Dropout Q-Functions for Doubly Efficient Reinforcement Learning

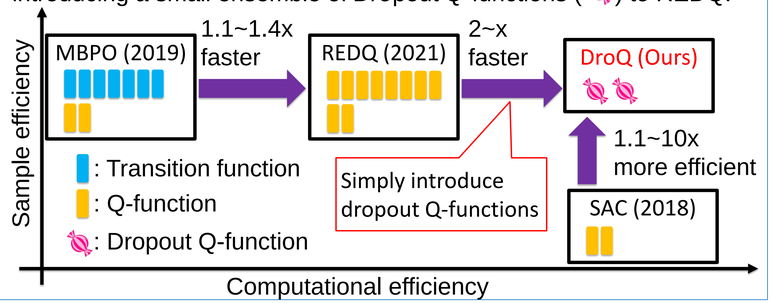
Takuya Hiraoka [1,2], Takahisa Imagawa [2], Taisei Hashimoto [2,3], Takashi Onishi [1,2], and Yoshimasa Tsuruoka [2,3]
[1] NEC Corporation, [2] The National Institute of Advanced Industrial Science and Technology, [3] The University of Tokyo

NEC-AIST AI Cooperative Research Laboratory
ICLR

1. Introduction:

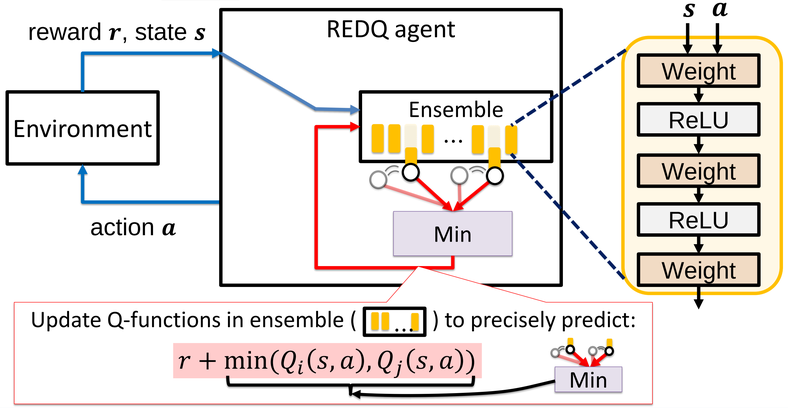
- In general, RL methods that are not only **sample efficient** but also **computational efficient** (i.e., **doubly efficient**) are preferable.

- We propose DroQ, a simple but doubly efficient RL method, by introducing a small ensemble of *Dro*pout *Q*-functions ( ) to REDQ.



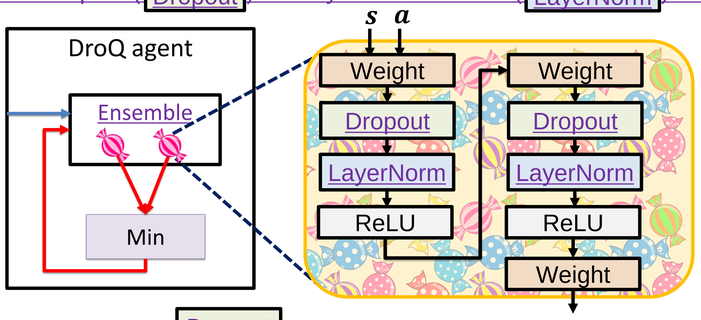
2. Randomized Ensembled Double Q-Learning (REDQ):

- REDQ (Chen, 2021) is a sample-efficient RL method equipped with **high update-to-data (UTD) ratio** and **randomized ensemble**.
- **High UTD ratio:** number of Q updates (→) per environment interaction (→) is high (e.g., 20 updates per interaction).
- **Randomized ensemble:** a randomly selected subset ( ) of ensemble ( ) is used at the target (Min) in the Q update (→).

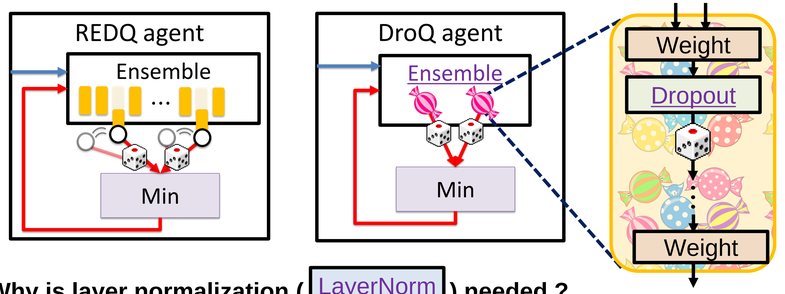


3. DroQ, the proposed method:

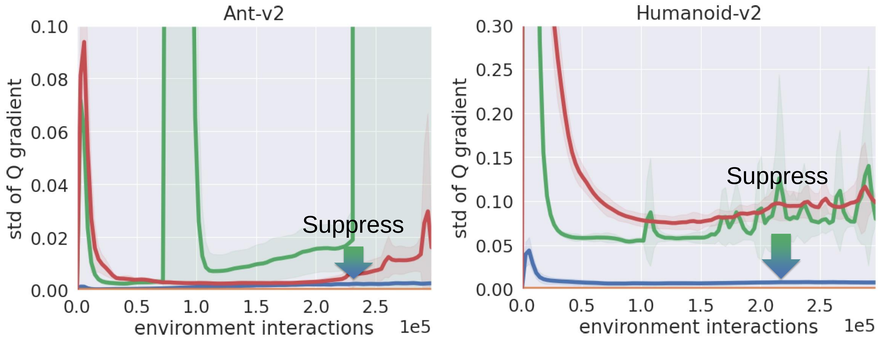
- DroQ is a REDQ variant using a small ensemble of dropout Q-functions ( ) in which dropout (Dropout) and layer normalization (LayerNorm) are used.



- **Q. Why is dropout (Dropout) needed ?**
  A. To inject Q-function uncertainty ( ) to the target (Min), similarly to REDQ.



- **Q. Why is layer normalization (LayerNorm) needed ?**
  A. To suppress ( ) the learning instability caused by dropout.

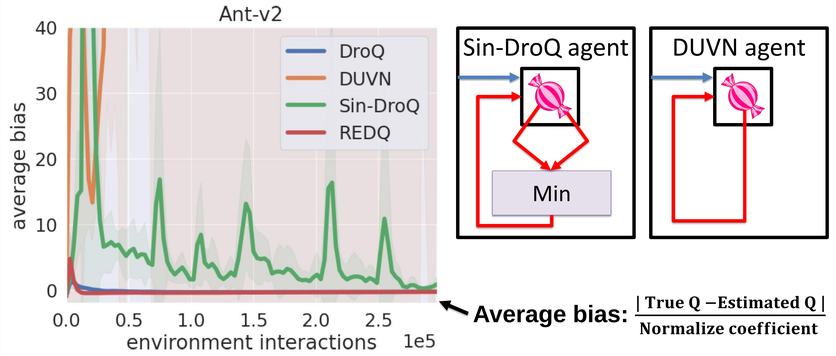


The standard deviation of the gradient of Q-loss w.r.t. parameters

DroQ w/o Dropout w/o LayerNorm w/o Dropout nor LayerNorm

3.5. DroQ, the proposed method (Contd.):

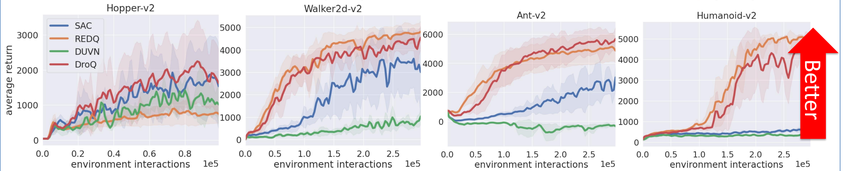
- **Q. Why is a small ensemble ( ) needed ?**
  (Why not use a single dropout Q-function ( ) alone ?)
  A. Using it ( ) alone induces a large bias in Q-estimation.



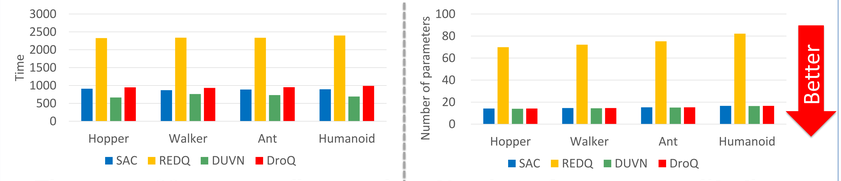
Average bias: $\frac{|\text{True Q} - \text{Estimated Q}|}{\text{Normalize coefficient}}$

4. Experiments:

- **Q. How sample-efficient is DroQ ( )?**
  A. Much better than SAC ( ) and almost the same as REDQ ( ).



- **Q. How computationally efficient is DroQ?**
  A. Much better than REDQ and almost the same as SAC.



Times per 20 x → + → (in msec)

Number of parameters (/1e4)

5. Conclusion:

- DroQ (REDQ +) is simple but doubly efficient.

Our source code is available at