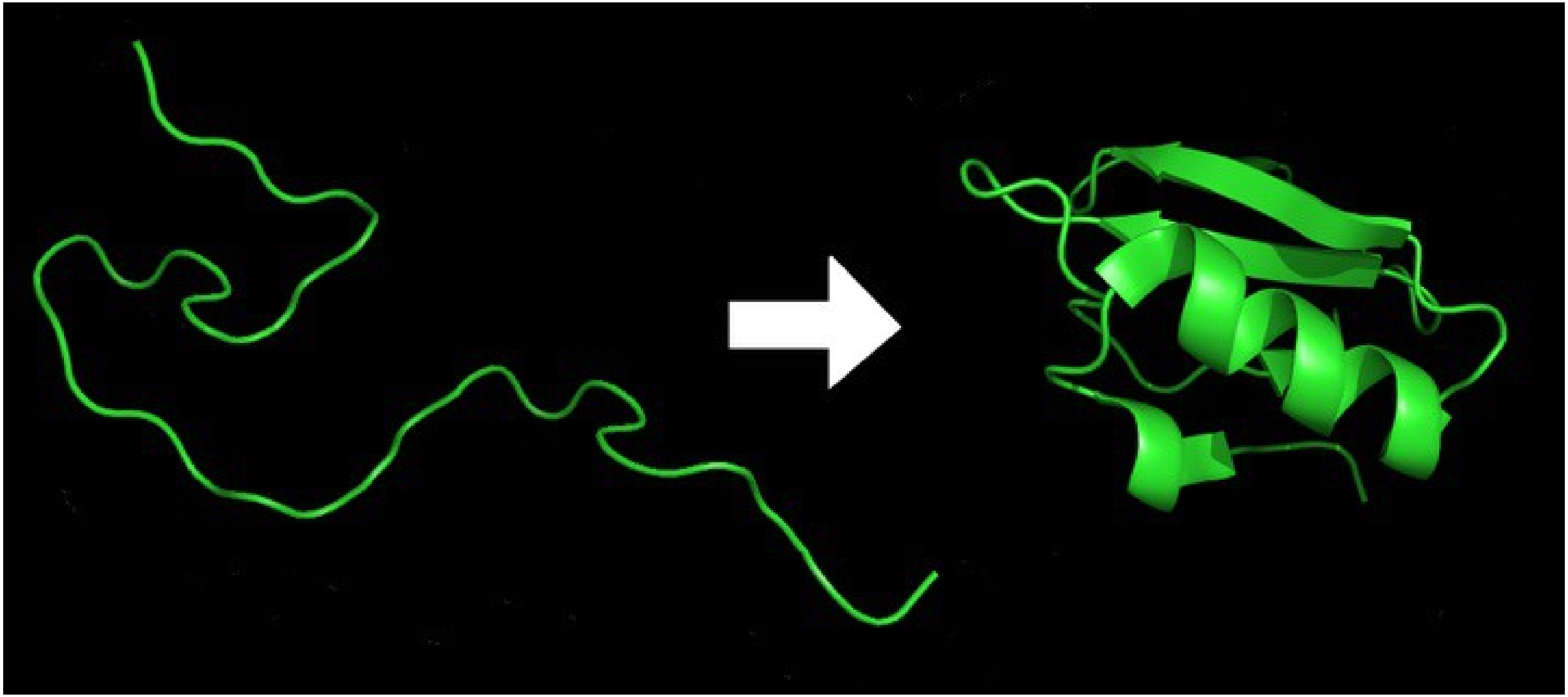
Computer Science and Biology

Protein Folding



Primary structure

(linear sequence of amino acids)

Tertiary structure

(3-dimensional shape)

AlphaFold

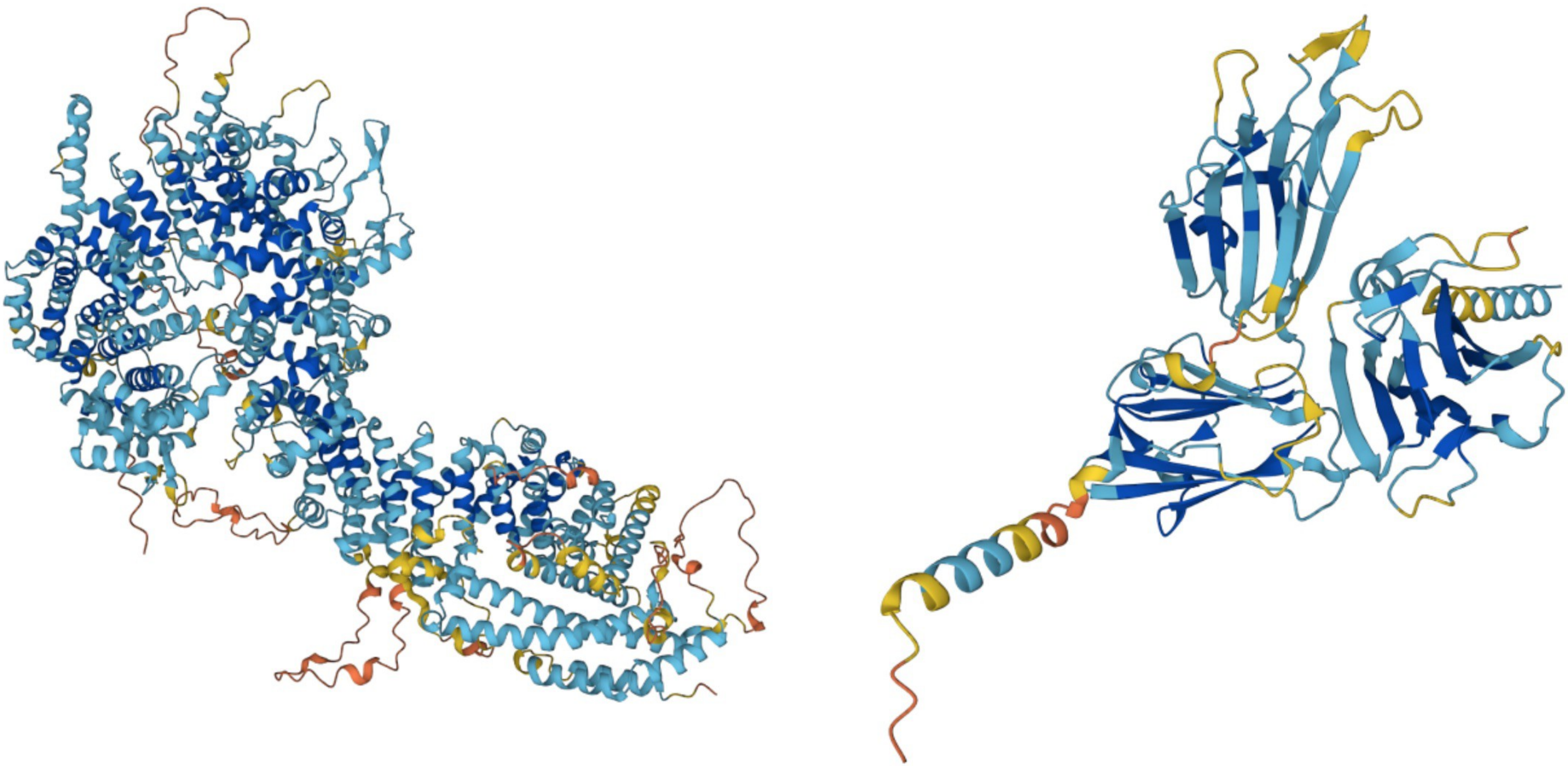
- Deep neural network for predicting protein structure

- AlphaFold 1 placed #1 in CASP (Critical Assessment of Structure Prediction) competition in 2018

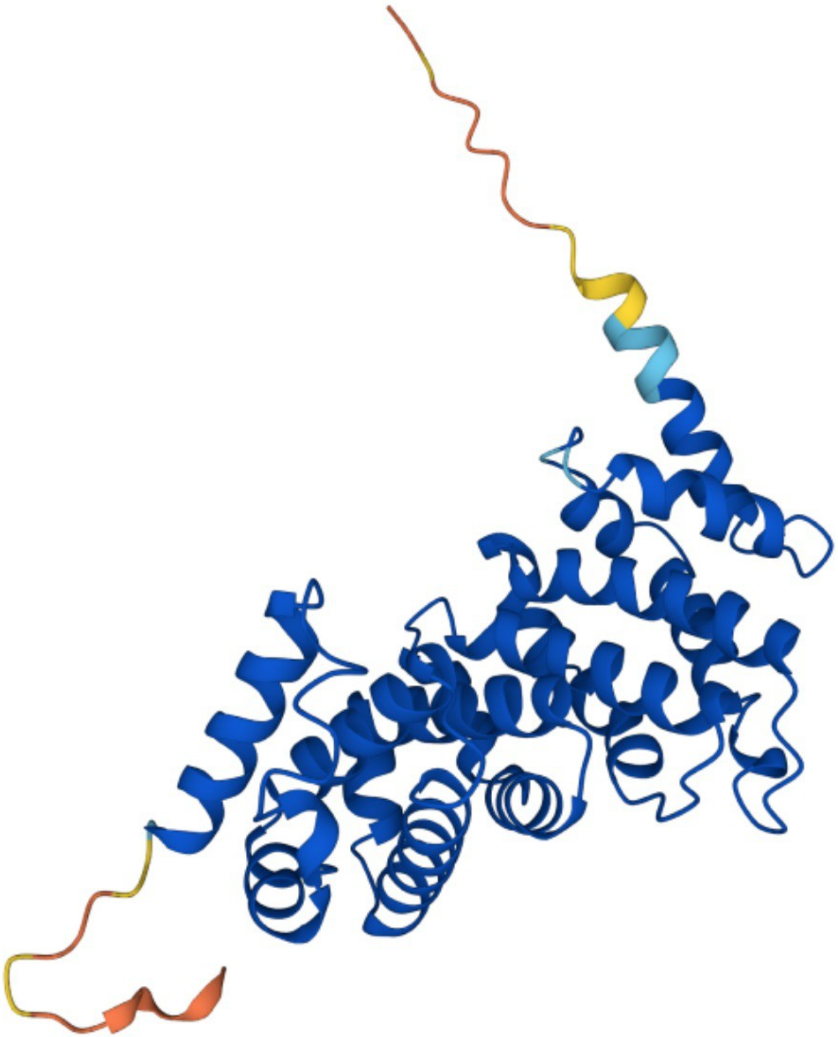
- AlphaFold 2 placed #1 in CASP competition in 2020

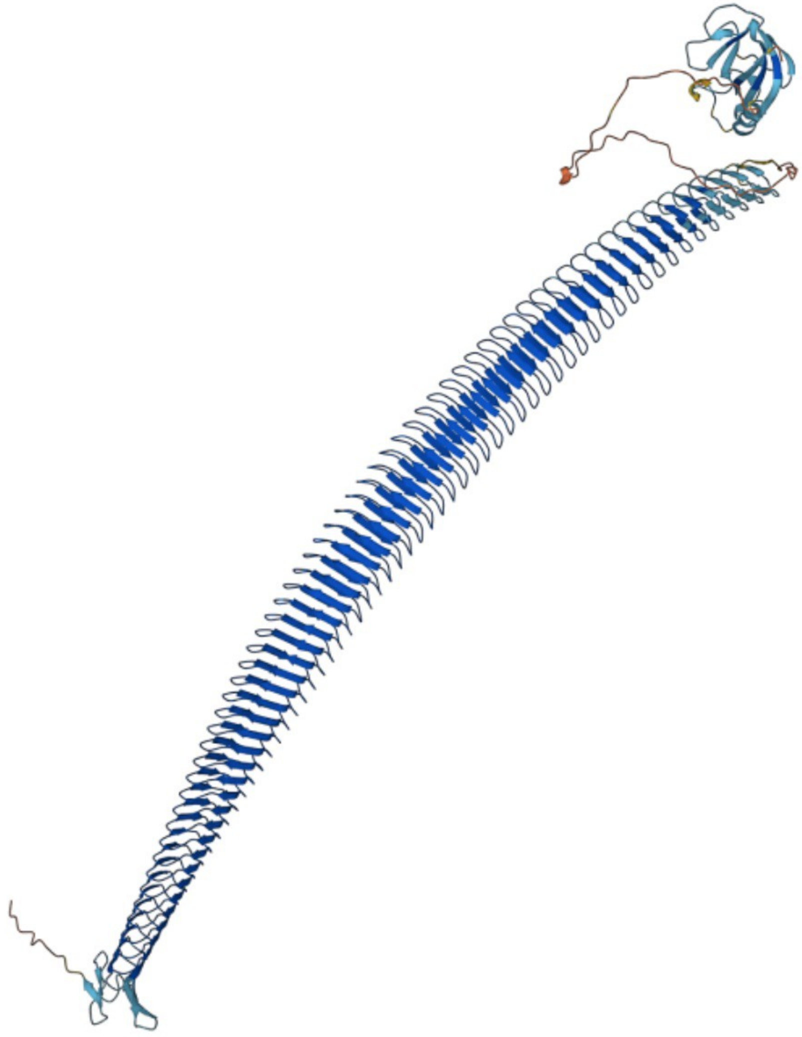
- Predicted protein structures were > 90% accurate in comparison to lab-determined protein structures

AlphaFold

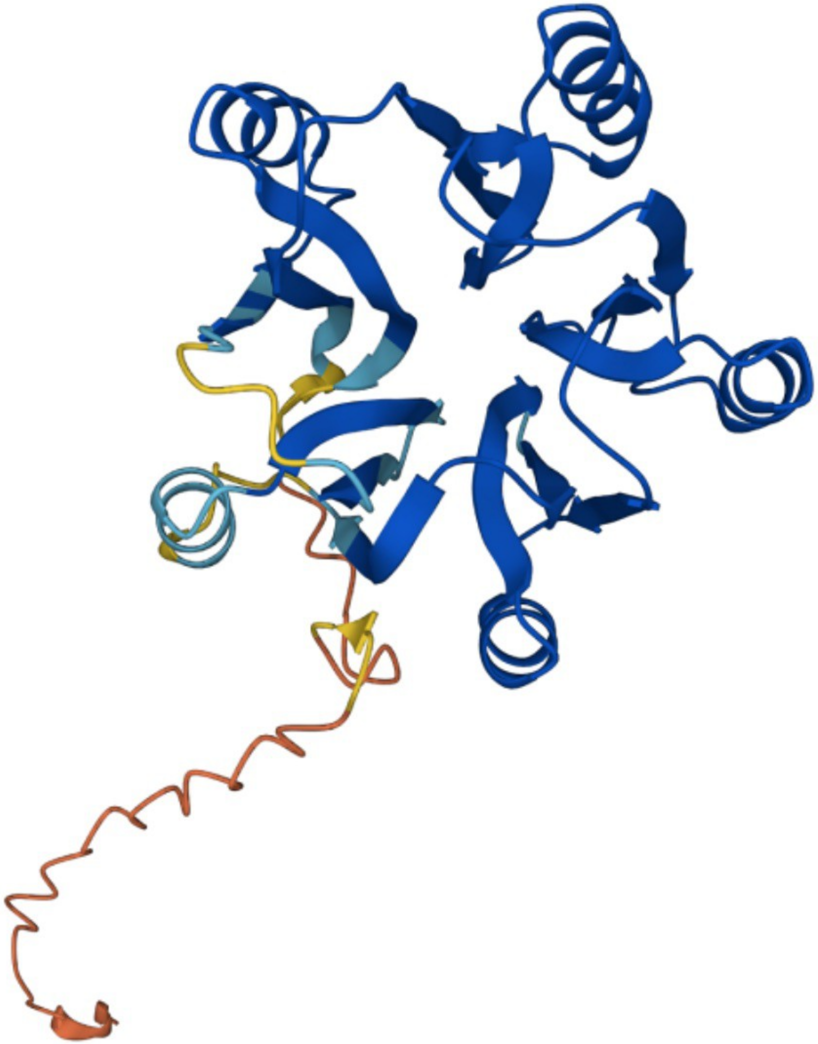


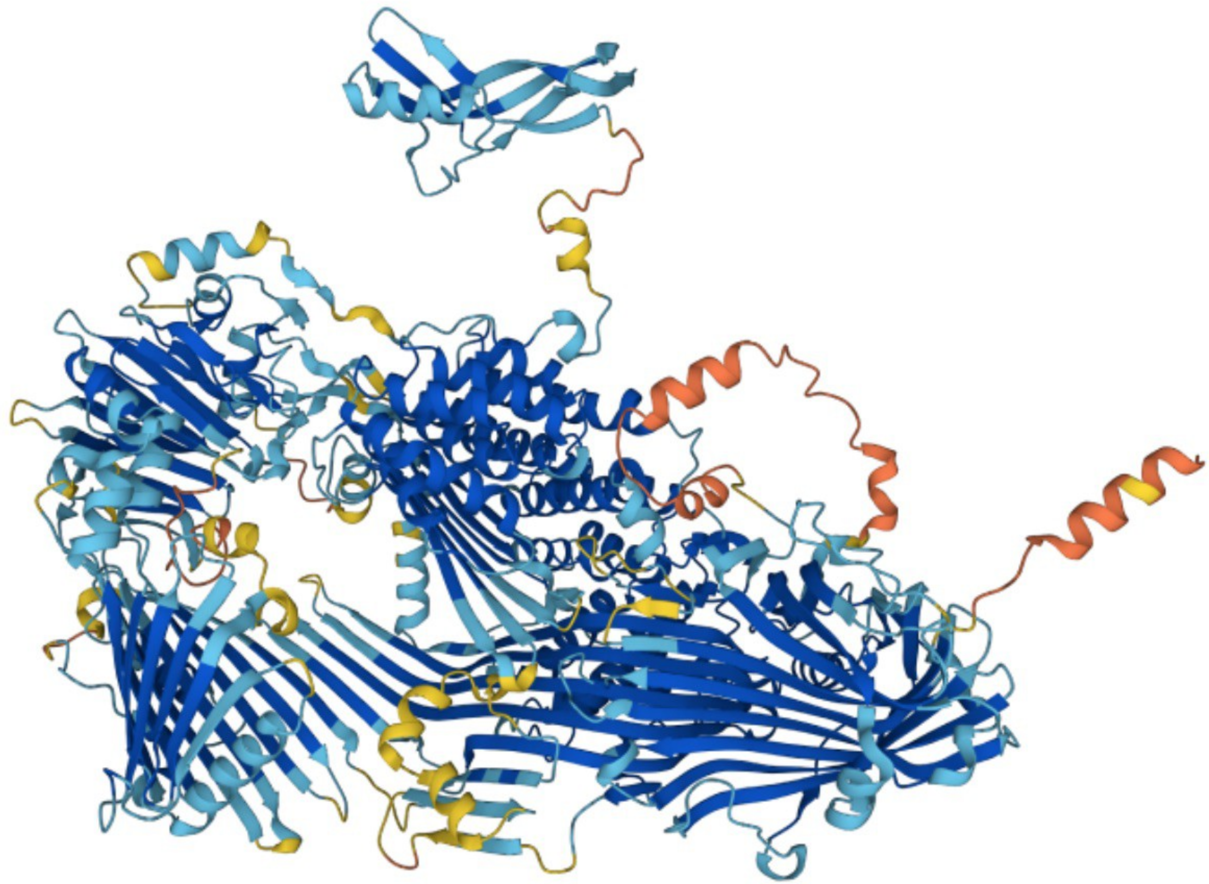
AlphaFold



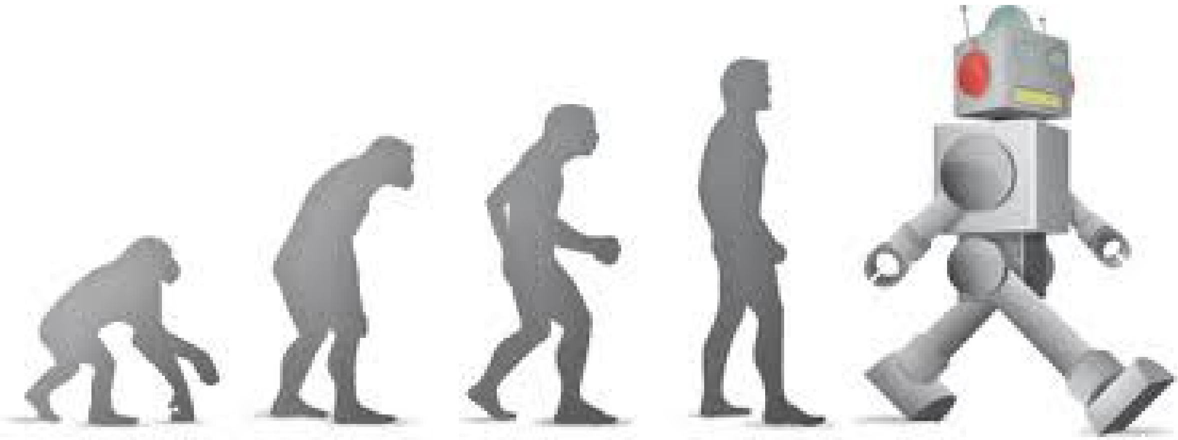


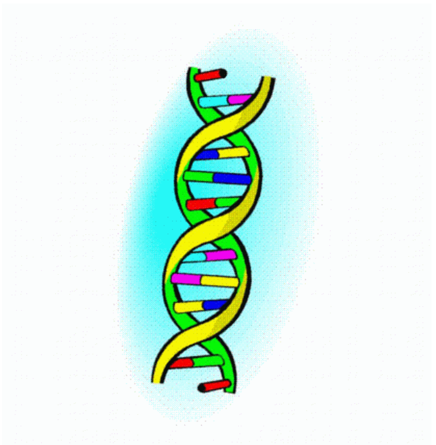
AlphaFold





Genetic Algorithms





Genetic Algorithms

- A collection of strings – "genomes"

000011101001100

110010011110011

100110100110100

111101011111101

00000100100010

101001101010101

000010000101011

110100100010001

Genetic Algorithms

- A collection of strings – “genomes”
- A “fitness function” – rates strings according to some criteria

$f($000011101001100$) = 6$

$f($110010011110011$) = 9$

$f($100110100110100$) = 7$

$f($111101011111101$) = 12$

$f($000000100100010$) = 3$

$f($101001101010101$) = 8$

$f($000010000101011$) = 5$

$f($110100100010001$) = 6$

Genetic Algorithms

- A collection of strings – "genomes"
- A "fitness function" – rates strings according to some criteria
- Survival of the fittest – "selection"

$f($000011101001100$) = 6$

$f($110010011110011$) = 9$

$f($100110100110100$) = 7$

$f($111101011111101$) = 12$

$f($00000100100010$) = 3$

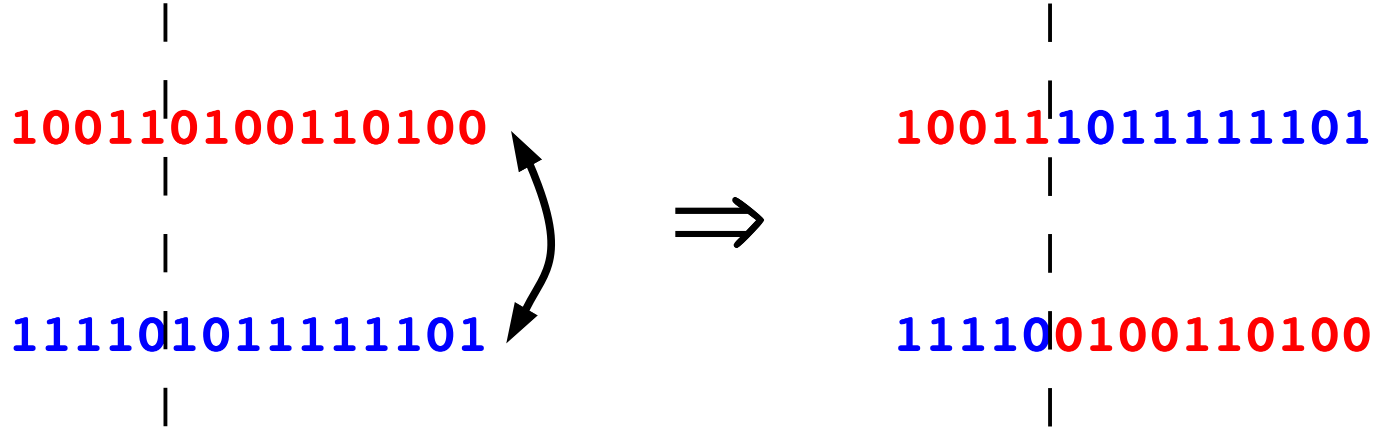
$f($101001101010101$) = 8$

$f($000010000101011$) = 5$

$f($110100100010001$) = 6$

Genetic Algorithms

- A collection of strings – “genomes”
- A “fitness function” – rates strings according to some criteria
- Survival of the fittest – “selection”
- Genetic recombination – “crossover”



Genetic Algorithms

- A collection of strings – "genomes"
- A "fitness function" – rates strings according to some criteria
- Survival of the fittest – "selection"
- Genetic recombination – "crossover"
- Random variation – "mutation"

100111011111101 ⟹ 100111111111111

111100100110100 ⟹ 011100100110100

Outline of a Genetic Algorithm

1. Create a population of random genomes

2. Evaluate the fitness of each genome in the population

3. Build a new population of genomes:

   (a) select 2 genomes probabilistically, based on fitness

   (b) create 2 new offspring from them, using crossover

   (c) mutate each offspring with some small probability

   (d) add the offspring to the new population

   continue steps (a) - (d)

4. When the new population has reached the same size as the current population, replace the current population by the new population … and repeat

Outline of a Genetic Algorithm

- The **average fitness** of the population will increase over time
- Even the **best-fit** individuals are not guaranteed to survive to the next generation
- Even the **worst-fit** individuals have some (small) probability of surviving
- Some GAs use **elitism** to ensure that the top individuals are not lost

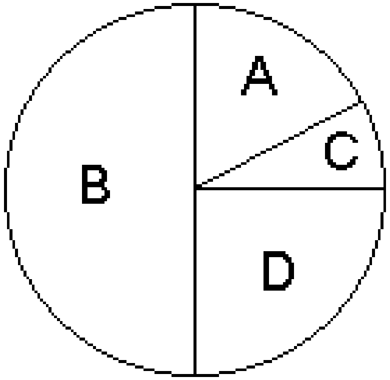
A Simple Example

	Genome	Fitness	New population
A:	**00000110**	2	
B:	**11101110**	6	
C:	**00100000**	1	
D:	**00110100**	3	

Average fitness of current population = 12 / 4 = 3.0

A Simple Example

Genome	Fitness	New population
A: **00000110**	2	
B: **11101110**	6	
C: **00100000**	1	
D: **00110100**	3	



Fitness-proportionate selection
("roulette-wheel sampling")

B: **11101110** and C: **00100000** are selected

A Simple Example

Genome	Fitness
A: **00000110**	2
B: **11101110**	6
C: **00100000**	1
D: **00110100**	3

New population

E: **01101110**
C: **00100000**

No crossover

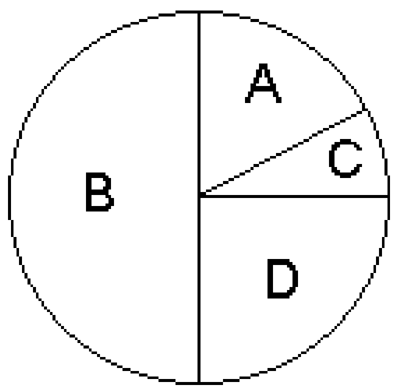
B is mutated

B: **11101110** → E: **01101110**

C: **00100000**

A Simple Example

Genome	Fitness	New population
A: **00000110**	2	E: **01101110**
B: **11101110**	6	C: **00100000**
C: **00100000**	1	
D: **00110100**	3	



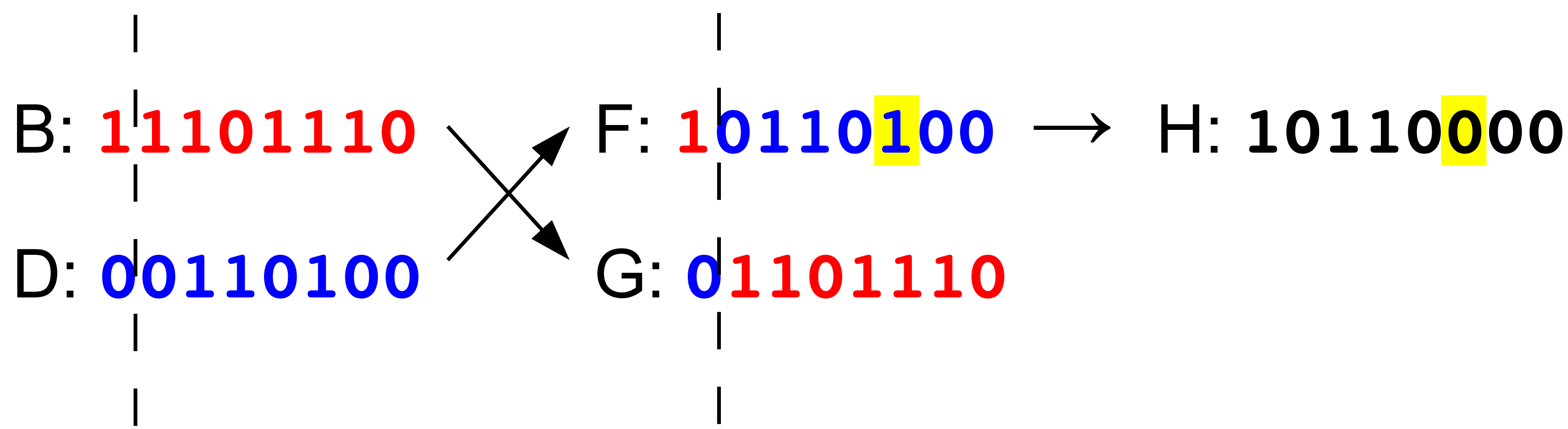
Fitness-proportionate selection
("roulette-wheel sampling")

B: **11101110** and D: **00110100** are selected

A Simple Example

Genome	Fitness	New population
A: **00000110**	2	E: **01101110**
B: **11101110**	6	C: **00100000**
C: **00100000**	1	H: **10110000**
D: **00110100**	3	G: **01101110**

Crossover occurs   F is mutated



A Simple Example

Genome	Fitness	New population	Fitness
A: **00000110**	2	E: **01101110**	5
B: **11101110**	6	C: **00100000**	1
C: **00100000**	1	H: **10110000**	3
D: **00110100**	3	G: **01101110**	5

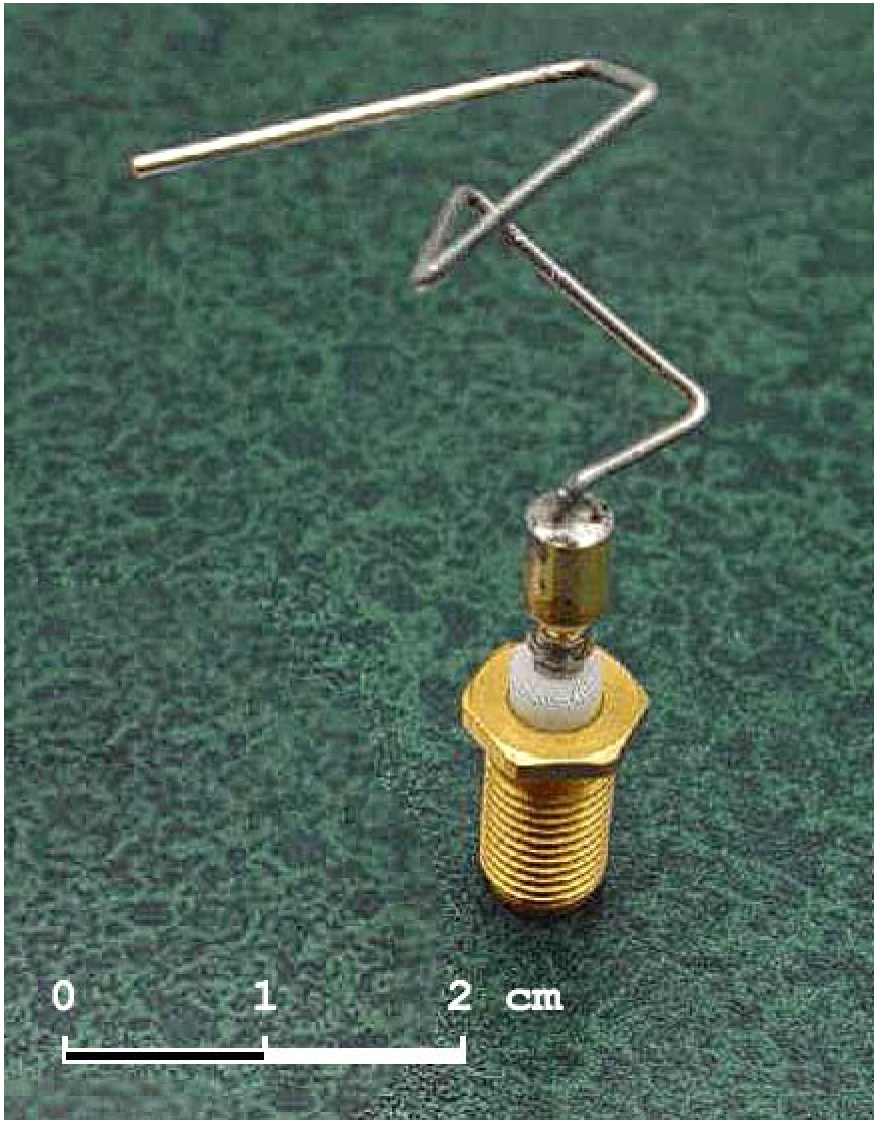
Average fitness of new population = 14 / 4 = 3.5

Best-fit genome from previous population was lost

Evolving an English Phrase

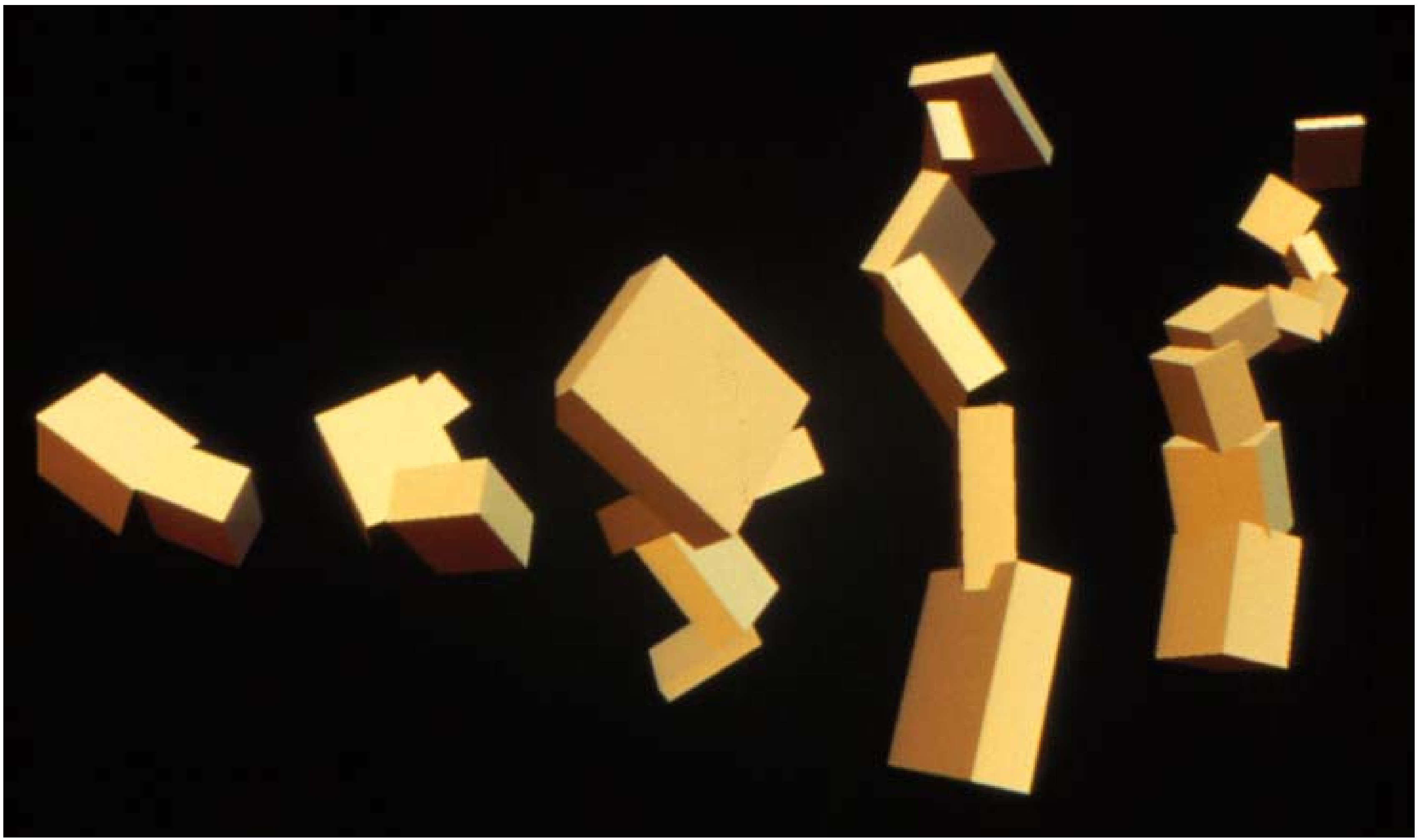
- Genome
  a sequence of letters representing an English phrase
- Fitness function
  the number of letters in the genome that match
  “the rain in spain stays mainly in the plain”
- Example
  “the yain in szbin stays mainly ik the ploin”
  fitness = 38
- GA parameters
  population size: 100
  crossover probability: 0.75
  mutation probability: 0.005 per letter

Demo


0 1 2 cm

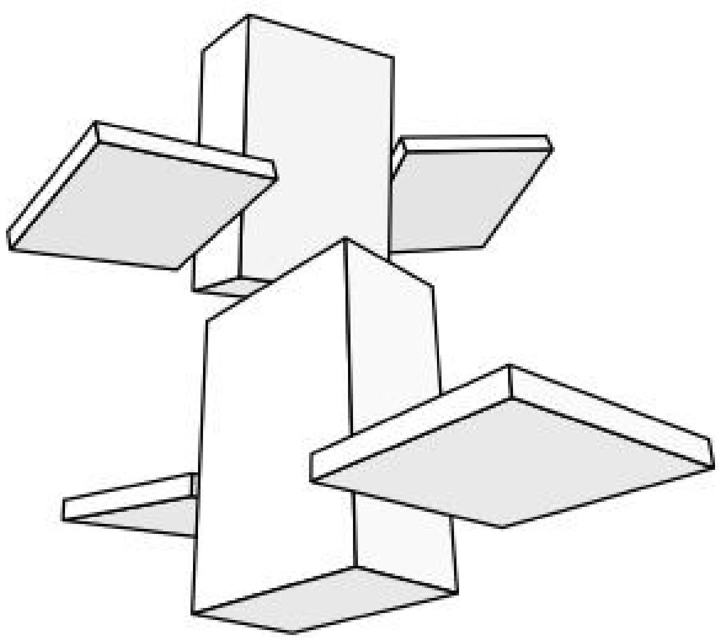
Evolving Virtual Creatures

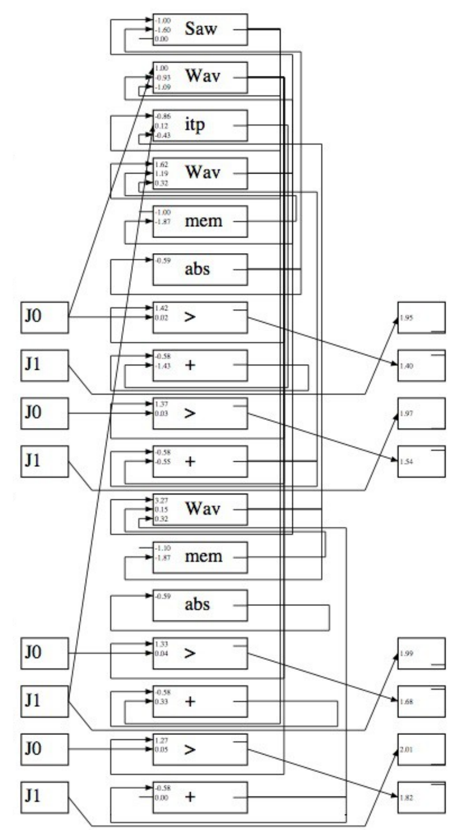
- An interesting application of GAs by Karl Sims (1994)



Evolving Virtual Creatures

- Virtual creatures move around in a **3-D simulated world**
- Creatures' **bodies** are rectangular blocks connected by movable joints, with **sensors** for light and proprioception
- Creatures' **brains** are complex neural networks

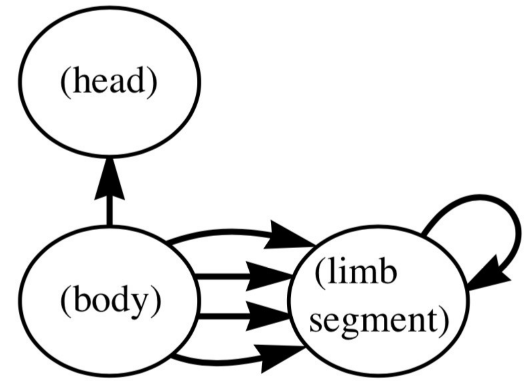




Evolving Virtual Creatures

- A **genome** is a set of nodes and links that encode a creature's body structure and brain structure

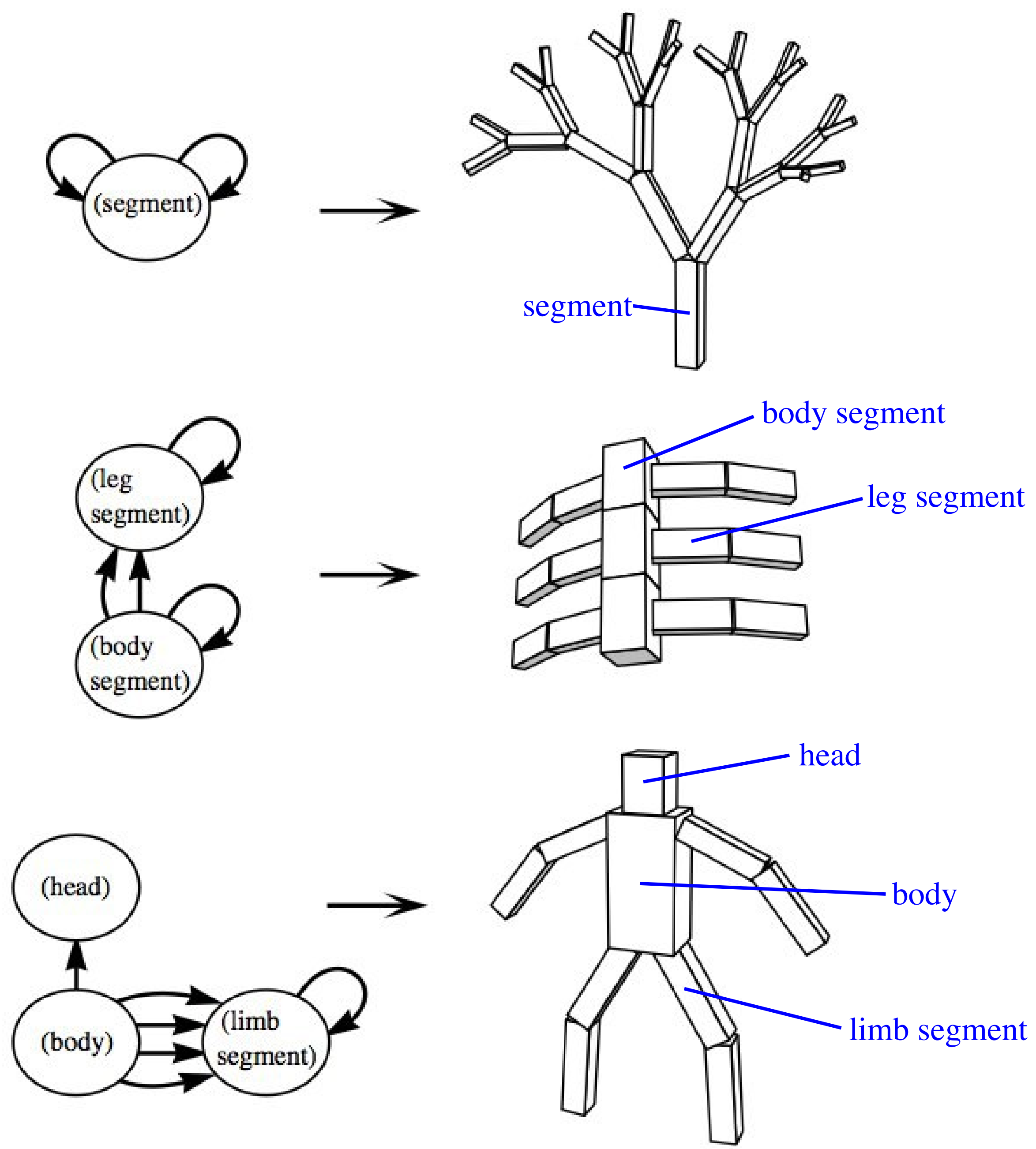
- Complex **genotype** → **phenotype** mapping



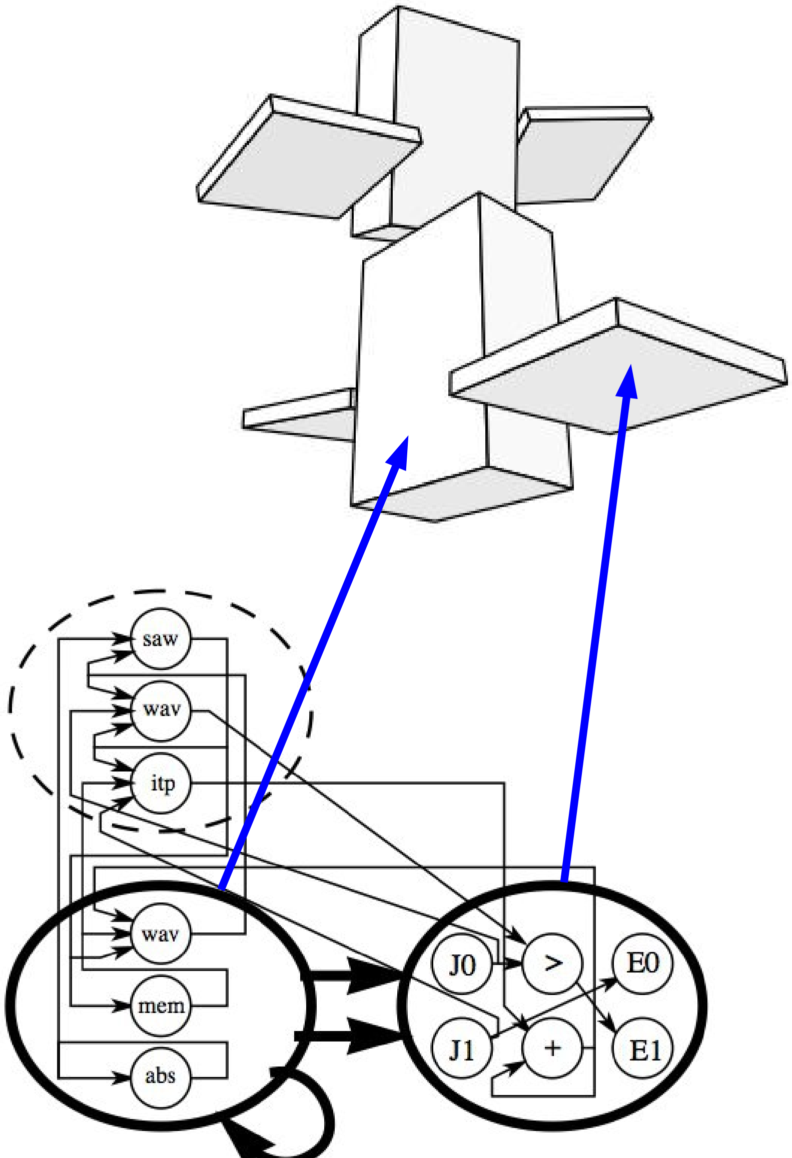
- Brains and bodies **co-evolve** together
  - Body structure evolves
  - Brain structure evolves (neural network topology)
  - Brain parameters evolve (neural network weights)

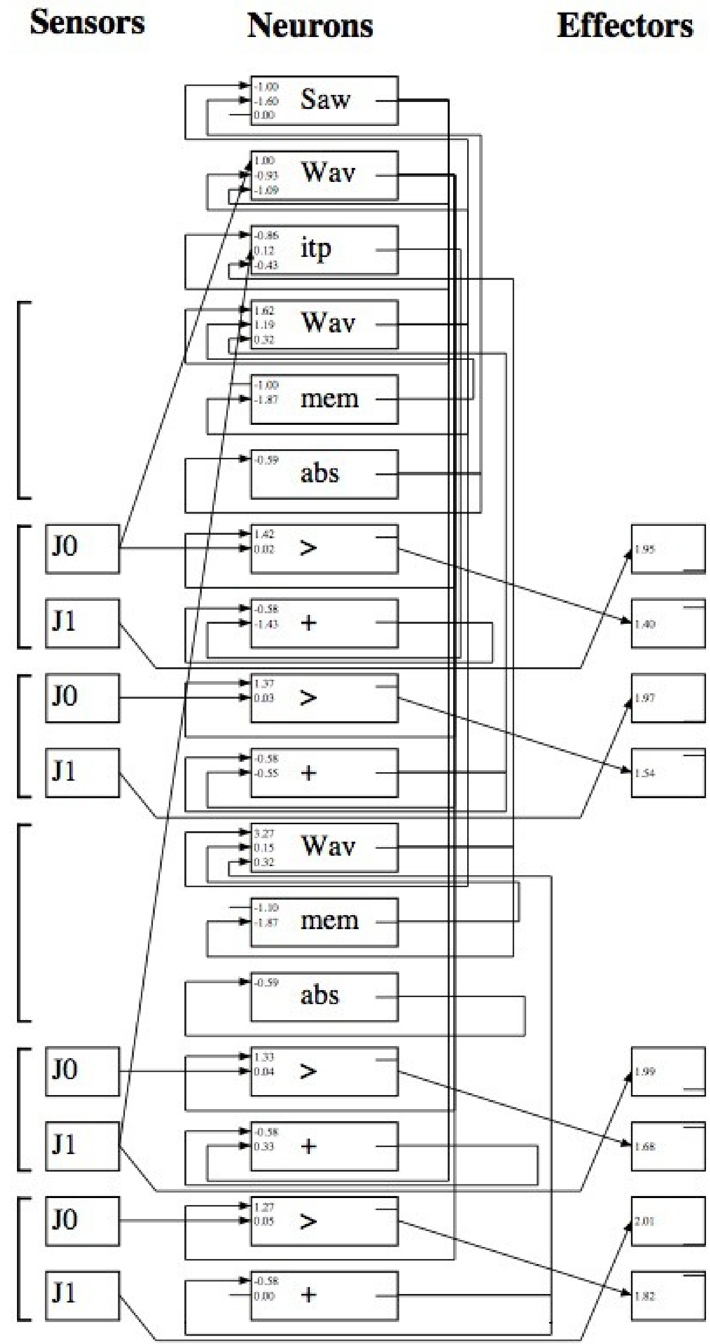
- Fitness: how well a creature can **swim**, **walk**, **jump**, **follow** a light source, or **compete** for control of a block

Genetic Encoding of Body and Brain

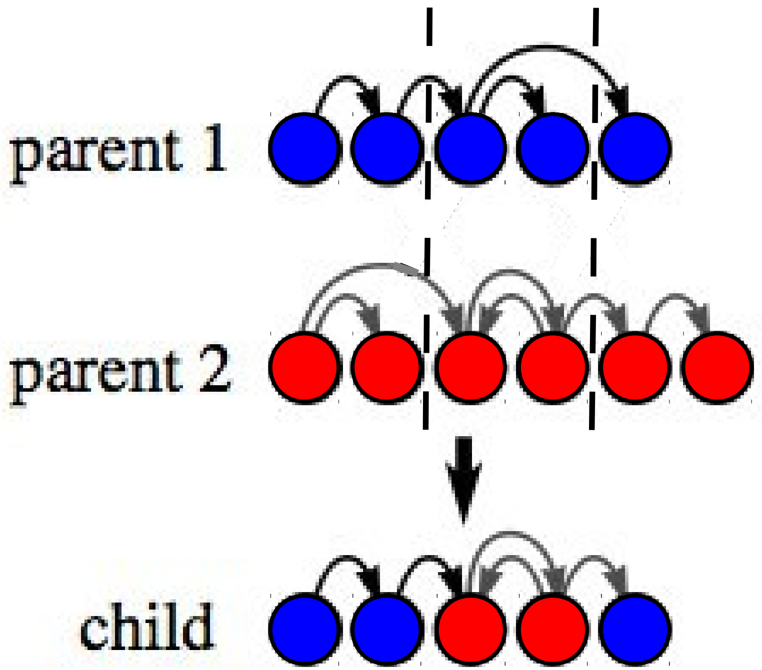


Genetic Encoding of Body and Brain

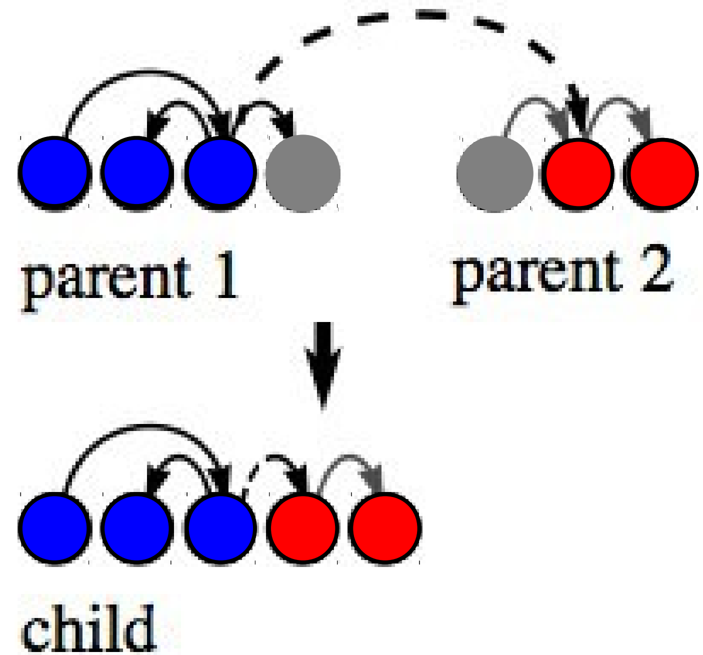




Genetic Recombination



Crossover



Grafting

The Genetic Algorithm

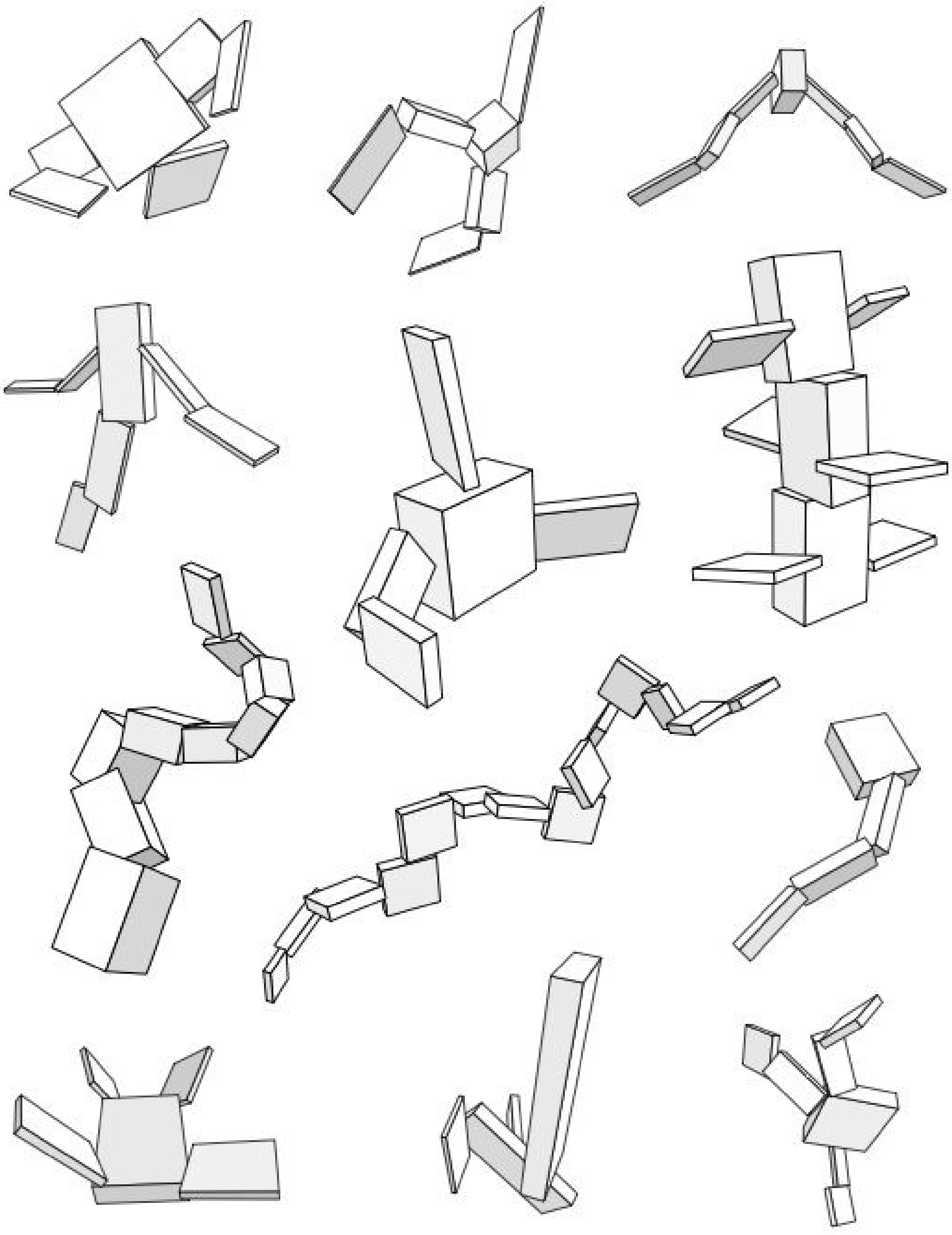
- Population size: 300 genomes
- Evolved for 100 generations
- Fitness evaluation:

genetic description → creature → 3-D simulation

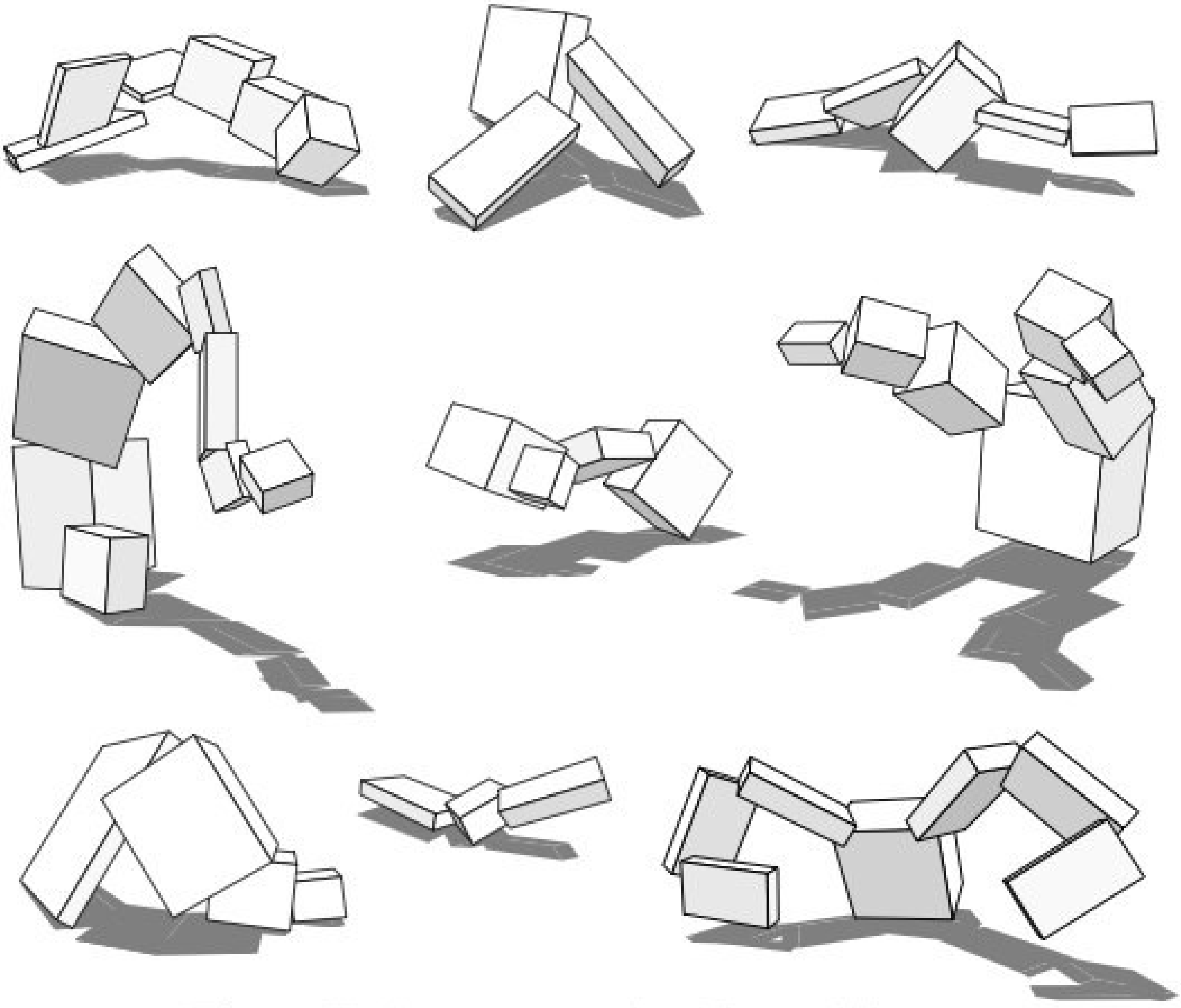
fitness = distance creature walks / swims / jumps / etc.
in a fixed amount of simulation time

- Virtual 3-D world simulates effects of gravity, friction, viscosity

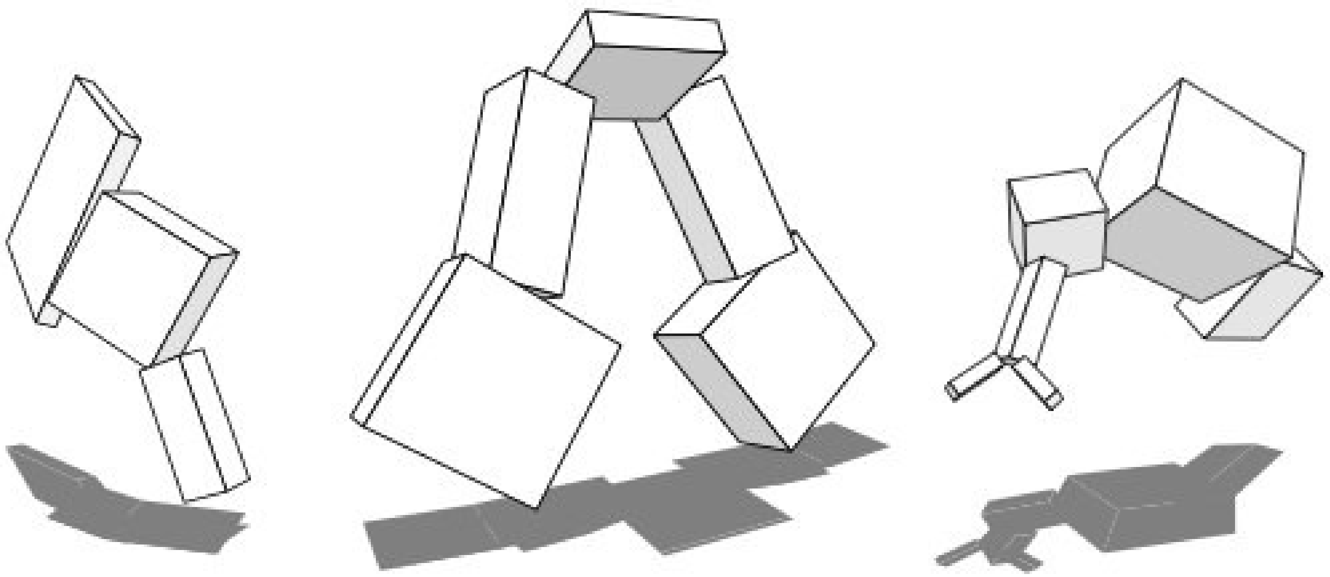
Results: Swimmers



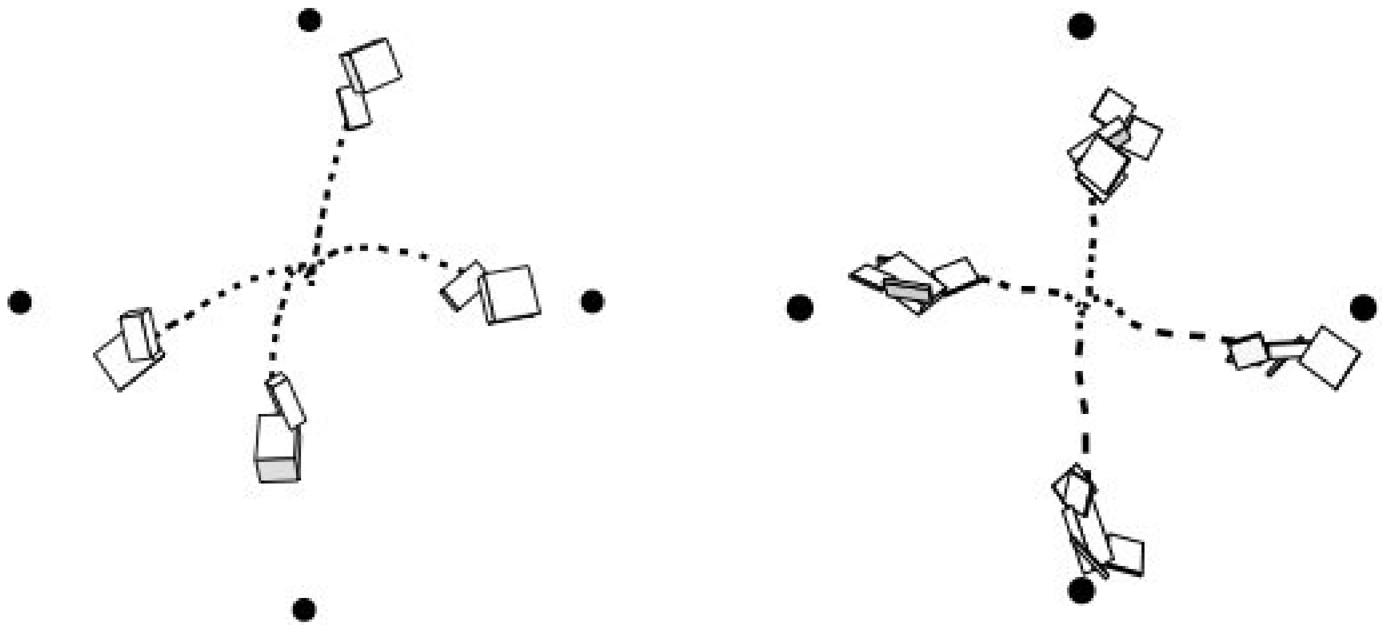
Results: Walkers



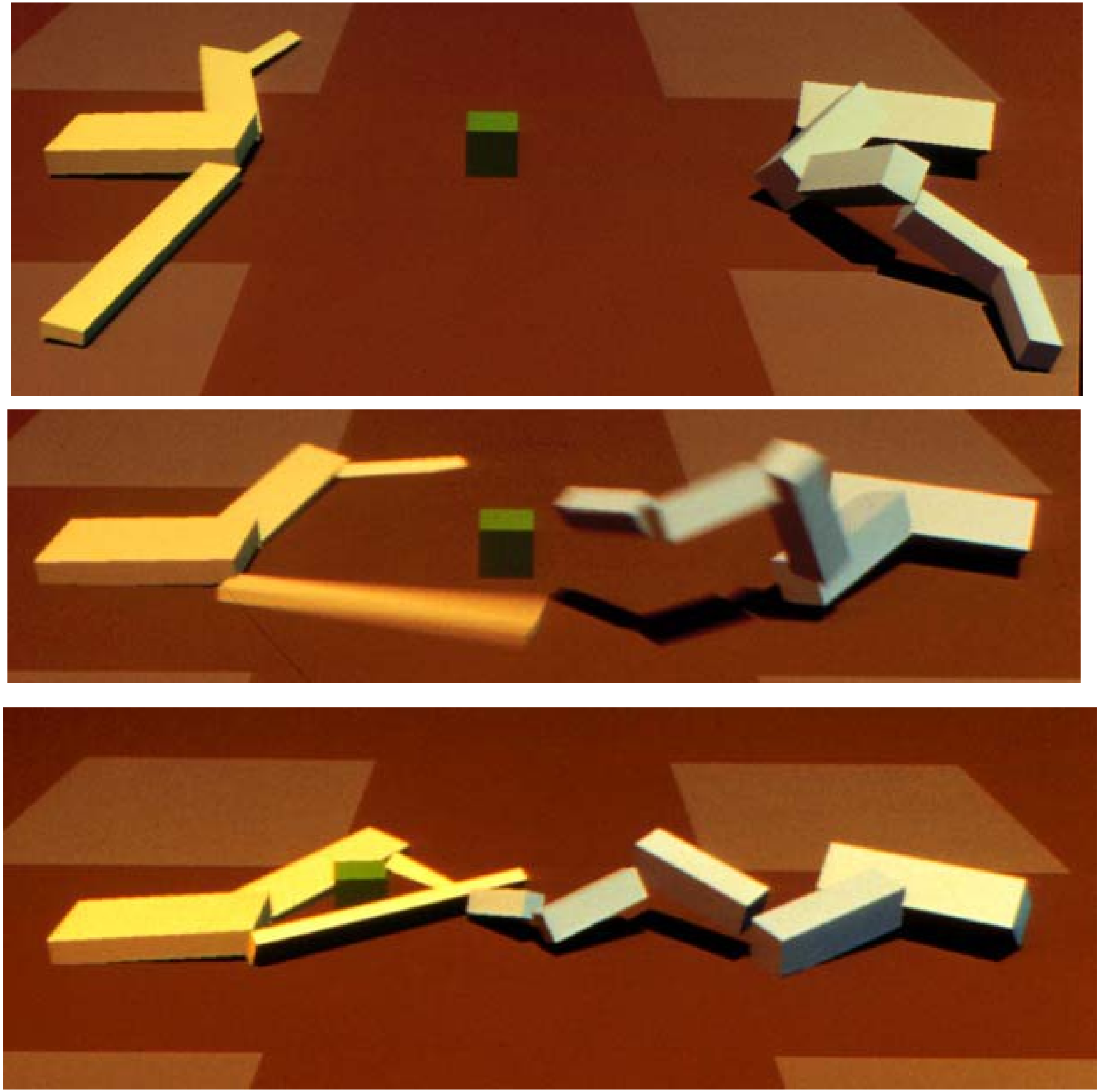
Results: Jumpers



Results: Light Followers



Results: Competitors



Video