Revisiting Fundamentals of Experience Replay

William Fedus, Prajit Ramachandran*, **Rishabh Agarwal**,*
Yoshua Bengio, Hugo Larochelle, Mark Rowland, Will Dabney

ICML
International Conference On Machine Learning

Slides adapted from William Fedus

Learning algorithm and *data generation* linked -- but relation poorly understood.

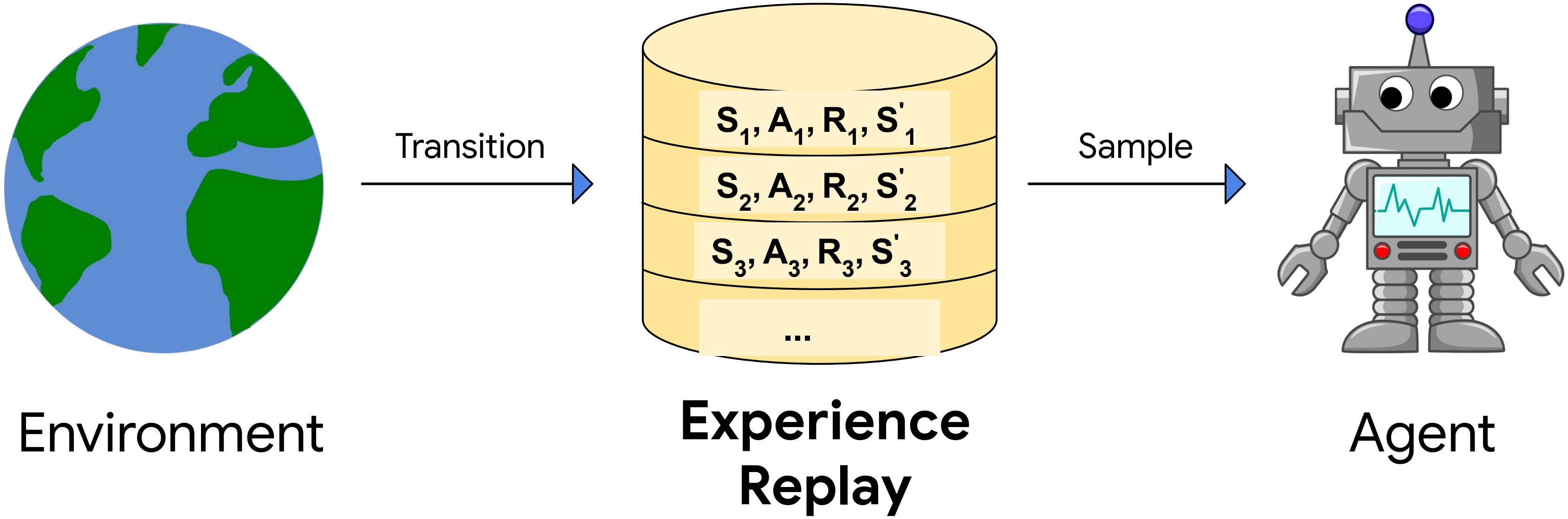
Our work empirically probes this *interplay*.

Source of learning algorithm: Rainbow 🌈

Data generation mechanism: Experience replay

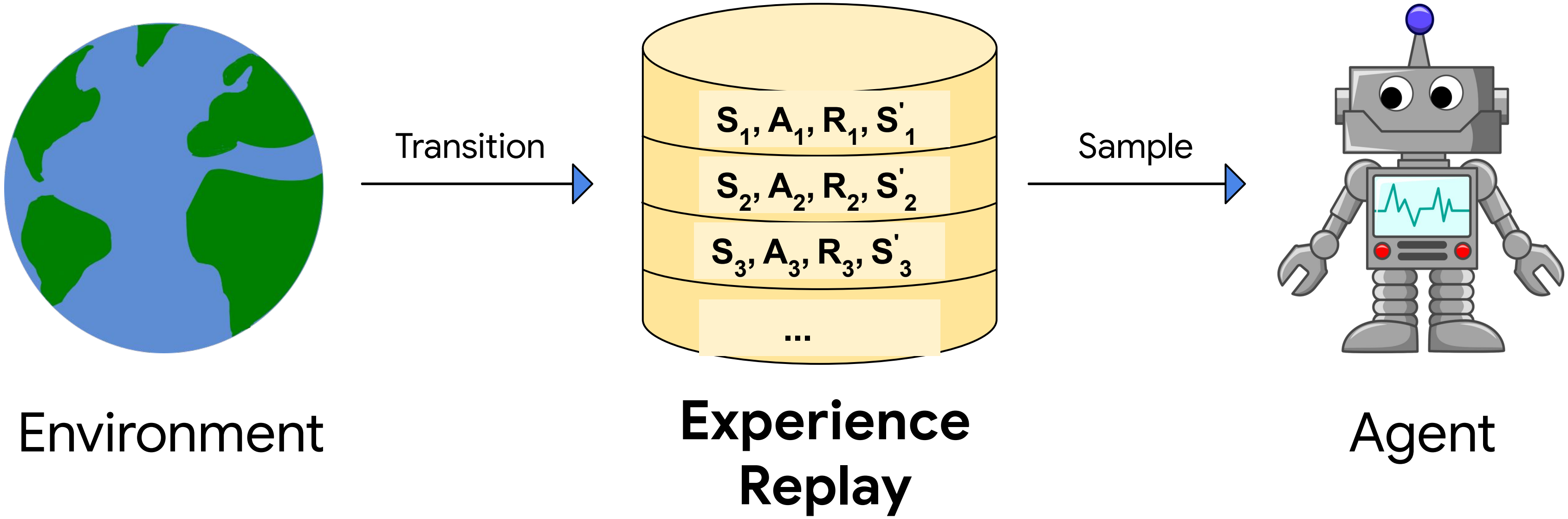
Hessel, Matteo, et al. "Rainbow: Combining improvements in deep reinforcement learning." AAAI, 2018.

Experience Replay in Deep RL



Fixed-size buffer of the most recent transitions collected by the policy.

Experience Replay in Deep RL



Improves sample efficiency and decorrelates samples.

The Learning Algorithm

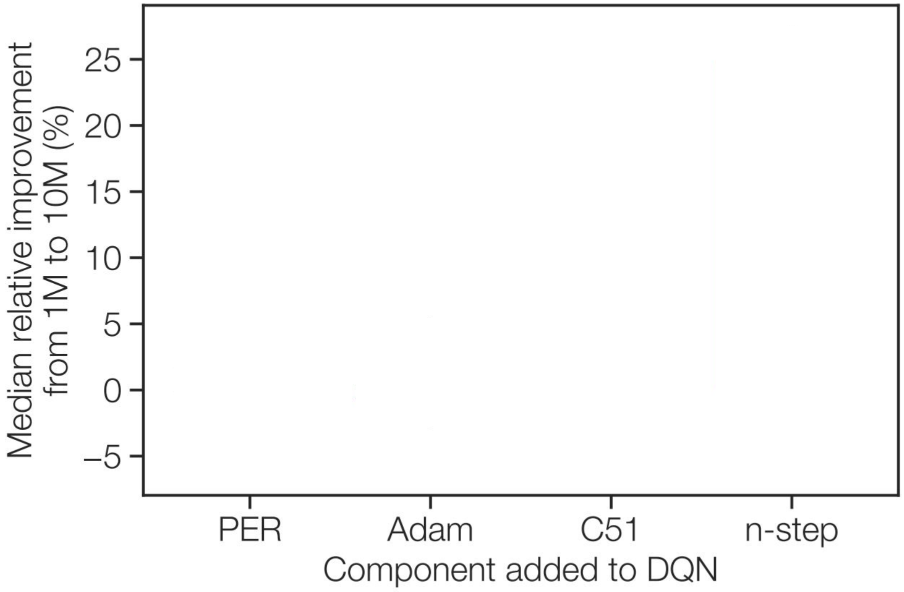
Rainbow agent is the kitchen sink of RL algorithms. Starting with DQN, add:

1. **Prioritized replay**: Preferentially sample high TD-error experience
2. **n-step returns**: Use *n* future rewards rather than single reward
3. **Adam**: Improved first-order gradient optimizer
4. **C51**: Predict the *distribution* over future returns, rather than expected value

Schaul et al., 2015; Watkins, 1989; Kingma and Ba, 2014; Bellemare et al., 2017

Learning Algorithms Interaction with Experience Replay

Analysis: Add each Rainbow component to a DQN agent and measure performance while *increasing* replay capacity.



TL;DR

Experience replay and learning algorithms interact in surprising ways: ***n*-step returns** are uniquely crucial to take advantage of increased replay capacity.

From a theoretical basis, this may be surprising -- more analysis next.

Detailed Analysis

A Deeper Look at Experience Replay

Shangtong Zhang, Richard S. Sutton
Dept. of Computing Science
University of Alberta
{shangtong.zhang, rsutton}@ualberta.ca

Smaller and larger replay capacities hurt -- don’t touch it!

An Optimistic Perspective on Offline Reinforcement Learning

Rishabh Agarwal[1] Dale Schuurmans[1 2] Mohammad Norouzi[1]

Recent RL methods work well even with extremely large replay buffers!

Two Independent Factors of Experience Replay

1. How *large* is the replay capacity?
2. **What is the *oldest policy* in the replay buffer?**

Defining a Replay Ratio

The *replay ratio* is the number of gradient updates per environment step. This controls how much experience is trained on before being discarded.

		Replay Capacity				
		100,000	316,228	**1,000,000**	3,162,278	10,000,000
Oldest Policy	25,000,000	250.000	79.057	25.000	7.906	2.500
	2,500,000	25.000	7.906	2.500	0.791	**0.250**
	250,000	2.500	0.791	**0.250**	0.079	0.025
	25,000	**0.250**	0.079	0.025	0.008	0.003

Defining a Replay Ratio

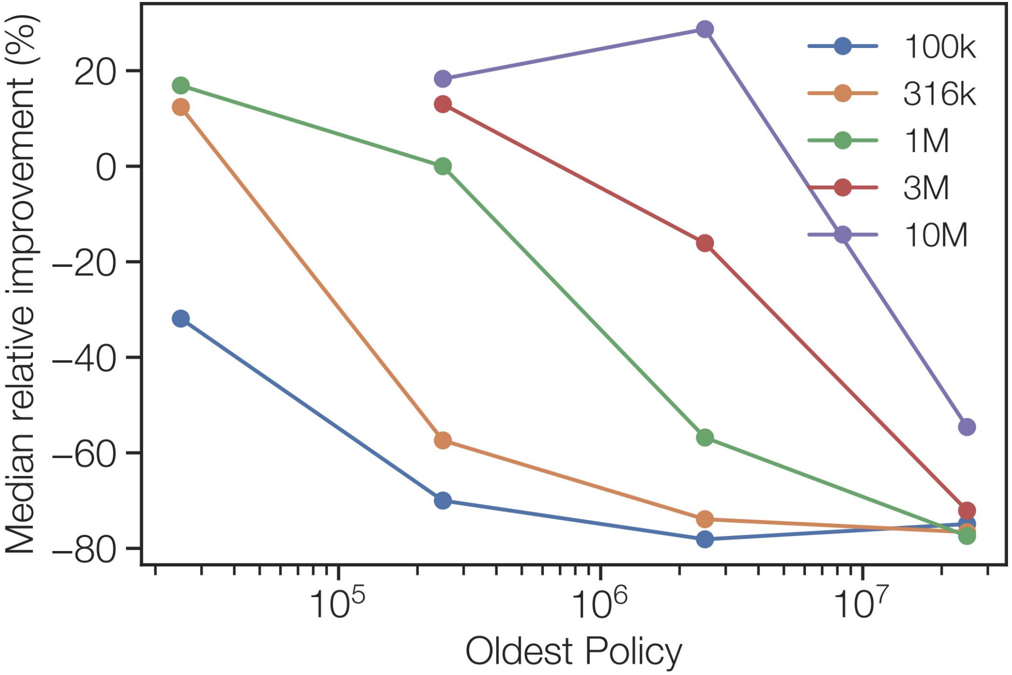
The *replay ratio* is the number of gradient updates per environment step.

1 env step / 250 gradient updates

400 env step / 1 gradient update

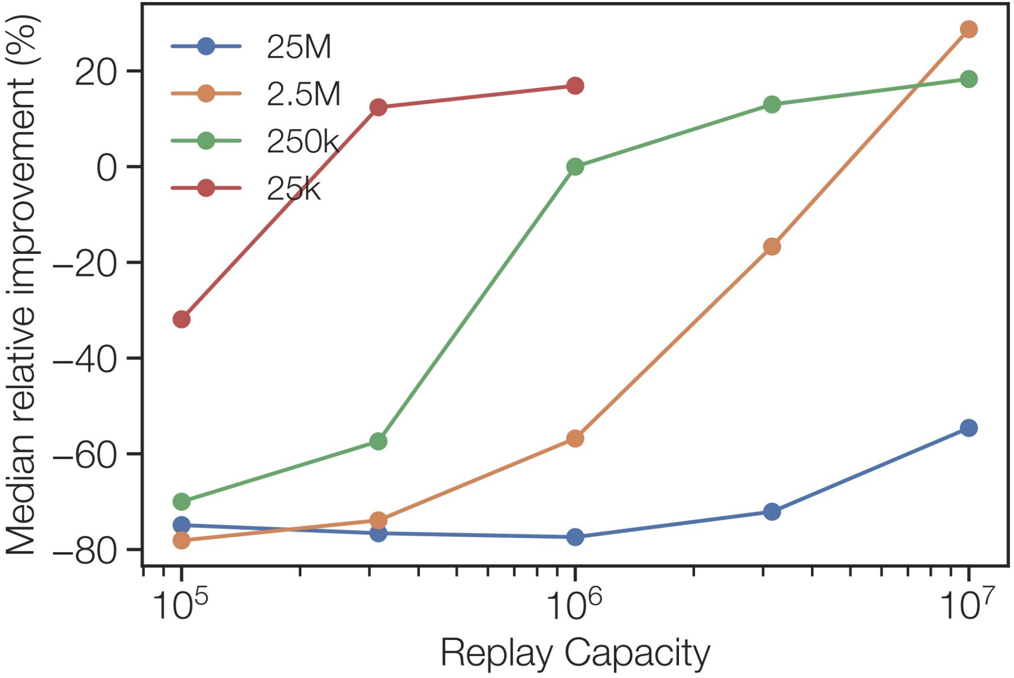
		Replay Capacity				
		100,000	316,228	**1,000,000**	3,162,278	10,000,000
Oldest Policy	25,000,000	250.000	79.057	25.000	7.906	2.500
	2,500,000	25.000	7.906	2.500	0.791	**0.250**
	250,000	2.500	0.791	**0.250**	0.079	0.025
	25,000	**0.250**	0.079	0.025	0.008	0.003

Rainbow Performance as we Vary Oldest Policy



On policy to Off-policy --->

Rainbow Performance as we Vary Capacity



Larger Buffers -->

Reduce to the Base DQN Agent

Rainbow benefits with larger memory, does DQN? Increase the replay capacity of a DQN agent (1M -> 10M). Control for *replay ratio* or the *oldest policy* in buffer.

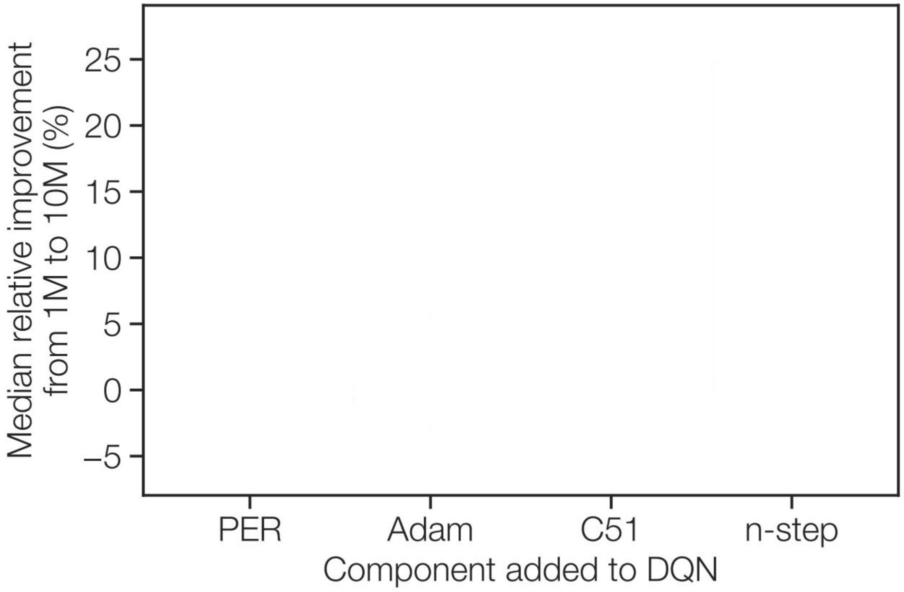
Agent	Fixed replay ratio	Fixed oldest policy
DQN	+0.1%	-0.4%
Rainbow	+28.7%	+18.3%

Two *learning algorithms* with two very different outcomes. What causes this gap?

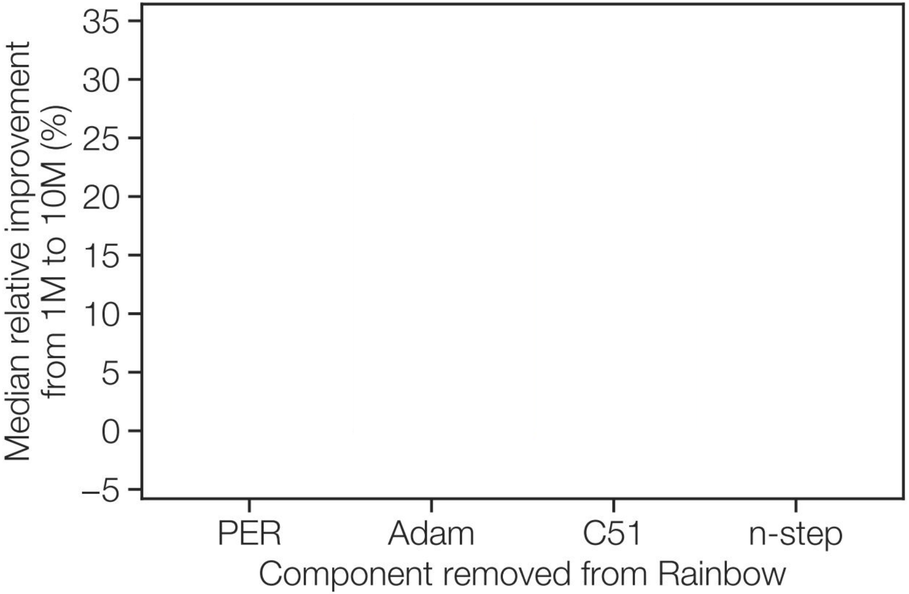
DQN Additive Analysis

DQN does *not* benefit when increasing the replay capacity while Rainbow does.

Analysis: Add each Rainbow component to DQN and measure performance while increasing replay capacity.



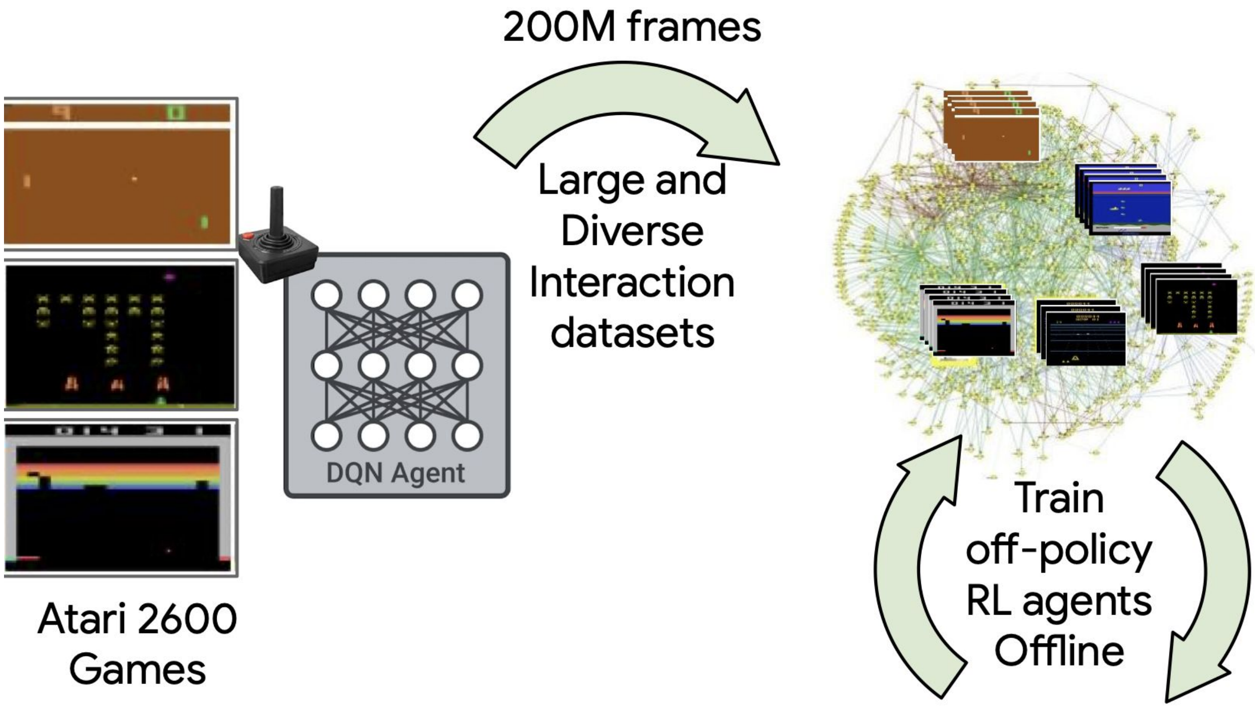
Rainbow Ablative Experiment



Experiment: Ablate each Rainbow component and measure performance while increasing replay capacity.

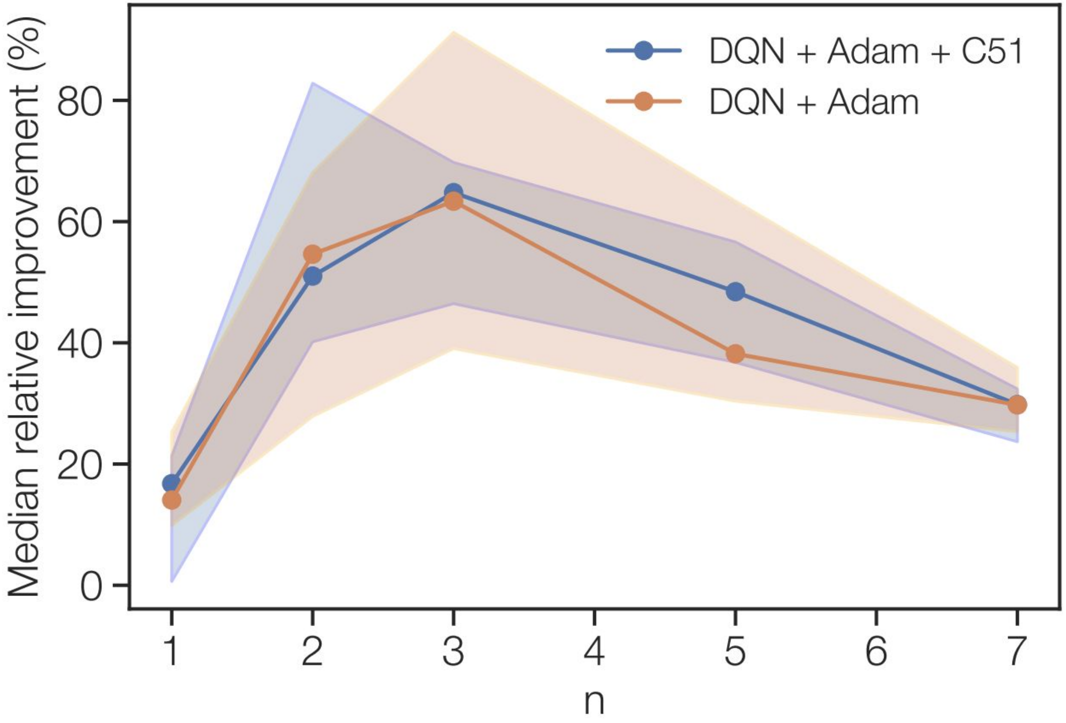
Empirical result: *n*-step returns are *important* in determining whether Q-learning will benefit from larger replay capacity.

Offline Reinforcement Learning



Agarwal et al. "An optimistic perspective on offline reinforcement learning." ICML (2020).

n-step Returns Beneficial in Offline RL



Theoretical Gap

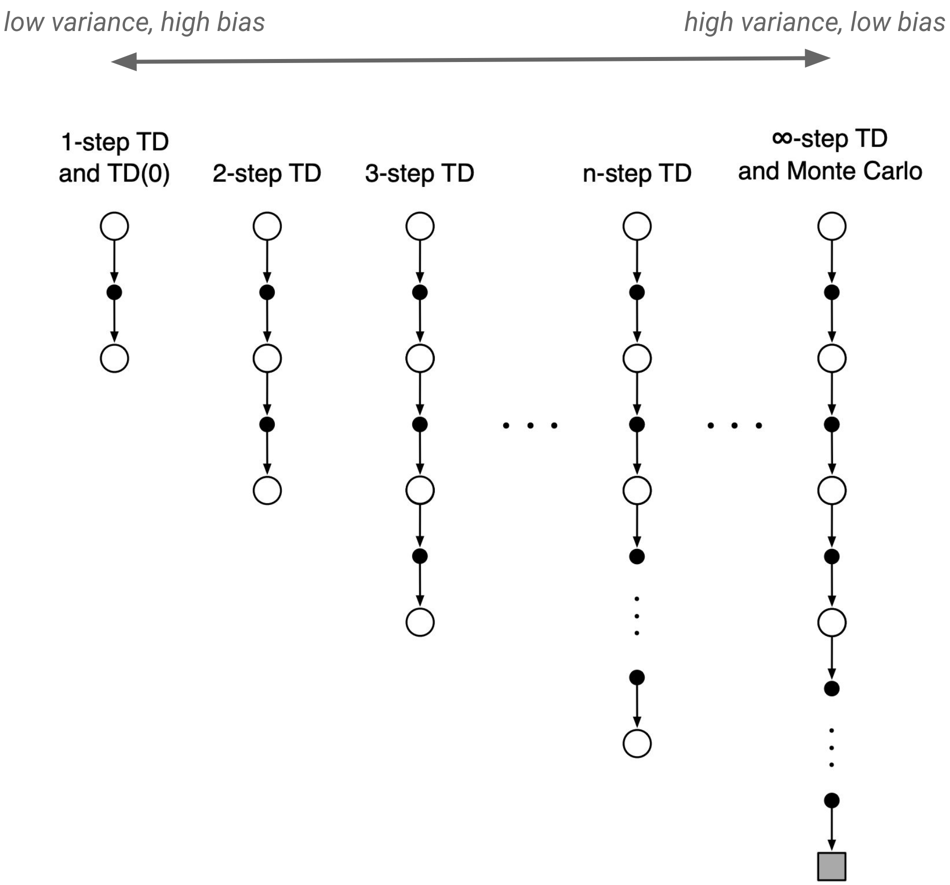
Uncorrected n-step returns are mathematically wrong in off-policy learning,

- We use *n*-step experience from past behavior policies, *b*
- But we learn the value for a policy, *π*

Common solution is to use techniques like importance sampling, Tree Backups or more recent work like Retrace (Munos et al., 2016) *n*-step methods interpolate between Temporal Difference (TD)- and Monte Carlo (MC) -learning.

Classic *bias-variance* tradeoff.

Figure from Sutton and Barto, 1998; 2008



n-step returns benefit from low bias, but suffer from high variance in *learning target*.

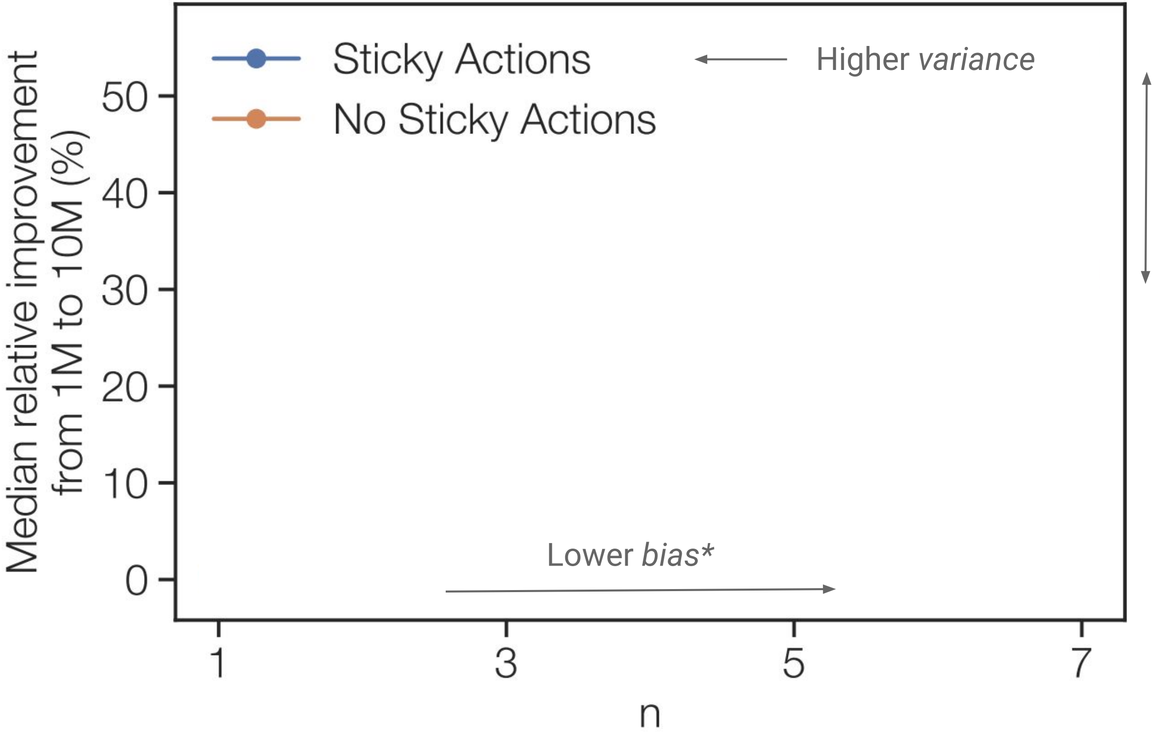
Hypothesis: the larger replay capacity decreases the value estimate *variance*.

Experiment: Toggle env randomness via *sticky actions*.

Hypothesis: *n*-step benefit should be eliminated or reduced in a *deterministic* environment.

Sticky actions -- Machado et al., 2017

Bias-Variance Effects in Experience Replay



Deterministic environments (orange) benefit less from larger capacity since these do not have as much variance to reduce

In Summary

Our analysis upends conventional wisdom: larger buffers are very important, provided one uses *n*-step returns.

We uncover a bias-variance tradeoff arising between *n*-step returns and replay capacity.

n-step returns still yield performance improvements, even in the infinite replay capacity setting (offline RL).

We point out a theoretical gap in our understanding.

Rainbow Interaction with Experience Replay Aspects

Replay Capacity

Oldest Policy		100,000	316,228	**1,000,000**	3,162,278	10,000,000
	25,000,000	-74.9	-76.6	-77.4	-72.1	-54.6
	2,500,000	-78.1	-73.9	-56.8	-16.7	**28.7**
	250,000	-70.0	-57.4	**0.0**	13.0	18.3
	25,000	**-31.9**	12.4	16.9	--	--

The *easiest* gain in deep RL? Change replay capacity from 1M to 10M.

Rainbow Interaction with Experience Replay Aspects

		Replay Capacity				
		100,000	316,228	**1,000,000**	3,162,278	10,000,000
Oldest Policy	25,000,000	-74.9	-76.6	-77.4	-72.1	-54.6
	2,500,000	-78.1	-73.9	-56.8	-16.7	**28.7**
	250,000	-70.0	-57.4	**0.0**	13.0	18.3
	25,000	**-31.9**	12.4	16.9	--	--

Significant aberration from the trend. Due to exploration issues.

An Idea to Test This Hypothesis

Consider the value estimate for a state *s*.

If the environment is deterministic, a single *n*-step rollout provides a 0-variance estimate.

We would expect no benefit of more samples from this state *s* and therefore diminished benefit of a larger replay buffer.

Deep Reinforcement Learning

1. Learning algorithm

   DQN, Rainbow, PPO

2. Function approximator

   MLP, conv. net, RNN

3. Data generation mechanism

   Experience replay, prioritized experience replay

Rainbow Performance as we Vary Capacity

Performance improves with capacity

Replay Capacity

Oldest Policy	100,000	316,228	**1,000,000**	3,162,278	10,000,000
25,000,000					
2,500,000					
250,000	-70.0	-57.4	**0.0**	13.0	18.3
25,000					

Rainbow Performance as we Vary Oldest Policy

More “on-policy” data
improves performance

Replay Capacity

Oldest Policy	100,000	316,228	**1,000,000**	3,162,278	10,000,000
25,000,000			-77.4		
2,500,000			-56.8		
250,000			**0.0**		
25,000			16.9		