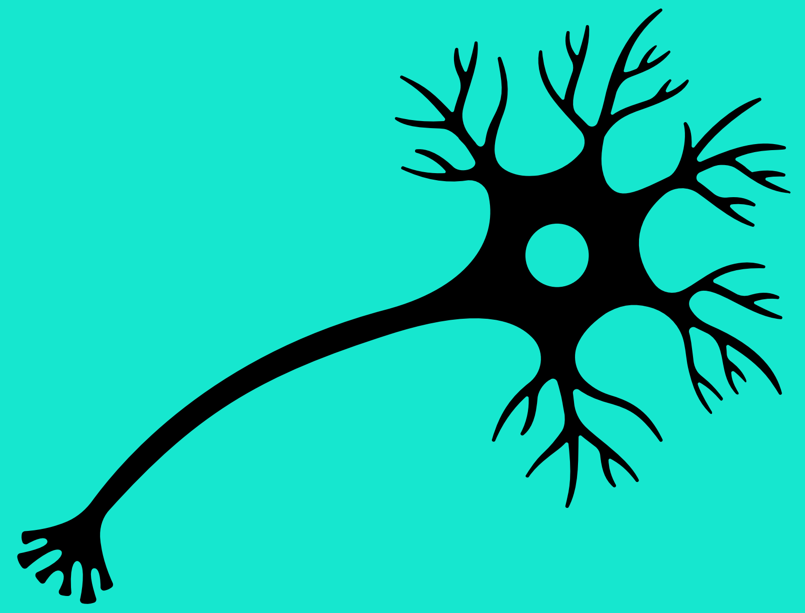
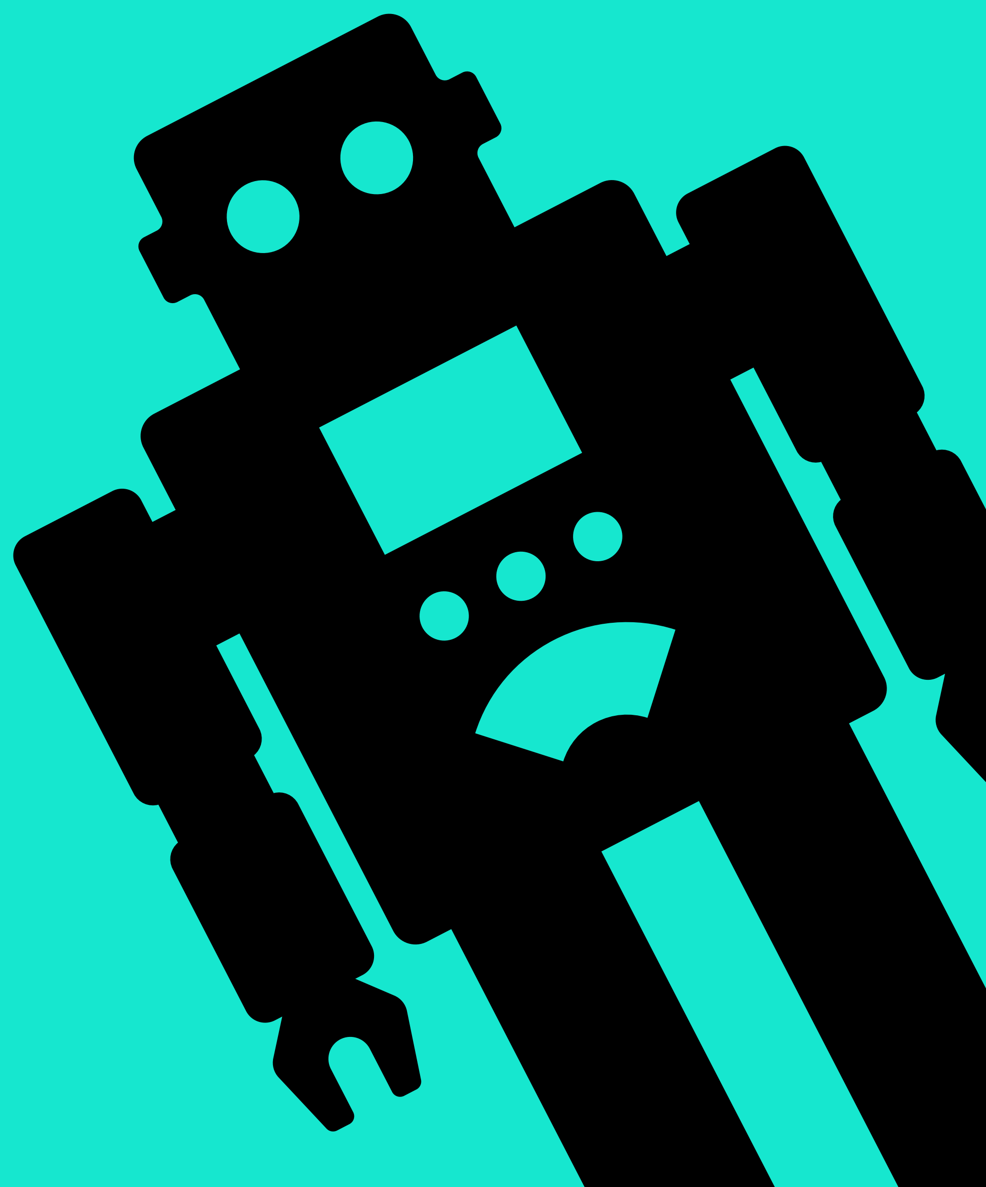


Co-evolution of Robots

Optimization for AI exam

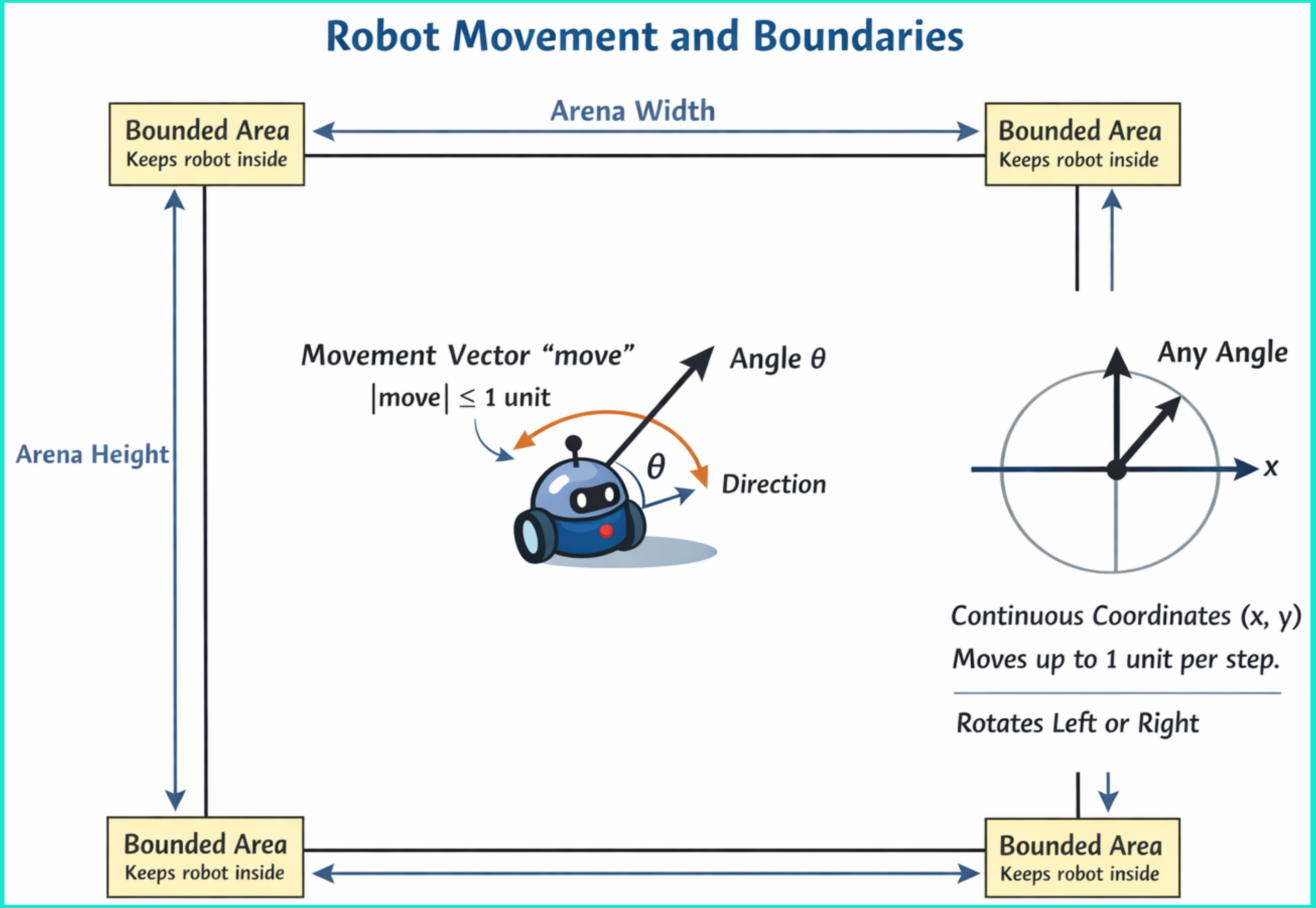
https://github.com/Gabriele-tomai00/Co-evolution_of_robots



Gabriele Tomai 27/01/2026

Simulation environment

Like “robot wars”,
two robots are fighting
against each other and the
winner is the surviving one



Main classes (1/3)

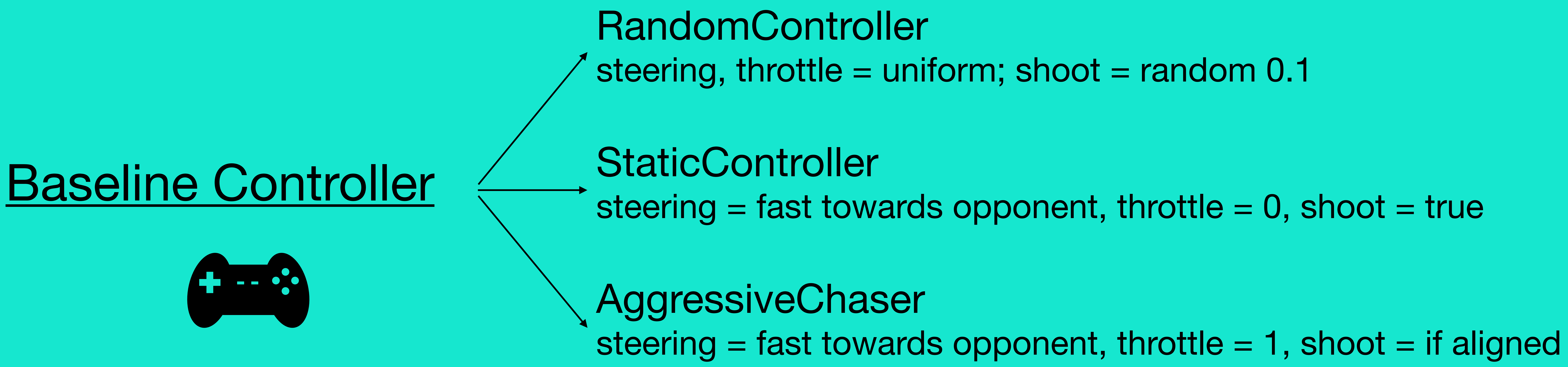
Arena

- Width
- Hight
- Robots
- Max steps
- Current step
- Demage information

Robot

- Position
- Health
- Demage inflicted
- Controller

Main classes (2/3)



<u>Baseline Controller</u>

RandomController

steering, throttle = uniform; shoot = random 0.1

StaticController

steering = fast towards opponent, throttle = 0, shoot = true

AggressiveChaser

steering = fast towards opponent, throttle = 1, shoot = if aligned

- Steering
- Throttle
- Shoot
- get(robot, arena)
- get_opponent(robot, arena)

Main classes (3/3)

Sensor

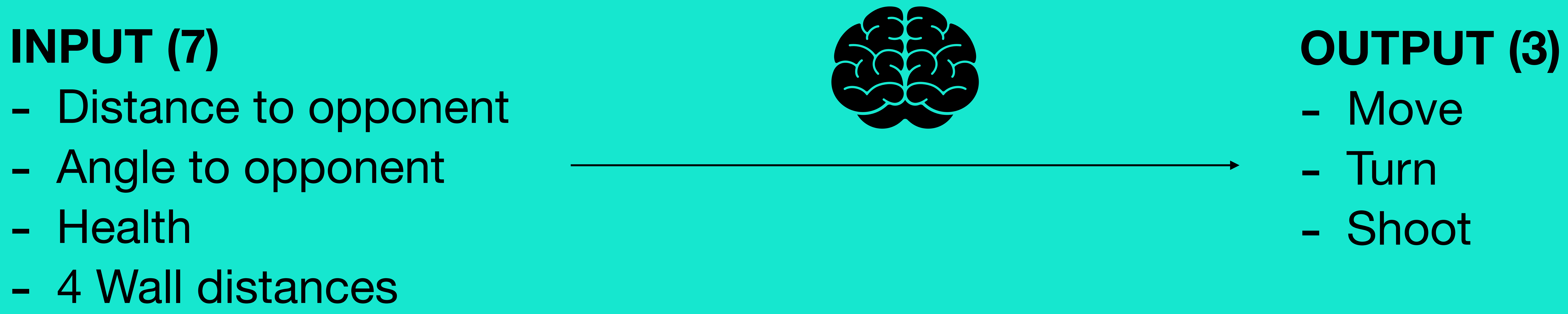
- get(robot, arena)
- get_opponent(robot, arena)

neat.nn.FeedForwardNetwork

- Library class
- The brain of the evolve network
- Receive 7 inputs
- Produces 3 outputs: raw float values

Sensor and NN

Valued of sensor
are normalized



INPUT (7)

- Distance to opponent
- Angle to opponent
- Health
- 4 Wall distances

OUTPUT (3)

- Move
- Turn
- Shoot

Fitness calculation

Hybrid Fitness (Internal + External)

One-population competitive coevolution

Fitness function →

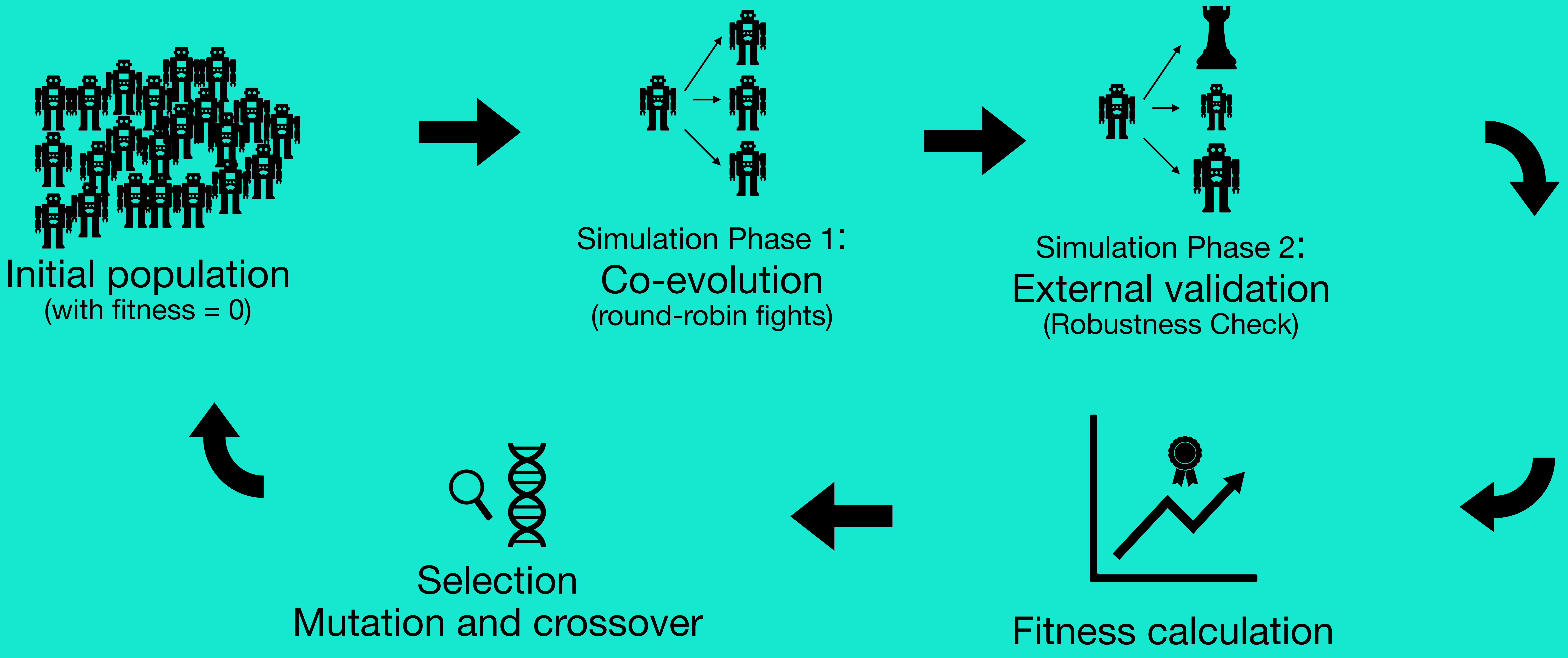
- +100 in case of victory
- +10 in case of aggression
- +0.1 to every time step survived (up to 30 points)

Designed to reward three main behaviors:

- Effectiveness (Victory)
- Aggression
- Survival

Evolutional NEAT cycle

For N generations



Initial population
(with fitness = 0)

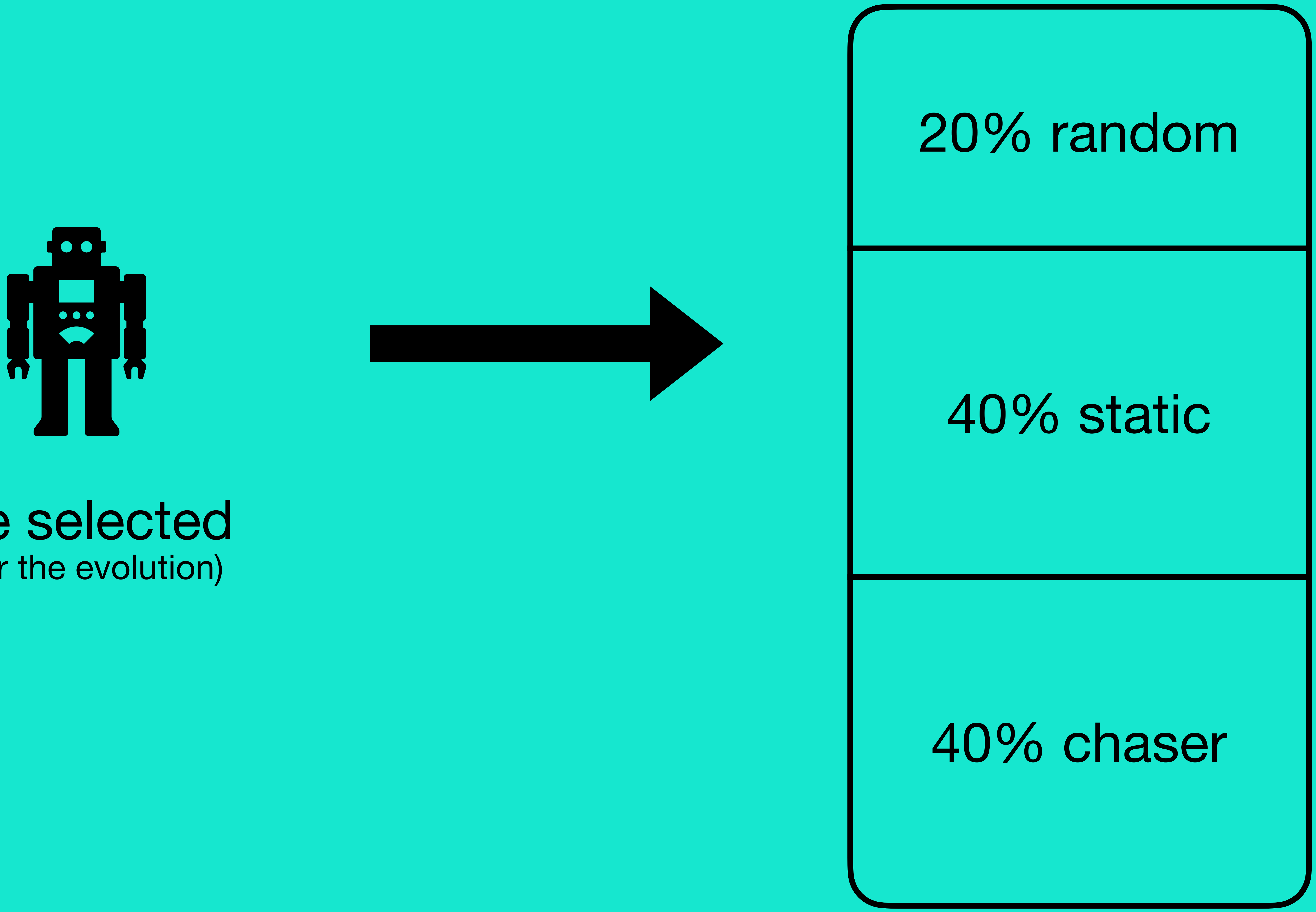
Simulation Phase 1:
Co-evolution
(round-robin fights)

Simulation Phase 2:
External validation
(Robustness Check)

Fitness calculation

Selection
Mutation and crossover

Final test



The selected
(after the evolution)

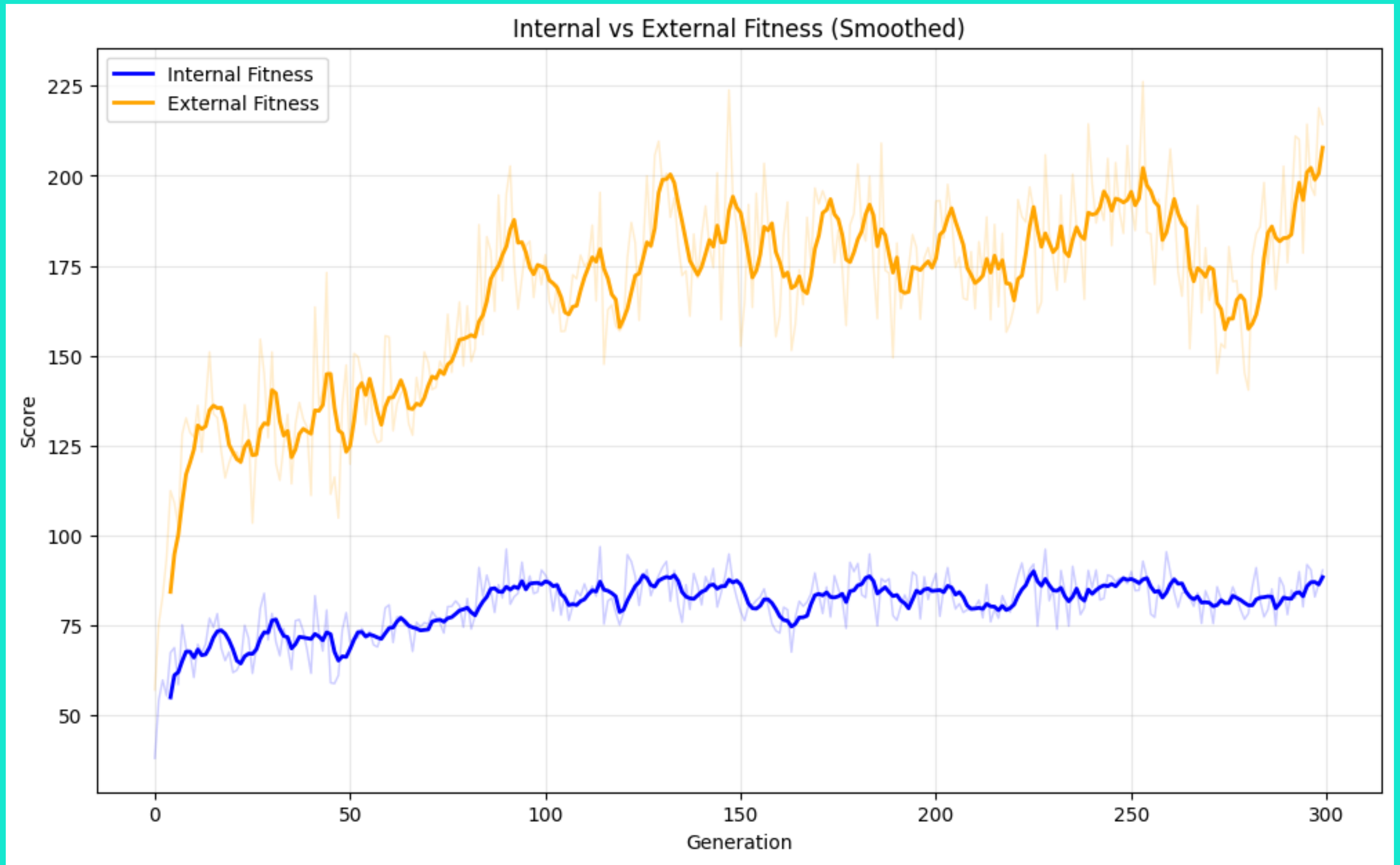
Random robots
(they don’t change during the evolution)

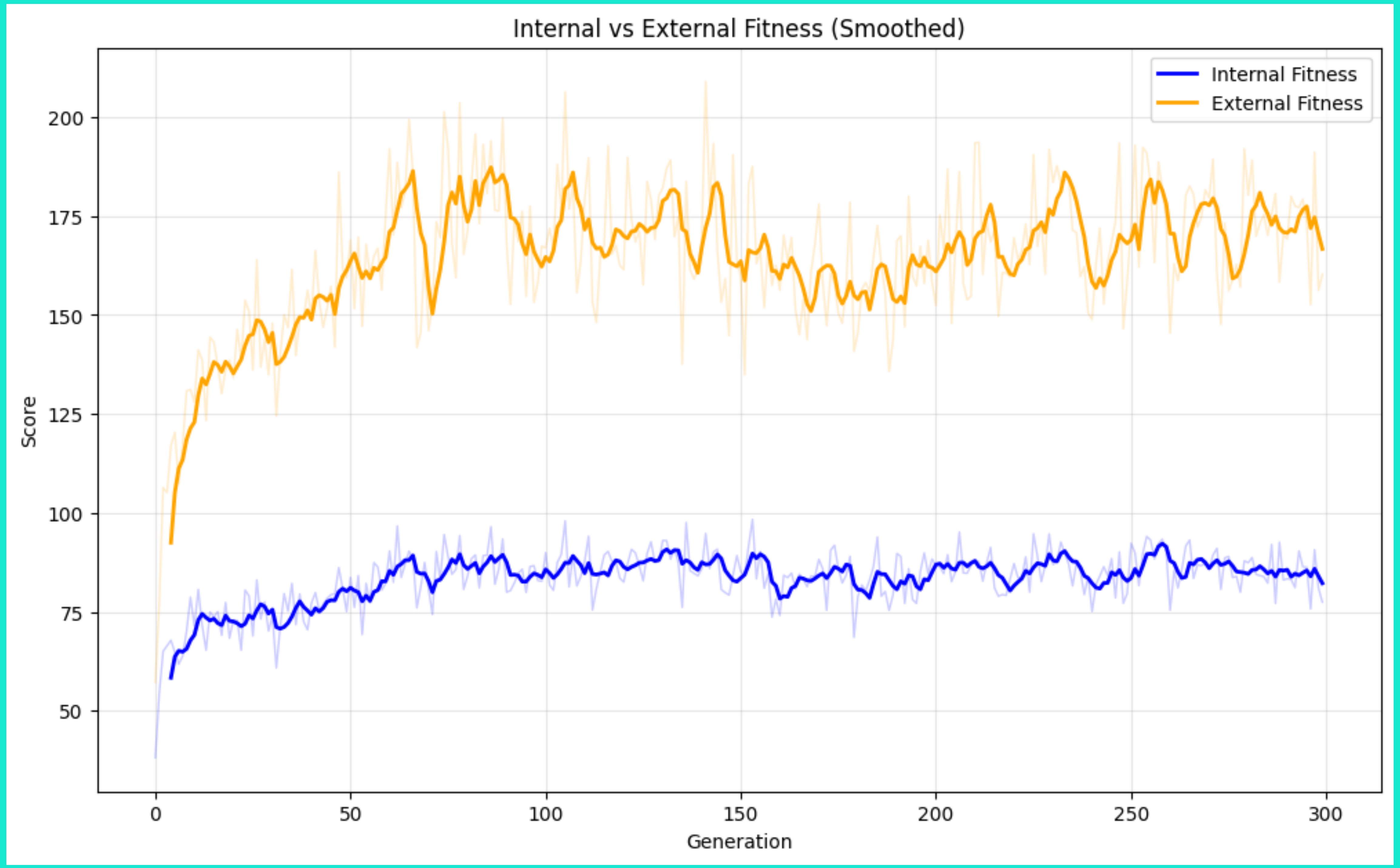
20% random

40% static

40% chaser

Some results (1/2)





- Population: 50
- Generations: 300
- Elitism: 2

- Population: 50
- Generations: 300
- Elitism: 4