NEURAL INFORMATION
PROCESSING SYSTEMS

Sample-Efficient and Safe Deep Reinforcement Learning via Reset Deep Ensemble Agents

Woojun Kim*[1] Yongjae Shin*[2] Jongeui Park[2] Youngchul Sung[2]

[1]Carnegie Mellon University [2]KAIST

Preliminaries

Primacy Bias

- DNN-based function approximators overfit early experiences, limiting their adaptability to later experiences [1].
- Primacy Bias is getting worse as we increase the replay ratio, which is the number of updates per time-step.

[1] Nikishin et al., “The primacy bias in deep reinforcement learning,” ICML 2022

Preliminaries

Primacy Bias

- DNN-based function approximators overfit early experiences, limiting their adaptability to later experiences [1].
- Primacy Bias is getting worse as we increase the replay ratio, which is the number of updates per time-step.

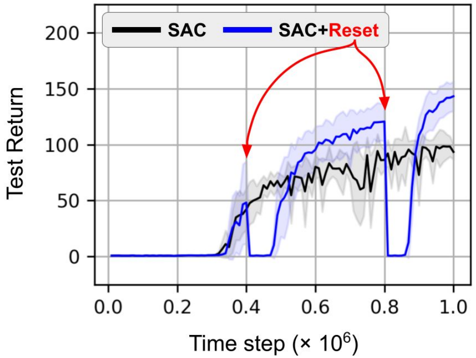


Figure 1. Test return on humanoid-run

Reset RL Agent

- (Nikishin et al 2022) proposed a simple reset method, which periodically resets a deep RL agent while preserving the replay buffer.
- Reset improves sample efficiency by allowing RL agents to increase the replay ratio.
- However, reset causes performance collapse after reset.
- Performance collapse leads to safety concerns, which restrict the use of the reset method in practical RL applications.

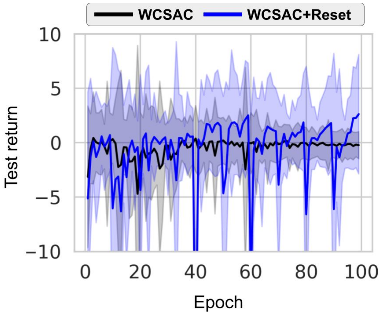


Figure 2. Test return on Safe RL.

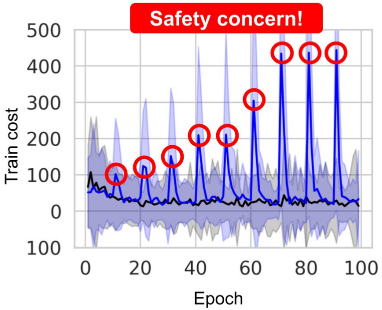


Figure 3. Train cost on Safe RL.

[1] Nikishin et al., "The primacy bias in deep reinforcement learning," ICML 2022

Our Approach: RDE

- Goal: preventing performance collapse and improving sample efficiency.
- RDE constructs **(1) N-ensemble** agent and **(2) sequentially reset** each ensemble agent and **(3) adaptively composite** N-ensemble agents into a single agent

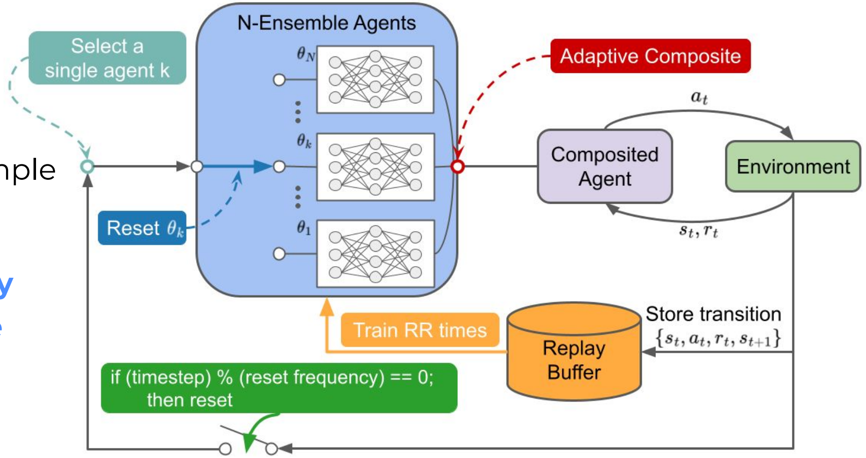
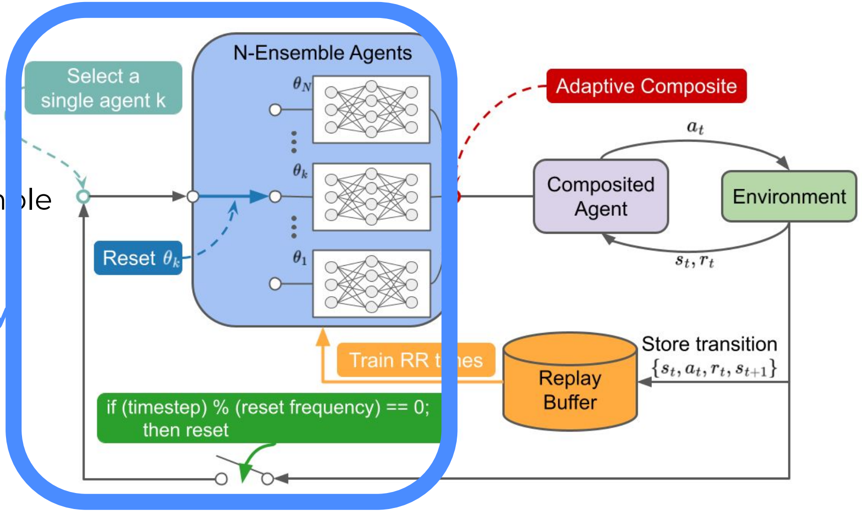


Figure 3. Overall diagram of RDE

Our Approach: RDE



- Goal: preventing performance collapse and improving sample efficiency.
- RDE constructs **(1) N-ensemble** agent and **(2) sequentially reset** each ensemble agent and **(3) adaptively composite** N-ensemble agents into a single agent

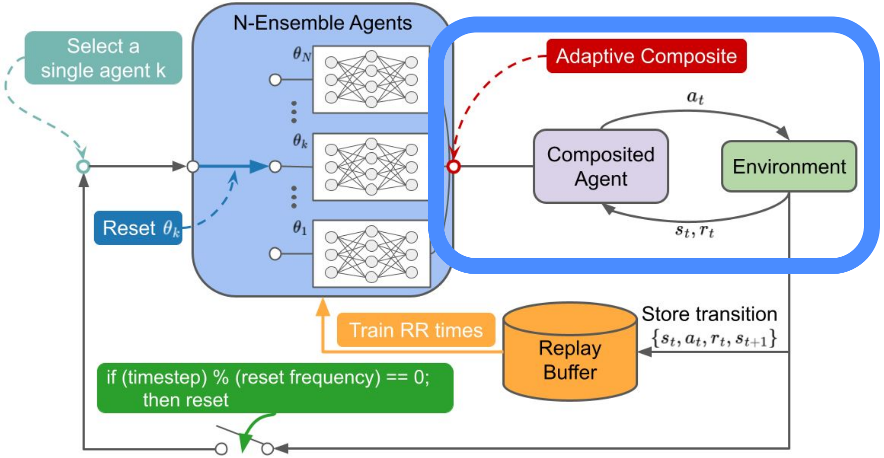
1. N-ensemble Agents

- Construct **N-ensemble agents** with different initial parameters.
- This diversity enhances robustness and efficiency.

2. Sequential Reset

- We **sequentially reset** parameter $\theta_1, \theta_2, \cdots, \theta_N$.
- Involving N-1 non-reset agents at each reset, prevents performance collapse.
- As we reset all ensemble agents, our method can still tackle the issue of primacy bias effectively.

Our Approach: RDE



- Goal: preventing performance collapse and improving sam efficiency.
- RDE constructs **(1) N-ensemble** agent and **(2) sequentially reset** each ensemble agent and **(3) adaptively composite** N-ensemble agents into a single agent

3. Adaptive Composition

- Recently resetted agent can still induce performance collapse.
- Propose **adaptive integration of N-ensemble agents** into a single agent.
- Assign a higher selection probability to the action with a higher action value, thereby reducing the chance that the most recently reset policy will be selected.

Eq. (1)

- The probabilities are calculated as $p_{select} = [p_1, p_2, \cdots, p_N] = \text{softmax}\left[\hat{Q}(s, a_1)/\alpha, \hat{Q}(s, a_2)/\alpha, \cdots, \hat{Q}(s, a_N)/\alpha\right]$ where $\alpha = \beta / \max(\hat{Q}(s, a_1), \hat{Q}(s, a_2), \cdots \hat{Q}(s, a_N))$ and $\hat{Q}$ is the action-value function of the earliest-reset agent among the ensemble.

Experiments

Environment / Base algorithm

- (Continuous) DeepMind Control Suite / SAC
- (Discrete) Minigrid, Atari 100k / DQN

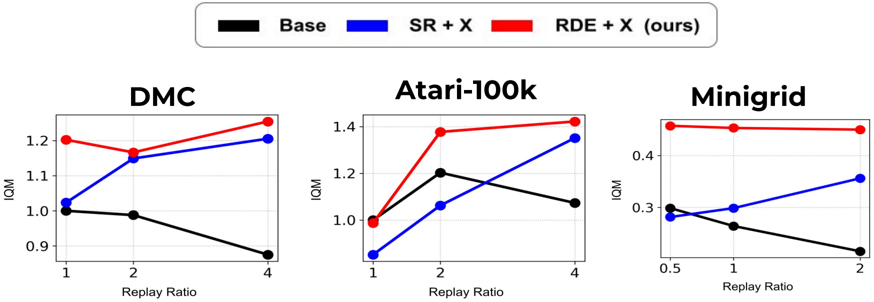


Figure 4. IQM results of the considered algorithms

- RDE outperforms both baselines (base algorithm and vanilla reset(SR)).

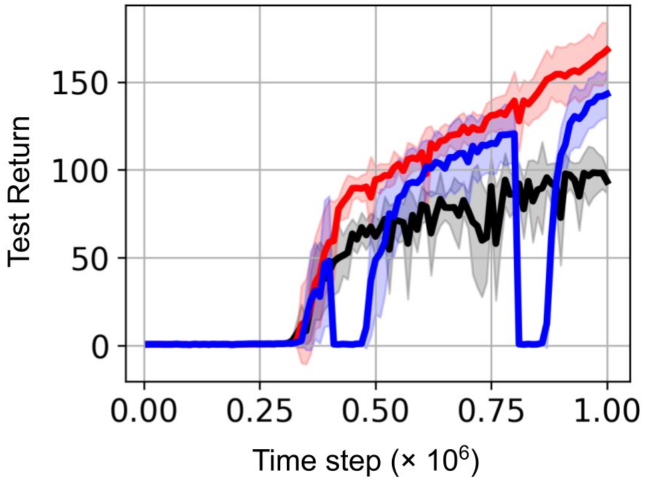


Figure 5. Test return on humanoid-run

- RDE prevents performance collapse and improves the final performance

Our approach for Safe RL

- We apply our approach to safe RL, which aims to maximize reward while minimizing costs including safety constraint.
- Adaptive composition based reward and cost functions.
- The adapted probability is defined as $p_{select}^{safe} = \kappa p_{select} + (1-\kappa)p_{select}^{c}$, where κ is the mixing coefficient and p_{select}^{c} is given by:

$$p_{select}^{c} = [p_1^c, p_2^c, \cdots, p_N^c] = \text{softmax}\left[-\hat{C}(s, a_1)/\alpha_c, -\hat{C}(s, a_2)/\alpha_c, \cdots, -\hat{C}(s, a_N)/\alpha_c\right]$$

where C denote cost function and $\alpha_c = \beta / \max\{|\hat{C}(s, a_1)|, |\hat{C}(s, a_2)|, \cdots, |\hat{C}(s, a_N)|\}$.

Experiments: Safe RL

- RDE not only achieved superior test performance to the baselines but also reduced training safety cost.

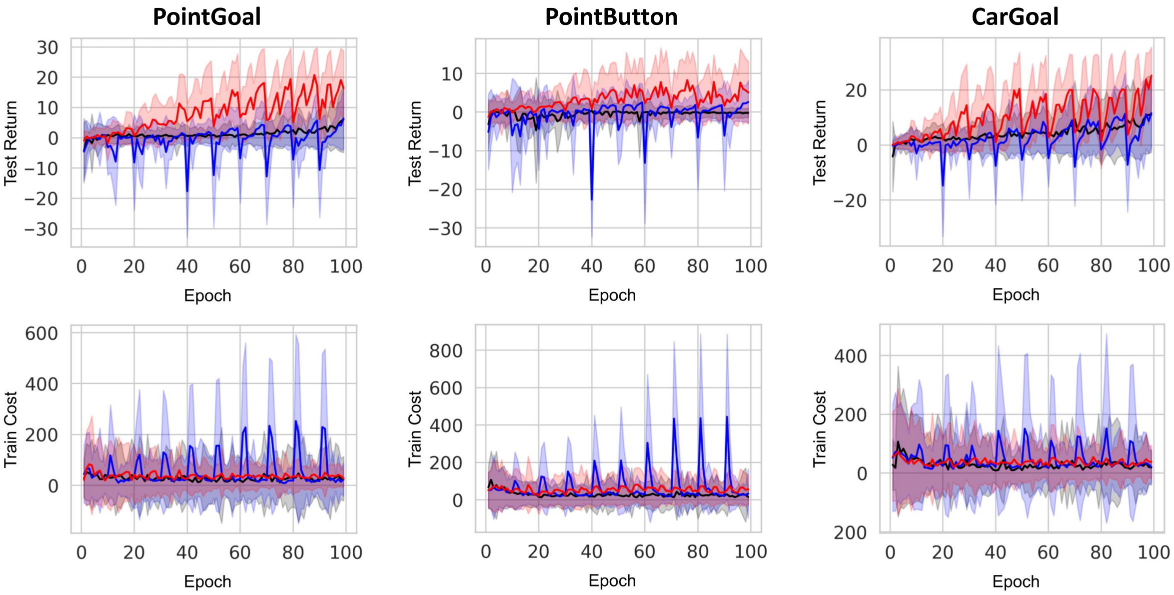


Figure 6. Test return & Train cost on Safe RL

Thank You!