TD3, SAC, TRPO, PPO

Milan Straka

November 23, 2020

EUROPEAN UNION
European Structural and Investment Fund
Operational Programme Research, Development and Education

Charles University in Prague
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics

Twin Delayed Deep Deterministic Policy Gradient

The paper Addressing Function Approximation Error in Actor-Critic Methods by Scott Fujimoto et al. from February 2018 proposes improvements to DDPG which

- decrease maximization bias by training two critics and choosing the minimum of their predictions;
- introduce several variance-lowering optimizations:
  - delayed policy updates;
  - target policy smoothing.

The TD3 algorithm has been together with SAC one of the state-of-the-art algorithms for off-policy continuous-actions RL training (as of 2020).

TD3 – Maximization Bias

Similarly to Q-learning, the DDPG algorithm suffers from maximization bias. In Q-learning, the maximization bias was caused by the explicit $\max$ operator. For DDPG methods, it can be caused by the gradient descent itself. Let $\boldsymbol\theta_{approx}$ be the parameters maximizing the $q_{\boldsymbol\theta}$ and let $\boldsymbol\theta_{true}$ be the hypothetical parameters which maximise true q_π, and let π_{approx} and π_{true} denote the corresponding policies.

Because the gradient direction is a local maximizer, for sufficiently small $\alpha < \varepsilon_1$ we have

$$\mathbb{E}\big[q_{\boldsymbol\theta}(s, \pi_{approx})\big] \geq \mathbb{E}\big[q_{\boldsymbol\theta}(s, \pi_{true})\big].$$

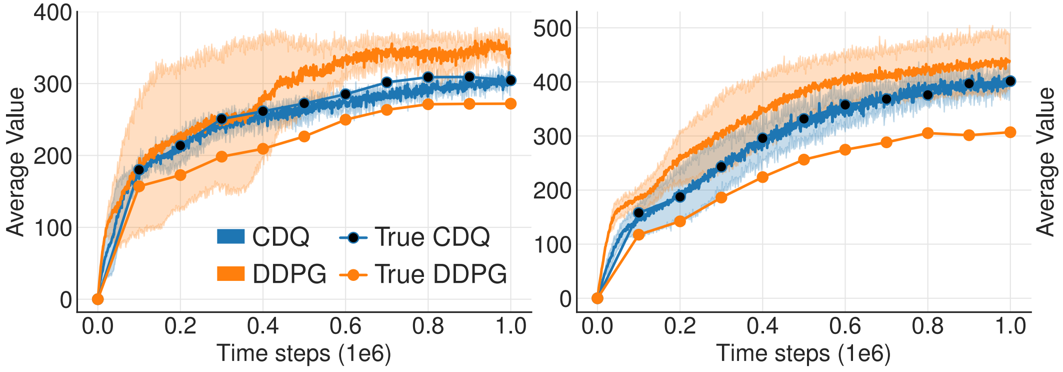
However, for real q_π and for sufficiently small $\alpha < \varepsilon_2$ it holds that

$$\mathbb{E}\big[q_\pi(s, \pi_{true})\big] \geq \mathbb{E}\big[q_\pi(s, \pi_{approx})\big].$$

Therefore, if $\mathbb{E}\big[q_{\boldsymbol\theta}(s, \pi_{true})\big] \geq \mathbb{E}\big[q_\pi(s, \pi_{true})\big]$, for $\alpha < \min(\varepsilon_1, \varepsilon_2)$

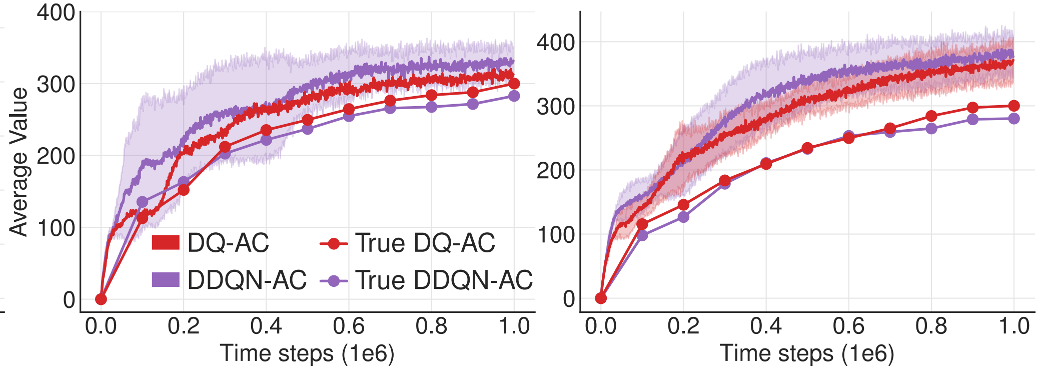
$$\mathbb{E}\big[q_{\boldsymbol\theta}(s, \pi_{approx})\big] \geq \mathbb{E}\big[q_\pi(s, \pi_{approx})\big].$$

TD3 – Maximization Bias



(a) Hopper-v1 (b) Walker2d-v1

Figure 1 of the paper "Addressing Function Approximation Error in Actor-Critic Methods" by Scott Fujimoto et al.



(a) Hopper-v1 (b) Walker2d-v1

Figure 2 of the paper "Addressing Function Approximation Error in Actor-Critic Methods" by Scott Fujimoto et al.

Analogously to Double DQN we could compute the learning targets using the current policy and the target critic, i.e., $r + \gamma q_{\boldsymbol\theta'}(s', \pi_{\boldsymbol\varphi}(s'))$ (instead of using target policy and target critic as in DDPG), obtaining DDQN-AC algorithm. However, the authors found out that the policy changes too slowly and the target and current networks are too similar.

Using the original Double Q-learning, two pairs of actors and critics could be used, with the learning targets computed by the opposite critic, i.e., $r + \gamma q_{\boldsymbol\theta_2}(s', \pi_{\boldsymbol\varphi_1}(s'))$ for updating $q_{\boldsymbol\theta_1}$. The resulting DQ-AC algorithm is slightly better, but still suffering from overestimation.

TD3 – Algorithm

The authors instead suggest to employ two critics and one actor. The actor is trained using one of the critics, and both critics are trained using the same target computed using the *minimum* value of both critics as

$$r + \gamma \min_{i=1,2} q_{\boldsymbol\theta'_i}(s', \pi_{\boldsymbol\varphi'}(s')).$$

Furthermore, the authors suggest two additional improvements for variance reduction.

- For obtaining higher quality target values, the authors propose to train the critics more often. Therefore, critics are updated each step, but the actor and the target networks are updated only every d-th step ($d = 2$ is used in the paper).
- To explicitly model that similar actions should lead to similar results, a small random noise is added to performed actions when computing the target value:

$$r + \gamma \min_{i=1,2} q_{\boldsymbol\theta'_i}(s', \pi_{\boldsymbol\varphi'}(s') + \varepsilon) \quad \text{for} \quad \varepsilon \sim \operatorname{clip}(\mathcal{N}(0, \sigma), -c, c).$$

TD3 – Algorithm

Algorithm 1 TD3

Initialize critic networks Q_{θ_1}, Q_{θ_2}, and actor network π_ϕ with random parameters θ_1, θ_2, ϕ
Initialize target networks $\theta'_1 \leftarrow \theta_1$, $\theta'_2 \leftarrow \theta_2$, $\phi' \leftarrow \phi$
Initialize replay buffer $\mathcal{B}$
for $t = 1$ **to** T **do**
  Select action with exploration noise $a \sim \pi_\phi(s) + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma)$ and observe reward r and new state s'
  Store transition tuple (s, a, r, s') in $\mathcal{B}$

  Sample mini-batch of N transitions (s, a, r, s') from $\mathcal{B}$
  $\tilde{a} \leftarrow \pi_{\phi'}(s') + \epsilon, \quad \epsilon \sim \operatorname{clip}(\mathcal{N}(0, \tilde{\sigma}), -c, c)$
  $y \leftarrow r + \gamma \min_{i=1,2} Q_{\theta'_i}(s', \tilde{a})$
  Update critics $\theta_i \leftarrow \operatorname{argmin}_{\theta_i} N^{-1} \sum (y - Q_{\theta_i}(s, a))^2$
  if t mod d **then**
    Update ϕ by the deterministic policy gradient:
    $\nabla_\phi J(\phi) = N^{-1} \sum \nabla_a Q_{\theta_1}(s, a)|_{a=\pi_\phi(s)} \nabla_\phi \pi_\phi(s)$
    Update target networks:
    $\theta'_i \leftarrow \tau \theta_i + (1 - \tau)\theta'_i$
    $\phi' \leftarrow \tau \phi + (1 - \tau)\phi'$
  end if
end for

Algorithm 1 of the paper "Addressing Function Approximation Error in Actor-Critic Methods" by Scott Fujimoto et al.

Hyper-parameter	Ours	DDPG
Critic Learning Rate	10^{-3}	10^{-3}
Critic Regularization	None	$10^{-2} \cdot \|\|\theta\|\|^2$
Actor Learning Rate	10^{-3}	10^{-4}
Actor Regularization	None	None
Optimizer	Adam	Adam
Target Update Rate (τ)	$5 \cdot 10^{-3}$	10^{-3}
Batch Size	100	64
Iterations per time step	1	1
Discount Factor	0.99	0.99
Reward Scaling	1.0	1.0
Normalized Observations	False	True
Gradient Clipping	False	False
Exploration Policy	$\mathcal{N}(0, 0.1)$	OU, $\theta = 0.15, \mu = 0, \sigma = 0.2$

Table 3 of the paper "Addressing Function Approximation Error in Actor-Critic Methods" by Scott Fujimoto et al.

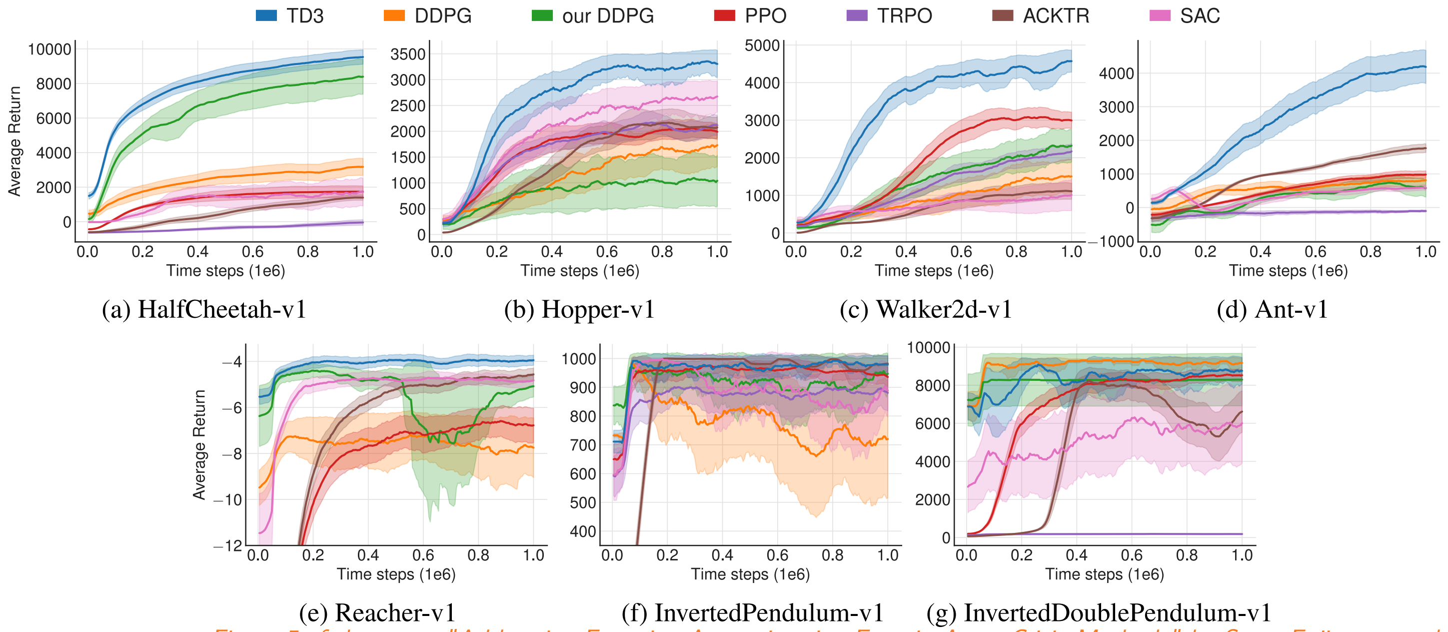


Figure 5 of the paper "Addressing Function Approximation Error in Actor-Critic Methods" by Scott Fujimoto et al.

Environment	TD3	DDPG	Our DDPG	PPO	TRPO	ACKTR	SAC
HalfCheetah	**9636.95 ± 859.065**	3305.60	8577.29	1795.43	-15.57	1450.46	2347.19
Hopper	**3564.07 ± 114.74**	2020.46	1860.02	2164.70	2471.30	2428.39	2996.66
Walker2d	**4682.82 ± 539.64**	1843.85	3098.11	3317.69	2321.47	1216.70	1283.67
Ant	**4372.44 ± 1000.33**	1005.30	888.77	1083.20	-75.85	1821.94	655.35
Reacher	**-3.60 ± 0.56**	-6.51	**-4.01**	-6.18	-111.43	-4.26	-4.44
InvPendulum	**1000.00 ± 0.00**	**1000.00**	**1000.00**	**1000.00**	985.40	**1000.00**	**1000.00**
InvDoublePendulum	**9337.47 ± 14.96**	**9355.52**	8369.95	8977.94	205.85	9081.92	8487.15

Table 1 of the paper "Addressing Function Approximation Error in Actor-Critic Methods" by Scott Fujimoto et al.

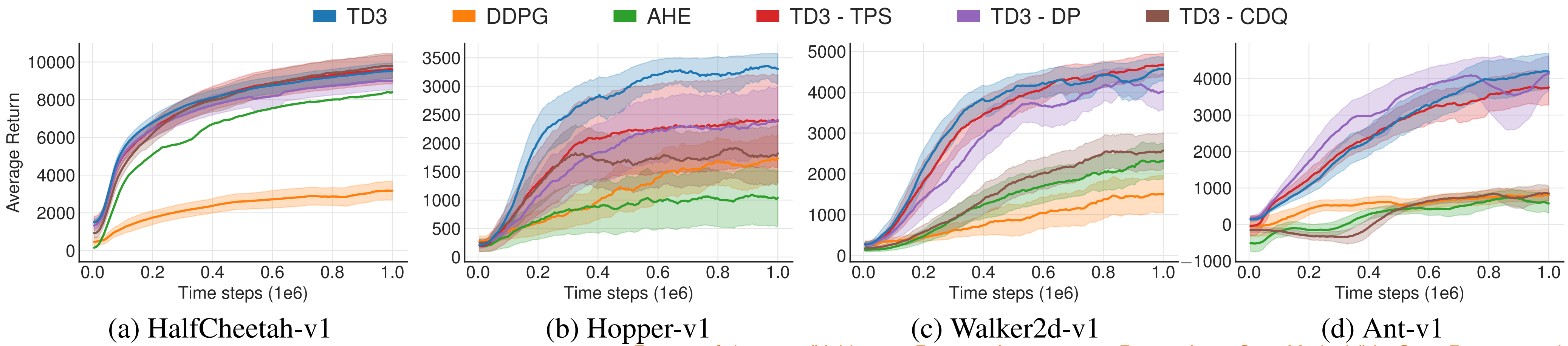


Figure 7 of the paper "Addressing Function Approximation Error in Actor-Critic Methods" by Scott Fujimoto et al.

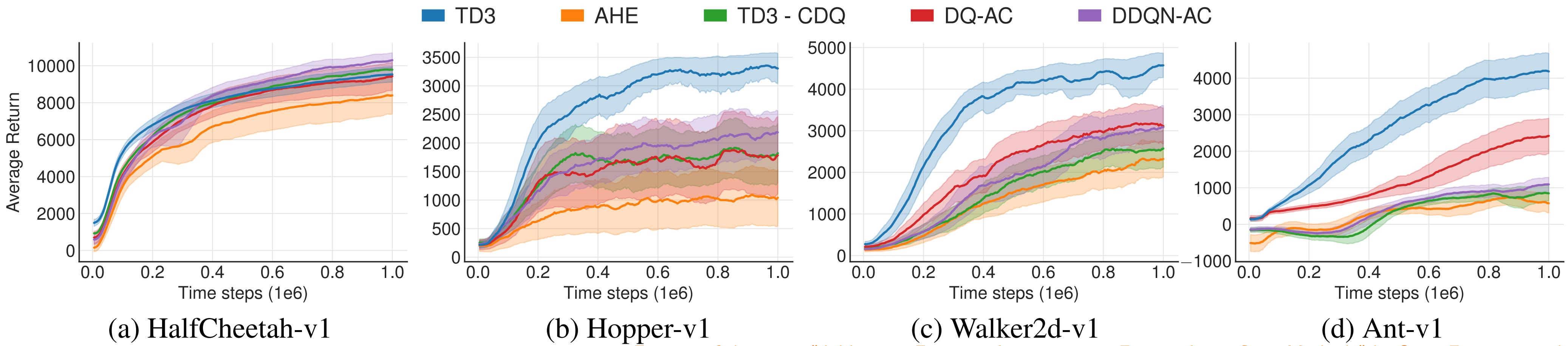


Figure 8 of the paper "Addressing Function Approximation Error in Actor-Critic Methods" by Scott Fujimoto et al.

TD3 – Ablations

Method	HCheetah	Hopper	Walker2d	Ant
TD3	9532.99	**3304.75**	**4565.24**	**4185.06**
DDPG	3162.50	1731.94	1520.90	816.35
AHE	8401.02	1061.77	2362.13	564.07
AHE + DP	7588.64	1465.11	2459.53	896.13
AHE + TPS	9023.40	907.56	2961.36	872.17
AHE + CDQ	6470.20	1134.14	3979.21	3818.71
TD3 - DP	9590.65	2407.42	**4695.50**	3754.26
TD3 - TPS	8987.69	2392.59	4033.67	**4155.24**
TD3 - CDQ	9792.80	1837.32	2579.39	849.75
DQ-AC	9433.87	1773.71	3100.45	2445.97
DDQN-AC	**10306.90**	2155.75	3116.81	1092.18

Table 2 of the paper "Addressing Function Approximation Error in Actor-Critic Methods" by Scott Fujimoto et al.

Soft Actor Critic

The paper Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor by Tuomas Haarnoja et al. introduces a different off-policy algorithm for continuous action space.

The general idea is to introduce entropy directly in the value function we want to maximize.

TO BE FINISHED LATER

Algorithm 1 Soft Actor-Critic

Initialize parameter vectors ψ, $\bar{\psi}$, θ, ϕ.
for each iteration **do**
  for each environment step **do**
    $\mathbf{a}_t \sim \pi_\phi(\mathbf{a}_t|\mathbf{s}_t)$
    $\mathbf{s}_{t+1} \sim p(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{s}_t, \mathbf{a}_t, r(\mathbf{s}_t, \mathbf{a}_t), \mathbf{s}_{t+1})\}$
  end for
  for each gradient step **do**
    $\psi \leftarrow \psi - \lambda_V \hat{\nabla}_\psi J_V(\psi)$
    $\theta_i \leftarrow \theta_i - \lambda_Q \hat{\nabla}_{\theta_i} J_Q(\theta_i)$ for $i \in \{1, 2\}$
    $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_\phi J_\pi(\phi)$
    $\bar{\psi} \leftarrow \tau\psi + (1 - \tau)\bar{\psi}$
  end for
end for

Algorithm 1 of the paper "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor" by Tuomas Haarnoja et al.

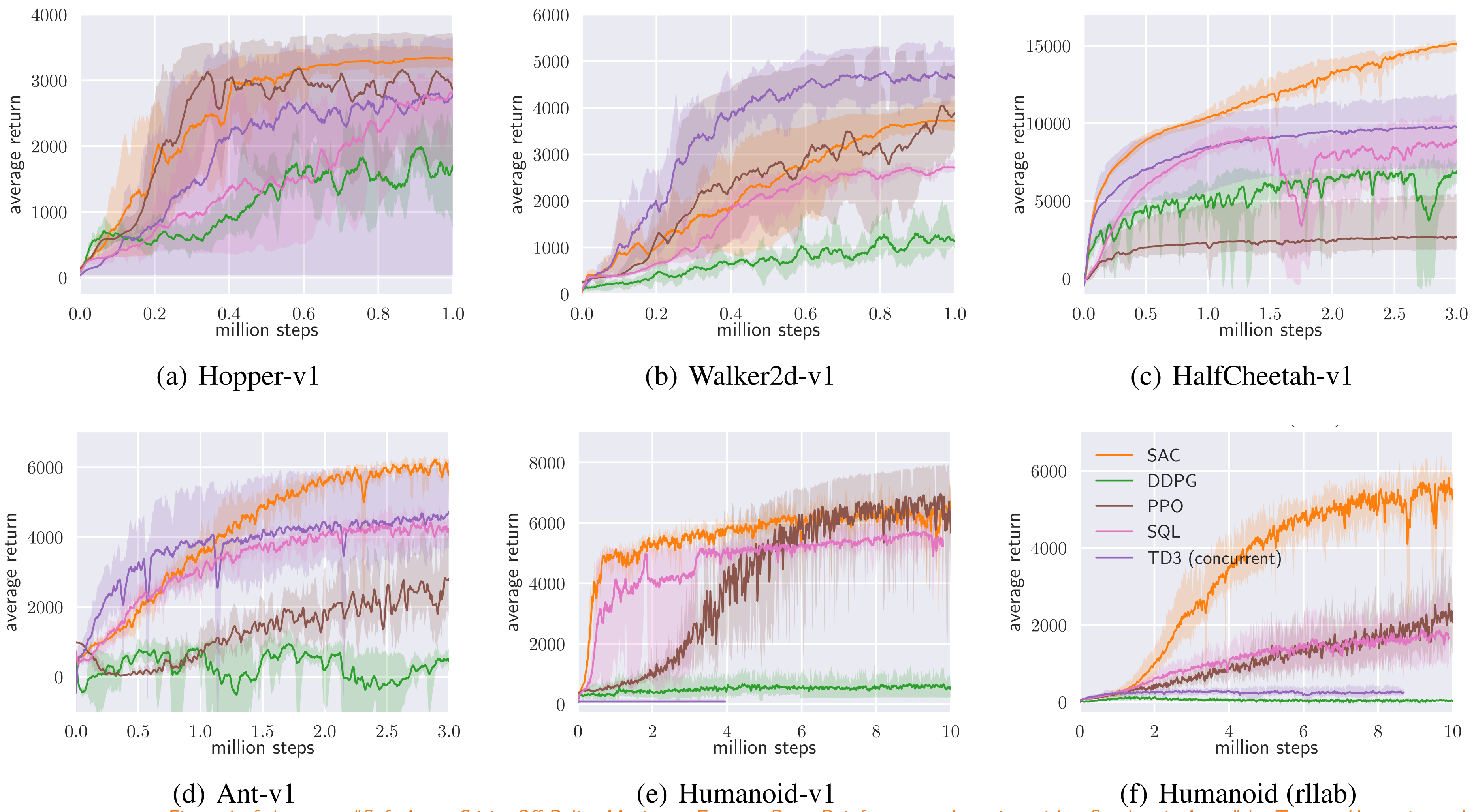


Figure 1 of the paper "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor" by Tuomas Haarnoja et al.