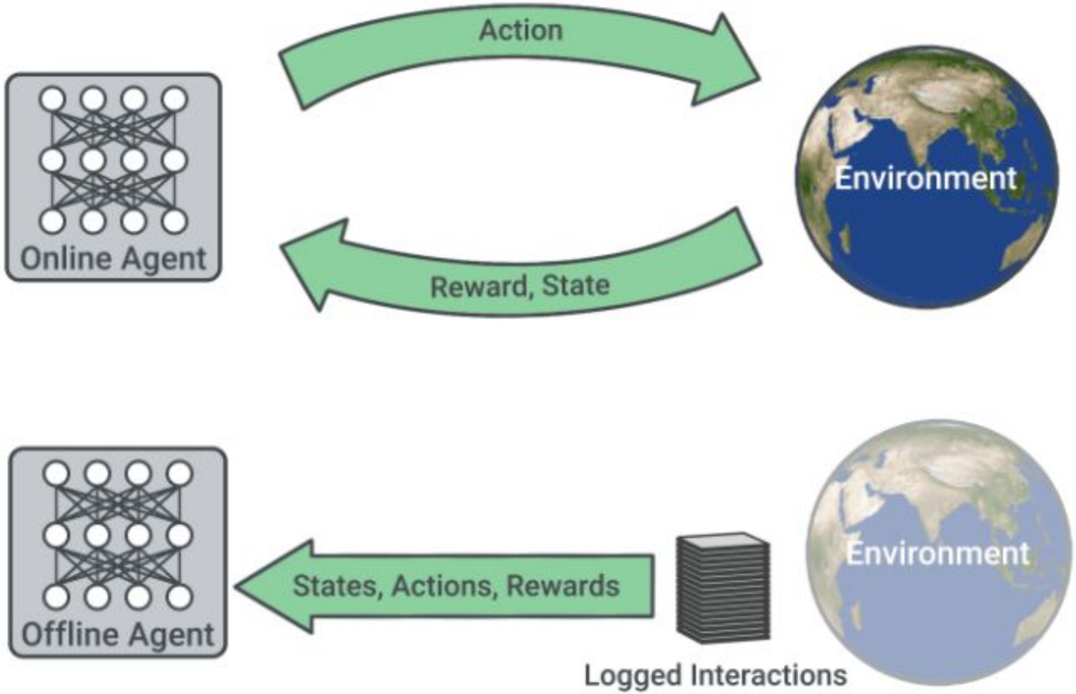
The role of dataset generation in Offline RL

Analyzing the performance gap between [Agarwal et al., 2020] & [Fujimoto et al, 2019]

Kajetan Schweighofer k01556273

Supervisors:
Marius-Constantin Dinu
Vihang Patil

What is Offline Reinforcement Learning?



Modified from [Agarwal et. al, 2020]

Why is it important?

No need for costly/impossible interaction with environment

Data for many critical task already logged

- E.g. medical records, financial data, driving, …

Appealing due to success of Deep Learning

Increase sample efficiency for off-policy DRL algorithms

- Even experience replay does not leverage all past data, but sliding window

Related work

Benchmarking the performance of simple off-policy algorithms on Atari environment

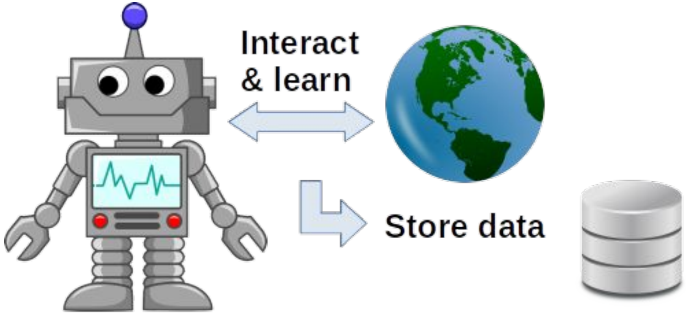
1. [Agarwal et. al, 2020]: An optimistic perspective on Offline Reinforcement Learning
   - Report very good results
2. [Fujimoto et. al, 2019]: Benchmarking Batch Deep Reinforcement Learning Algorithms
   - Report mostly poor results

Both generate Offline Dataset by an online DQN behavioural policy

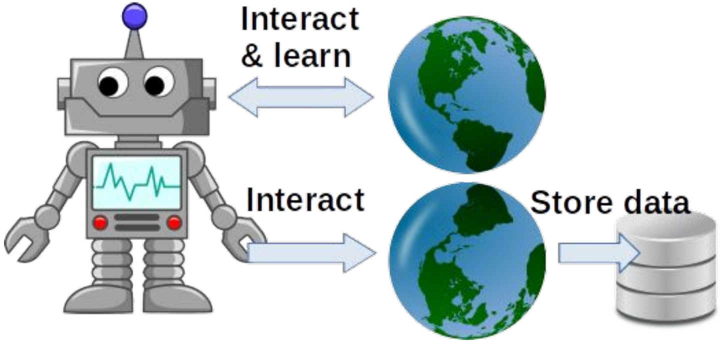
Dataset generation

Recent publications mainly use one of two dataset generation strategies:

1. Store full behavioural experience replay buffer



2. Generate other trajectories by behavioural policy
    - Can be final or intermediate policy



Scope of practical work

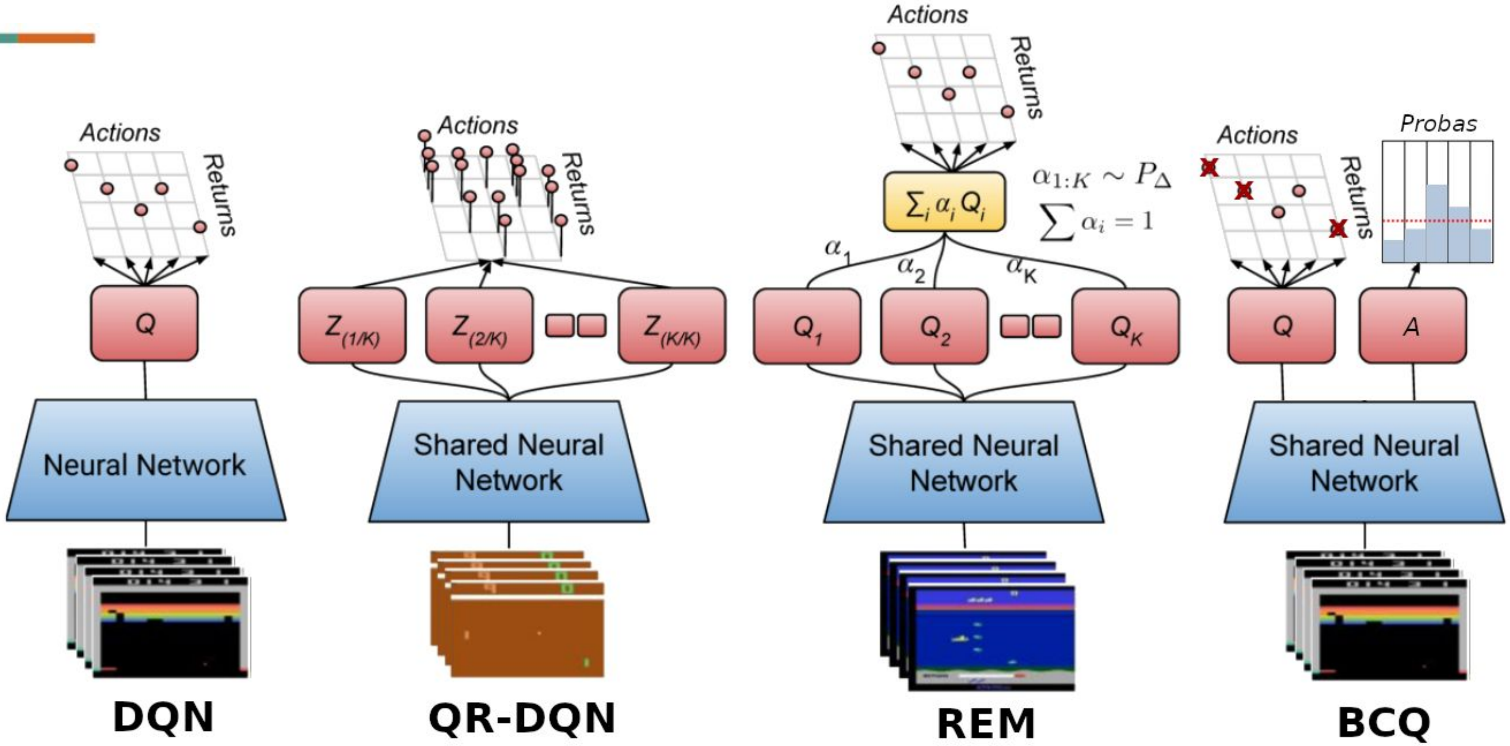
Test both dataset generation techniques, ablation study w.r.t. number of generating policies

- Atari game “Breakout”

Implement off-policy (deep Q-learning) algorithms in Pytorch

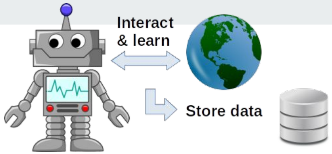
- Deep Q Network (**DQN**) [Mnih et. al, 2013]
- Quantile Regression DQN (**QR-DQN**) [Dabney et. al, 2017]
- Random Ensemble Mixture (**REM**) [Agarwal et. al, 2020]
- Batch Constrained deep Q-learning (**BCQ**) [Fujimoto et. al, 2019]
    - Designed to be trained in an offline paradigm

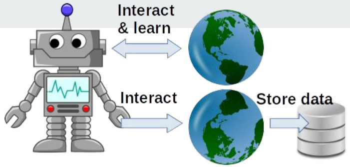
Algorithms



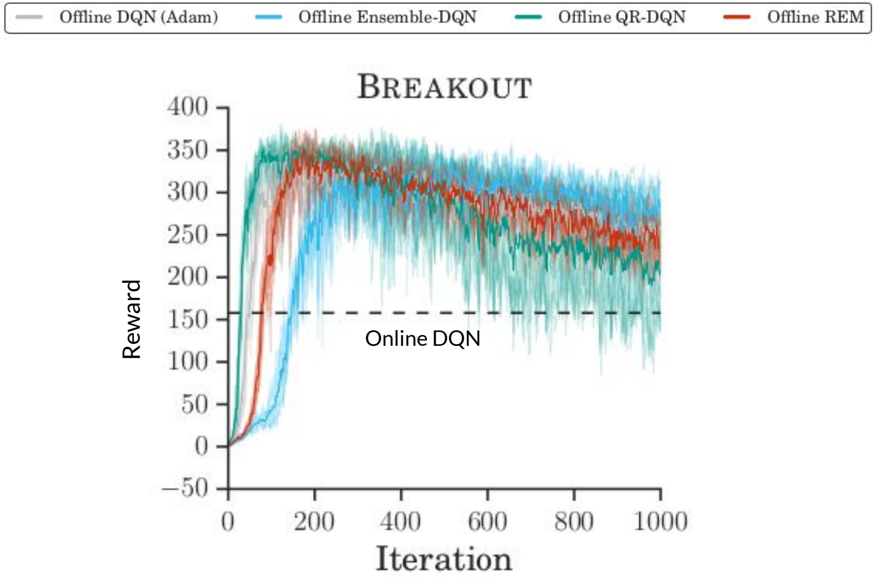
Modified from [Agarwal et. al, 2020]

Results in Papers



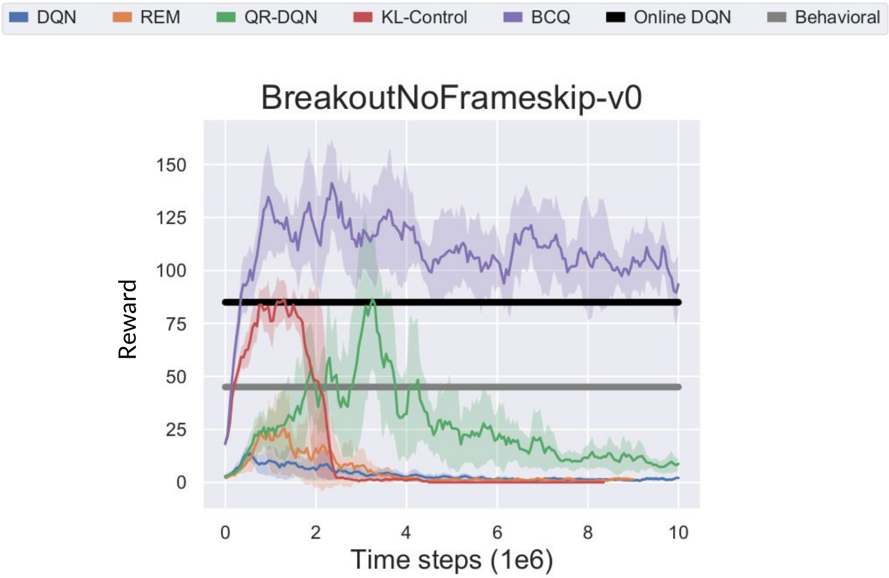


50 million transitions, 12.5 million policies (buffer)



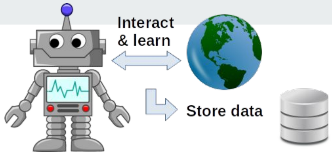
[Agarwal et. al, 2020]

10 million transitions, one final policy

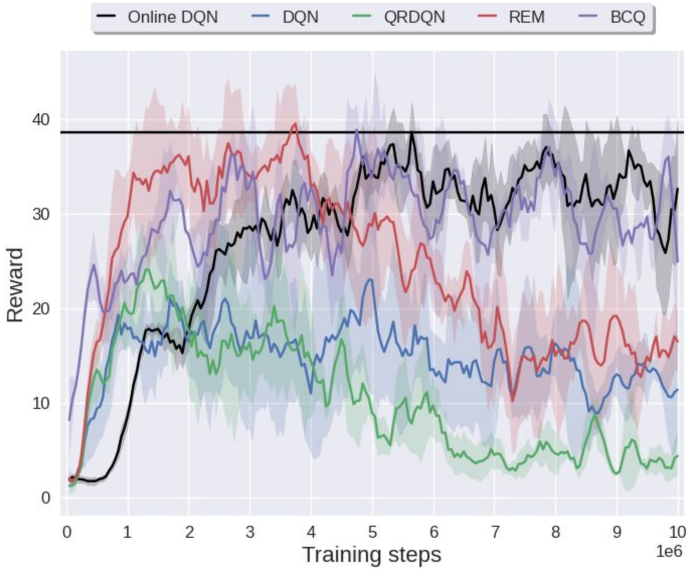


[Fujimoto et. al, 2020]

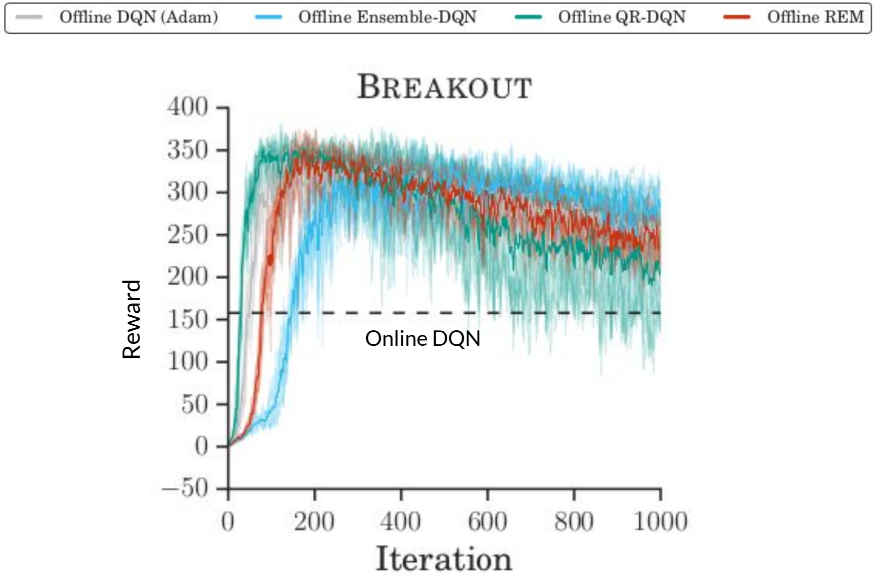
Performance on behavioural buffer



10 million transitions, 2.5 million policies (buffer)

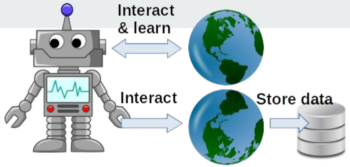


50 million transitions, 12.5 million policies



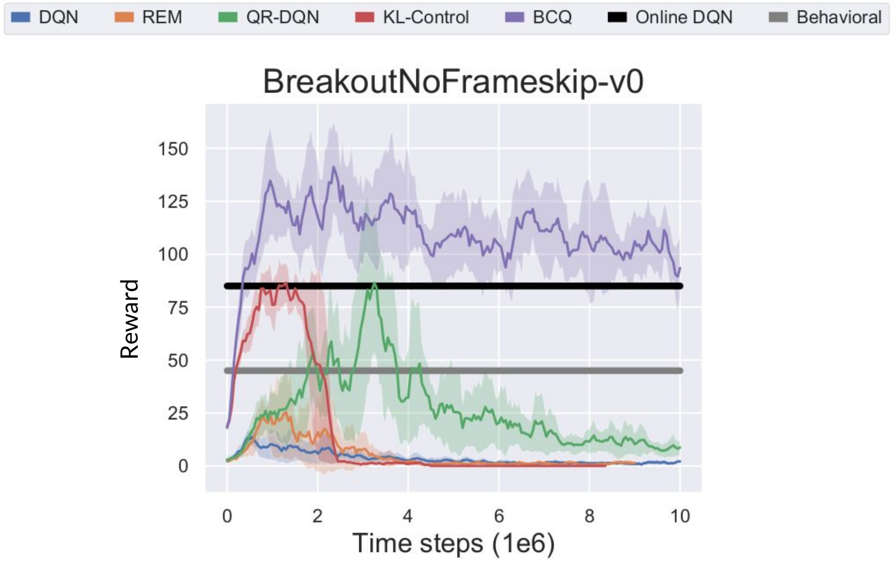
[Agarwal et. al, 2020]

Performance for one final policy



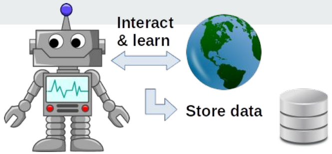
10 million transitions, one final policy

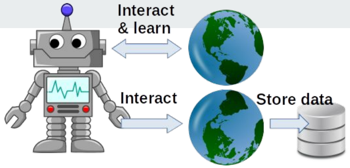
10 million transitions, one final policy



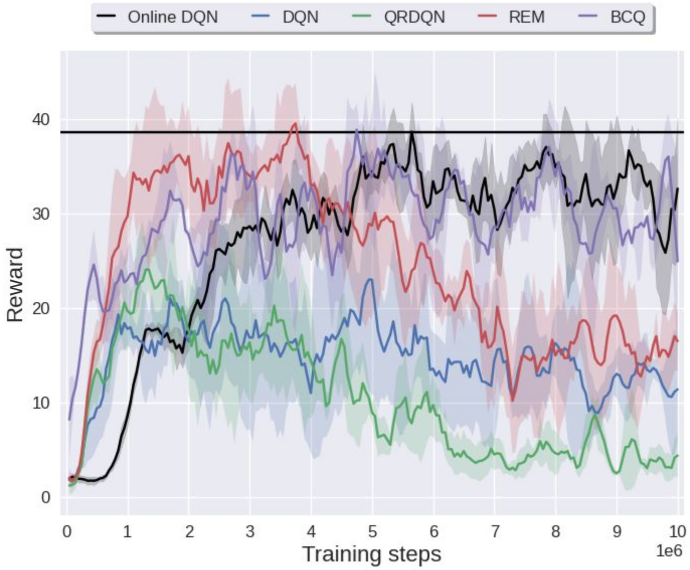
[Fujimoto et. al, 2020]

Performance for 2.5 m policies



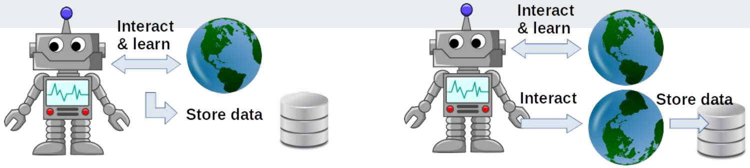


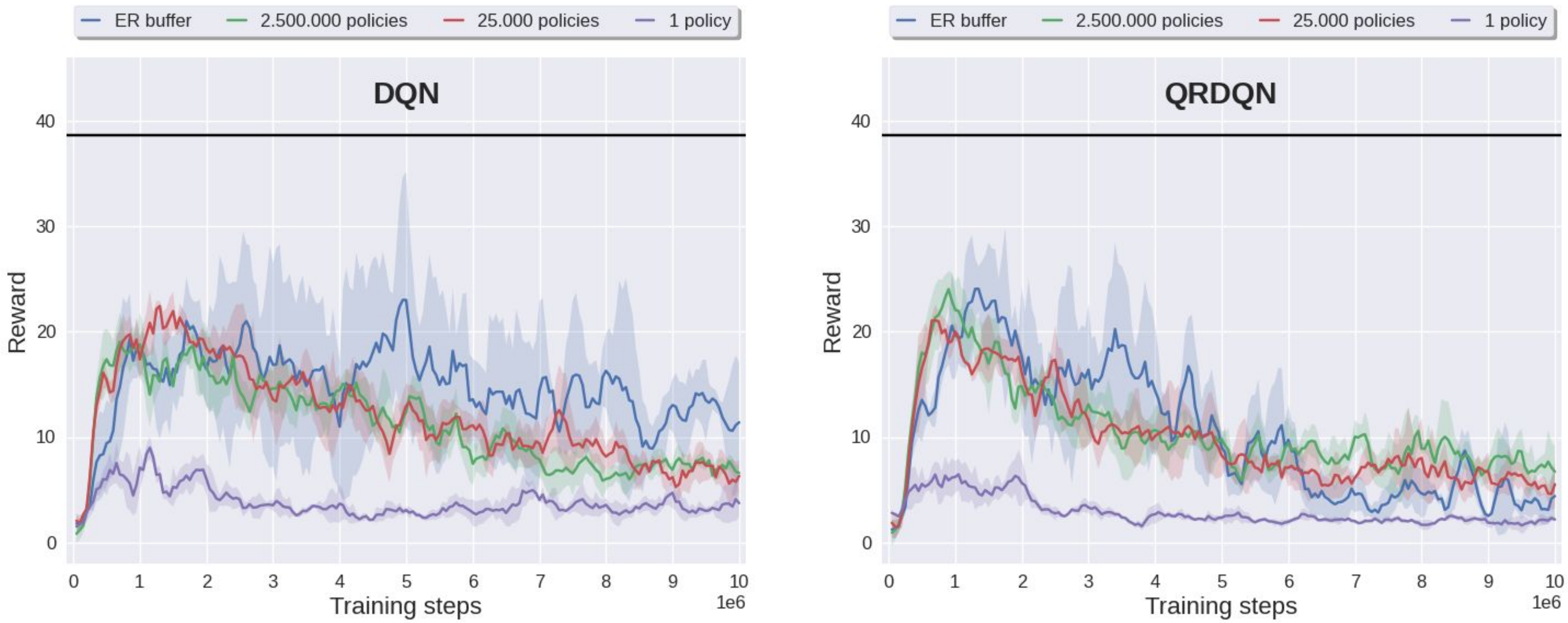
10 million transitions, 2.5 million policies (buffer)



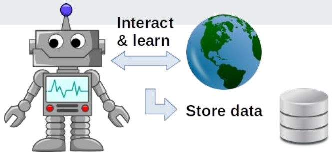
10 million transitions, 2.5 million interm. policies

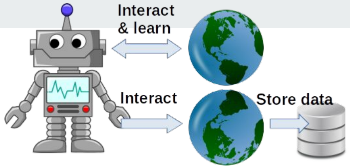
Ablation study

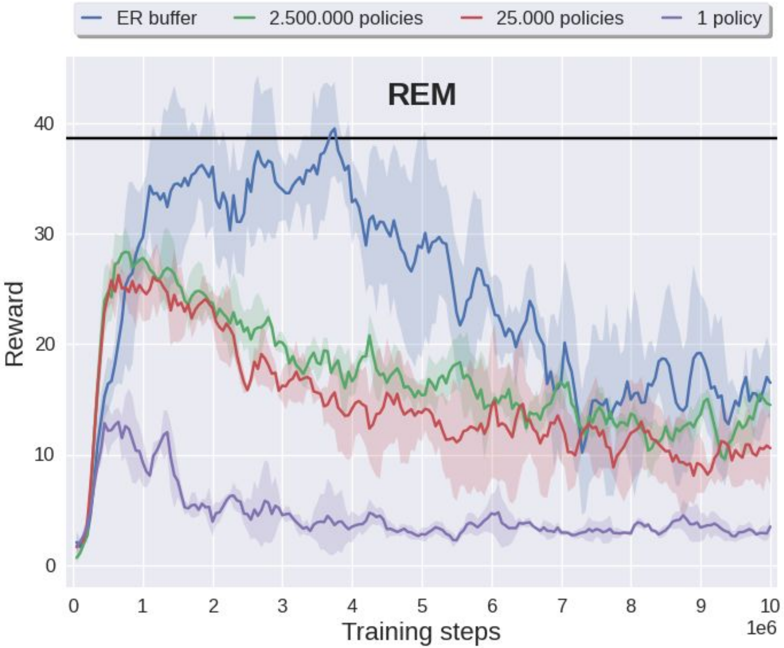


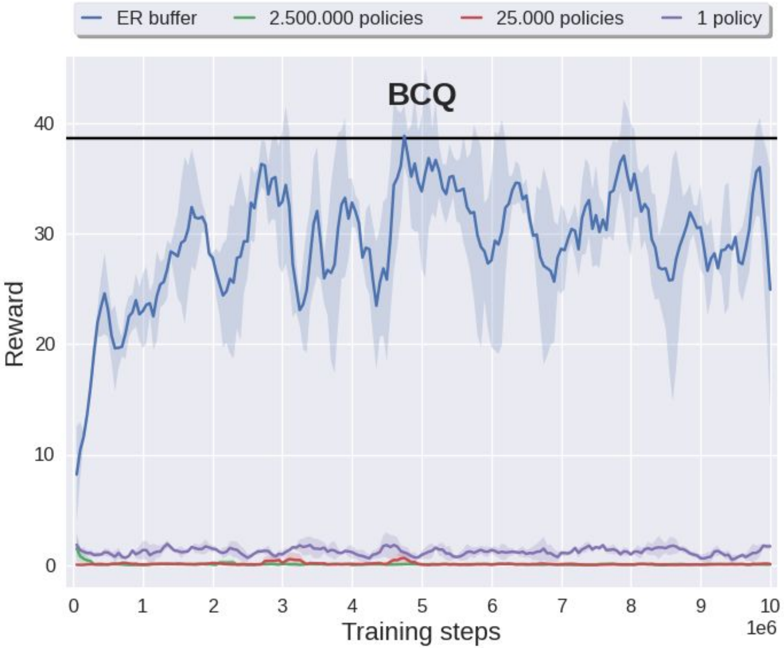


Ablation study

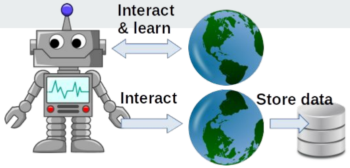






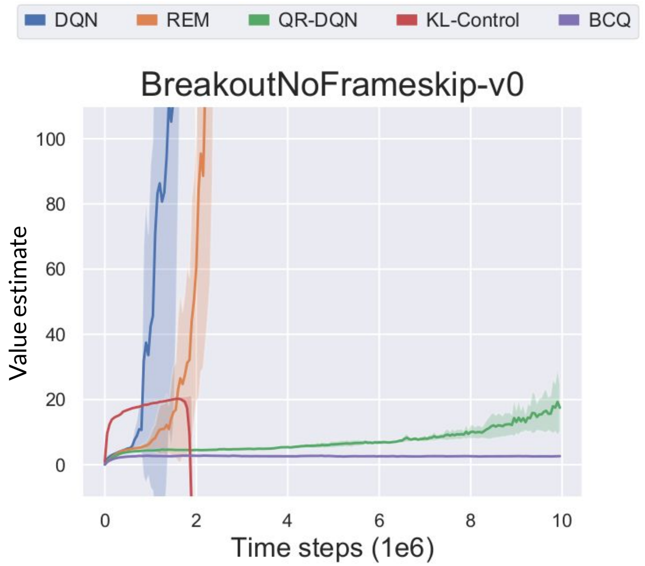


No diverging value estimates



No divergence in value estimates (one final policy)

[Fujimoto et. al, 2019] reported diverging values



[Fujimoto et. al, 2020]

Issues & reflection

Hard to implement the task that was solved in the papers

- Many custom wrappers for Atari, most follow [Machado et. al, 2017]

Slight differences like optimizers, initialization, activation functions, gradient clipping, …

High level frameworks (Dopamine) hard to grasp

Network architectures seldomly revisited to meet advances in the field

Expectations are mostly met

Future work

Investigate issue for BCQ

Find reason for stable value estimates

Try other, easier environments

Test different hyperparameters for offline algorithms

- hyperparameters are selected “online”

Limit test dataset generation

References

[Agarwal et. al, 2020] Agarwal, R., Schuurmans, D., and Norouzi, M. (2020). An Optimistic perspective on offline reinforcement learning

[Fujimoto et. al, 2019] Fujimoto, S., Conti, E., Ghavamzadeh, M., and Pineau, J. (2019) Benchmarking batch deep reinforcement learning algorithms

[Mnih et. al., 2013] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning.

[Dabney et. al., 2017] Dabney, W., Rowland, M., Bellemare, M. G., and Munos, R. (2017). Distributional reinforcement learning with quantile regression.

[Machado et. al, 2017] Machado, M., Bellemare, M., Talvitie, E., Veness, J., Hausknecht M., and Bowling, M. (2017) Revisiting the arcade learning environment

Questions?