Averaged-DQN: Variance Reduction and Stabilization for Deep Reinforcement Learning

Oron Anschel

Joint work Nir Baram, and Nahum Shimkin

Department of Electrical Engineering
Technion - Israel Institute of Technology

Technion Israel Institute of Technology

Overview

1 Background & Motivation

2 Averaged-DQN

3 Overestimation and Approximation Errors

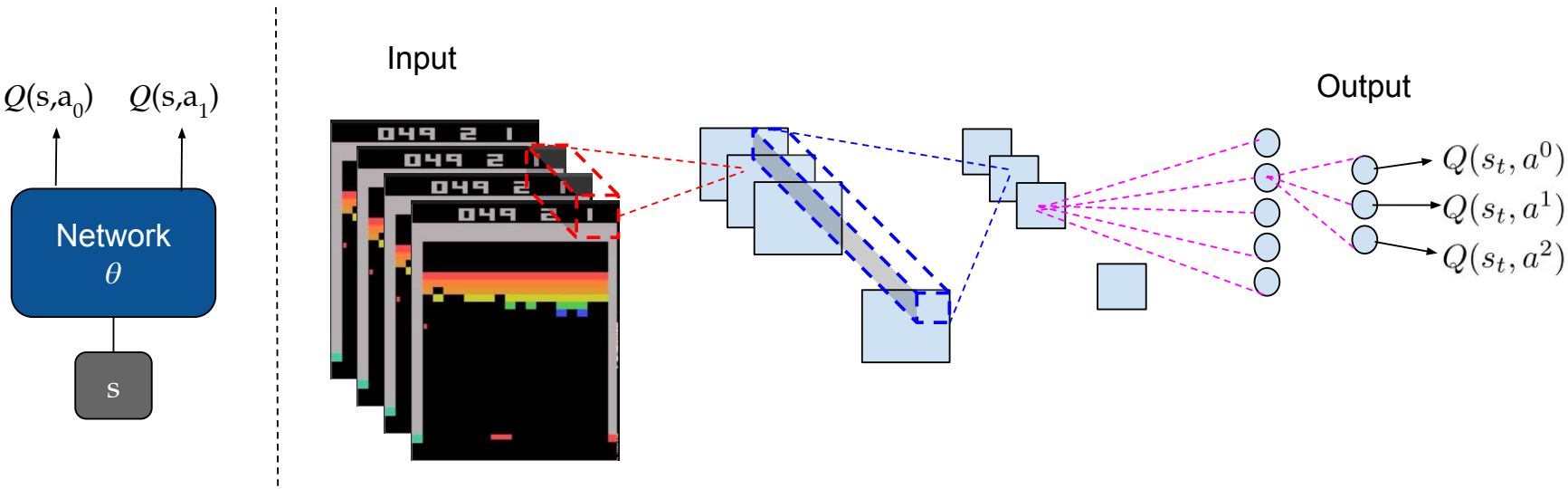
4 TAE Variance Reduction

5 Experiments

Technion Israel Institute of Technology

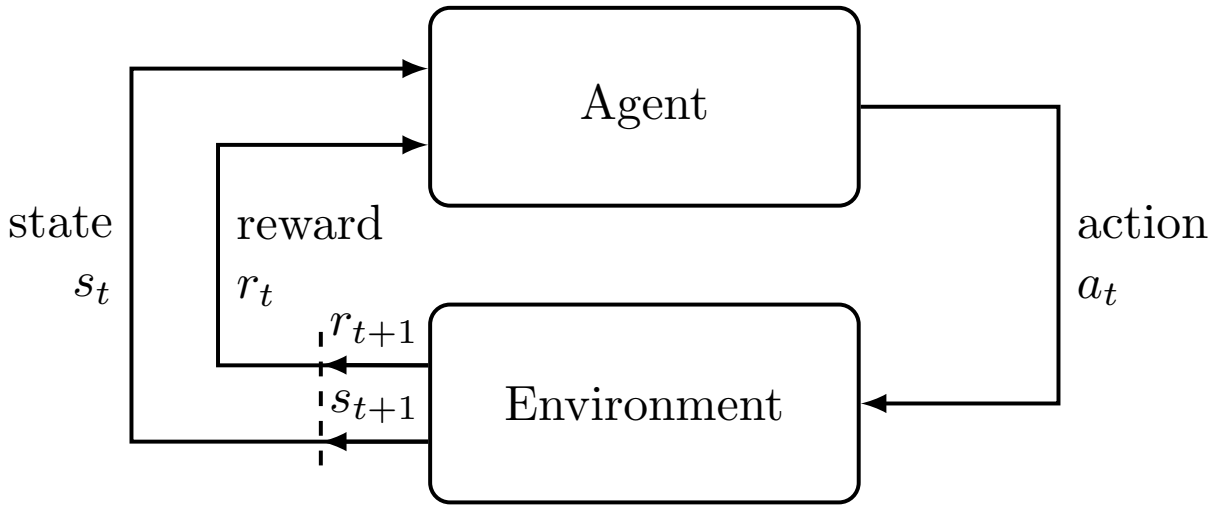
DQN

- Mnih, et al., (2013)
  - Target Network.
  - Experience Buffer (Lin et al., 1993).



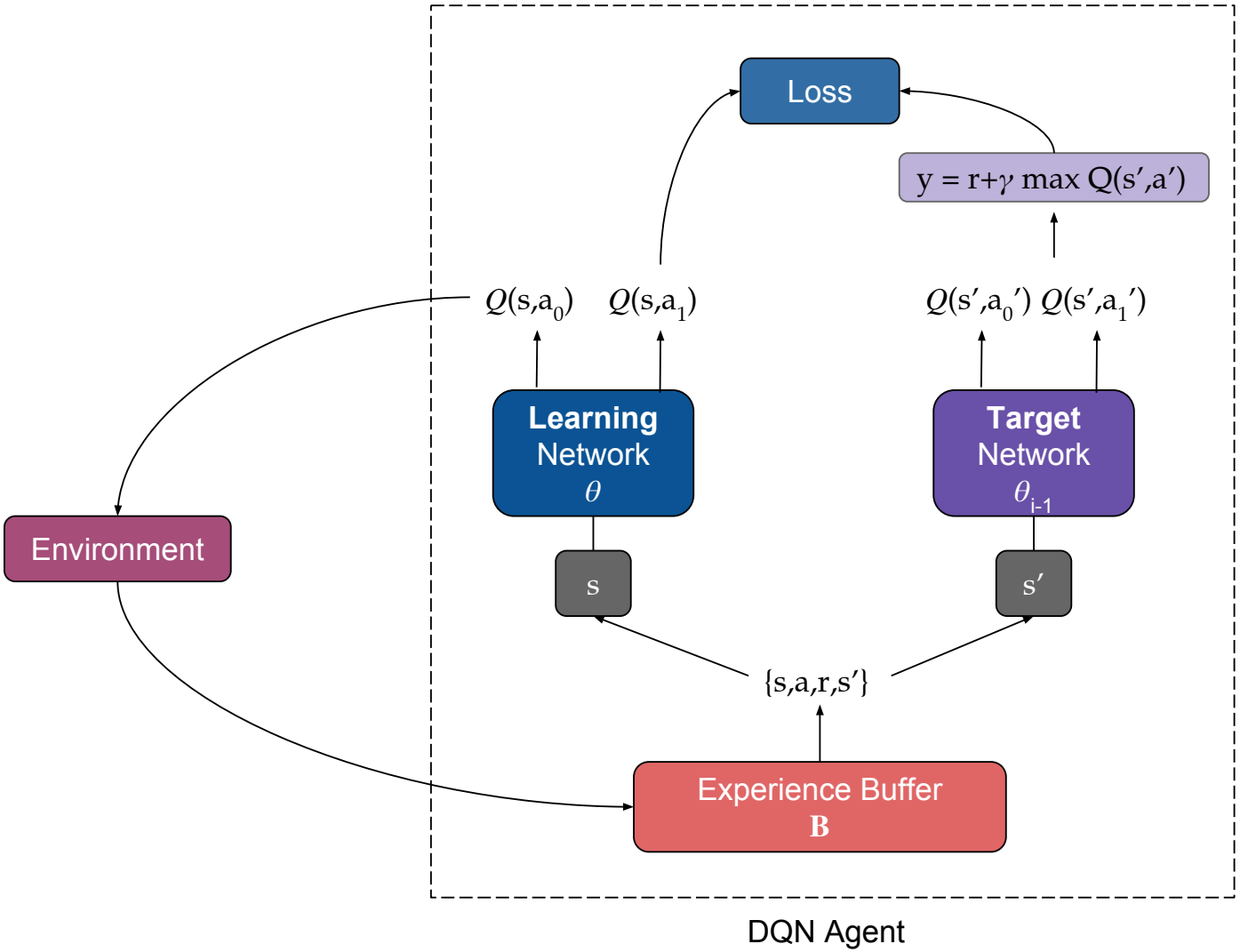
Reinforcement Learning

The **Agent-Environment** Interface

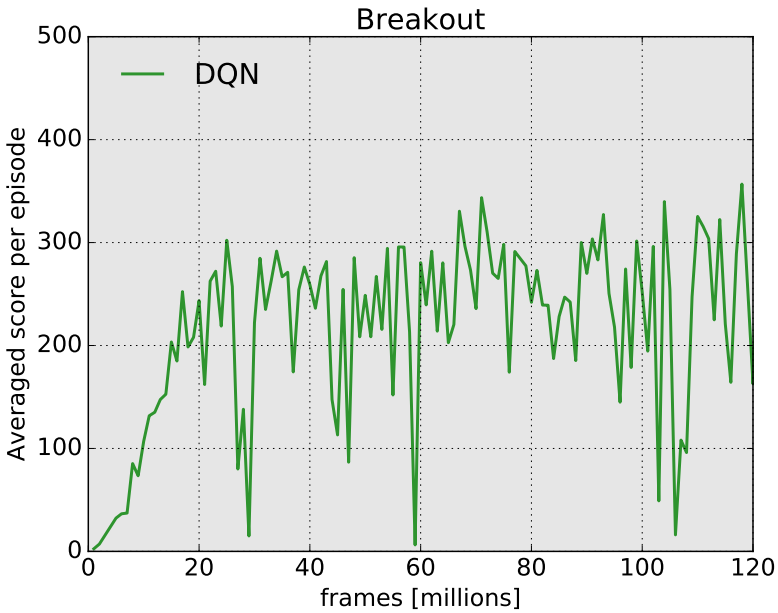


DQN Agent- Environment Interface

The **Agent-Environment** Interface



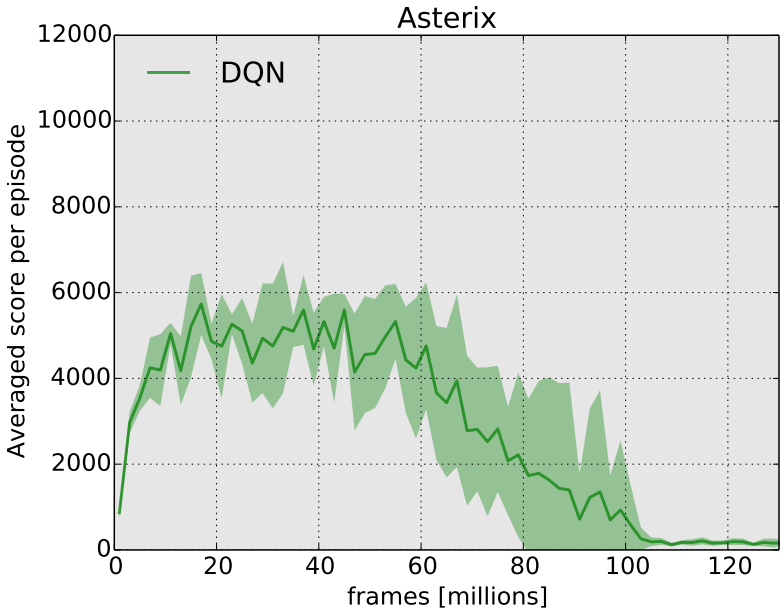
Motivational Example 1



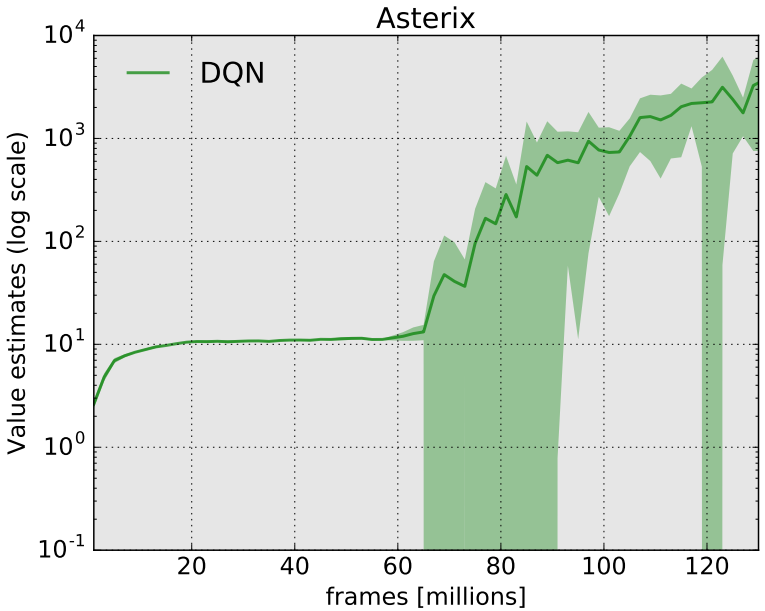
Example 1:
Instability and Variability

- Upside: Super-human conrol.
- Downside: **Best model** evaluation methodology.

Motivational Example 2



(average score)



(value estimate)

Example 2:
Overestimation[1]

[1] H van Hasselt et al., 2015

- Overestimation $\rightarrow$ asymptotically sub-optimal policies.

Motivation Cont'd

- Are the two examples related?

Overview

1 Background & Motivation

2 Averaged-DQN

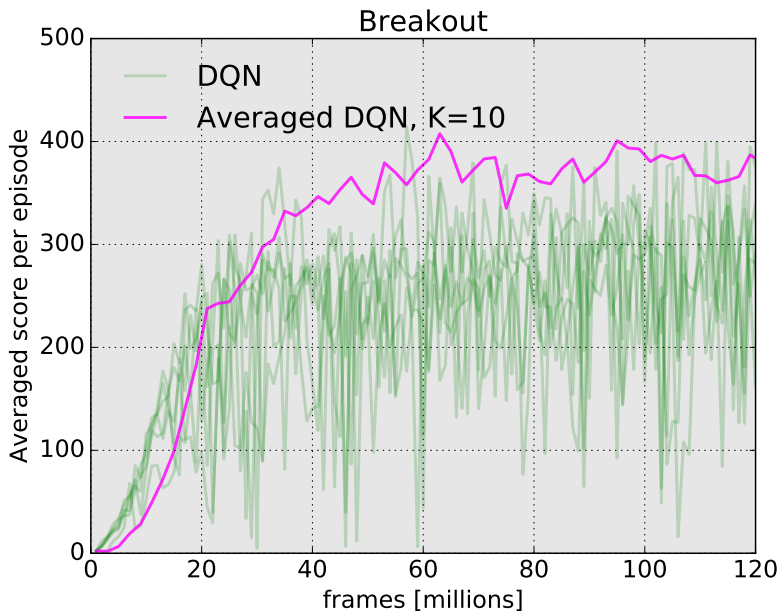
3 Overestimation and Approximation Errors

4 TAE Variance Reduction

5 Experiments

Technion Israel Institute of Technology

Motivational Example 1

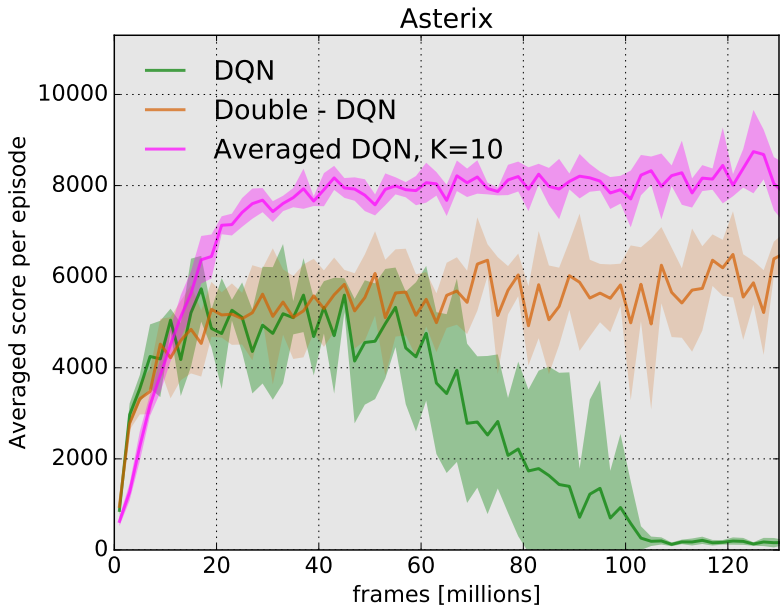


Example 1:
Instability and Variability

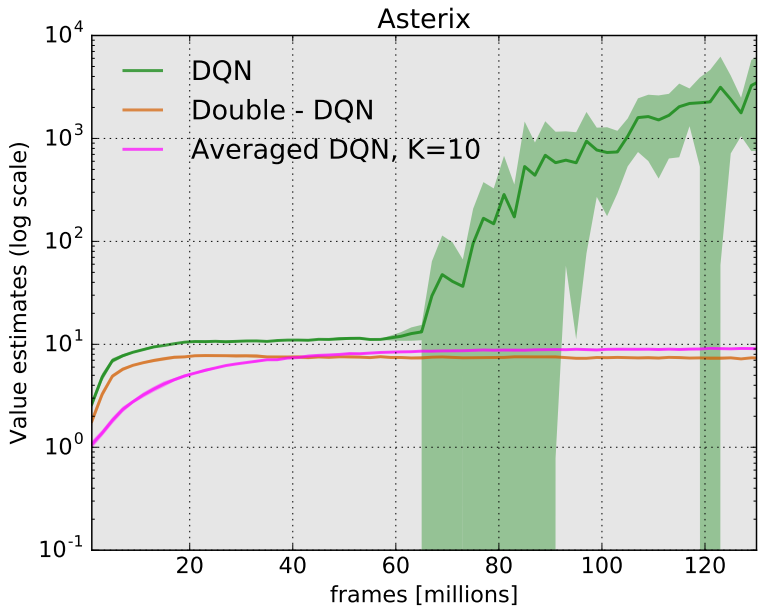
- Upside effects: **Stable training** and **better policies**.
- Bonus: Best model $\rightarrow$ Last model evaluation.

Technion Israel Institute of Technology

Motivational Example 2



(average score)



(value estimate)

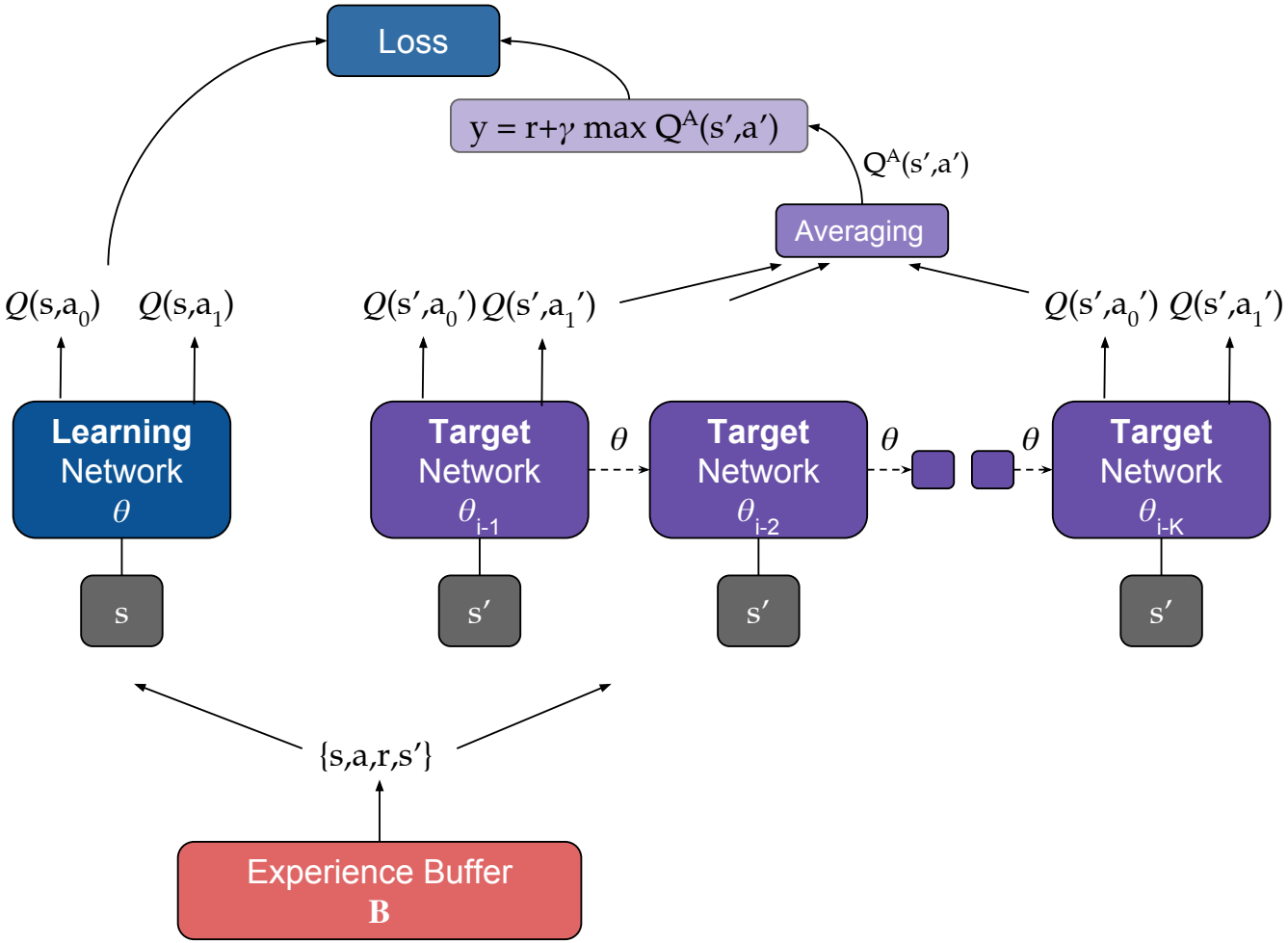
Example 2:
Overestimation[1]

[1] H van Hasselt et al., 2015

- Upside effect: Reduced overestimation.

Averaged DQN Agent

The **Averaged-DQN** Agent

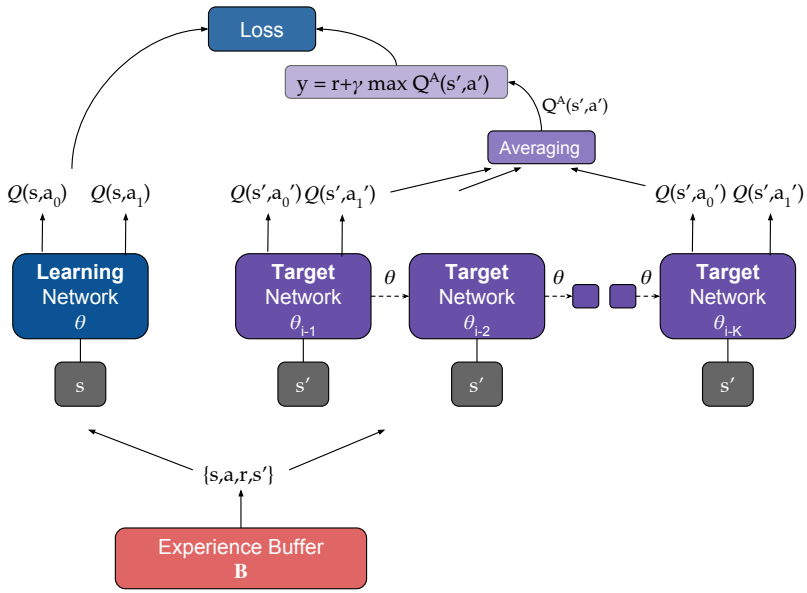


Averaged DQN

Algorithm Averaged DQN

1: Initialize $Q(s, a; \theta)$ with random weights θ_0
2: Initialize exploration procedure Explore($\cdot$)
3: Initialize Experience Replay (ER) buffer $\mathcal{B}$
4: **for** $i = 1, 2, \ldots, N$ **do**
5: Explore($\cdot$), update $\mathcal{B}$
6: $Q^A_{i-1}(s, a) = \frac{1}{K} \sum_{k=1}^{K} Q(s, a; \theta_{i-k})$
7: $y^i_{s,a} = \mathbb{E}_{\mathcal{B}} \left[r + \gamma \max_{a'} Q^A_{i-1}(s', a') | s, a \right]$
8: $\theta_i \approx \text{argmin}_\theta \ \mathbb{E}_{\mathcal{B}} \left[(y^i_{s,a} - Q(s, a; \theta))^2 \right]$

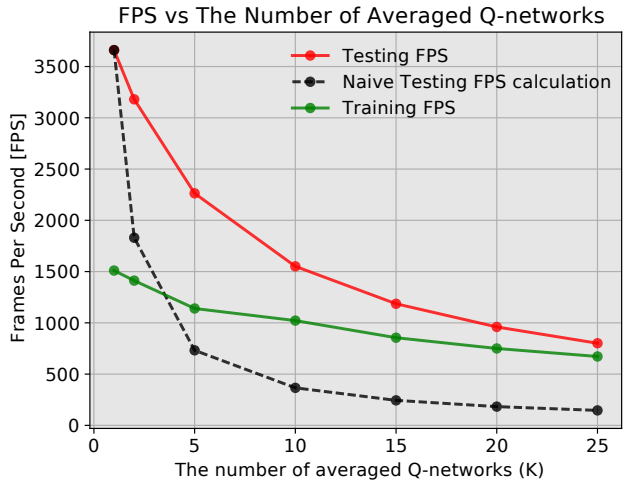
Output: $Q^A_N(s, a) = \frac{1}{K} \sum_{k=0}^{K-1} Q(s, a; \theta_{N-k})$



*In red: differences from DQN.

Computational Cost

- **Training:** $K-1$ more forward passes (each minimization of the DQN loss). Back-propagation updates $\rightarrow$ same as in DQN.
- **Testing:** $K-1$ more forward passes (for each evaluation of the Q-function).



*In general, the forward-passes through the K Q-networks can be done in parallel.

*In practice, since the same state is forwarded, the GPU implicitly parallelize the computation.

Overview

1 Background & Motivation

2 Averaged-DQN

3 Overestimation and Approximation Errors

4 TAE Variance Reduction

5 Experiments

Technion Israel Institute of Technology

Target Approximation Error (TAE)

- TAE: $Z^i_{s,a} = Q(s,a;\theta_i) - y^i_{s,a}$
- The error in $Q(s,a;\theta_i)$ relative to $y^i_{s,a}$.

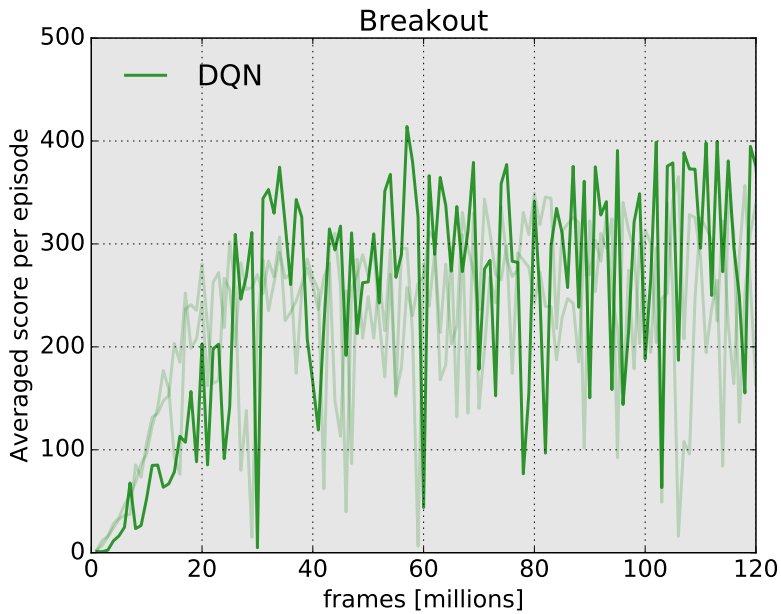
Algorithm DQN (Mnih, 2013)

1: Initialize $Q(s,a;\theta)$ with random weights θ_0
2: Initialize exploration procedure Explore($\cdot$)
3: Initialize Experience Replay (ER) buffer $\mathcal{B}$
4: **for** $i = 1, 2, \ldots, N$ **do**
5: Explore($\cdot$), update $\mathcal{B}$
6: $y^i_{s,a} = \mathbb{E}_{\mathcal{B}}\left[r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}) | s, a\right]$
7: $\theta_i \approx \textbf{argmin}_\theta \ \mathbb{E}_{\mathcal{B}}\left[(y^i_{s,a} - Q(s,a;\theta))^2\right]$

Output: $Q^{\text{DQN}}(s,a;\theta_N)$

Error source

- Sub-optimality of θ_i.
- Model expressiveness power.
- Finite ER buffer $\mathcal{B}$.



Technion Israel Institute of Technology

Target Approximation Error (TAE)

- TAE: $Z^i_{s,a} = Q(s,a;\theta_i) - y^i_{s,a}$
- The error in $Q(s,a;\theta_i)$ relative to $y^i_{s,a}$.

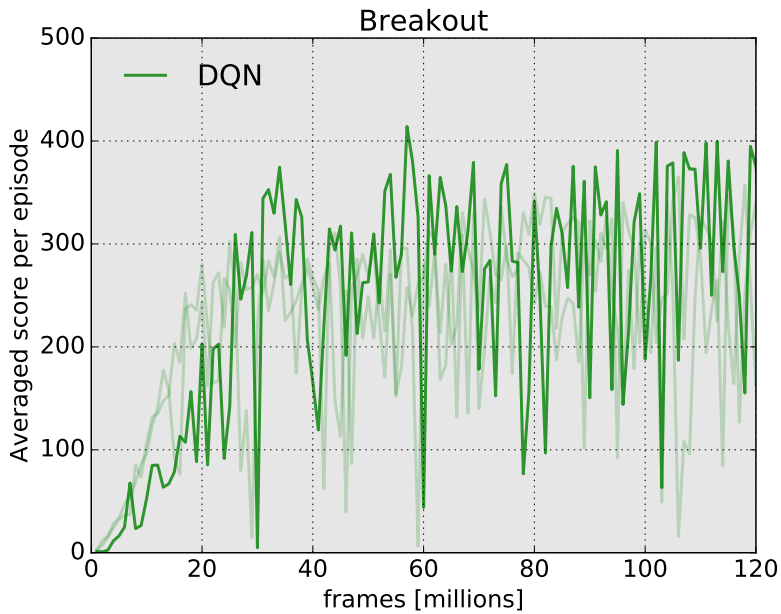
Algorithm DQN (Mnih, 2013)

1: Initialize $Q(s,a;\theta)$ with random weights θ_0
2: Initialize exploration procedure Explore($\cdot$)
3: Initialize Experience Replay (ER) buffer $\mathcal{B}$
4: **for** $i = 1, 2, \ldots, N$ **do**
5: Explore($\cdot$), update $\mathcal{B}$
6: $y^i_{s,a} = \mathbb{E}_{\mathcal{B}}\left[r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}) | s, a\right]$
7: $\theta_i \approx \textbf{argmin}_\theta \ \mathbb{E}_{\mathcal{B}}\left[\left(y^i_{s,a} - Q(s,a;\theta)\right)^2\right]$

Output: $Q^{\text{DQN}}(s,a;\theta_N)$

Error source

- Sub-optimality of θ_i.
- Model expressiveness power.
- Finite ER buffer $\mathcal{B}$.



Technion Israel Institute of Technology

Target Approximation Error (TAE)

- TAE: $Z^i_{s,a} = Q(s,a;\theta_i) - y^i_{s,a}$
- The error in $Q(s,a;\theta_i)$ relative to $y^i_{s,a}$.

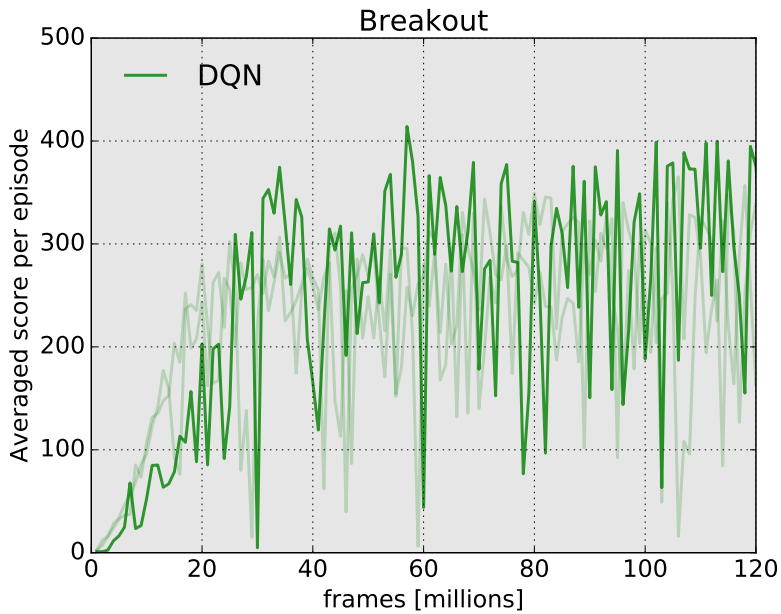
Algorithm DQN (Mnih, 2013)

1: Initialize $Q(s,a;\theta)$ with random weights θ_0
2: Initialize exploration procedure Explore($\cdot$)
3: Initialize Experience Replay (ER) buffer $\mathcal{B}$
4: **for** $i = 1, 2, \ldots, N$ **do**
5: Explore($\cdot$), update $\mathcal{B}$
6: $y^i_{s,a} = \mathbb{E}_{\mathcal{B}}\left[r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}) | s, a\right]$
7: $\theta_i \approx \textbf{argmin}_\theta \ \mathbb{E}_{\mathcal{B}}\left[\left(y^i_{s,a} - Q(s,a;\theta)\right)^2\right]$

Output: $Q^{\text{DQN}}(s,a;\theta_N)$

Error source

- Sub-optimality of θ_i.
- Model expressiveness power.
- Finite ER buffer $\mathcal{B}$.



Technion Israel Institute of Technology

Overestimation

- Overestimation: $R^i_{s,a} = y^i_{s,a} - \hat{y}^i_{s,a}$
- $y^i_{s,a} = \mathbb{E}_{\mathcal{B}}\left[r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}) | s, a\right]$
- $\hat{y}^i_{s,a} = \mathbb{E}_{\mathcal{B}}\left[r + \gamma \max_{a'} (y^{i-1}_{s',a'}) | s, a\right]$ **(without TAE)**

Algorithm DQN (Mnih, 2013)

1: Initialize $Q(s, a; \theta)$ with random weights θ_0
2: Initialize exploration procedure Explore($\cdot$)
3: Initialize Experience Replay (ER) buffer $\mathcal{B}$
4: **for** $i = 1, 2, \ldots, N$ **do**
5: Explore($\cdot$), update $\mathcal{B}$
6: $y^i_{s,a} = \mathbb{E}_{\mathcal{B}}\left[r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}) | s, a\right]$
7: $\theta_i \approx \text{argmin}_\theta \; \mathbb{E}_{\mathcal{B}}\left[(y^i_{s,a} - Q(s, a; \theta))^2\right]$

Output: $Q^{\text{DQN}}(s, a; \theta_N)$

Error source

- TAE variance.
- Similar Q-values for different actions.

[1] Thrun and Shwartz, 1993

Overestimation

- Overestimation: $R^i_{s,a} = y^i_{s,a} - \hat{y}^i_{s,a}$
- $y^i_{s,a} = \mathbb{E}_{\mathcal{B}}\left[r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}) | s, a\right]$
- $\hat{y}^i_{s,a} = \mathbb{E}_{\mathcal{B}}\left[r + \gamma \max_{a'} (y^{i-1}_{s',a'}) | s, a\right]$ **(without TAE)**

Algorithm DQN (Mnih, 2013)

1: Initialize $Q(s, a; \theta)$ with random weights θ_0
2: Initialize exploration procedure Explore($\cdot$)
3: Initialize Experience Replay (ER) buffer $\mathcal{B}$
4: **for** $i = 1, 2, \ldots, N$ **do**
5: Explore($\cdot$), update $\mathcal{B}$
6: $y^i_{s,a} = \mathbb{E}_{\mathcal{B}}\left[r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}) | s, a\right]$
7: $\theta_i \approx \text{argmin}_\theta \; \mathbb{E}_{\mathcal{B}}\left[(y^i_{s,a} - Q(s, a; \theta))^2\right]$

Output: $Q^{\text{DQN}}(s, a; \theta_N)$

Error source

- TAE variance.
- Similar Q-values for different actions.

[1] Thrun and Shwartz, 1993

Overestimation

- Overestimation: $R^i_{s,a} = y^i_{s,a} - \hat{y}^i_{s,a}$
- $y^i_{s,a} = \mathbb{E}_{\mathcal{B}}\left[r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}) | s, a\right]$
- $\hat{y}^i_{s,a} = \mathbb{E}_{\mathcal{B}}\left[r + \gamma \max_{a'} (y^{i-1}_{s',a'}) | s, a\right]$ **(without TAE)**

Algorithm DQN (Mnih, 2013)

1: Initialize $Q(s, a; \theta)$ with random weights θ_0
2: Initialize exploration procedure Explore($\cdot$)
3: Initialize Experience Replay (ER) buffer $\mathcal{B}$
4: **for** $i = 1, 2, \ldots, N$ **do**
5: Explore($\cdot$), update $\mathcal{B}$
6: $y^i_{s,a} = \mathbb{E}_{\mathcal{B}}\left[r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}) | s, a\right]$
7: $\theta_i \approx \text{argmin}_\theta \; \mathbb{E}_{\mathcal{B}}\left[(y^i_{s,a} - Q(s, a; \theta))^2\right]$

Output: $Q^{\text{DQN}}(s, a; \theta_N)$

Error source

- TAE variance.
- Similar Q-values for different actions.

[1] Thrun and Shwartz, 1993

Overestimation

- Overestimation: $R^i_{s,a} = y^i_{s,a} - \hat{y}^i_{s,a}$
- $y^i_{s,a} = \mathbb{E}_{\mathcal{B}}\left[r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}) | s, a\right]$
- $\hat{y}^i_{s,a} = \mathbb{E}_{\mathcal{B}}\left[r + \gamma \max_{a'} (y^{i-1}_{s',a'}) | s, a\right]$ **(without TAE)**

Algorithm DQN (Mnih, 2013)

1: Initialize $Q(s, a; \theta)$ with random weights θ_0
2: Initialize exploration procedure Explore($\cdot$)
3: Initialize Experience Replay (ER) buffer $\mathcal{B}$
4: **for** $i = 1, 2, \ldots, N$ **do**
5: Explore($\cdot$), update $\mathcal{B}$
6: $y^i_{s,a} = \mathbb{E}_{\mathcal{B}}\left[r + \gamma \max_{a'} Q(s', a'; \theta_{i-1}) | s, a\right]$
7: $\theta_i \approx \text{argmin}_\theta \; \mathbb{E}_{\mathcal{B}}\left[(y^i_{s,a} - Q(s, a; \theta))^2\right]$

Output: $Q^{\text{DQN}}(s, a; \theta_N)$

Error source

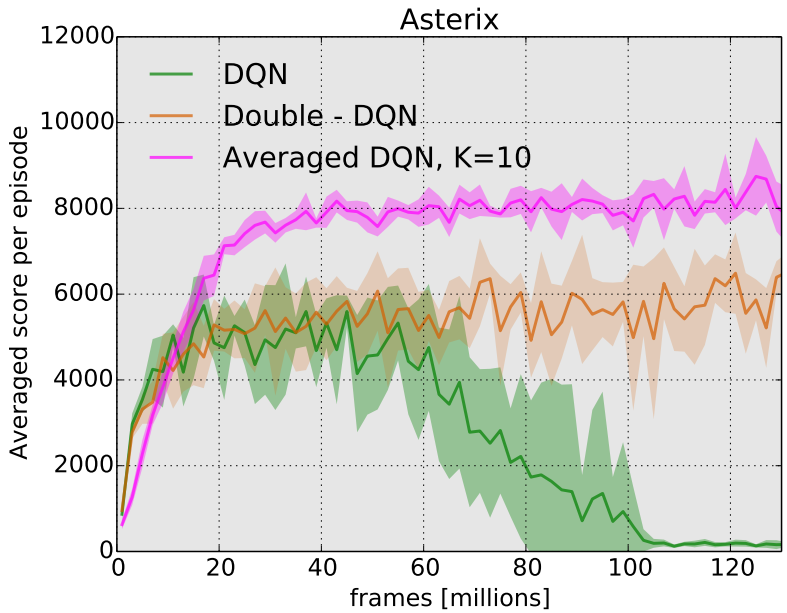
- TAE variance.
- Similar Q-values for different actions.

[1] Thrun and Shwartz, 1993

Overestimation Effect

Overestimation effect:

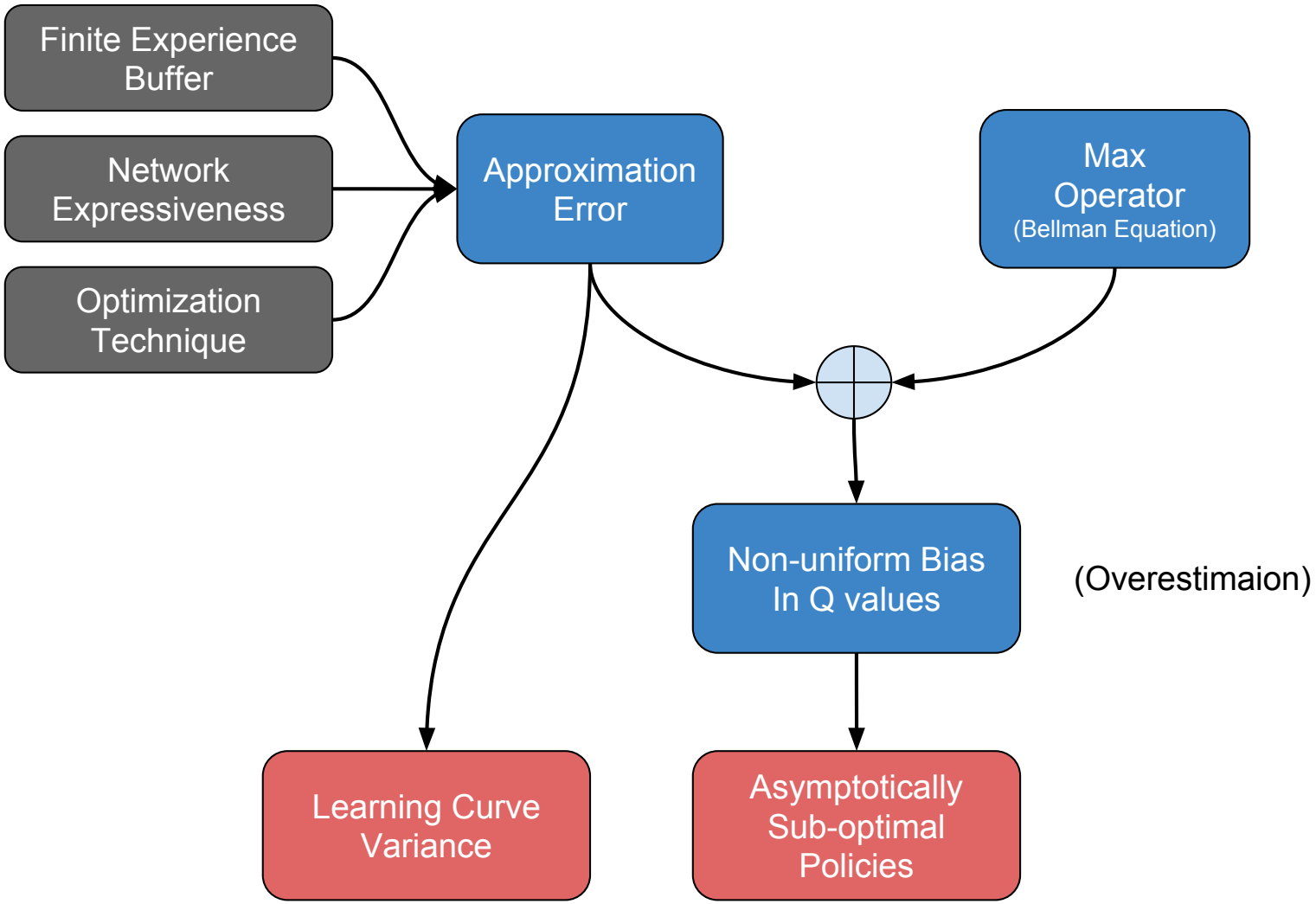
1. Non-uniform bias $\rightarrow$ Asymptotic sub-optimal policies[1,2].



[1] Thrun and Shwartz, 1993

[2] H van Hasselt et al., 2015

TAE and Overestimation Summary



Other Options

Directly dealing with the TAE sources:

- Prioritized experience replay (Schaul, et al. 2015).
- Dueling network architectures (Wang, et al., 2015).
- Smaller SGD learning rate/ Better optimizers.

Changing the Bellamn Operator:

- Double-DQN (Hasselt, et al. 2015).

Ensemble Techniques:

- Bootstrapped-DQN (Osband, et al. 2016).

Technion Israel Institute of Technology

Other Options

Directly dealing with the TAE sources:

- Prioritized experience replay (Schaul, et al. 2015).
- Dueling network architectures (Wang, et al., 2015).
- Smaller SGD learning rate/ Better optimizers.

Changing the Bellamn Operator:

- Double-DQN (Hasselt, et al. 2015).

Ensemble Techniques:

- Bootstrapped-DQN (Osband, et al. 2016).

Technion Israel Institute of Technology

Other Options

Directly dealing with the TAE sources:

- Prioritized experience replay (Schaul, et al. 2015).
- Dueling network architectures (Wang, et al., 2015).
- Smaller SGD learning rate/ Better optimizers.

Changing the Bellamn Operator:

- Double-DQN (Hasselt, et al. 2015).

Ensemble Techniques:

- Bootstrapped-DQN (Osband, et al. 2016).

Technion Israel Institute of Technology

Other Options

Directly dealing with the TAE sources:

- Prioritized experience replay (Schaul, et al. 2015).
- Dueling network architectures (Wang, et al., 2015).
- Smaller SGD learning rate/ Better optimizers.

Changing the Bellamn Operator:

- Double-DQN (Hasselt, et al. 2015).

Ensemble Techniques:

- Bootstrapped-DQN (Osband, et al. 2016).

Technion Israel Institute of Technology

Other Options

Directly dealing with the TAE sources:

- Prioritized experience replay (Schaul, et al. 2015).
- Dueling network architectures (Wang, et al., 2015).
- Smaller SGD learning rate/ Better optimizers.

Changing the Bellamn Operator:

- Double-DQN (Hasselt, et al. 2015).

Ensemble Techniques:

- Bootstrapped-DQN (Osband, et al. 2016).

Technion Israel Institute of Technology

Other Options

Directly dealing with the TAE sources:

- Prioritized experience replay (Schaul, et al. 2015).
- Dueling network architectures (Wang, et al., 2015).
- Smaller SGD learning rate/ Better optimizers.

Changing the Bellamn Operator:

- Double-DQN (Hasselt, et al. 2015).

Ensemble Techniques:

- Bootstrapped-DQN (Osband, et al. 2016).

Technion Israel Institute of Technology

Other Options

Directly dealing with the TAE sources:

- Prioritized experience replay (Schaul, et al. 2015).
- Dueling network architectures (Wang, et al., 2015).
- Smaller SGD learning rate/ Better optimizers.

Changing the Bellamn Operator:

- Double-DQN (Hasselt, et al. 2015).

Ensemble Techniques:

- Bootstrapped-DQN (Osband, et al. 2016).

Technion Israel Institute of Technology

Other Options

Directly dealing with the TAE sources:

- Prioritized experience replay (Schaul, et al. 2015).
- Dueling network architectures (Wang, et al., 2015).
- Smaller SGD learning rate/ Better optimizers.

Changing the Bellamn Operator:

- Double-DQN (Hasselt, et al. 2015).

Ensemble Techniques:

- Bootstrapped-DQN (Osband, et al. 2016).

Technion Israel Institute of Technology

Overview

1 Background & Motivation

2 Averaged-DQN

3 Overestimation and Approximation Errors

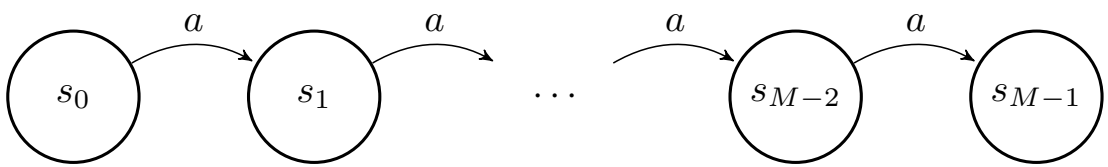
4 TAE Variance Reduction

5 Experiments

Technion Israel Institute of Technology

Averaged-DQN Variance Reduction

Unidirectional single-chain MDP



Result

For $K > 1, M > 1$:

$$\mathrm{Var}[Q_i^{\text{Averaged-DQN}}(s_0, a)] < \mathrm{Var}[Q_i^{\text{Ensemble-DQN}}(s_0, a)]$$
$$= \frac{1}{K}\mathrm{Var}[Q^{\text{DQN}}(s_0, a; \theta_i)]$$

*Averaged-DQN averages over averages.

Technion Israel Institute of Technology

Overview

1 Background & Motivation

2 Averaged-DQN

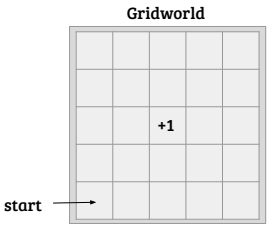
3 Overestimation and Approximation Errors

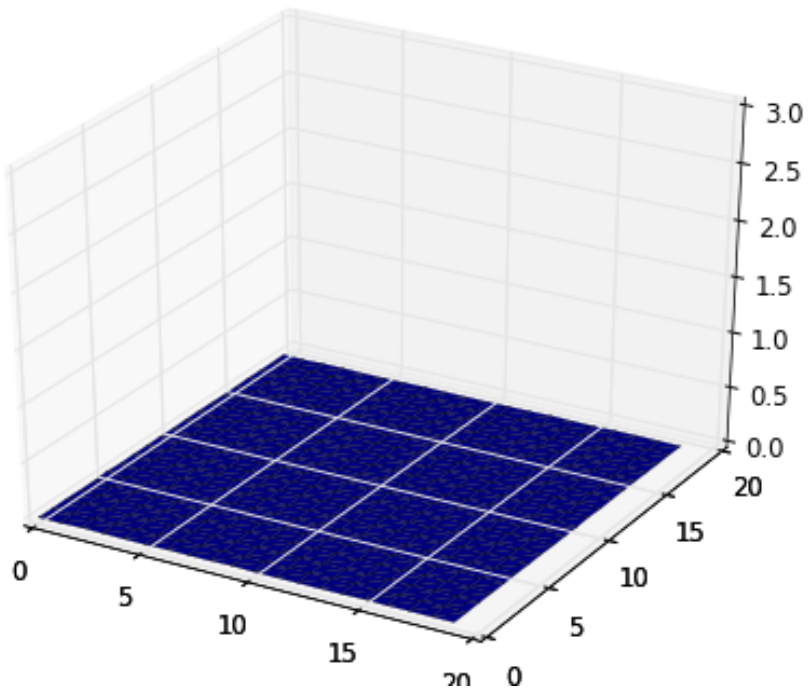
4 TAE Variance Reduction

5 Experiments

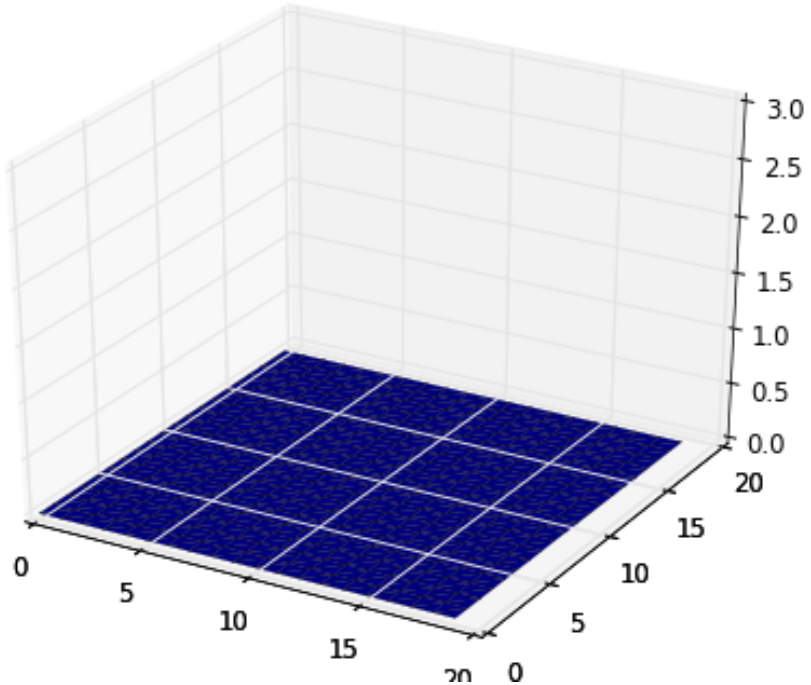
Technion Israel Institute of Technology

Gridworld Experiments





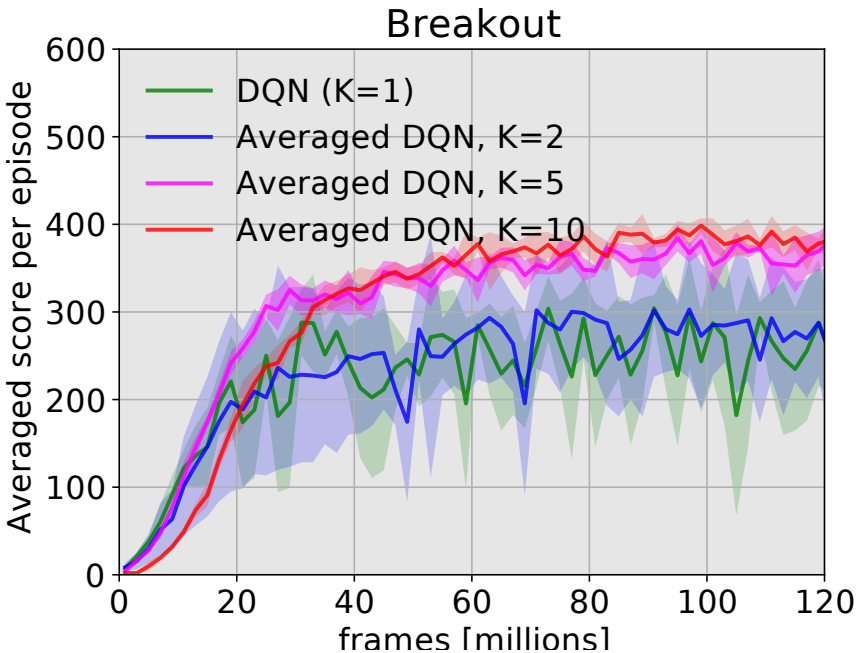
DQN

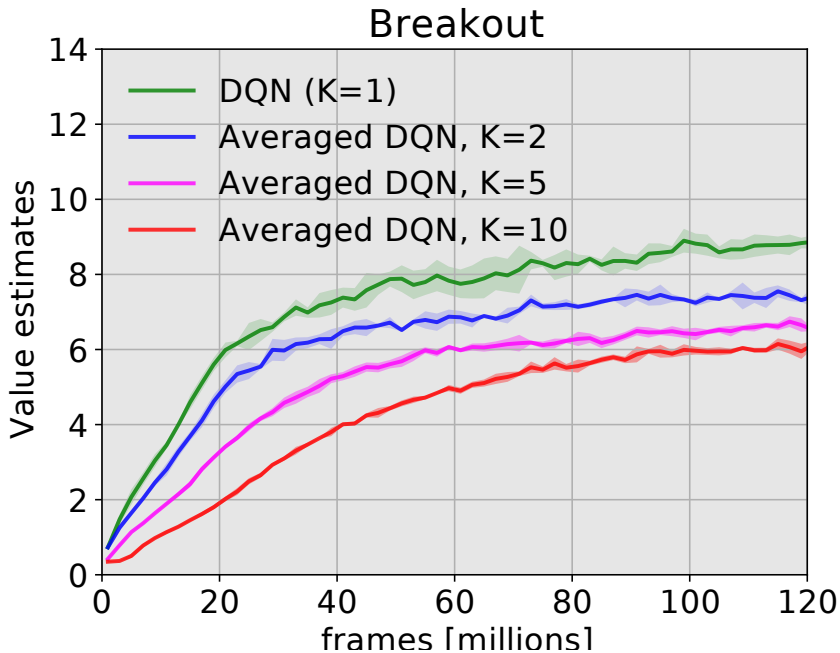


Averaged-DQN (K=15)

Technion Israel Institute of Technology

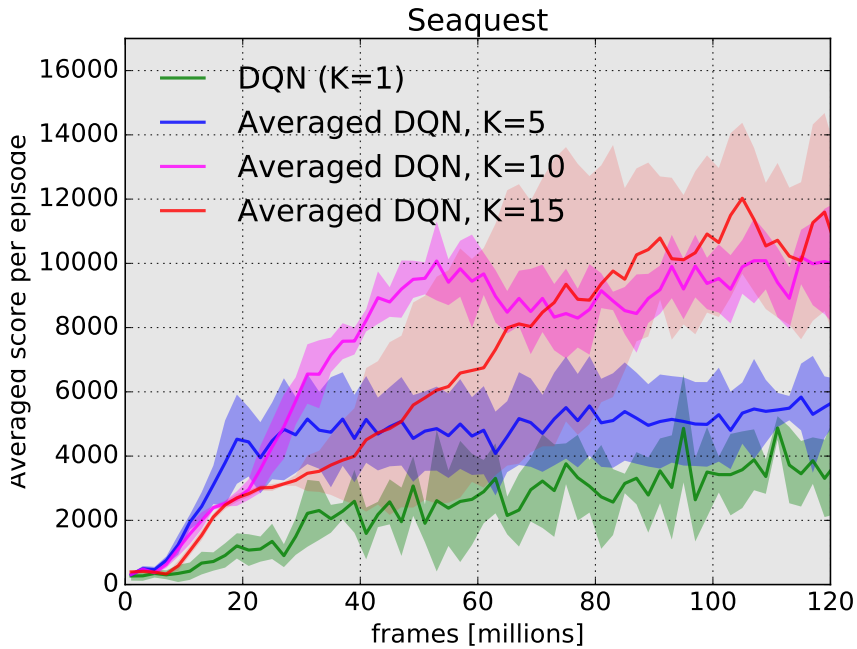
Arcade Learning Environment (Breakout)

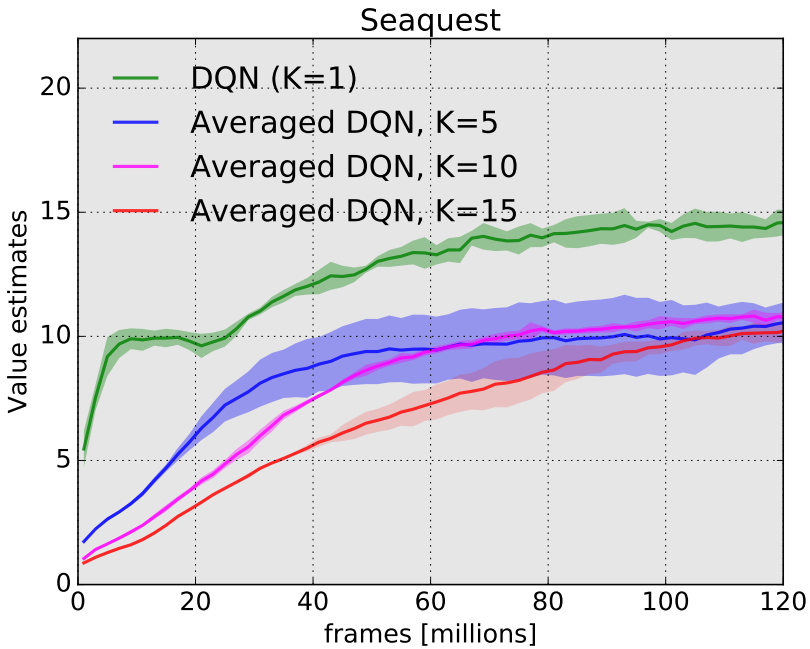




Technion Israel Institute of Technology

Arcade Learning Environment (Seaquest)





Thank you!

oronanschel@campus.technion.ac.il

Talk to me at poster #99

Slides at https://goo.gl/VQQ3Wj

Technion Israel Institute of Technology