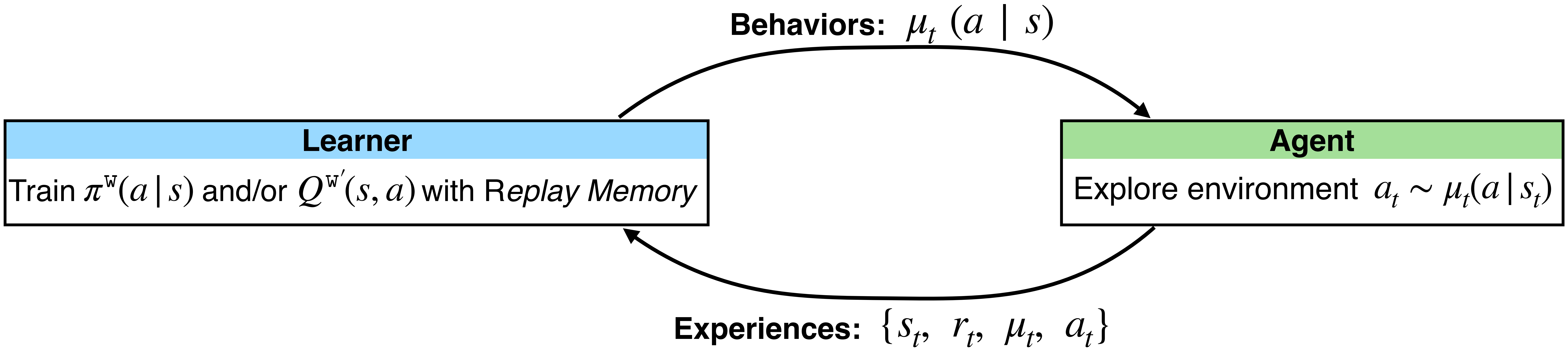
Remember and Forget for Experience Replay

Guido Novati & Petros Koumoutsakos
Computational Science, ETH Zürich

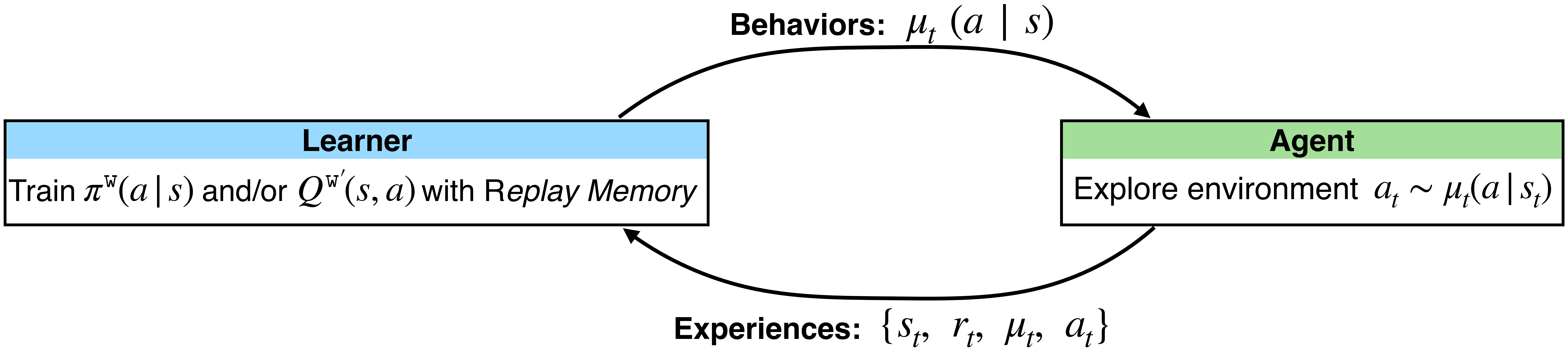
Off-policy Reinforcement Learning

- Off-policy RL with Experience Replay typically alternates:



Off-policy Reinforcement Learning

- Off-policy RL with Experience Replay typically alternates:



- Replay behaviors are typically associated with past policy iterations.
- Off-policy RL attempts to estimate on-policy quantities from off-policy data.

E.g. maximize on-policy returns: $$J(w) = \mathop{\mathbb{E}}_{t \sim \text{RM}} \left[\frac{\pi^w(a_t \mid s_t)}{\mu_t\ (a_t \mid s_t)}\ Q^{\pi^w}(s_t, a_t) \right]$$

Remember and Forget Experience Replay

RL algorithm

1) Which learns a **parameterized policy**.
E.g. DDPG (Lillicrap *et al.* 2016) trains deterministic policy **m**(s) and adds exploration noise:

$$\pi^{\mathtt{w}}(a\,|\,s) = \mathbf{m}^{\mathtt{w}}(s) + \mathcal{N}(0, \boldsymbol{\sigma}^2)$$

2) With **off-policy gradients estimated by ER**.

$$g(\mathtt{w}) = \mathop{\mathbb{E}}_{t\,\sim\,\text{RM}} \left[\, \hat{g}(t, \mathtt{w}) \,\right]$$

E.g. deterministic policy gradient (Silver *et al.* 2014):

$$\hat{g}^{\text{DPG}}(t, \mathtt{w}) = \nabla_{\mathtt{w}} \mathbf{m}^{\mathtt{w}}(s_t)\, \nabla_a Q^{\mathtt{w}'}(s_t, a) \Big|_{a=\mathbf{m}^{\mathtt{w}}(s_t)}$$

Remember and Forget Experience Replay

RL algorithm

1) Which learns a **parameterized policy**.
E.g. DDPG (Lillicrap *et al.* 2016) trains deterministic policy **m**(s) and adds exploration noise:

$$\pi^{\mathtt{w}}(a\,|\,s) = \mathbf{m}^{\mathtt{w}}(s) + \mathcal{N}(0, \boldsymbol{\sigma}^2)$$

2) With **off-policy gradients estimated by ER**.

$$g(\mathtt{w}) = \mathop{\mathbb{E}}_{t\,\sim\,\text{RM}} \left[\, \hat{g}(t, \mathtt{w}) \,\right]$$

E.g. deterministic policy gradient (Silver *et al.* 2014):

$$\hat{g}^{\text{DPG}}(t, \mathtt{w}) = \nabla_{\mathtt{w}} \mathbf{m}^{\mathtt{w}}(s_t)\, \nabla_a Q^{\mathtt{w}'}(s_t, a)\Big|_{a=\mathbf{m}^{\mathtt{w}}(s_t)}$$

ReF-ER

1) **Rejects samples** from gradient estimation if importance weight $\rho_t^{\mathtt{w}} = \pi^{\mathtt{w}}(a_t \mid s_t)/\mu_t(a_t \mid s_t)$ **outside of a trust region.**

2) **Penalizes policy towards training behaviors.**

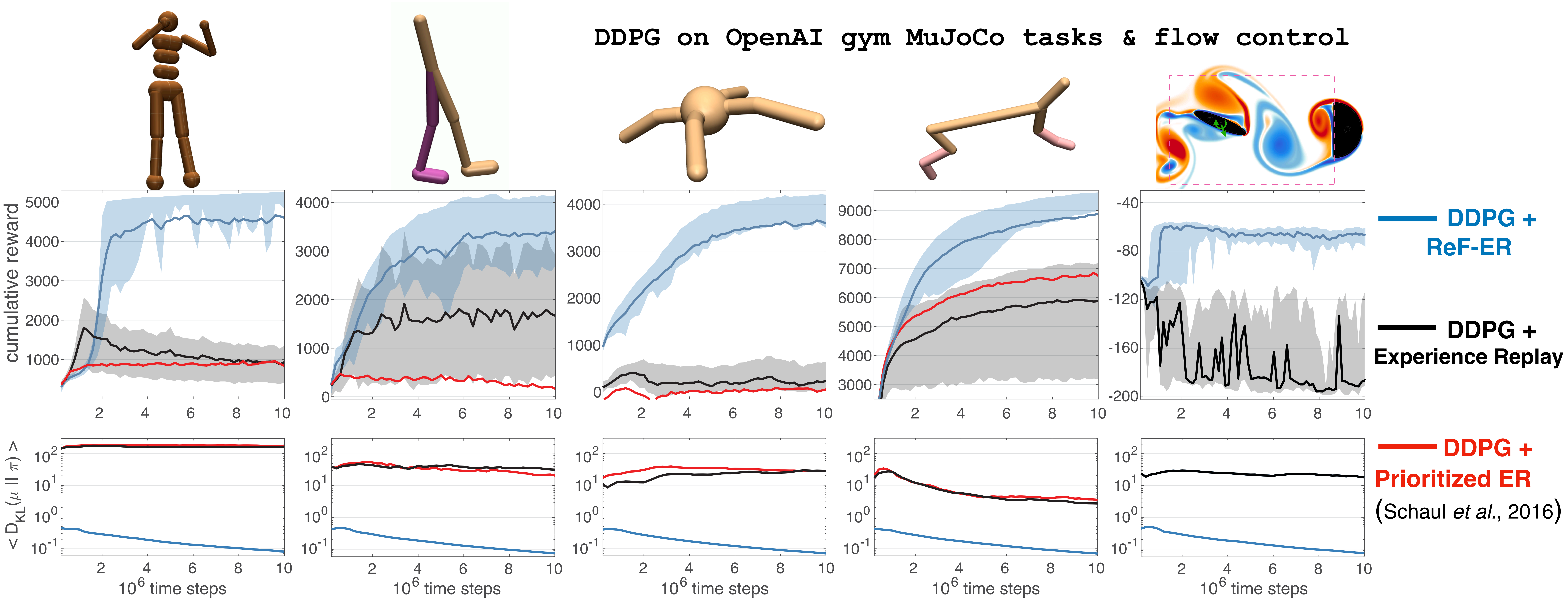
$$\hat{g}(t, \mathtt{w}) \leftarrow \begin{cases} \beta \hat{g}(t, \mathtt{w}) - (1-\beta)\, \nabla D_{\text{KL}} \left[\mu_t \| \pi^{\mathtt{w}}(\cdot\,|\,s_t)\right] & \text{if } \frac{1}{C} < \rho_t < C \\ -(1-\beta)\, \nabla D_{\text{KL}} \left[\mu_t \| \pi^{\mathtt{w}}(\cdot\,|\,s_t)\right] & \text{otherwise} \end{cases}$$

Notes:
- Trust region parameter C can be annealed.
- Coefficient β is iteratively updated to keep a fixed fraction of samples within the trust region.

Results

- ReF-ER with: Off-policy pol.-gradients (ACER, Wang *et al.* 2017), Q-learning (NAF, Gu *et al.* 2016), DPG (DDPG, Lillicrap *et al.* 2016).
- We observe: **effectively constrained D_{KL}, increased stability and performance**.
- At the price of: sometimes slower progress at the beginning of training.

DDPG on OpenAI gym MuJoCo tasks & flow control



Conclusion

GENERAL IMPLICATION:

Off-policy RL benefits from maintaining similarity between policy and training behaviors.

ReF-ER:

- Easy to implement, modular improvement for off-policy RL.
- Aligns on-policy distribution (‘test set’) and replay experiences (`training set`).
- Brings off-policy RL one step closer to supervised learning.

More info:

- poster : Pacific Ballroom # 50
- paper : https://arxiv.org/abs/1807.05827
- source code : https://github.com/cselab/smarties