Exploring Munchausen Reinforcement Learning

Team 10
Mohamed Amine Ketata
Konstantin Pervunin

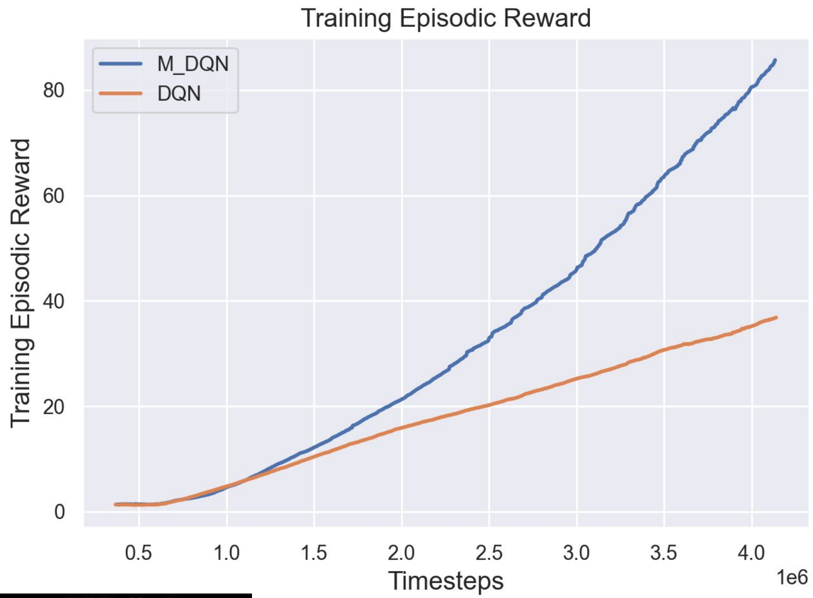
Tutor: Johannes Tenhumberg

Core Idea of Munchausen-RL [1]

- Use the current policy for bootstrapping.

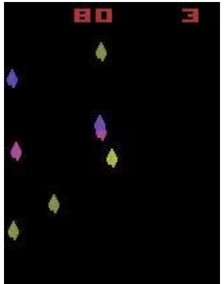
- Replace r_t by $r_t + \alpha \ln \pi(a_t \mid s_t)$ in any TD scheme.

DQN vs. Munchausen-DQN: Results on Atari Games



Breakout

Asteroids



M-RL for continuous action spaces: Munchausen-SAC

- SAC [2] target for the Q-function:

$$y_{SAC}\left(r, s', d\right) = r + \gamma(1-d)\left(\min_{i=1,2} Q_{\phi_{t\arg,i}}\left(s', \tilde{a}'\right) - \alpha log\pi_\theta\left(\tilde{a}' \mid s'\right)\right), \ \tilde{a}' \sim \pi_\theta\left(. \mid s'\right)$$

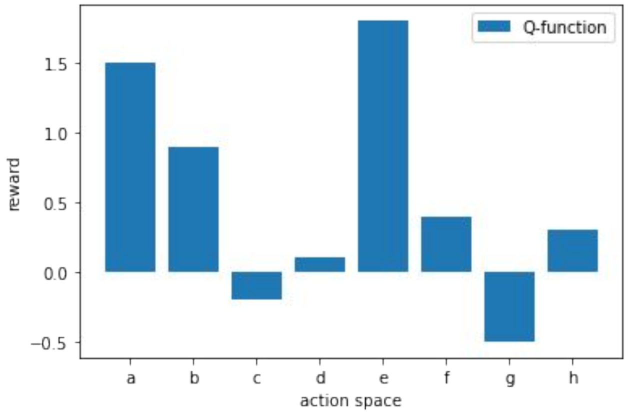
- M-SAC target for the Q-function:

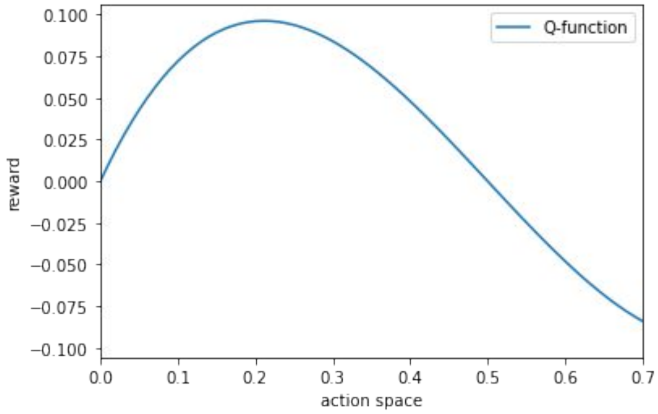
$$y_{M-SAC}\left(r, s', d\right) = r + \left[\tau\alpha log\pi_\theta(a \mid s)\right]_{l0}^{0} + \gamma(1-d)\left(\min_{i=1,2} Q_{\phi_{t\arg,i}}\left(s', \tilde{a}'\right) - \alpha log\pi_\theta\left(\tilde{a}' \mid s'\right)\right), \ \tilde{a}' \sim \pi_\theta\left(. \mid s'\right)$$

M-SAC specific hyperparameters: τ and $l0$

Action gap for continuous action space

Original definition: **action gap** = difference between optimal and second best predicted rewards





Problem: **action gap** does not exceed 0.

Action gap for continuous action space

Action gap describes how confident is the agent in the optimality of the selected action.

SAC algorithm has 2 networks: actor and critic.

Actor oriented action gap - confidence in the choice of the action.

Critic oriented action gap - confidence in the maximality of the expected reward.

Actor oriented action gap

Generate distorted actions:

- add random noise to the weights of the actor network
- predict actions for given states
- remove added noise

Define actor oriented action gap as:

$$AG_{actor} = \frac{1}{N} \cdot \sum_{i=1}^{N} ||a_i - d_i||_2$$

where a_i is a “real” action and d_i is a distorted action for the state s_i

Noise generation

Purpose: scale the noise according to the weights.

Noise must be

- proportional to the norm
- inversely proportional to the size

Under these conditions, the norm of the matrix before adding noise is close to the norm after adding noise.

Critic oriented action gap

Generate distorted actions as for the Actor oriented action gap.

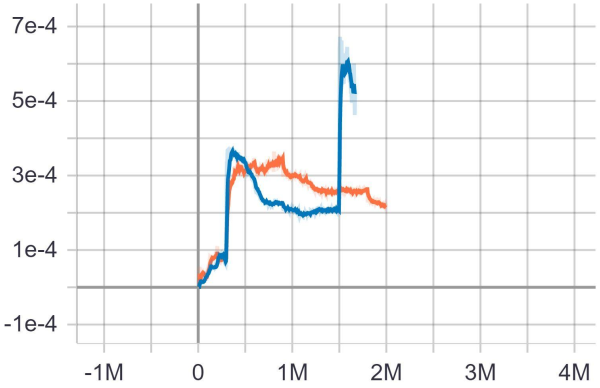
Instead of action space consider the distorted action and the optimal action (generated by actor network without noise).

Use the similar definition as for the discrete action gap:

$$AG_{critic} = \frac{1}{N} \cdot \sum_{i=1}^{N} Q(s_i, a_i) \; - \; Q(s_i, d_i)$$

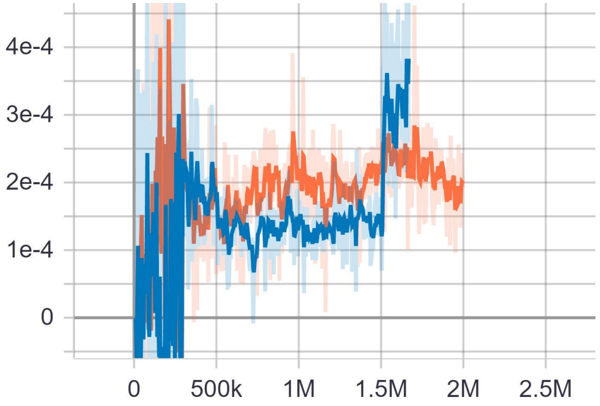
where s_i are states, a_i is a “real” action and d_i is a distorted action for the state s_i

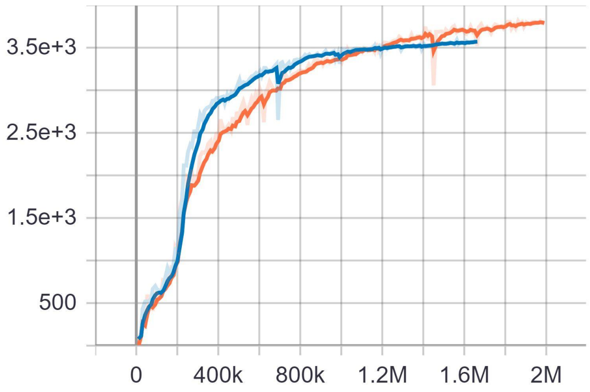
Intermediate results



AntBulletEnv-v0

Munchausen-SAC vs SAC



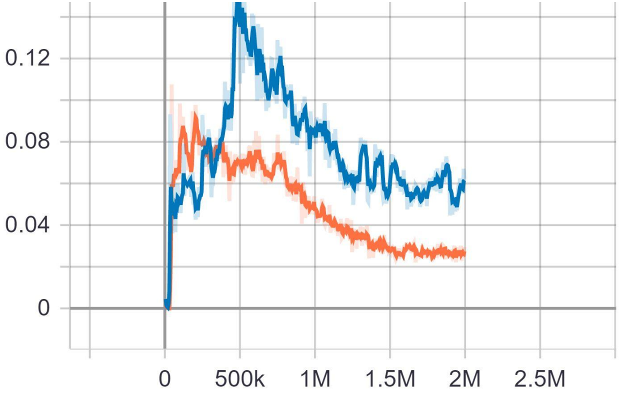


top-left: AG_{actor}

bottom-left: AG_{critic}

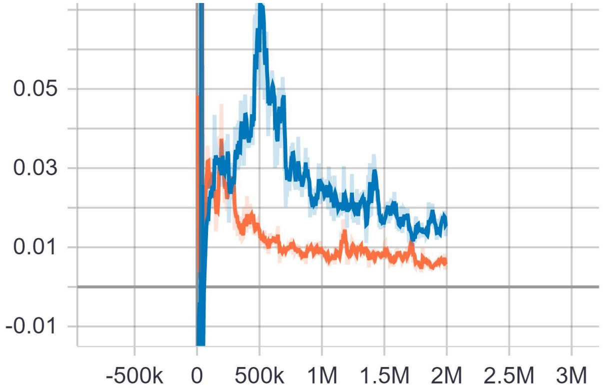
bottom-right: mean reward

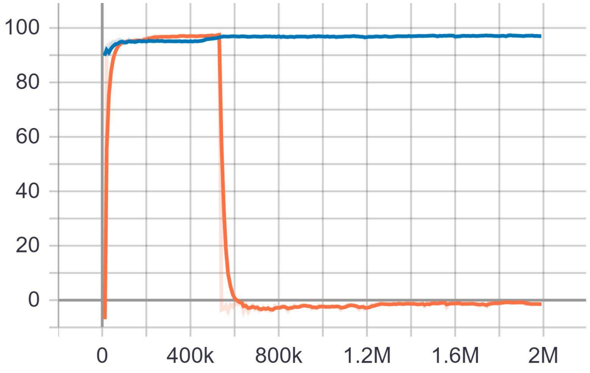
Intermediate results [2]



MountainCarContinuous-v0

Munchausen-SAC vs SAC





top-left: AG_{actor}

bottom-left: AG_{critic}

bottom-right: mean reward

Weighted difference of Q-values as action gap

Problem: AG_{actor} and AG_{critic} correlate but describe contradicting properties.

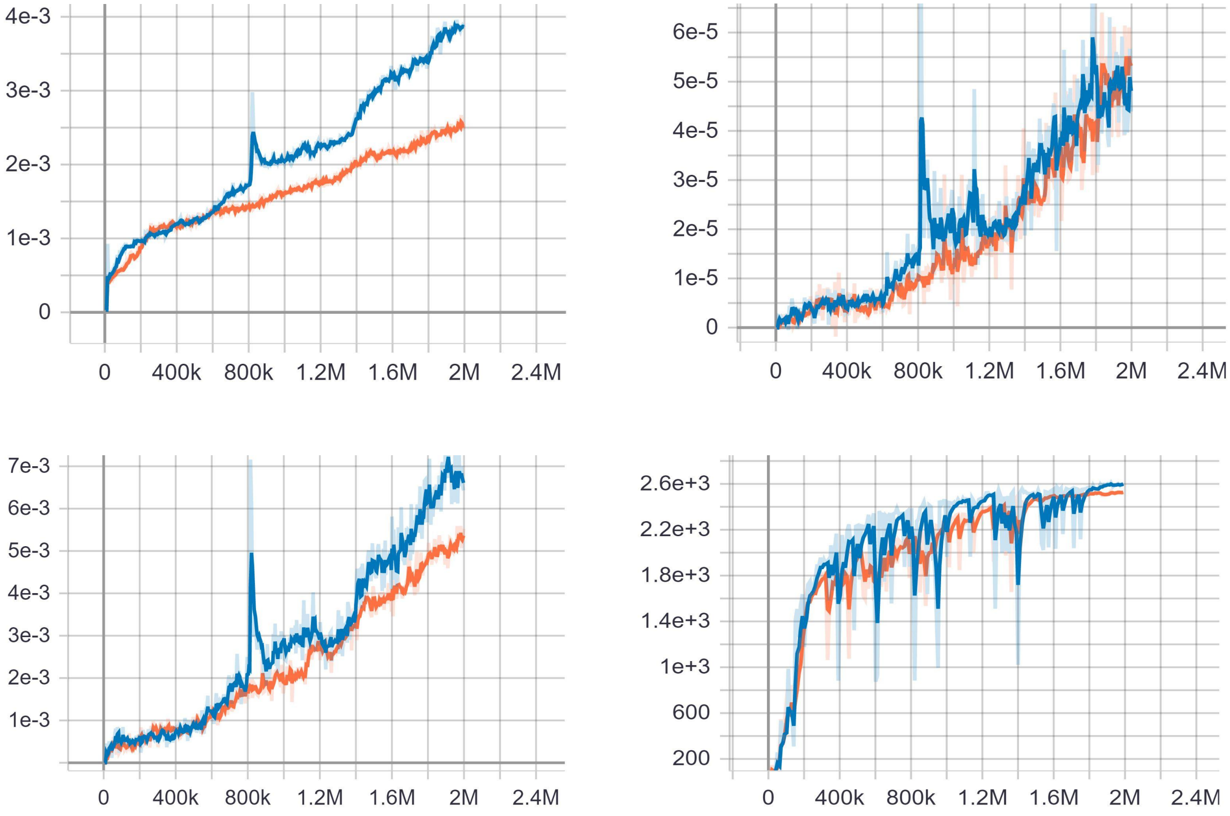
New interpretation:

- based on critic oriented action gap
- “allow” high Q-values near to the optimal action
- “penalize” other high Q-values

New definition:

$$AG = \frac{1}{N} \cdot \sum_{i=1}^{N} ||a_i - d_i||_2 \cdot (Q(s_i, a_i) \;-\; Q(s_i, d_i))$$

Final results



Walker2DBulletEnv-v0

Munchausen-SAC vs SAC

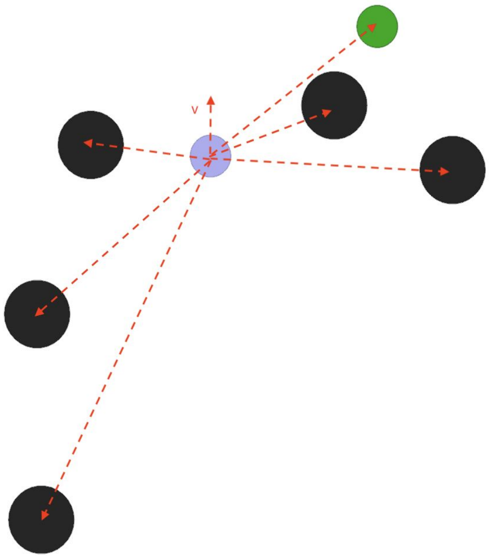
top-left: AG_{actor}

bottom-left: AG_{critic}

top-right: AG

bottom-right: mean reward

Path Planning Task: Particles Environment

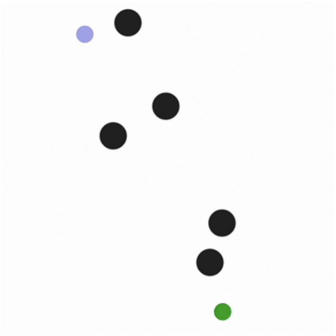


- State space:
  [velocity, goal_pos, obstacles_pos] $\in \mathbb{R}^{2+2+5\times 2}$
- Reward:

$$r_t = -dist(agent,\, goal) + \begin{cases} +10\,, if\, goal\, reached \\ -10\,, if\, agent\, hits\, obstacle \end{cases}$$

- Action: (Motor) Force to apply on the agent.

Particles Environment: SAC vs Munchausen-SAC

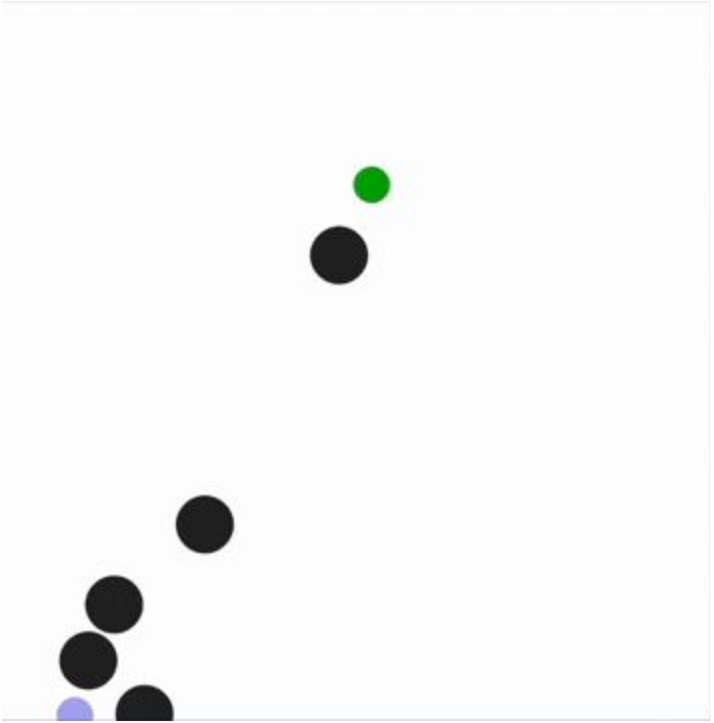


- Training for 2M steps.
- Munchausen-SAC hyperparameters:
  - τ = 0.5 (was 0.9 for M-DQN), clipping threshold $l0$ = -2.0 (was -1.0 for M-DQN).
- Testing on 10,000 configurations:
  - same configurations used for both agents.

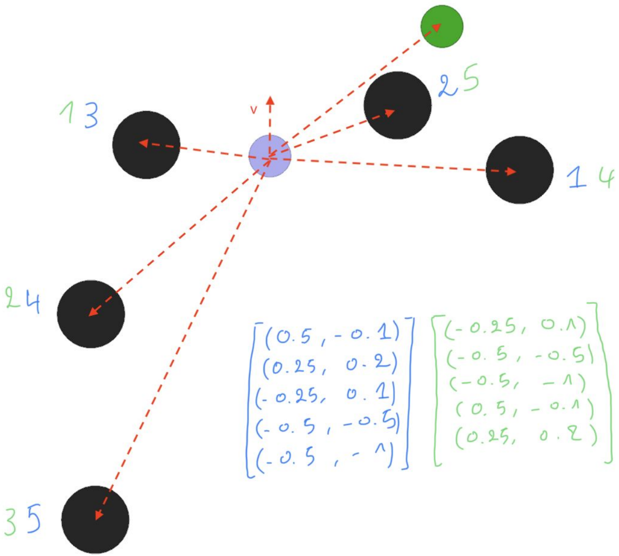
	random configurations	harder configurations (agents trained on random configurations)
SAC	98.48% (avg. time: **7.7**)	74.16% (avg. time: **11.9**)
Munchausen-SAC	**98.66%** (avg. time: 7.9)	**76.48%** (avg. time: 12.8)
Solved by both	97.52%	62.42%

Particles Environment: Munchausen-SAC

- Trained with random configurations.
- Tested on harder ones.

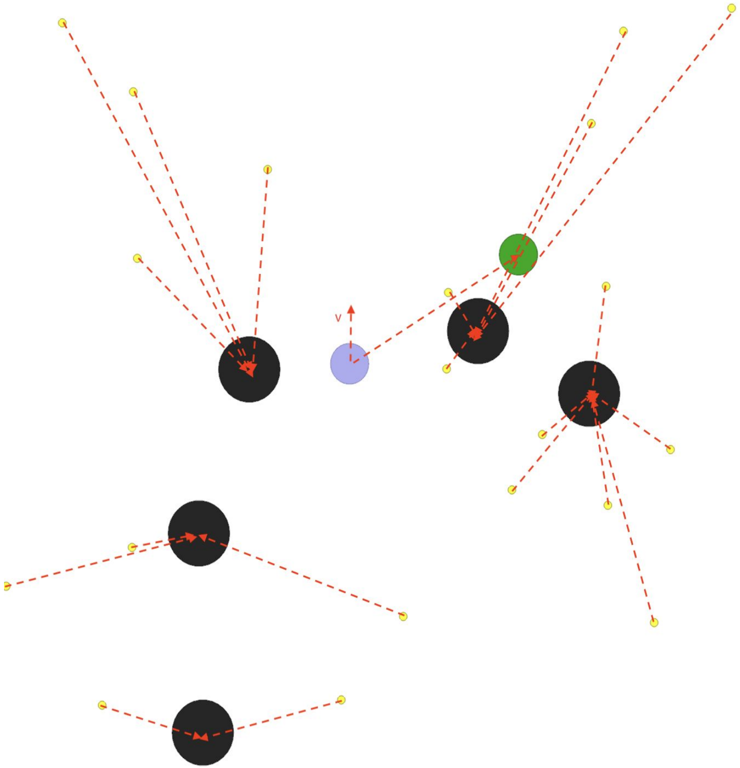


Particles Environment: Limitations



- Fixed number of obstacles.
- **No invariance to obstacles permutations.**

Path Planning Task: Particles Environment with Basis Points Set [3]



- State space:
  [velocity, goal_pos, obstacles_pos] $\in \mathbb{R}^{2+2+20\times 2}$
- Reward and Action: Same as other env.

- Pros:
  - Handles varying number of obstacles.
  - Invariance to obstacles permutations.
- Cons:
  - Higher dimensional state space.

Particles Environment with Basis Points Set: SAC vs Munchausen-SAC

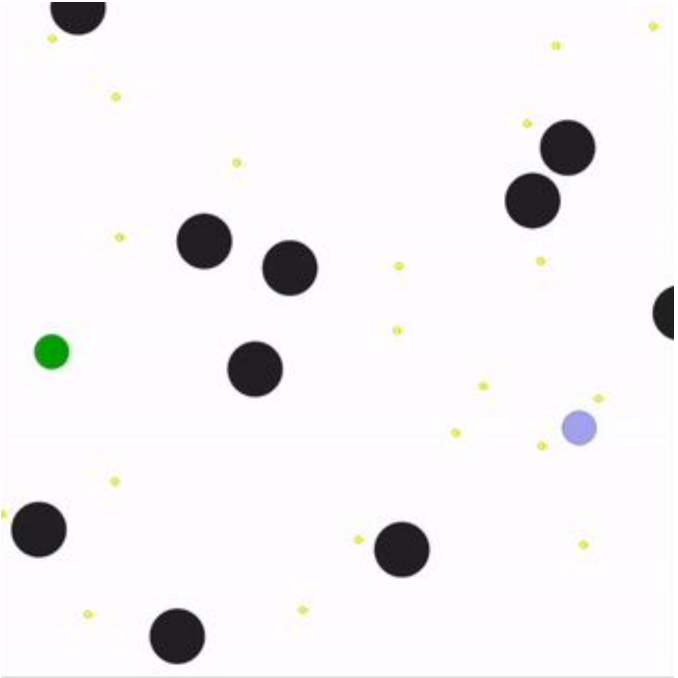
- Training for 2M steps.
- Munchausen-SAC hyperparameters:
  - τ = 0.5 (was 0.9 for M-DQN), clipping threshold $l0$ = -2.0 (was -1.0 for M-DQN).
- Testing on 10,000 random configurations:
  - same configurations used for both agents.
  - same fixed basis points set (20 basis points).
  - Both agents trained on env with 5 obstacles, tested on envs with 5 and 10 obstacles.

	5 obstacles	10 obstacles (agents trained on 5 obstacles)
SAC	71.65% (avg. time: 7.3)	54,55% (avg. time: 6.71)
Munchausen-SAC	**77.14%** (avg. time: **7.2**)	**59,50%** (avg. time: **6.68**)
Solved by both	66.42%	48.03%

Particles Environment with Basis Points Set: SAC vs Munchausen-SAC

Particles Environment with Basis Points Set: Munchausen-SAC

- Trained on env with 5 obstacles.
- Tested on envs with 10 and 15 obstacles.



References

[1] Vieillard, N., Pietquin, O., and Geist, M. (2020). Munchausen reinforcement learning. arXiv preprint arXiv:2007.14430.

[2] Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). Soft actor critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In International Conference on Machine Learning.

[3] Prokudin, Sergey and Lassner, Christoph and Romero, Javier (2019). Efficient Learning on Point Clouds with Basis Point Sets. From Proceedings of the IEEE International Conference on Computer Vision, pages 4332-4341.