Weighted Bellman Losses for Improved Signal-to-Noise in Q-Updates

Pieter Abbeel

UC Berkeley EECS

Joint work with
Kimin Lee, Misha Laskin, Aravind Srinivas

Cause of Instability and Noise in Q-Learning

- Error propagation in Q-learning

unseen $(s_{t+1},a) \rightarrow$ high error

$$Q(s_t, a_t) \leftarrow r_t + \gamma \max_a Q(s_{t+1}, a)$$

error propagates

Weighted Bellman Backup

- Error propagation issue in Q-learning

unseen $(s_{t+1}, a) \rightarrow$ high error

$$Q(s_t, a_t) \leftarrow r_t + \gamma \max_a Q(s_{t+1}, a)$$

error propagates

- Reweighting Bellman backup can handle this issue

$$w(s,a)\left(Q(s,a) - [r(s,a) + \gamma \widehat{Q}(s', a')]\right)^2$$

Some confidence score about target value

Weighted Bellman Backup

- Error propagation issue in Q-learning

unseen $(s_{t+1},a) \rightarrow$ high error

$$Q(s_t, a_t) \leftarrow r_t + \gamma \max_a Q(s_{t+1}, a)$$

error propagates

- Reweighting Bellman backup can handle this issue

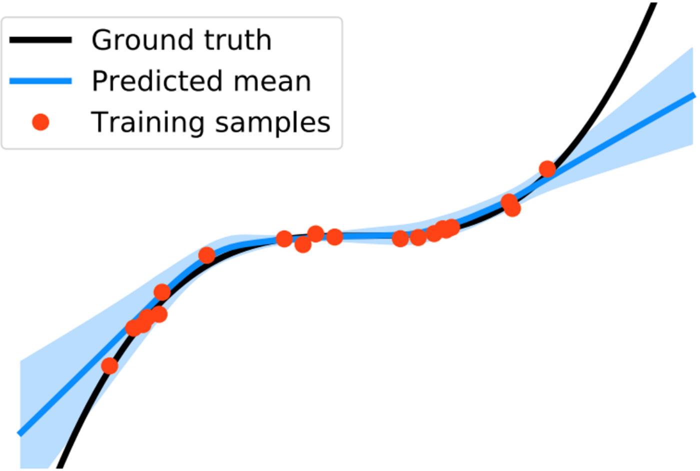
$$w(s,a)\left(Q(s,a) - \left[r(s,a) + \gamma \widehat{Q}(s',a')\right]\right)^2$$

Some confidence score about target value

How to quantify the uncertainty on target value?

Weighted Bellman Backup

- Main idea: **uncertainty estimation using ensembles** [Osband et al., 2016, Lakshminarayanan et al., 2017]
  - Toy regression task



Ensembles can produce **well-calibrated uncertainty** estimates (i.e., **variance**) on **unseen** samples

[Osband et al., 2016] Osband, I., Blundell, C., Pritzel, A. and Van Roy, B., Deep exploration via bootstrapped DQN. In NeurIPS, 2016.
[Lakshminarayanan et al., 2017] Lakshminarayanan, B., Pritzel, A. and Blundell, C., Simple and scalable predictive uncertainty estimation using deep ensembles. In NeurIPS, 2017

Weighted Bellman Backup

- Definition of confidence score

$$w(s,a) = \underbrace{\sigma}_{\text{Sigmoid}} \left(-\bar{Q}_{\mathtt{std}}(s,a) * \underbrace{T}_{\text{Temperature}} \right) + \underline{0.5}$$

  - Small variance: weight → 1.0
  - High variance: weight → 0.5

Weighted Bellman Backup

- Definition of confidence score

$$w(s,a) = \underbrace{\sigma}_{\text{Sigmoid}}\left(-\bar{Q}_{\texttt{std}}(s,a) * \underbrace{T}_{\text{Temperature}}\right) + \underline{0.5}$$

  - Small variance: weight → 1.0
  - High variance: weight → 0.5

- Weighted Bellman backup loss

$$w(s,a)\left(Q(s,a) - [r(s,a) + \gamma\widehat{Q}(s',a')]\right)^2$$

UCB Exploration

- Main idea: utilize uncertainty estimation for exploration
- UCB exploration based on Q-ensemble [Chen et al., 2017]

$$a_t = \max_a \{ \underbrace{Q_{\texttt{mean}}(s_t, a)}_{\text{exploit}} + \underbrace{\lambda Q_{\texttt{std}}(s_t, a)}_{\text{explore}} \}$$

- We further extend to continuous action space and apply to more advanced off-policy RL algorithms

[Chen et al., 2017] Chen, R.Y., Sidor, S., Abbeel, P. and Schulman, J., UCB exploration via Q-ensembles. arXiv preprint arXiv:1706.01502, 2017.

Pseudo Algorithm

Algorithm 1 SUNRISE: SAC version

```
for each iteration do
    for each timestep t do
        // UCB EXPLORATION
        Collect N action samples: A_t = {a_{t,i} ~ π_{φ_i}(a|s_t) | i ∈ {1,...,N}}
        Choose the action that maximizes UCB: a_t = argmax_{a_{t,i} ∈ A_t} Q_mean(s_t, a_{t,i}) + λ Q_std(s_t, a_{t,i})
        Collect state s_{t+1} and reward r_t from the environment by taking action a_t
        Sample bootstrap masks M_t = {m_{t,i} ~ Bernoulli(β) | i ∈ {1,...,N}}
        Store transitions τ_t = (s_t, a_t, s_{t+1}, r_t) and masks in replay buffer B ← B ∪ {(τ_t, M_t)}
    end for
    // UPDATE AGENTS VIA BOOTSTRAP AND WEIGHTED BELLMAN BACKUP
    for each gradient step do
        Sample random minibatch {(τ_j, M_j)}_{j=1}^B ~ B
        for each agent i do
            Update the Q-function by minimizing (1/B) Σ_{j=1}^B m_{j,i} L_WQ(τ_j, θ_i)
            Update the policy by minimizing (1/B) Σ_{j=1}^B m_{j,i} L_π(s_j, φ_i)
        end for
    end for
end for
```

Interact with environment using UCB inference

Optimize ensemble agents via weighted Bellman backups

Experimental Results

- OpenAI Gym (state, continuous action)
- DeepMind Control Suite (pixel, continuous action)
- Atari (pixel, discrete action)

Experimental Results on OpenAI Gym

- Performance on OpenAI Gym at 200K timesteps

	Cheetah	Walker	Hopper	Ant
PETS [12]	2288.4 ± 1019.0	282.5 ± 501.6	114.9 ± 621.0	1165.5 ± 226.9
POPLIN-A [49]	1562.8 ± 1136.7	-105.0 ± 249.8	202.5 ± 962.5	1148.4 ± 438.3
POPLIN-P [49]	4235.0 ± 1133.0	597.0 ± 478.8	2055.2 ± 613.8	$\mathbf{2330.1 \pm 320.9}$
METRPO [28]	2283.7 ± 900.4	-1609.3 ± 657.5	1272.5 ± 500.9	282.2 ± 18.0
TD3 [14]	3015.7 ± 969.8	-516.4 ± 812.2	1816.6 ± 994.8	870.1 ± 283.8
SAC [15]	4035.7 ± 268.0	-382.5 ± 849.5	2020.6 ± 692.9	836.5 ± 68.4
SUNRISE	$\mathbf{5370.6 \pm 483.1}$	$\mathbf{1926.5 \pm 694.8}$	$\mathbf{2601.9 \pm 306.5}$	1627.0 ± 292.7

  - Always improve the performance of SAC by large margin

Experimental Results on OpenAI Gym

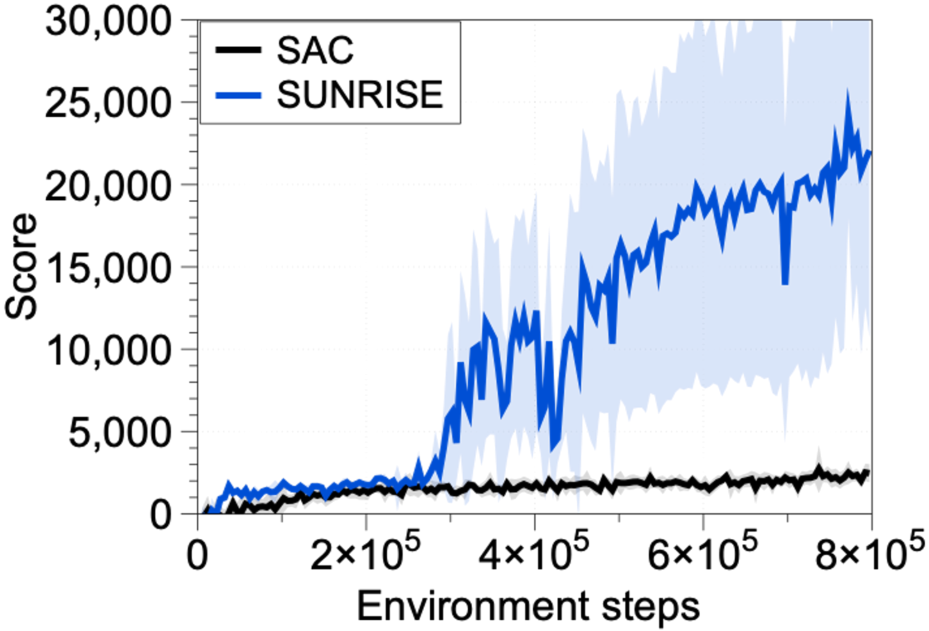
- Performance on OpenAI Gym at 200K timesteps

	Cheetah	Walker	Hopper	Ant
PETS [12]	2288.4 ± 1019.0	282.5 ± 501.6	114.9 ± 621.0	1165.5 ± 226.9
POPLIN-A [49]	1562.8 ± 1136.7	-105.0 ± 249.8	202.5 ± 962.5	1148.4 ± 438.3
POPLIN-P [49]	4235.0 ± 1133.0	597.0 ± 478.8	2055.2 ± 613.8	$\mathbf{2330.1 \pm 320.9}$
METRPO [28]	2283.7 ± 900.4	-1609.3 ± 657.5	1272.5 ± 500.9	282.2 ± 18.0
TD3 [14]	3015.7 ± 969.8	-516.4 ± 812.2	1816.6 ± 994.8	870.1 ± 283.8
SAC [15]	4035.7 ± 268.0	-382.5 ± 849.5	2020.6 ± 692.9	836.5 ± 68.4
SUNRISE	$\mathbf{5370.6 \pm 483.1}$	$\mathbf{1926.5 \pm 694.8}$	$\mathbf{2601.9 \pm 306.5}$	1627.0 ± 292.7

  - Always improve the performance of SAC by large margin
  - Outperform SOTA model-based RL methods like POPLIN and PETS on Cheetah, Walker, Hopper

Experimental Results on OpenAI Gym

- Results on SlimHumanoid [Wang et al., 2019]



SUNRISE can be effective at handling complex environments like Humanoid

Gains from SUNRISE become more significant when learning longer

[Wang et al., 2019] Wang, T., Bao, X., Clavera, I., Hoang, J., Wen, Y., Langlois, E., Zhang, S., Zhang, G., Abbeel, P. and Ba, J., Benchmarking model-based reinforcement learning. arXiv preprint arXiv:1907.02057, 2019

Experimental Results on DM Control

- Performance on DeepMind Control Suite at 100K and 500K environment steps

500K step	PlaNet [16]	Dreamer [17]	SLAC [31]	CURL [41]	DrQ [25]	RAD [30]	SUNRISE
Finger-spin	561 ± 284	796 ± 183	673 ± 92	926 ± 45	938 ± 103	975 ± 16	**983** ±1
Cartpole-swing	475 ± 71	762 ± 27	-	845 ± 45	868 ± 10	873 ± 3	**876** ± 4
Reacher-easy	210 ± 44	793 ± 164	-	929 ± 44	942 ± 71	916 ± 49	**982** ± 3
Cheetah-run	305 ± 131	570 ± 253	640 ± 19	518 ± 28	660 ± 96	624 ± 10	**678** ± 46
Walker-walk	351 ± 58	897 ± 49	842 ± 51	902 ± 43	921 ± 45	938 ± 9	**953** ± 13
Cup-catch	460 ± 380	879 ± 87	852 ± 71	959 ± 27	963 ± 9	966 ± 9	**969** ± 5
100K step							
Finger-spin	136 ± 216	341 ± 70	693 ± 141	767 ± 56	901 ± 104	811 ± 146	**905** ± 57
Cartpole-swing	297 ± 39	326 ± 27	-	582 ± 146	**759** ± 92	373 ± 90	591 ± 55
Reacher-easy	20 ± 50	314 ± 155	-	538 ± 233	601 ± 213	567 ± 54	**722** ± 50
Cheetah-run	138 ± 88	235 ± 137	319 ± 56	299 ± 48	344 ± 67	381 ± 79	**413** ± 35
Walker-walk	224 ± 48	277 ± 12	361 ± 73	403 ± 24	612 ± 164	641 ± 89	**667** ± 147
Cup-catch	0 ± 0	246 ± 174	512 ± 110	769 ± 43	**913**± 53	666 ± 181	633 ± 241

- SUNRISE consistently improves the performance of RAD

Experimental Results on Atari

- Performance on Atari games at 100K interactions

Game	Human	Random	SimPLe [23]	CURL [41]	Rainbow [47]	SUNRISE
Alien	7127.7	227.8	616.9	558.2	789.0	**872.0**
Amidar	1719.5	5.8	88.0	**142.1**	118.5	122.6
Assault	742.0	222.4	527.2	**600.6**	413.0	594.8
Asterix	8503.3	210.0	**1128.3**	734.5	533.3	755.0
BankHeist	753.1	14.2	34.2	131.6	97.7	**266.7**
BattleZone	37187.5	2360.0	5184.4	14870.0	7833.3	**15700.0**
Boxing	12.1	0.1	**9.1**	1.2	0.6	6.7
Breakout	30.5	1.7	**16.4**	4.9	2.3	1.8
ChopperCommand	7387.8	811.0	**1246.9**	1058.5	590.0	1040.0
CrazyClimber	35829.4	10780.5	**62583.6**	12146.5	25426.7	22230.0
DemonAttack	1971.0	152.1	208.1	817.6	688.2	**919.8**
Freeway	29.6	0.0	20.3	26.7	28.7	**30.2**
Frostbite	4334.7	65.2	254.7	1181.3	1478.3	**2026.7**
Gopher	2412.5	257.6	771.0	**669.3**	348.7	654.7
Hero	30826.4	1027.0	2656.6	6279.3	3675.7	**8072.5**
Jamesbond	302.8	29.0	125.3	**471.0**	300.0	390.0
Kangaroo	3035.0	52.0	323.1	872.5	1060.0	**2000.0**
Krull	2665.5	1598.0	**4539.9**	4229.6	2592.1	3087.2
KungFuMaster	22736.3	258.5	**17257.2**	14307.8	8600.0	10306.7
MsPacman	6951.6	307.3	1480.0	1465.5	1118.7	**1482.3**
Pong	14.6	-20.7	**12.8**	-16.5	-19.0	-19.3
PrivateEye	69571.3	24.9	58.3	**218.4**	97.8	100.0
Qbert	13455.0	163.9	1288.8	1042.4	646.7	**1830.8**
RoadRunner	7845.0	11.5	5640.6	5661.0	9923.3	**11913.3**
Seaquest	42054.7	68.4	**683.3**	384.5	396.0	570.7
UpNDown	11693.2	533.4	3350.3	2955.2	3816.0	**5074.0**

Consistently outperform Rainbow

Experimental Results on Atari

- Performance on Atari games at 100K interactions

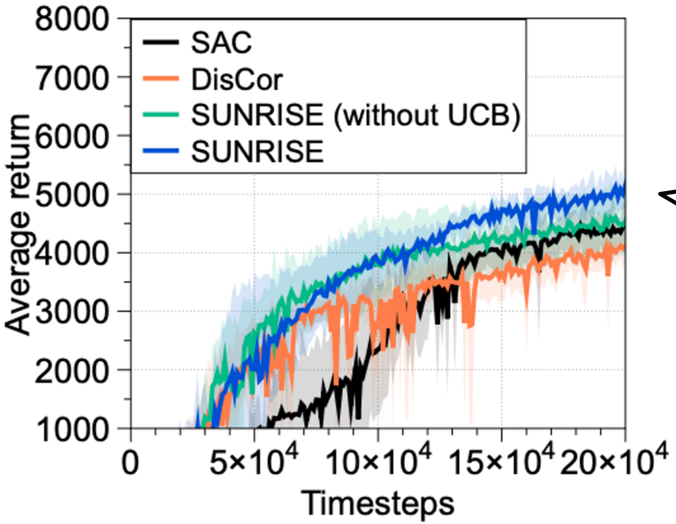
Game	Human	Random	SimPLe [23]	CURL [41]	Rainbow [47]	SUNRISE
Alien	7127.7	227.8	616.9	558.2	789.0	**872.0**
Amidar	1719.5	5.8	88.0	**142.1**	118.5	122.6
Assault	742.0	222.4	527.2	**600.6**	413.0	594.8
Asterix	8503.3	210.0	**1128.3**	734.5	533.3	755.0
BankHeist	753.1	14.2	34.2	131.6	97.7	**266.7**
BattleZone	37187.5	2360.0	5184.4	14870.0	7833.3	**15700.0**
Boxing	12.1	0.1	**9.1**	1.2	0.6	6.7
Breakout	30.5	1.7	**16.4**	4.9	2.3	1.8
ChopperCommand	7387.8	811.0	**1246.9**	1058.5	590.0	1040.0
CrazyClimber	35829.4	10780.5	**62583.6**	12146.5	25426.7	22230.0
DemonAttack	1971.0	152.1	208.1	817.6	688.2	**919.8**
Freeway	29.6	0.0	20.3	26.7	28.7	**30.2**
Frostbite	4334.7	65.2	254.7	1181.3	1478.3	**2026.7**
Gopher	2412.5	257.6	771.0	**669.3**	348.7	654.7
Hero	30826.4	1027.0	2656.6	6279.3	3675.7	**8072.5**
Jamesbond	302.8	29.0	125.3	**471.0**	300.0	390.0
Kangaroo	3035.0	52.0	323.1	872.5	1060.0	**2000.0**
Krull	2665.5	1598.0	**4539.9**	4229.6	2592.1	3087.2
KungFuMaster	22736.3	258.5	**17257.2**	14307.8	8600.0	10306.7
MsPacman	6951.6	307.3	1480.0	1465.5	1118.7	**1482.3**
Pong	14.6	-20.7	**12.8**	-16.5	-19.0	-19.3
PrivateEye	69571.3	24.9	58.3	**218.4**	97.8	100.0
Qbert	13455.0	163.9	1288.8	1042.4	646.7	**1830.8**
RoadRunner	7845.0	11.5	5640.6	5661.0	9923.3	**11913.3**
Seaquest	42054.7	68.4	**683.3**	384.5	396.0	570.7
UpNDown	11693.2	533.4	3350.3	2955.2	3816.0	**5074.0**

Consistently outperform Rainbow

SOTA on 13 out of 26 environments

Ablation Study

- Can weighted Bellman backup reduce error propagation?
  - Noisy-reward setting: r'(s,a) = r(s,a) + z, where z ~ N(0, 1)
  - Baseline: DisCor [Kumar et al., 2020], Weighted Bellman backup based on estimated cumulative Bellman errors



SUNRISE with weighted Bellman backups (green curve) reduces the error propagation

[Kumar et al., 2020] Kumar, A., Gupta, A. and Levine, S., DisCor: Corrective Feedback in Reinforcement Learning via Distribution Correction. arXiv preprint arXiv:2003.07305, 2020.

SUNRISE Take-Aways

- Ensembles can be used to prevent error propagation in Q-learning
- Future directions
  - Other applications: Offline RL, Imitation learning
  - Extension to on-policy learning
- Pre-print: https://arxiv.org/abs/2007.04938
- Code: https://github.com/pokaxpoka/sunrise

Thank you