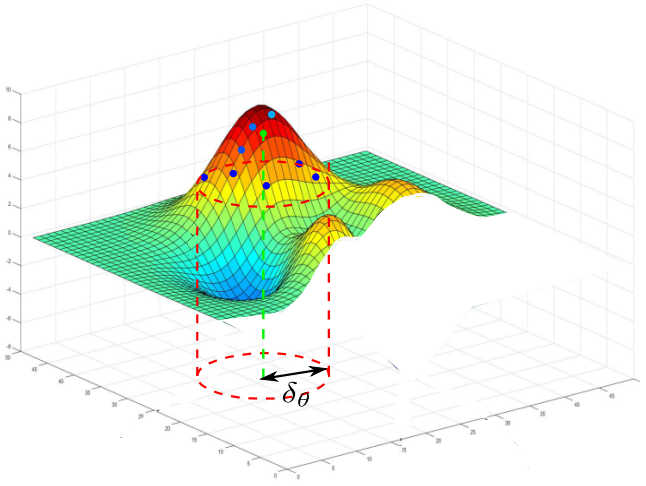
Hyper-parameter tuning and Population-Based Training

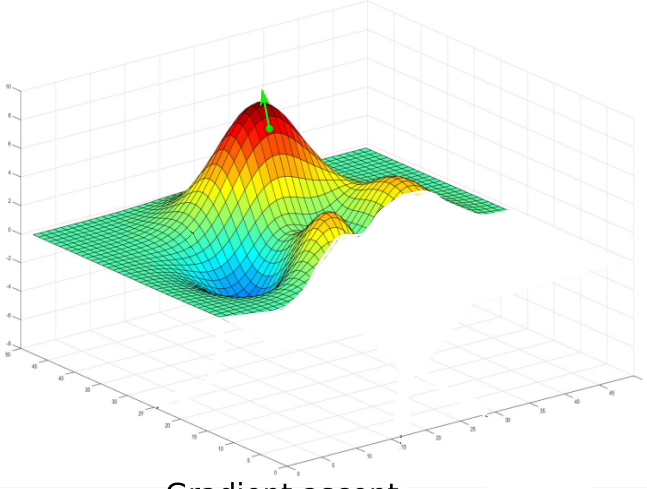
Olivier Sigaud

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

Context: survey of Evo+RL combinations



Selection - Variation

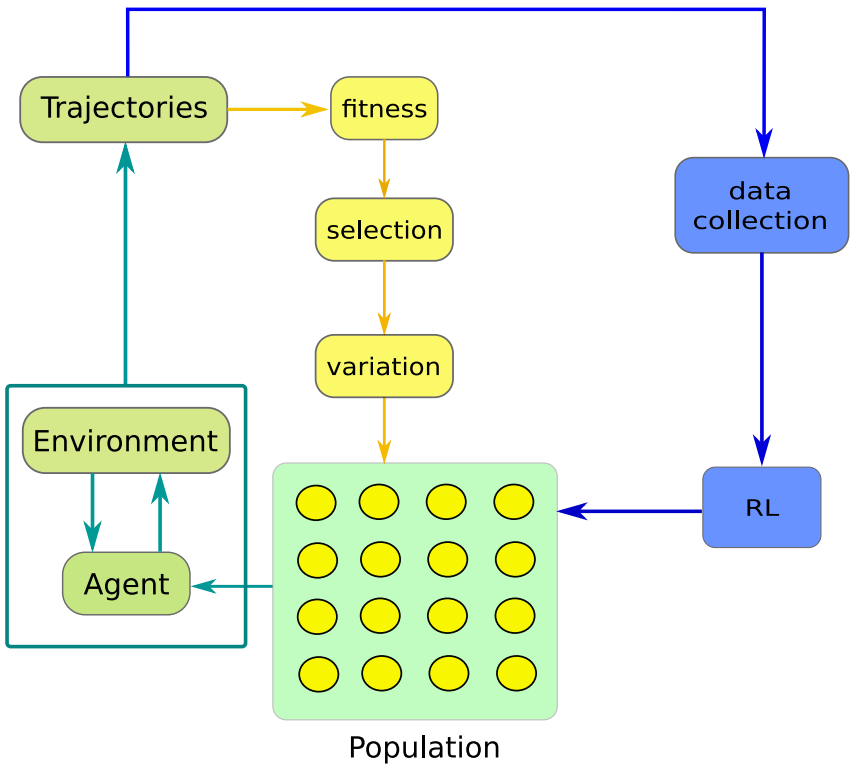


Gradient ascent

- There are two main approaches to policy search:
  - **Direct policy search methods:** given π_θ, variations $\pi_{\theta'}$ are generated and evaluated. The defining characteristics of this approach is that π_θ^{t+1} depends on the post hoc evaluation of the $\pi_{\theta_i}^t$.
  - This is not sample efficient, but robust
  - **Policy gradient methods:** π_θ^{t+1} is obtained from π_θ^t by following a gradient of performance computed from estimates of the value of individual actions taken by π_θ^t over a set of states.
  - Need to stay close to π_θ^t for the gradient to be useful.

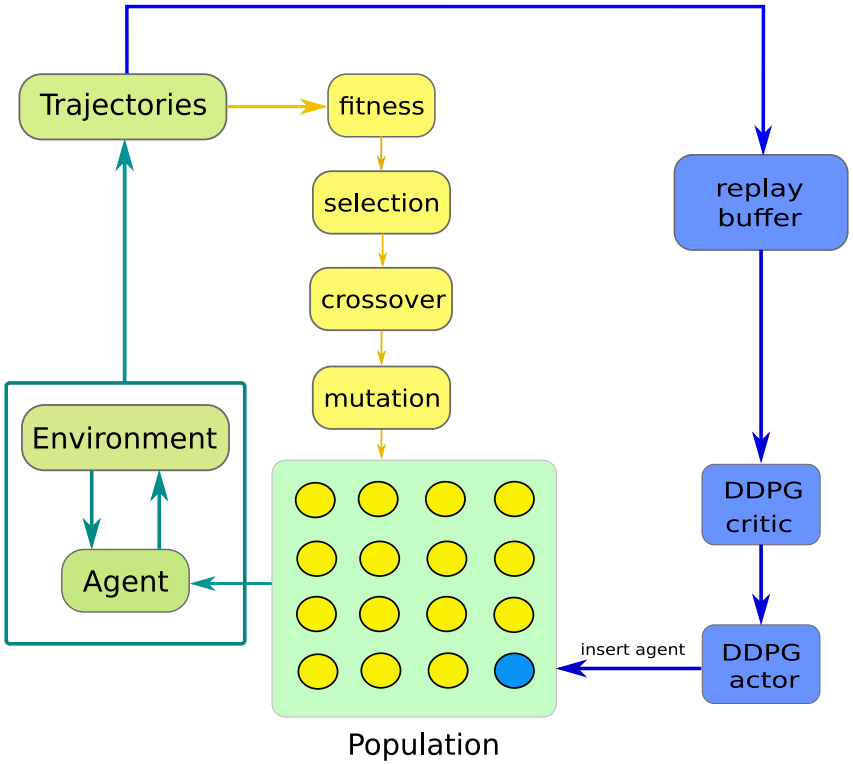
Combinations

- Several algorithms combine direct policy search and policy gradient methods
- The focus of PBT is hyperparameter tuning.
- An outlier in the Evo+RL methods
- A general template:



Sigaud, O. (2022) Combining evolution and deep reinforcement learning for policy search: a survey. *ACM Transactions in Evolutionary Learning and Optimization*

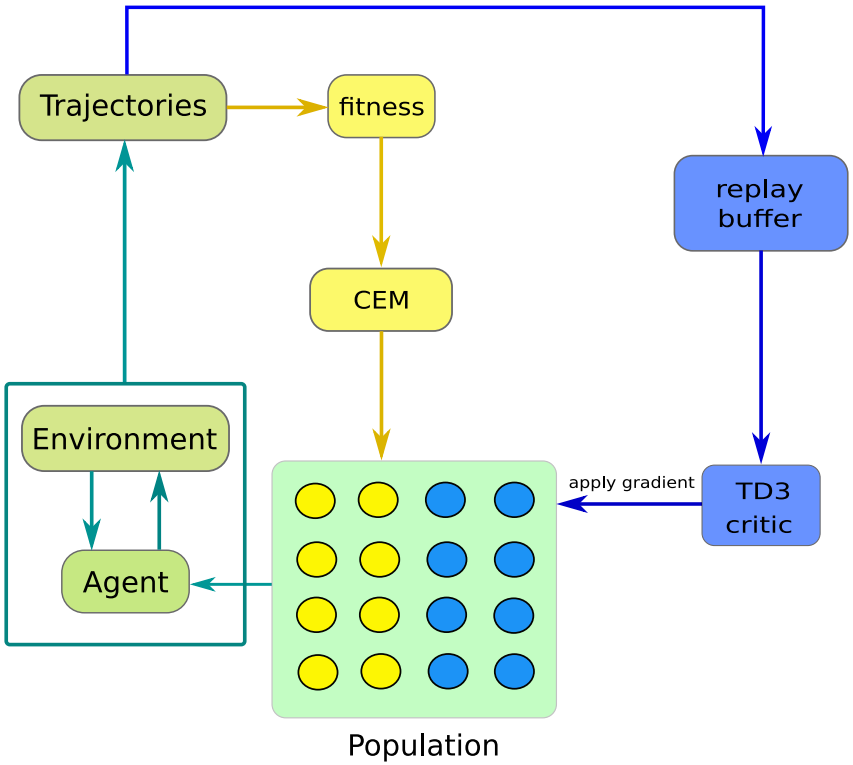
Example 1: ERL



- The DDPG agent is injected into the population, if it performs better it is kept and accelerates evolution

Khadka, S. & Tumer, K. (2018a) Evolution-guided policy gradient in reinforcement learning. In *Neural Information Processing Systems*

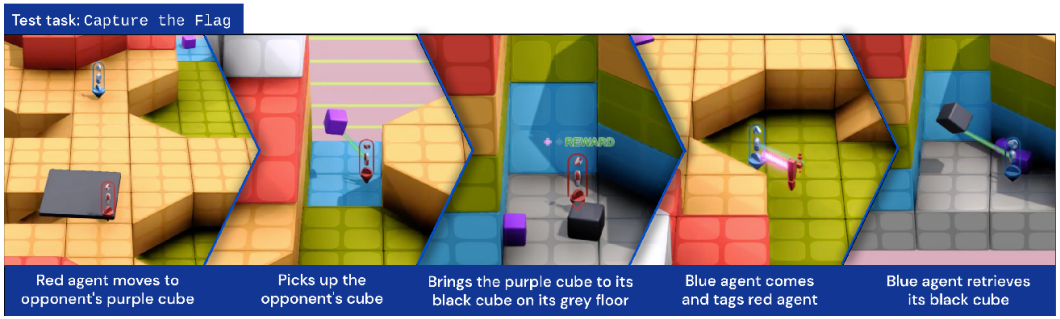
Example 2: CEM-RL



- Half the population get critic gradient updates, if it improves the corresponding agents are kept and accelerate evolution

Pourchot, A. & Sigaud, O. (2018) CEM-RL: Combining evolutionary and gradient-based methods for policy search. *arXiv preprint arXiv:1810.01222* (submitted to ICLR)

Introduction to PBT



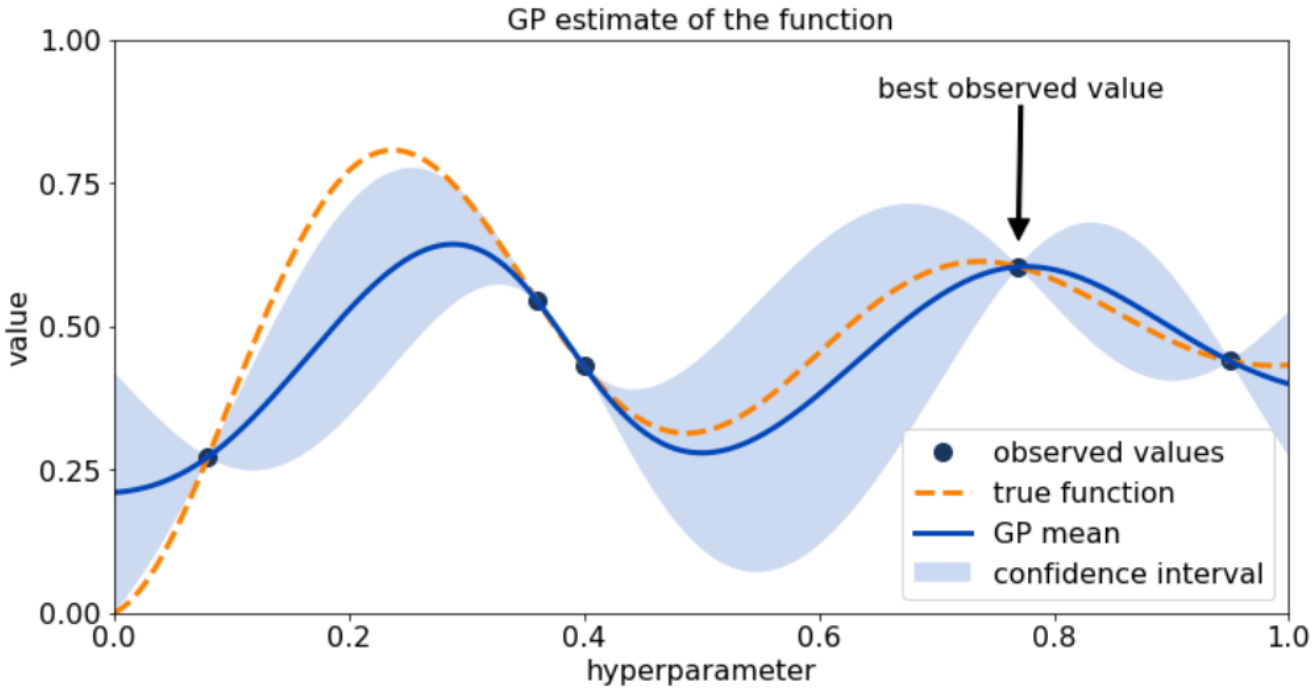
- Hyperparameter search is crucial in Deep RL
- Population-Based Training (PBT) provides an efficient solution to this problem
- It has been used in several notorious applications of Deep RL
- We note $\mathbf{h}$ the hyperparameter vector and $\boldsymbol{\theta}$ the parameter vector

Jaderberg, M., Dalibard, V., Osindero, S., Czarnecki, W. M., Donahue, J., Razavi, A., Vinyals, O., Green, T., Dunning, I., Simonyan, K., et al. (2017) Population-based training of neural networks. *arXiv preprint arXiv:1711.09846*

Jaderberg, M., Czarnecki, W. M., Dunning, I., Marris, L., Lever, G., Castaneda, A. G., Beattie, C., Rabinowitz, N. C., Morcos, A. S., Ruderman, A., et al. (2019) Human-level performance in 3D multiplayer games with population-based reinforcement learning. *Science*, 364(6443), 859–865

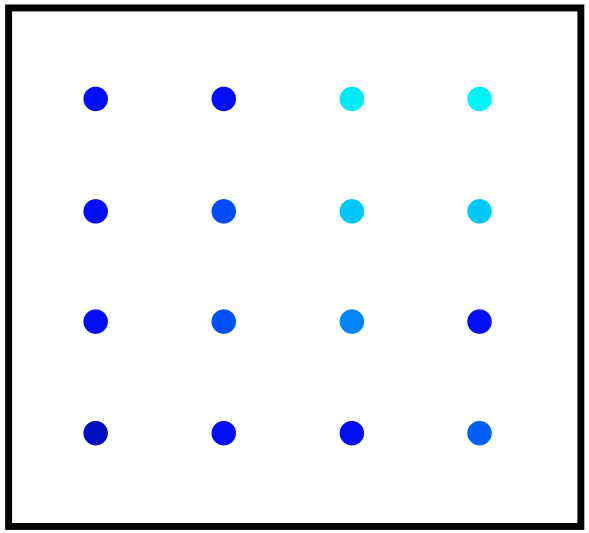
Stooke, A., Mahajan, A., Barros, C., Deck, C., Bauer, J., Sygnowski, J., Trebacz, M., Jaderberg, M., Mathieu, M., et al. (2021) Open-ended learning leads to generally capable agents. *arXiv preprint arXiv:2107.12808*

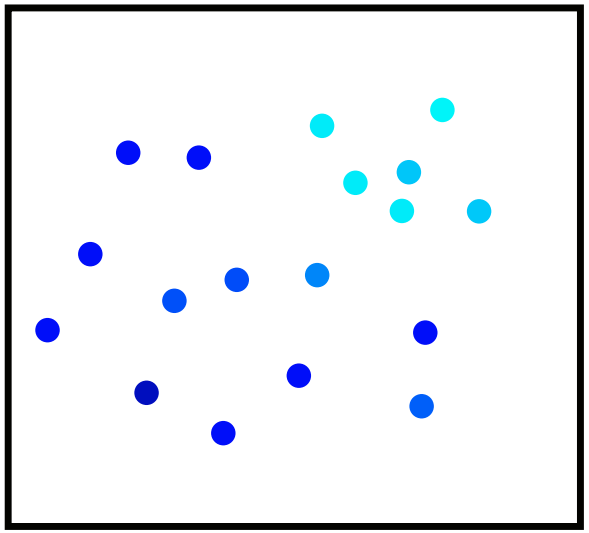
Bayesian optimization



- The choice of h_{t+1} depends on information gathered with h_t
- No room for parallel search
- Key example: optuna

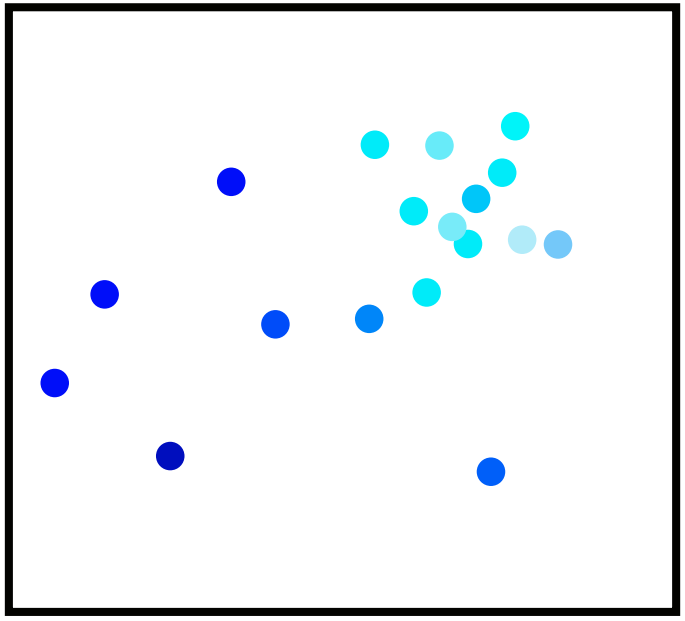
Grid search or random search





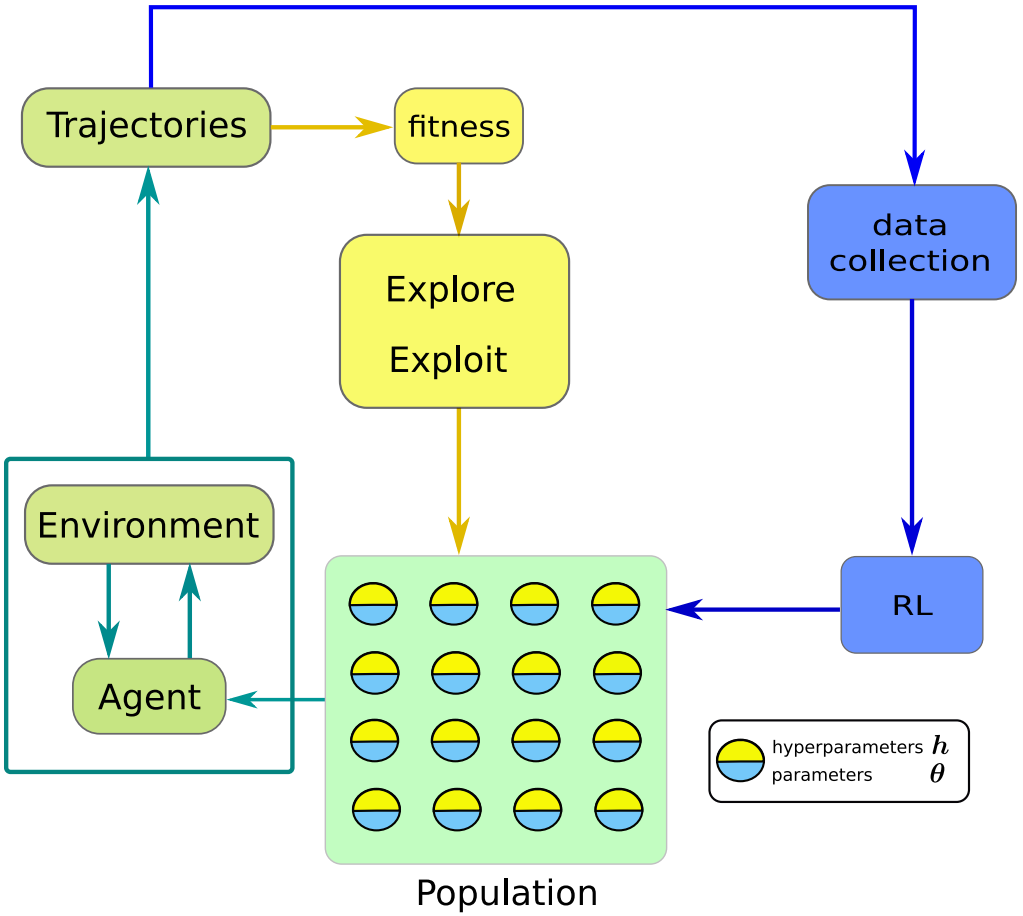
- All evaluations are independent and can be performed in parallel
- Random search is generally better than grid search
- But no mechanism to insist on promising areas

Evolutionary methods



- The best of both worlds:
  - Parallel search at each generation
  - Convergence to sweet spots
- PBT uses an evolutionary approach
- But hyperparameter search is performed during the training of agents
- A more adaptive dynamics

The PBT architecture



- The PBT approach is applied to more than RL (GAN, supervised learning...) but here we focus on RL.
- The variation-selection operators (**Exploit**, **Explore**) are applied to both parameters and hyperparameters

Overview

Algorithm 1 Population Based Training (PBT)

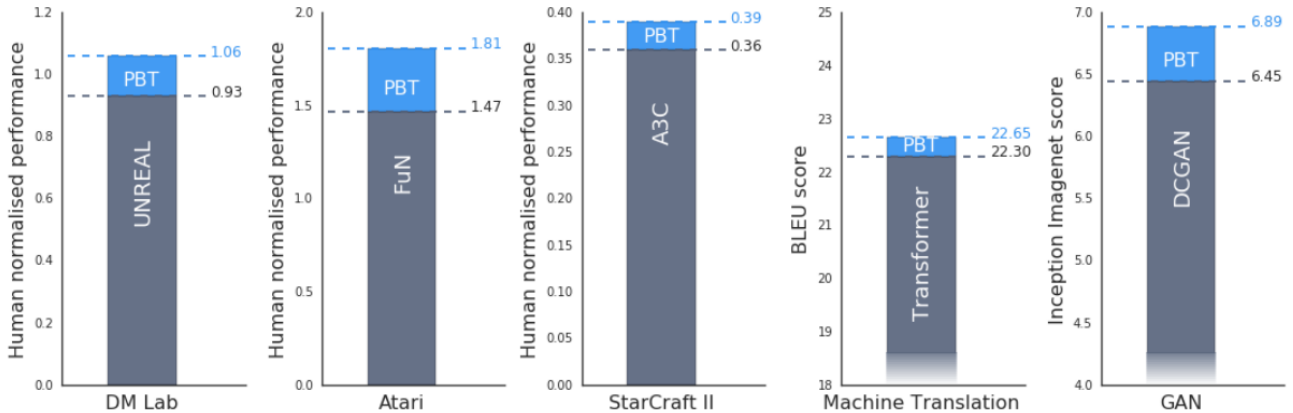
procedure TRAIN($\mathcal{P}$) ▷ initial population $\mathcal{P}$
  for $(\theta, h, p, t) \in \mathcal{P}$ (asynchronously in parallel) **do**
    while not end of training **do**
      $\theta \leftarrow \texttt{step}(\theta|h)$ ▷ one step of optimisation using hyperparameters h
      $p \leftarrow \texttt{eval}(\theta)$ ▷ current model evaluation
      if $\texttt{ready}(p, t, \mathcal{P})$ **then**
        $h', \theta' \leftarrow \texttt{exploit}(h, \theta, p, \mathcal{P})$ ▷ use the rest of population to find better solution
        if $\theta \neq \theta'$ **then**
          $h, \theta \leftarrow \texttt{explore}(h', \theta', \mathcal{P})$ ▷ produce new hyperparameters h
          $p \leftarrow \texttt{eval}(\theta)$ ▷ new model evaluation
        end if
      end if
      update $\mathcal{P}$ with new $(\theta, h, p, t+1)$ ▷ update population
    end while
  end for
  return θ with the highest p in $\mathcal{P}$
end procedure

- Two evolutionary operators: **Exploit**, **Explore**
- **Exploit** copies $\boldsymbol{\theta}$ and $\mathbf{h}$ from a better individual
- Requires centralizing and updating perf of individuals at all times (eval)
- Risk of significant overhead
- **Explore** varies $\boldsymbol{\theta}$ and $\mathbf{h}$
- Evolution when the individual is ready (typically, 10^6 gradient steps)
- With synchronized evolutions, lesser evaluation overhead

RL implementation

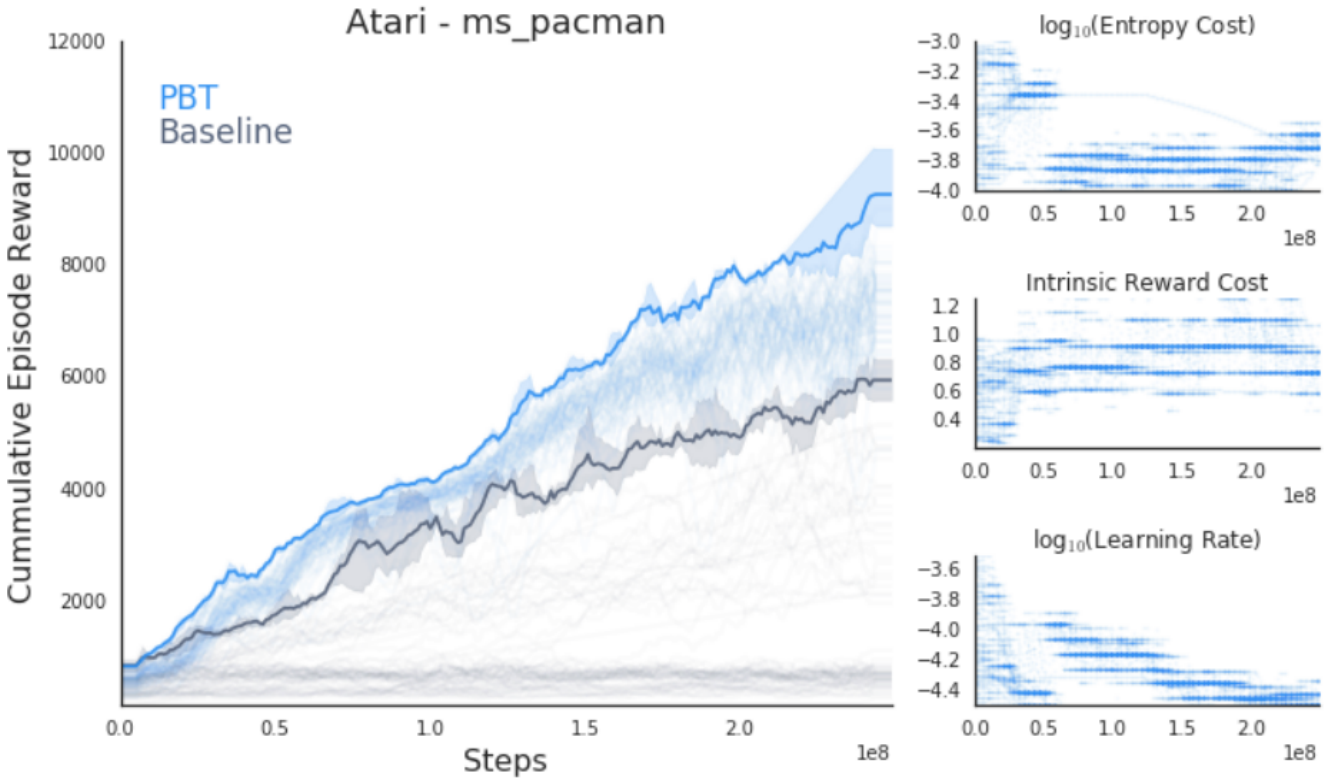
- Two Explore strategies
  - **Perturb:** each hyperparameter independently is randomly perturbed by a factor of 1.2 or 0.8.
  - **Resample:** each hyperparameter is resampled from the original prior distribution
  - They seem to only use Perturb
- Two Exploit strategies
  - **T-test selection:** Uniformly sample another agent that replaces the current agent if its last 10 episodic rewards are better using Welch's T-test.
  - **Truncation selection:** Rank all agents, if the current agent is in the bottom K% replace by one in the top K% (K=20 or 5).

Consistent gain in performance



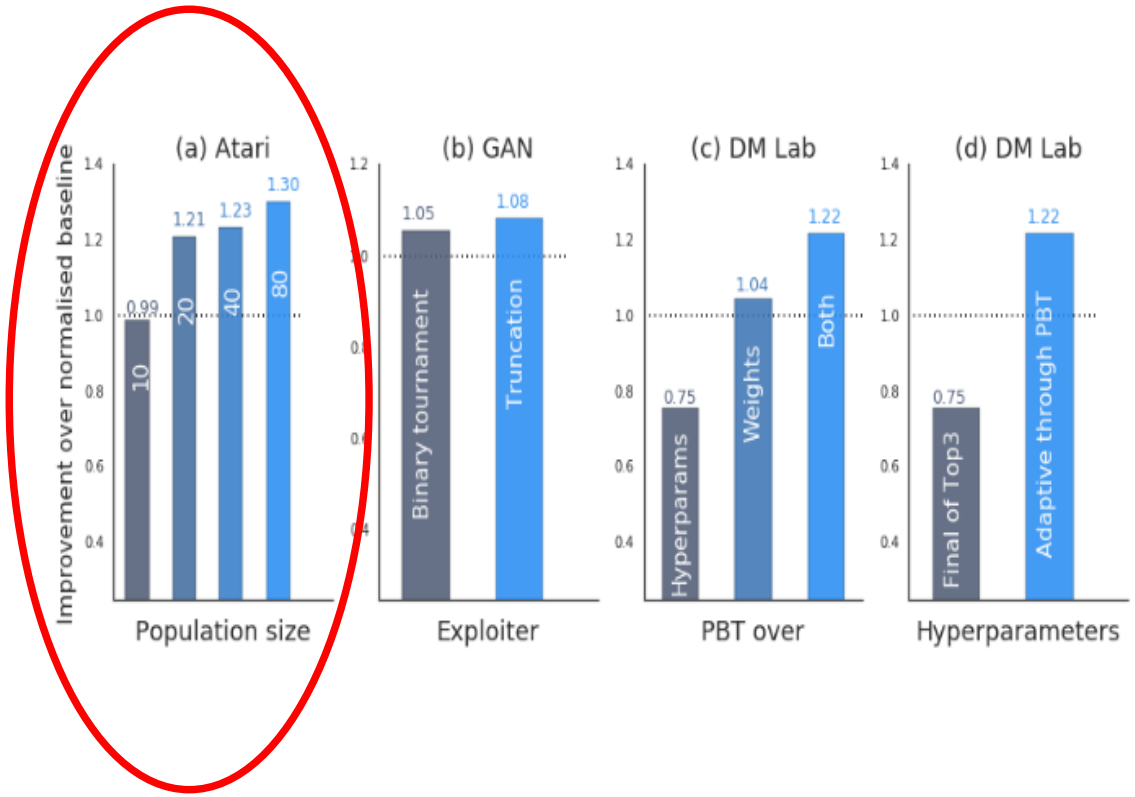
- Tried over different domains with different algorithms
- I only cover the RL part

Variation of hyperparameters over time



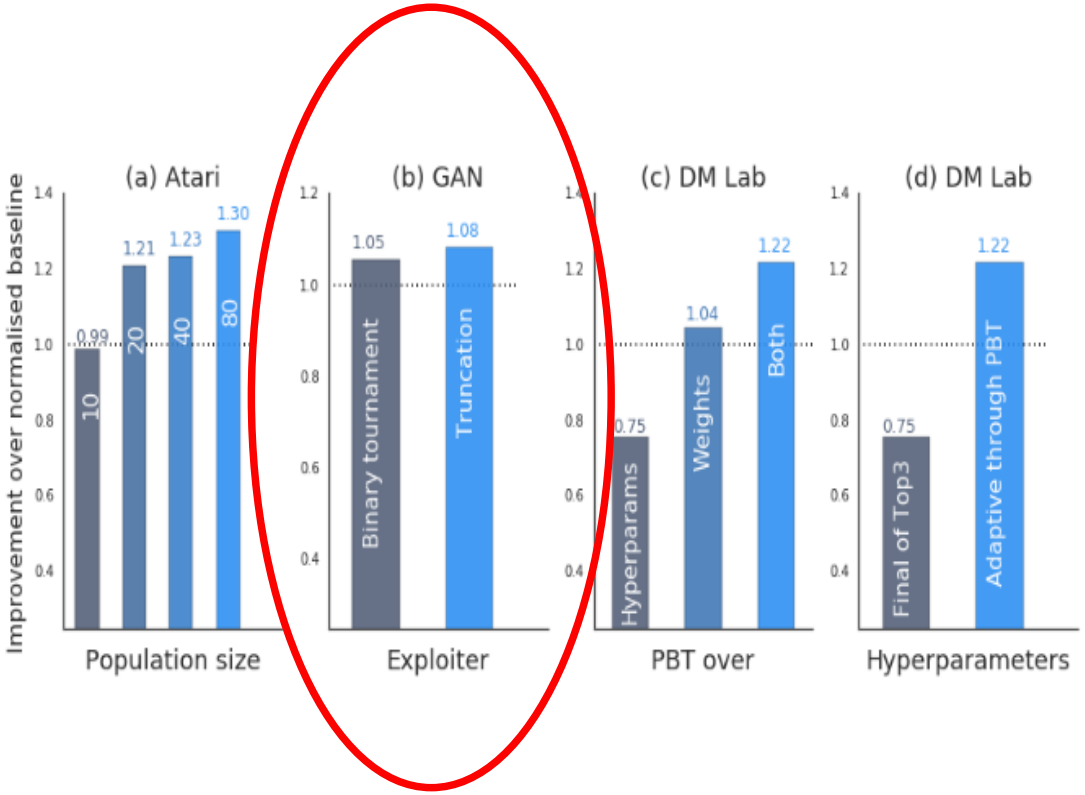
- We can see the $\mathbf{h}$ drifting over time
- Does not converge to a single value

Effect of population size



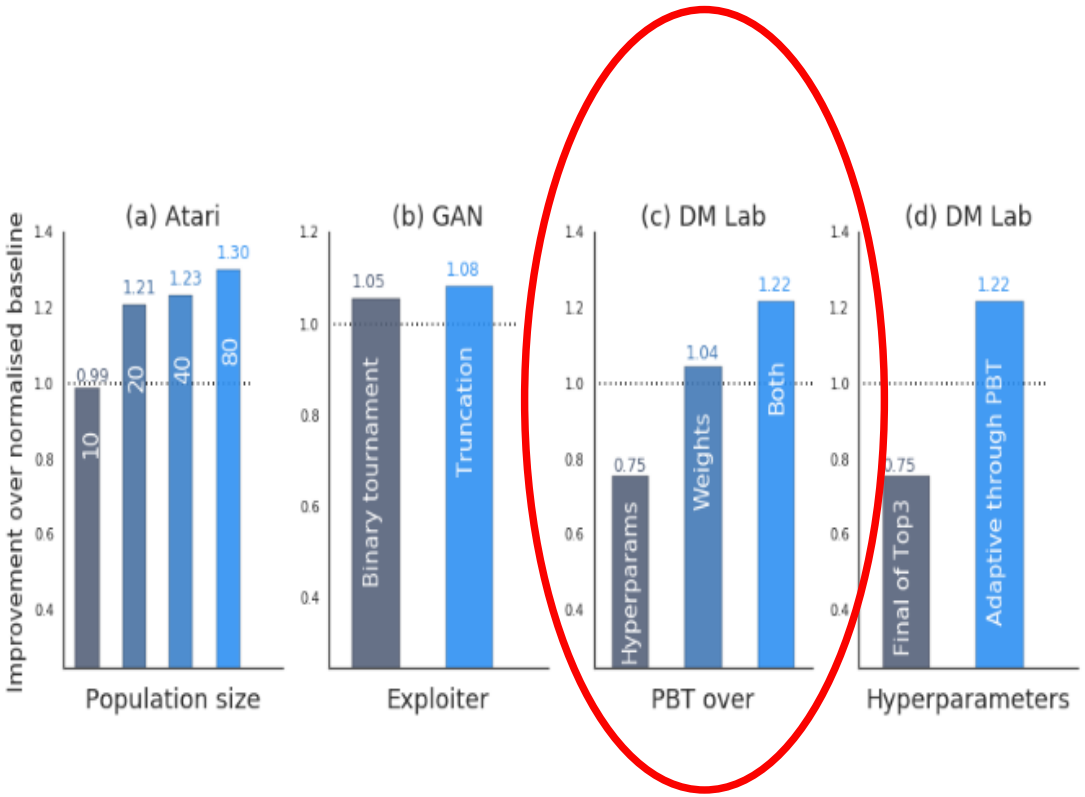
- A too small population results in suboptimal performance
- Adding more agents improves less and less
- More agents means more samples. They do not compare at constant budget! (this is not clearly specified in the paper)

Choice of Exploit method



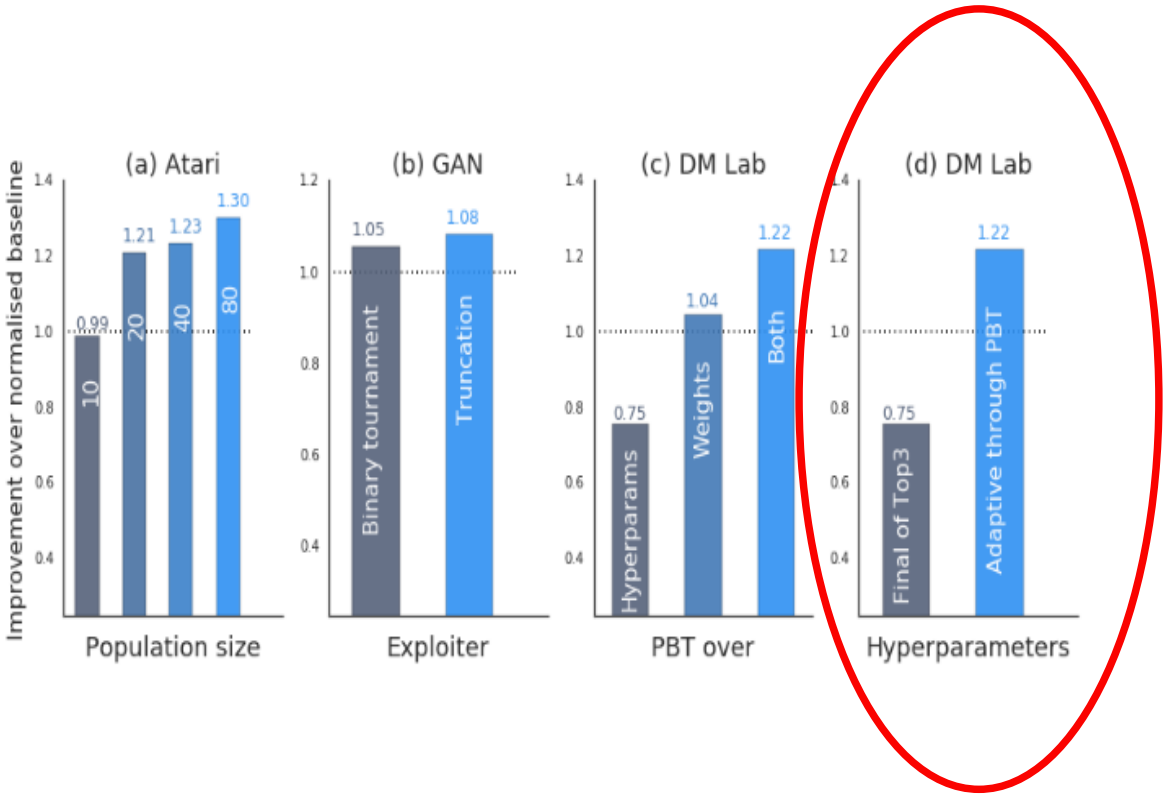
- Truncation selection works slightly better
- Both **Exploit** operators are very heuristic
- Room for improvement

Copy both $\mathbf{h}$ and θ?



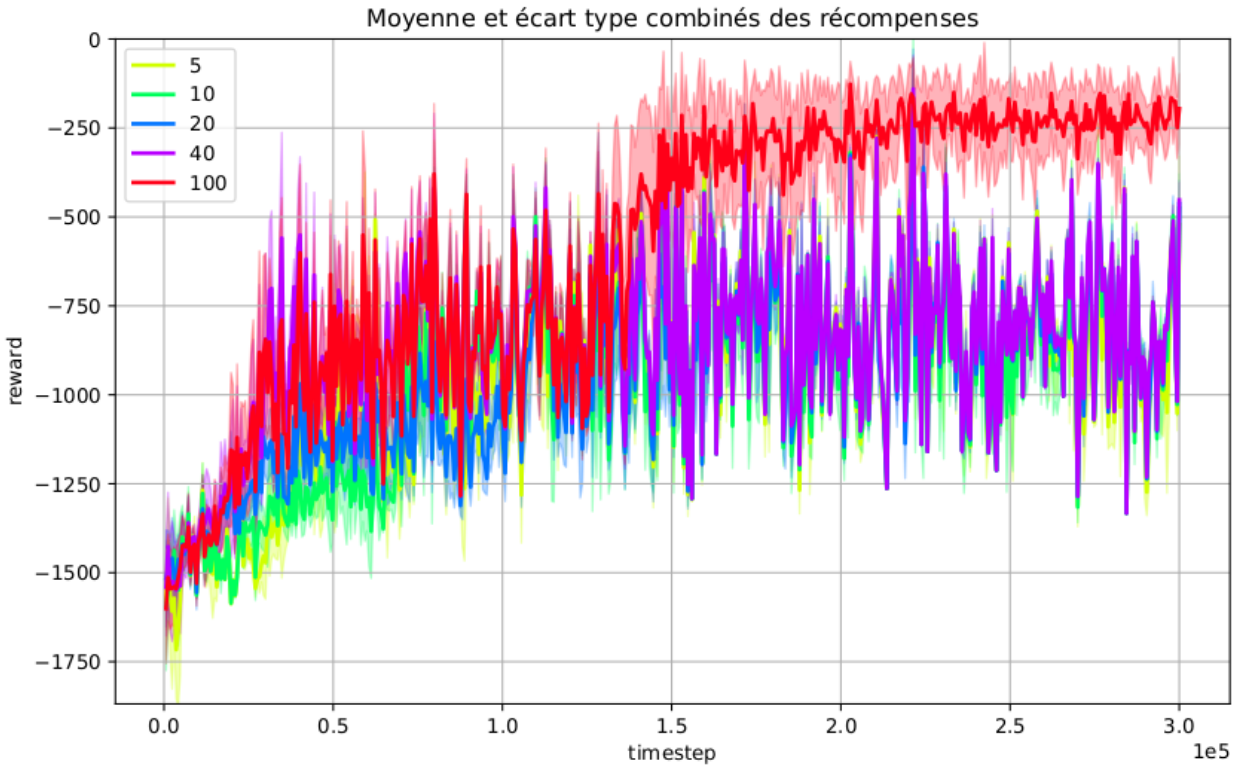
- Applying **Exploit** to both $\mathbf{h}$ and θ is better than just to $\mathbf{h}$

Role of adaptivity



- Measure performance with the fixed final $\mathbf{h}$
- The fact that $\mathbf{h}$ changes over time is beneficial to performance

Local empirical study



- A population of 100 agents is required to make a difference on Pendulum
- Study a more advanced evolution method to see if we can reduce the size of efficient populations

Any question?

Send mail to: Olivier.Sigaud@upmc.fr

Jaderberg, M., Czarnecki, W. M., Dunning, I., Marris, L., Lever, G., Castaneda, A. G., Beattie, C., Rabinowitz, N. C., Morcos, A. S., Ruderman, A., et al. (2019).
Human-level performance in 3d multiplayer games with population-based reinforcement learning.
Science, 364(6443):859–865.

Jaderberg, M., Dalibard, V., Osindero, S., Czarnecki, W. M., Donahue, J., Razavi, A., Vinyals, O., Green, T., Dunning, I., Simonyan, K., et al. (2017).
Population-based training of neural networks.
arXiv preprint arXiv:1711.09846.

Jung, W., Park, G., and Sung, Y. (2020).
Population-guided parallel policy search for reinforcement learning.
In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Khadka, S. and Tumer, K. (2018).
Evolution-guided policy gradient in reinforcement learning.
In *Neural Information Processing Systems*.

Liu, Y., Ramachandran, P., Liu, Q., and Peng, J. (2017).
Stein variational policy gradient.
arXiv preprint arXiv:1704.02399.

Parker-Holder, J., Nguyen, V., Desai, S., and Roberts, S. J. (2021).
Tuning mixed input hyperparameters on the fly for efficient population based AutoRL.
Advances in Neural Information Processing Systems, 34.

Parker-Holder, J., Nguyen, V., and Roberts, S. (2020).
Provably efficient online hyperparameter optimization with population-based bandits.
arXiv preprint arXiv:2002.02518.

Pourchot, A. and Sigaud, O. (2018).
CEM-RL: Combining evolutionary and gradient-based methods for policy search.

arXiv preprint arXiv:1810.01222 (ICLR 2019).

Sigaud, O. (2022).

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

Stooke, A., Mahajan, A., Barros, C., Deck, C., Bauer, J., Sygnowski, J., Trebacz, M., Jaderberg, M., Mathieu, M., et al. (2021).

Open-ended learning leads to generally capable agents.
arXiv preprint arXiv:2107.12808.

Follow-up: PB2 (Population-Based Bandits)

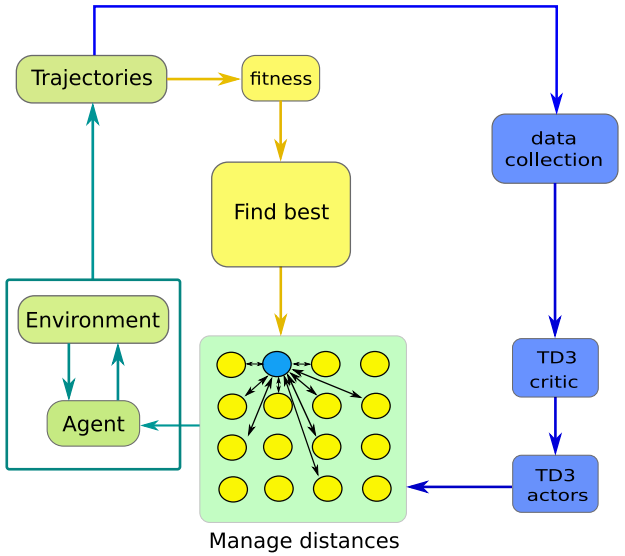
- The PBT heuristics are questionable, provide no regret guarantees.
- PBT needs pop size > 20
- Formalize a **time varying schedule for hyperparams**
- Get a Bayesian Optimization model of hyperparam choice
- Use a GP model of the schedule with time varying bandits (that's technical)
- Get a good regret bound, need pop size $= 4$ or 5
- PB2 is limited to continuous hyperparams (by nature of the GP)
- 2021: extensions to mixed continuous and categorical variables (PB2-Multi and PB2-Mix)
- Still choice of parents based on performance


Parker-Holder, J., Nguyen, V., and Roberts, S. Provably efficient online hyperparameter optimization with population-based bandits. *arXiv preprint arXiv:2002.02518*, 2020


Parker-Holder, J., Nguyen, V., Desai, S., and Roberts, S. J. Tuning mixed input hyperparameters on the fly for efficient population based AutoRL. *Advances in Neural Information Processing Systems*, 34, 2021

ISIR INSTITUT DES SYSTÈMES INTELLIGENTS ET DE ROBOTIQUE

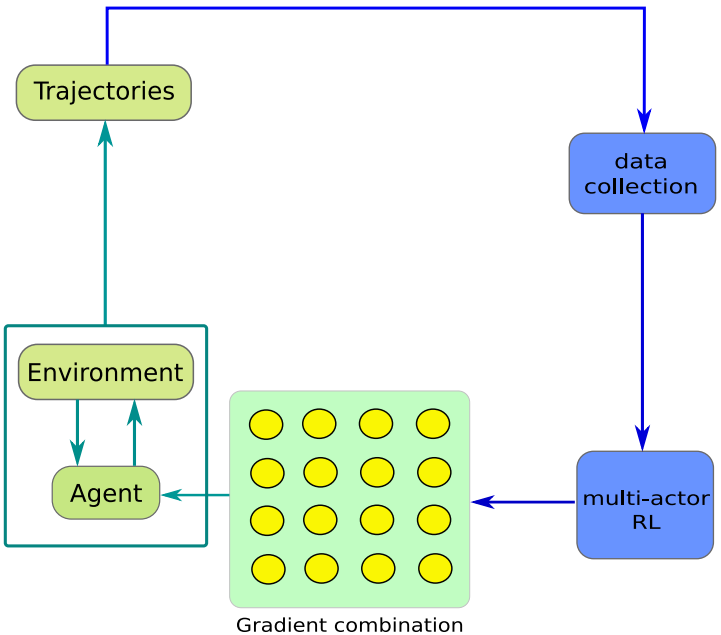
Follow-up: P3S-TD3



- Instead of hard copy of π^t_{best} (and hyperparams), adds a KL between π^t_{best} and the $\pi^t_{\theta_i}$ to help moving towards π^t_{best}
- But also maintains a minimal distance between π^t_{best} and the $\pi^t_{\theta_i}$, to prevent collapse (part not covered by the derivation)
- The mechanism is somewhat ad hoc
- Still a selection of the best based on fitness
- Convincing recent results, not much cited

Jung, W., Park, G., and Sung, Y. (2020) Population-guided parallel policy search for reinforcement learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net URL https://openreview.net/forum?id=rJeINp4KwH

A glimpse at SVPG



- P3S-TD3 is intermediate between PBT and SVPG (ignores SVPG)
- SVPG does not have post hoc evaluation any more

Liu, Y., Ramachandran, P., Liu, Q., and Peng, J. (2017) Stein Variational Policy Gradient. *arXiv preprint arXiv:1704.02399*, (UAI).