Evolution Strategies for Neural Policy Search

Mid-thesis commitee

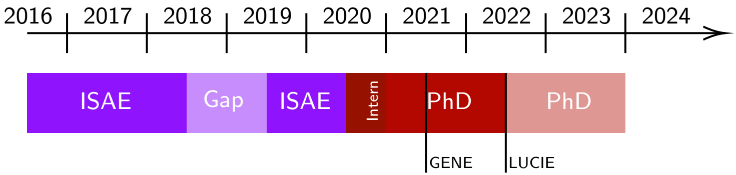
Author: Paul Templier[1]
Advisors: Emmanuel Rachelson[1], Dennis G. Wilson[1]

[paul.templier@isae-supaero.fr]

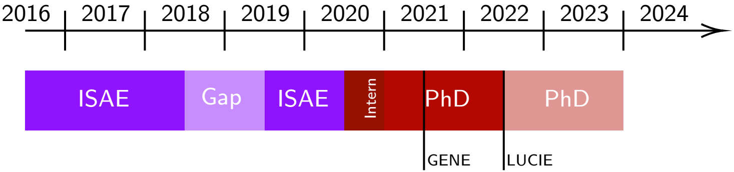
June 29, 2022

[1] University of Toulouse, ISAE-SUPAERO

[Context] Mid-thesis report



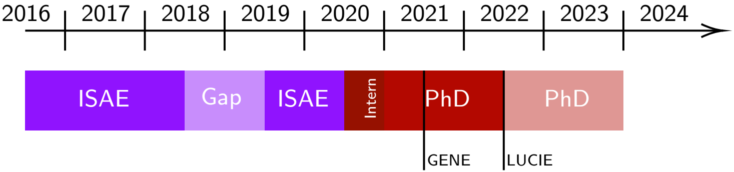
[Context] Mid-thesis report



Initial topic

Bio-inspired methods for artificial neural networks

[Context] Mid-thesis report



Initial topic

Bio-inspired methods for artificial neural networks

Goal of this report

Organize past and present work, and highlight future research directions.

Content

1. [Context] Context of this PhD

Content

1. [Context] Context of this PhD
2. [Policy search] Evolution Strategies for Policy Search

Content

1. [Context] Context of this PhD
2. [Policy search] Evolution Strategies for Policy Search
3. [Search space] Representing policies and changing the search space

Content

1. [Context] Context of this PhD
2. [Policy search] Evolution Strategies for Policy Search
3. [Search space] Representing policies and changing the search space
4. [Search direction] Using samples to help the search

Content

1. [Context] Context of this PhD
2. [Policy search] Evolution Strategies for Policy Search
3. [Search space] Representing policies and changing the search space
4. [Search direction] Using samples to help the search
5. [Noisy fitness] Adapting to stochastic problems

Content

1. [Context] Context of this PhD
2. [Policy search] Evolution Strategies for Policy Search
3. [Search space] Representing policies and changing the search space
4. [Search direction] Using samples to help the search
5. [Noisy fitness] Adapting to stochastic problems
6. [Directions] Future work and timeline

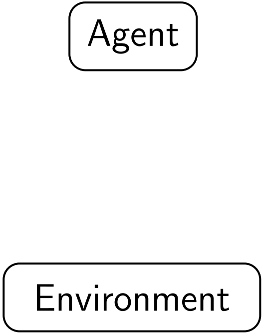
Content

1. [Context] Context of this PhD
2. [Policy search] Evolution Strategies for Policy Search
3. [Search space] Representing policies and changing the search space
4. [Search direction] Using samples to help the search
5. [Noisy fitness] Adapting to stochastic problems
6. [Directions] Future work and timeline

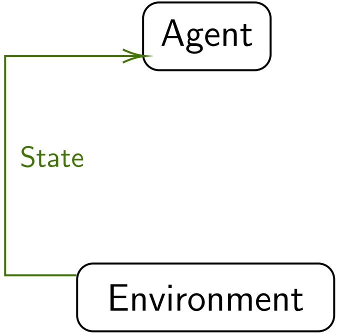
[Policy search] Policy

Environment

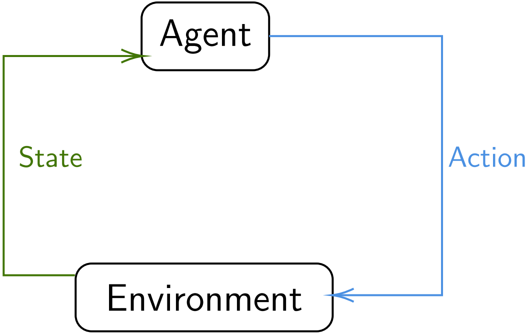
[Policy search] Policy



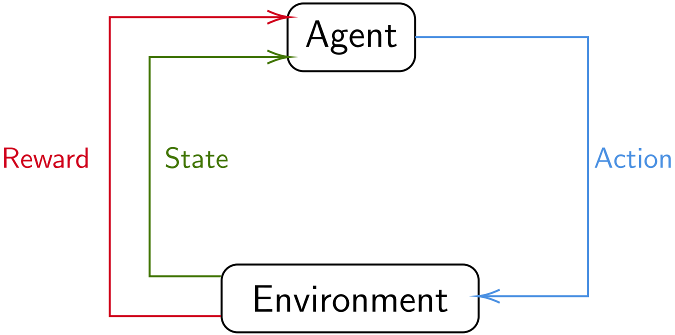
[Policy search] Policy



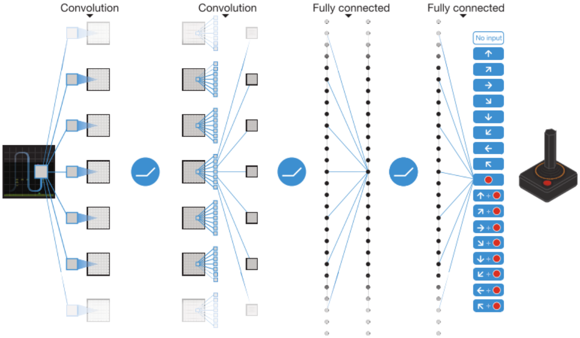
[Policy search] Policy



[Policy search] Policy



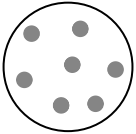
[Policy search] Neural networks



Neural Network used in Deep Q Networks [Mnih et al., 2015]

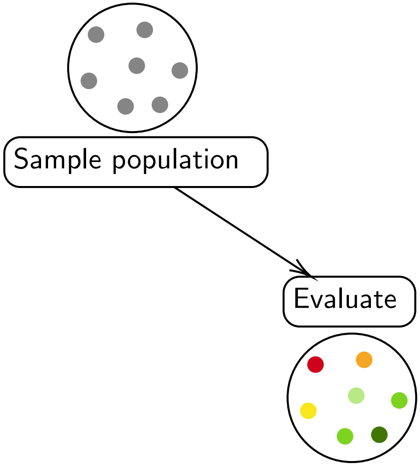
[Policy search] Evolution Strategies

[Policy search] Evolution Strategies

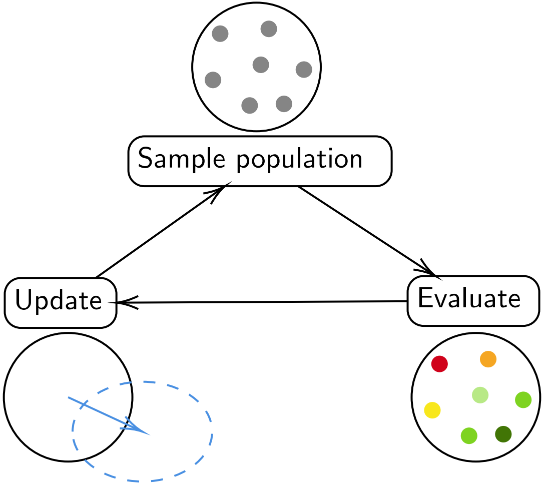


Sample population

[Policy search] Evolution Strategies



[Policy search] Evolution Strategies



[Policy search] Variants of Evolution Strategies

Fixed covariance

- (μ, λ) ES
- Canonical ES
- OpenAI ES

[Policy search] Variants of Evolution Strategies

Fixed covariance

- (μ, λ) ES
- Canonical ES
- OpenAI ES

Covariance matrix adaptation

- CMA-ES

[Policy search] Variants of Evolution Strategies

Fixed covariance

- (μ, λ) ES
- Canonical ES
- OpenAI ES

Covariance matrix adaptation

- CMA-ES

Natural gradient

- XNES
- SNES

[Policy search] Variants of Evolution Strategies

Fixed covariance

- (μ, λ) ES
- Canonical ES
- OpenAI ES

Covariance matrix adaptation

- CMA-ES

Natural gradient

- XNES
- SNES

Adjacent methods

- Cross-Entropy Method
- Augmented Random Search

[Policy search] Variants of Evolution Strategies

Fixed covariance

- (μ, λ) ES
- Canonical ES
- OpenAI ES

Covariance matrix adaptation

- CMA-ES

Natural gradient

- XNES
- SNES

Adjacent methods

- Cross-Entropy Method
- Augmented Random Search

Neuroevolution for policy search

- Large dimensions (10^6 parameters)
- Iterative evaluation

[Policy search] Canonical ES

σ - Mutation step-size
θ_0 - Initial policy parameters
F - Fitness function
λ - Offsprings population size
μ - Parents population size

[Policy search] Canonical ES

σ - Mutation step-size

θ_0 - Initial policy parameters

F - Fitness function

λ - Offsprings population size

μ - Parents population size

$w_i = \frac{log(\mu+0.5)-log(i)}{\sum_{j=1}^{\mu} log(\mu+0.5)-log(j)} \quad \forall i = 1...\lambda$

[Policy search] Canonical ES

σ - Mutation step-size
θ_0 - Initial policy parameters
F - Fitness function
λ - Offsprings population size
μ - Parents population size
$w_i = \frac{log(\mu+0.5)-log(i)}{\sum_{j=1}^{\mu} log(\mu+0.5)-log(j)} \quad \forall i = 1...\lambda$
for t=0, 1, ... **do**

end for

[Policy search] Canonical ES

σ - Mutation step-size
θ_0 - Initial policy parameters
F - Fitness function
λ - Offsprings population size
μ - Parents population size
$w_i = \frac{log(\mu+0.5) - log(i)}{\sum_{j=1}^{\mu} log(\mu+0.5) - log(j)} \quad \forall i = 1...\lambda$
for t=0, 1, ... **do**
  for i=0, 1, ... λ **do**
    Sample noise: $\epsilon_i \sim N(0, I)$
    Evaluate score: $s_i \leftarrow F(\theta_t + \sigma\epsilon_i)$
  end for

end for

[Policy search] Canonical ES

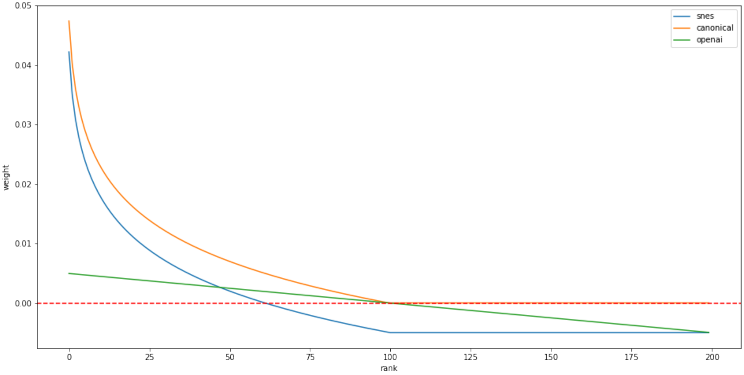
σ - Mutation step-size
θ_0 - Initial policy parameters
F - Fitness function
λ - Offsprings population size
μ - Parents population size
$w_i = \frac{log(\mu+0.5)-log(i)}{\sum_{j=1}^{\mu} log(\mu+0.5)-log(j)} \quad \forall i = 1...\lambda$
for t=0, 1, ... **do**
  for i=0, 1, ... λ **do**
    Sample noise: $\epsilon_i \sim N(0, I)$
    Evaluate score: $s_i \leftarrow F(\theta_t + \sigma\epsilon_i)$
  end for
  Sort $(\epsilon_1, ..., \epsilon_\lambda)$ according to s (ϵ_i with best s_i first)

end for

[Policy search] Canonical ES

σ - Mutation step-size
θ_0 - Initial policy parameters
F - Fitness function
λ - Offsprings population size
μ - Parents population size
$w_i = \frac{log(\mu+0.5)-log(i)}{\sum_{j=1}^{\mu} log(\mu+0.5)-log(j)} \quad \forall i = 1...\lambda$
for t=0, 1, ... **do**
    for i=0, 1, ... λ **do**
        Sample noise: $\epsilon_i \sim N(0, I)$
        Evaluate score: $s_i \leftarrow F(\theta_t + \sigma\epsilon_i)$
    end for
    Sort $(\epsilon_1, ..., \epsilon_\lambda)$ according to s (ϵ_i with best s_i first)
    Update policy: $\theta_{t+1} \leftarrow \theta_t + \sigma \sum_{j=1}^{\mu} w_j \epsilon_j$
end for

[Policy search] Utility

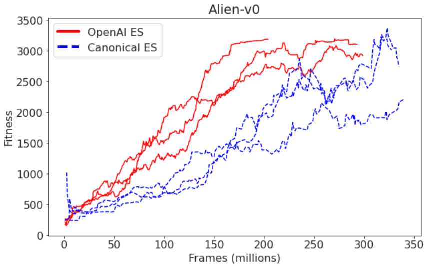


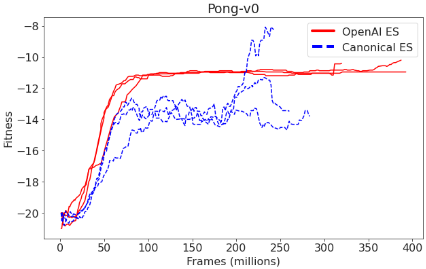
Utility values of the individuals in a ranked population for λ=200; μ=100

[Policy search] Benchmarking Evolutionary Reinforcement Learning

Reproduction settings

Reproducing Canonical ES [Chrabaszcz et al., 2018] and OpenAI ES [Salimans et al., 2017] on the Arcade Learning Environment.





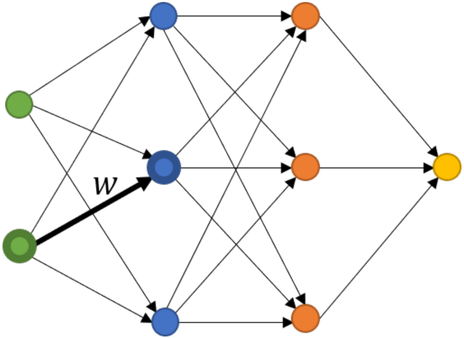
Evolution of Canonical ES and OpenAI ES on Alien and Pong with 800 CPUh compute budget

Content

1. [Context] Context of this PhD
2. [Policy search] Evolution Strategies for Policy Search
3. [Search space] Representing policies and changing the search space
4. [Search direction] Using samples to help the search
5. [Noisy fitness] Adapting to stochastic problems
6. [Directions] Future work and timeline

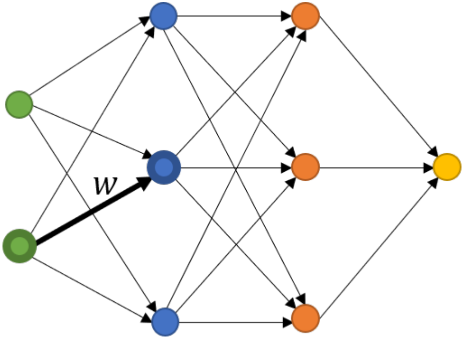
[Search space] A Geometric Encoding for Neural Network Evolution

[Search space] A Geometric Encoding for Neural Network Evolution

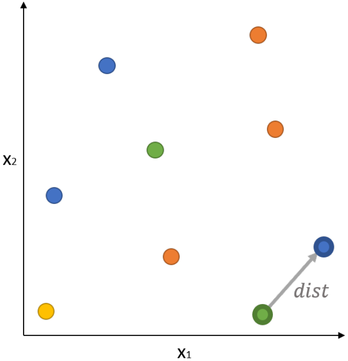


Fully connected
neural network

[Search space] A Geometric Encoding for Neural Network Evolution



Fully connected
neural network



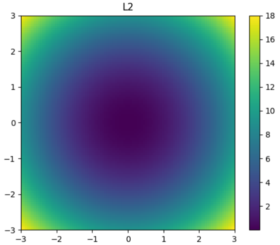
GENE encoding
[Templier et al., 2021]

[Search space] GENE: Distance functions

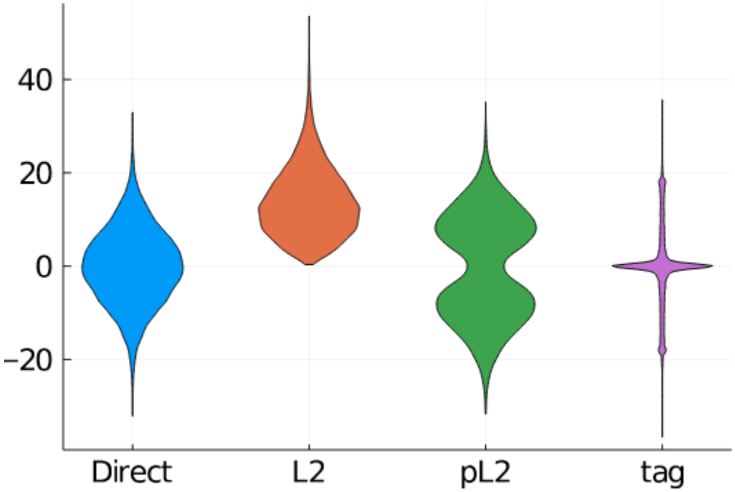
$$w_{i,j} = dist(n_i, n_j) \tag{1}$$

Euclidean distance

$$\sqrt{\sum_{k=1}^{D} \left(n_1^k - n_2^k\right)^2} \tag{2}$$

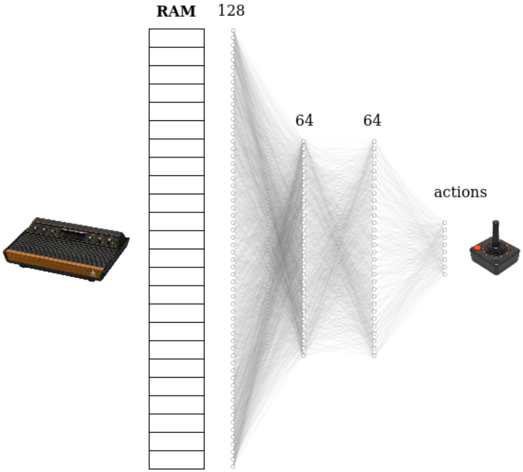


[Search space] GENE: Weight distribution



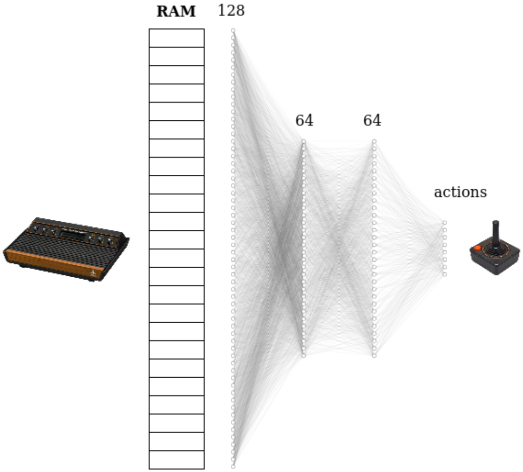
Distribution of weight values in networks evolved with different encodings.

[Search space] Experimental setup



RAM state representation

[Search space] Experimental setup



RAM state representation

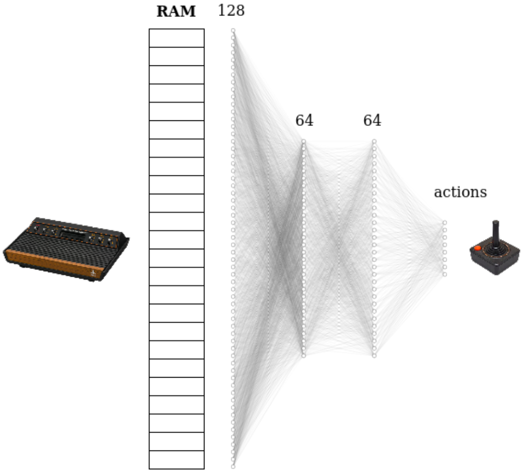
SNES

- Separable NES
- Complexity in $O(n)$

XNES

- Exponential NES
- Complexity in $O(n^2)$

[Search space] Experimental setup



RAM state representation

SNES

- Separable NES
- Complexity in $O(n)$

XNES

- Exponential NES
- Complexity in $O(n^2)$

Encodings

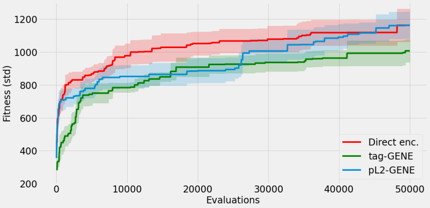
- Direct encoding
- GENE: dim=3
- 10 runs

[Search space] Computational cost

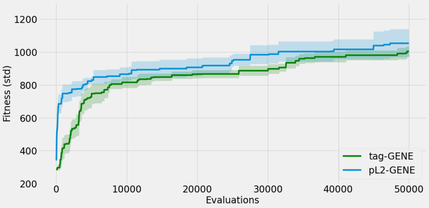
Evolutionary Strategy update of μ and σ

Encoding	D	Genes		Mean time (s)	Memory (KiB)
pL2-GENE	3	804	SNES	0.000357	630.56
pL2-GENE	10	2211	SNES	0.000678	1372.16
Direct	-	5609	SNES	0.001350	3133.44
pL2-GENE	3	804	XNES	1.475000	1352663.04
pL2-GENE	10	2211	XNES	14.244000	11806965.76
Direct	-	5609	XNES	119.976000	79765176.32

[Search space] Competitive results - Arcade Learning Environment

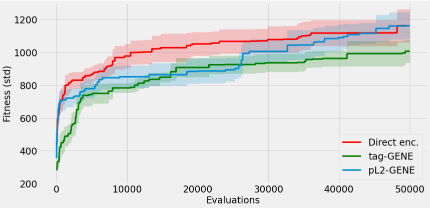


SNES on SpaceInvaders

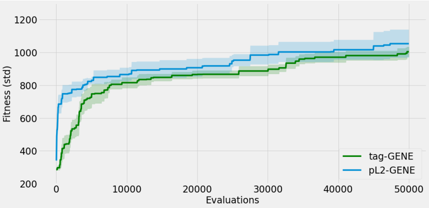


XNES on SpaceInvaders

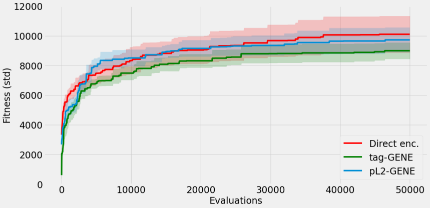
[Search space] Competitive results - Arcade Learning Environment



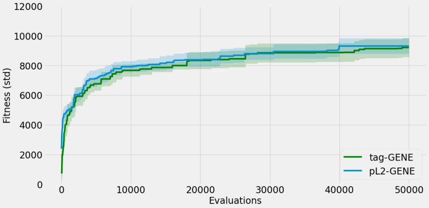
SNES on SpaceInvaders



XNES on SpaceInvaders

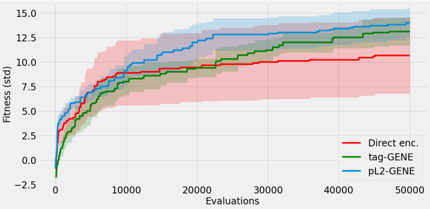


SNES on Krull

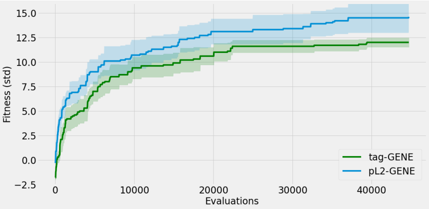


XNES on Krull

[Search space] Improving results - Arcade Learning Environment

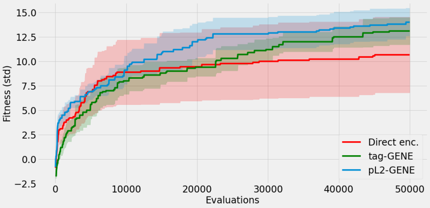


SNES on IceHockey

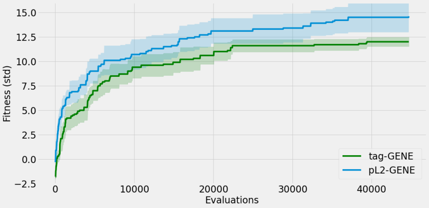


XNES on IceHockey

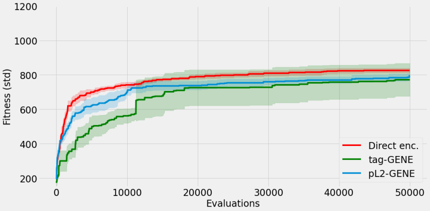
[Search space] Improving results - Arcade Learning Environment



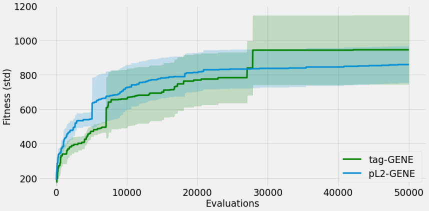
SNES on IceHockey



XNES on IceHockey



SNES on Seaquest



XNES on Seaquest

[Search space] Future Work

Distance functions

Design new distance functions, or optimize them through co-evolution.

Hybrid encoding

Switch between indirect and direct encodings during the evolution.

Gradient descent

Use backpropagation and gradient descent to optimize genomes instead of evolution.

Complex networks

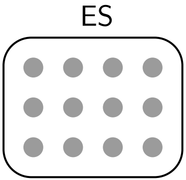
Design encodings for convolution layers and recurrent networks.

Content

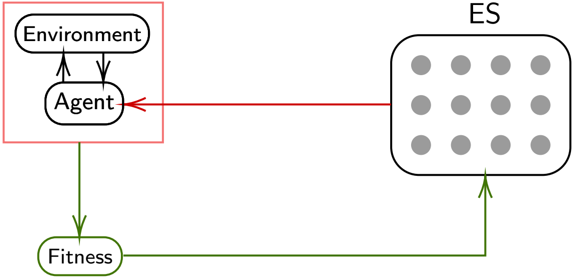
1. [Context] Context of this PhD
2. [Policy search] Evolution Strategies for Policy Search
3. [Search space] Representing policies and changing the search space
4. [Search direction] Using samples to help the search
5. [Noisy fitness] Adapting to stochastic problems
6. [Directions] Future work and timeline

[Search direction] Using samples to drive the search

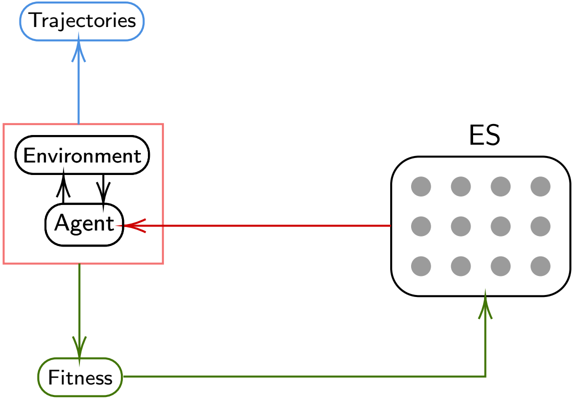
[Search direction] Using samples to drive the search



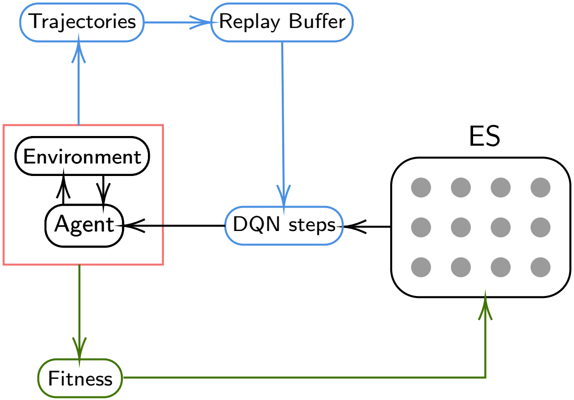
[Search direction] Using samples to drive the search



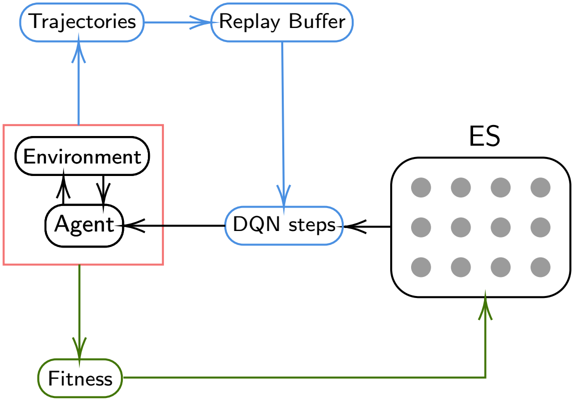
[Search direction] Using samples to drive the search



[Search direction] Using samples to drive the search



[Search direction] Using samples to drive the search

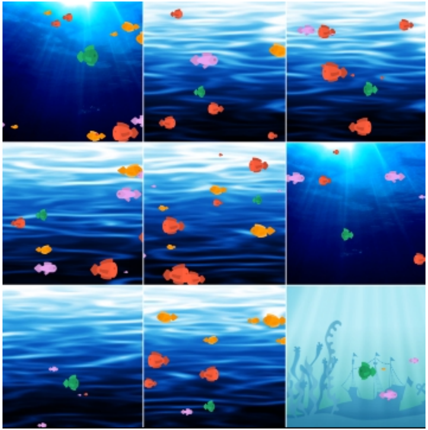


CEM-RL: CEM + Actor-Critic [Pourchot and Sigaud, 2019]

Content

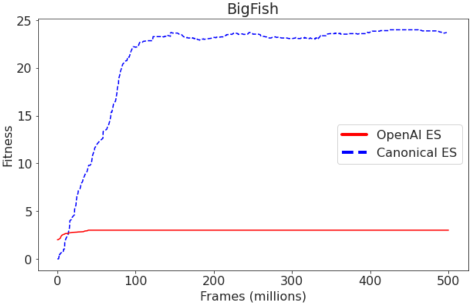
1. [Context] Context of this PhD
2. [Policy search] Evolution Strategies for Policy Search
3. [Search space] Representing policies and changing the search space
4. [Search direction] Using samples to help the search
5. [Noisy fitness] Adapting to stochastic problems
6. [Directions] Future work and timeline

[Noisy fitness] Stochastic fitness



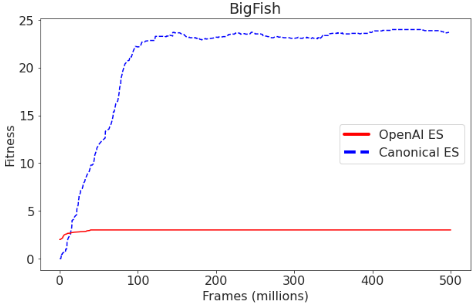
BigFish levels generated from different seeds

[Noisy fitness] ES on stochastic environments

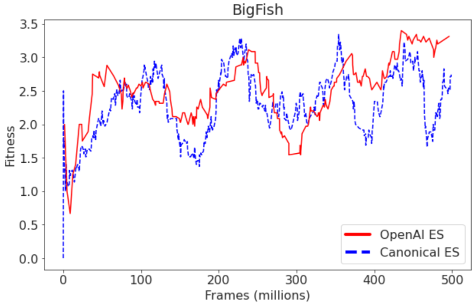


ES on BigFish, same level

[Noisy fitness] ES on stochastic environments



ES on BigFish, same level



ES on BigFish, random level

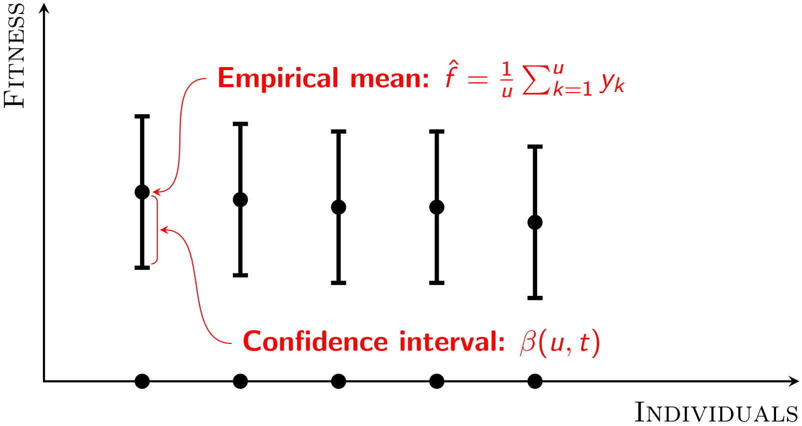
[Noisy fitness] The LUCIE selection procedure

[Noisy fitness] The `LUCIE` selection procedure

Objective: identify the **best** μ individuals with as **few evaluations** as possible.
[Lecarpentier et al., 2022]

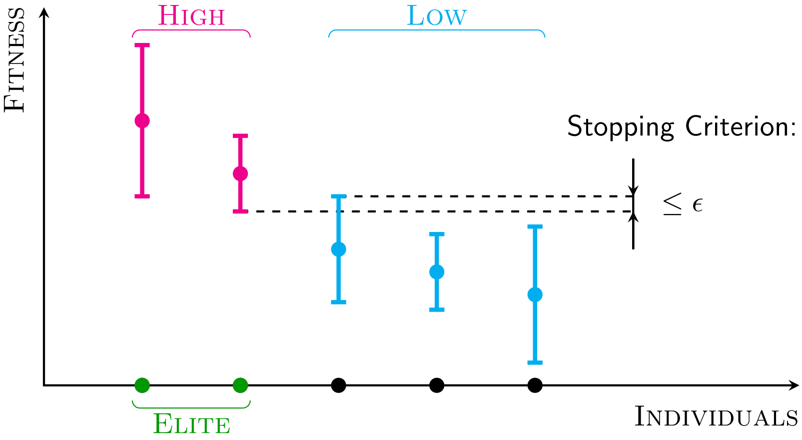
[Noisy fitness] The `LUCIE` selection procedure

Objective: identify the **best** μ individuals with as **few evaluations** as possible.
[Lecarpentier et al., 2022]



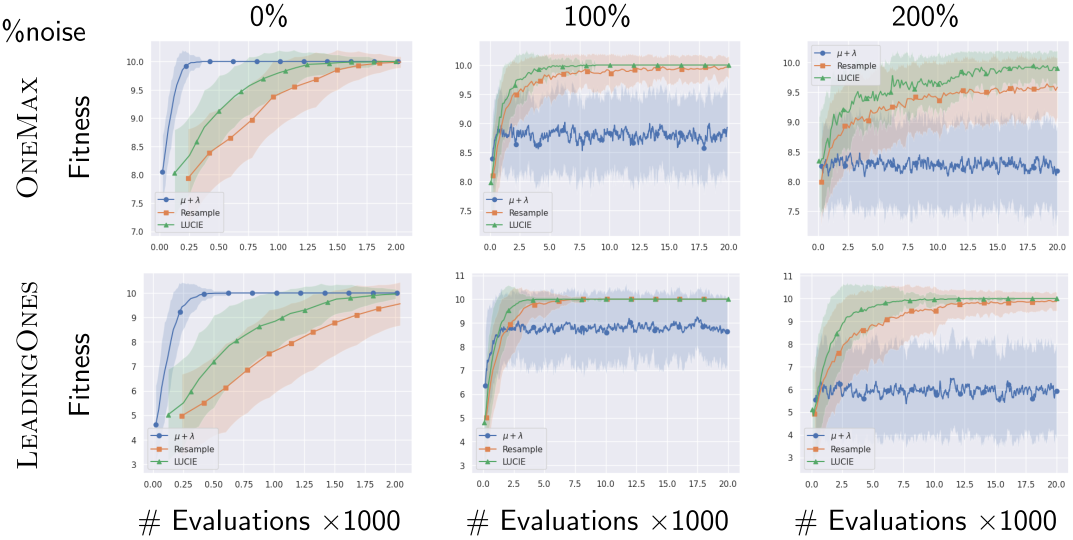
[Noisy fitness] The `LUCIE` selection procedure

Objective: identify the **best** μ individuals with as **few evaluations** as possible.
[Lecarpentier et al., 2022]

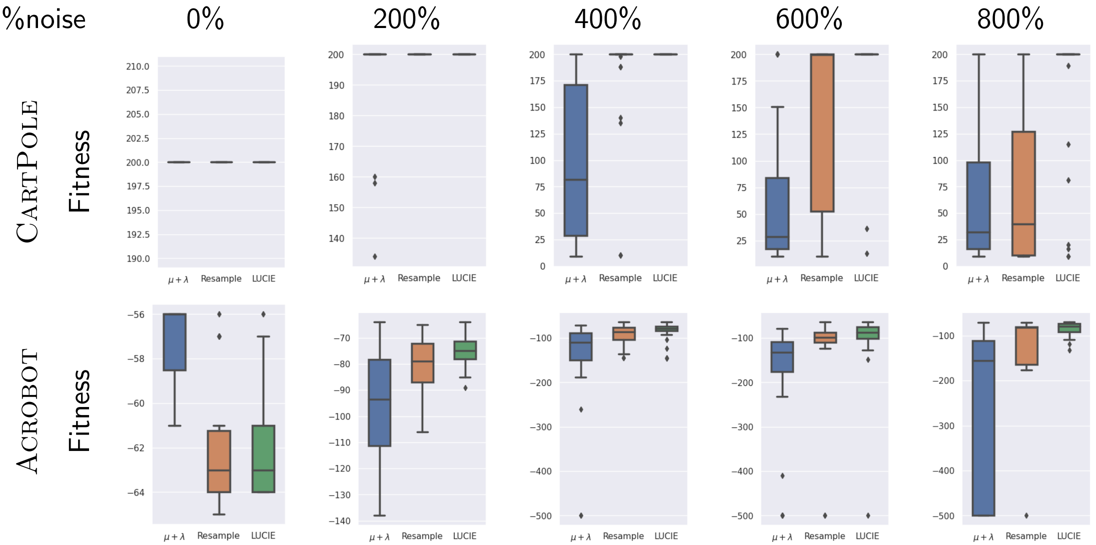


[Noisy fitness] OneMax and LeadingOnes

[Noisy fitness] OneMax and LeadingOnes



[Noisy fitness] Classic Control



[Noisy fitness] `LUCIE` for Evolution Strategies

[Noisy fitness] LUCIE for Evolution Strategies

Bandit problem

Selecting which individuals to evaluate

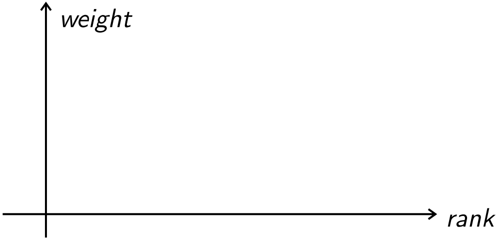
- Split
- Rank

[Noisy fitness] `LUCIE` for Evolution Strategies

Bandit problem

Selecting which individuals to evaluate

- Split
- Rank

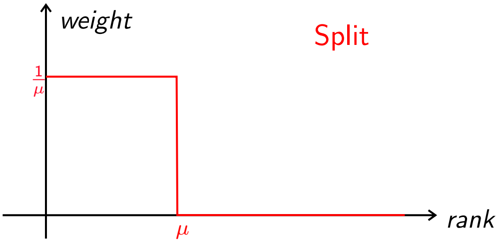


[Noisy fitness] `LUCIE` for Evolution Strategies

Bandit problem

Selecting which individuals to evaluate

- Split
- Rank

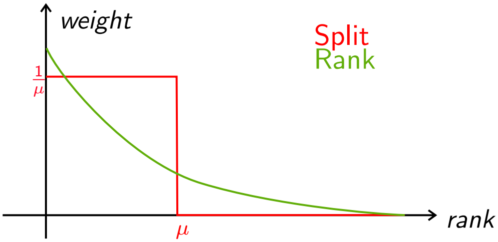


[Noisy fitness] LUCIE for Evolution Strategies

Bandit problem

Selecting which individuals to evaluate

- Split
- Rank



[Noisy fitness] LUCIE for Evolution Strategies

Bandit problem

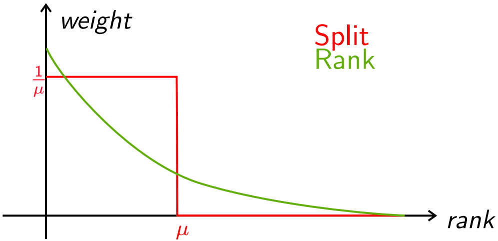
Selecting which individuals to evaluate

- Split
- Rank

Population mixing

Keeping evaluated individuals

- Elitist ES
- Importance Mixing [Pourchot et al., 2018]



[Noisy fitness] LUCIE for Evolution Strategies

Bandit problem

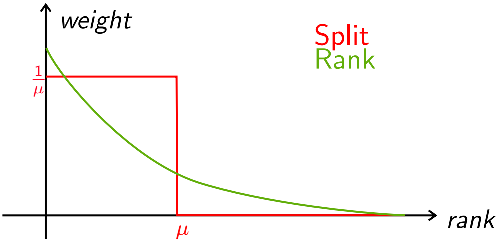
Selecting which individuals to evaluate

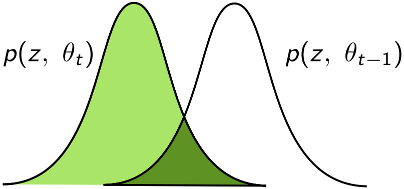
- Split
- Rank

Population mixing

Keeping evaluated individuals

- Elitist ES
- Importance Mixing
  [Pourchot et al., 2018]





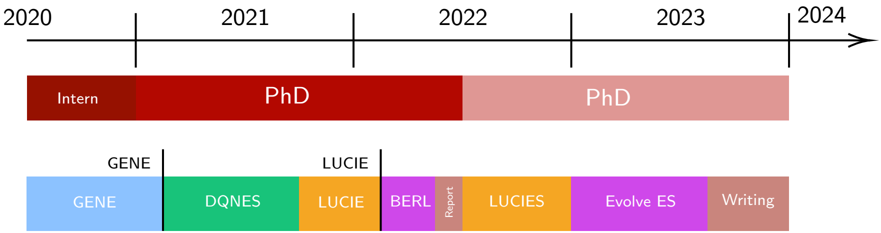
Content

1. [Context] Context of this PhD
2. [Policy search] Evolution Strategies for Policy Search
3. [Search space] Representing policies and changing the search space
4. [Search direction] Using samples to help the search
5. [Noisy fitness] Adapting to stochastic problems
6. [Directions] Future work and timeline

[Directions] Future work

LUCI ES

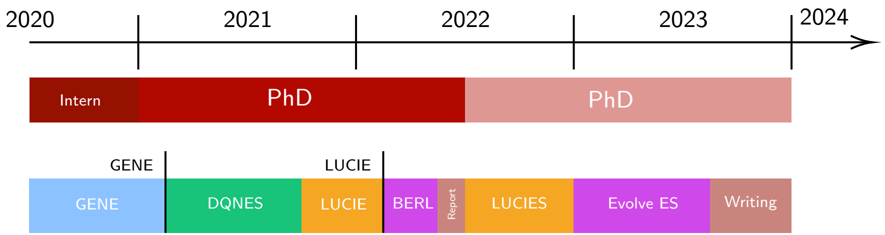
- Explore (μ, λ) ES
- Ranking in Bandit problems
- Heritage (Importance Mixing, elitism)
- Scalability



[Directions] Future work

Evolving Evolution Strategies

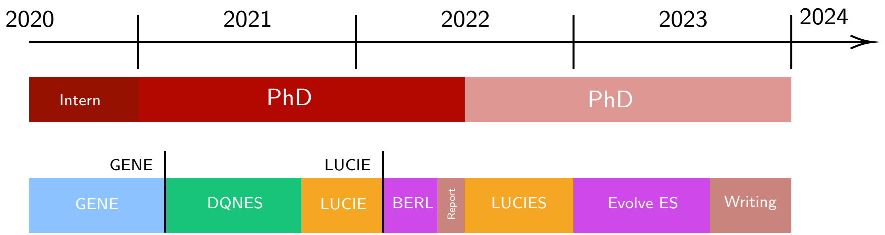
- Make ES methods emerge from scratch
- Neuromodulation: adapting ES during the evolution



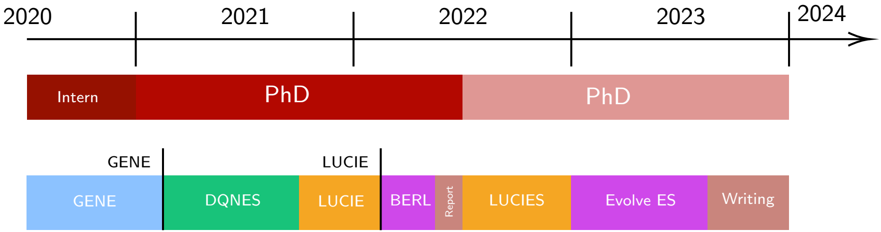
[Directions] Future work

ES for Policy Search

- Neuroevolution constraints and theory
- Ablation study of existing methods



[Directions] Future work



References I

Chrabaszcz, P., Loshchilov, I., and Hutter, F. (2018).
Back to Basics: Benchmarking Canonical Evolution Strategies for Playing Atari.
pages 1419–1426.

Lecarpentier, E., Templier, P., Rachelson, E., and Wilson, D. G. (2022).
LUCIE: An Evaluation and Selection Method for Stochastic Problems.
In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2022)*.

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.

Pourchot, A., Perrin, N., and Sigaud, O. (2018).
Importance mixing: Improving sample reuse in evolutionary policy search methods.

References II

Pourchot, A. and Sigaud, O. (2019).
CEM-RL: Combining evolutionary and gradient-based methods for policy search.
arXiv:1810.01222 [cs, stat].

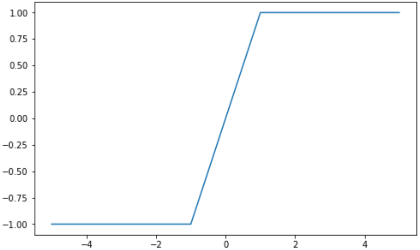
Salimans, T., Ho, J., Chen, X., Sidor, S., and Sutskever, I. (2017).
Evolution Strategies as a Scalable Alternative to Reinforcement Learning.

Templier, P., Rachelson, E., and Wilson, D. G. (2021).
A Geometric Encoding for Neural Network Evolution.
page 9.

[Search space] Signed distances

Bounded identity function

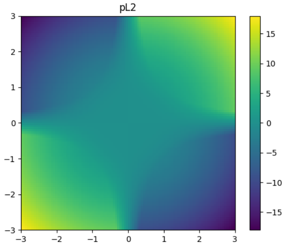
$$\alpha : \begin{cases} \text{if } x \geq 1 : \alpha(x) = 1 \\ \text{if } x \leq -1 : \alpha(x) = -1 \\ \text{else: } \alpha(x) = x \end{cases} \tag{3}$$



[Search space] Distance functions

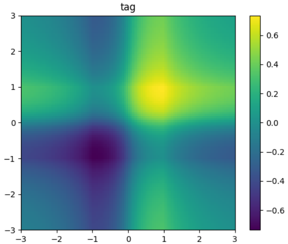
pL2-GENE

$$\alpha \left(\prod_{k=1}^{D} n_1^k - n_2^k \right) \sqrt{\sum_{j=1}^{D} \left(n_1^j - n_2^j \right)^2} \quad (4)$$



tag-GENE

$$\sum_{j=2}^{D} \alpha(n_1^j - n_2^1)e^{-|n_1^j - n_2^1|} \quad (5)$$



[Noisy fitness] The `LUCIE` selection procedure

[Noisy fitness] The `LUCIE` selection procedure

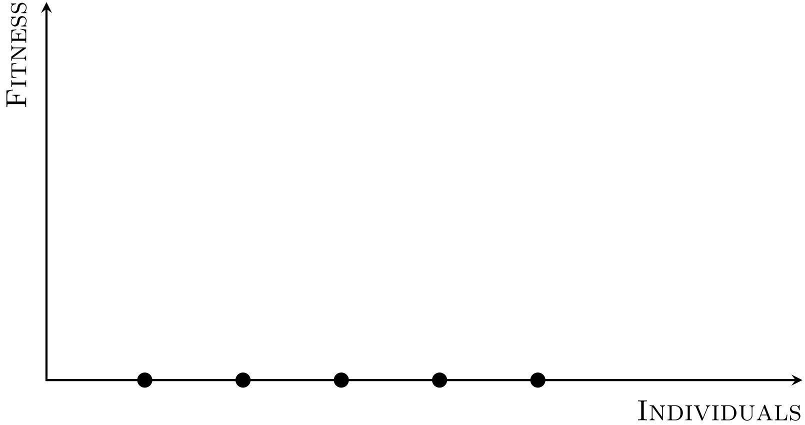
Objective: identify the **best μ** individuals with as **few evaluations** as possible.

[Noisy fitness] The `LUCIE` selection procedure

Objective: identify the **best** μ individuals with as **few evaluations** as possible.

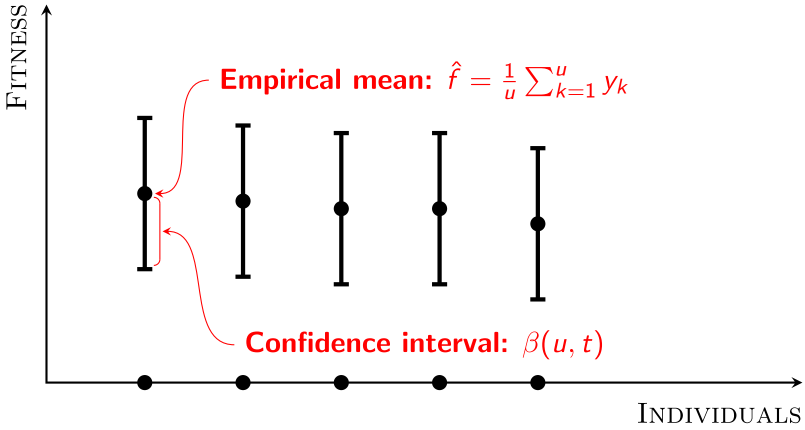
[Noisy fitness] The LUCIE selection procedure

Objective: identify the **best** μ individuals with as **few evaluations** as possible.



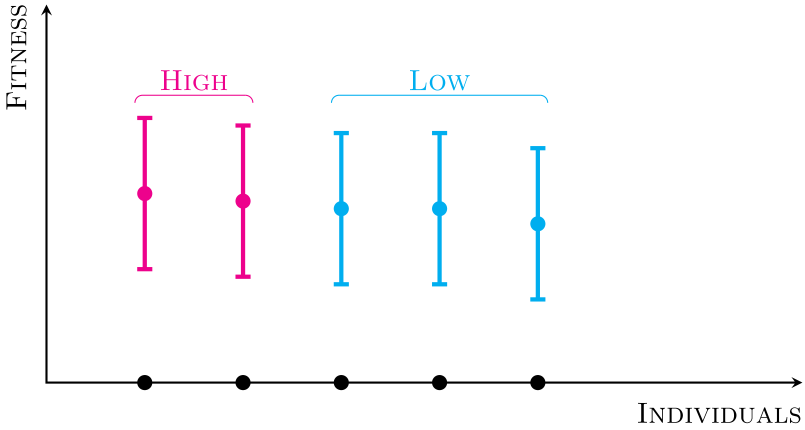
 The LUCIE selection procedure

Objective: identify the **best** μ individuals with as **few evaluations** as possible.



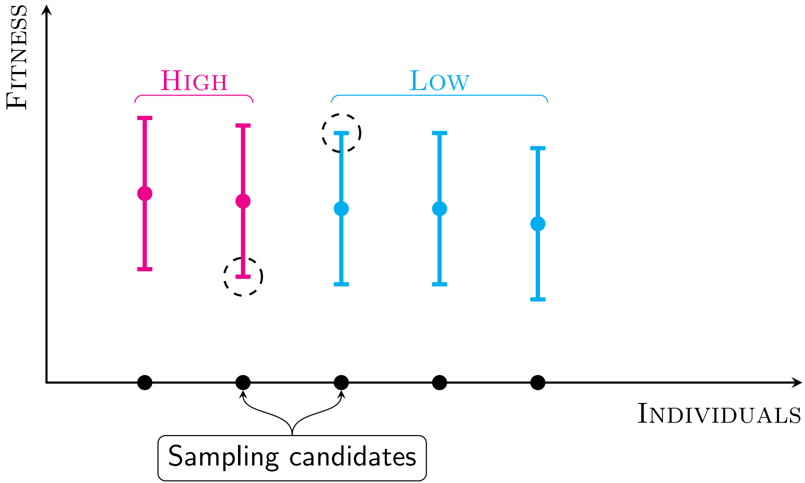
[Noisy fitness] The LUCIE selection procedure

Objective: identify the **best** μ individuals with as **few evaluations** as possible.



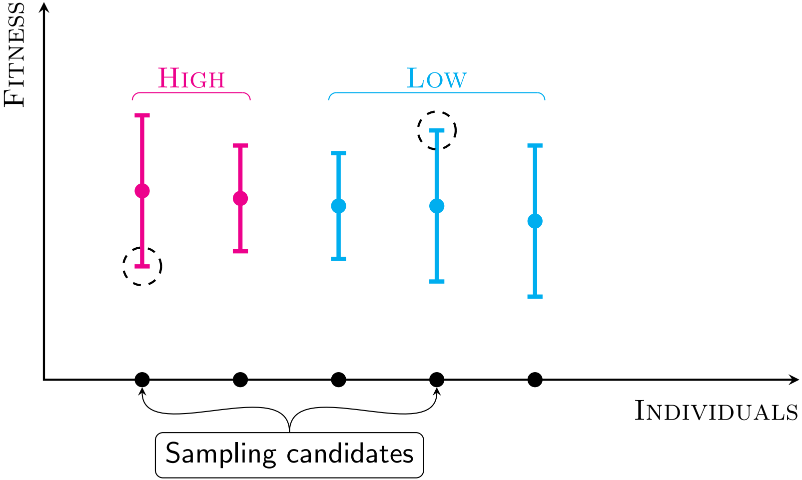
[Noisy fitness] The LUCIE selection procedure

Objective: identify the **best** μ individuals with as **few evaluations** as possible.



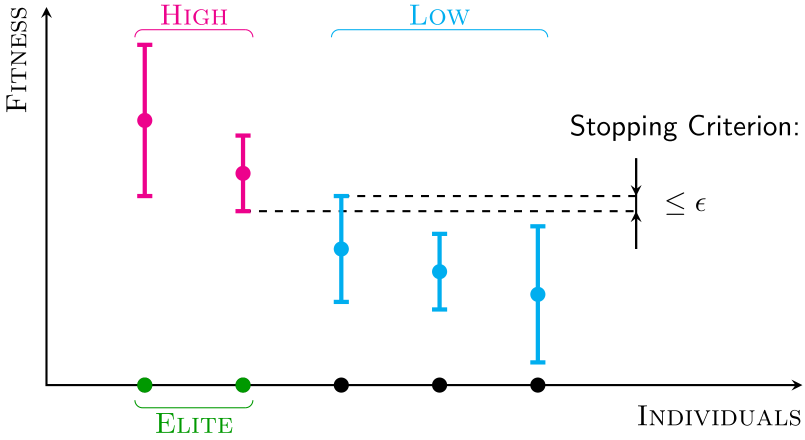
[Noisy fitness] The LUCIE selection procedure

Objective: identify the **best** μ individuals with as **few evaluations** as possible.



[Noisy fitness] The `LUCIE` selection procedure

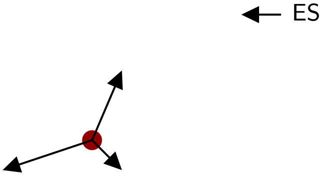
Objective: identify the **best** μ individuals with as **few evaluations** as possible.



[Search direction] DQNES

[Search direction] DQNES

[Search direction] DQNES



[Search direction] DQNES