

A UPDATING FORMULA IN VARIANCE ESTIMATION

Here we show that the weighted standard deviation in equation 6 can be obtained by updating σ with formula 7 in Variance Estimation. For simplicity, the inputs to all the following functions are hidden as they are the same state-action pair (s, a) .

While being updated with equation 1, we have

$$Q_n = (1 - \alpha)^n Q_0 + \sum_{i=1}^n \alpha (1 - \alpha)^{n-i} G_i \quad (13)$$

$$= \alpha \sum_{i=1}^n (1 - \alpha)^{n-i} G_i \quad (14)$$

The equal sign in formula 14 is established by setting $Q_0 = 0$.

If we expand $1/\alpha$ as a power series

$$\frac{1}{\alpha} = \frac{1 - (1 - \alpha)^n}{1 - (1 - \alpha)} \cdot \frac{1}{1 - (1 - \alpha)^n} \quad (15)$$

$$= \sum_{i=0}^{n-1} (1 - \alpha)^i \cdot \frac{1}{1 - (1 - \alpha)^n} \quad (16)$$

$$\approx \sum_{i=0}^{n-1} (1 - \alpha)^i \quad (17)$$

$$= \sum_{i=1}^n (1 - \alpha)^{n-i} \quad (18)$$

The approximate sign is used here to reflect that the second term of the multiplication approximates 1 as $n \rightarrow \infty$ because $\alpha \in (0, 1]$. The value function can be re-written as an exponentially weighted sum of the return sequence by applying the above expansion of power series:

$$Q_n \approx \frac{\sum_{i=1}^n (1 - \alpha)^{n-i} G_i}{\sum_{i=1}^n (1 - \alpha)^{n-i}} \quad (19)$$

We denote $A_n = \sum_{i=1}^n (1 - \alpha)^{n-i}$, then

$$\sigma_{n+1}^2 = \frac{\sum_{i=1}^{n+1} (1 - \alpha)^{n+1-i} (G_i - Q_{n+1})^2}{A_{n+1}} \quad (20)$$

$$= \frac{\sum_{i=1}^n (1 - \alpha)^{n+1-i} (G_i - Q_{n+1})^2}{A_{n+1}} + \frac{(G_{n+1} - Q_{n+1})^2}{A_{n+1}} \quad (21)$$

$$= (1 - \alpha) \cdot \frac{\sum_{i=1}^n (1 - \alpha)^{n-i} (G_i - Q_{n+1})^2}{A_{n+1}} + \frac{(G_{n+1} - Q_{n+1})^2}{A_{n+1}} \quad (22)$$

$$= (1 - \alpha) \cdot \frac{\sum_{i=1}^n (1 - \alpha)^{n-i} (G_i - Q_n + Q_n - Q_{n+1})^2}{A_{n+1}} + \frac{(G_{n+1} - Q_{n+1})^2}{A_{n+1}} \quad (23)$$

$$= (1 - \alpha) \cdot \left[\frac{\sum_{i=1}^n (1 - \alpha)^{n-i} (G_i - Q_n)^2}{A_{n+1}} + \frac{\sum_{i=1}^n (1 - \alpha)^{n-i} (Q_n - Q_{n+1})^2}{A_{n+1}} \right. \\ \left. + \frac{2 \sum_{i=1}^n (1 - \alpha)^{n-i} (G_i - Q_n)(Q_n - Q_{n+1})}{A_{n+1}} \right] + \frac{(G_{n+1} - Q_{n+1})^2}{A_{n+1}} \quad (24)$$

$$\approx (1 - \alpha) \cdot \frac{A_n}{A_{n+1}} \cdot [\sigma_n^2 + (Q_{n+1} - Q_n)^2 + 0] + \frac{(G_{n+1} - Q_{n+1})^2}{A_{n+1}} \quad (25)$$

$$\approx (1 - \alpha) [\sigma_n^2 + (Q_{n+1} - Q_n)^2] + \alpha (G_{n+1} - Q_{n+1})^2 \quad (26)$$

where the first approximate sign comes from formula 19 and the second one comes from formula 17.

Though the updating formula is biased as n doesn't approach infinity in practice, the bias is negligible. Because all the above approximations are rooted in formula 17 and $(1 - \alpha)^n$ converges to 0 rapidly as n grows.

B AN EMPIRICAL ANALYSIS

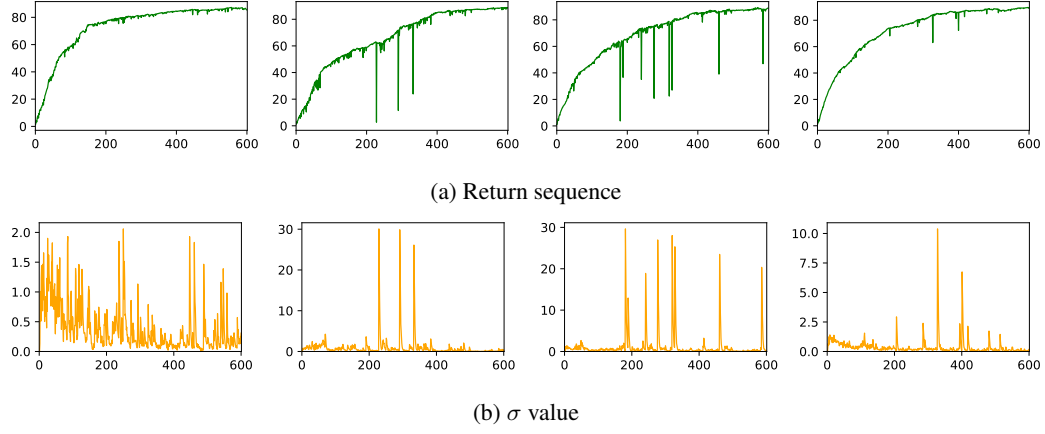


Figure 5: Raw return sequences (a) and corresponding σ values (b) over visitations.

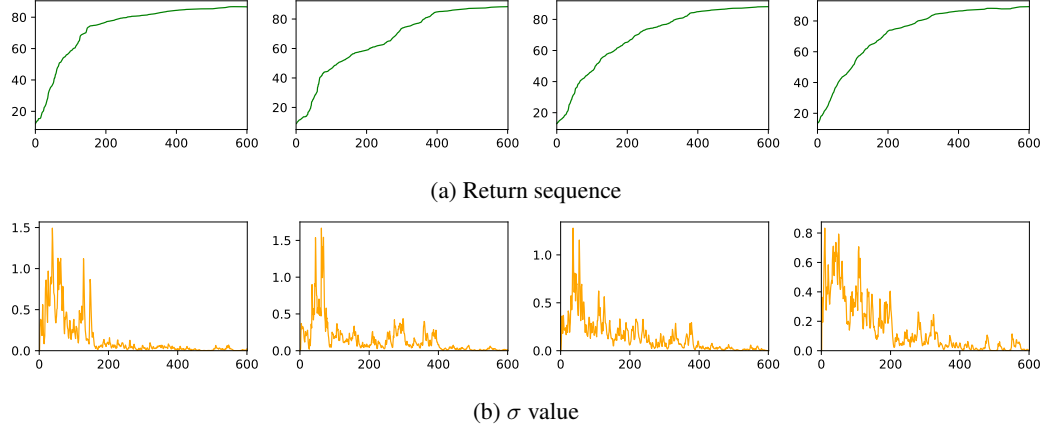


Figure 6: Smoothed return sequences (a) and corresponding σ values (b) over visitations.

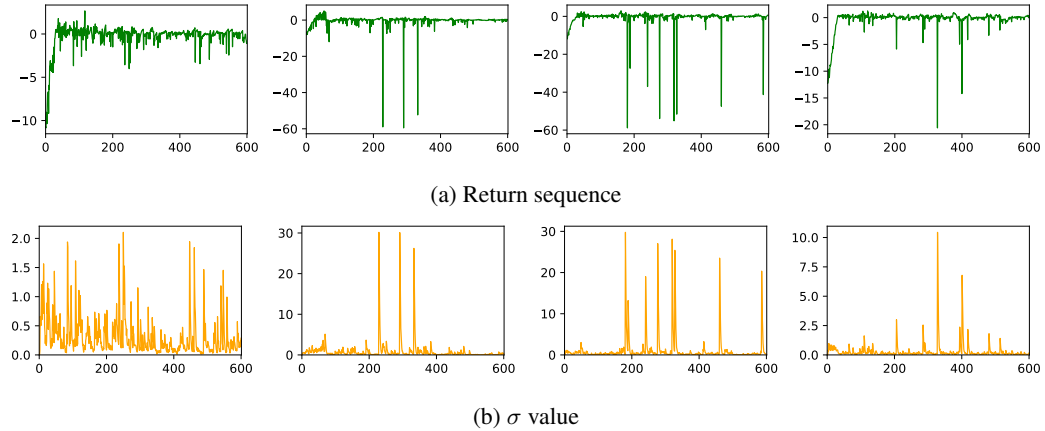


Figure 7: Residual return sequences (a) and corresponding σ values (b) over visitations.

Sutton & Barto (2018) prove that the policy and value functions monotonically improve over sweeps in policy iteration. In this simple form, return sequences for individual state-action pairs are monotonically increasing and usually converge fast. However, the same statement is not true when the policy is updated with Monte Carlo settings where the returns are random samples. In this situation, fluctuations in returns are inevitable due to the stochastic property; an incremental updating scheme is adopted to stabilize learning as well as avoid the excessive impact of malicious samples.

Here we illustrate the characteristics of the return sequences and analyze how variance guides exploration. The raw return sequences shown in Figure 5a(a) are randomly sampled from frequently visited state-action pairs in the Cartpole balancing problem and truncated to be of the same length. To disentangle the impact of the convergence trend from transient fluctuations, smoothed versions of those sequences are extracted from the raw sequences (shown in Figure 6a), while the residuals are shown in Figure 7a. Then we apply our Variance Estimation method on each of the sequences independently and show the σ values over visits below the sequences.

While σ values are integrated into the Q values used by our exploration policy (shown in Equation 5), a greater σ value usually results in an increase in exploration budget. Meanwhile, the nature of weighted sum balances the importance of learned Q value and the history of its change which is captured by σ .

In Figure 6b, we observe that the σ value is greater when the return sequence changes at a faster rate. As the sequence converges, the σ value approaches zero. An interesting but unobvious observation is that the σ value spikes when there is change in the convergence rate of the return sequence. Since variance is essentially a Euclidean distance metric, it is capable of capturing second-order information.

Sequences in Figure 7a isolate the impacts of transient fluctuations from the overall trend. We observe that whenever there an excessive fluctuation appears, σ value spikes to a high magnitude to demand immediate exploration. Once the return goes back to its normal range, the σ value decreases simultaneously. Those quick responses are useful since excessive fluctuations are harmful to the estimation of Q values. A timely investigation of exploration budget eliminates this negative impact before it propagates to more states. Meanwhile, frequent fluctuations beget an increase in σ value and result in more exploration to determine its value.

In conclusion, the exploration policy on our constructed upper bound effectively allocates exploration budget to accelerate convergence in important states as well as alleviate the impact of fluctuations.

C HYPERPARAMETERS

Table 2: Atari DQN Hyperparameters

Hyperparameter	Value	Description
c_{V-DQN}	0.1	Weighting factor of σ -stream in exploration policy in V-DQN
c_{TD-DQN}	0.1	Weighting factor of σ -stream in exploration policy in TD-DQN
mini-batch size	32	Size of mini-batch sample for gradient step
replay buffer size	1M	Maximum number of transitions stored in the replay buffer
initial replay buffer size	50K	Number of transitions stored in the replay buffer before optimization starts
optimization frequency	4	Number of actions the agent takes between successive network optimization steps
update frequency	30000	Number of steps between consecutive target updates
ϵ_{init}	1.00	Initial exploration rate of ϵ -greedy method
ϵ_{final}	0.01	Final exploration rate of ϵ -greedy method
N_{ϵ}	1M	Number of actions that the exploration rate of ϵ -greedy method decays from initial value to final value
α	0.0000625	Adam optimizer learning rate
ϵ_{ADAM}	0.00015	Adam optimizer parameter
evaluation frequency	250K	Number of actions between successive evaluation runs
evaluation length	125K	Number of actions per evaluation run
evaluation episode length	27K	Maximum number of action in an episode in evaluation runs
max no-op	30	Maximum number of no-op actions before the episode starts

D EXPERIMENTAL RESULTS ON ATARI GAMES

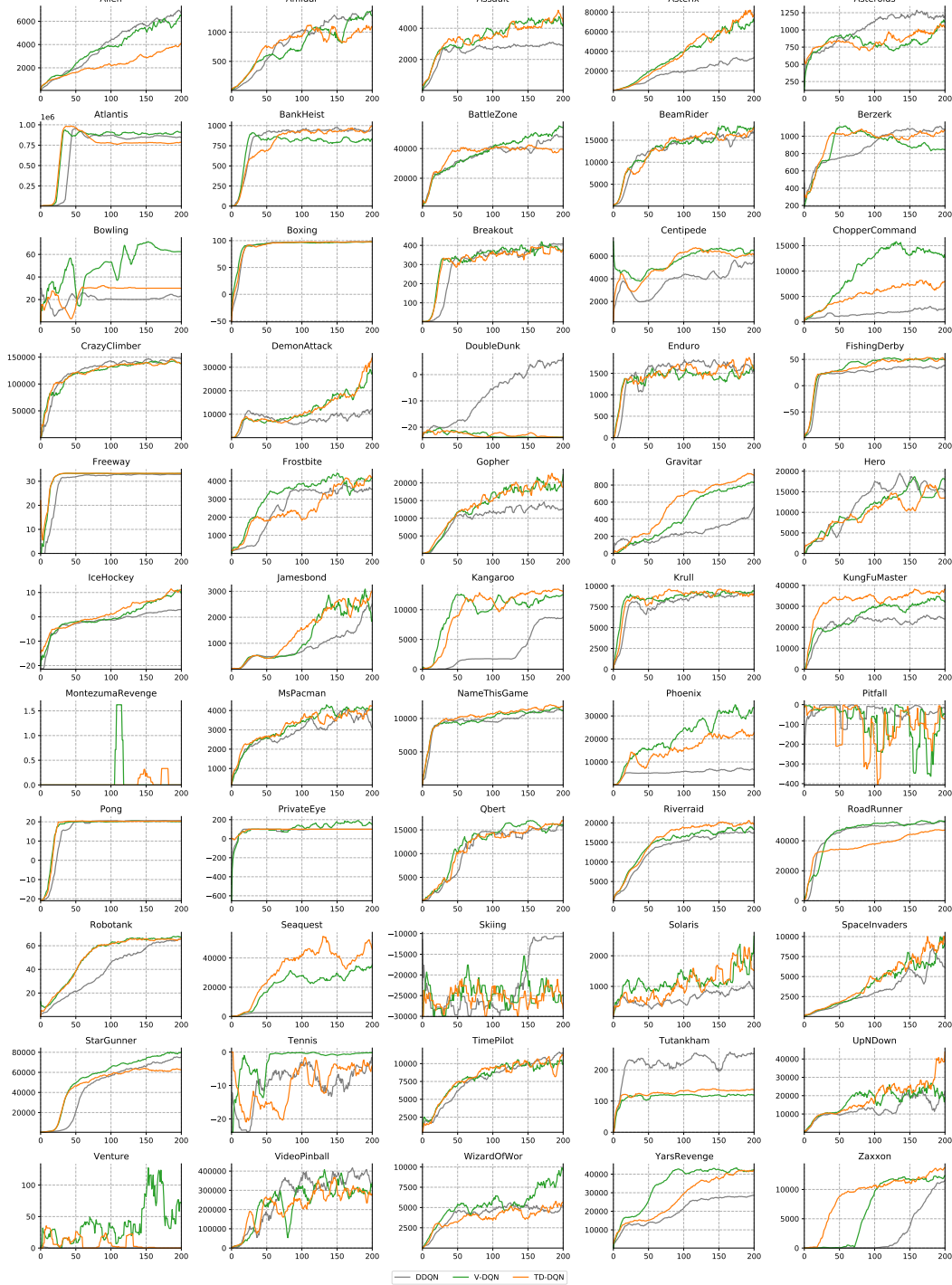


Figure 8: Training curve on Atari games from a single training run each. Episodes start with up to 30 no-op actions. Each data point is an average of episode rewards from 500K frames of evaluation runs, and smoothed over 10 data points.

Game	DDQN	V-DQN	TD-DQN
Alien	123%	130%	76%
Amidar	99%	97%	97%
Assault	764%	1659%	1324%
Asterix	537%	1280%	1490%
Asteroids	2%	1%	1%
Atlantis	7445%	7334%	7714%
BankHeist	165%	158%	170%
BattleZone	168%	199%	149%
BeamRider	143%	153%	156%
Berzerk	51%	49%	49%
Bowling	7%	52%	18%
Boxing	902%	903%	904%
Breakout	1764%	1851%	1644%
Centipede	57%	64%	64%
ChopperCommand	39%	237%	111%
CrazyClimber	639%	658%	646%
DemonAttack	563%	1415%	1700%
DoubleDunk	1568%	-145%	-90%
Enduro	267%	284%	281%
FishingDerby	151%	164%	164%
Freeway	130%	131%	131%
Frostbite	108%	116%	112%
Gopher	1071%	1689%	1665%
Gravitar	14%	22%	25%
Hero	83%	81%	82%
IceHockey	139%	227%	253%
Jamesbond	890%	1315%	1071%
Kangaroo	354%	529%	552%
Krull	909%	917%	931%
KungFuMaster	150%	199%	214%
MontezumaRevenge	-1%	-0%	-1%
MsPacman	32%	33%	29%
NameThisGame	207%	210%	218%
Phoenix	134%	852%	531%
Pitfall	5%	5%	5%
Pong	116%	115%	115%
PrivateEye	-1%	-1%	-1%
Qbert	146%	150%	160%
Riverraid	136%	152%	156%
RoadRunner	826%	827%	724%
Robotank	1006%	1071%	1051%
Seaquest	6%	122%	226%
Skiing	43%	37%	35%
Solaris	-5%	50%	40%
SpaceInvaders	889%	1527%	1318%
StarGunner	922%	984%	760%
Tennis	197%	147%	146%
TimePilot	387%	365%	366%
Tutankham	215%	108%	109%
UpNDown	366%	516%	707%
Venture	-1%	36%	18%
VideoPinball	425%	425%	425%
WizardOfWor	168%	309%	206%
YarsRevenge	61%	98%	95%
Zaxxon	151%	175%	183%

Table 3: Normalized scores.

Game	Random	Human	DDQN	V-DQN	TD-DQN
Alien	128.3	6371.3	7807.3	8236.9	4895.4
Amidar	11.8	1540.4	1521.6	1495.0	1494.0
Assault	166.9	628.9	3697.5	7829.8	6284.9
Asterix	164.5	7536.0	39782.0	94509.1	110013.6
Asteroids	871.3	36517.3	1464.3	1317.6	1278.0
Atlantis	13463.0	26575.0	989675.0	975100.0	1024975.0
BankHeist	21.7	644.5	1050.1	1007.7	1082.3
BattleZone	3560.0	33030.0	53153.8	62133.3	47460.0
BeamRider	254.6	14961.0	21296.0	22765.7	23234.0
Berzerk	196.1	2237.5	1228.1	1205.0	1190.7
Bowling	35.2	146.5	42.7	93.6	54.7
Boxing	-1.5	9.6	98.6	98.7	98.8
Breakout	1.6	27.9	465.5	488.3	434.1
Centipede	1925.5	10321.9	6695.1	7271.9	7282.8
ChopperCommand	644.0	8930.0	3900.0	20289.7	9835.0
CrazyClimber	9337.0	32667.0	158346.2	162828.0	159952.0
DemonAttack	208.3	3442.8	18418.2	45977.5	55200.6
DoubleDunk	-16.0	-14.4	9.1	-18.3	-17.4
Enduro	-81.8	740.2	2113.5	2250.2	2225.0
FishingDerby	-77.1	5.1	46.6	57.7	57.7
Freeway	0.1	25.6	33.2	33.6	33.6
Frostbite	66.4	4202.8	4516.2	4883.8	4702.4
Gopher	250.0	2311.0	22331.4	35061.4	34555.7
Gravitar	245.5	3116.0	637.5	869.5	976.1
Hero	1580.3	25839.4	21606.6	21244.0	21476.5
IceHockey	-9.7	0.5	4.5	13.5	16.1
Jamesbond	33.5	368.5	3014.2	4438.8	3622.5
Kangaroo	100.0	2739.0	9450.0	14057.6	14679.4
Krull	1151.9	2109.1	9855.5	9930.0	10065.0
KungFuMaster	304.0	20786.8	31070.7	40984.4	44177.4
MontezumaRevenge	25.0	4182.0	0.0	9.1	3.3
MsPacman	197.8	15375.0	5038.5	5246.7	4585.7
NameThisGame	1747.8	6796.0	12181.1	12359.4	12774.7
Phoenix	1134.4	6686.2	8568.6	48424.7	30623.9
Pitfall	-348.8	5998.9	0.0	0.0	0.0
Pong	-18.0	15.5	20.8	20.6	20.6
PrivateEye	662.8	64169.1	100.0	200.0	100.0
Qbert	183.0	12085.0	17551.4	18051.7	19250.0
Riverraid	588.3	14382.2	19322.9	21545.0	22073.8
RoadRunner	200.0	6878.0	55381.7	55437.1	48526.7
Robotank	2.4	8.9	67.8	72.0	70.7
Seaquest	215.5	40425.8	2789.8	49230.8	91277.1
Skiing	-15287.4	-3686.6	-10314.1	-10990.9	-11215.2
Solaris	2047.2	11032.6	1572.0	6497.6	5615.0
SpaceInvaders	182.6	1464.9	11580.8	19757.7	17086.2
StarGunner	697.0	9528.0	82076.7	87577.8	67768.6
Tennis	-21.4	-6.7	7.5	0.2	0.0
TimePilot	3273.0	5650.0	12460.7	11941.4	11975.0
Tutankham	12.7	138.3	283.0	147.8	150.1
UpNDown	707.2	9896.1	34346.4	48118.3	65673.9
Venture	18.0	1039.0	9.6	382.9	197.1
VideoPinball	20452.0	15641.1	584388.2	632013.8	631348.0
WizardOfWor	804.0	4556.0	7115.1	12388.1	8551.7
YarsRevenge	1476.9	47135.2	29332.9	46319.6	44894.4
Zaxxon	475.0	8443.0	12488.4	14409.8	15028.3

Table 4: Raw Scores.