Evolution + deep RL

Using evolution to optimize actions

Olivier Sigaud

Sorbonne Université
http://people.isir.upmc.fr/sigaud

Evolving actions to improve performance: overview

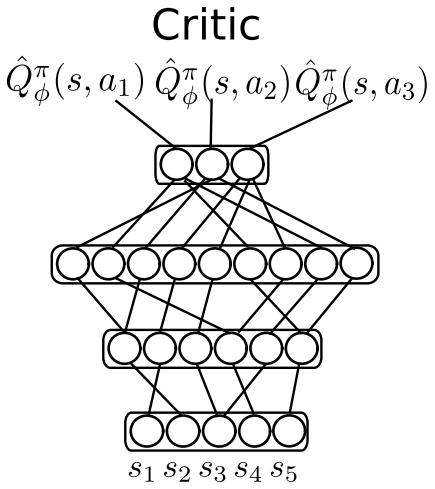
Prop. / Algo.	Critic update	Action Selection	Policy Update
QT-OPT Kalashnikov et al. (2018)	$\bar{a}_t$ =CEM (random, 64, 6, 2)	$\bar{a}_t$ =CEM (random, 64, 6, 2)	No policy
CGP Simmons-Edler et al. (2019)	$\bar{a}_t$ =CEM (random, 64, 6, 2)	$\bar{a}_t = \pi(s_t)$	BC or DPG
EAS-RL Ma et al. (2022)	$\bar{a}_t$ =PSO (10,10)	$\bar{a}_t = \pi(s_t)$	BC + DPG
SAC-CEPO Shi and Singh (2021)	SAC update	$\bar{a}_t$ =CEM (π, 60 →140, 3% →7%, 6 →14)	BC
GRAC Shao et al. (2021)	$\bar{a}_t$ =CEM (π, 256, 5, 2)	$\bar{a}_t$ =CEM (π, 256, 5, 2)	PG with two losses
ZOSPI Sun et al. (2020)	DDPG update	$\bar{a}_t = \pi(s_t)$ + perturb. network	BC($\bar{a}_t$ = argmax($random$, 50))

- Policy params and actions are vectors of numbers which can be optimized
- Actions are smaller vectors than policies, thus they are easier to optimize
- Here we optimize actions within the RL loop

Sigaud, O. (2022) Combining evolution and deep reinforcement learning for policy search: a survey. *arXiv preprint arXiv:2203.14009*

The Q-network in DQN

state / action	a_0	a_1	a_2	a_3
$\mathbf{s}_0$	0.66	0.88*	0.81	0.73
$\mathbf{s}_1$	0.73	0.63	0.9*	0.43
$\mathbf{s}_2$	0.73	0.9	0.95*	0.73
$\mathbf{s}_3$	0.81	0.9	1.0*	0.81
$\mathbf{s}_4$	0.81	1.0*	0.81	0.9
$\mathbf{s}_5$	0.9	1.0*	0.0	0.9



DQN

- Parametrized representation of the critic $\hat{Q}^{\pi_{\boldsymbol{\theta}}}_{\boldsymbol{\phi}}(\mathbf{s}_t, \mathbf{a}_t)$
- Q-network equivalent to the Q-Table (with an infinity of state rows)
- For each observed $(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})$:

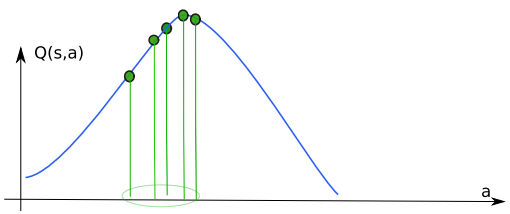
$$Q(\mathbf{s}_t, \mathbf{a}_t) \leftarrow Q(\mathbf{s}_t, \mathbf{a}_t) + \alpha[r_t + \gamma \max_{\mathbf{a} \in A} Q(\mathbf{s}_{t+1}, \mathbf{a}) - Q(\mathbf{s}_t, \mathbf{a}_t)]$$

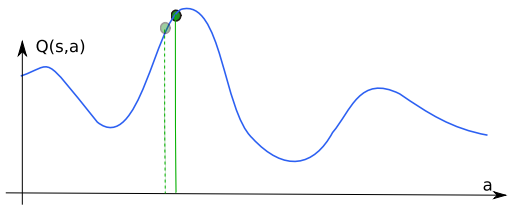
- Select action by finding $\arg\max_{\mathbf{a} \in A} Q(\mathbf{s}, \mathbf{a})$ (as in Q-LEARNING)
- Limitation: requires one output neuron per action

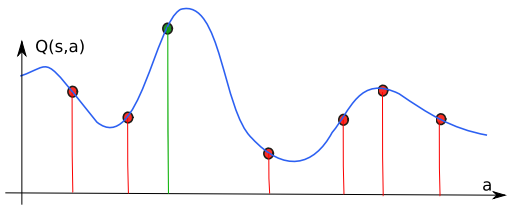
Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015) Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529–533.

Moving to continuous actions

- Starting from DQN, and considering continuous actions
- Two things become too hard:
  - Selecting actions by finding $\arg\max_{\mathbf{a}\in A} Q(\mathbf{s},\mathbf{a})$
  - Computing $\max_{\mathbf{a}\in A} Q(\mathbf{s}_{t+1},\mathbf{a})$ in the update rule



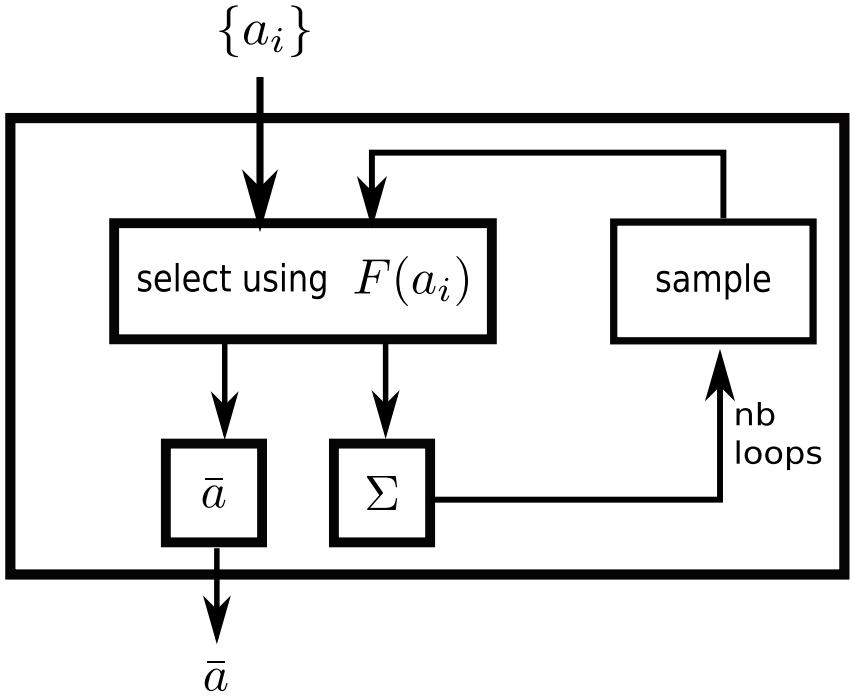




- Three classes of solutions
  1. Use an easily optimized model (e.g. convex) (NAF, Wang et al. 2016)
  2. DDPG: train a side estimator of the best action (also true of SAC)
  3. Sample a limited set of actions (QT-OPT, Kalashnikov et al., 2018)
- Here we focus on the third class of solutions

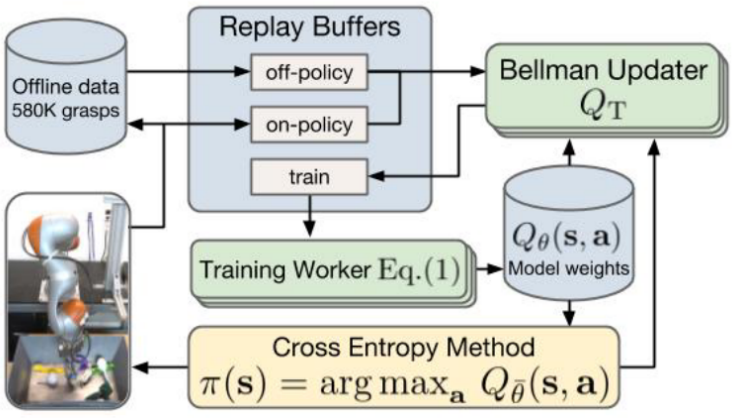
The CEM component

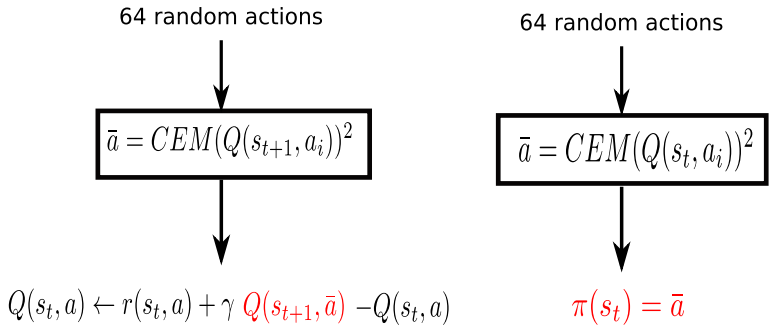
- We pack the CEM algorithm into a compact notation



- Summarized as: $\bar{a} = CEM(F(a_i))^{nb}$
- The notation is useful for presenting the next algorithms

QT-Opt

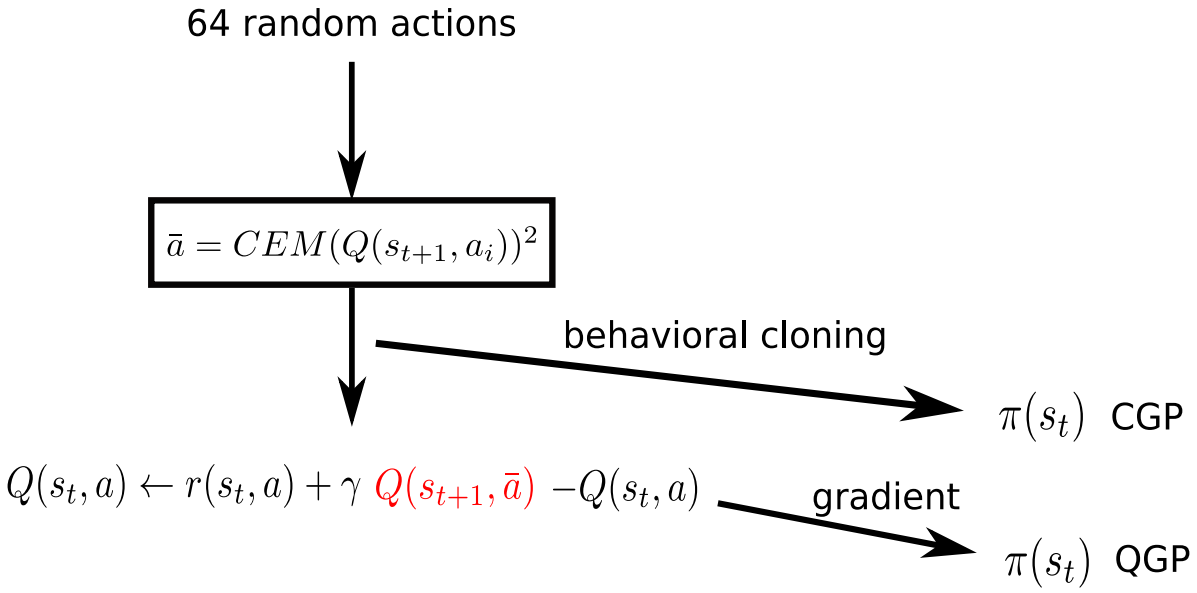




- CEM is used to draw actions for Bellman update + acting in the world
- Action in $\mathbb{R}^5 \times 3$bits
- Pop size $= 64$, take the best 6 to define the next covariance matrix

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Qt-Opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018

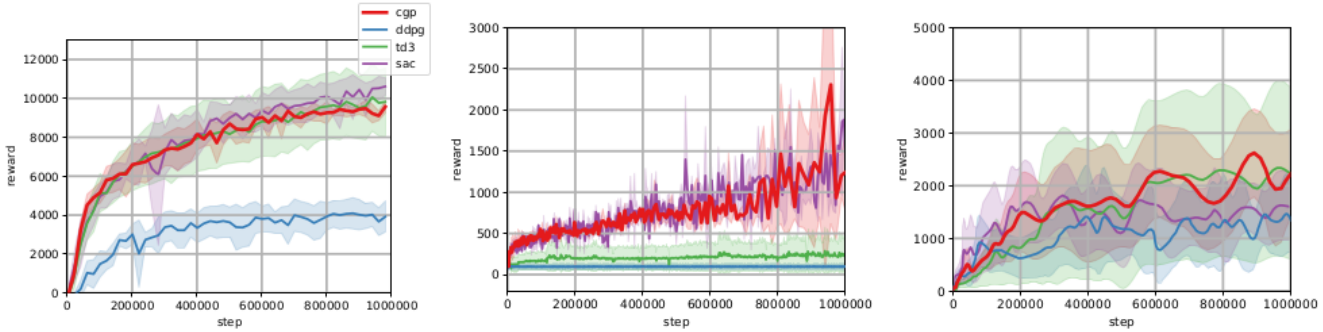
Cross-Entropy Guided Policies (CGP)



- Main point: in QT-OPT, action inference is too slow
- The Q-function is updated from CEM sampling, as in QT-OPT
- But the actor is learned:
  - From the gradient of the critic, as in DDPG or TD3 (QPG)
  - Through behavioral cloning of CEM actions (CGP). This option works better.

Simmons-Edler, R., Eisner, B., Mitchell, E., Seung, S., and Lee, D. Q-learning for continuous actions with cross-entropy guided policies. *arXiv preprint arXiv:1903.10605*, 2019

CGP learning curves



(a) HalfCheetah-v2 (b) Humanoid-v2 (c) Hopper-v2

(d) Pusher-v2 (e) Ant-v2

- 4 seeds
- Not compared to QT-OPT
- The gain with respect to TD3 is not spectacular

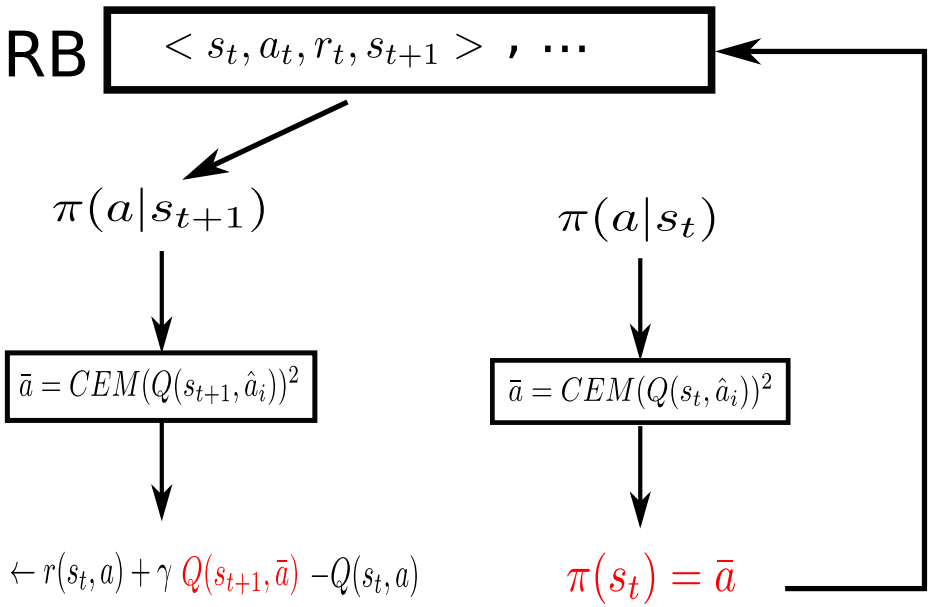
CGP running time

Table 1. Runtime in average seconds per episode of HalfCheetah-v2 (without rendering) on an otherwise-idle machine with a Nvidia GTX 1080 ti GPU. CGP achieves a constant inference runtime independent of the number of CEM iterations used, which matches the performance of other methods.

METHOD	MEAN TRAIN (S)	MEAN INFERENCE (S)
RANDOM	-	0.48
DDPG	5.75	2.32
TD3	5.67	2.35
SAC	11.00	2.35
CEM-2	7.1	6.3
CEM-4	9.3	10.1
CGP-2	11.03	2.35
CGP-4	14.46	2.35

- Results on HalfCheetah-v2
- Longer to train, but not longer at inference (by contrast with QT-OPT)

GRAC: Self-guided Policy Improvement with Evolution Strategies

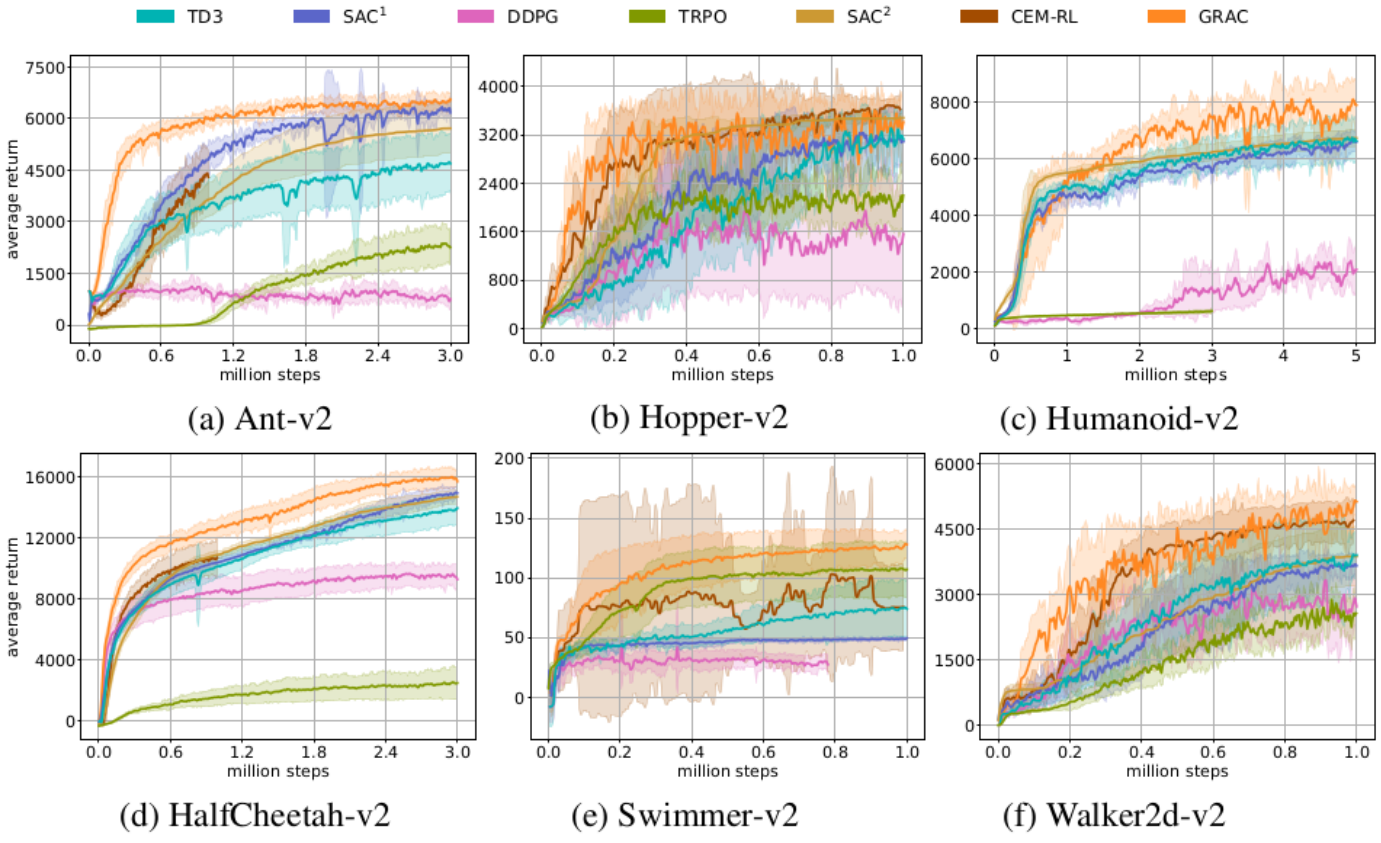


- Idea: improve the action choice of the policy using CEM from $\pi(.|s)$
- Combines the DDPG and the QT-OPT approaches (related to TD-MPC)
- The initial actions $\hat{\mathbf{a}}_t$ are sampled from the current policy
- Only update the critic if $Q(\mathbf{s}_t, \bar{\mathbf{a}}_t) - Q(\mathbf{s}_t, \hat{\mathbf{a}}_t) > 0$

Shao, L., You, Y., Yan, M., Yuan, S., Sun, Q., and Bohg, J. GRAC: Self-guided and self-regularized actor-critic. In *Conference on Robot Learning*, pp. 267–276. PMLR, 2021

Hansen, N., Wang, X., and Su, H. (2022) Temporal difference learning for model predictive control. *arXiv preprint arXiv:2203.04955*

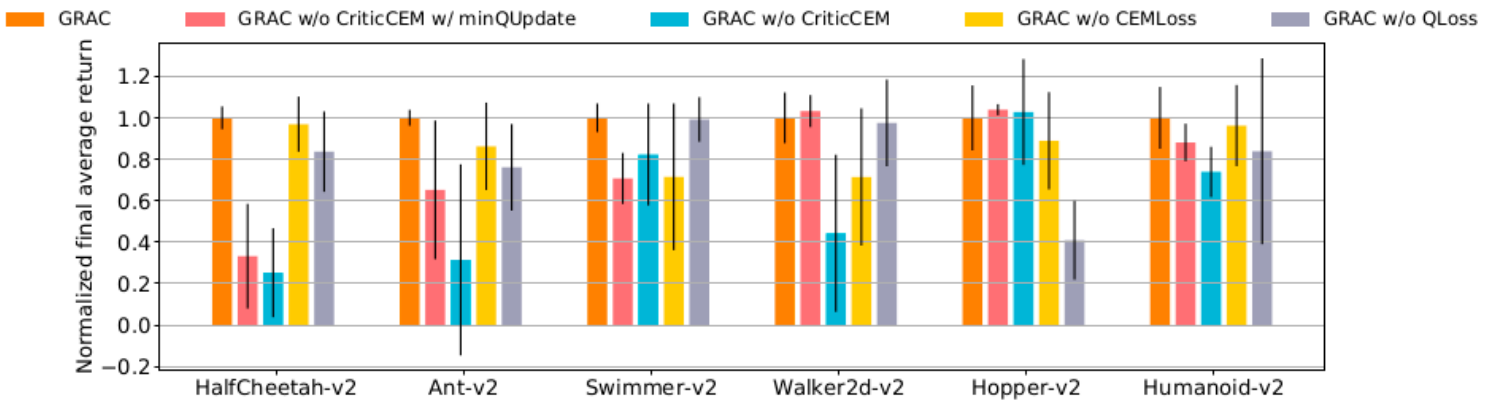
GRAC learning curves



- 10 seeds (after review)
- Pop size 256, keep 5 bests, 2 iterations of CEM
- Compared to CEMRL, but not to QT-OPT, CGP or other similar algos

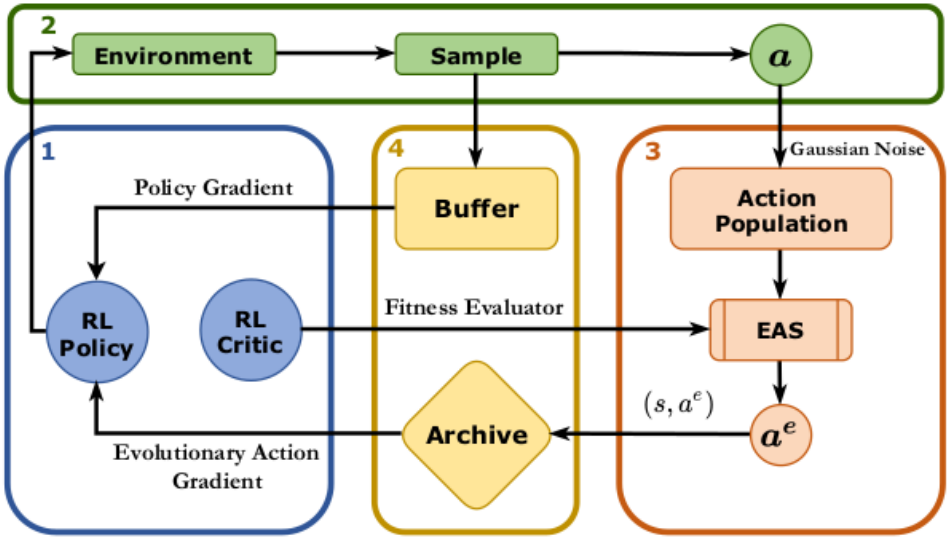
GRAC ablations

- Three contributions:
  - Self-guided Policy Improvement with Evolution Strategies
  - Self-regularized TD Learning (very interesting, but out of scope)
  - Max-min Double Q-Learning (idem)



- 4 seeds, average of the 10 last evaluations
- The various components of GRAC complement each other

EAS-RL

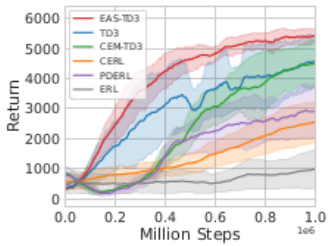


- Use Particle Swarm Optimization (PSO) rather than CEM (a sort of CEM with inertia and acceleration...)
- The evolutionary actions are used for behavioral cloning
- As in GRAC, only update if $Q(\mathbf{s}_t, \bar{\mathbf{a}}_t) - Q(\mathbf{s}_t, \hat{\mathbf{a}}_t) > 0$ (Q-filtering)
- Otherwise, use standard DPG as in TD3

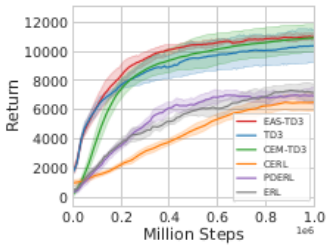
Ma, Y., Liu, T., Wei, B., Liu, Y., Xu, K., and Li, W. Evolutionary action selection for gradient-based policy learning. *arXiv preprint arXiv:2201.04286*, 2022

EAS-RL learning curves

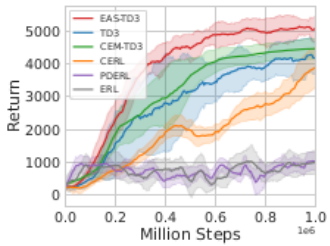
Environment	Ant	HalfCheetah	Walker2d	Hopper	Humanoid	AntBullet	W2DBullet
State Dimension	111	17	17	11	376	28	22
Action Dimension	8	6	6	3	17	8	6



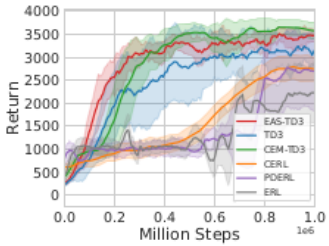
(a) Ant-v3



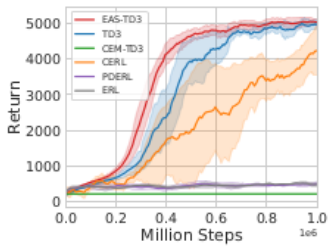
(b) HalfCheetah-v3



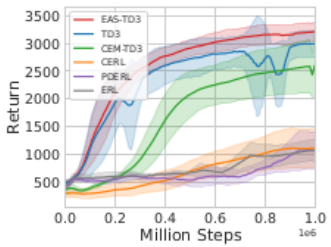
(c) Walker2d-v3



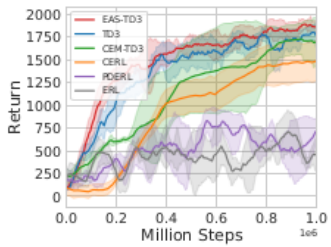
(d) Hopper-v3



(e) Humanoid-v3



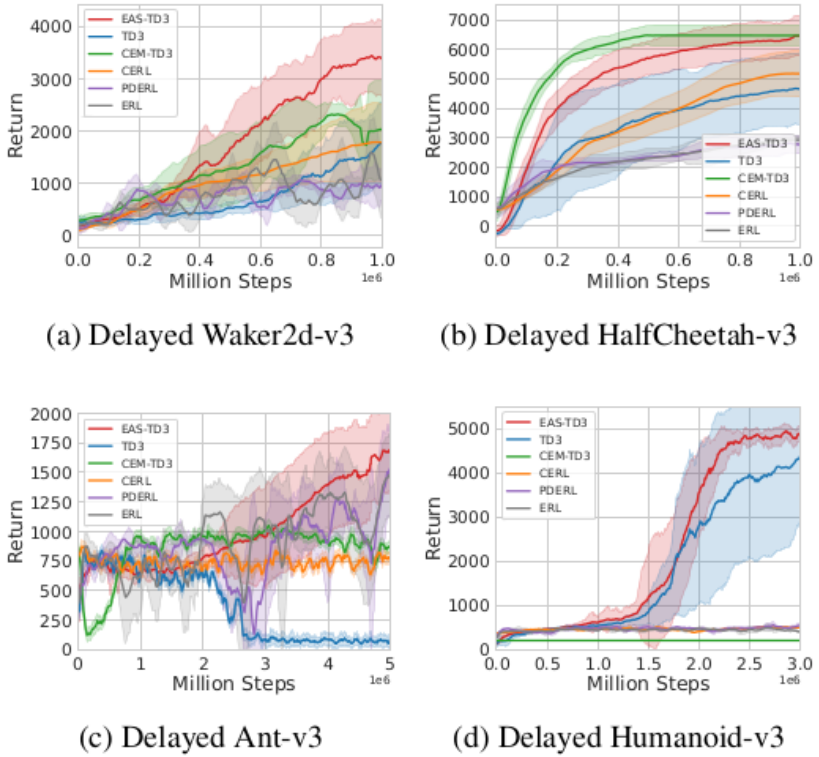
(f) AntBulletEnv-v0



(g) Walker2DBulletEnv-v0

- Number of seeds not found
- Useful for comparison with other papers

EAS-RL in delayed environments



(a) Delayed Waker2d-v3 (b) Delayed HalfCheetah-v3

(c) Delayed Ant-v3 (d) Delayed Humanoid-v3

- CEMRL works well, apart from Humanoid...
- Useful for comparison with other papers

Any question?

Send mail to: Olivier.Sigaud@isir.upmc.fr

Hansen, N., Wang, X., and Su, H. (2022).

Temporal difference learning for model predictive control.
arXiv preprint arXiv:2203.04955.

Kalashnikov, D., Irpan, A., Pastor, P., Ibarz, J., Herzog, A., Jang, E., Quillen, D., Holly, E., Kalakrishnan, M., Vanhoucke, V., et al. (2018).

QT-Opt: Scalable deep reinforcement learning for vision-based robotic manipulation.
arXiv preprint arXiv:1806.10293.

Ma, Y., Liu, T., Wei, B., Liu, Y., Xu, K., and Li, W. (2022).

Evolutionary action selection for gradient-based policy learning.
arXiv preprint arXiv:2201.04286.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015).

Human-level control through deep reinforcement learning.
Nature, 518(7540):529–533.

Shao, L., You, Y., Yan, M., Yuan, S., Sun, Q., and Bohg, J. (2021).

GRAC: Self-guided and self-regularized actor-critic.
In *Conference on Robot Learning*, pages 267–276. PMLR.

Sigaud, O. (2022).

Combining evolution and deep reinforcement learning for policy search: a survey.
ACM Transactions in Evolutionary Learning and Optimization.

Simmons-Edler, R., Eisner, B., Mitchell, E., Seung, S., and Lee, D. (2019).

Q-learning for continuous actions with cross-entropy guided policies.
arXiv preprint arXiv:1903.10605.