

Evolution Strategies for Neural Policy Search

Mid-thesis committee

Author: Paul Templier¹

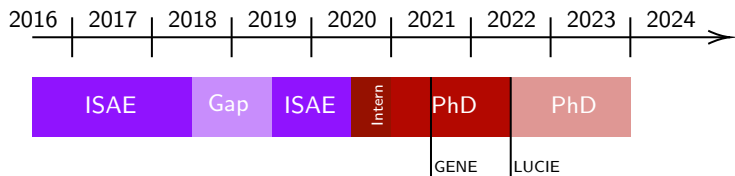
Advisors: Emmanuel Rachelson¹, Dennis G. Wilson¹

[paul.templier@isae-supaero.fr]

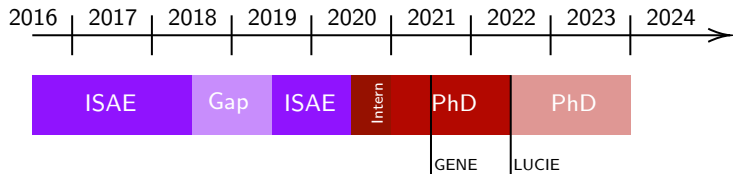
June 29, 2022

¹ University of Toulouse, ISAE-SUPAERO

[Context] Mid-thesis report



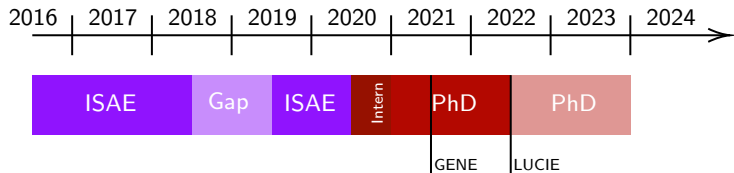
[Context] Mid-thesis report



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

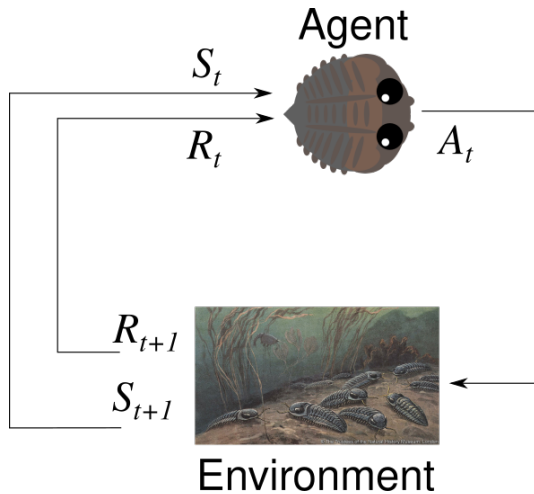
1. [Context] Context of this PhD
2. [Policy search] Evolution Strategies for Policy Search
3. [Search space] Representing policies and changing the search space
4. [Search direction] Using samples to help the search
5. [Noisy fitness] Adapting to stochastic problems

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

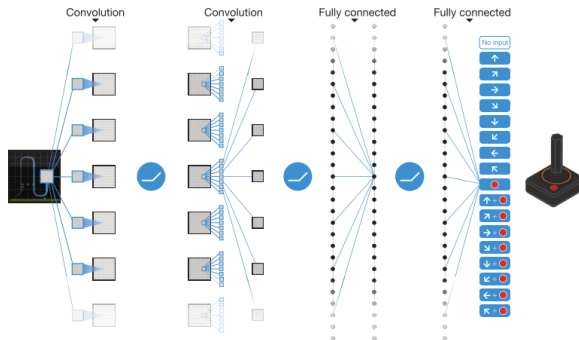
[Policy search] Policy search

[Policy search] Policy search



<https://github.com/d9w/evolution/blob/master/imgs/er1.png>

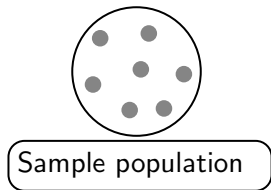
[Policy search] Neural networks



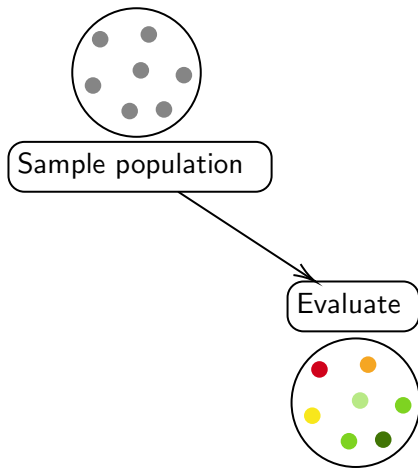
Neural Network used in Deep Q Networks [3]

[Policy search] Evolution Strategies

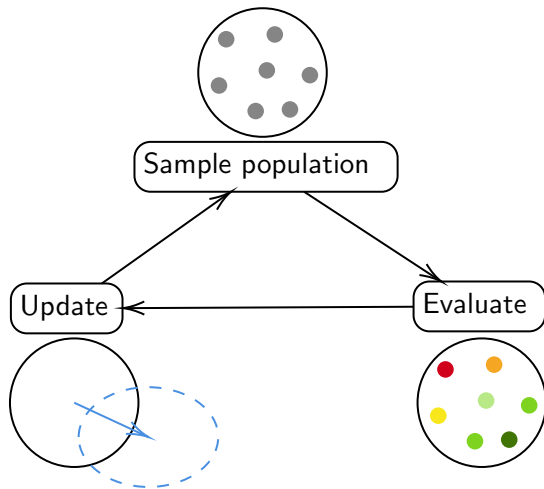
[Policy search] Evolution Strategies



[Policy search] Evolution Strategies



[Policy search] Evolution Strategies



[Policy search] Variants of Evolution Strategies

Evolution Strategies

- ▶ (μ, λ) ES
- ▶ SNES
- ▶ Canonical ES
- ▶ OpenAI ES
- ▶ CMA-ES
- ▶ XNES
- ▶ Cross-Entropy Method
- ▶ Augmented Random Search

[Policy search] Variants of Evolution Strategies

Evolution Strategies

- ▶ (μ, λ) ES
- ▶ SNES
- ▶ Canonical ES
- ▶ OpenAI ES
- ▶ CMA-ES
- ▶ XNES
- ▶ Cross-Entropy Method
- ▶ Augmented Random Search

Neuroevolution for policy search

- ▶ large dimensions ($1.6 \cdot 10^6$ parameters)
- ▶ expensive evaluation

[Policy search] Benchmarking Evolutionary Reinforcement Learning

Reproduction settings

Reproducing Canonical ES [1] and OpenAI ES [4] on the Arcade Learning Environment.

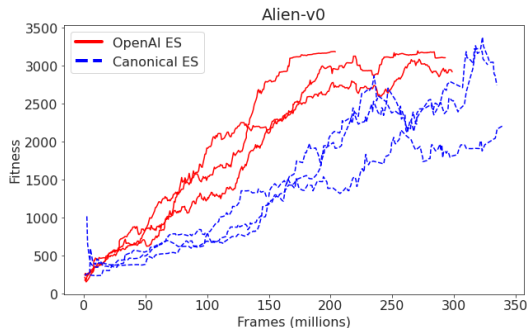
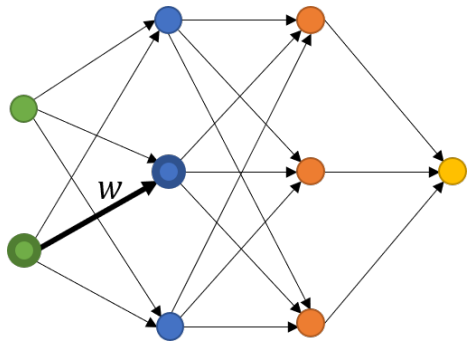


Figure: Evolution of Canonical ES and OpenAI ES on Alien with 800 CPUh compute budget

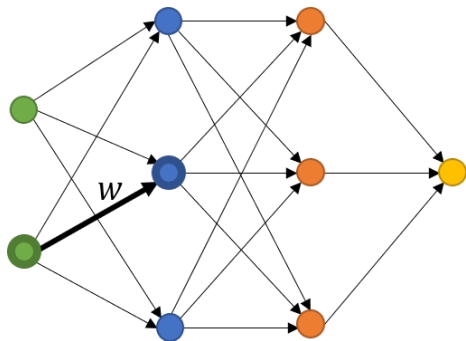
[Search space] A Geometric Encoding for Neural Network Evolution

[Search space] A Geometric Encoding for Neural Network Evolution

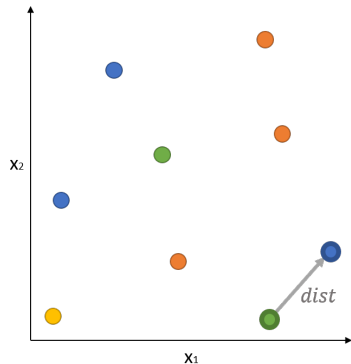


Fully connected
neural network

[Search space] A Geometric Encoding for Neural Network Evolution



Fully connected
neural network



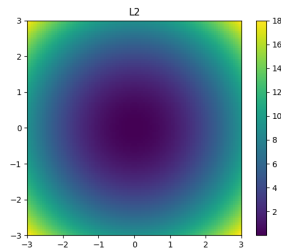
GENE encoding

[Search space] GENE: Distance functions

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

Euclidean distance

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



[Search space] GENE: Weight distribution

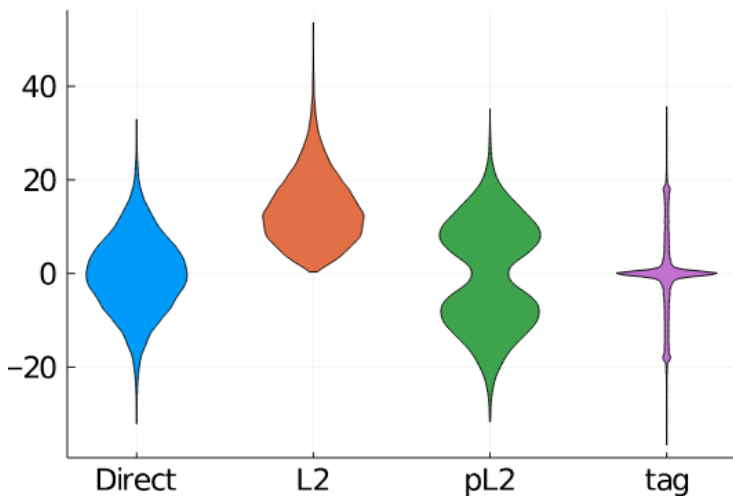


Figure: Distribution of weight values in networks evolved with different encodings.

[Search space] Competitive results - Arcade Learning Environment

[Search space] Competitive results - Arcade Learning Environment

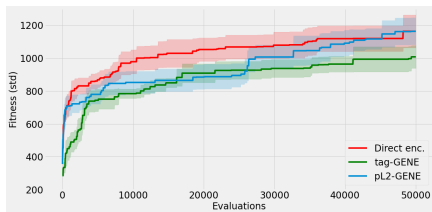


Figure: SNES on SpaceInvaders

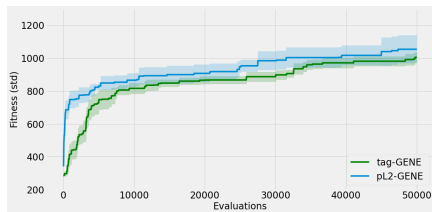


Figure: XNES on SpaceInvaders

[Search space] Competitive results - Arcade Learning Environment

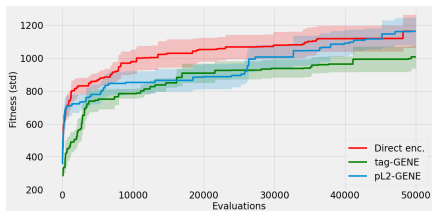


Figure: SNES on SpaceInvaders

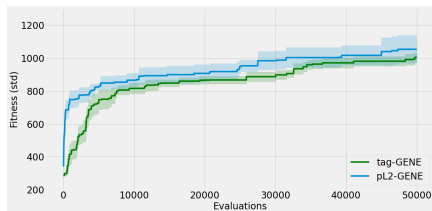


Figure: XNES on SpaceInvaders

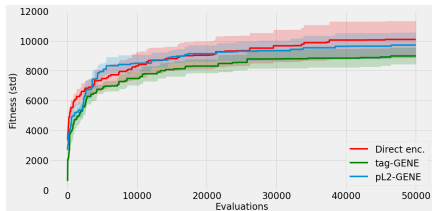


Figure: SNES on Krull

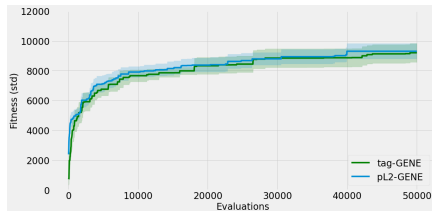


Figure: XNES on Krull

[Search space] Improving results - Arcade Learning Environment

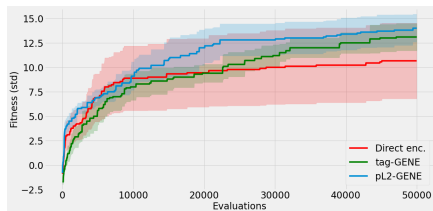


Figure: SNES on IceHockey

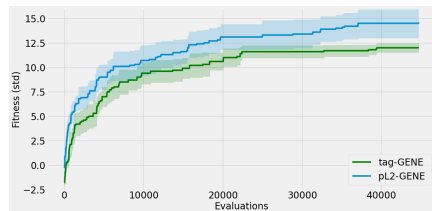


Figure: XNES on IceHockey

[Search space] Improving results - Arcade Learning Environment

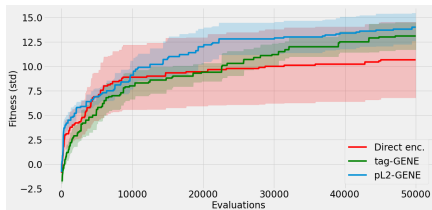


Figure: SNES on IceHockey

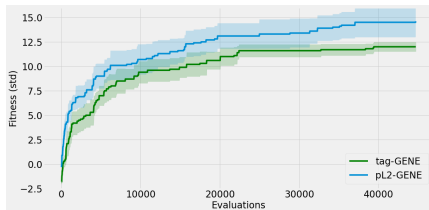


Figure: XNES on IceHockey

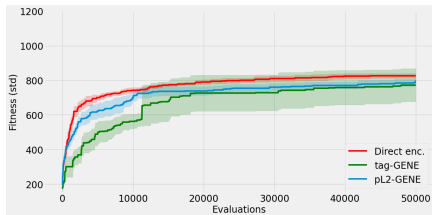


Figure: SNES on Seaquest

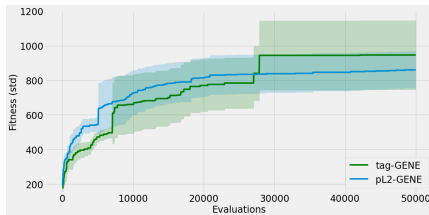


Figure: XNES on Seaquest

[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] 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

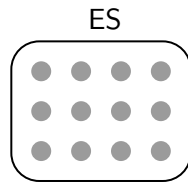
Use backpropagation and gradient descent to optimize genomes instead of evolution.

Complex networks

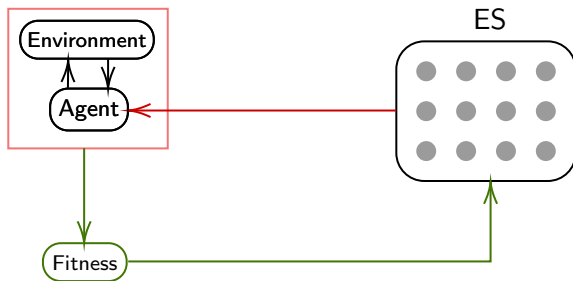
Design encodings for convolution layers and recurrent networks.

[Search direction] Using samples to drive the search

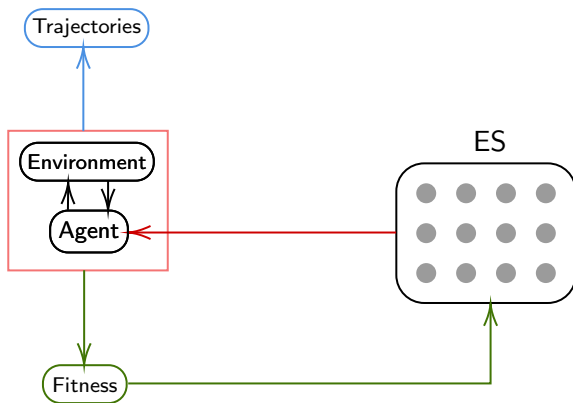
[Search direction] Using samples to drive the search



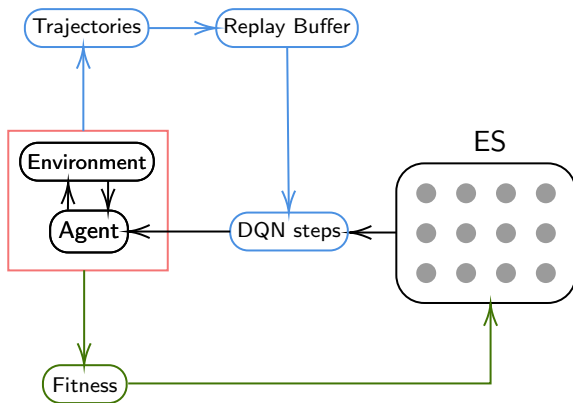
[Search direction] Using samples to drive the search



[Search direction] Using samples to drive the search



[Search direction] Using samples to drive the search



[Noisy fitness] ES on noisy environments

[Noisy fitness] ES on noisy environments

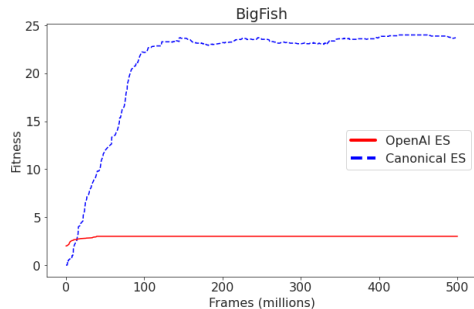


Figure: ES on BigFish, same level

[Noisy fitness] ES on noisy environments

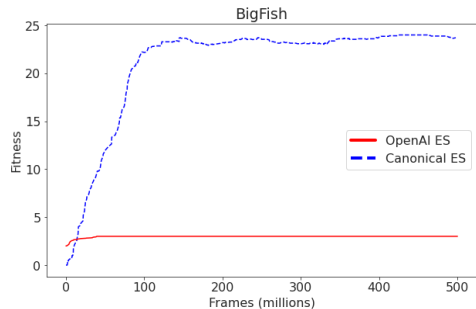


Figure: ES on BigFish, same level

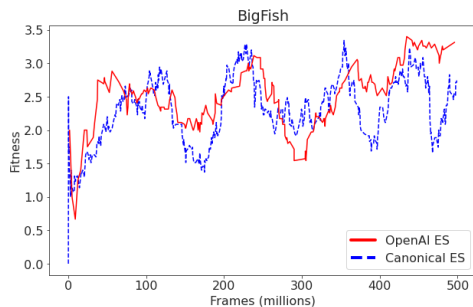


Figure: ES on BigFish, random level

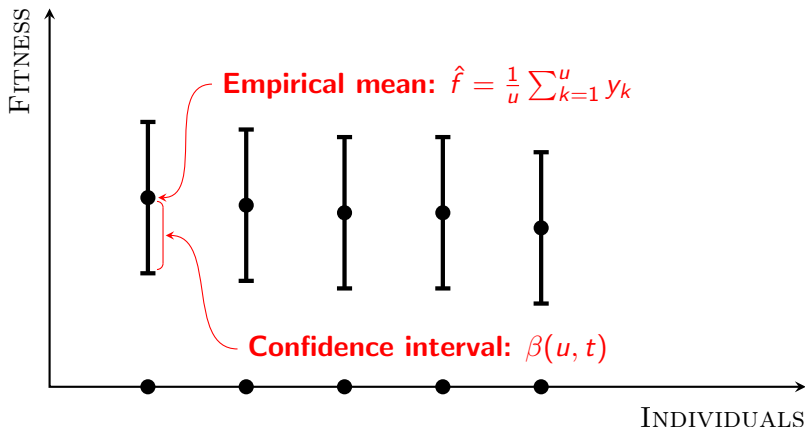
[Noisy fitness] The LUCIE selection procedure

[Noisy fitness] The LUCIE selection procedure

Objective: identify the **best** μ individuals with as **few evaluations** as possible. [2]

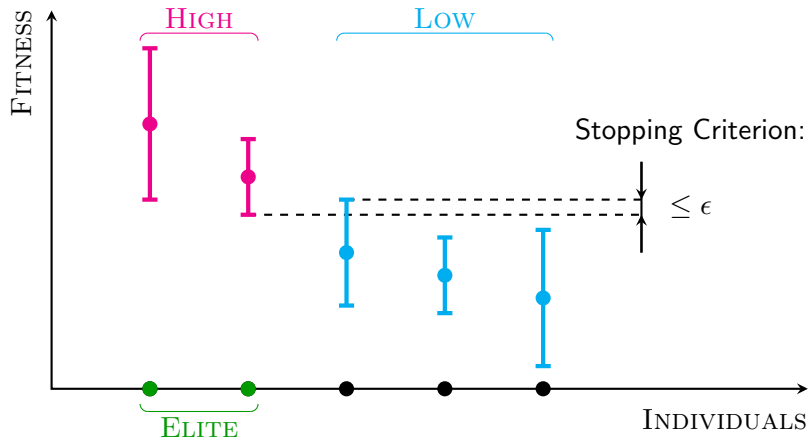
[Noisy fitness] The LUCIE selection procedure

Objective: identify the **best** μ individuals with as **few evaluations** as possible. [2]



[Noisy fitness] The LUCIE selection procedure

Objective: identify the **best** μ individuals with as **few evaluations** as possible. [2]



[Noisy fitness] ONEMAX and LEADINGONES

[Noisy fitness] ONEMAX and LEADINGONES

%noise

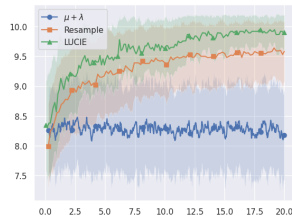
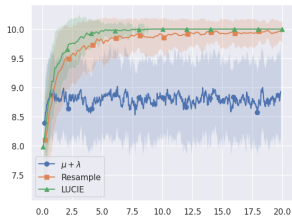
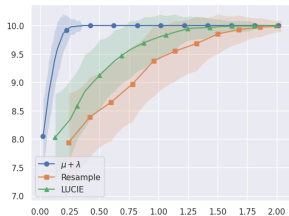
0%

100%

200%

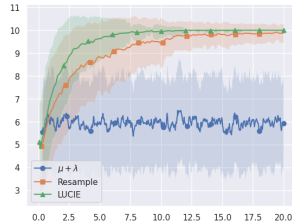
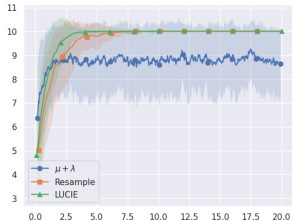
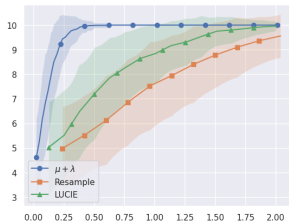
ONEMAX

Fitness



LEADINGONES

Fitness



Evaluations $\times 1000$

Evaluations $\times 1000$

Evaluations $\times 1000$

[Noisy fitness] LUCIE for Evolution Strategies

[Noisy fitness] LUCIE for Evolution Strategies

Bandit problem

Selecting which individuals to evaluate

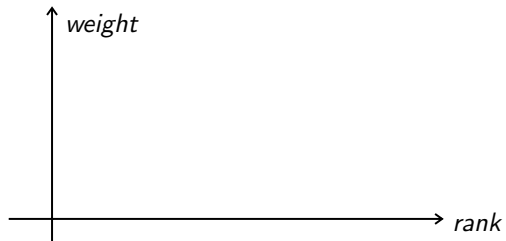
- ▶ Split
- ▶ Rank

[Noisy fitness] LUCIE for Evolution Strategies

Bandit problem

Selecting which individuals to evaluate

- ▶ Split
- ▶ Rank

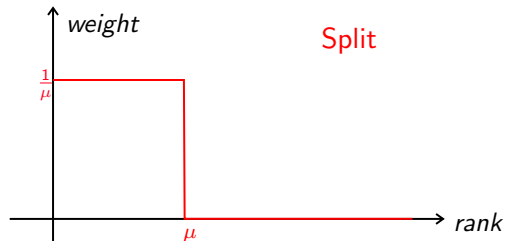


[Noisy fitness] LUCIE for Evolution Strategies

Bandit problem

Selecting which individuals to evaluate

- ▶ Split
- ▶ Rank

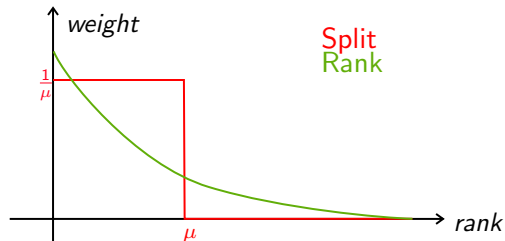


[Noisy fitness] LUCIE for Evolution Strategies

Bandit problem

Selecting which individuals to evaluate

- Split
- Rank



[Noisy fitness] LUCIE for Evolution Strategies

Bandit problem

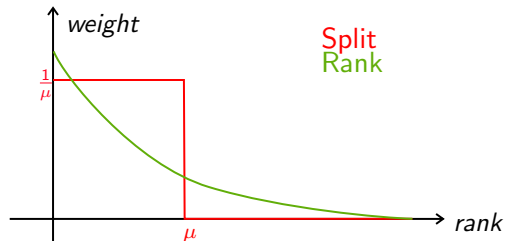
Selecting which individuals to evaluate

- ▶ Split
- ▶ Rank

Heritage

Keeping evaluated individuals

- ▶ Elitist ES
- ▶ Importance Mixing

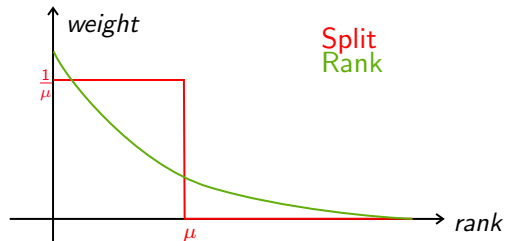


[Noisy fitness] LUCIE for Evolution Strategies

Bandit problem

Selecting which individuals to evaluate

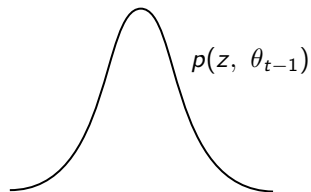
- ▶ Split
- ▶ Rank



Heritage

Keeping evaluated individuals

- ▶ Elitist ES
- ▶ Importance Mixing

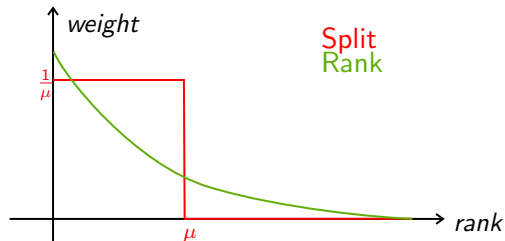


[Noisy fitness] LUCIE for Evolution Strategies

Bandit problem

Selecting which individuals to evaluate

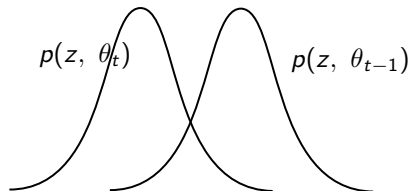
- Split
- Rank



Heritage

Keeping evaluated individuals

- Elitist ES
- Importance Mixing

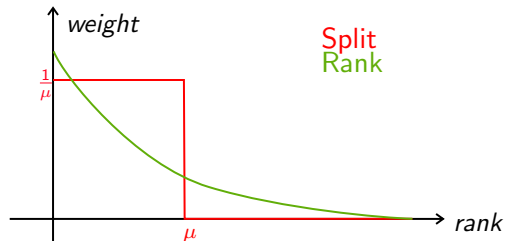


[Noisy fitness] LUCIE for Evolution Strategies

Bandit problem

Selecting which individuals to evaluate

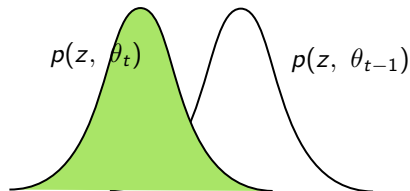
- Split
- Rank



Heritage

Keeping evaluated individuals

- Elitist ES
- Importance Mixing

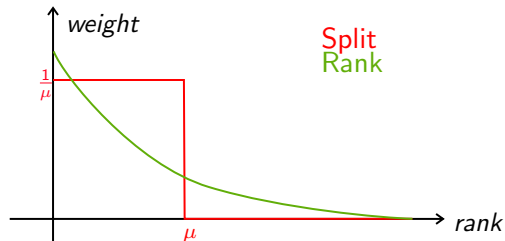


[Noisy fitness] LUCIE for Evolution Strategies

Bandit problem

Selecting which individuals to evaluate

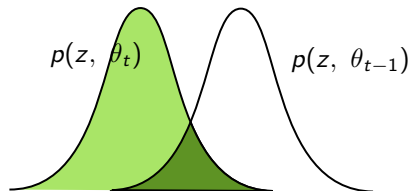
- Split
- Rank



Heritage

Keeping evaluated individuals

- Elitist ES
- Importance Mixing



[Directions] Future work

[Directions] Future work

LUCI ES

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

[Directions] Future work

LUCI ES

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

ES for Policy Search

- ▶ Neuroevolution constraints and theory
- ▶ Ablation study of existing methods

[Directions] Future work

LUCI ES

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

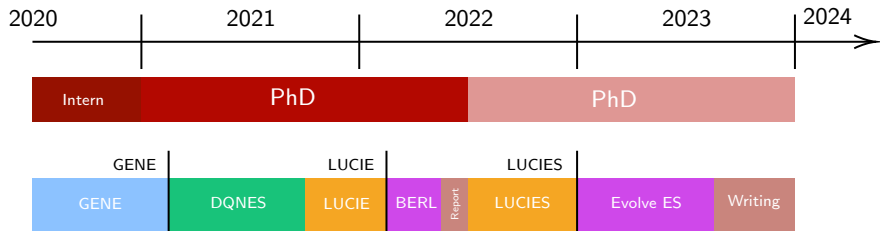
ES for Policy Search

- ▶ Neuroevolution constraints and theory
- ▶ Ablation study of existing methods

Evolving Evolution Strategies

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


[Directions] Timeline



References I

-  P. Chrabaszcz, I. Loshchilov, and F. Hutter.
Back to Basics: Benchmarking Canonical Evolution Strategies for Playing Atari.
pages 1419–1426, 2018.
-  E. Lecarpentier, P. Templier, E. Rachelson, and D. G. Wilson.
LUCIE: An Evaluation and Selection Method for Stochastic Problems.
In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2022)*, 2022.
-  V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al.
Human-level control through deep reinforcement learning.
nature, 518(7540):529–533, 2015.

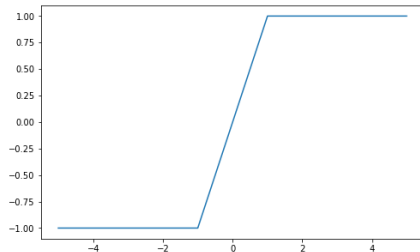
References II

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

[Search space] Signed distances

Bounded identity function

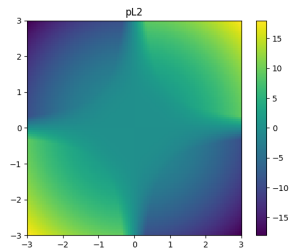
$$\alpha : \begin{cases} \text{if } x \geq 1 : \alpha(x) = 1 \\ \text{if } x \leq -1 : \alpha(x) = -1 \\ \text{else: } \alpha(x) = x \end{cases} \quad (3)$$



[Search space] Distance functions

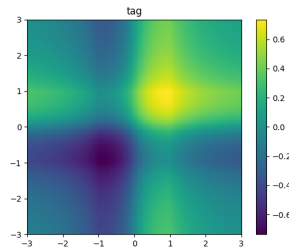
pL2-GENE

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



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.

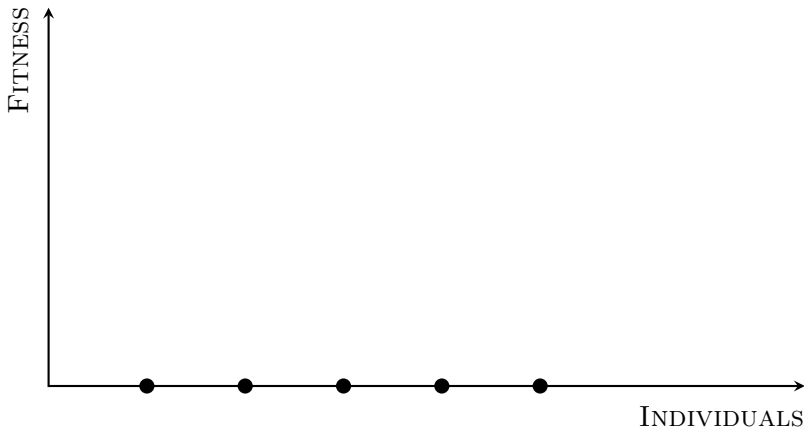
[Noisy fitness] The LUCIE selection procedure

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



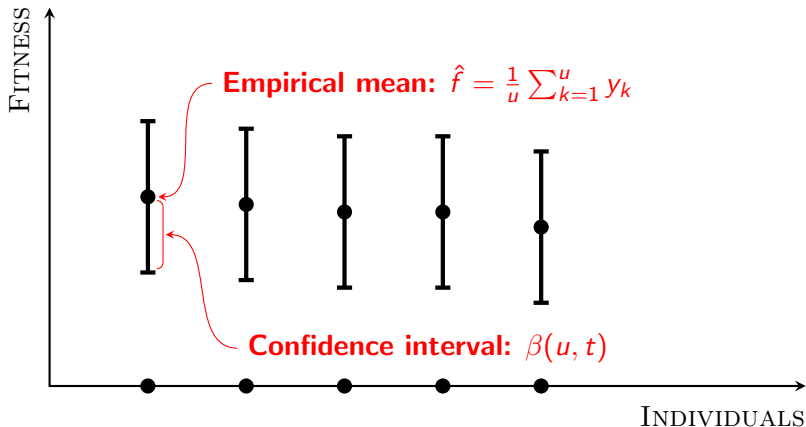
[Noisy fitness] The LUCIE selection procedure

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



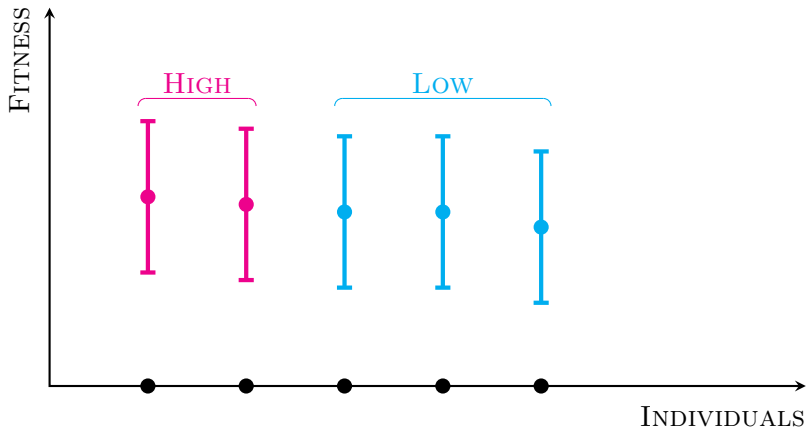
[Noisy fitness] The LUCIE selection procedure

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



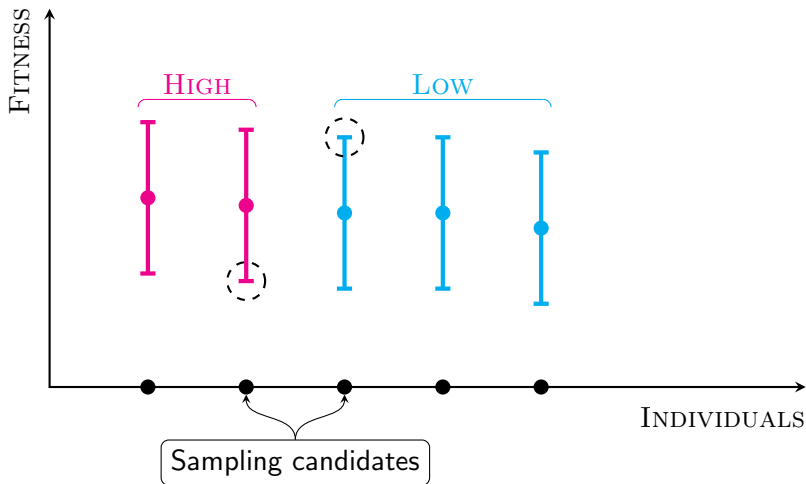
[Noisy fitness] The LUCIE selection procedure

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



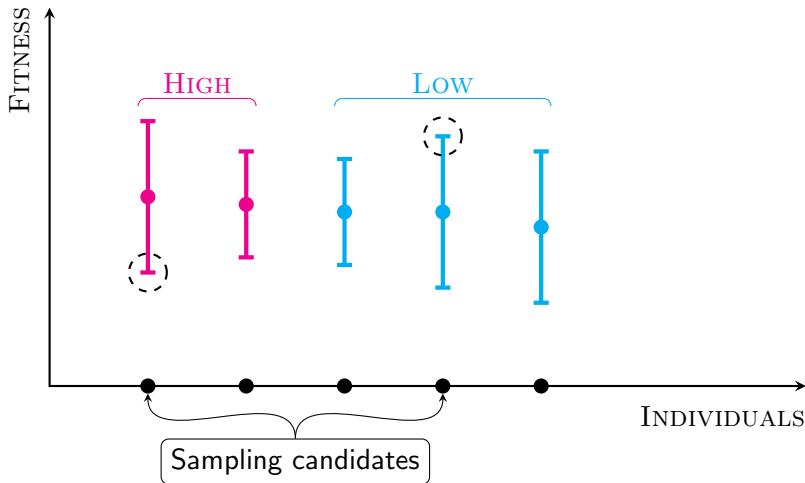
[Noisy fitness] The LUCIE selection procedure

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



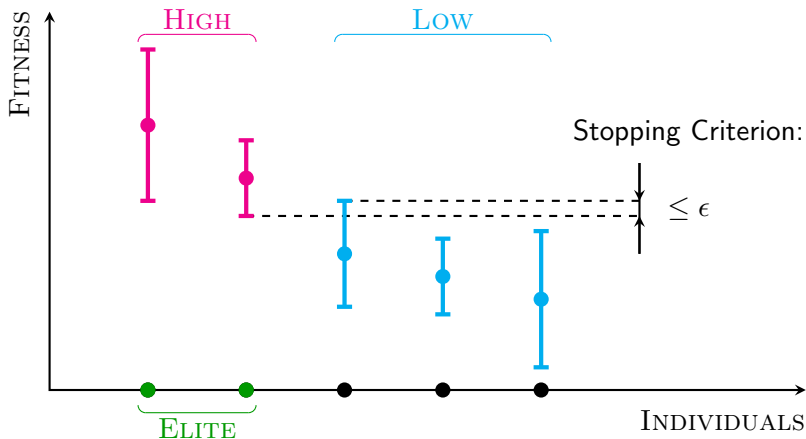
[Noisy fitness] The LUCIE selection procedure

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



[Noisy fitness] The LUCIE selection procedure

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



[Noisy fitness] Classic Control

%noise

0%

200%

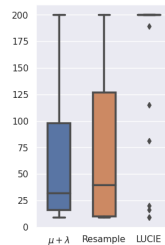
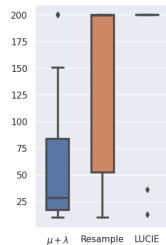
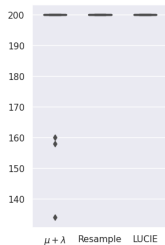
400%

600%

800%

CARTPOLE

Fitness



ACROBOT

Fitness

