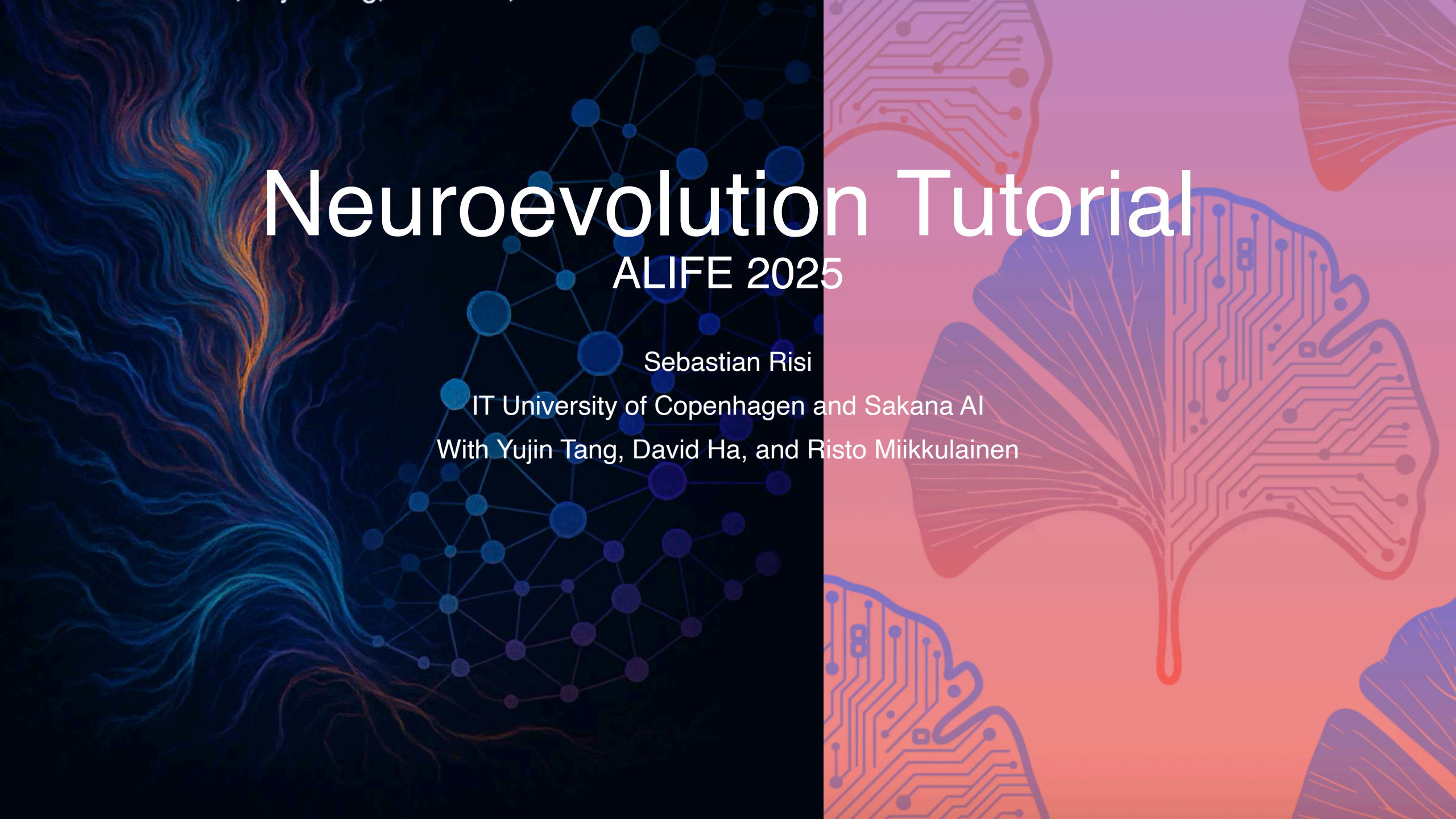


# Neuroevolution Tutorial

## ALIFE 2025

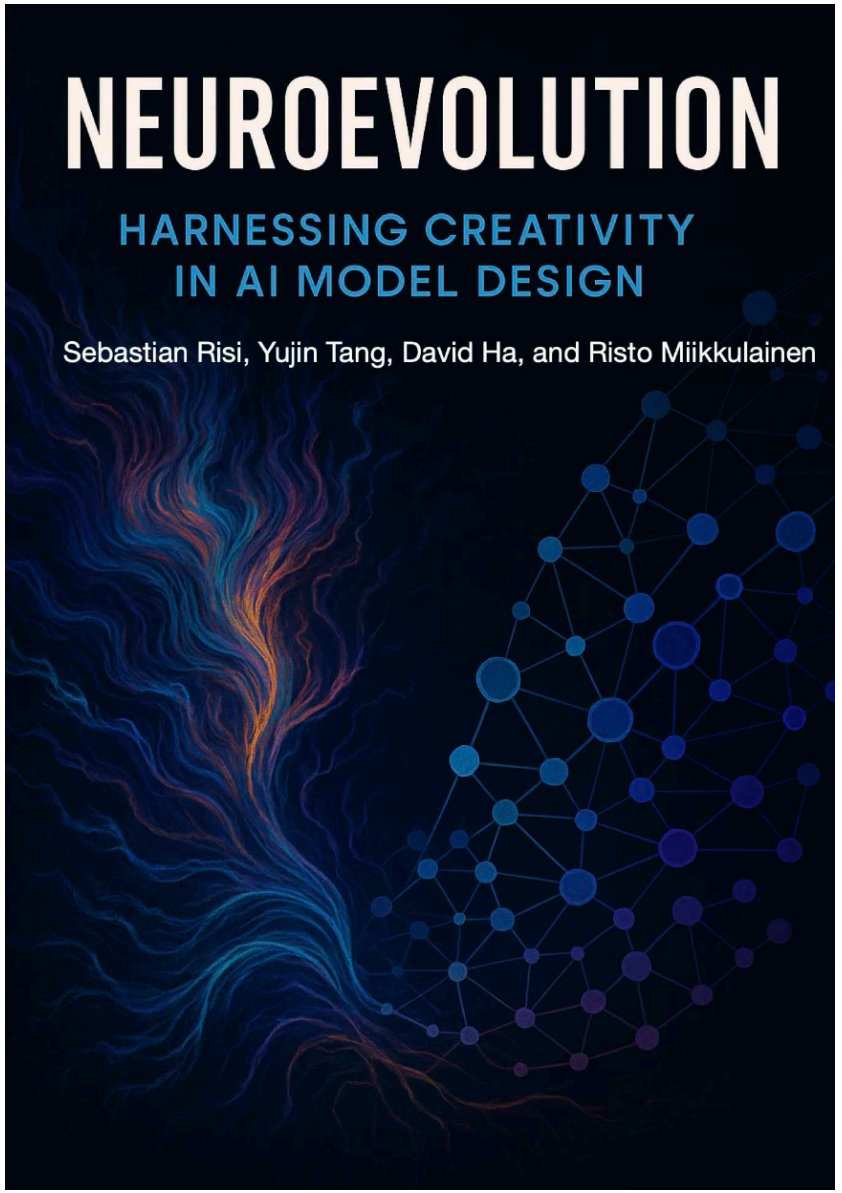
Sebastian Risi

IT University of Copenhagen and Sakana AI

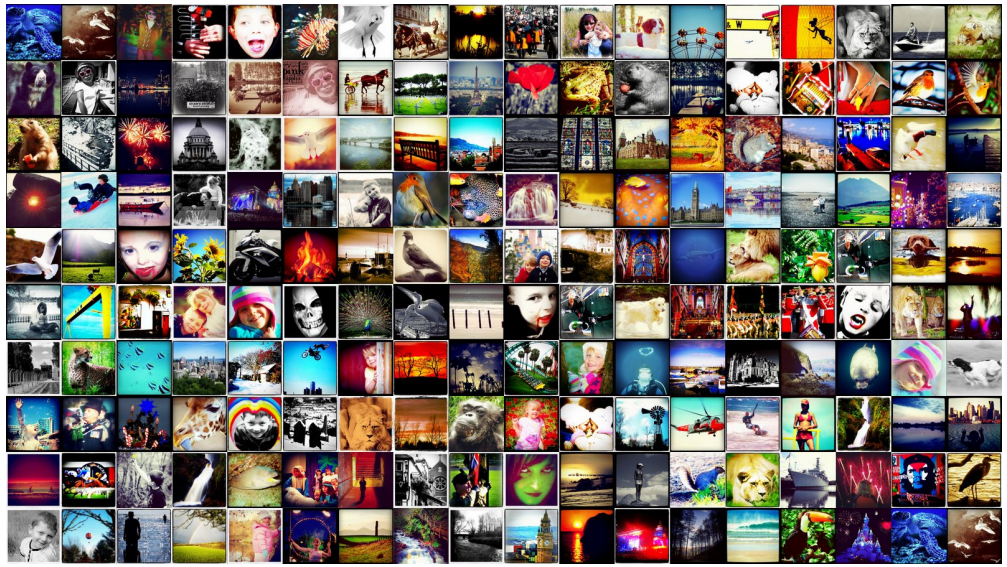
With Yujin Tang, David Ha, and Risto Miikkulainen

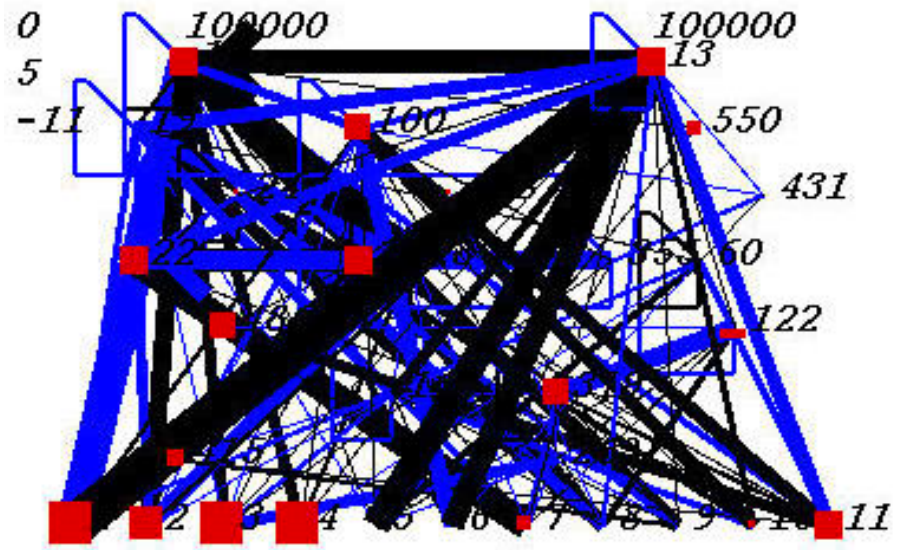
# Outline

1. Motivation for Neuroevolution
2. Basics
3. Indirect Encodings
4. Taking Advantage of Diversity
5. Evolving Intelligent Agents
6. Evolving Collective Systems
7. Open-ended Neuroevolution
8. Synergies with other ML
9. Insights into Biology
10. Conclusion



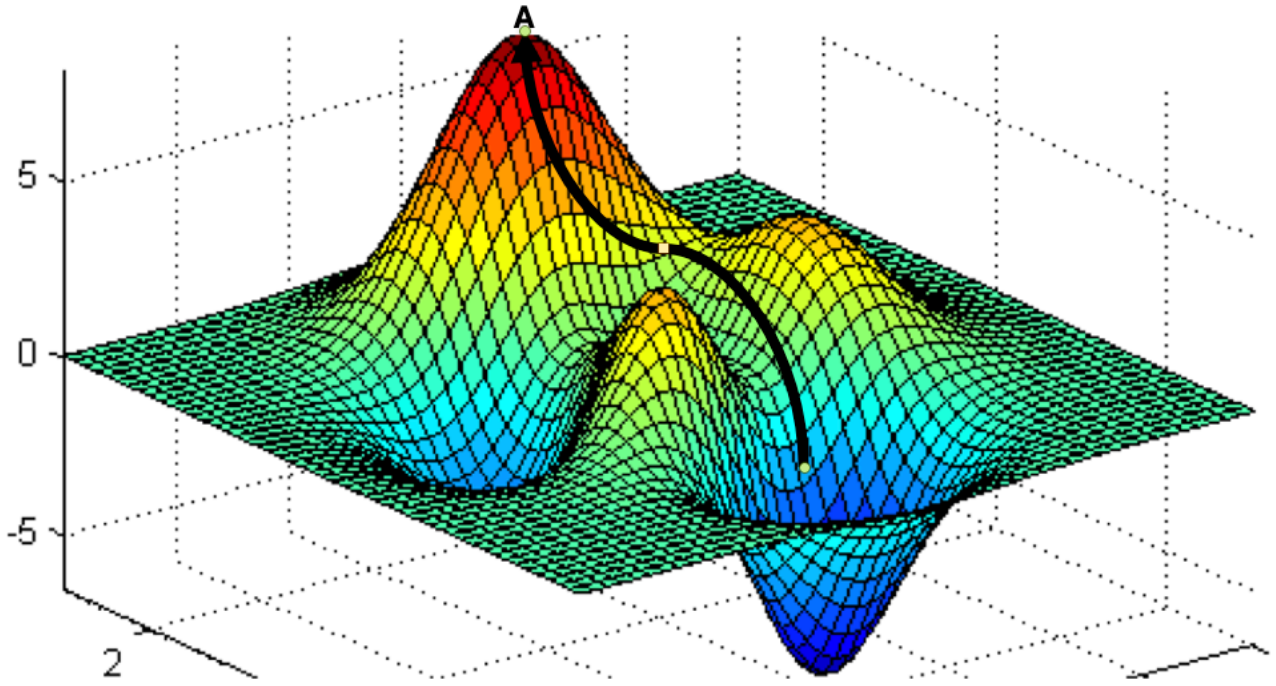
# Motivation: From Imitation to Creativity





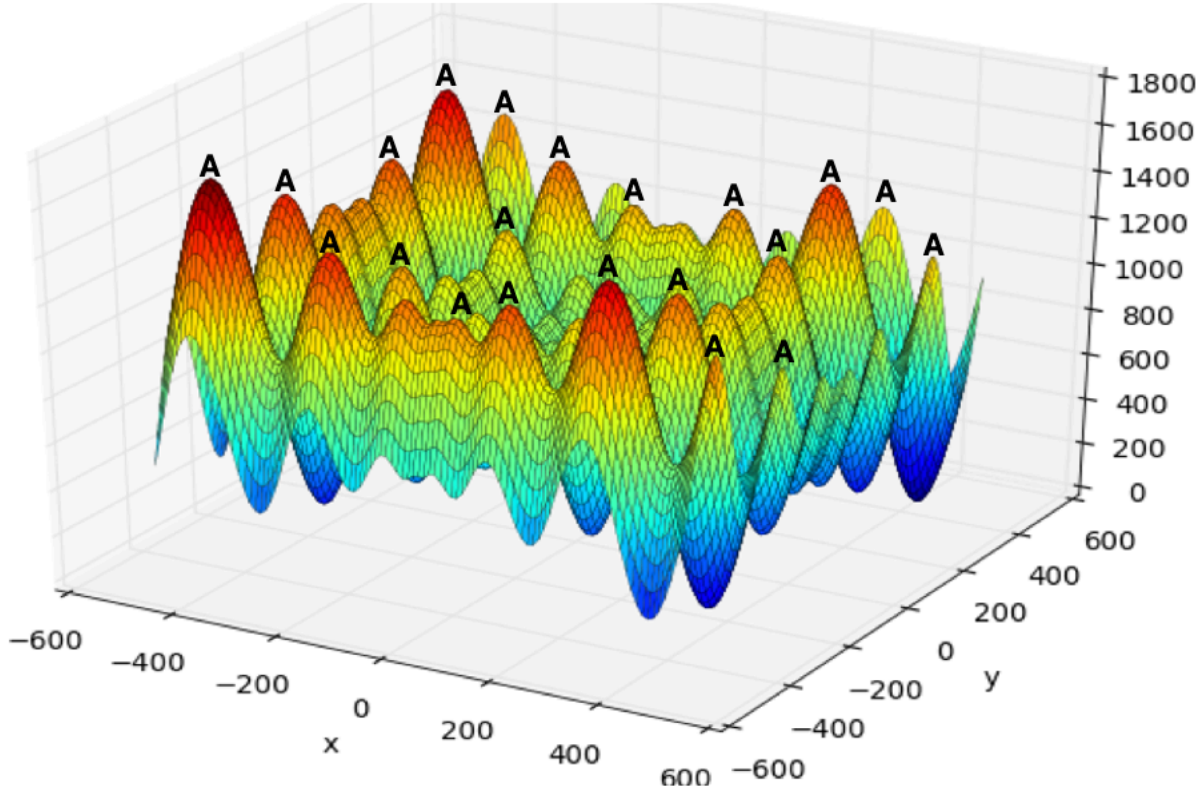

- Much of AI so far focuses on imitation
  - I.e. gradient descent on labeled datasets
  - Powerful in prediction: object recognition, diagnosis, forecasting, etc.
- Agentic AI focuses on behavior
  - Gradients not available
  - Needs to be discovered
- How can we create novel behaviors?

# Reinforcement Learning is One Approach



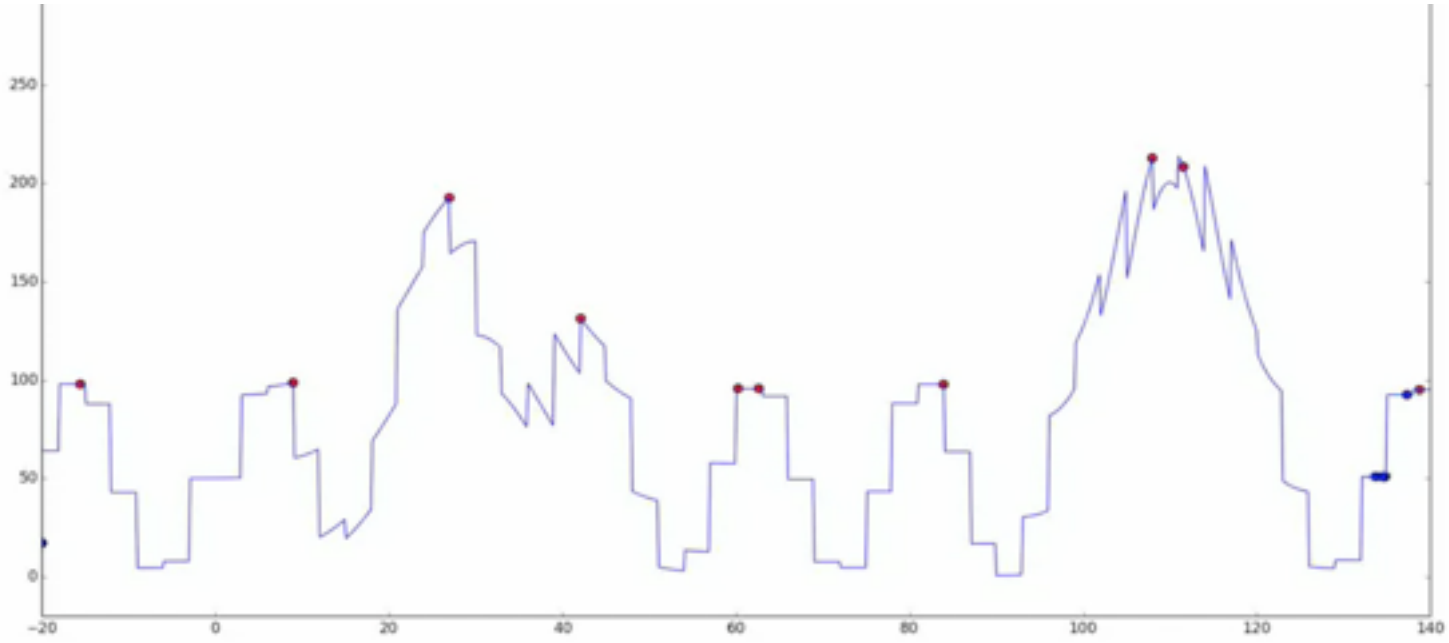

- Approximating gradient descent
- Explore around the current solution
- Improve it gradually
- Can climb the nearest hill well

# …but Creativity in RL is Limited




- Space is too large
  - Multiple starts won't help
- Space is too high-dimensional
  - Little improvement from one step
- Space is deceptive
  - Can only find the nearest hill

# Solution: Population-based Search



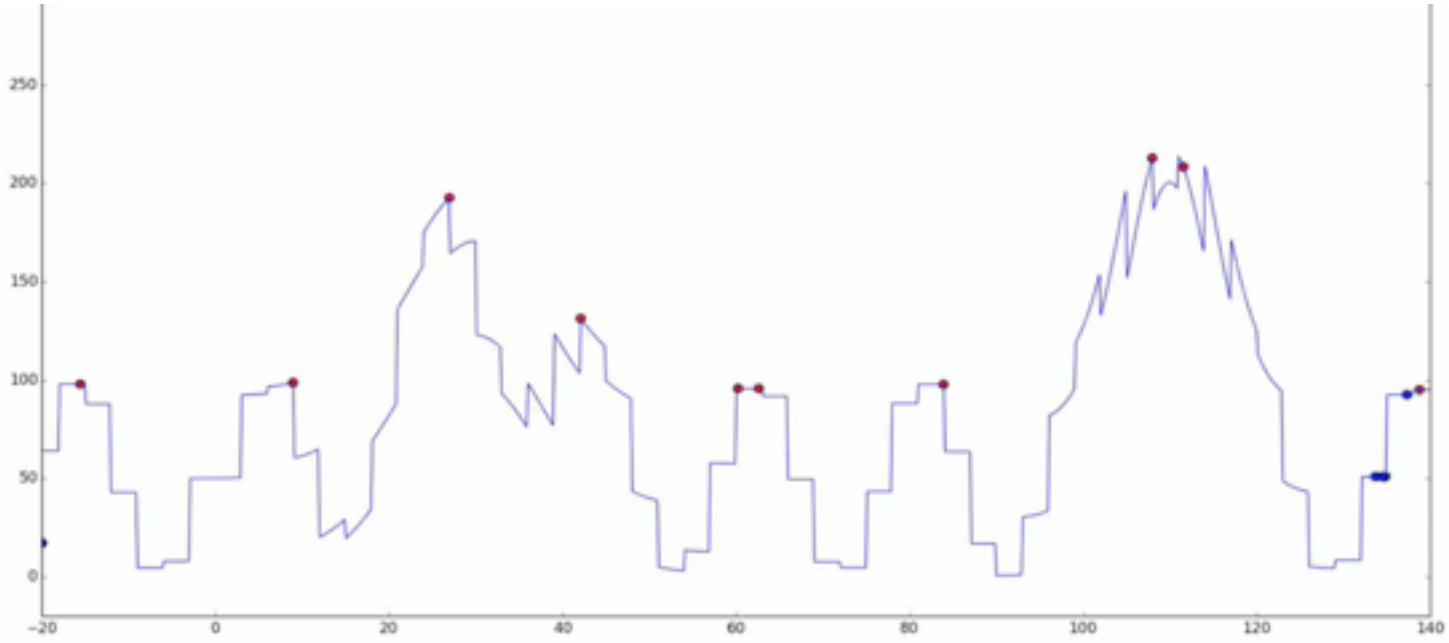

A.k.a Evolutionary Computation

- Many individuals spread out, sharing information
- Not limited to differentiable domains: configurations, choices ok

Not limited to incremental improvement

- Large jumps possible, can be more creative

# Scaling up through Evolution



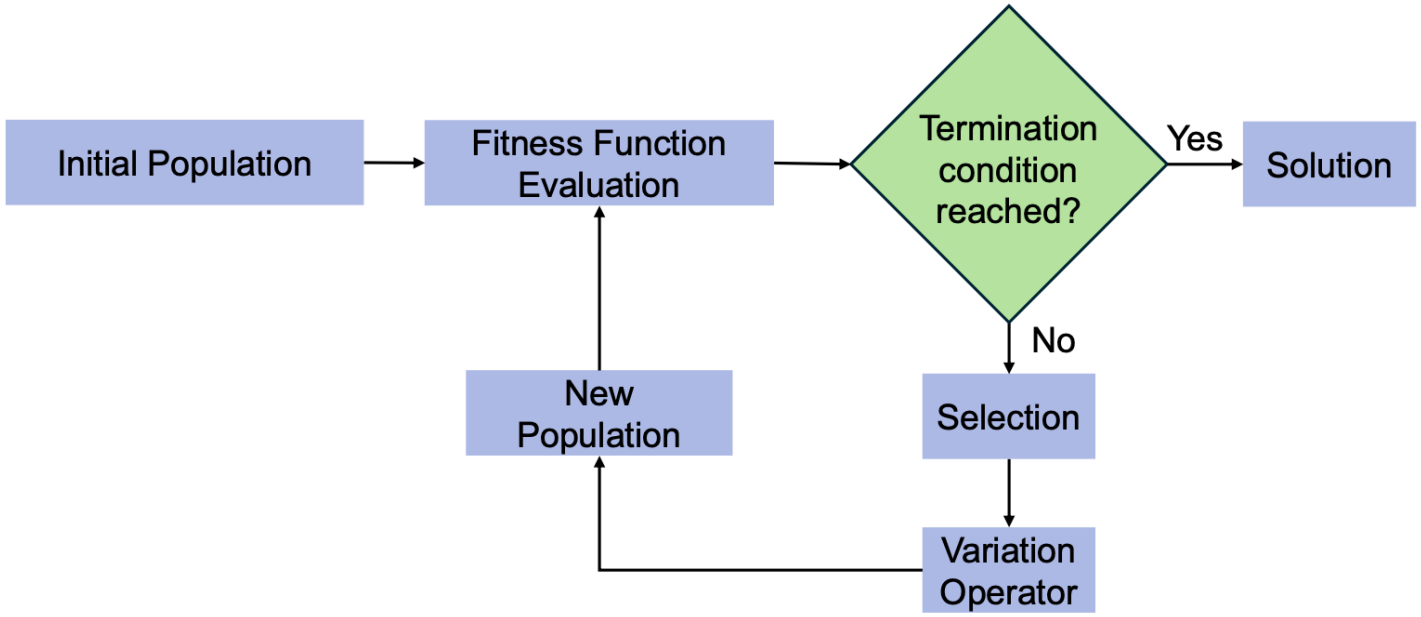

## Works in large scales

- Structured search works in large spaces (e.g. 2^2^70; Hodjat & Shahrzad 2016)
- Multiple variables optimized at once (e.g. up to 1B; Deb et al. 2017)
- Multiple objectives and novelty get around deception (Shahrzad and Hodjat 2020)

## Neuroevolution uses population-based search to optimize neural netwoks

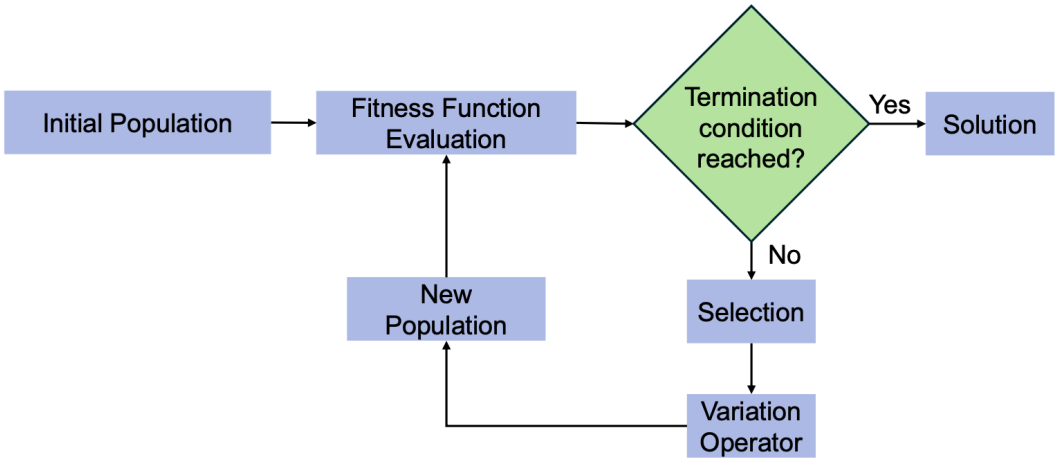
- Weights, topologies, designs

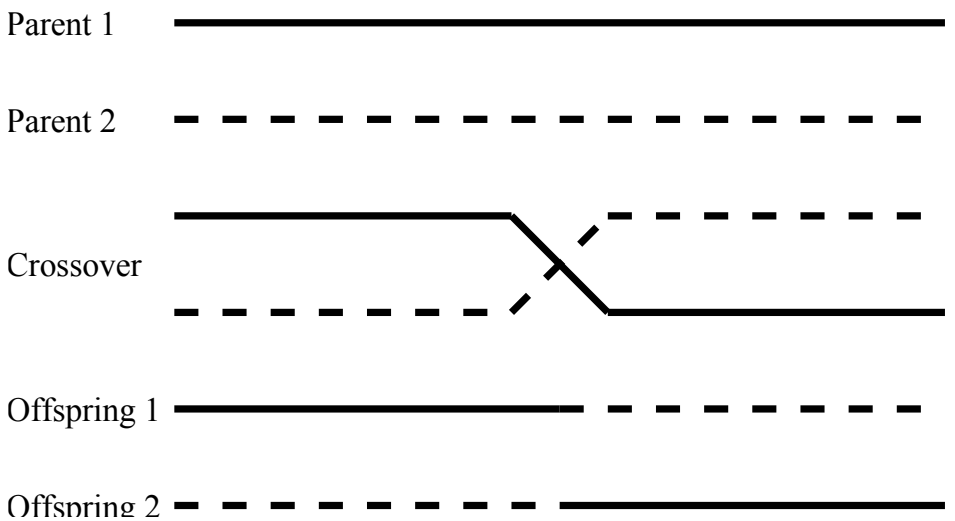
# Evolution Basics: Encoding, Evaluation, and Selection




- A population of encodings
- Decoded into individuals that are evaluated in the domain
- Good individuals retained, bad thrown away

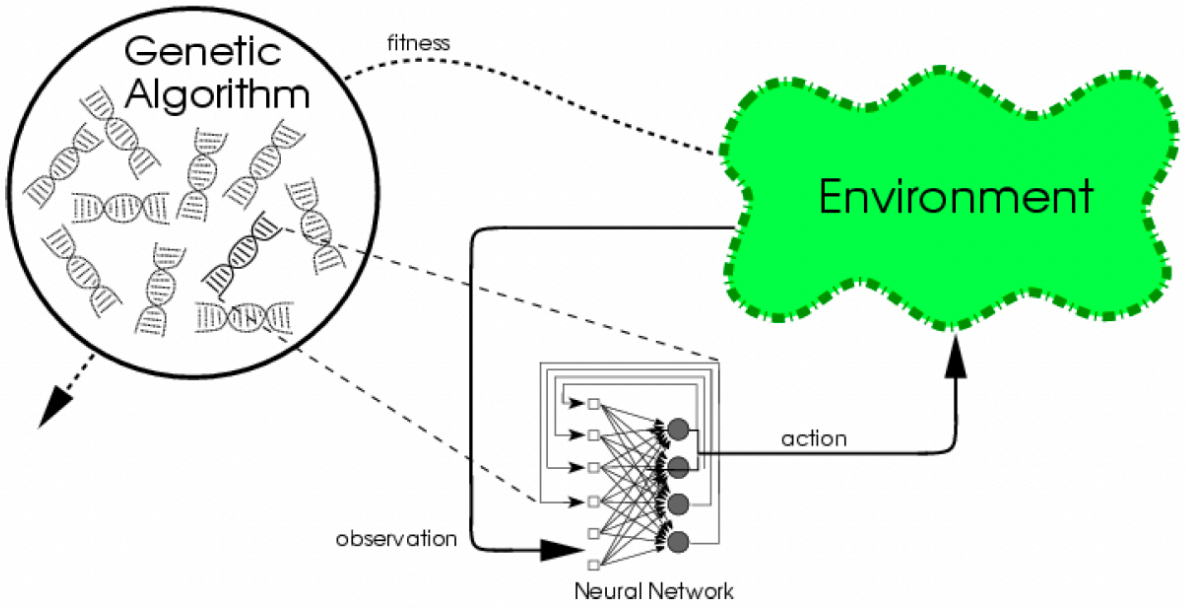
# Creating Variation



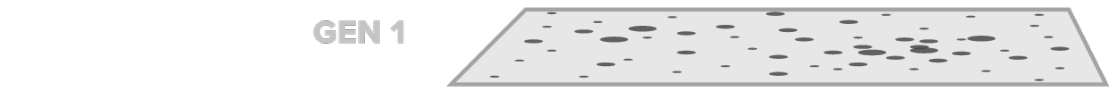

- New individuals generated from the parent encodings
  - Crossover: combine building blocks from two parents
  - Mutation: create new building blocks

# Basic Neuroevolution

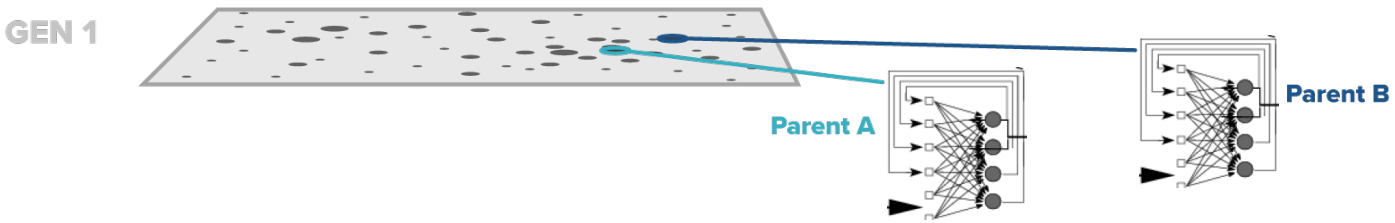



- Evolving connection weights in a population of networks
- Chromosomes are strings of connection weights (bits or real)
  - E.g. 10010110101100101111001
  - Usually fully connected, fixed, initially random topology
- A natural mapping between genotype and phenotype
  - GA and NN are a good match!

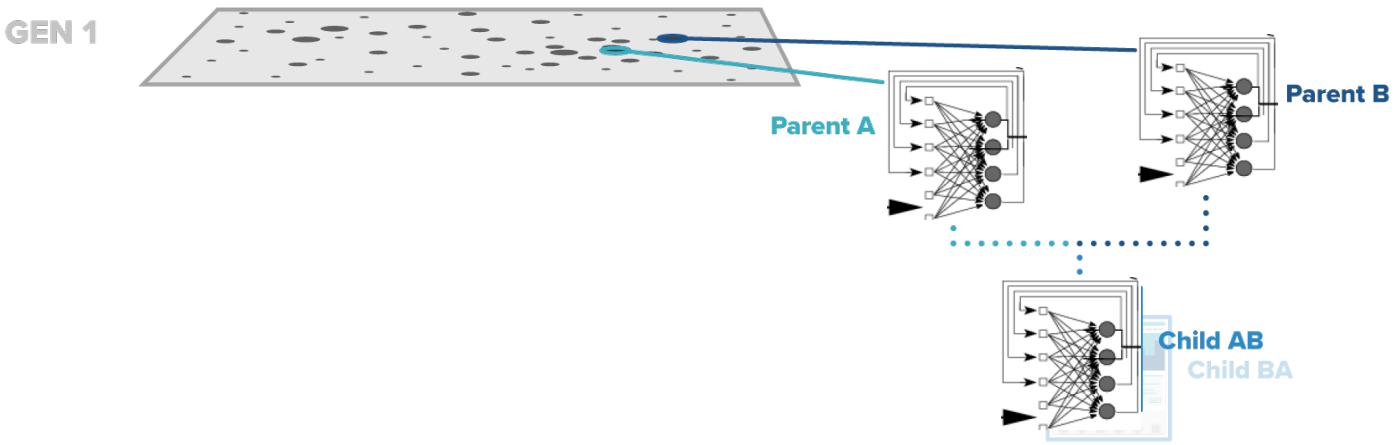
# Population-based Search



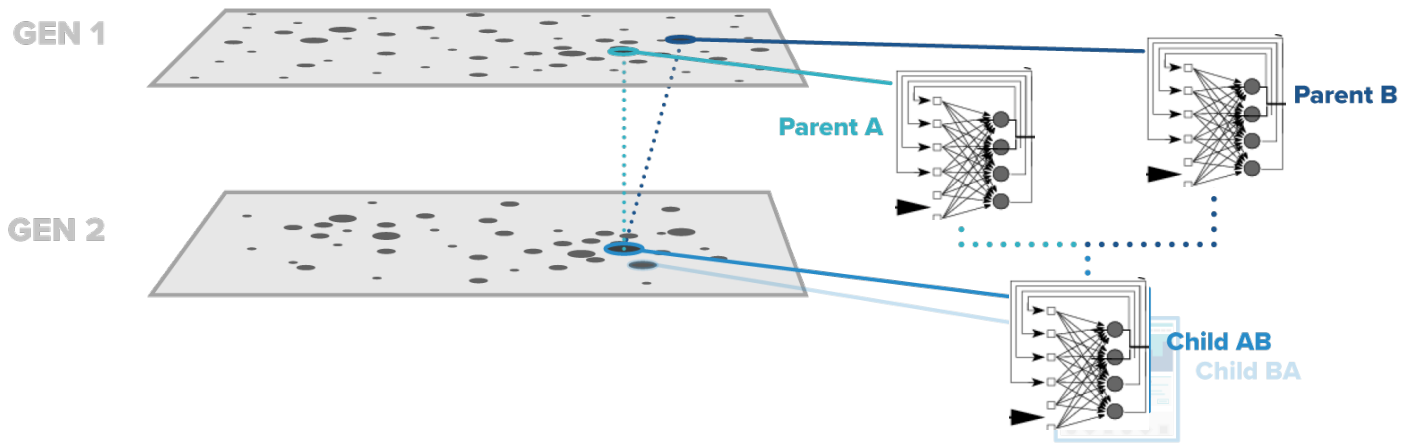
# Population-based Search



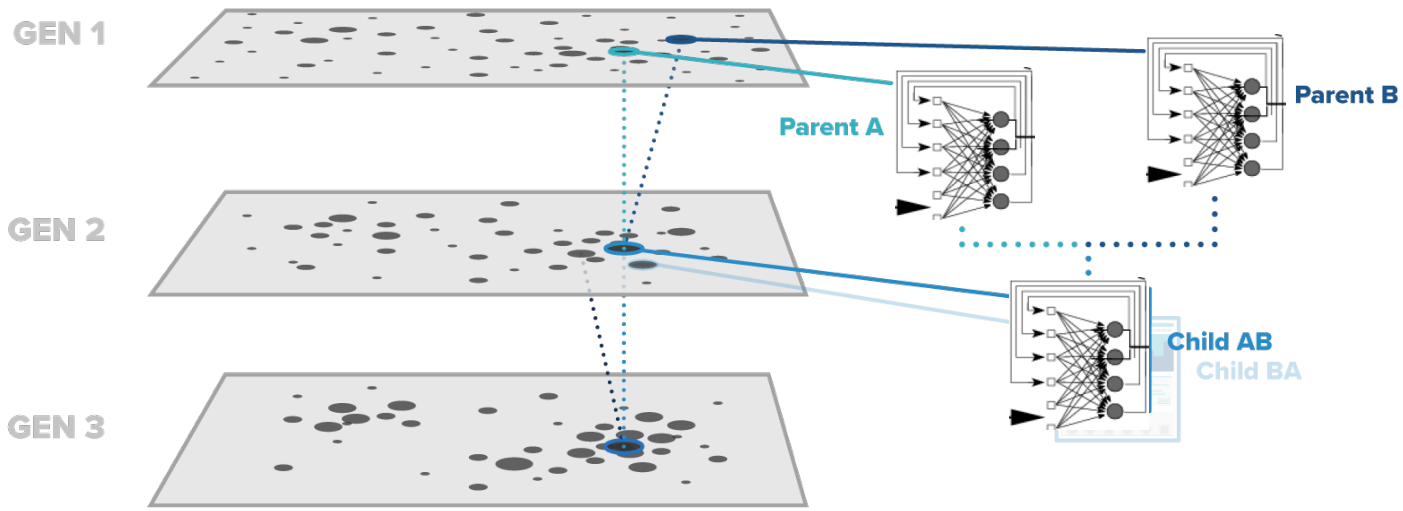
# Population-based Search



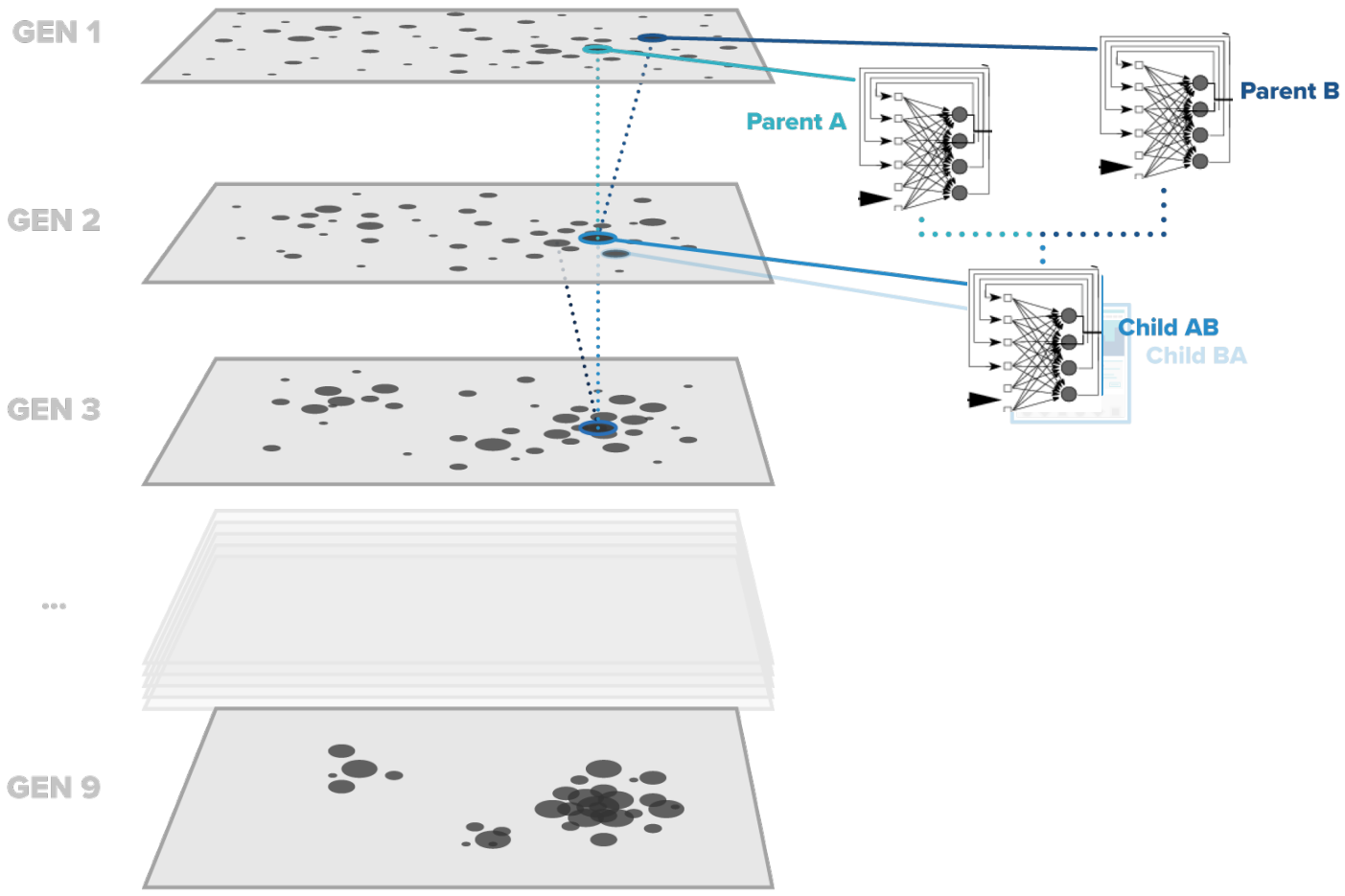
# Population-based Search



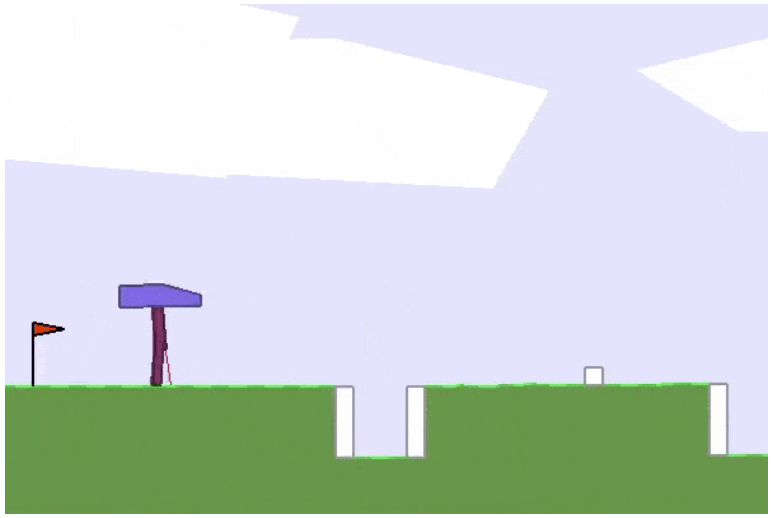
# Population-based Search

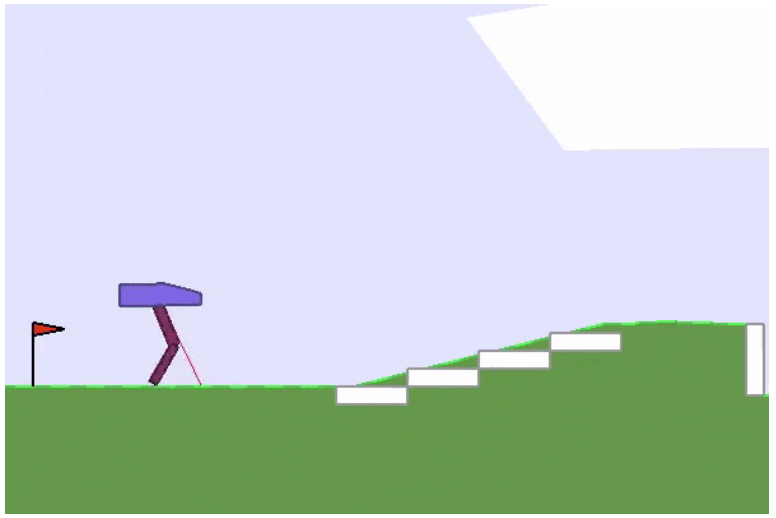


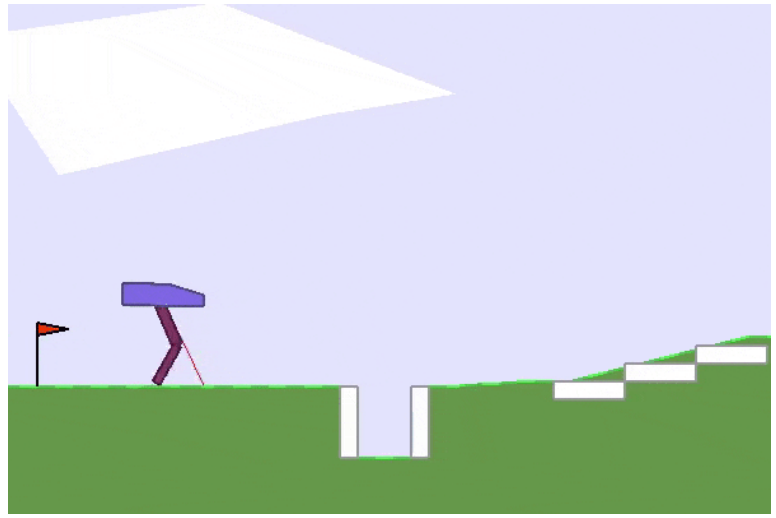
# Population-based Search



# Example: Learning to Walk



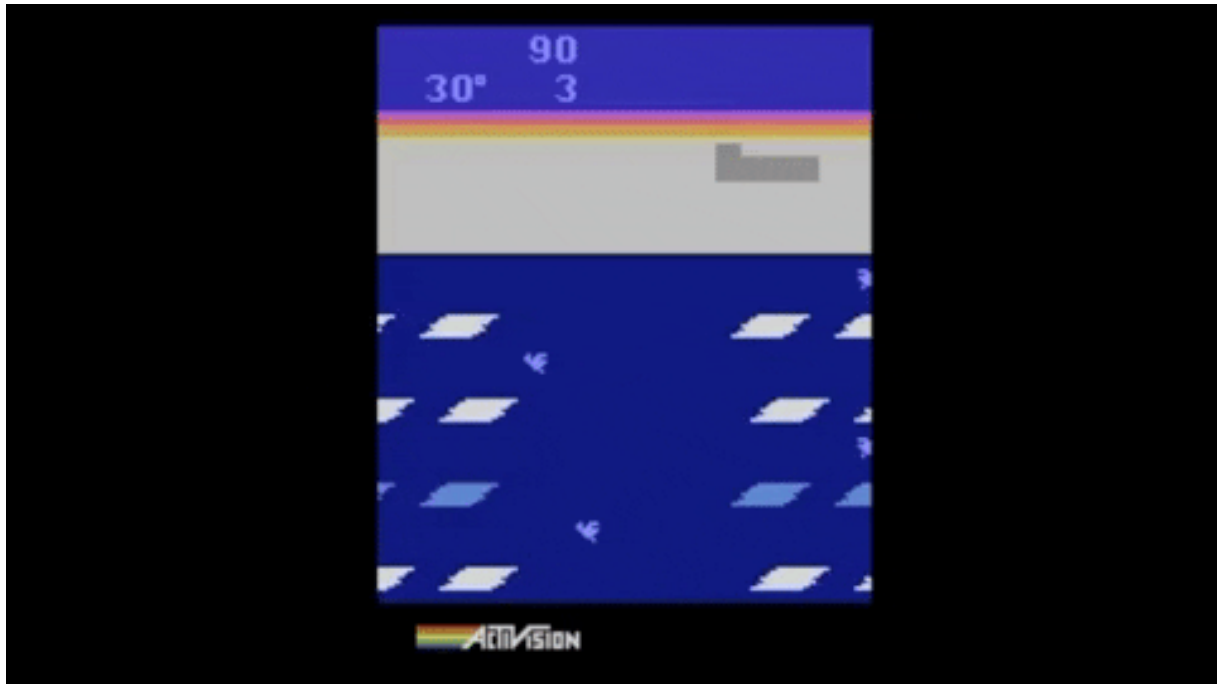




# NE vs RL

- Salimans et al. ”Evolution Strategies as a Scalable Alternative to Reinforcement Learning”, 2017.
- Such et al. ”Deep Neuroevolution: Genetic Algorithms Are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning”, 2018.

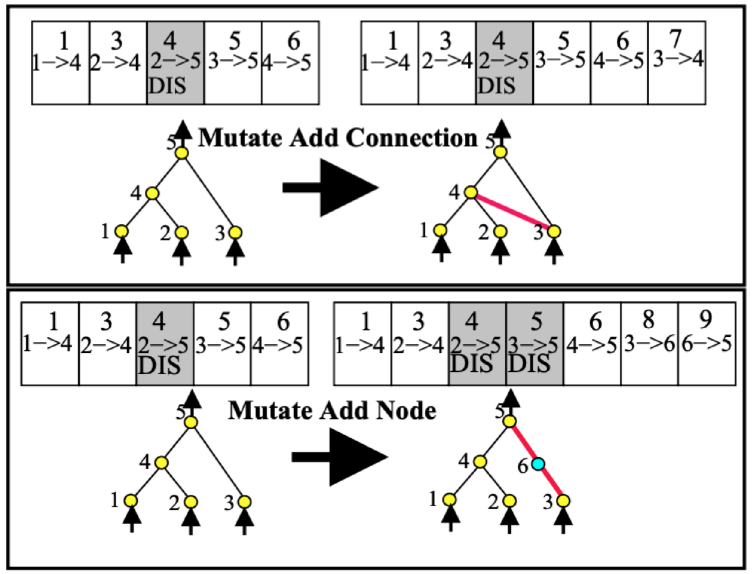
| | DQN | ES | A3C | RS | GA | GA |
|---|---|---|---|---|---|---|
| Frames | 200M | 1B | 1B | 1B | 1B | 6B |
| Time | ~7-10d | ~ 1h | ~ 4d | ~ 1h or 4h | ~ 1h or 4h | ~ 6h or 24h |
| Forward Passes | 450M | 250M | 250M | 250M | 250M | 1.5B |
| Backward Passes | 400M | 0 | 250M | 0 | 0 | 0 |
| Operations | 1.25B U | 250M U | 1B U | 250M U | 250M U | 1.5B U |
| amidar | **978** | 112 | 264 | 143 | 263 | 377 |
| assault | 4,280 | 1,674 | **5,475** | 649 | 714 | 814 |
| asterix | 4,359 | 1,440 | **22,140** | 1,197 | 1,850 | 2,255 |
| asteroids | 1,365 | 1,562 | **4,475** | 1,307 | 1,661 | 2,700 |
| atlantis | 279,987 | **1,267,410** | 911,091 | 26,371 | 76,273 | 129,167 |
| enduro | **729** | 95 | -82 | 36 | 60 | 80 |
| frostbite | 797 | 370 | 191 | 1,164 | **4,536** | **6,220** |
| gravitar | 473 | **805** | 304 | 431 | 476 | 764 |
| kangaroo | 7,259 | **11,200** | 94 | 1,099 | 3,790 | **11,254** |
| seaquest | **5,861** | 1,390 | 2,355 | 503 | 798 | 850 |
| skiing | -13,062 | -15,443 | -10,911 | -7,679 | †**-6,502** | †**-5,541** |
| venture | 163 | 760 | 23 | 488 | **969** | †**1,422** |
| zaxxon | 5,363 | 6,380 | **24,622** | 2,538 | 6,180 | 7,864 |

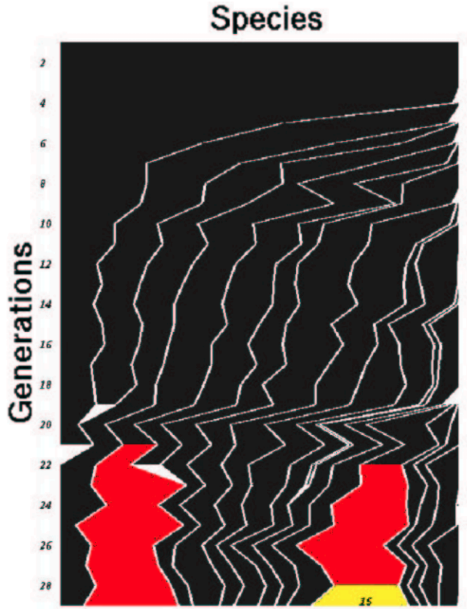


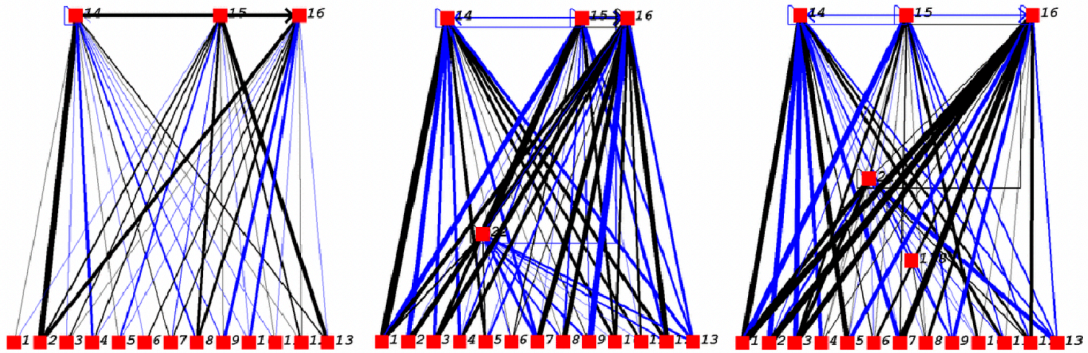
# Advanced Neuroevolution

E.g. Neuroevolution of Augmenting Topologies (NEAT)

- *Historical markings* match up different structures
- Speciation
  - Keeps incompatible networks apart
  - Protects innovation
- Incremental growth from minimal structure, i.e. *complexification*
  - Avoids searching in unnecessarily high-d space
  - Makes finding high-d solutions possible

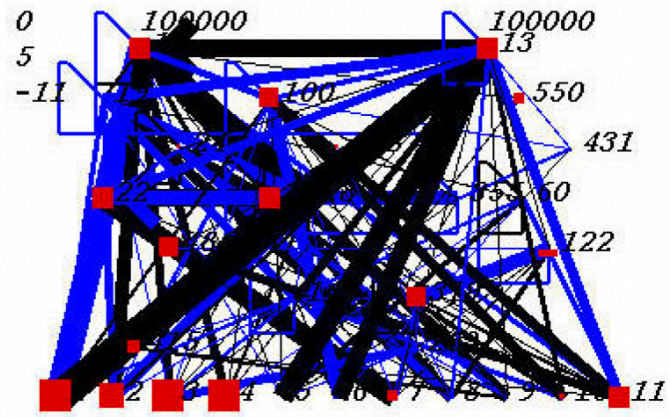


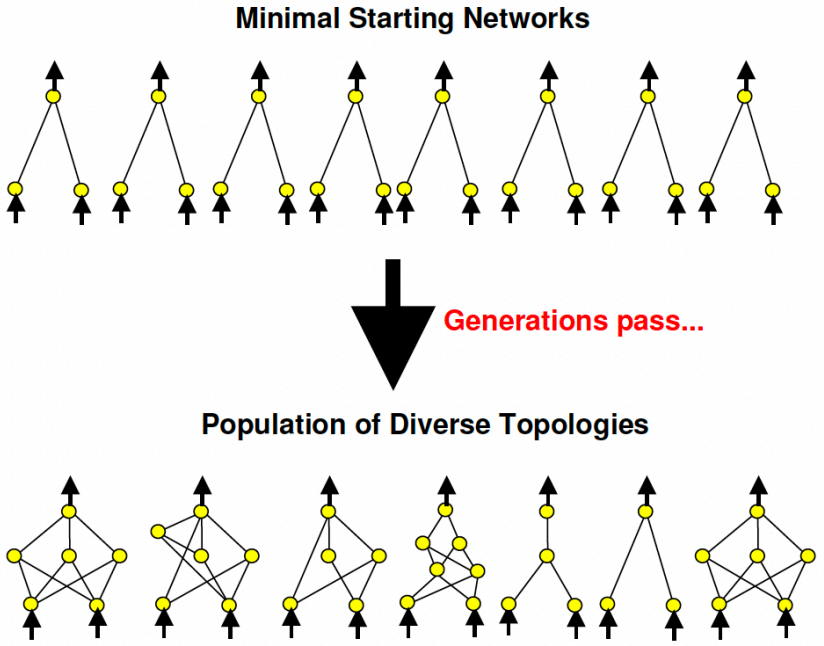



(Stanley and Miikkulainen, 2002)

# Why Is It a Good Idea?

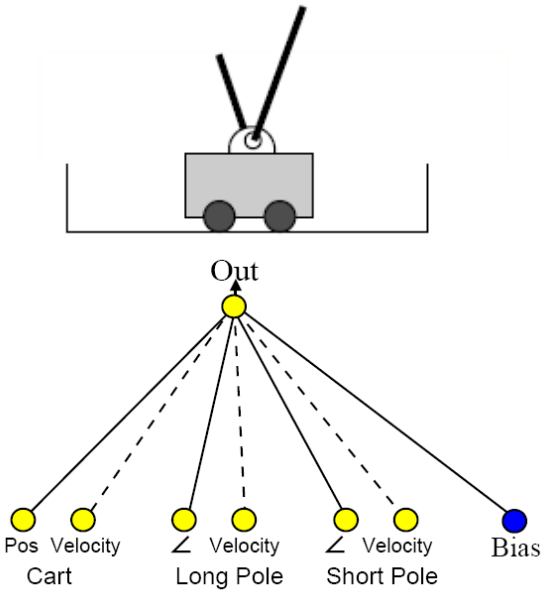


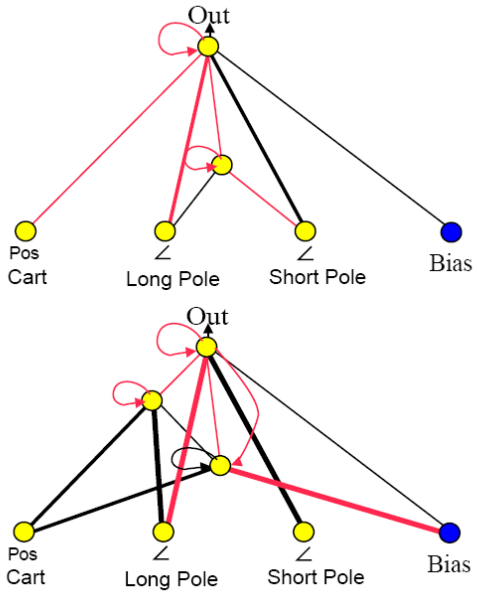




- NN search space is complex with nonlinear interactions
- Complexification keeps the search tractable
  - Start simple, add more sophistication
- Incremental discovery of complex solutions

# Discovering Compact, Interpretable Structure

- E.g. in double pole balancing
  - Easy when position, velocity of  both poles and the cart are given
  - Hard when only positions: need to figure out how they are moving
- Discovers recurrent structure
  - Either representing velocities separately
  - Or simply the derivative of the difference of the poles!
- Big improvement from other approaches
  - Standard value-function RL unsuccessful



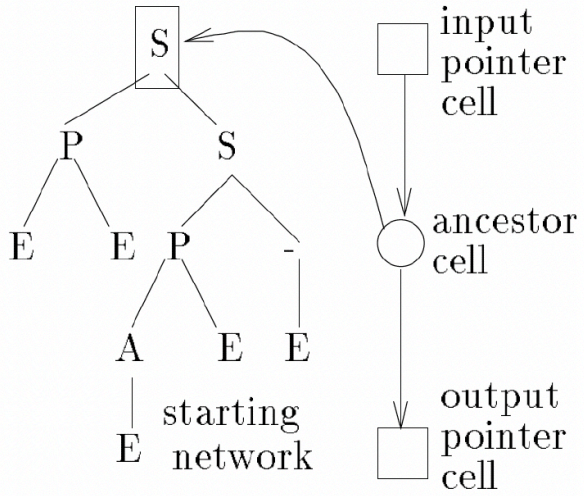



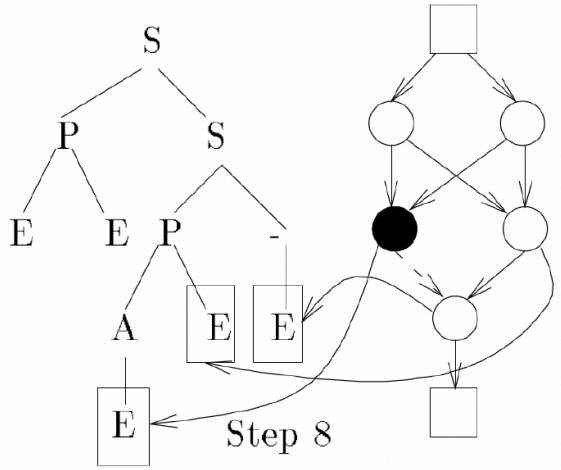

| Method | Evaluations |
|---|---|
| Gruau et al. 1996 Cellular Encoding | 840,000 |
| Gomez and Miikkulainen 1999 ESP | 169,466 |
| NEAT | 33,184 |

20 trials

# Indirect Encodings

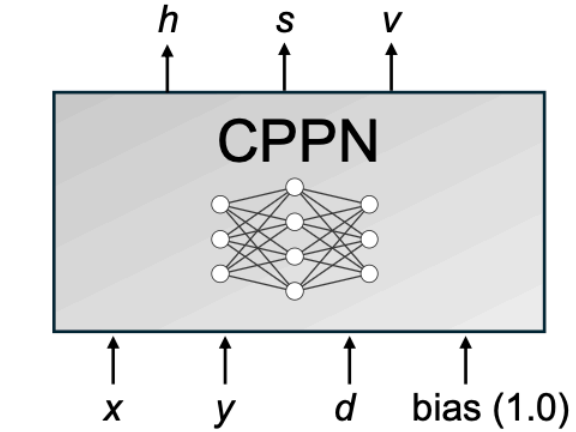
# Indirect Encoding: Development



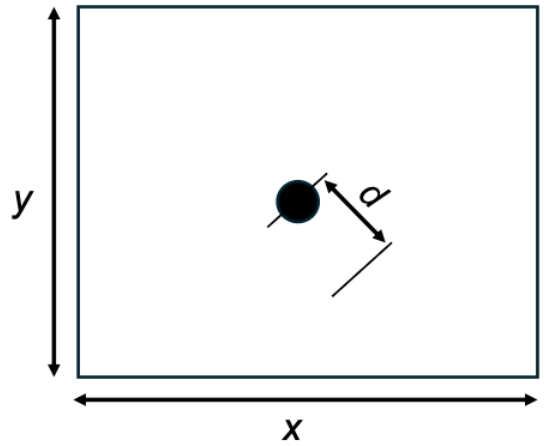




- Instructions for constructing the network evolved
  - Instead of specifying each unit and connection
- E.g. Cellular Encoding (CE Gruau & Whitley 1993)
- Grammar tree describes construction
  - Sequential and parallel cell division
  - Changing thresholds, weights
  - A “developmental” process that results in a network

# Compositional Pattern Producing Networks (CPPNs; Stanley 2007)

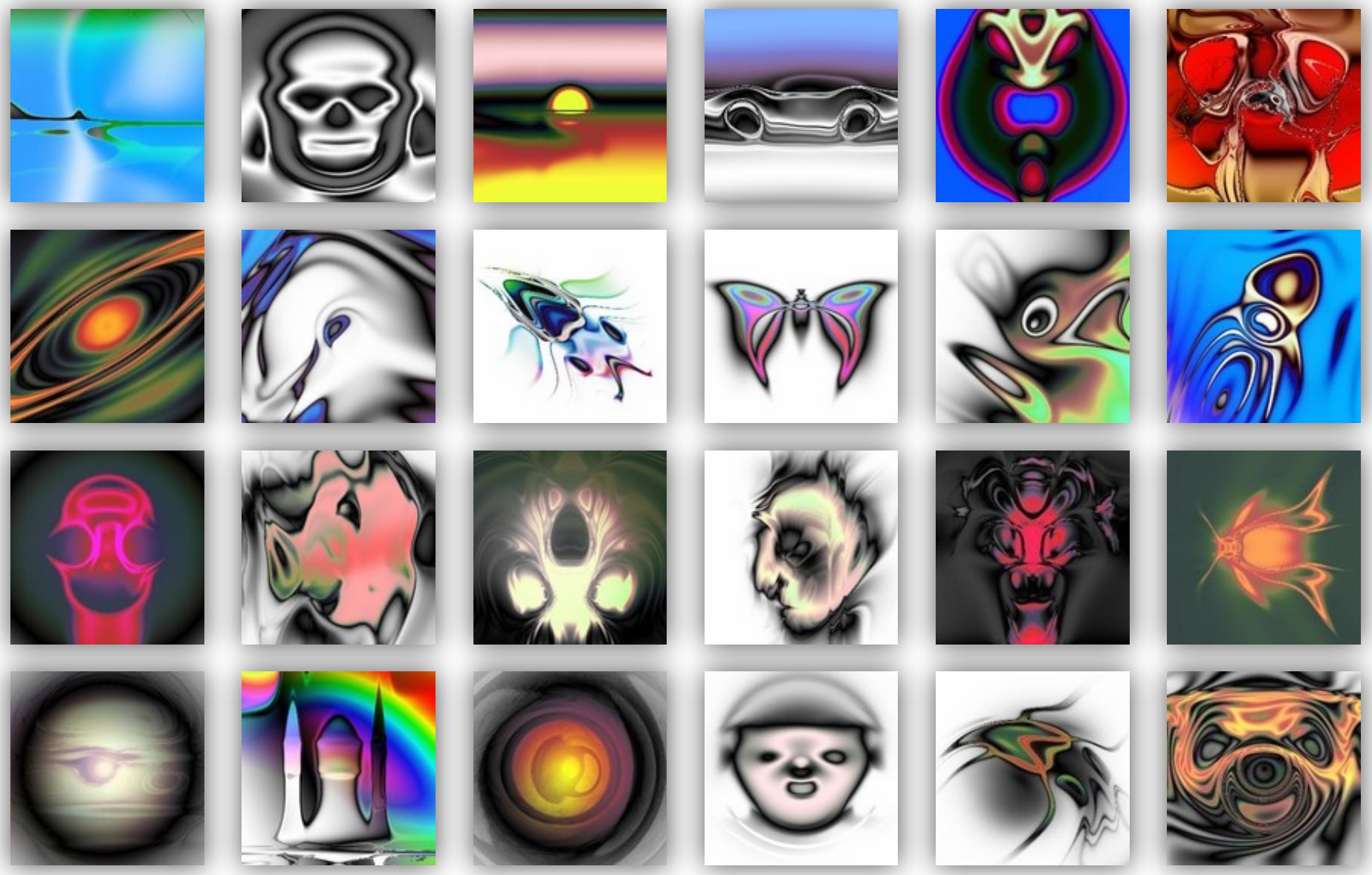



(a) CPPN

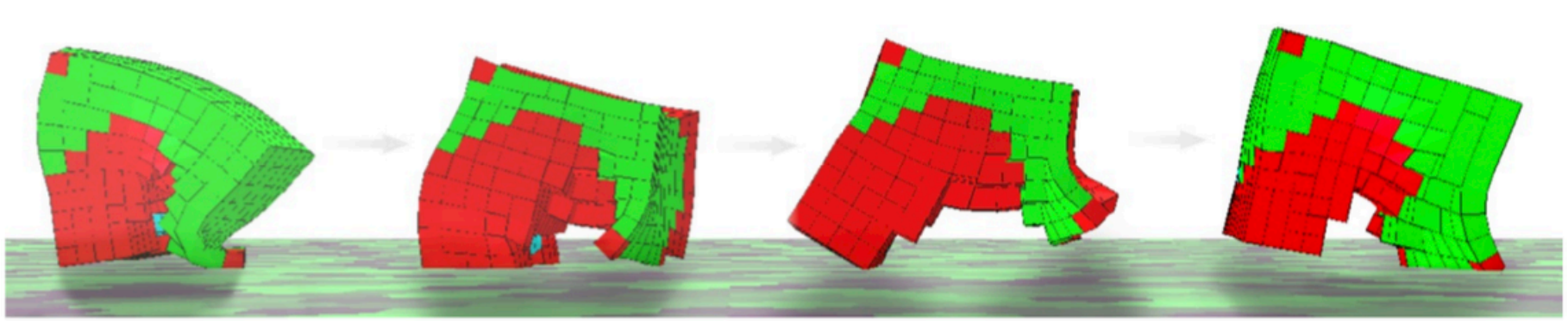


# CPPN-Generated Patterns

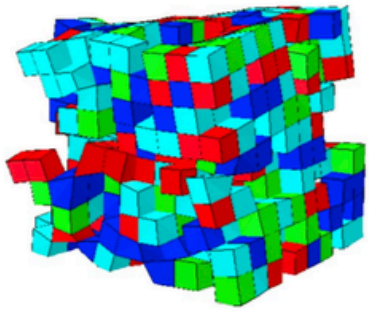
www.picbreeder.org



# From 2D Images to 3D Virtual Creatures



(a) Indirect encoding

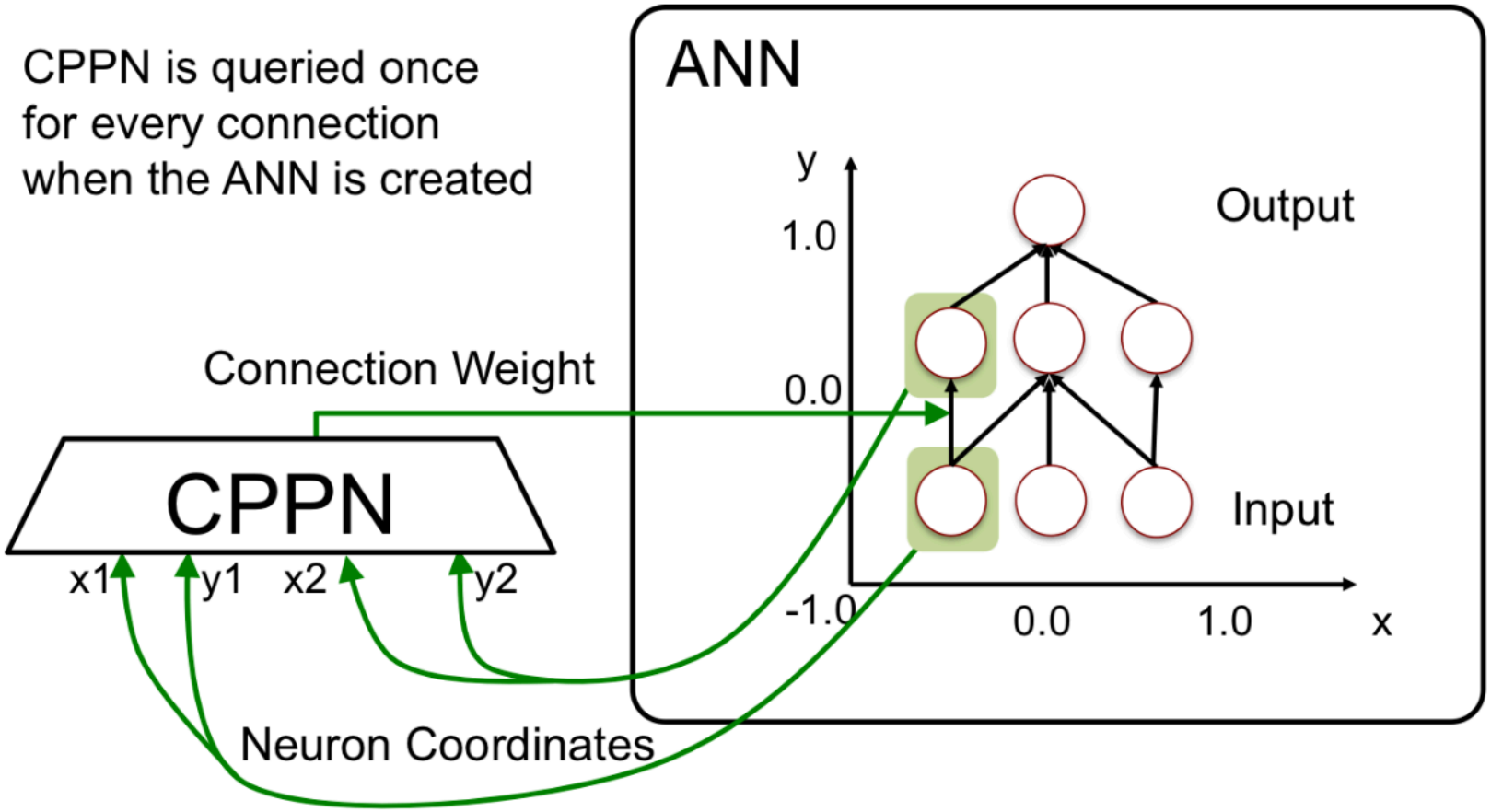


(b) Direct encoding

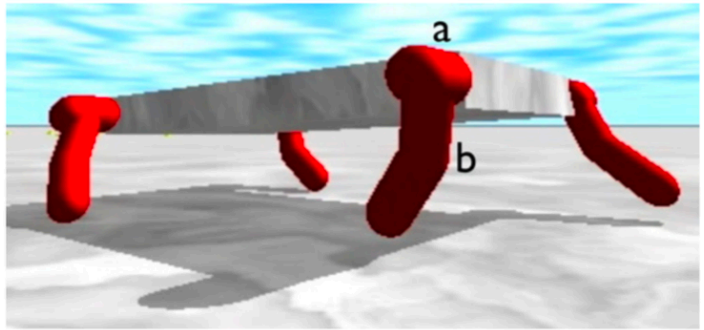
N. Cheney, R. MacCurdy, J. Clune, and H. Lipson. Unshackling evolution: evolving soft robots with multiple materials and a powerful generative en- coding. ACM SIGEVOlution, 7(1):11–23, 2

# Encoding Brains Through CPPNs: HyperNEAT

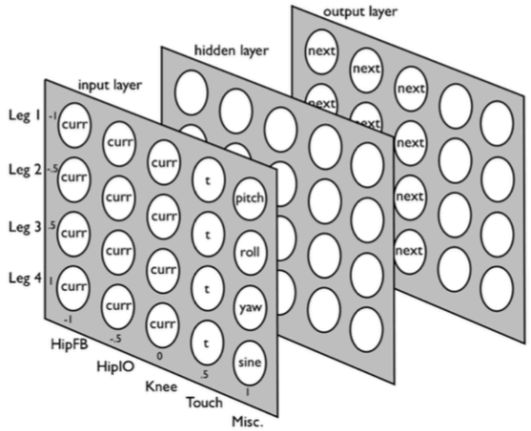
(Stanley et al. 2009)



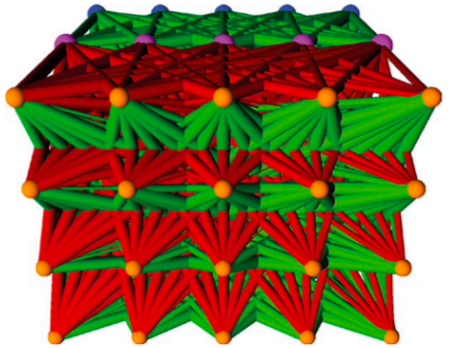
# HyperNEAT-encoded Quadruped Locomotion



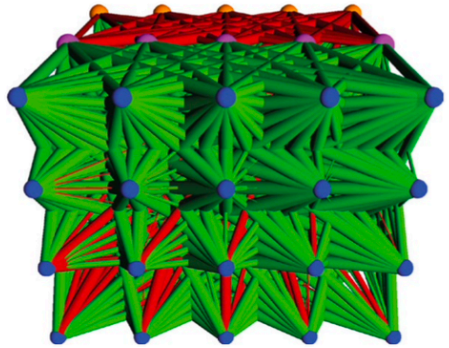

(a)




(b)



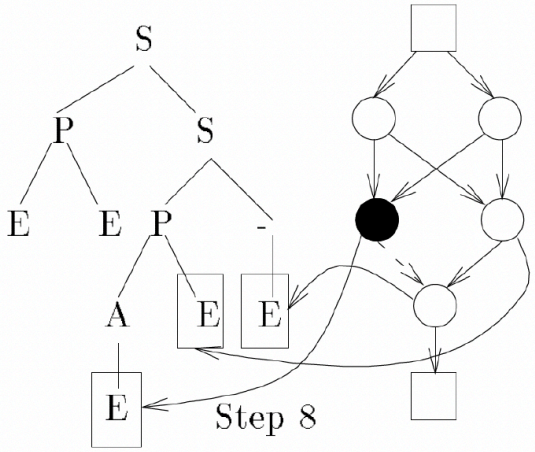
(c)

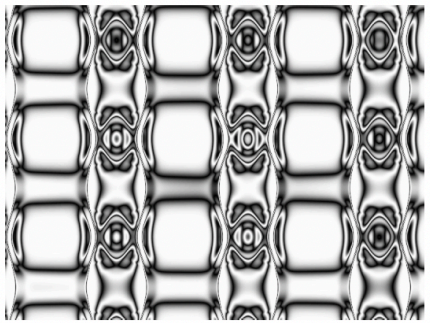


(d)

J. Clune, K. O. Stanley, R. T. Pennock, and C. Ofria. On the performance of indirect encoding across the continuum of regularity. IEEE Transactions on Evolutionary Computation, 15(3):346–3
2011

# Why are Indirect Encodings a Good Idea?

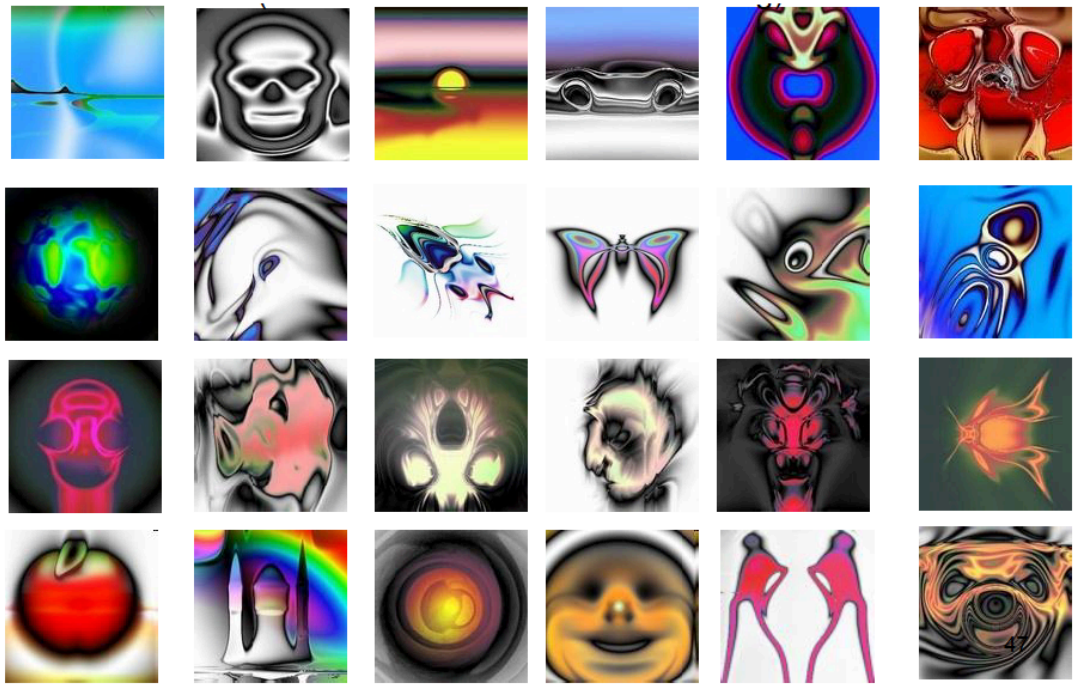


- Describes structure efficiently
  - Recurrency symbol in CE: XOR → parity
  - Repetition with variation in CPPNs
- Useful for evolving topology
  - E.g. large structured networks
  - E.g. repetition of motifs

# Taking advantage of diversity

# Diversity: (A) Searching for Novelty

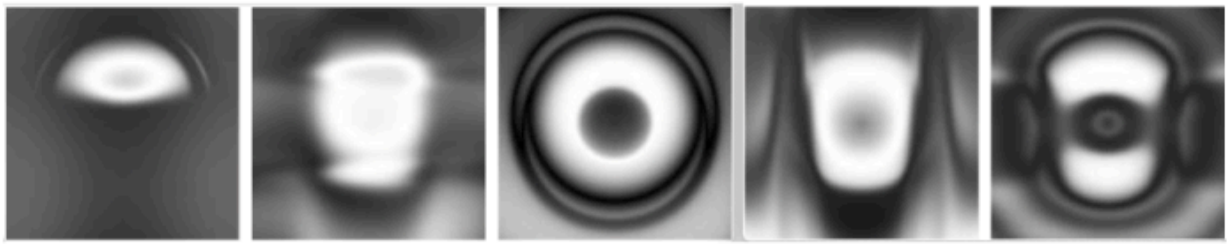


- Motivated by humans as fitness functions
- E.g. picbreeder.com, endlessforms.com (Secretan et al. 2011; Clune et all 2011)
  - CPPNs evolved; Human users select parents
- No specific goal
  - Interesting solutions preferred
  - Similar to biological evolution?

# Novelty Search (Lehman & Stanley 2011)

Gen 12 Gen 20 Gen 36 Gen49 Gen 74

(*a*) Intermediate images in the evolution of the skull image

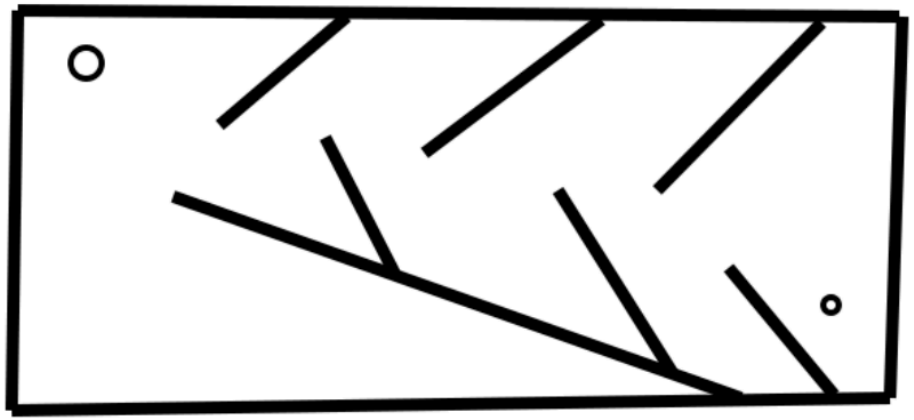


Run 1 Run 3 Run 7 Run 15 Run 17

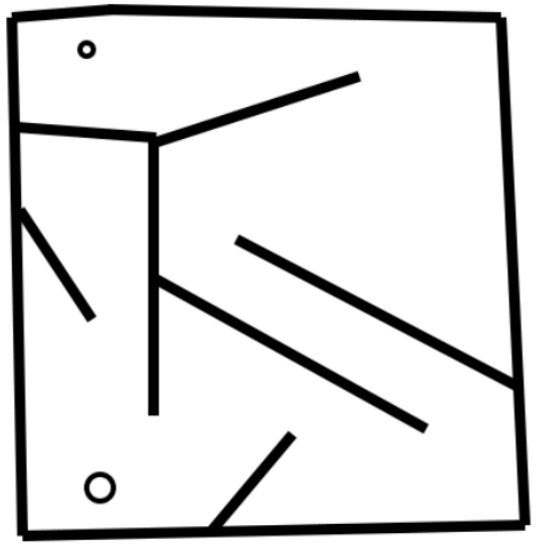
(*b*) Attempts to evolve the skull image directly

- Evolutionary algorithms maximize a performance objective
  - But sometimes hard to achieve it step-by-step
- Novelty search rewards candidates that are simply different
  - Stepping stones for constructing complexity

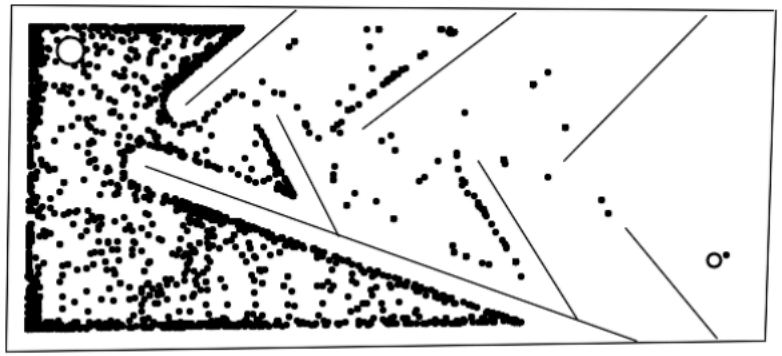
# Novelty Search



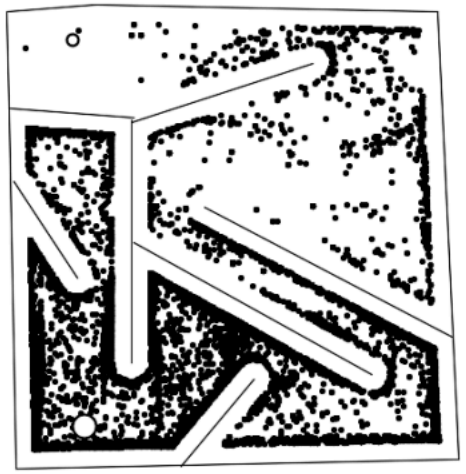
(a) Medium Map



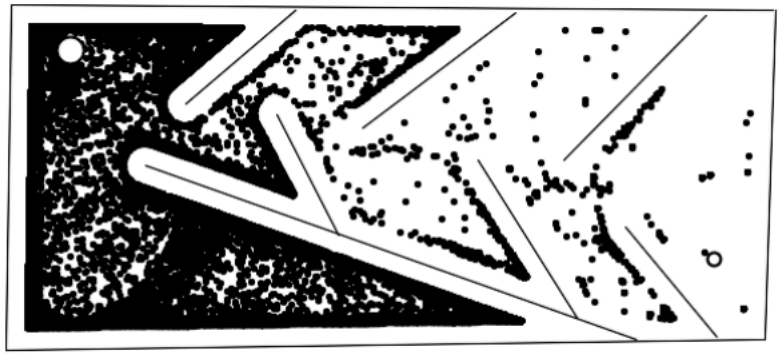
(b) Hard Map



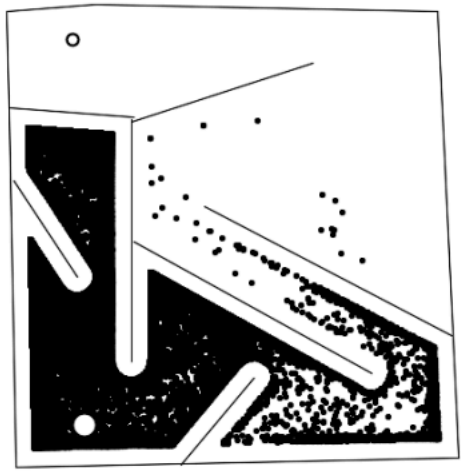
(a) Medium Map Novelty



(b) Hard Map Novelty

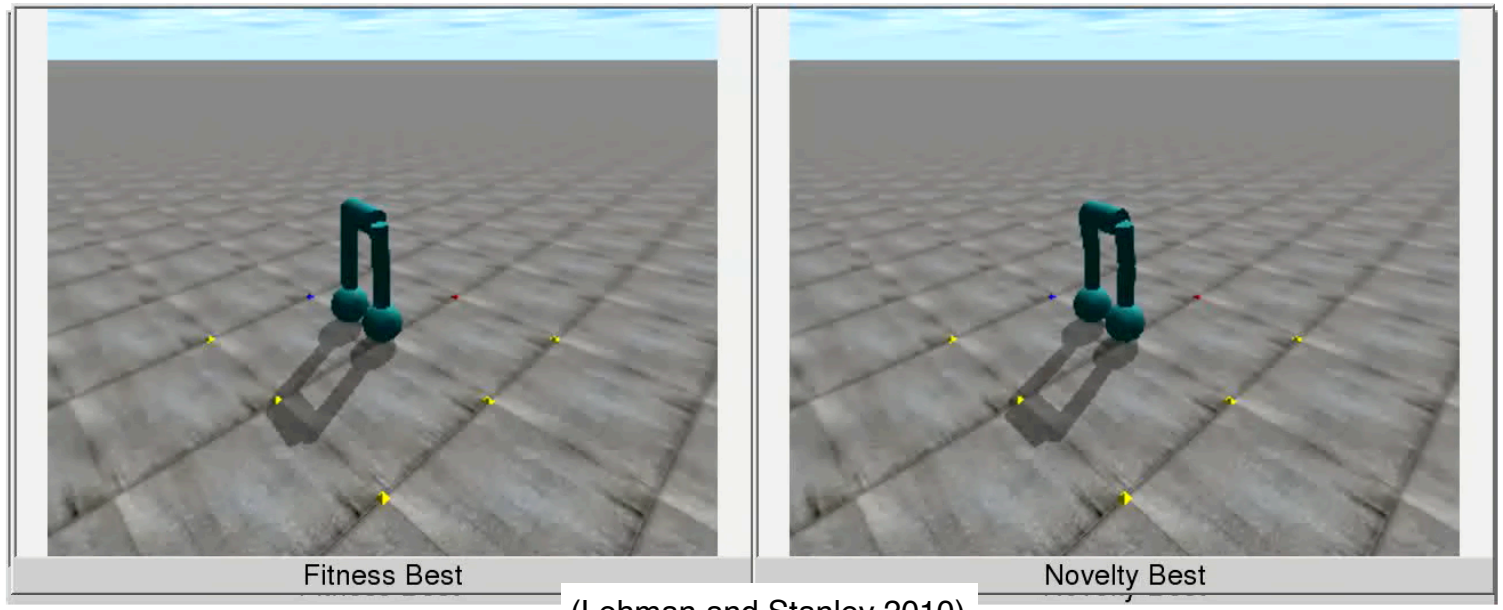


(c) Medium Map Fitness



(d) Hard Map Fitness

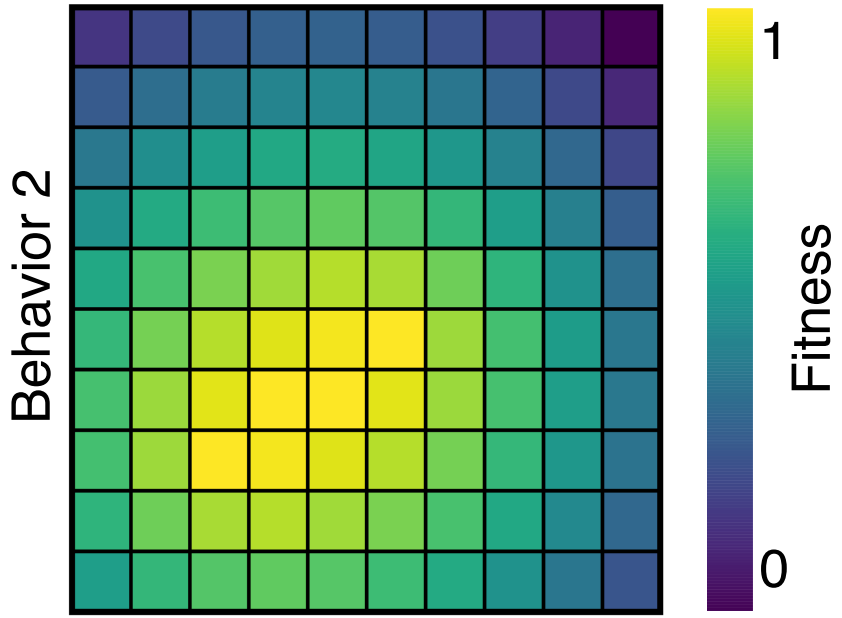
# Novelty Search Demo




(Lehman and Stanley 2010)

- Fitness-based evolution is rigid
  - Requires gradual progress
- Novelty-based evolution is more innovative, natural
  - Allows building on stepping stones
- How to guide novelty search towards useful solutions?
  - Quality Diversity methods
- DEMO

# (B) Combining Quality with Diversity



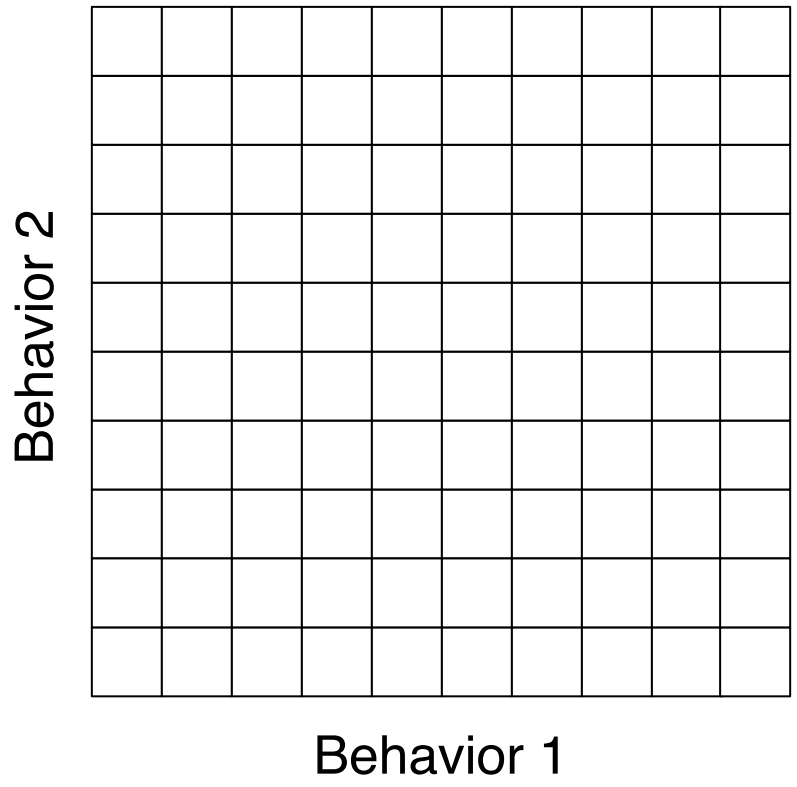

(Mouret and Clune 2015)

**Goal:** Find a set of different high-quality solutions for a given problem
Ex: Find fast walking gaits for a legged robot for every direction

**Popular QD Method:** Multi-dimensional Archive of Phenotypic Elites[1] (MAP-Elites)

**Idea:** Evolve an archive of different high-quality solutions (= elites)

# Multi-dimensional Archive of Phenotypic Elites (MAP-Elites)




## MAP-Elites algorithm

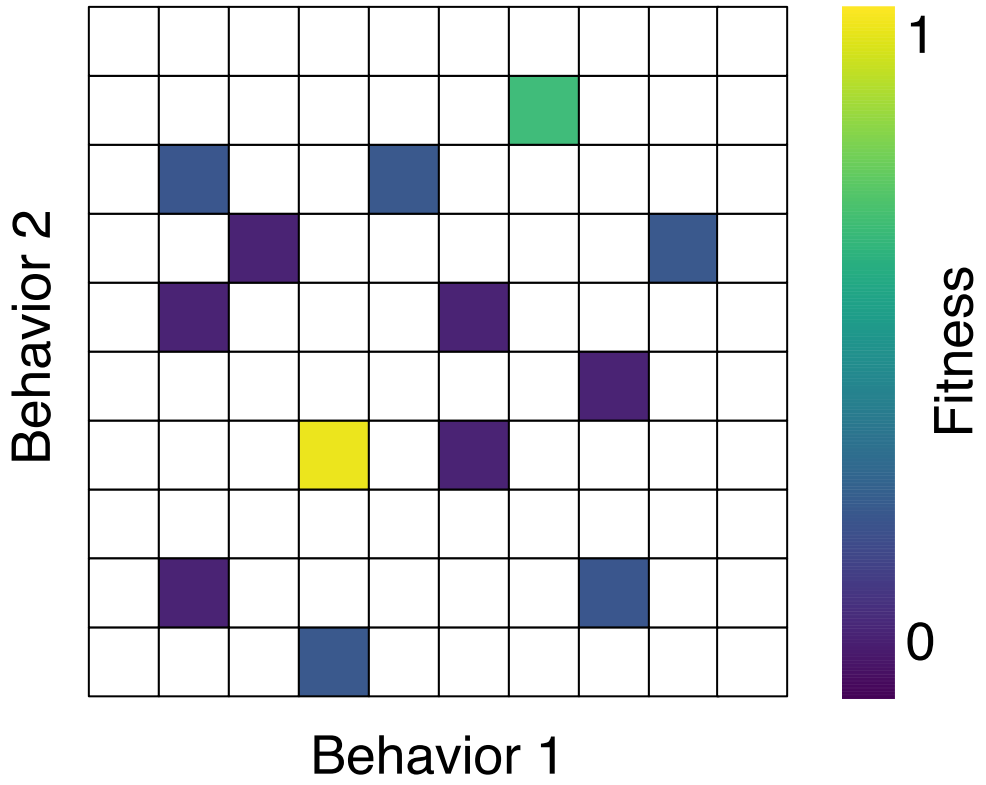
**Initialization**

1. **Divide the behavior space into cells (= the Archive)**
2. Initialize with random solutions until $n_{init}$ elites are found

**Main loop**

1. Pick two elites from the Archive
2. Apply crossover and mutation
3. Evaluate new solution
4. If new behavior or better fitness the solution becomes an elite

# Multi-dimensional Archive of Phenotypic Elites (MAP-Elites)




## MAP-Elites algorithm

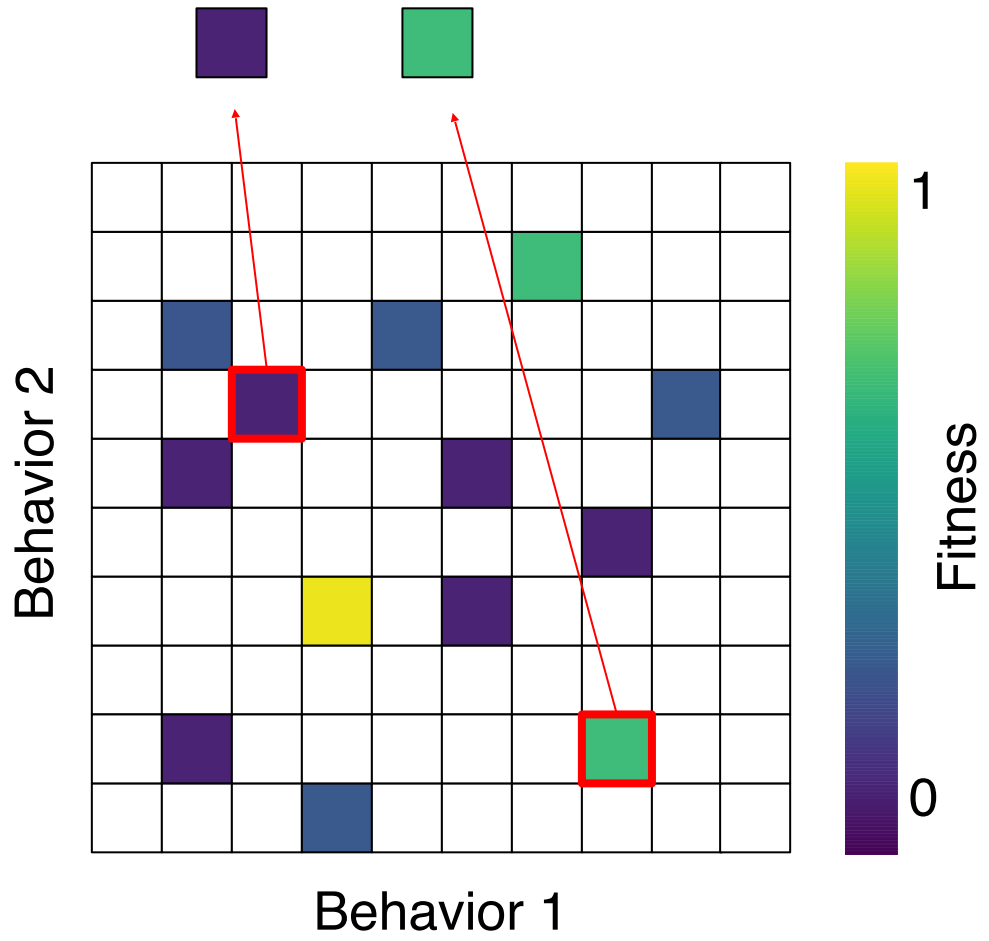
**Initialization**

1. Divide the behavior space into cells (= the Archive)
2. **Initialize with random solutions until $n_{init}$ elites are found**

**Main loop**

1. Pick two elites from the Archive
2. Apply crossover and mutation
3. Evaluate new solution
4. If new behavior or better fitness the solution becomes an elite

# Multi-dimensional Archive of Phenotypic Elites (MAP-Elites)




## MAP-Elites algorithm

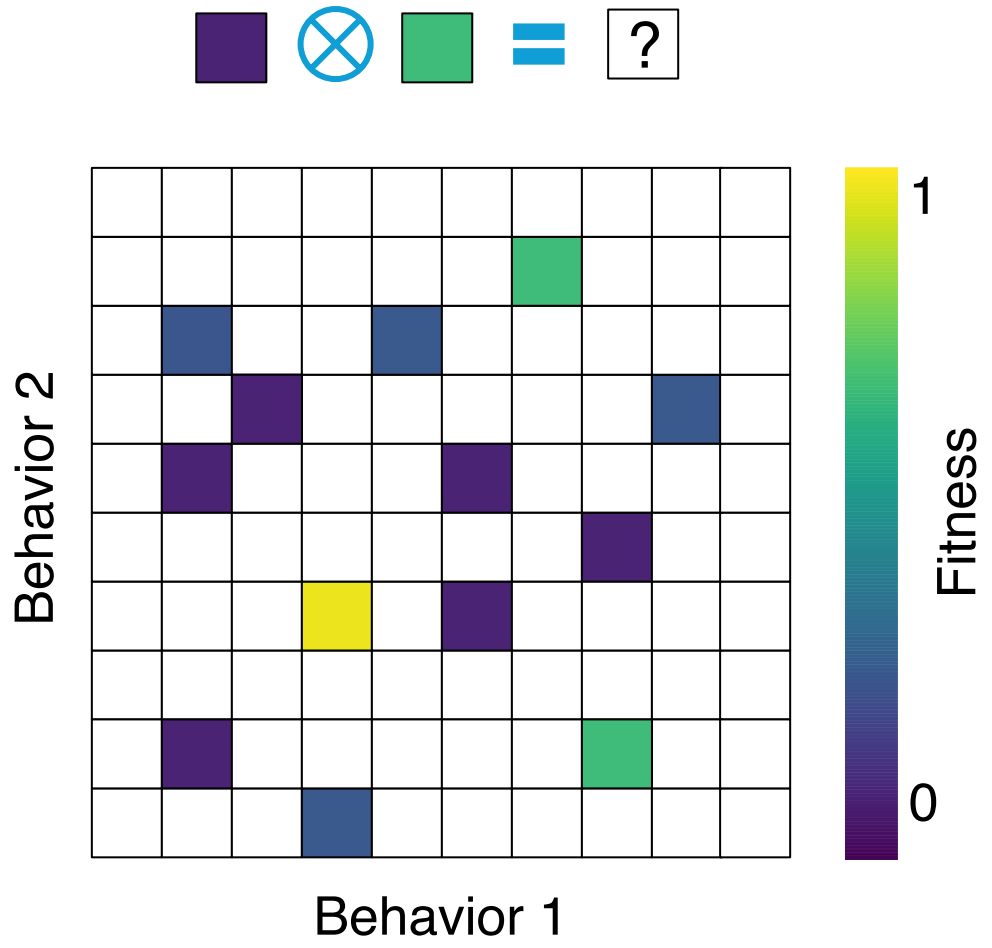
**Initialization**

1. Divide the behavior space into cells (= the Archive)
2. Initialize with random solutions until $n_{init}$ elites are found

**Main loop**

1. **Pick two elites from the Archive**
2. Apply crossover and mutation
3. Evaluate new solution
4. If new behavior or better fitness the solution becomes an elite

# Multi-dimensional Archive of Phenotypic Elites (MAP-Elites)




## MAP-Elites algorithm

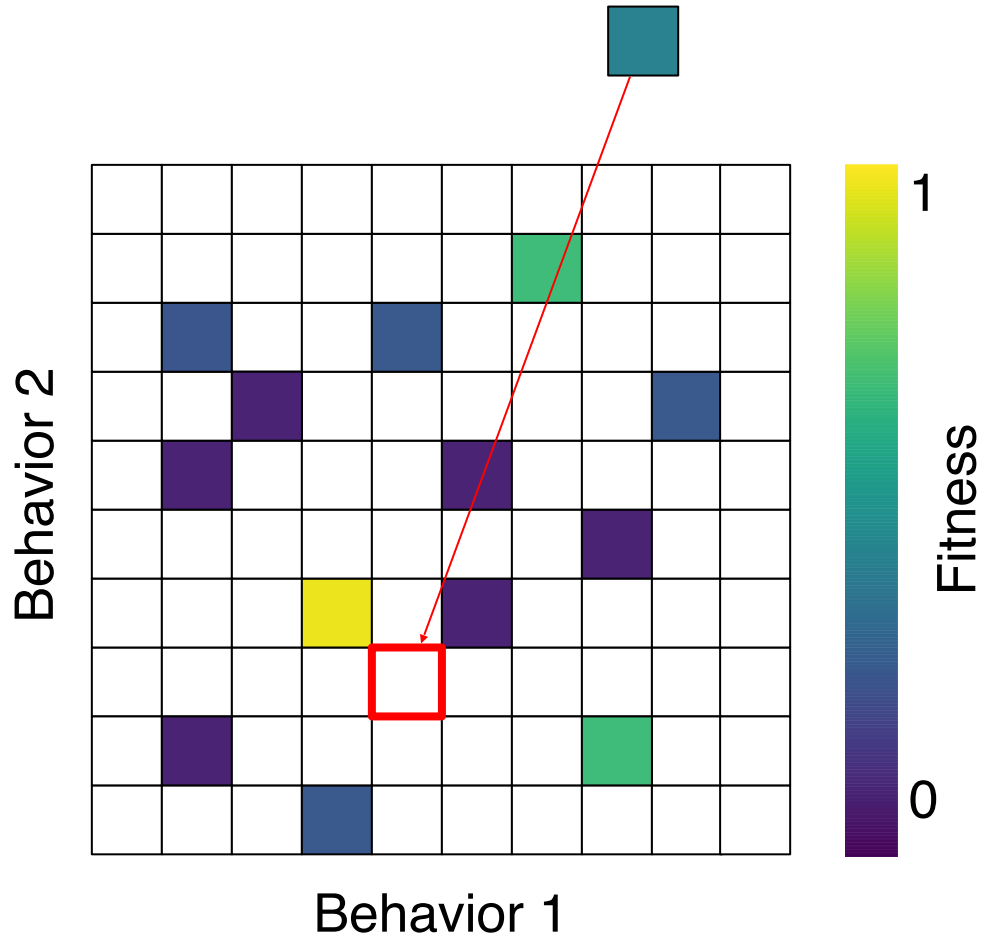
**Initialization**

1. Divide the behavior space into cells (= the Archive)
2. Initialize with random solutions until $n_{init}$ elites are found

**Main loop**

1. Pick two elites from the Archive
2. **Apply crossover and mutation**
3. Evaluate new solution
4. If new behavior or better fitness the solution becomes an elite

# Multi-dimensional Archive of Phenotypic Elites (MAP-Elites)




## MAP-Elites algorithm

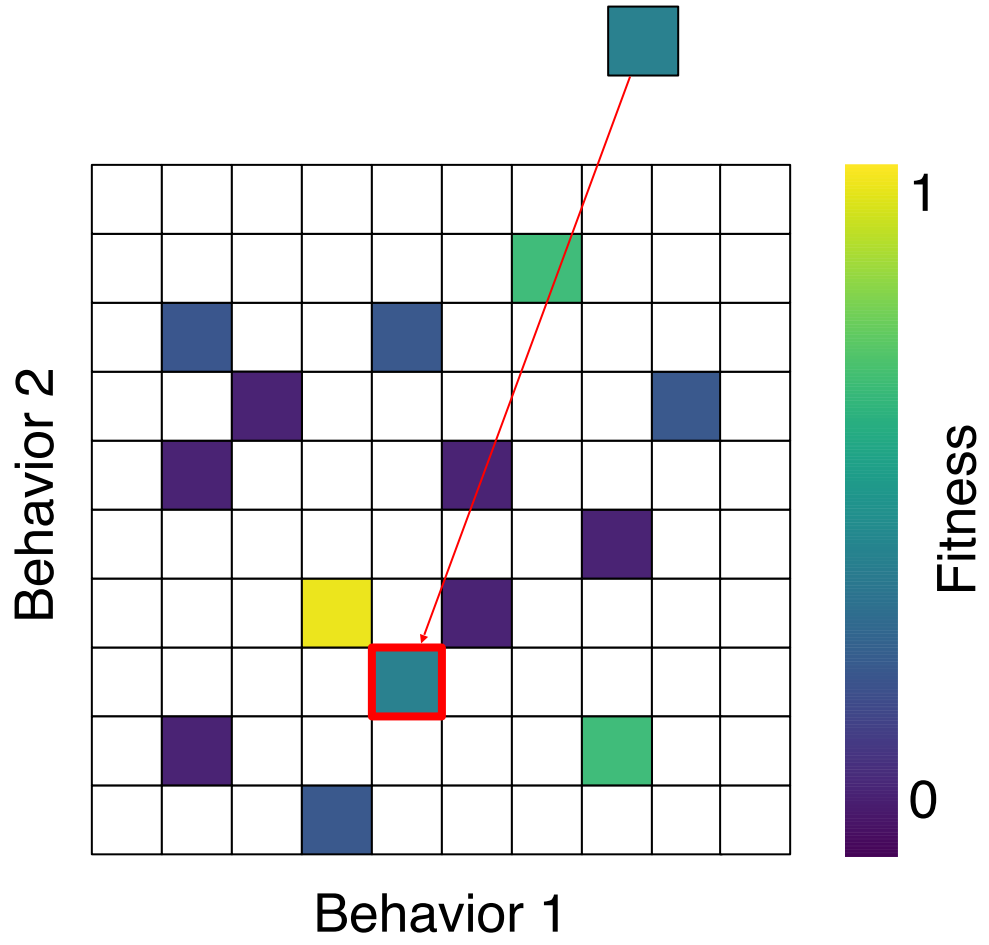
**Initialization**

1. Divide the behavior space into cells (= the Archive)
2. Initialize with random solutions until $n_{init}$ elites are found

**Main loop**

1. Pick two elites from the Archive
2. Apply crossover and mutation
3. **Evaluate new solution**
4. If new behavior or better fitness the solution becomes an elite

# Multi-dimensional Archive of Phenotypic Elites (MAP-Elites)




## MAP-Elites algorithm

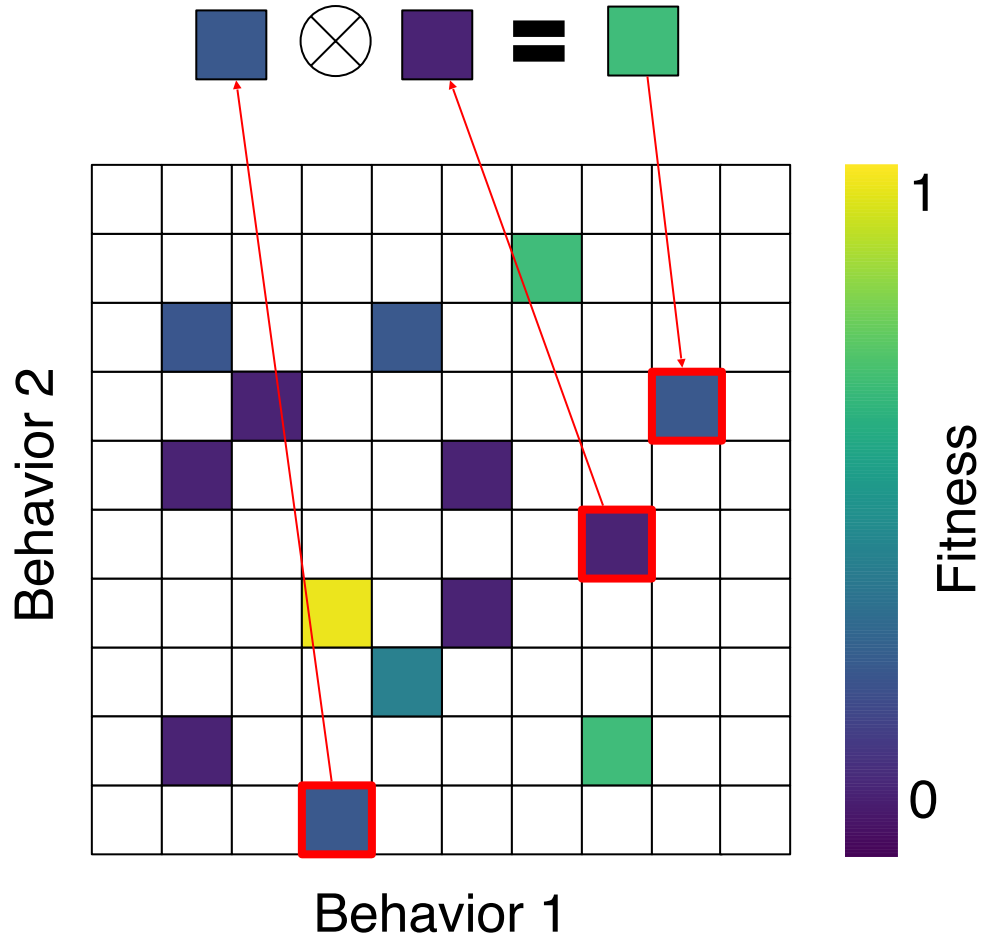
**Initialization**

1. Divide the behavior space into cells (= the Archive)
2. Initialize with random solutions until $n_{init}$ elites are found

**Main loop**

1. Pick two elites from the Archive
2. Apply crossover and mutation
3. Evaluate new solution
4. **If new behavior** or better fitness **the solution becomes an elite**

# Multi-dimensional Archive of Phenotypic Elites (MAP-Elites)




## MAP-Elites algorithm

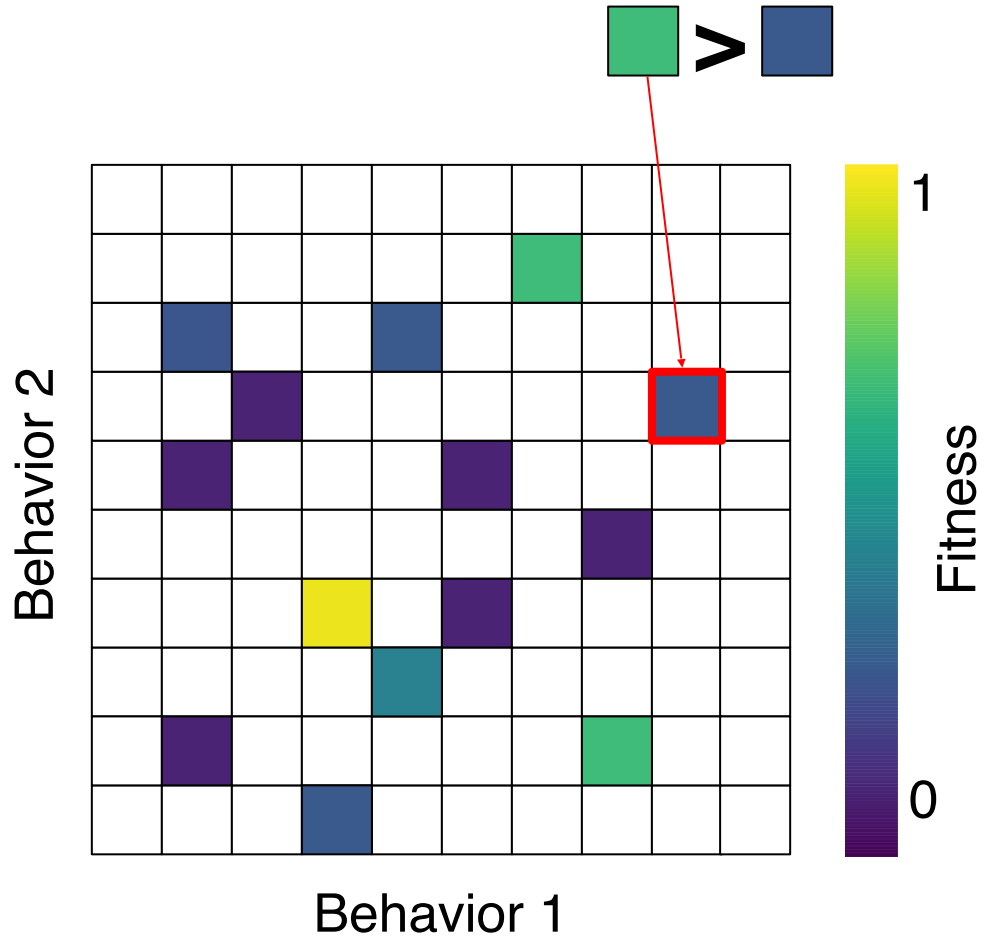
**Initialization**

1. Divide the behavior space into cells (= the Archive)
2. Initialize with random solutions until $n_{init}$ elites are found

**Main loop**

1. **Pick two elites from the Archive**
2. **Apply crossover and mutation**
3. **Evaluate new solution**
4. If new behavior or better fitness the solution becomes an elite

# Multi-dimensional Archive of Phenotypic Elites (MAP-Elites)




## MAP-Elites algorithm

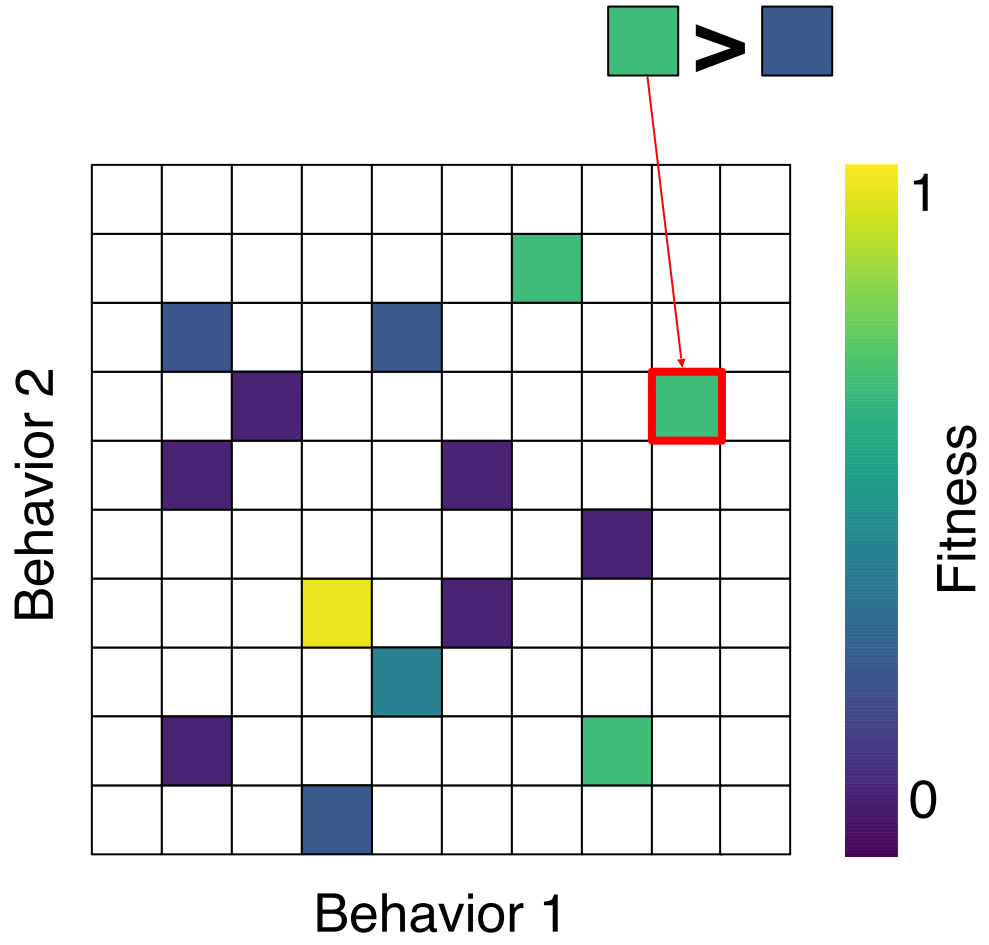
### Initialization

1. Divide the behavior space into cells (= the Archive)
2. Initialize with random solutions until $n_{init}$ elites are found

### Main loop

1. Pick two elites from the Archive
2. Apply crossover and mutation
3. Evaluate new solution
4. **If** new behavior or **better fitness** **the solution becomes an elite**

# Multi-dimensional Archive of Phenotypic Elites (MAP-Elites)




## MAP-Elites algorithm

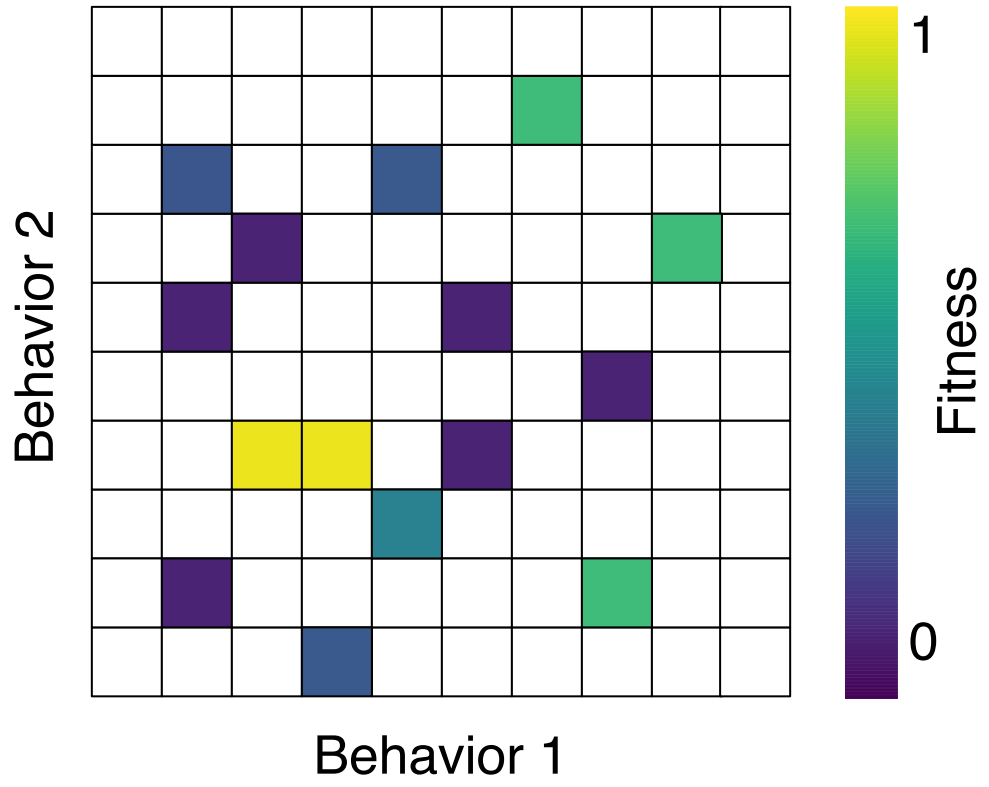
### Initialization

1. Divide the behavior space into cells (= the Archive)
2. Initialize with random solutions until $n_{init}$ elites are found

### Main loop

1. Pick two elites from the Archive
2. Apply crossover and mutation
3. Evaluate new solution
4. **If** new behavior or **better fitness** **the solution becomes an elite**

# Multi-dimensional Archive of Phenotypic Elites (MAP-Elites)




## MAP-Elites algorithm

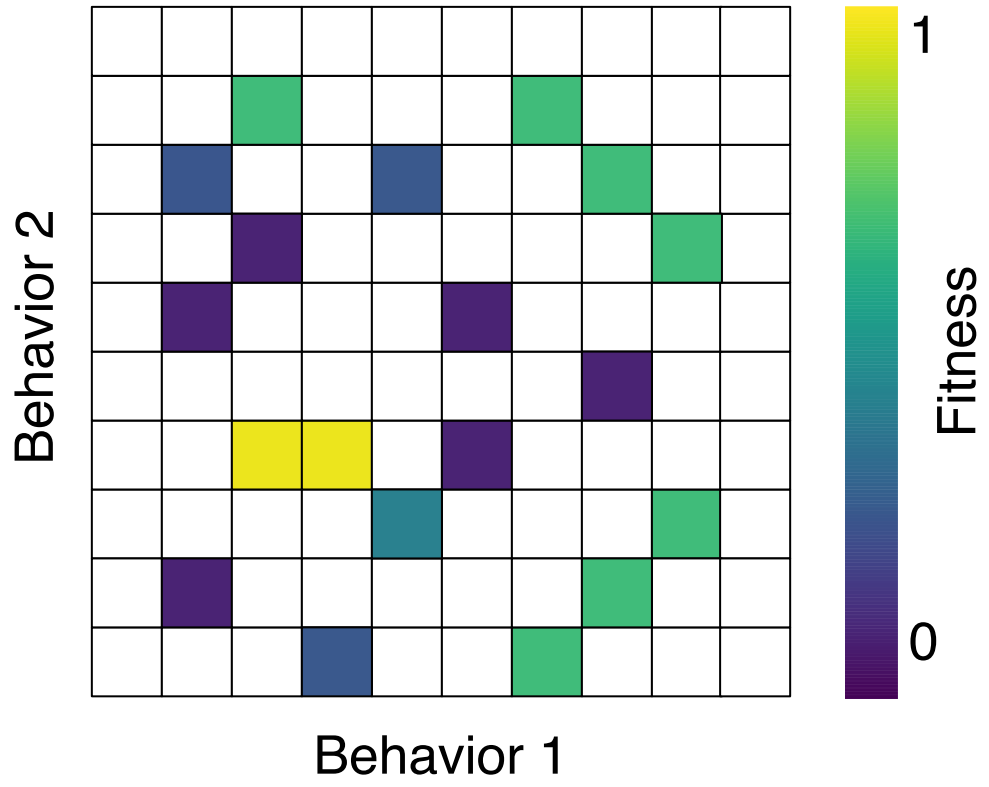
**Initialization**

1. Divide the behavior space into cells (= the Archive)
2. Initialize with random solutions until $n_{init}$ elites are found

**Main loop**

1. **Pick two elites from the Archive**
2. **Apply crossover and mutation**
3. **Evaluate new solution**
4. **If new behavior or better fitness the solution becomes an elite**

# Multi-dimensional Archive of Phenotypic Elites (MAP-Elites)




## MAP-Elites algorithm

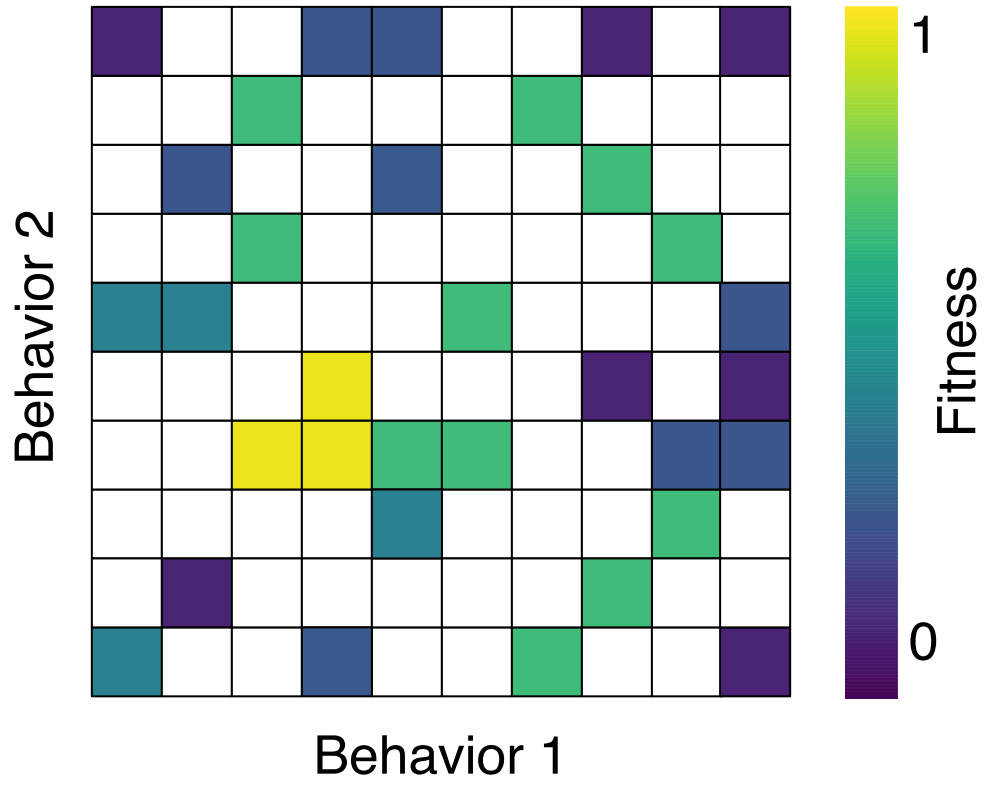
**Initialization**

1. Divide the behavior space into cells (= the Archive)
2. Initialize with random solutions until $n_{init}$ elites are found

**Main loop**

1. **Pick two elites from the Archive**
2. **Apply crossover and mutation**
3. **Evaluate new solution**
4. **If new behavior or better fitness the solution becomes an elite**

# Multi-dimensional Archive of Phenotypic Elites (MAP-Elites)




## MAP-Elites algorithm

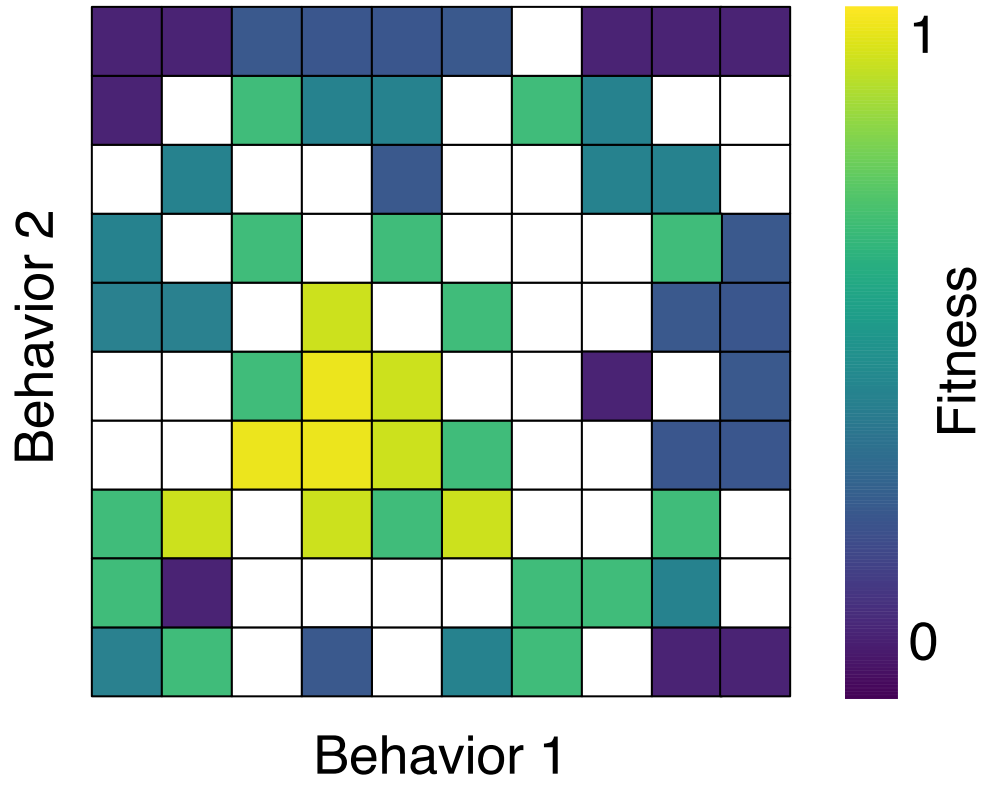
**Initialization**

1. Divide the behavior space into cells (= the Archive)
2. Initialize with random solutions until $n_{init}$ elites are found

**Main loop**

1. **Pick two elites from the Archive**
2. **Apply crossover and mutation**
3. **Evaluate new solution**
4. **If new behavior or better fitness the solution becomes an elite**

# Multi-dimensional Archive of Phenotypic Elites (MAP-Elites)




## MAP-Elites algorithm

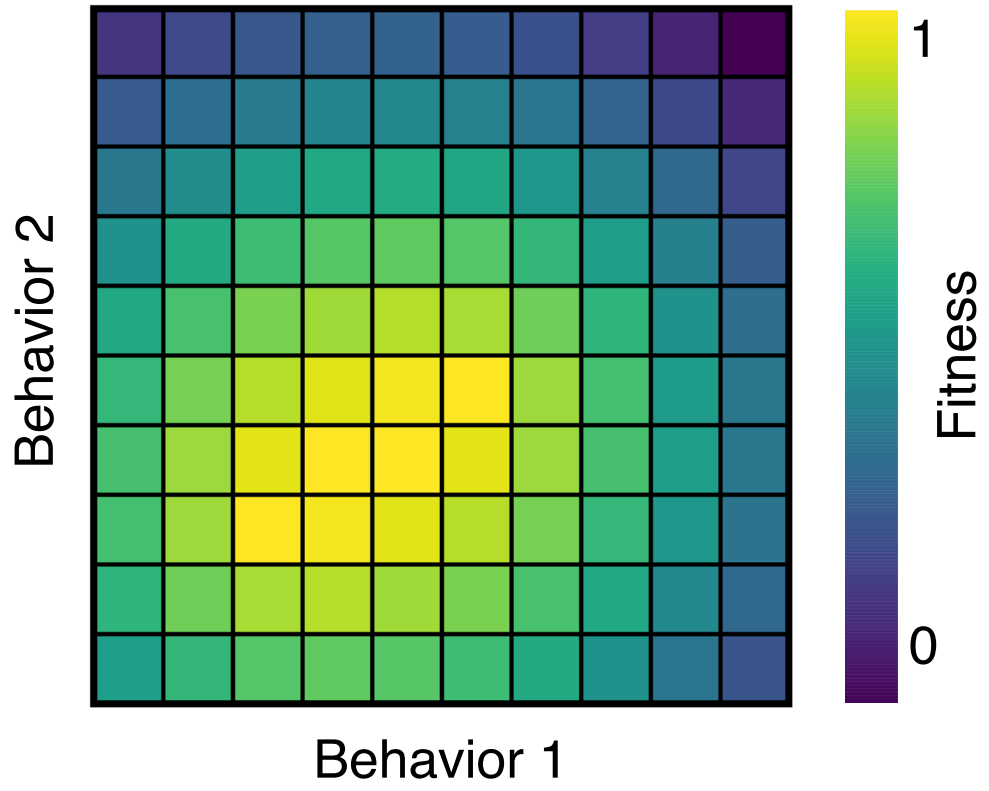
**Initialization**

1. Divide the behavior space into cells (= the Archive)
2. Initialize with random solutions until $n_{init}$ elites are found

**Main loop**

1. **Pick two elites from the Archive**
2. **Apply crossover and mutation**
3. **Evaluate new solution**
4. **If new behavior or better fitness the solution becomes an elite**

# Multi-dimensional Archive of Phenotypic Elites (MAP-Elites)




## MAP-Elites algorithm

**Initialization**

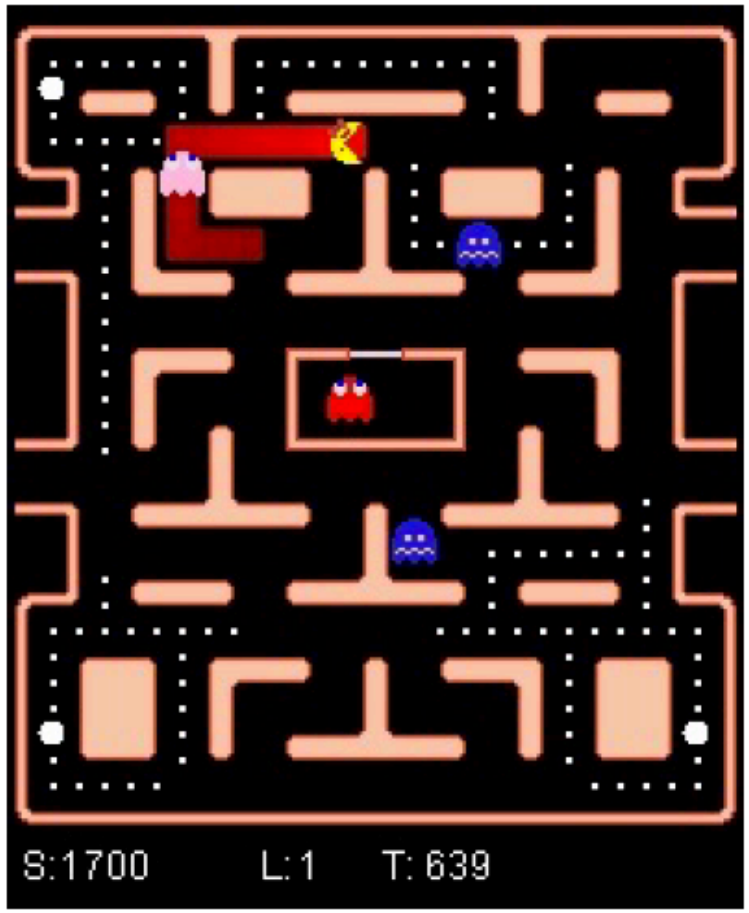
1. Divide the behavior space into cells (= the Archive)
2. Initialize with random solutions until $n_{init}$ elites are found

**Main loop**

1. Pick two elites from the Archive
2. Apply crossover and mutation
3. Evaluate new solution
4. If new behavior or better fitness the solution becomes an elite

# Evolving intelligent agents

# 4.2 Evolving Behavioral Strategies



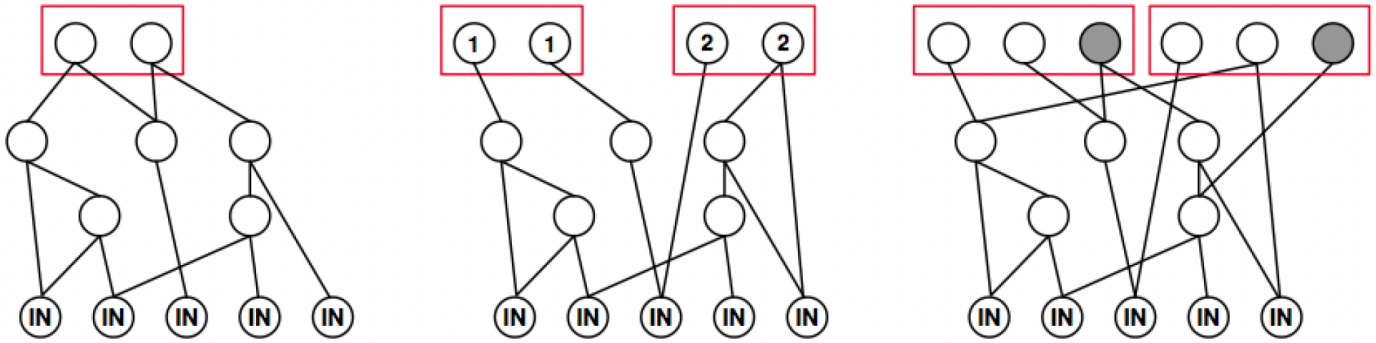

(Schrum and Miikkulainen 2015)

Strategy: different behaviors at different times

Ms. Pac-Man:

- Agents perform many different tasks
  - E.g. eat pills, avoid ghosts, eat powerpills, eat ghosts
  - Sometimes clearly separate in time
  - Sometimes multiple tasks at once
- How can we evolve them into a single network?

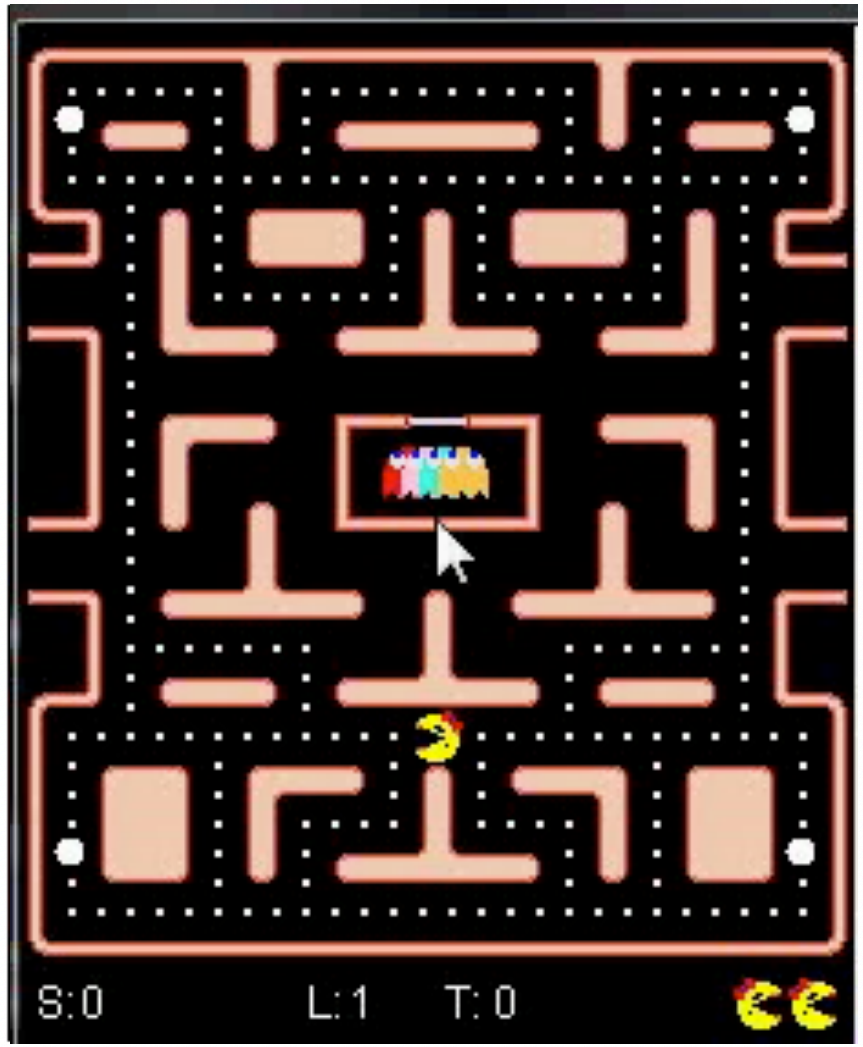
# MM-NEAT: Modular Multiobjective Approach




(a) Single-module Network (b) Multitask Network (c) Preference Neuron Network

- Evolution discovers modules and when to use them
  - Vs. human-designed division with multitasking
- Multiple modules with preference neurons
  - Modules implement different behaviors
  - Preference neurons used to choose among them
  - Module-mutation adds new modules
- Evolved towards multiple objectives
  - Correspond to dimensions of game play
  - E.g. pills and ghosts in Ms. Pac-Man

# Human-Designed Task Division



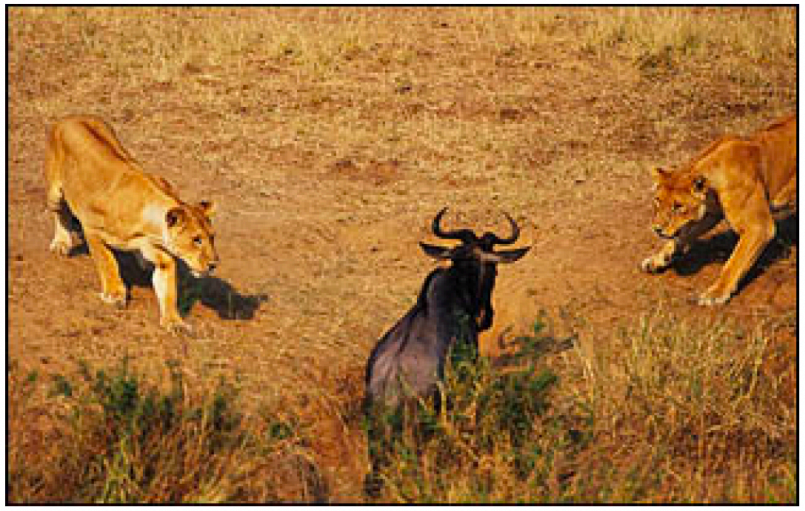

- Multitask approach
  - One module for threat ghosts
  - Another module for edible ghosts
  - Works ok, but...
  - DEMO

# Evolution-Discovered Task Division

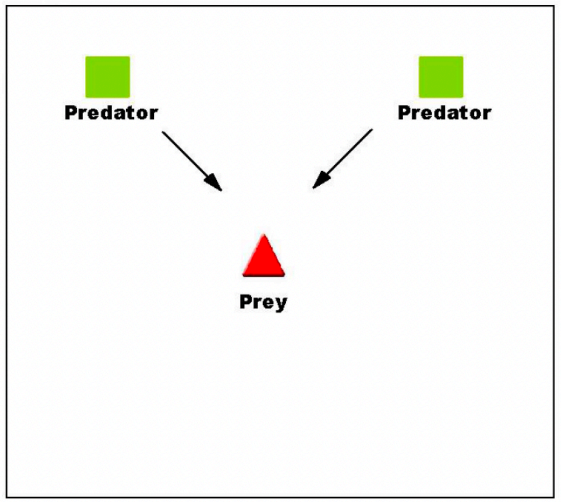

- One module used 95% of the time
  - Eat pills, avoid ghosts, chase ghosts
  - Different behaviors with a common base
- A second module 5% of the time
  - Luring ghosts near a power pill
  - Escaping from tight spaces
- A different multimodal perspective
- Not as obvious, but more powerful
- DEMO

# Evolving Collective Systems

# Evolving Collective Behavior



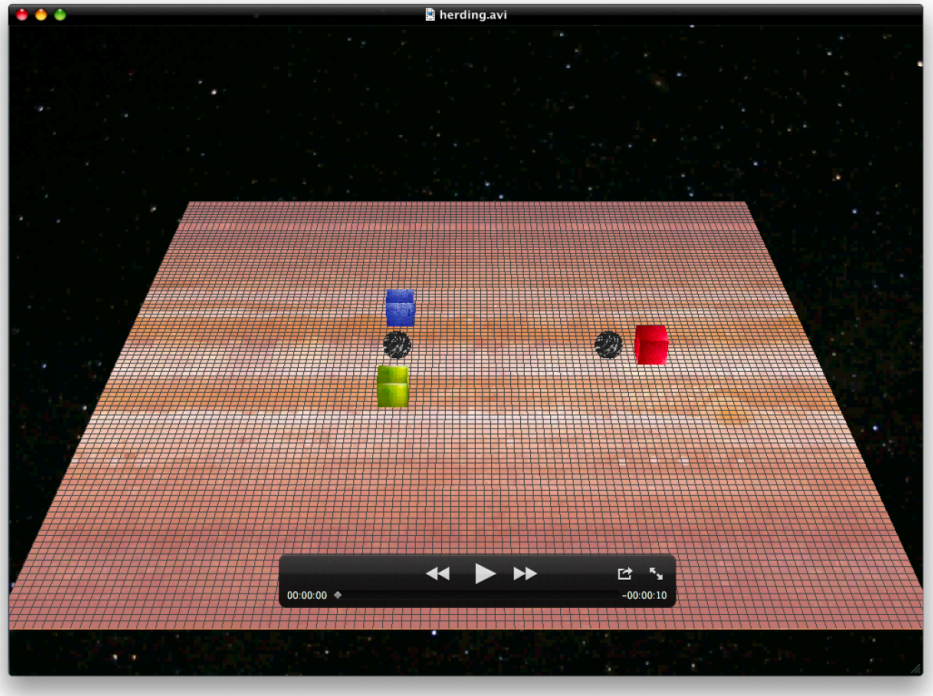
Natural predators and prey




Formalization of behavior

- Complex cooperation observed in pursuit and evasion
  - Motivated by biology, esp. hyenas vs. zebras (Kay Holekamp, MSU)
  - Largely innate, possible to see behaviors and their evolution
- Such behaviors evolve together, in coevolutionary environment
  - Simultaneous competitive and cooperative coevolution

# Experimental Setup

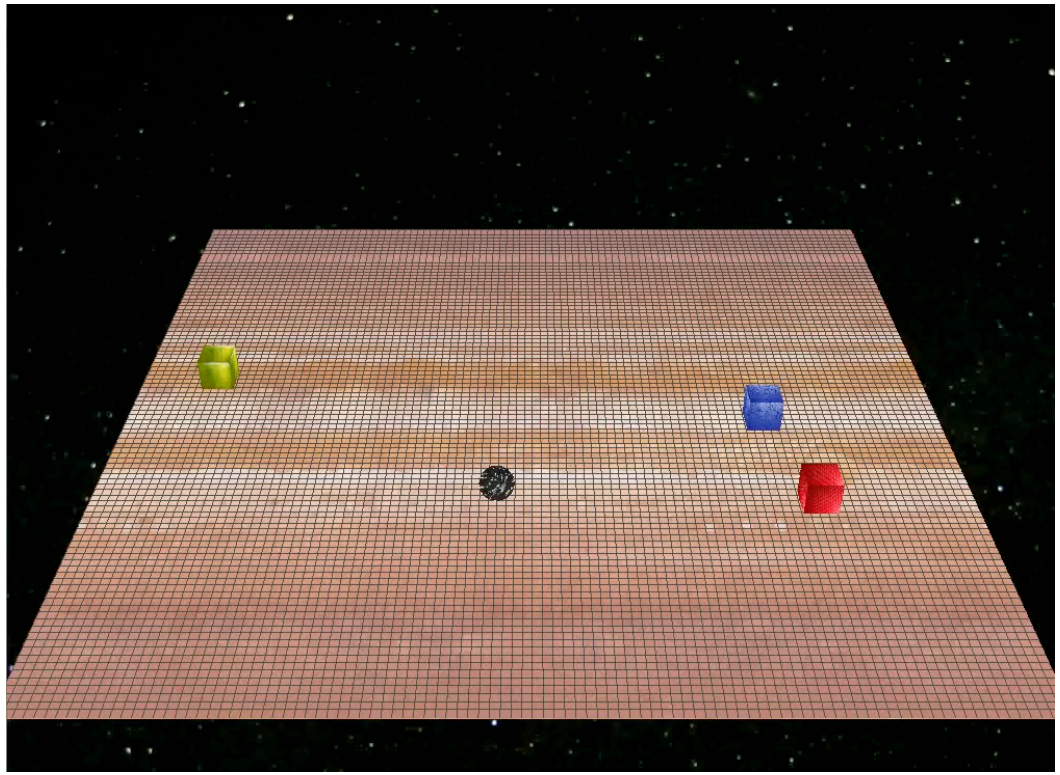


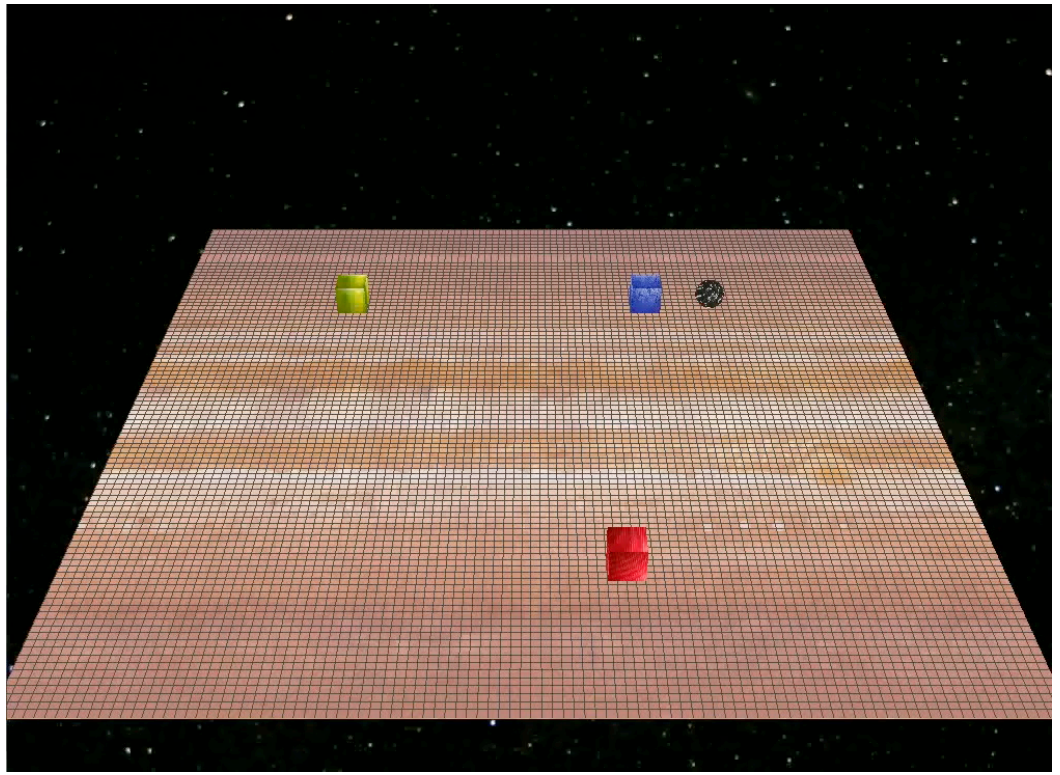

(Rawal et al. 2010)

- Toroidal grid world
- Predators, prey move with same speed in 4 directions
- No direct communication between team members
  - Communication still possible through stigmergy
- Does a coevolutionary arms race result?

DEMO

# Evolutionary Arms Race





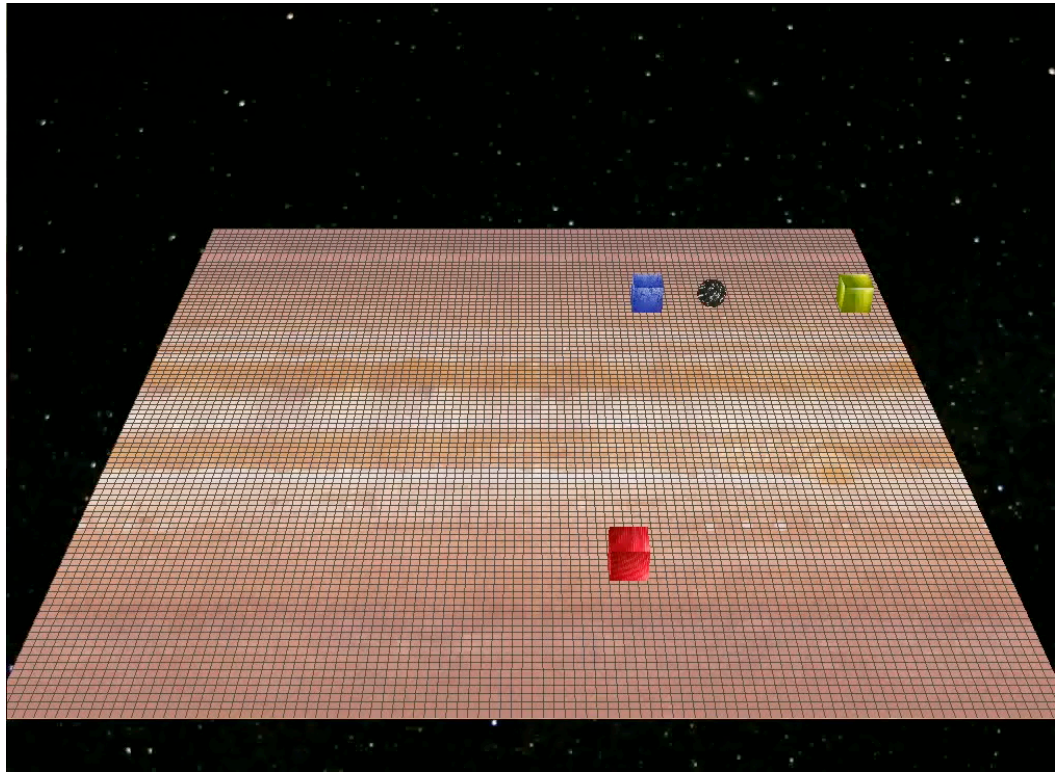
50-75: Single predator catches the prey

75-100: Prey evades by circling

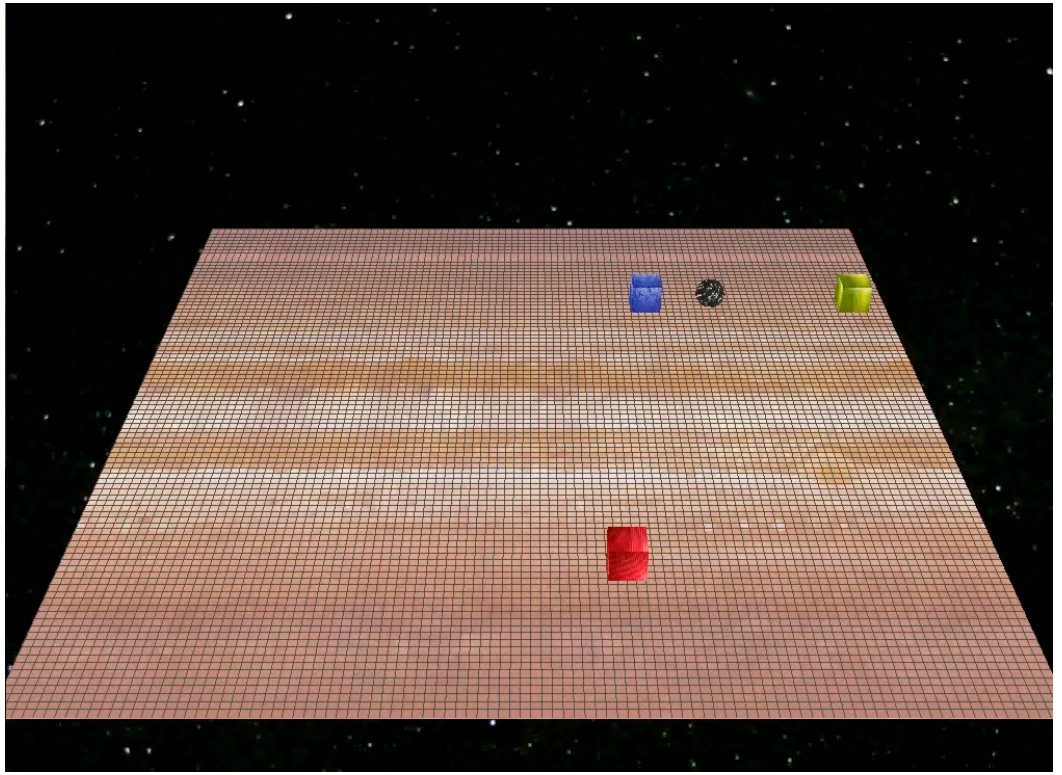
Predators and prey populations develop increasingly sophisticated behaviors

- Each improvement provides a challenge to the other population

# Evolutionary Arms Race

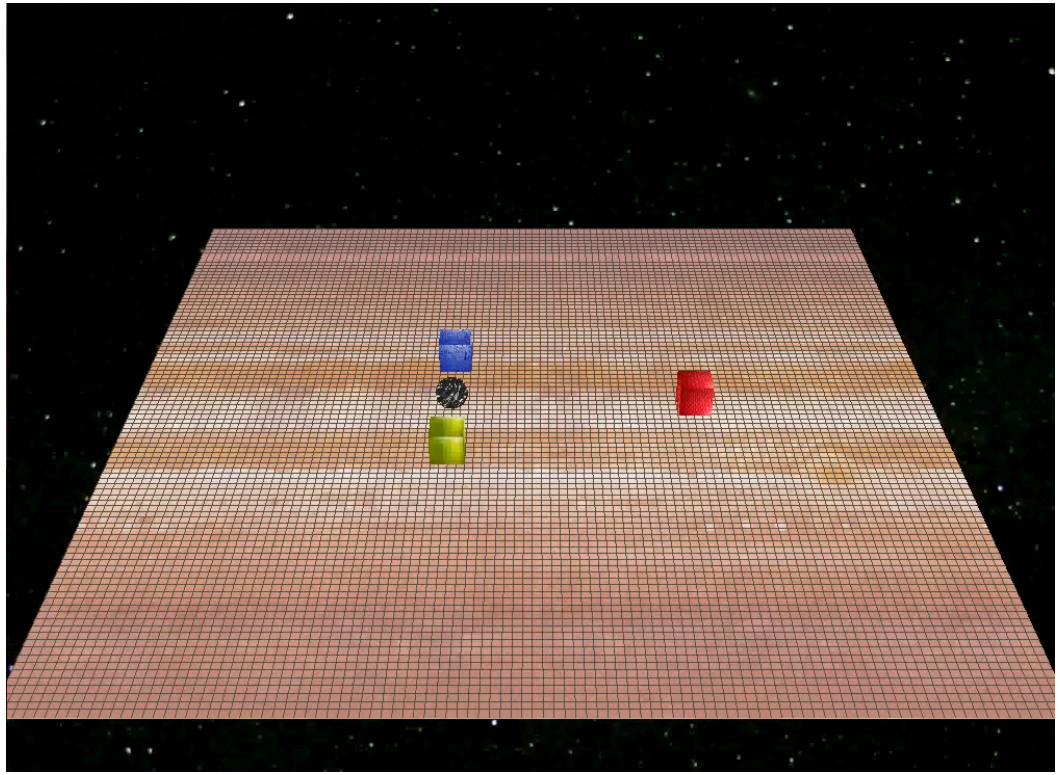


100-150: Two predators cooperate



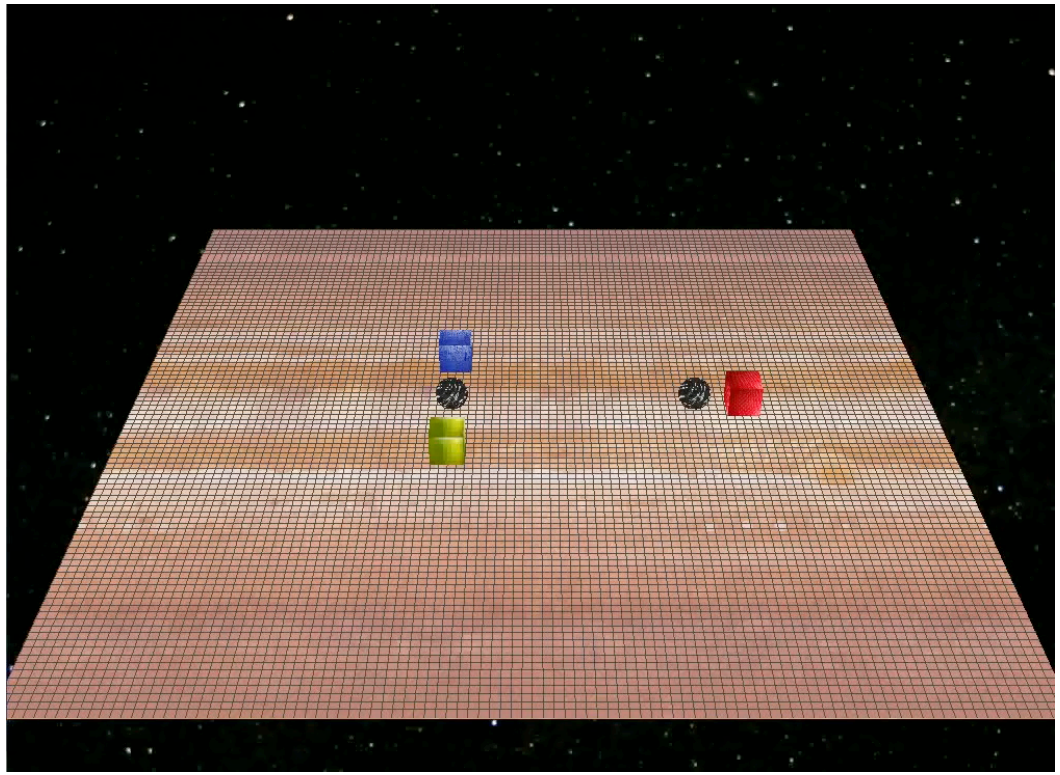
150-180: Prey baits and escapes

# Evolutionary Arms Race

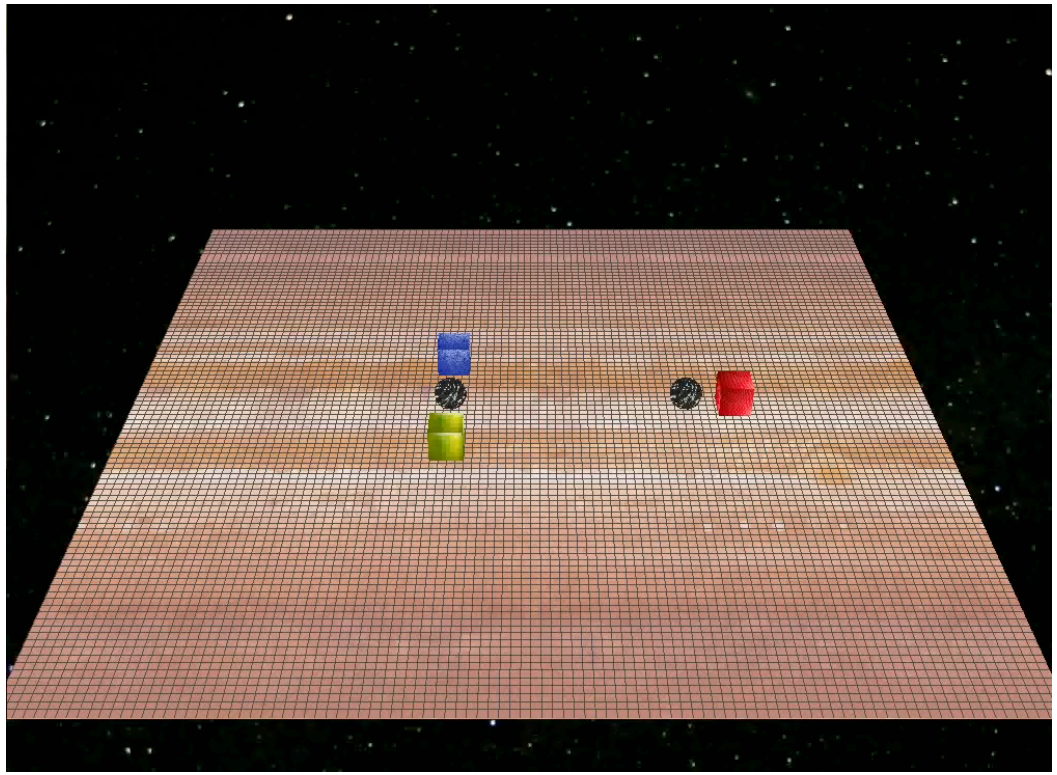


180-200: All predators cooperate

# Evolutionary Arms Race



200-250: Predators herd two prey



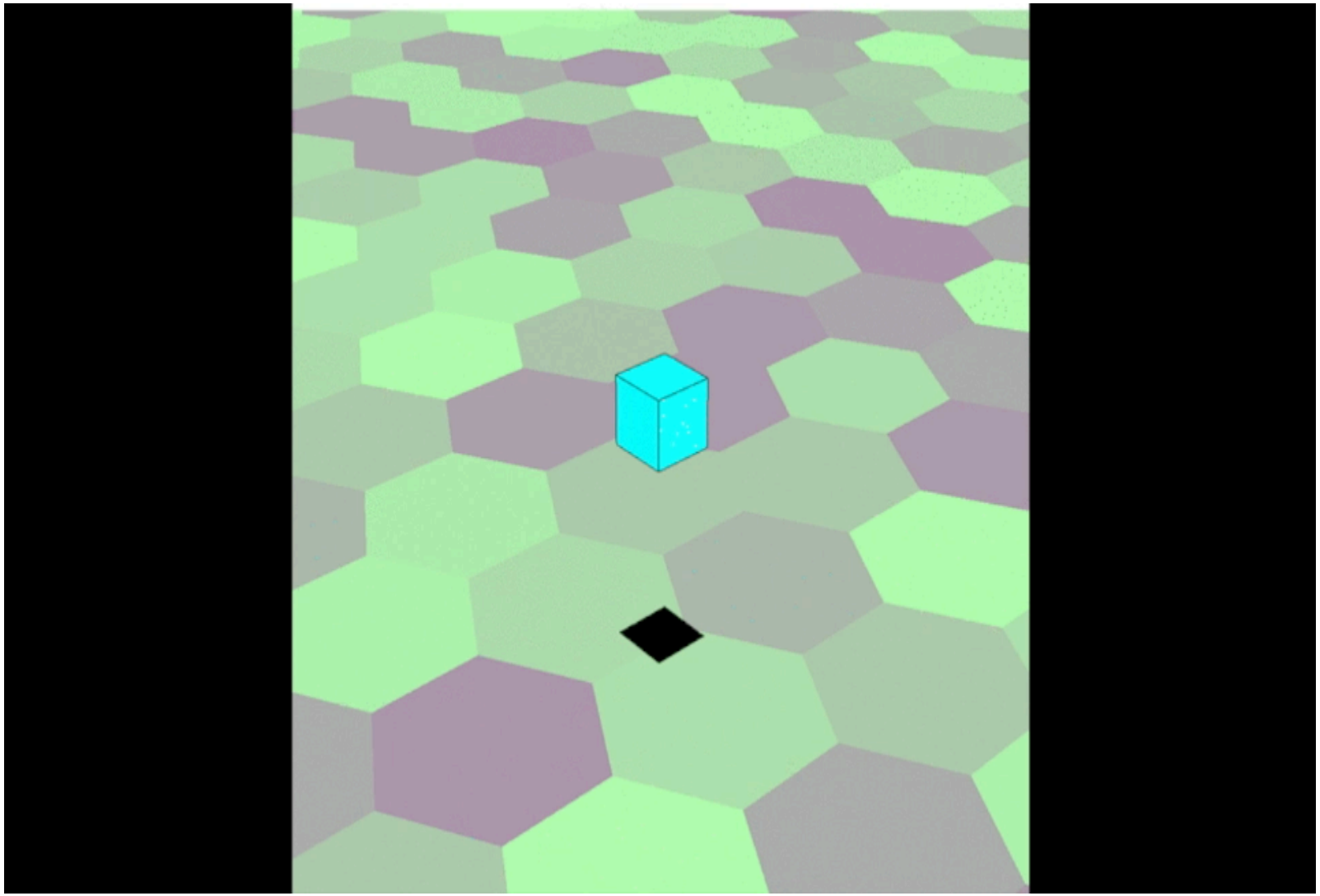
250-300: Prey evade by scattering

# Evolving Neural Cellular Automata

- Neural Cellular Automata (**Chua et al. 1988**)
- Differentiable NCA (**Mordvintsev et al. 2020**)

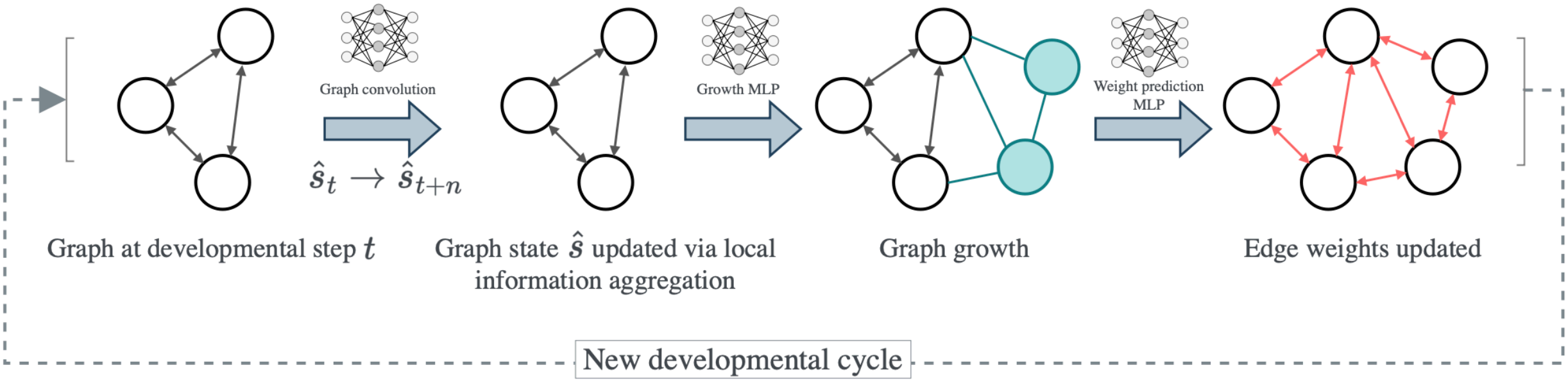
# Regenerating Soft Robots through Neural Cellular Automata

Horibe, Walker, Risi; EuroGP 2021



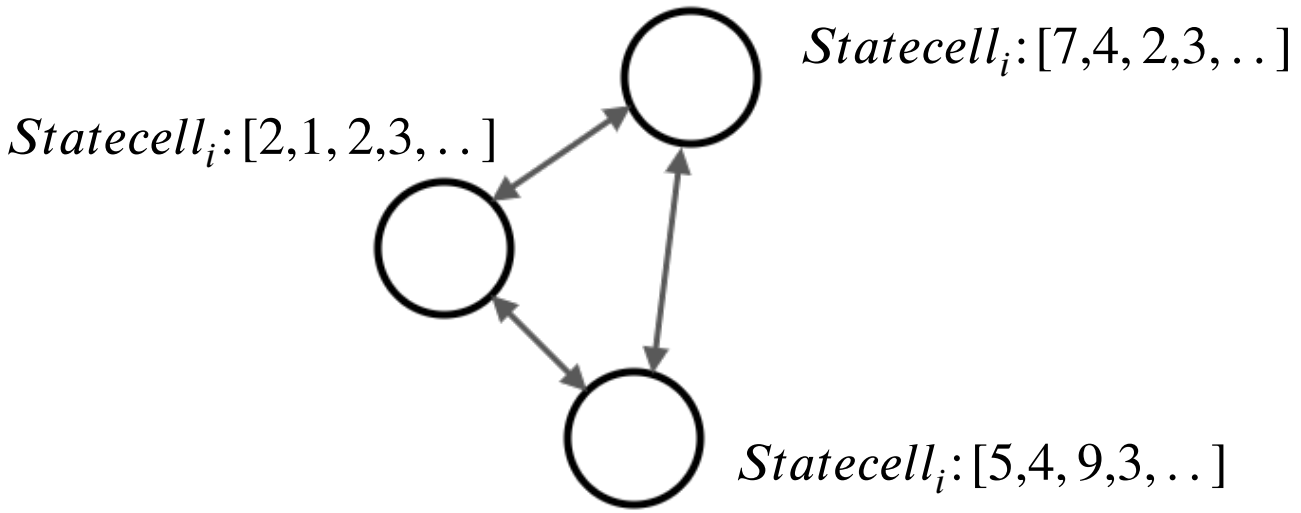
# Growing NNs with neural developmental programs

Najarro, Sudhakaran and Risi, ALIFE 2023



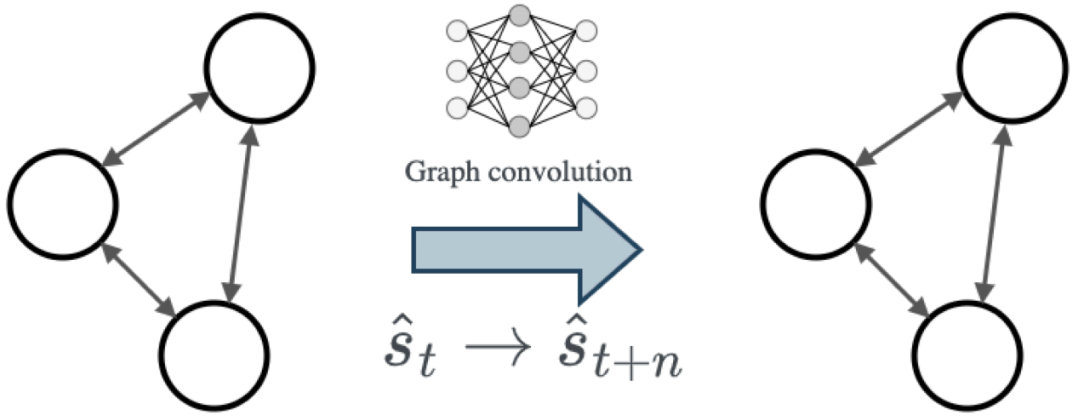
# Self-Assembling ANNs through Neural Developmental Programs

**Step 0**: We start with an initial graph seed



# Self-Assembling ANNs through Neural Developmental Programs

**Step 1**: Update node states via message passing

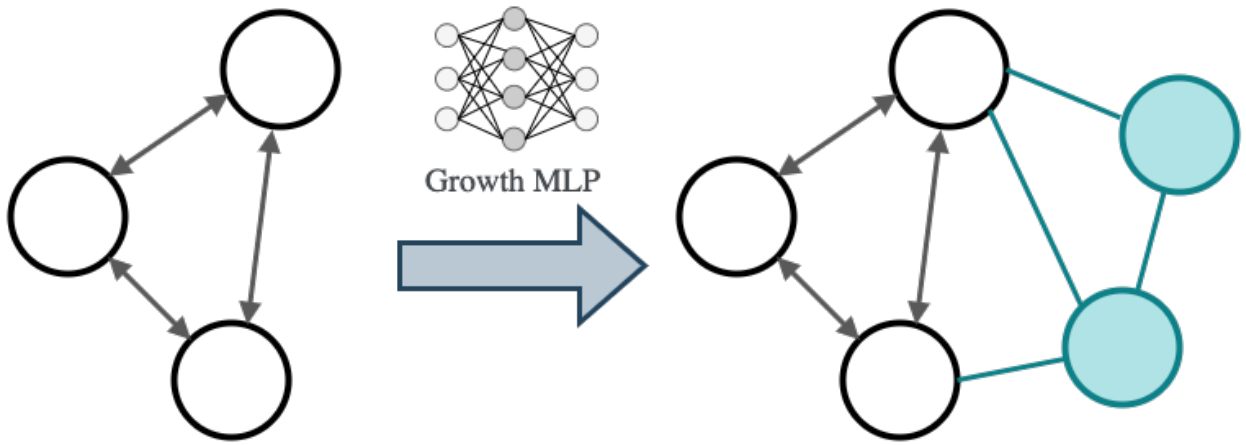



Graph at developmental step $t$

Graph state $\hat{s}$ updated via local information aggregation

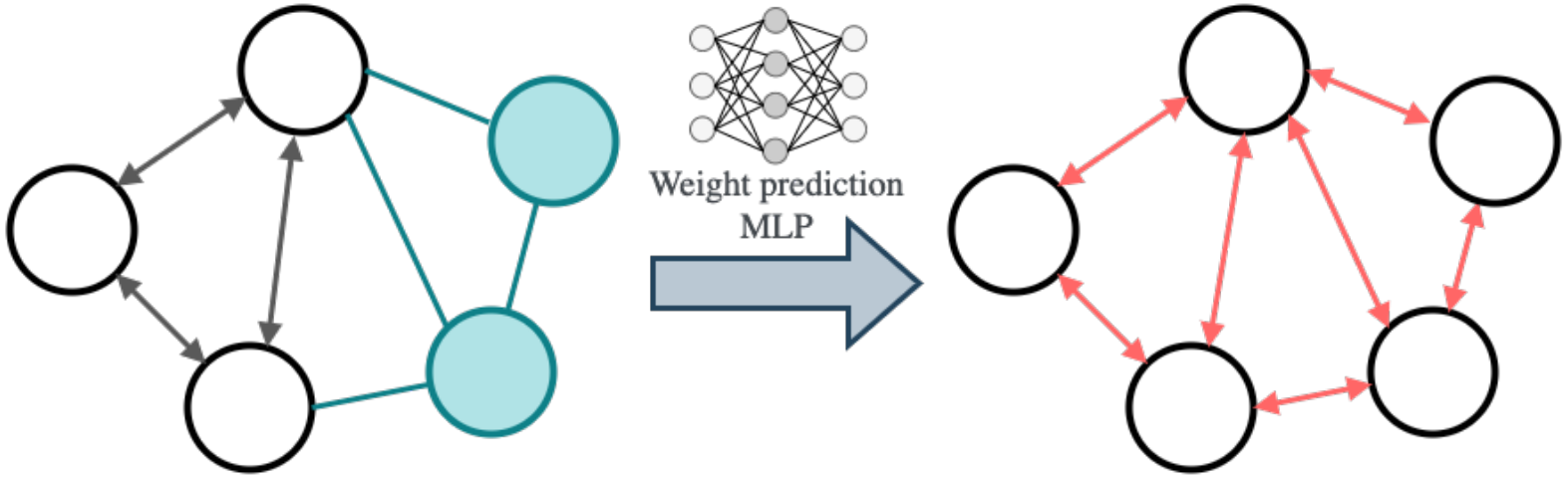
# Self-Assembling ANNs through Neural Developmental Programs

**Step 2**: A neural network decides which nodes will grow

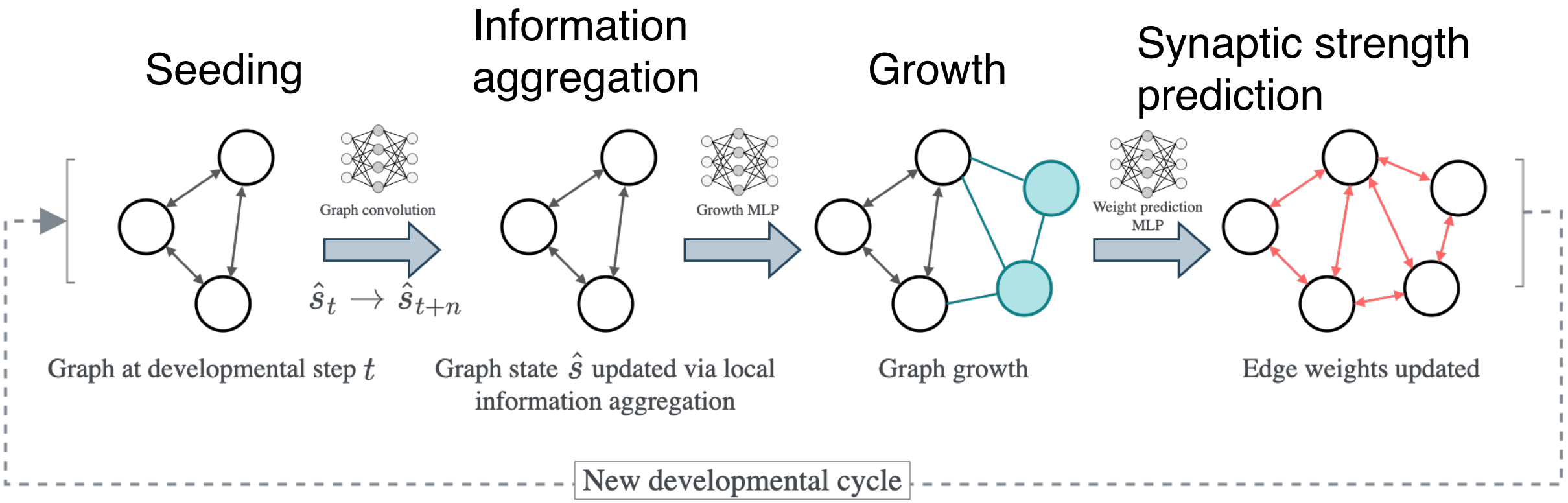


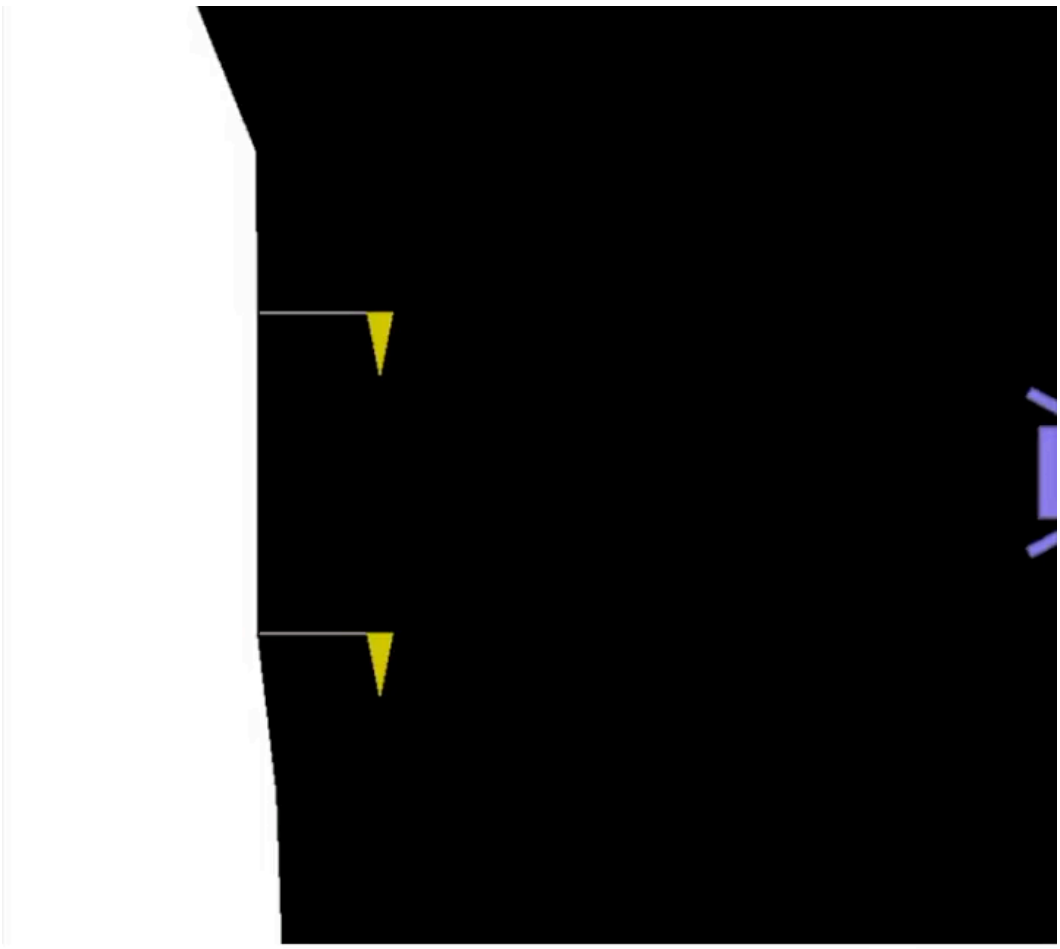
# Self-Assembling ANNs through Neural Developmental Programs

**Step 3**: If network weighted: another NN determines edge weights



# Self-Assembling ANNs through Neural Developmental Programs






Growth cycle: 0, Graph si

# Open-ended Neuroevolution

# Competitive Coevolution of Environments and Solutions

- Paired Open-Ended Trailblazer (POET; Wang et al.)

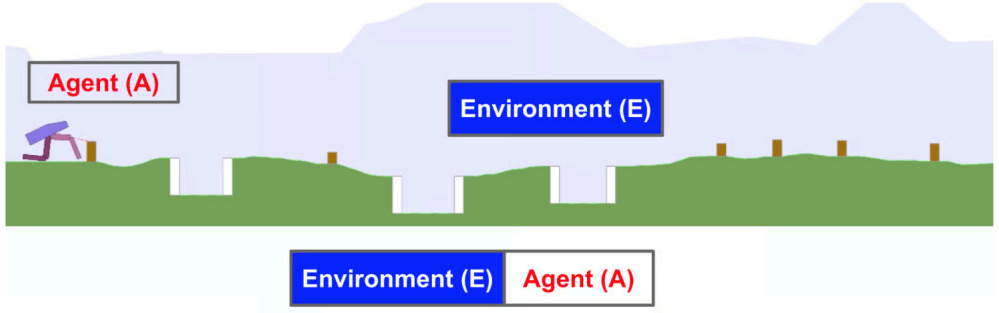



Figure 1: An example bipedal walking environment.

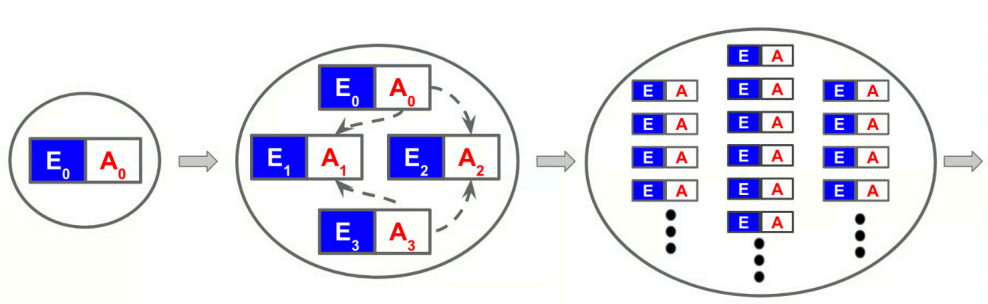
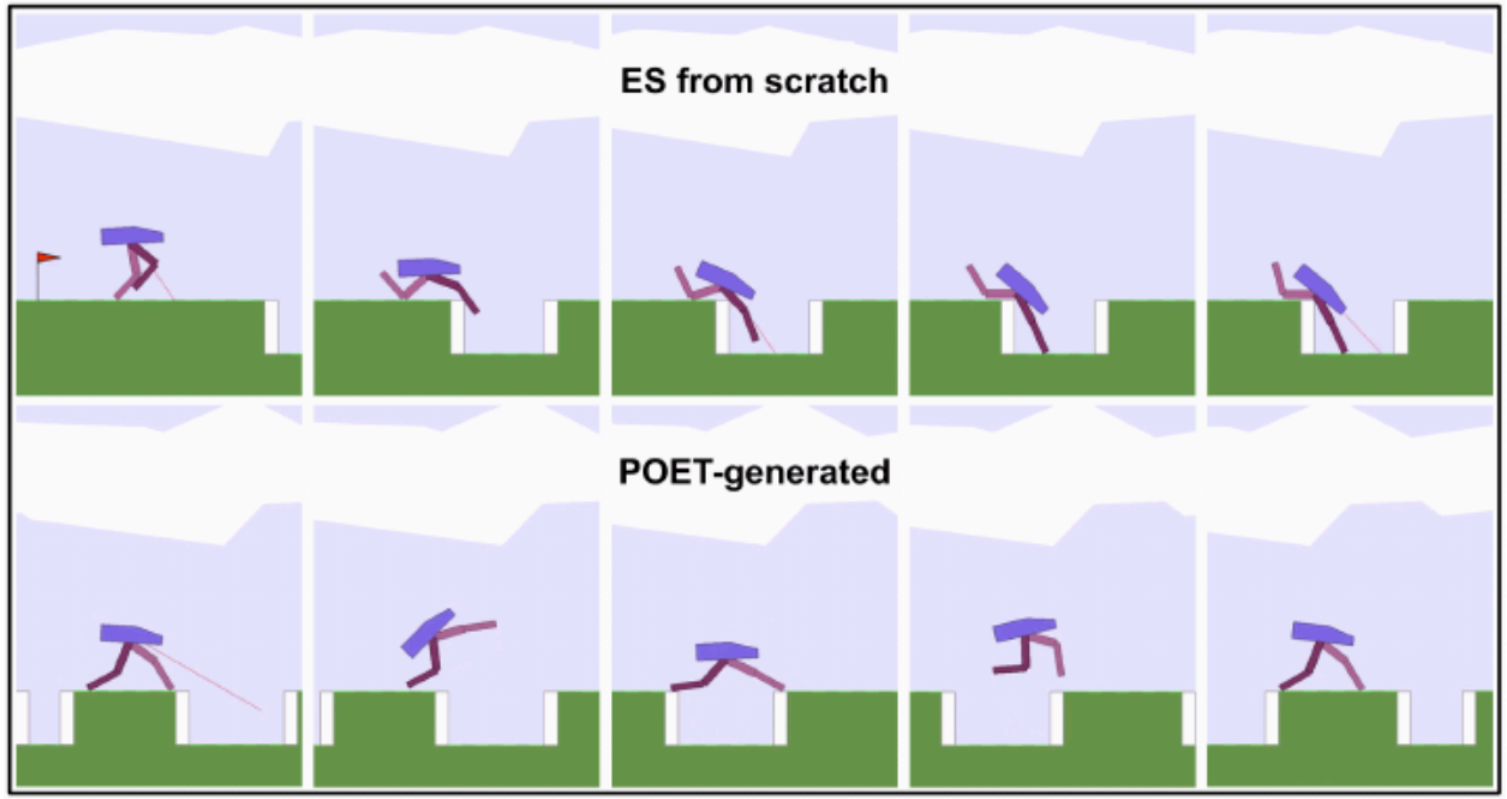
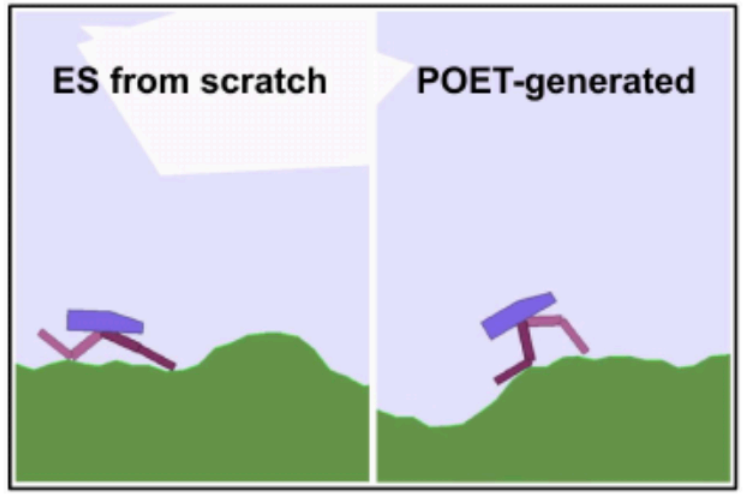


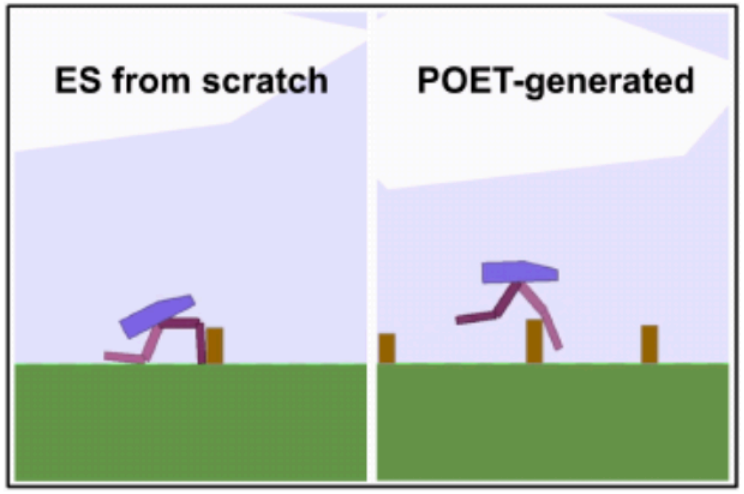

Figure 2: An overview of POET.



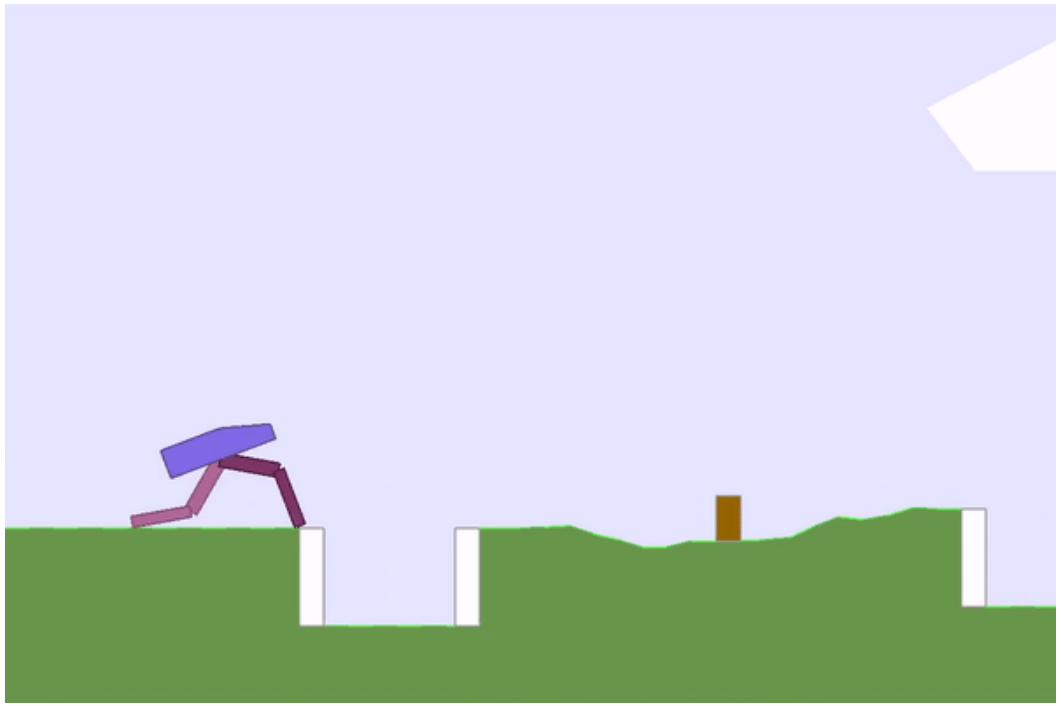

(*a*) Generated agents attempting gaps

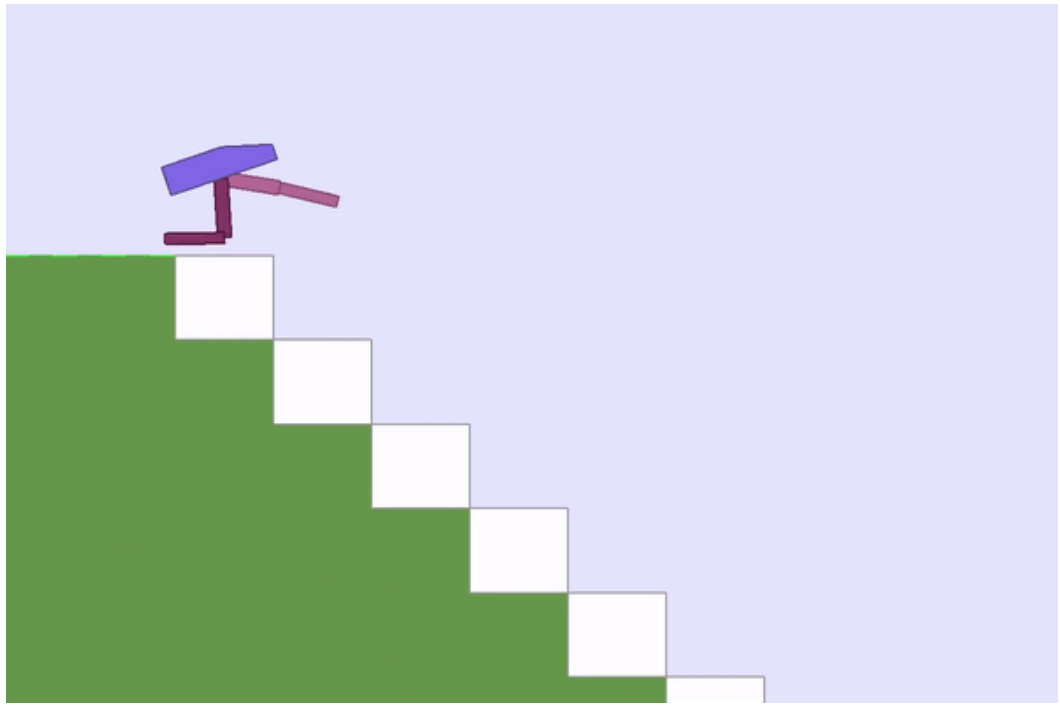



(*b*) Generated agents on rough surfaces




(*c*) Generated agents attempting stumps





# Synergies with ML

# 5.1 Neuroevolution Synergies with Deep Learning

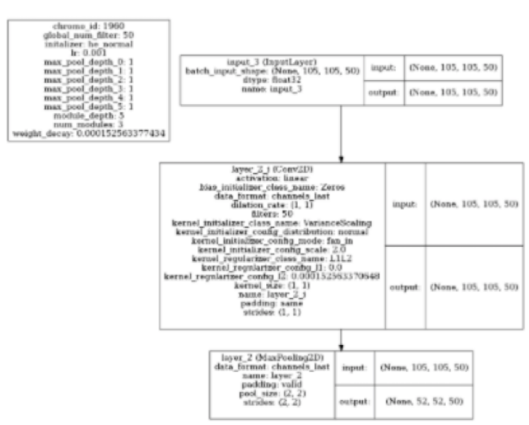
## (A) Fundamental: Neural Architecture Search

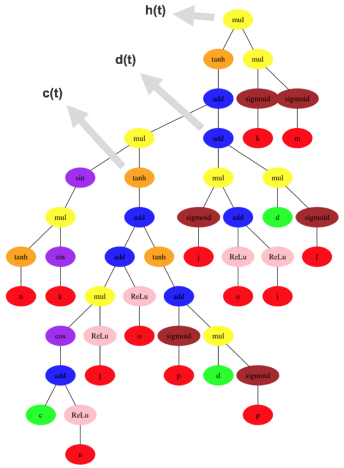
- Optimizing structure and hyperparameters
- Takes advantage of exploration in EC

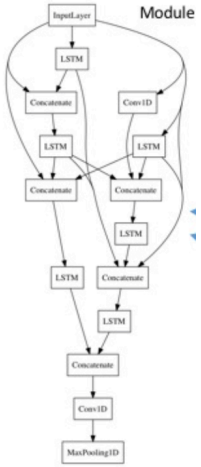
## (B) Extended: Data and training

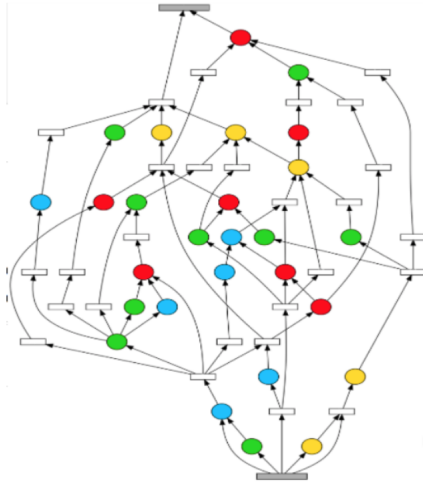
- Loss functions, activation functions, data augmentation, initialization, learning algorithm
- Takes advantage of flexibility of EC

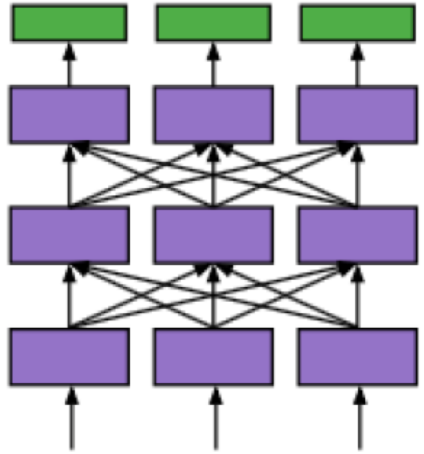
# (A) Evolutionary Neural Architecture Search







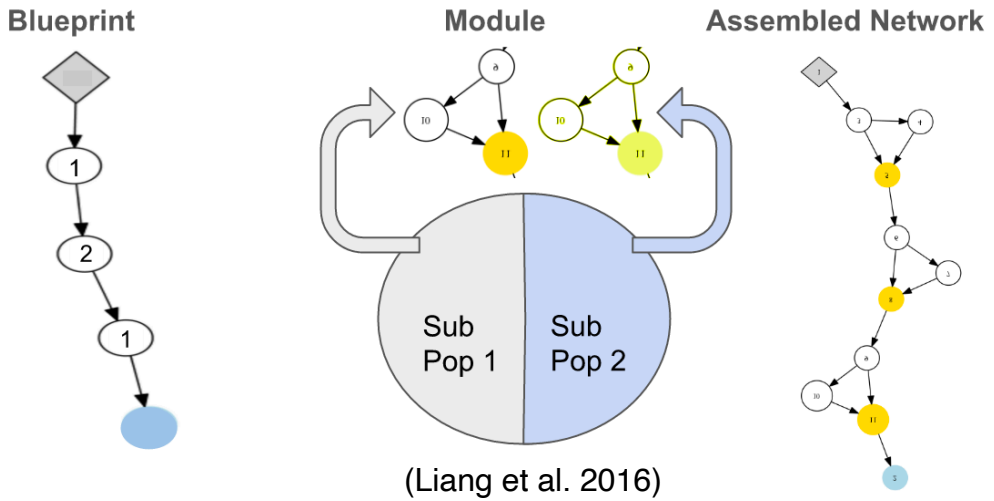
Evolution is a natural fit:

- Population-based search covers the space
- Crossover between structures discovers principles

Moreover,

- Can build on Neuroevolution work since the 1990s:
  partial solutions, complexification, indirect encoding, novelty search
- Applies to continuous values; discrete choices; graph structures; combinations
- Can evolve hyperparameters; nodes; modules; topologies; multiple tasks

# E.G. NAS with CoDeepNEAT




(Liang et al. 2016)

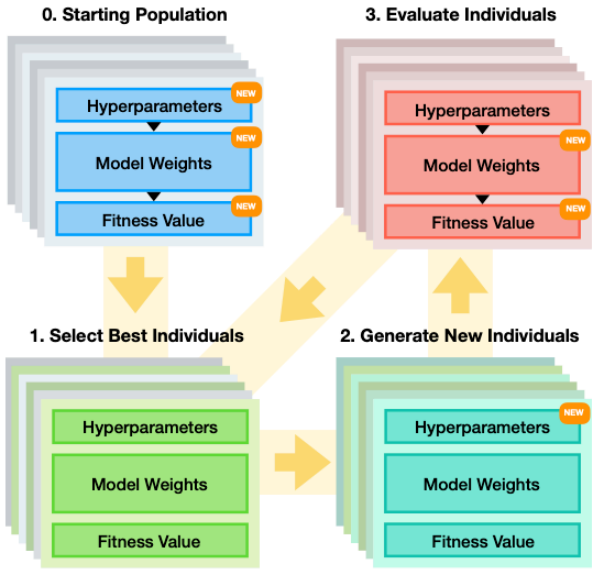
## Evolution at three levels

- Module subpopulations optimize building blocks
- Blueprint population optimizes their combinations
- Hyperparameter evolution optimizes their instantiation

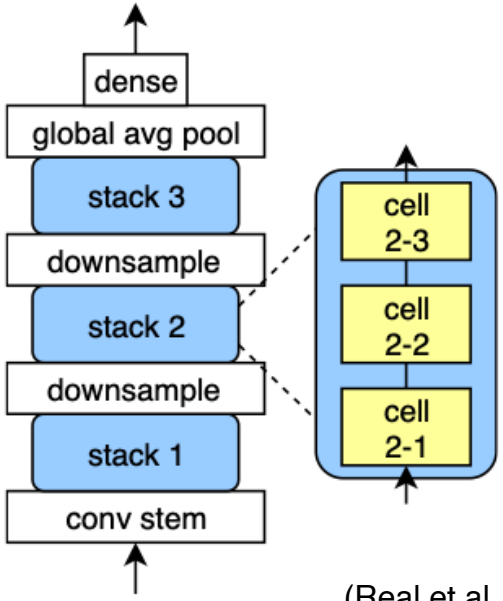
## Fitness of the complete network drives evolution

- Candidates need to be evaluated through training
- Expensive; use partial training, surrogates…

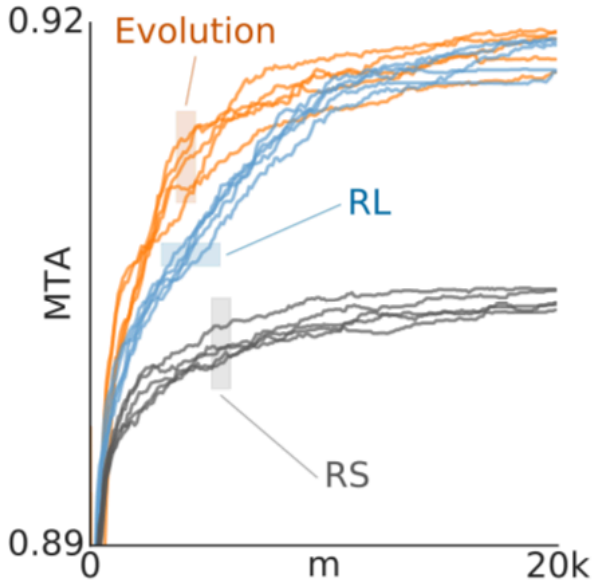
# Making NAS Evaluations Practical




(Liang et al. 2021)




(Real et al. 2019)




Population-based training (Jaderberg et al 2017; Liang et al. 2021)

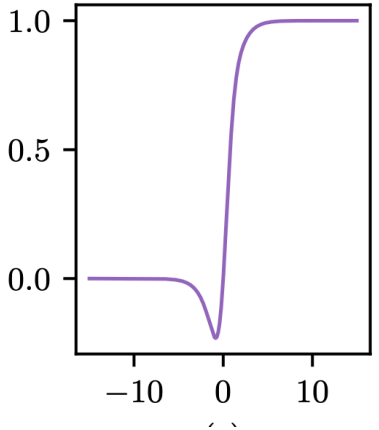
- Continual training and evolution

Scaling and regularization (Such et al. 2017; Real et al. 2019)

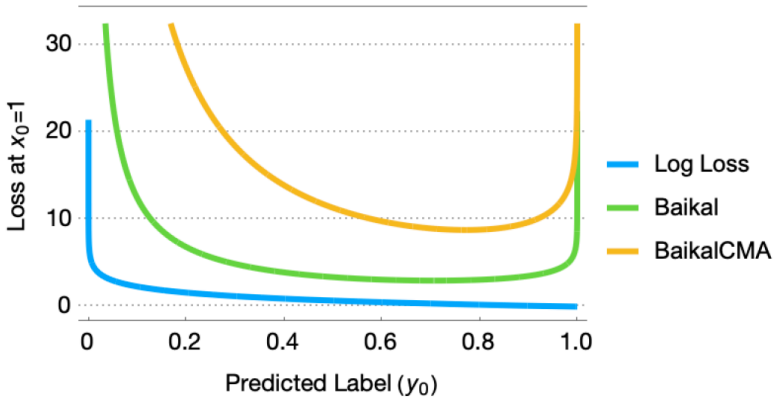
- State-of-the art at the time in CIFAR-10, CIFAR-100, ImageNet

Specialized crossover operators (Qiu and Miikkulainen 2023)

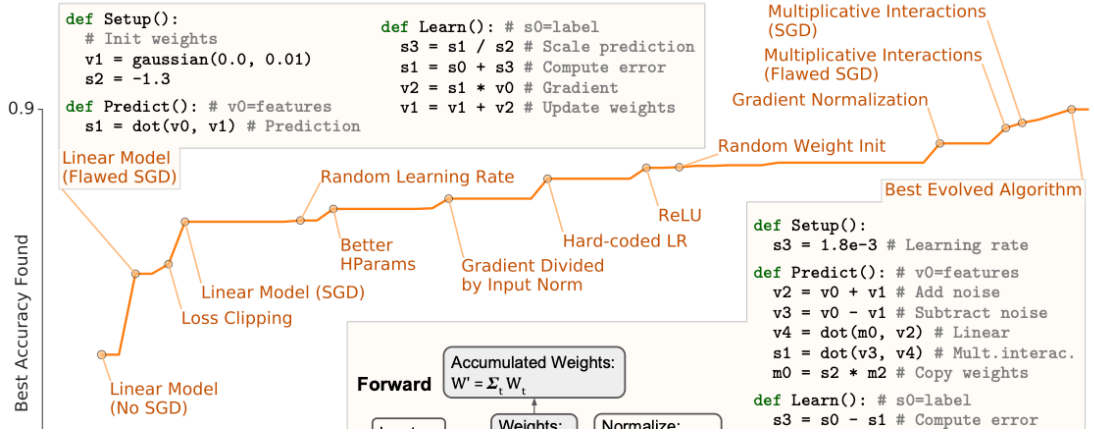
# (B) Optimizing Other Aspects of Deep Learning Design




(Bingham and Miikkulainen, 2022)




(Gonzalez and Miikkulainen, 2020)




(Real et al. 2020)

## Optimizing activation functions and loss functions

(Bingham and Miikkulainen, 2022, IJCNN-25)
(Gonzalez and Miikkulainen, 2020)

- Regularization and refinement

## Designing machine learning algorithms with GP

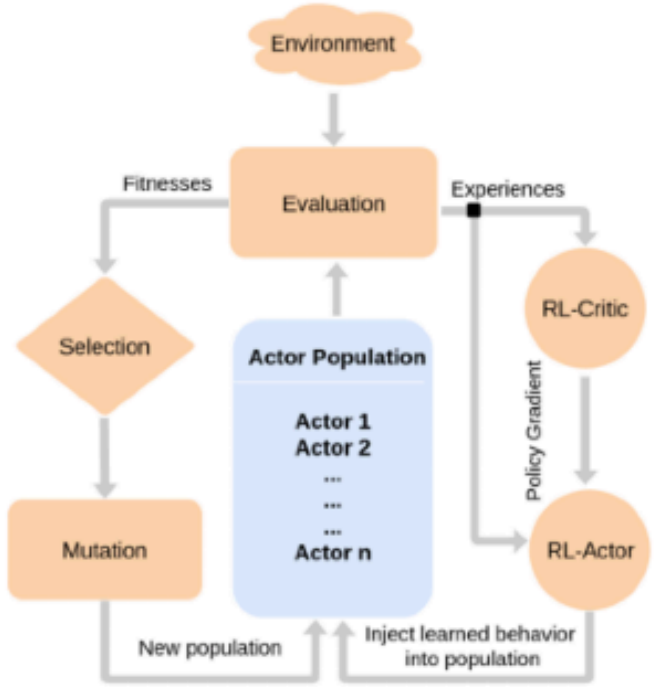
(Real et al. 2020)

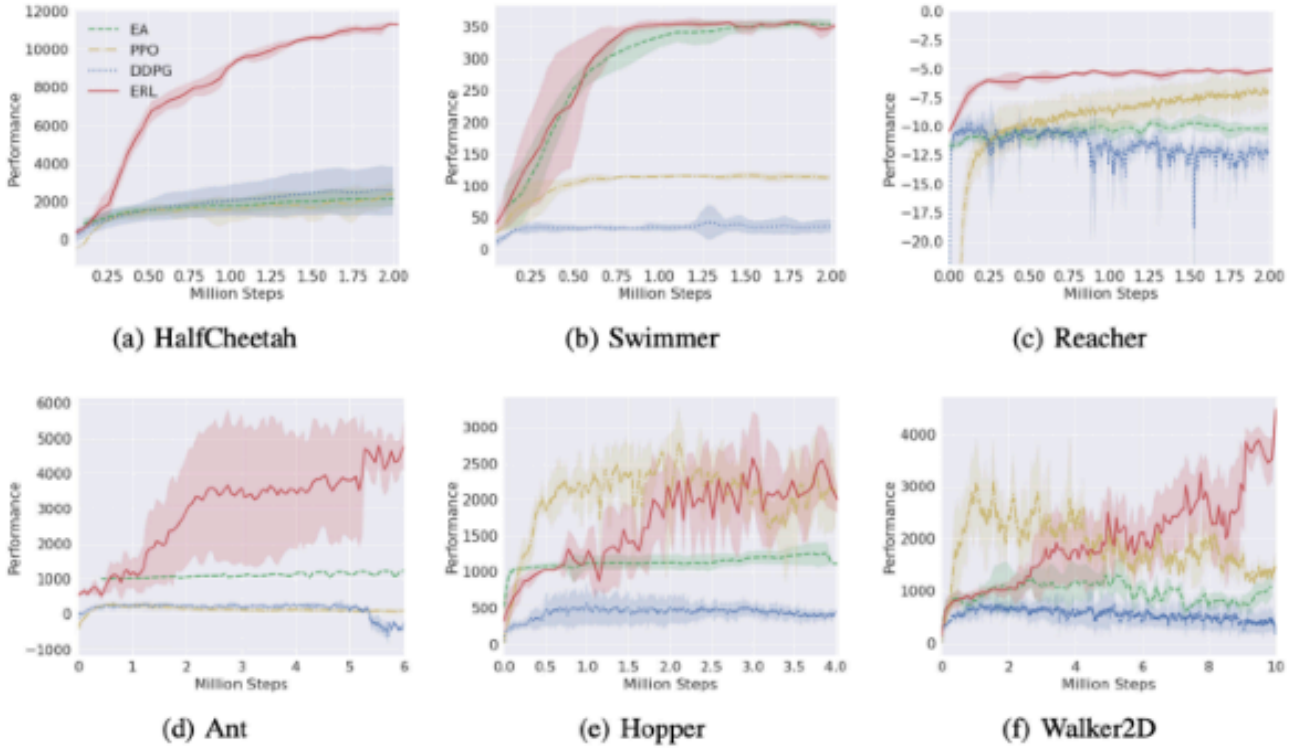
- Adapts to different task types
- Discovering new layer types

## Coevolution of multiple aspects of network design

# Neuroevolution Synergies with RL

## (A) Combining population-based search and RL-based search



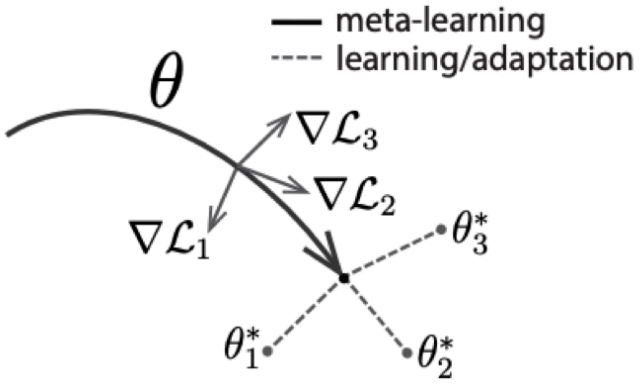




(Khadka and Tumer 2018)

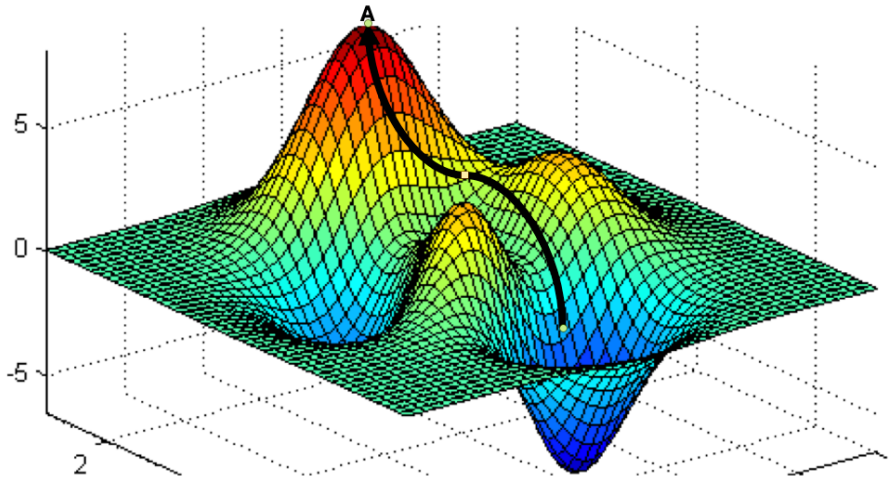
Evolutionary Reinforcement Learning:

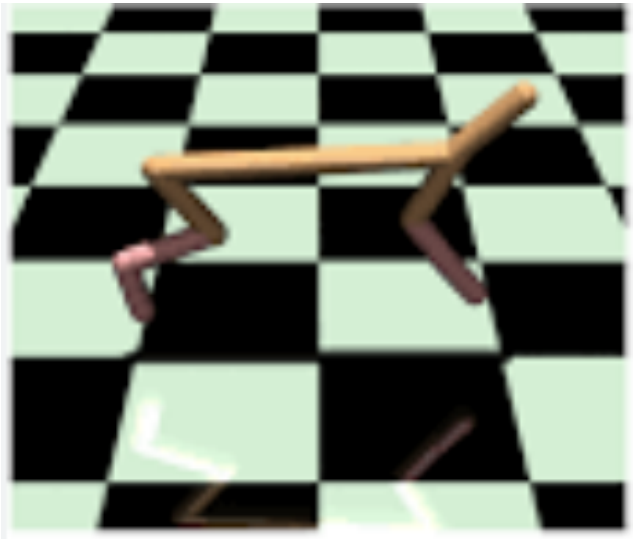
- A population of networks evolved to maximize rewards
- Evaluations create off-policy training data for Deep RL
- Trained networks periodically injected into the population
- ERL outperforms both EA and Deep RL alone

# (B) Evolving Starting Points for RL




(Finn et al. 2017)



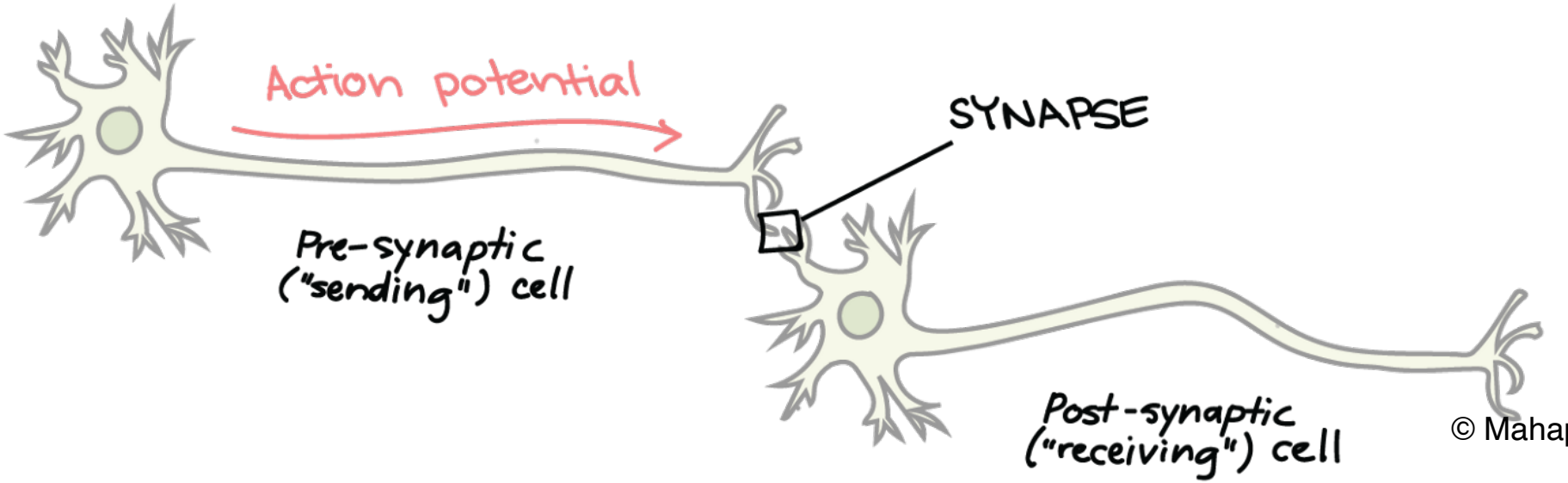
MAML-Baldwin, ES-MAML (Fernando et al. 2018; Song et al. 2019)

- Model-agnostic Meta-Learning (MAML) finds good starting points for learning
- Evolutionary methods like MAML-Baldwin and ES-MAML improve by evolving the starting points
  Evolve initial weights that adapt to different tasks during the agent's lifetime.
- E.g. in half-cheetah task, adapts to changing direction rewards within seconds

# Meta-Learning through Hebbian Plasticity in Random Networks

Najarro & Risi, NeurIPS 2020

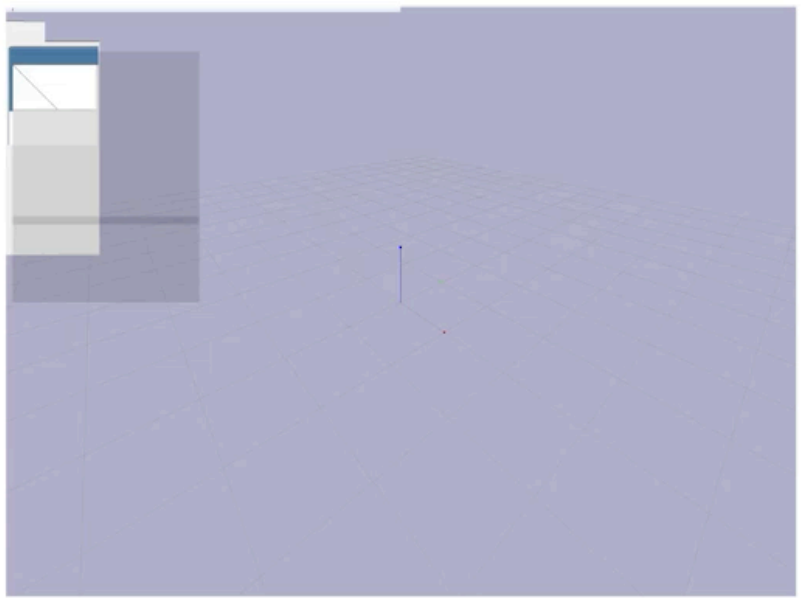
Start network with random weights instead and only evolve local Hebbian learning rules ➔ can weights learn to self-organize?




© Mahapattanakul

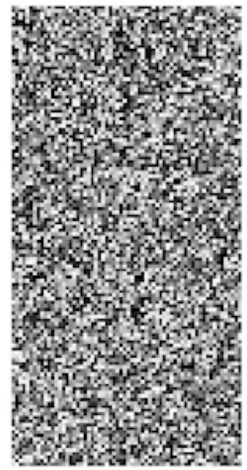
$$\Delta w_{ij} = \eta_w \cdot (A_w o_i o_j + B_w o_i + C_w o_j + D_w)$$

**Hebbian Network**



**Network's dynamical weights**

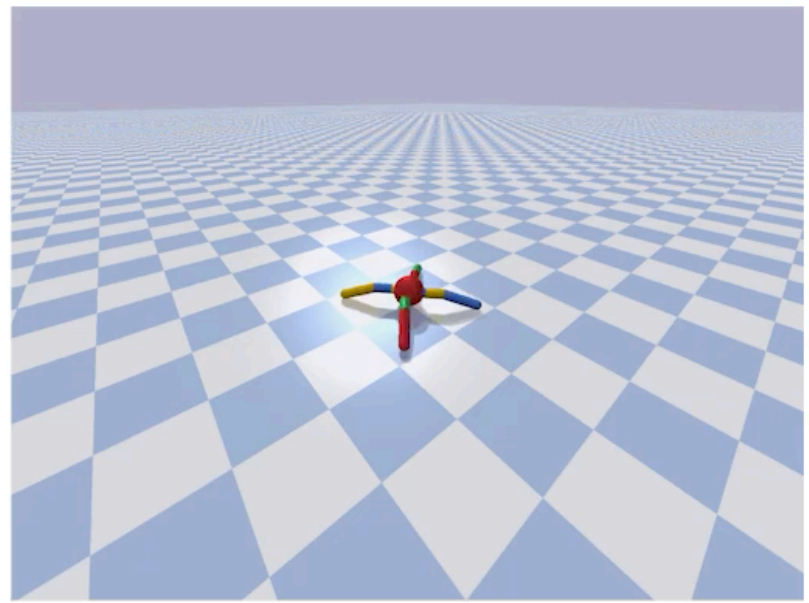
FC layer 1



FC layer 2

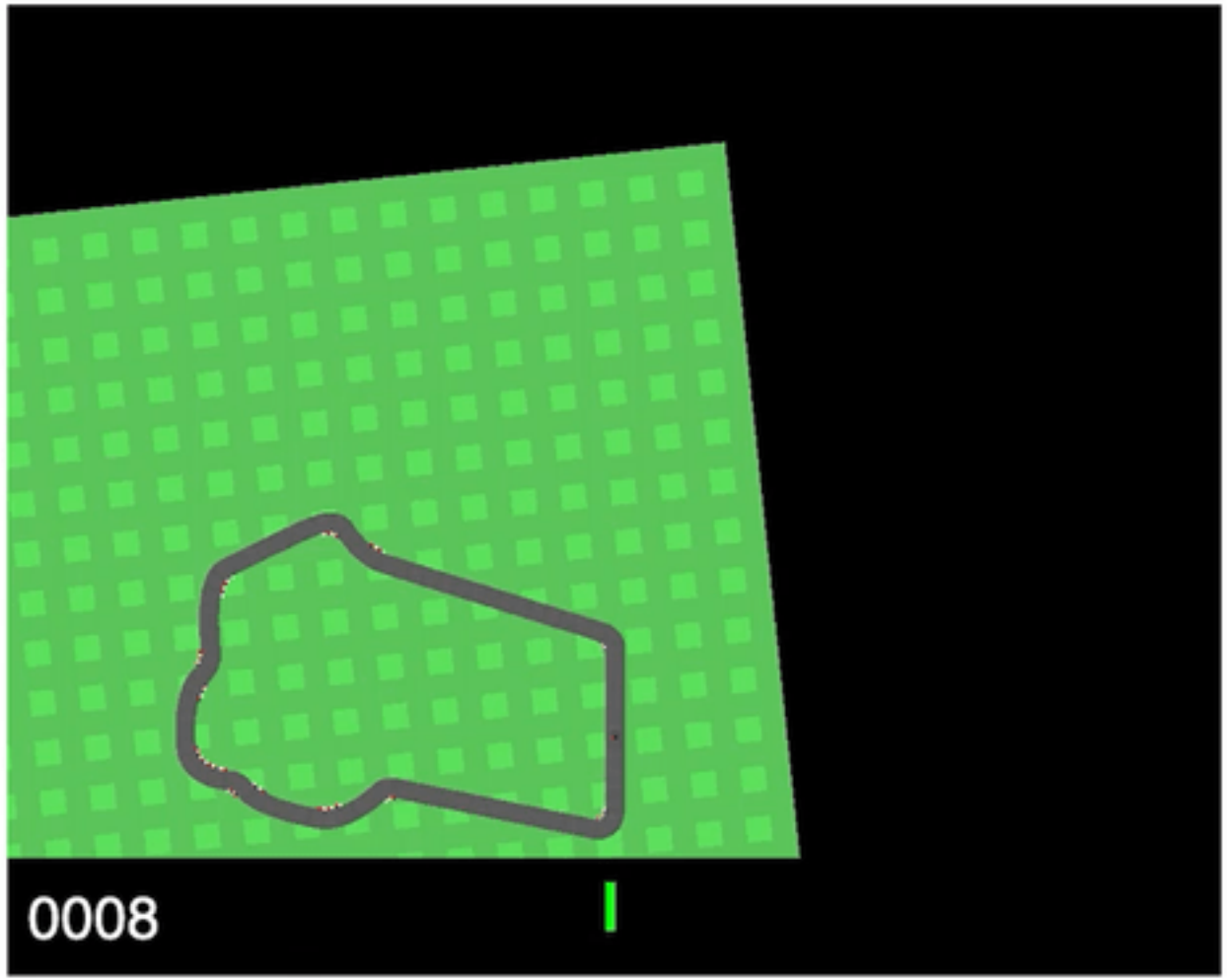
FC layer 3

**Static Network**

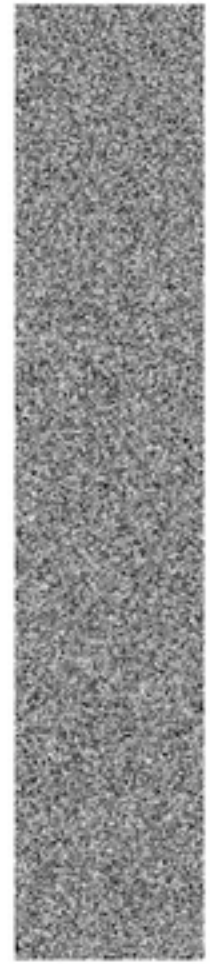


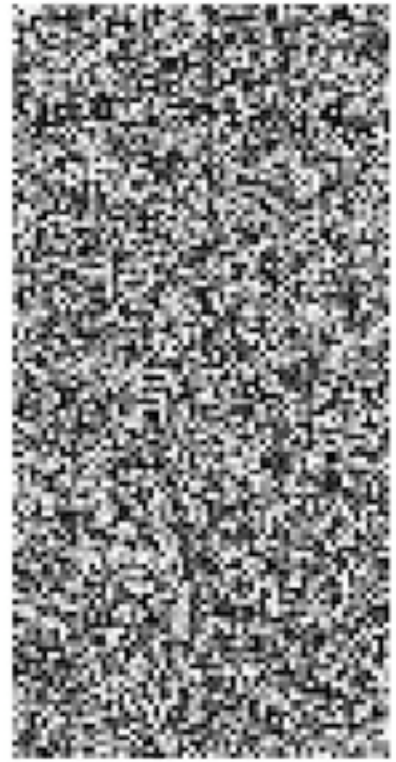
Seen during training

## RL environment

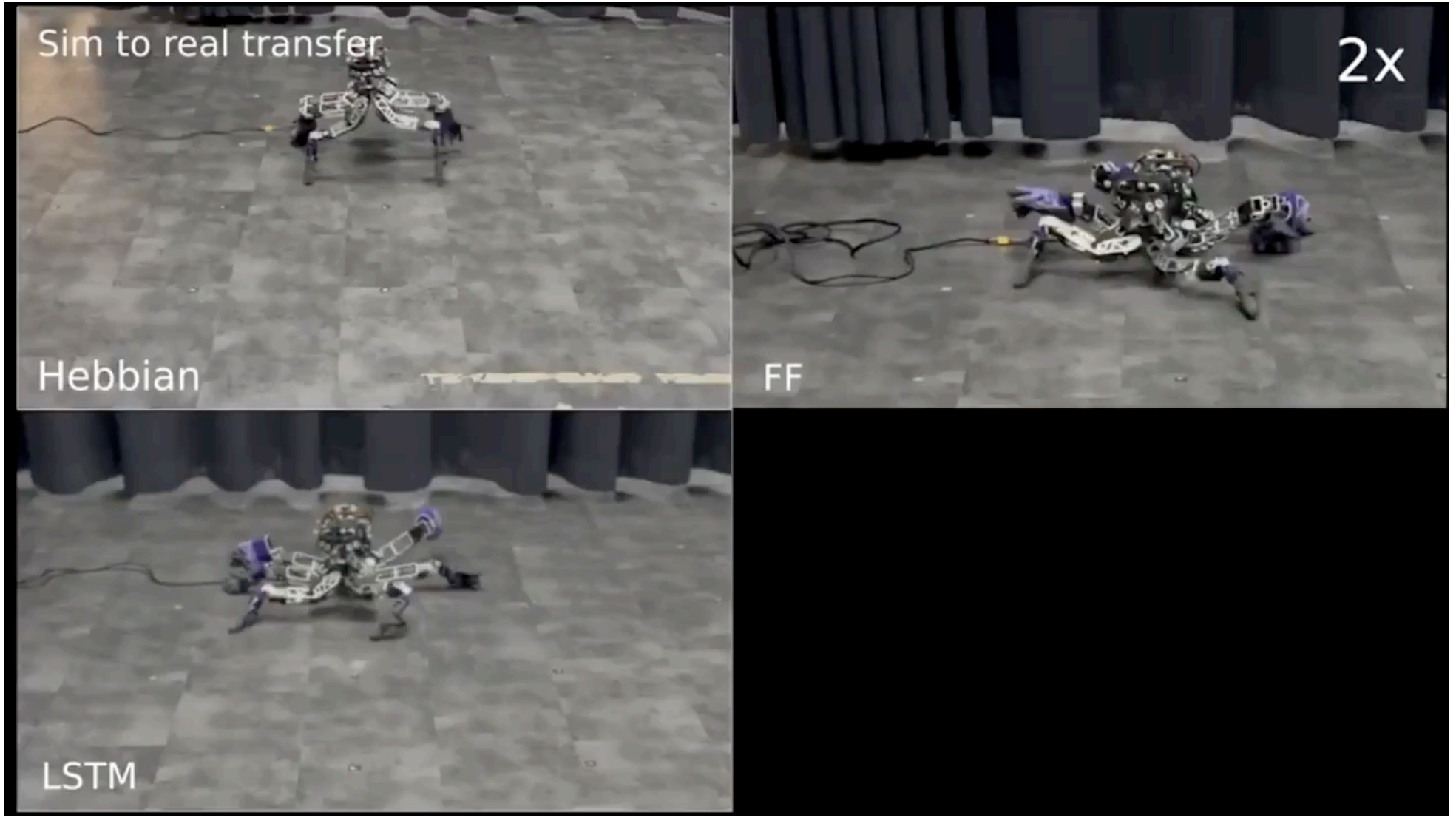



## Network's dynamical weights


FC layer 1


FC layer 2

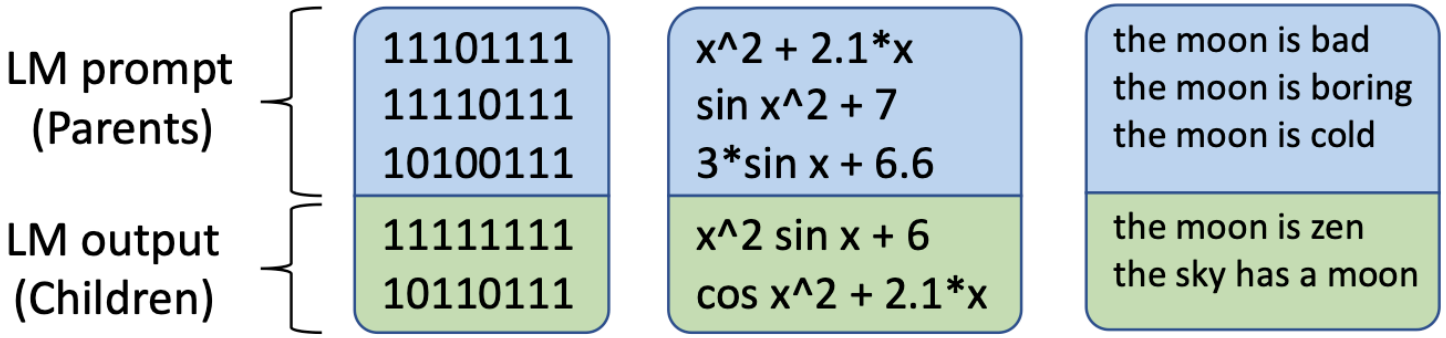

FC layer 3




Lung et al. 2025; IROS

# Neuroevolution Synergies with LLMs

## (A) Neuroevolution through Large Language Models



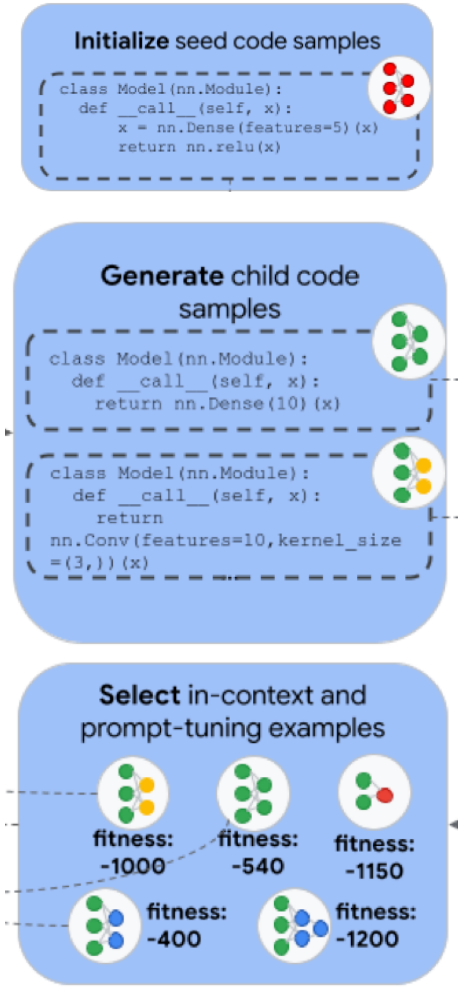

(Meyerson et al. 2024)

### Better evolution through LLMs?

- Evolution through large models (ELM)
- Language model crossover (LMX)
- Level generation for Mario (MarioGPT)

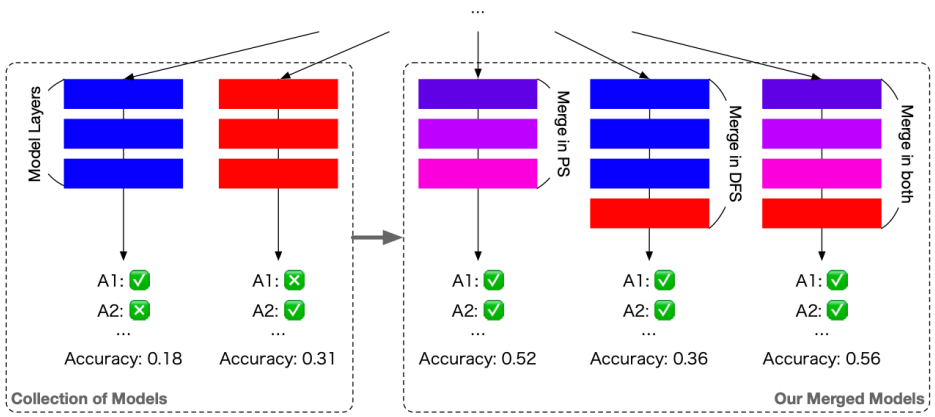
### E.g. Evolutionary prompting for NAS (EvoPrompting)

- Existing architectures as prompts; generate new
- Tune the prompts based on performance

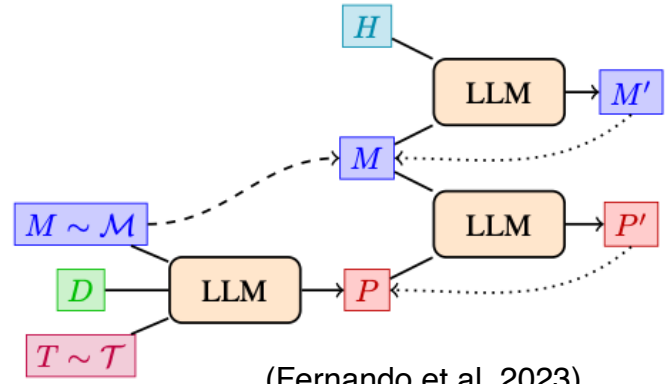



(Chen et al. 2023)

# Neuroevolution of Large Language Models




(Akiba et al. 2024)




(Fernando et al. 2023)

## Better LLMs through evolution?

Model merging: combine multiple fine-tuned LLMs to one

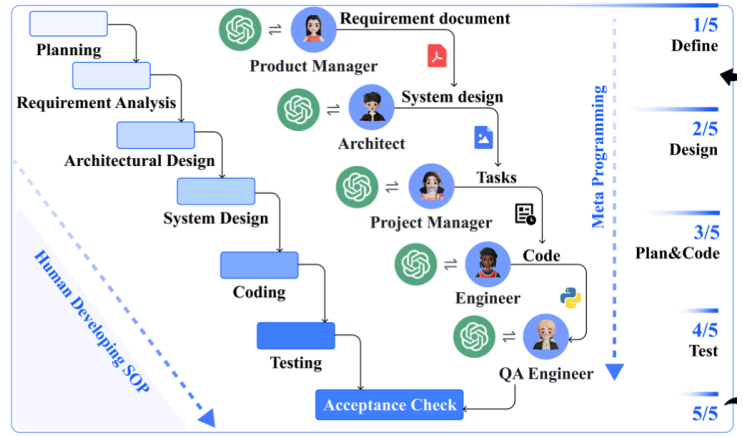
- E.g. Japanese LLM with Math

Evolving prompts: Promptbreeder

- Evolving mutation prompts to improve task prompts

Evolving multi-LLM interactions

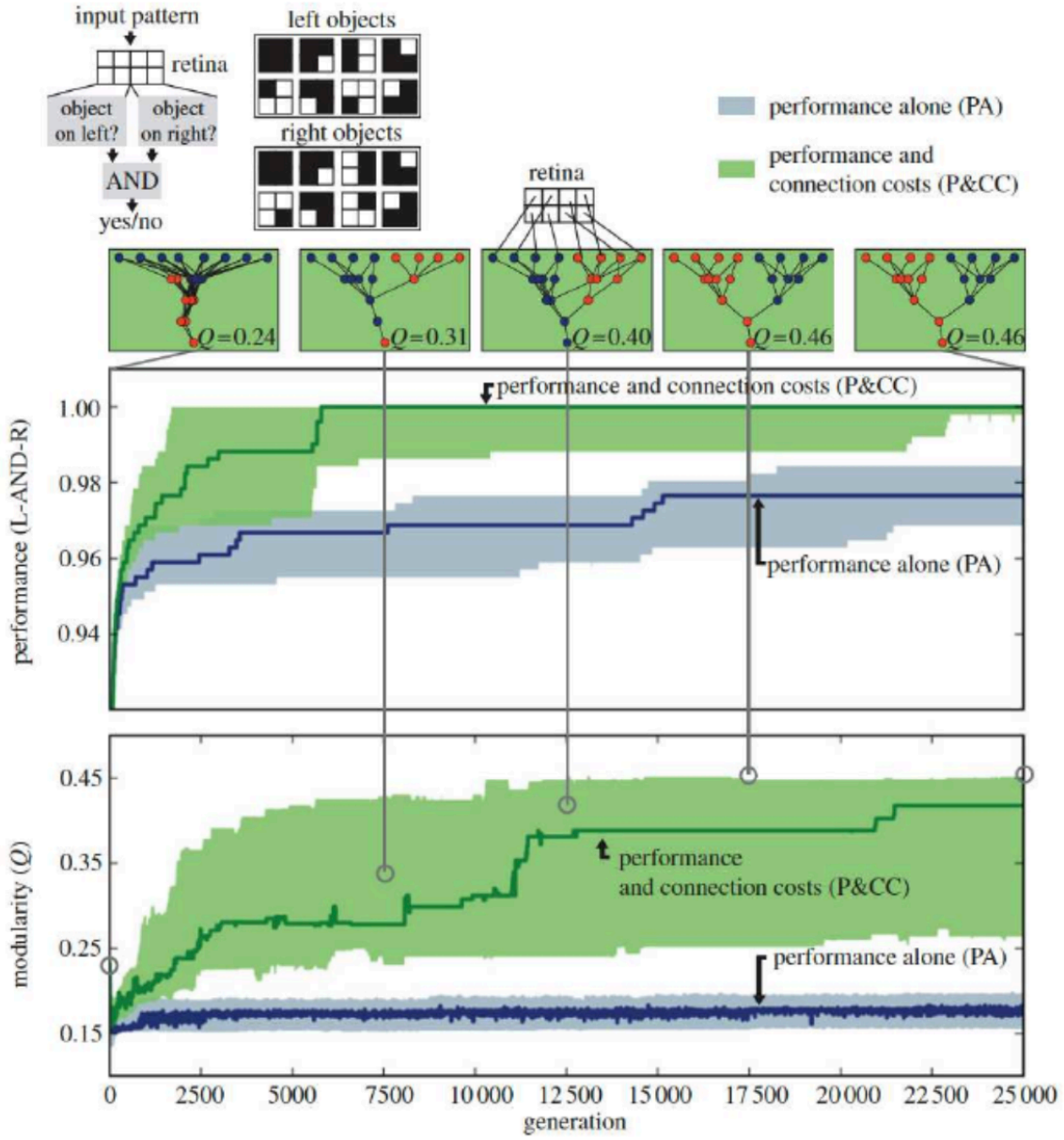
- E.g. roles for collaborative problem solving




(Hong et al. 2023)

# Insights into biology

# Evolution of modularity




Clune, J., Mouret, J. B., & Lipson, H. (2013). The evolutionary origins of modularity. *Proceedings of the Royal Society b: Biological sciences*, *280*(1755), 20122863

# Example: Evolution of Intelligent Coordinated Behavior

## Stealing a kill from lions

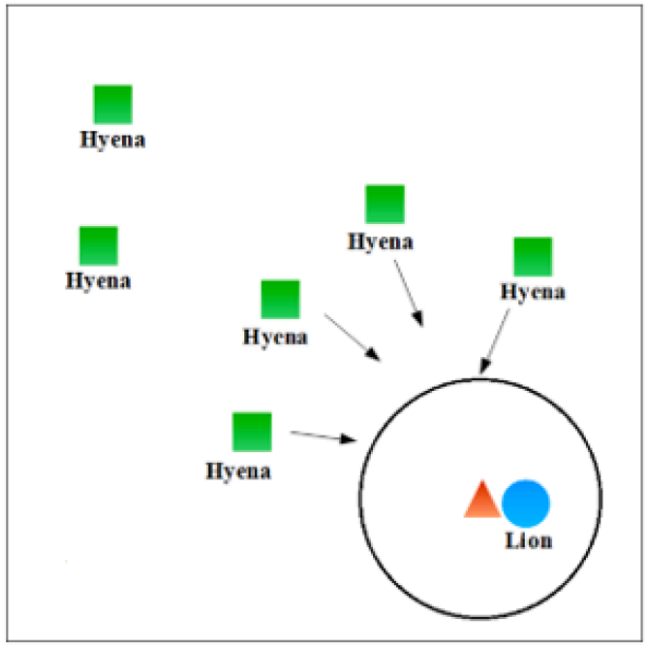
- Succeeds in an otherwise impossible task (sometimes)
- More sophisticated than other hyena behaviors
- Highly rewarding compared to normal hunting
- Largely genetically determined
- A breakthrough in evolution of intelligence?

A collaboration with Kay Holekamp's lab (MSU)

- Studying hyenas in Masai Mara since 1982

(Rajagopalan et al. 2021)

# Simulation Setup



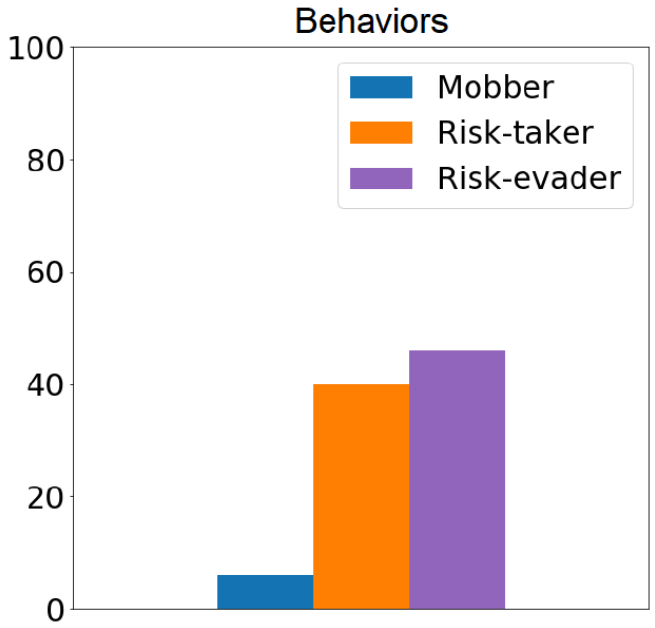

Lion at a kill, with an interaction circle around it
Ten hyenas chosen and placed randomly in the field
If four or more hyenas enter the circle simultaneously, they get the kill

- Otherwise they die

Does mobbing behavior evolve?

- What are the stepping stones for it?

# Initial Behaviors



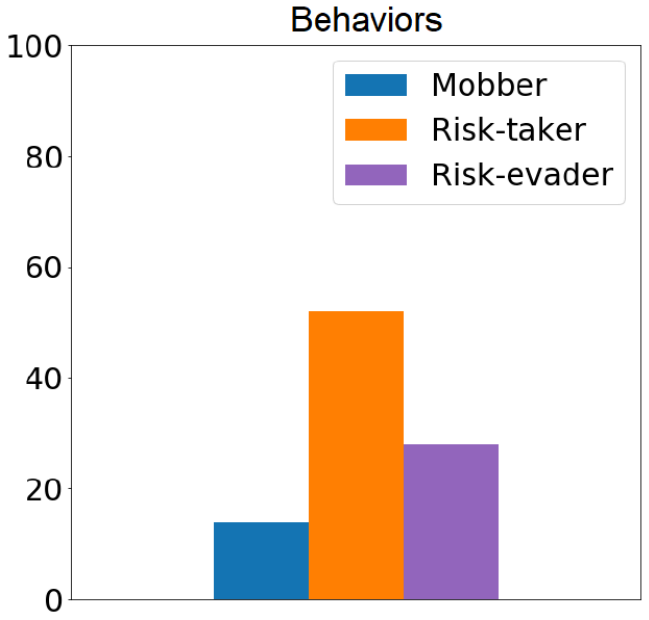

Risk evasion is common

- Never reach the circle; Medium fitness

Risk taking is common

- Charge the circle; Frequent low fitness
- Occasional high fitness by accident

# Early Behaviors




Risk taking grows

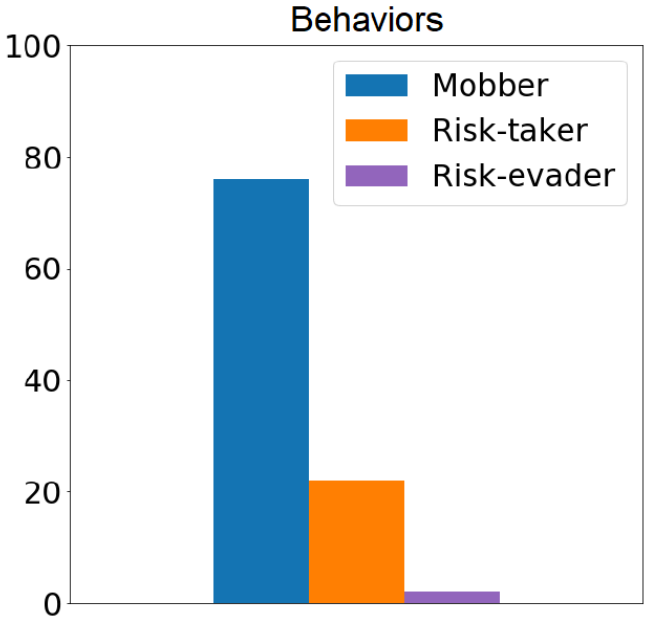
- As long as it is successful often enough

Risk evasion also persists

Evasion at the circle starts to emerge

Is mostly detrimental, but an important stepping stone

# Later Behaviors



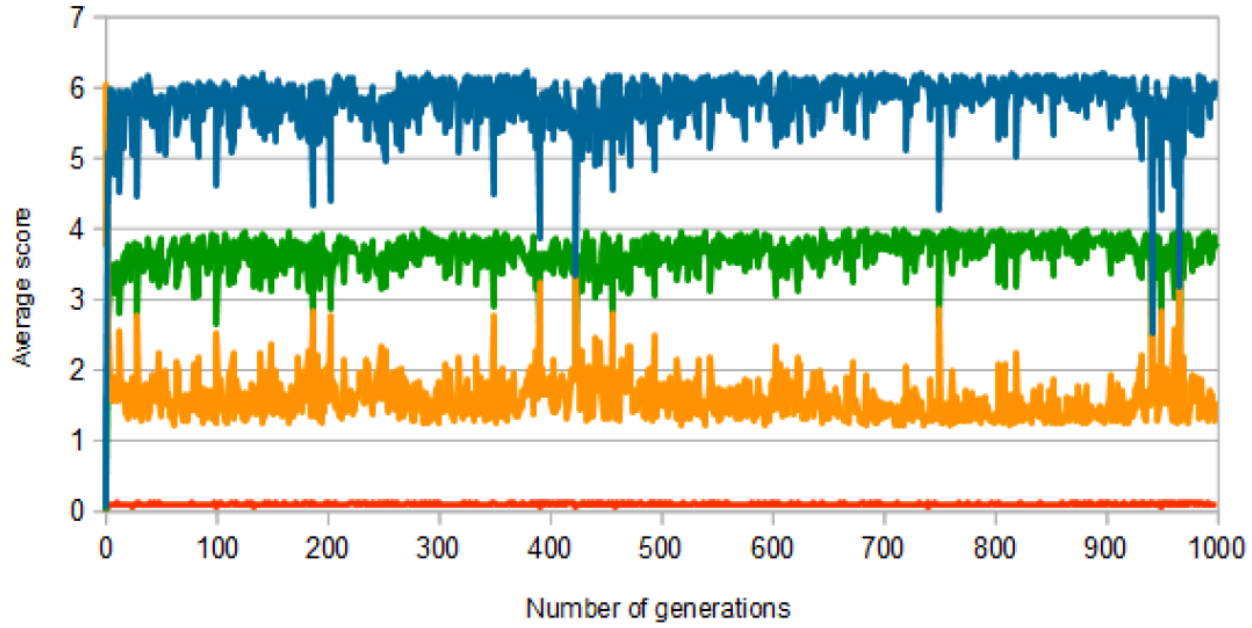

Mobbing emerges

- Not just coincidence of risk takers
- Hyenas wait until there's enough of them

Risk-evaders evolve into latecomers

Simple risk-taking and risk-evasion still exist

# These Behaviors Persist in Prolonged Evolution




Risk taking and risk evasion never go away completely

- They serve a role in maintaining the mobbing behavior
- If mobbing starts to get lost, it can be reintroduced

# Insight into Real-life Behaviors

These behaviors are observed in real-life hyenas as well

A computational explanation of why they are there:

- Stepping stones in discovery
- Safeguards in maintaining

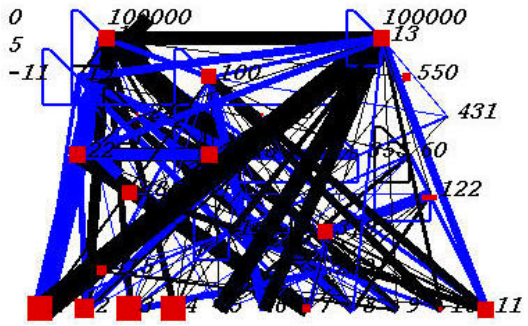
# Future Challenge: Evolution of Language

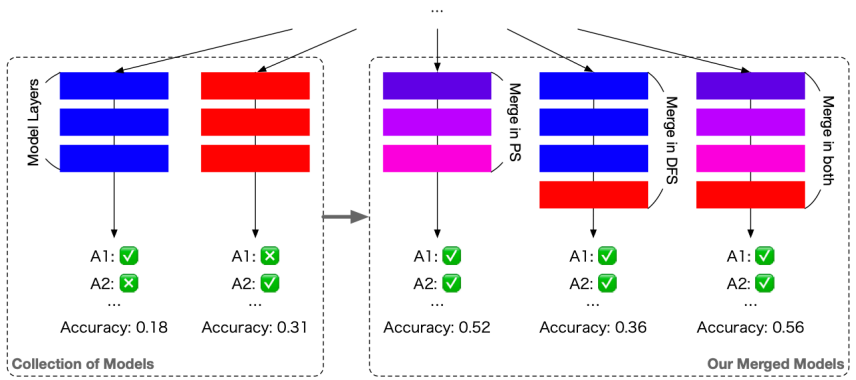

Signaling is possible to evolve in ecological simulations
Structured language is much harder
Perhaps language evolved not from signaling, but cognition

- Complex social structure, with modifiable roles
- Language structure can reuse the same conceptual structures

Enough compute, complex simulations to study now?

# Conclusion




AI is progressing from imitation to creativity; from models to agents

Neuroevolution is a powerful approach to discovering behavior

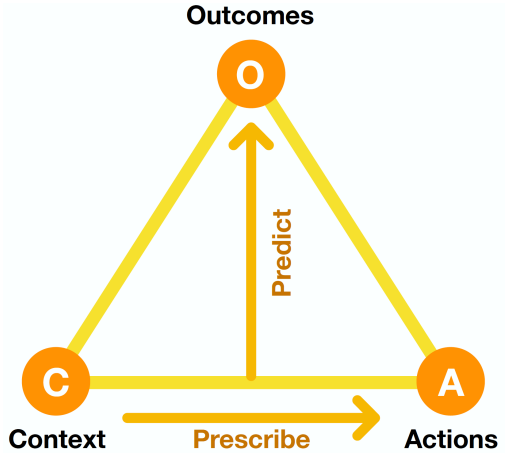
- Control, strategy, collective behavior, decision-making

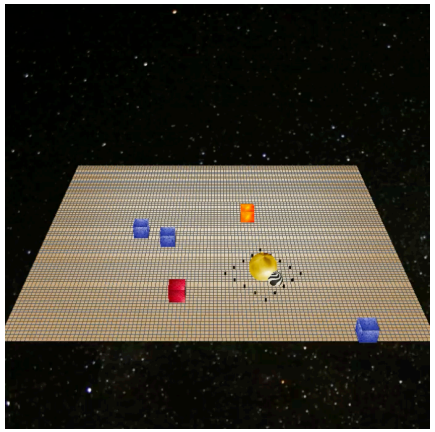
Neuroevolution can provide a boost to ML

- Deep learning designs; RL exploration; LLM optimization
- Automatic design of learning machines

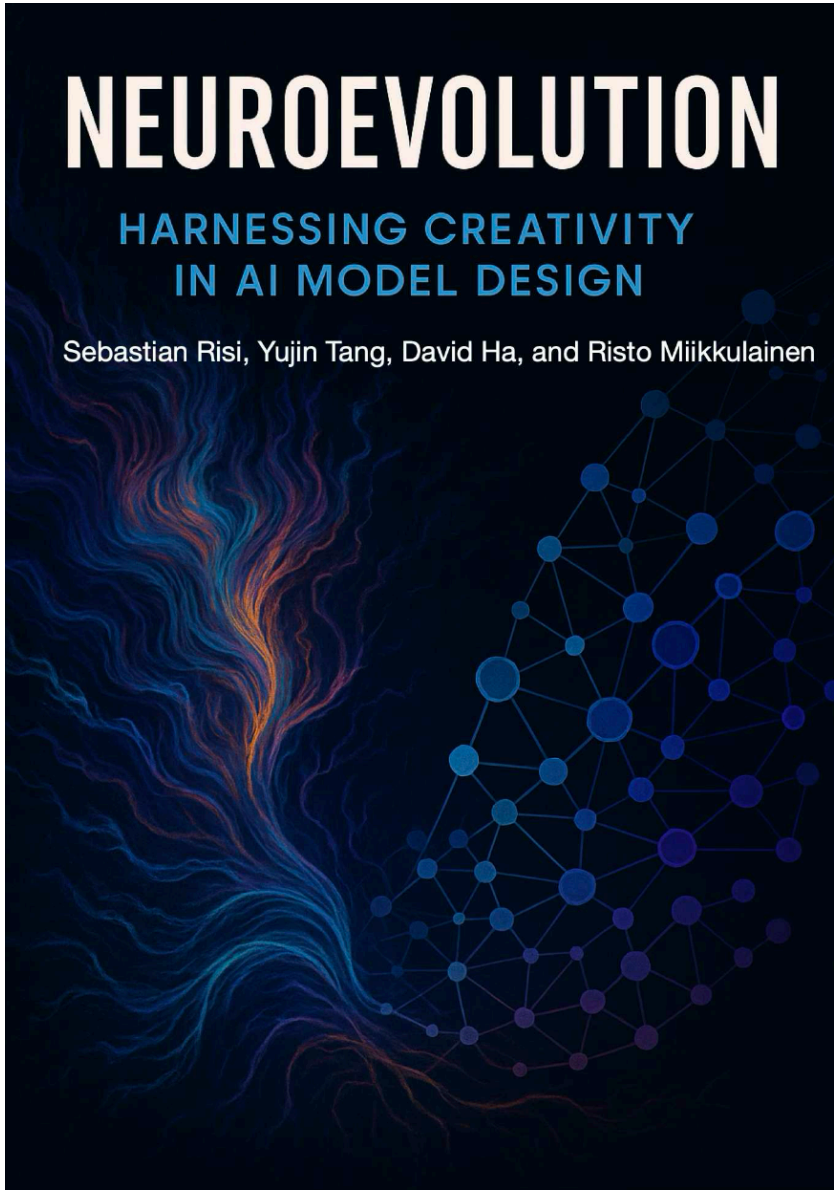
Neuroevolution can provide insight into biological evolution

- Evolutionary origins of circuits, behavior, intelligence
- Evolution of language as a current challenge
- A possible path to more capable artificial agents



# The Neuroevolution Book!




- MIT Press, 2026
- A comprehensive overview
- Software platform: Demos, exercises
- Open access

- https://neuroevolutionbook.com