THE ELECTRONICS RESURGENCE INITIATIVE

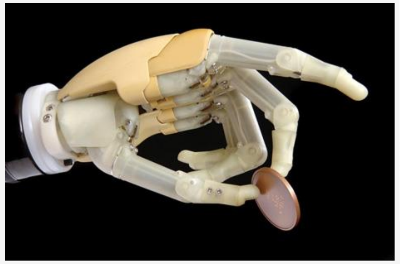
LIFELONG LEARNING MACHINES (L2M)

HAVA SIEGELMANN

PROGRAM MANAGER
DARPA MTO

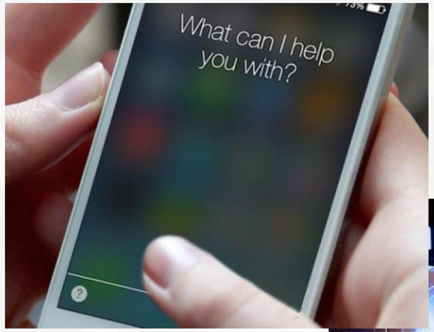
AI TODAY

Control methods?



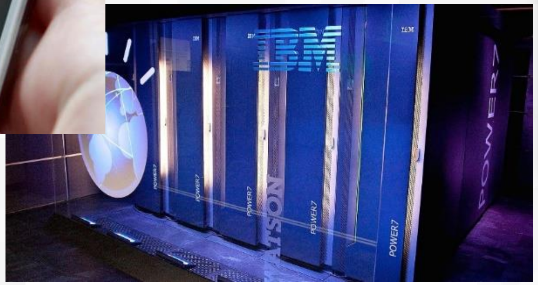
http://4.bp.blogspot.com/-G2pRpNi80Lk/Uvjd46xBQcI/AAAAAAAAAD0/utvMBR_dAe4/s1600/Bionic_hand.jpg

©edu.glogster.com



©Apple

Recognition
- speech



©IBM

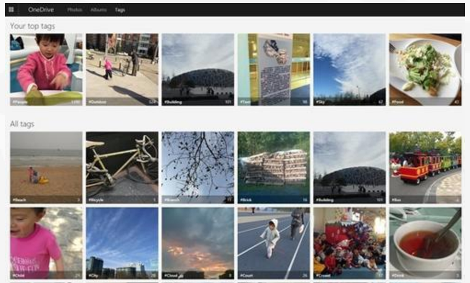
©DeepMind Technologies

Gaming
< 1982

©United States Navy

Self driving?

©Google



https://i2.kknews.cc/SIG=29vnh65/2175/3455714929.jpg

Recognition
- image

Foundational methods:

1. Programs, Rule-based systems (human experts + databases + processors)
2. Machine learning (parametric structure + learning rule + training-databases + more-processors)

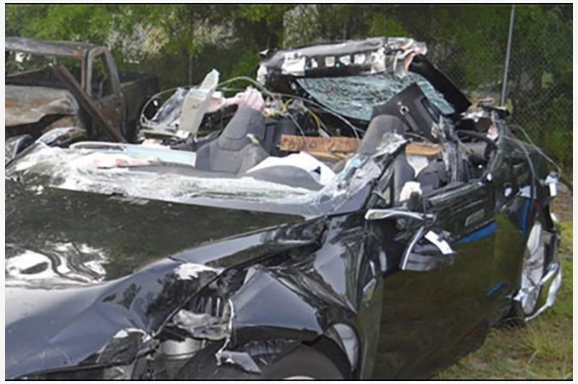
AI LIMITATIONS

AI systems only compute with what they've been programmed or trained for in advance

AI: Both algorithms and machine learning are frozen after preparation phase

1. Malfunctions in circumstances that exceed preparation: No way to prepare for every eventuality
2. No easy fix to learn from errors, enlarging repertoire of behaviors (catastrophic forgetting)
3. Worsens with increase in autonomous applications

https://ichef.bbci.co.uk/news/660/cpsprodpb/7708/production/_99727403_b02dbecb-aa3a-4aea-a013-170ffbfb0fd4.jpg



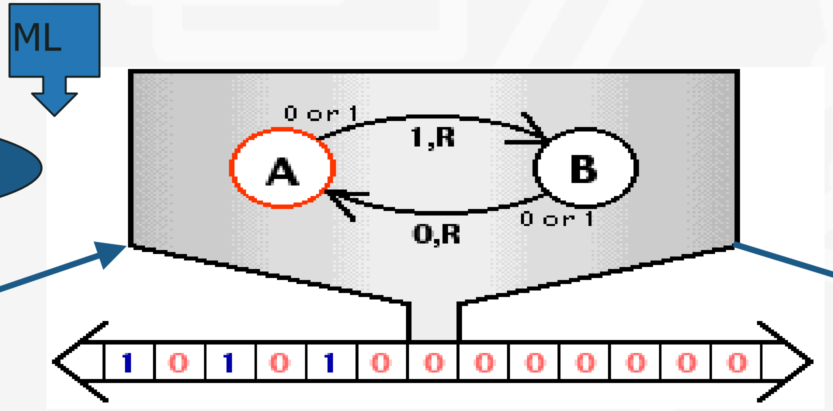
https://i.guim.co.uk/img/media/70d0711df5ee63efc8e012dc7e085a43863fbc8f/59_0_2986_1792/master/2986.jpg?w=620&q=55&auto=format&usm=12&fit=max&s=510e1039a4cb541e5a746f0d711d59d9

https://www.vosizneias.com/wp-content/uploads/2018/03/ubers-725x269.jpg

©Tempe Police Department

TODAY'S COMPUTATIONAL FOUNDATION: TURING MACHINES

In 1936, Alan Turing modeled "human-calculators" as theoretical automatic machines



©blogspot.com

Loadable program

Input

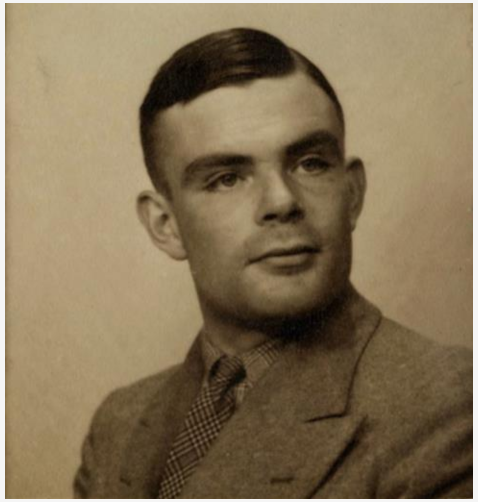
Memory tape

Output

Current AI has two pre-execution parts:

- **Program and rules**
- **Parameter learning (e.g., in ML)**

TURING ON INTELLIGENT MACHINES



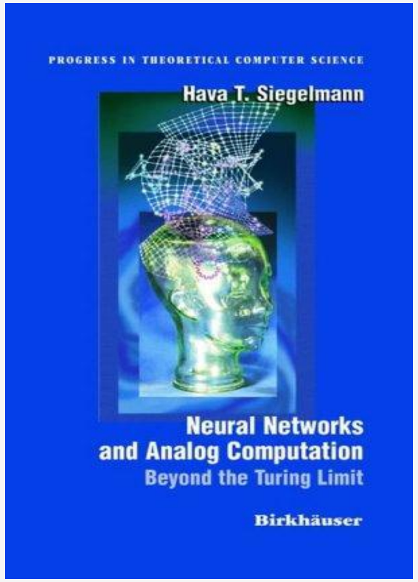
http://godsandfoolishgrandeur.blogspot.com/2013/10/alan-turing.html

"Electronic computers are intended to carry out any definite rule of thumb process which could have been done by a human operator working in a disciplined but unintelligent manner." ('50)

"My contention is that machines can be constructed that will simulate the behaviour of the human mind" ('51)

"What we want is a machine that can learn from experience" ('47)

SUPER-TURING CONTINUUM HIERARCHY



Continuum of computational hierarchy: From Turing Machines (fixed programs) to Super-Turing Computation (modifiable programs)

Turing machines change output based on input

Super-Turing machines change program based on inputs

Change of any of the following TM properties will lead to ST

1. Discrete values	Analog values (Real)
2. Deterministic	Randomness/asynchronous
3. Pre-programmed	Lifelong Learning, evolving
4. One algorithm	Series of machines

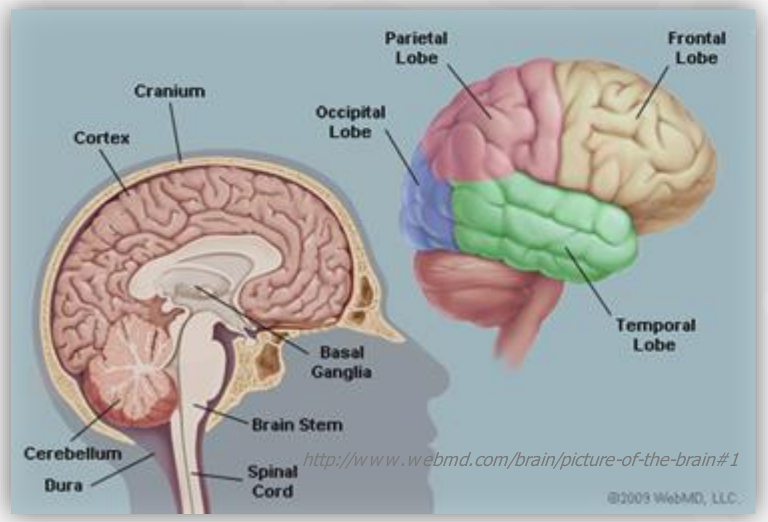
$\alpha \in$ Kolmogorov$[f(n),g(n)]$: UTM calculates α[n-prefix] from *f(n)* bits in *g(n)* time P=K[1,p(n)] AnalogP=K[n,n]

Turing suggested these properties for future computers that can learn

NATURE COMBINES TURING WITH SUPER-TURING COMPUTATION

- Turing machines change output based on input
- Super-Turing machines change program based on inputs

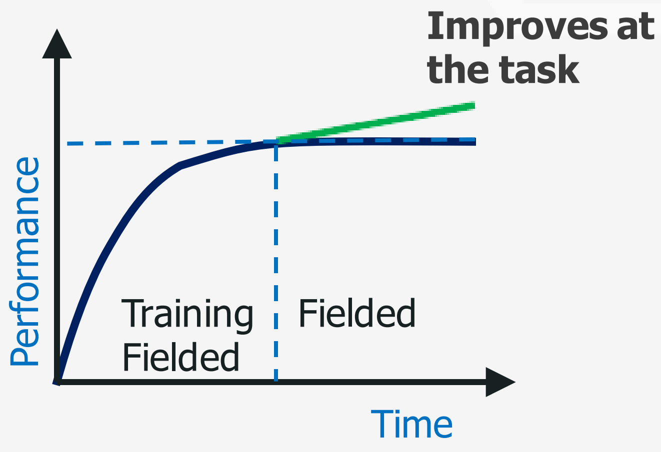
- Nature systems follow (Turing-like) programs
- They adapt as needed, changing their Turing programs
- They store revised Turing programs as components for future use



WE WANT: LIFELONG LEARNING FOR AI APPLICATIONS

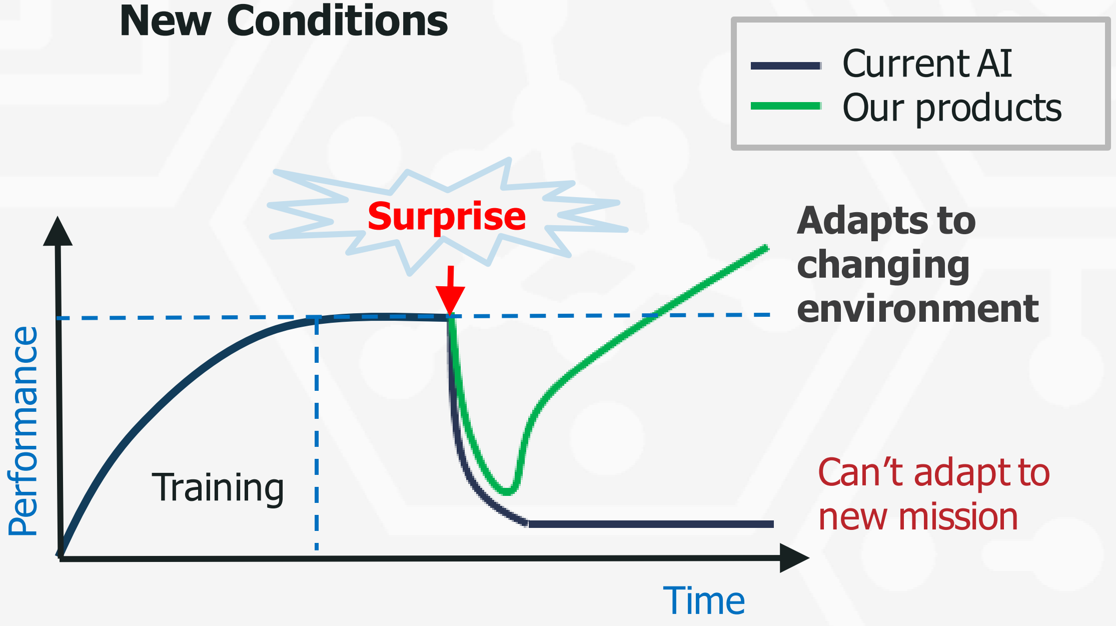
L2M is concerned with learning machines that will improve their performance over their lifetimes

Continuously Improve Performance



Current ML based on large datasets; data may be scarce

Adapt to New Conditions



Situation may change after training and fielding (external, internal)

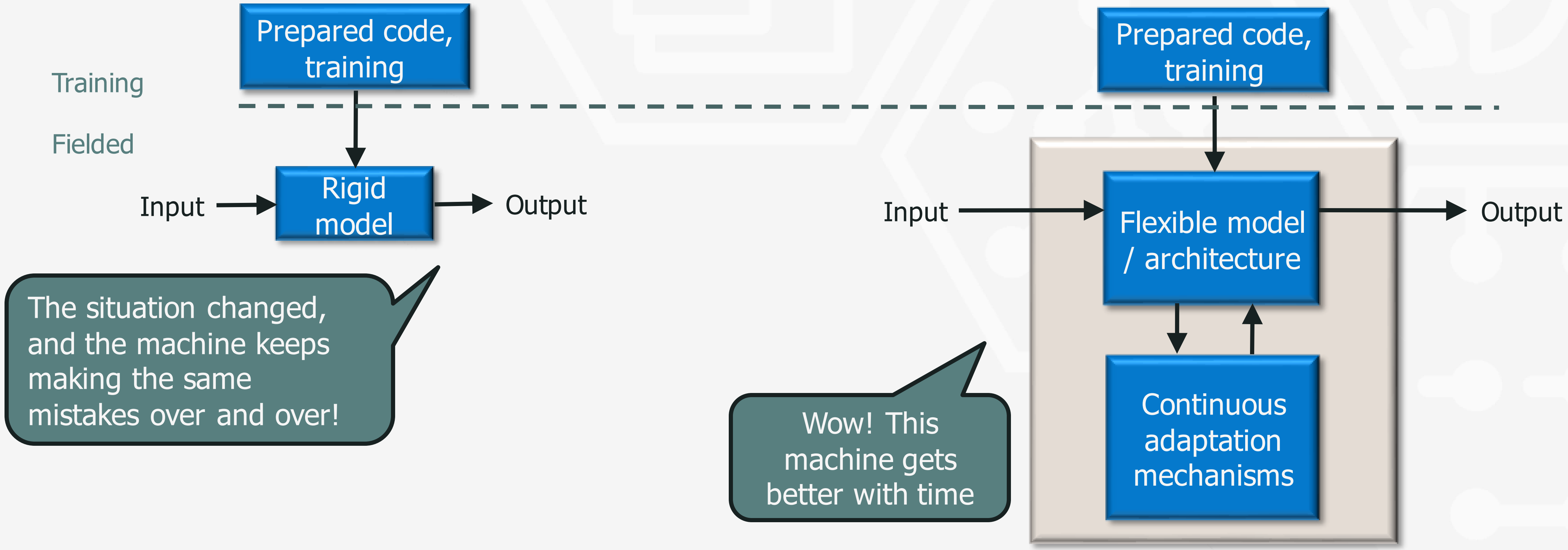
LIFELONG LEARNING SYSTEM

Today

- Execution follows completed training cycle
- Fixed during execution
- Hardware - static systems for an AI method

Next Generation

- Continues learning during execution
- Program adapts to new situations, new tasks
- Hardware supports updates, protects manipulations



This is the goal of the L2M Program

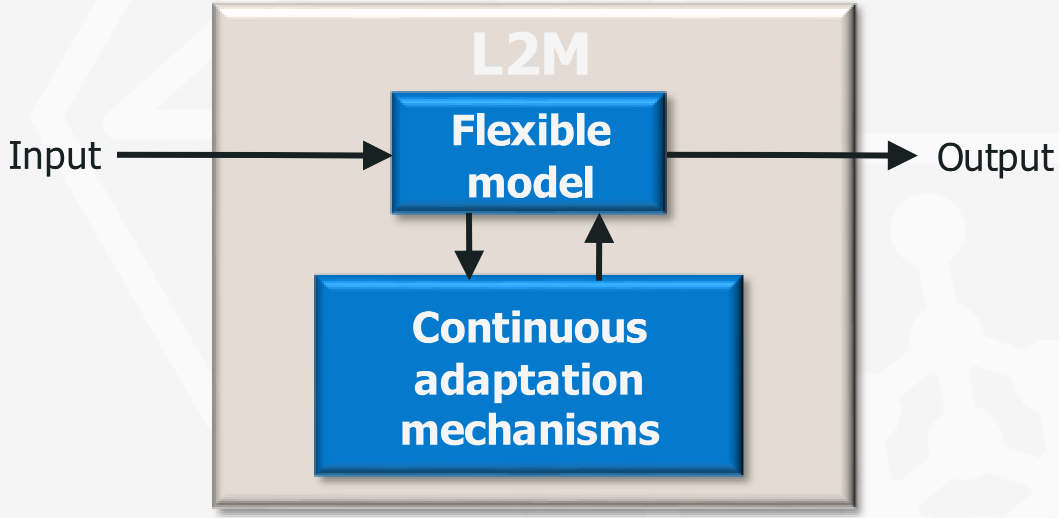
CORE CAPABILITIES OF AN L2M SYSTEM

Examples L2M systems:

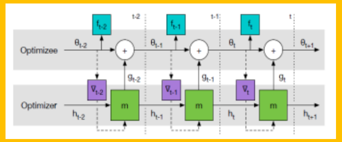
- A car that becomes better on snowy roads each time it drives on them (an expert)
- A plane that learns to fly more efficiently and safely

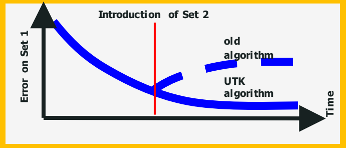
1. **Continual learning** – systems capable of learning during execution, data not i.i.d.
2. **Adaptation to new tasks and circumstances** – applying previously learned skills to novel situations without forgetting previously learned tasks
3. **Goal-driven perception** – choosing and perceiving input signals from mission view
4. **Selective plasticity** – balancing stability vs. plasticity; knowing when to learn
5. **Safety and monitoring** – ensuring correct behavior in a system that continues to change

L2M PROGRAM STRUCTURE



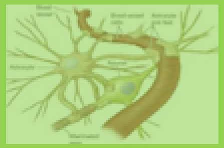
TA 1:
Lifelong learning systems





- Software, architectures, and algorithms
- Theory, including combining supervised, unsupervised, and reward-based learning

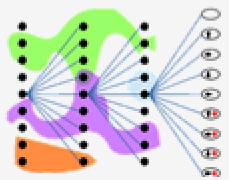
TA 2:
Biological principles

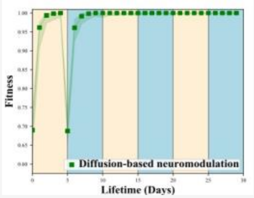


- Learn from nature
- Transfer to machine learning

SOME CURRENT IDEAS FOR SOLUTIONS

CONTEXTUAL ADAPTATION (CLUNE)





Source: University of Wyoming

Bio-inspired neuromodulation divides the network into modules of reusable memories based on context

AUTOGENERATING NN (LIPSON)

Source: Columbia University

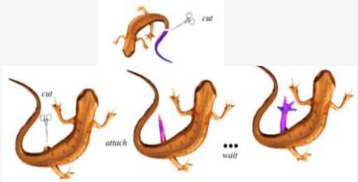
Build network that generates itself (auto-generation). Build neural nets that generate better neural nets

SELF-DIRECTED NN (LEARNED-MILLER)

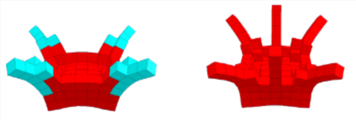
Source: UMass Amherst

Agent uses downtimes to challenge itself with surrogate tasks – to learn in the absence of explicit labels

SOMATIC COMPUTATION (LEVIN)



Source: Tufts University



(Corucci et al, 2016)

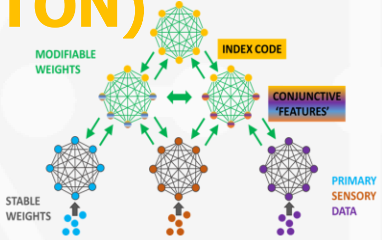
Bioelectric somatic like computation to recover from injury, flexible robots and adapt to new environments

TASK REUSE (EATON)

Source: University of Pennsylvania

Efficient re-use of previously learned computational primitives and their continual improvement

SLEEP AND MEMORY (MCNAUGHTON)

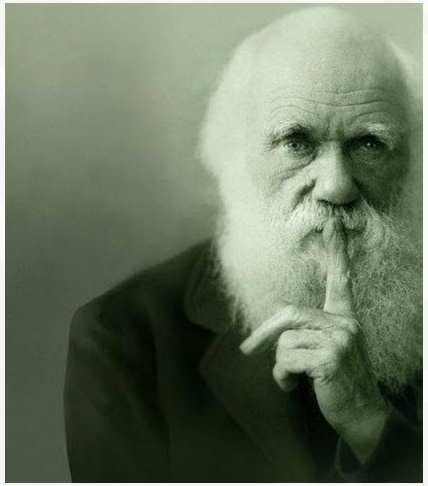


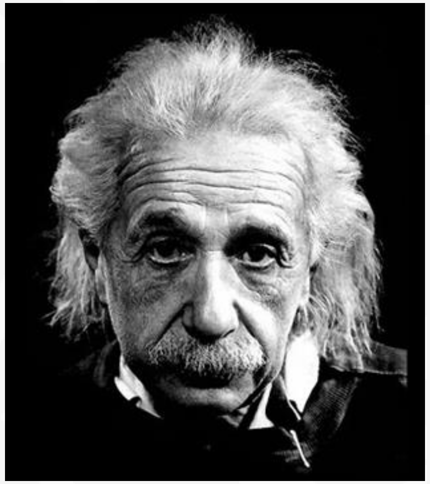
Source: UC Irvine

Fast and slow "index-code" learning (hippocampus-cortex) drives selective plasticity, reduces catastrophic forgetting

"…it is not the strongest that survives; but…the one that is able best to adapt…to the changing environment….”

L.C. Megginson, re “On the Origin of Species”





https://www.izlesene.com/iz/memcn3342

“Once you stop learning, you start dying.”

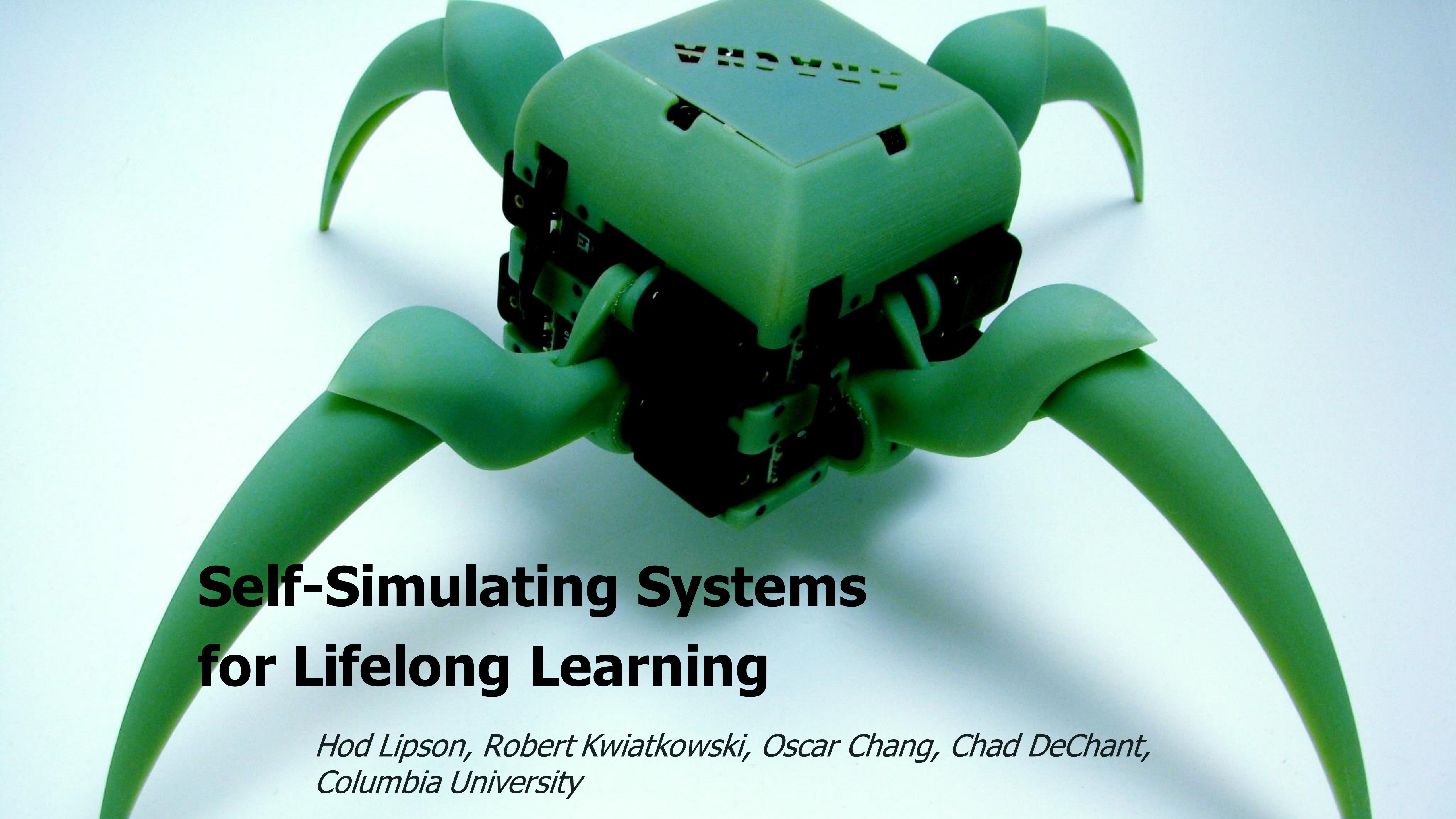
Albert Einstein

Thank you

SELF-SIMULATING SYSTEMS FOR LIFELONG LEARNING

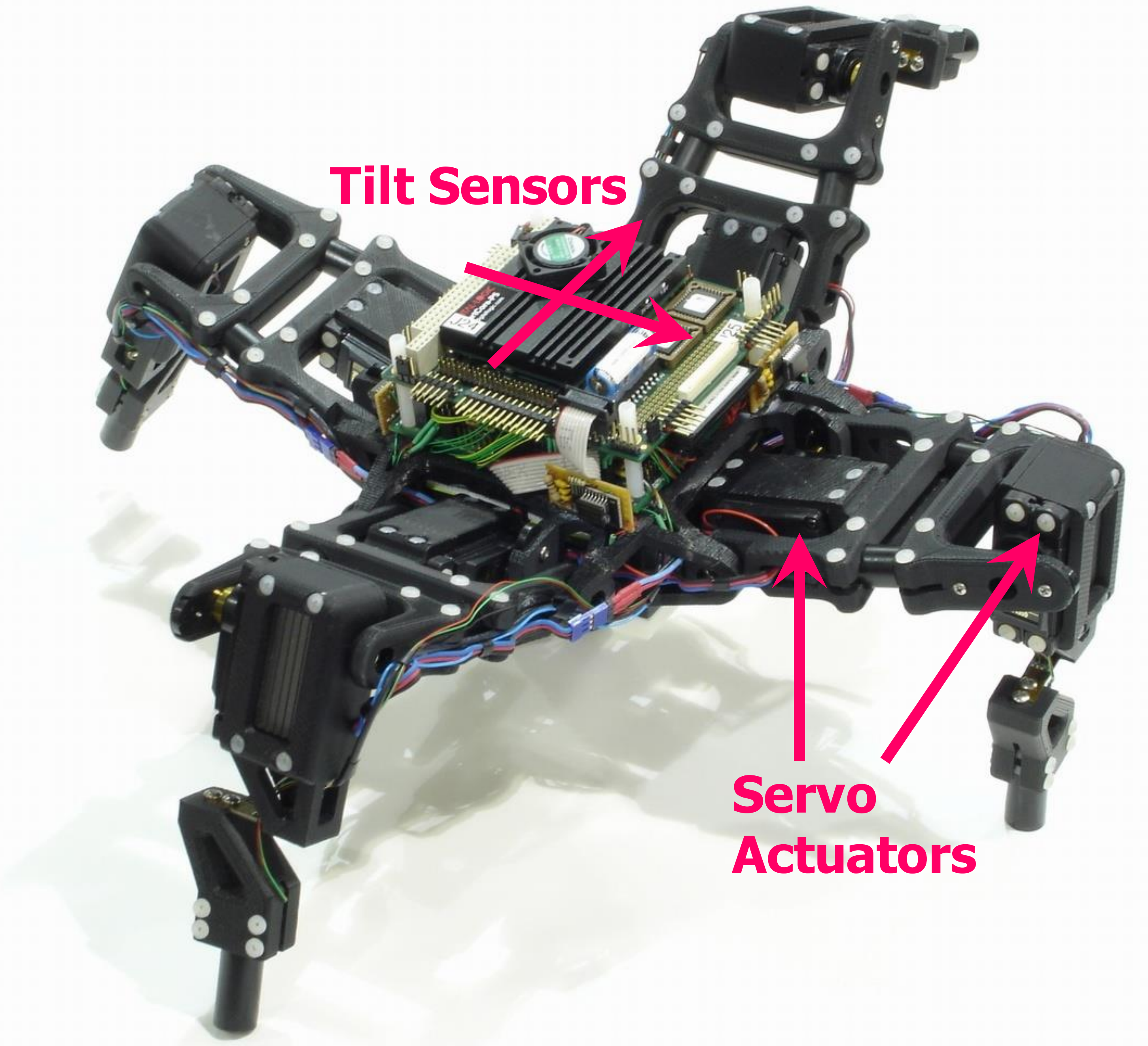
HOD LIPSON

PROFESSOR
COLUMBIA UNIVERSITY



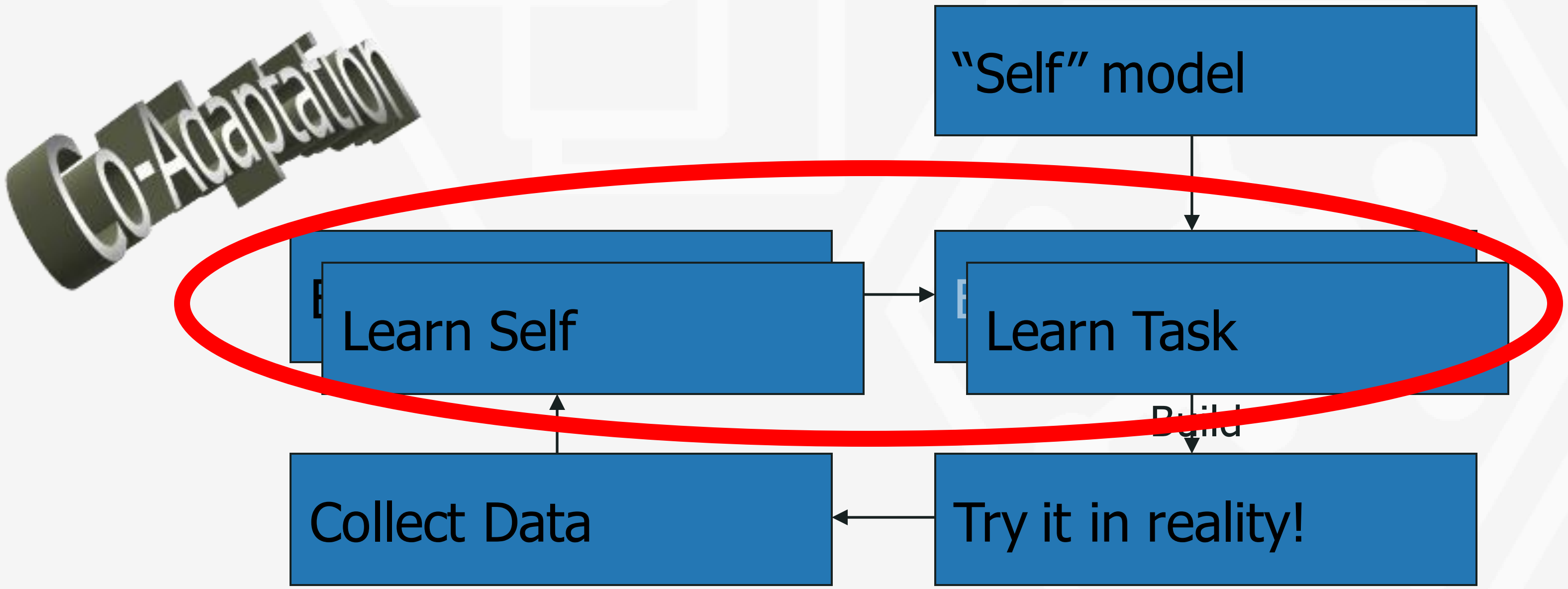
Self-Simulating Systems for Lifelong Learning

Hod Lipson, Robert Kwiatkowski, Oscar Chang, Chad DeChant,
Columbia University



With Josh Bongard, Viktor Zykov

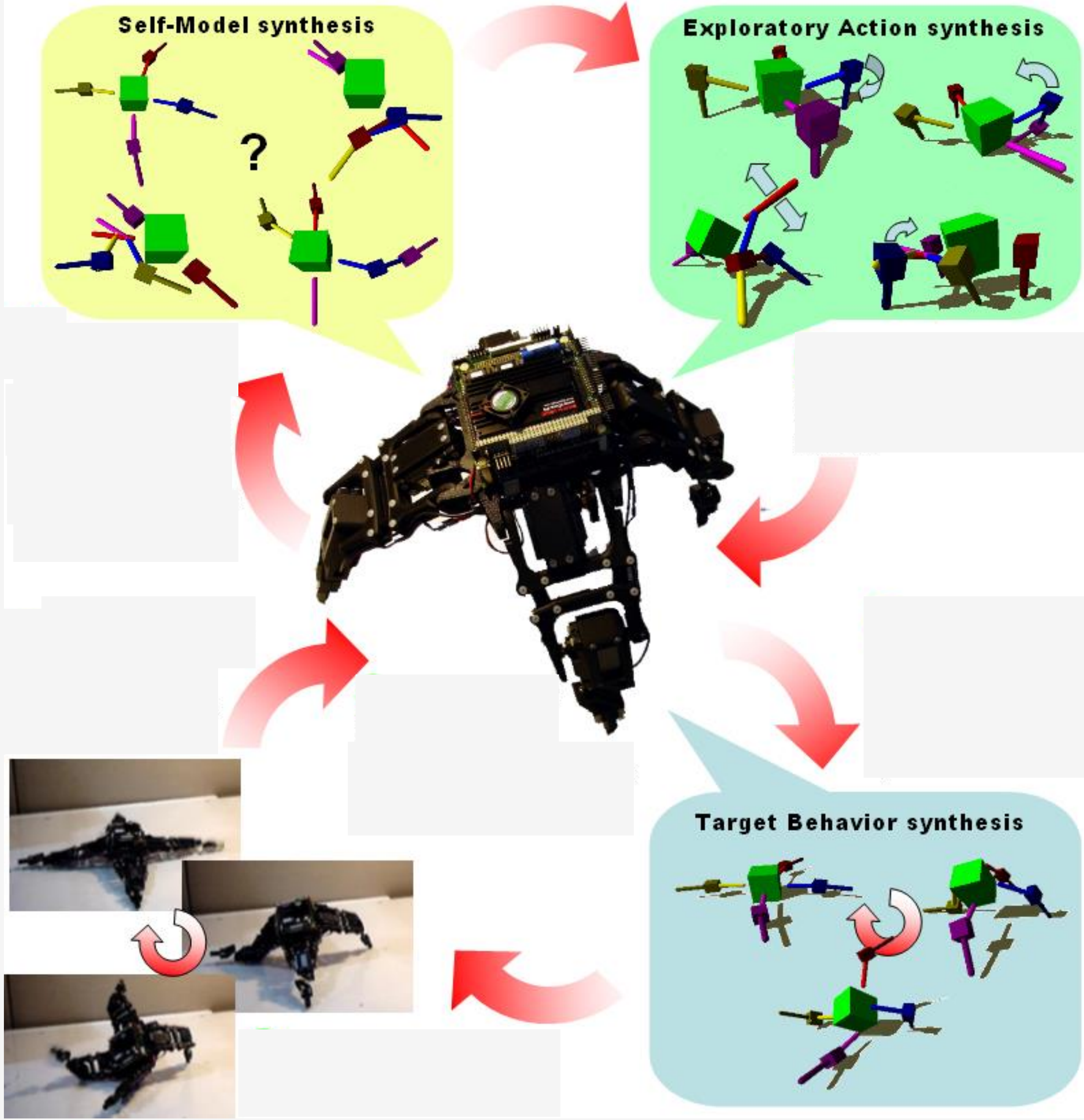
MODULAR ADAPTATION

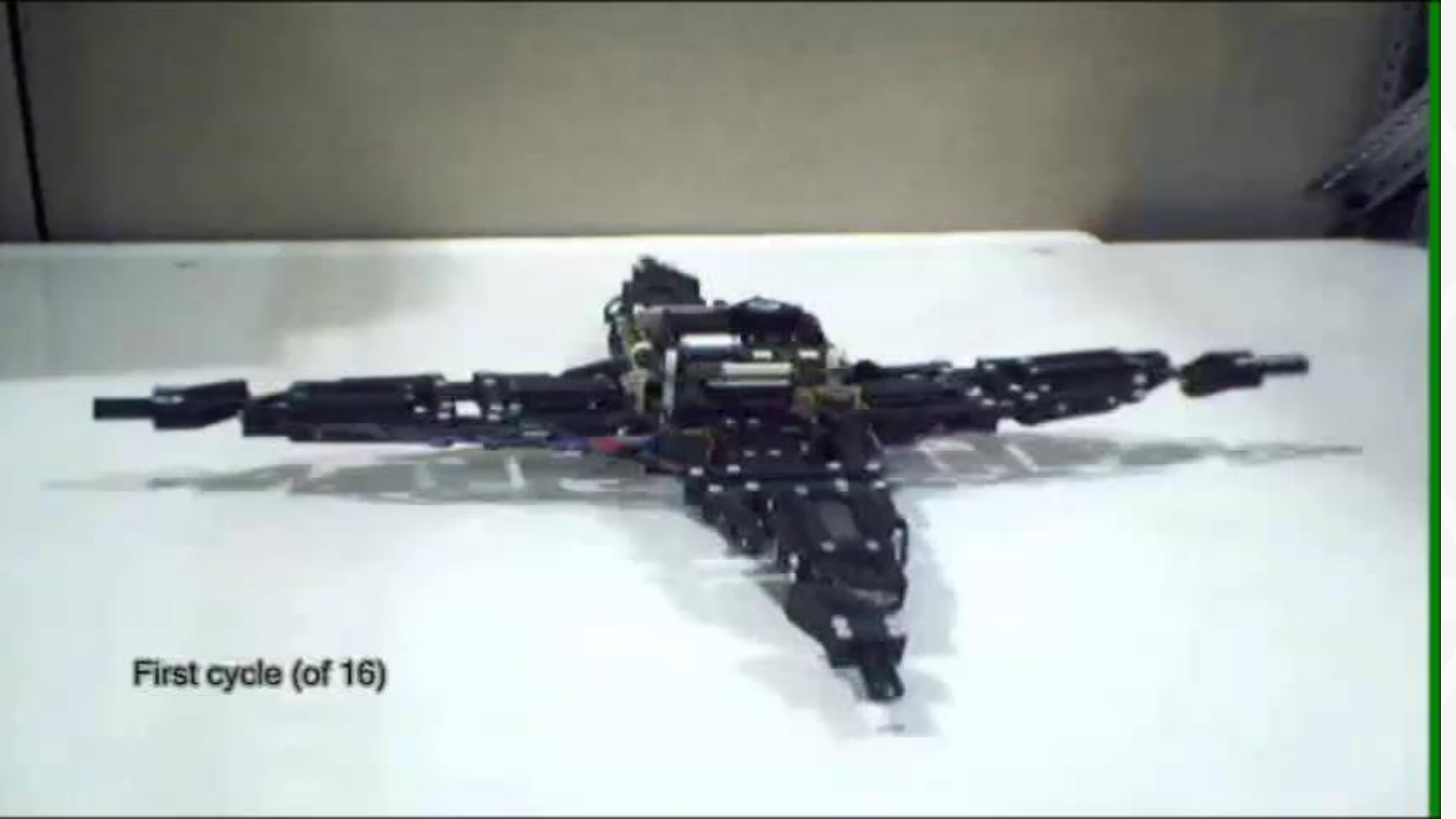


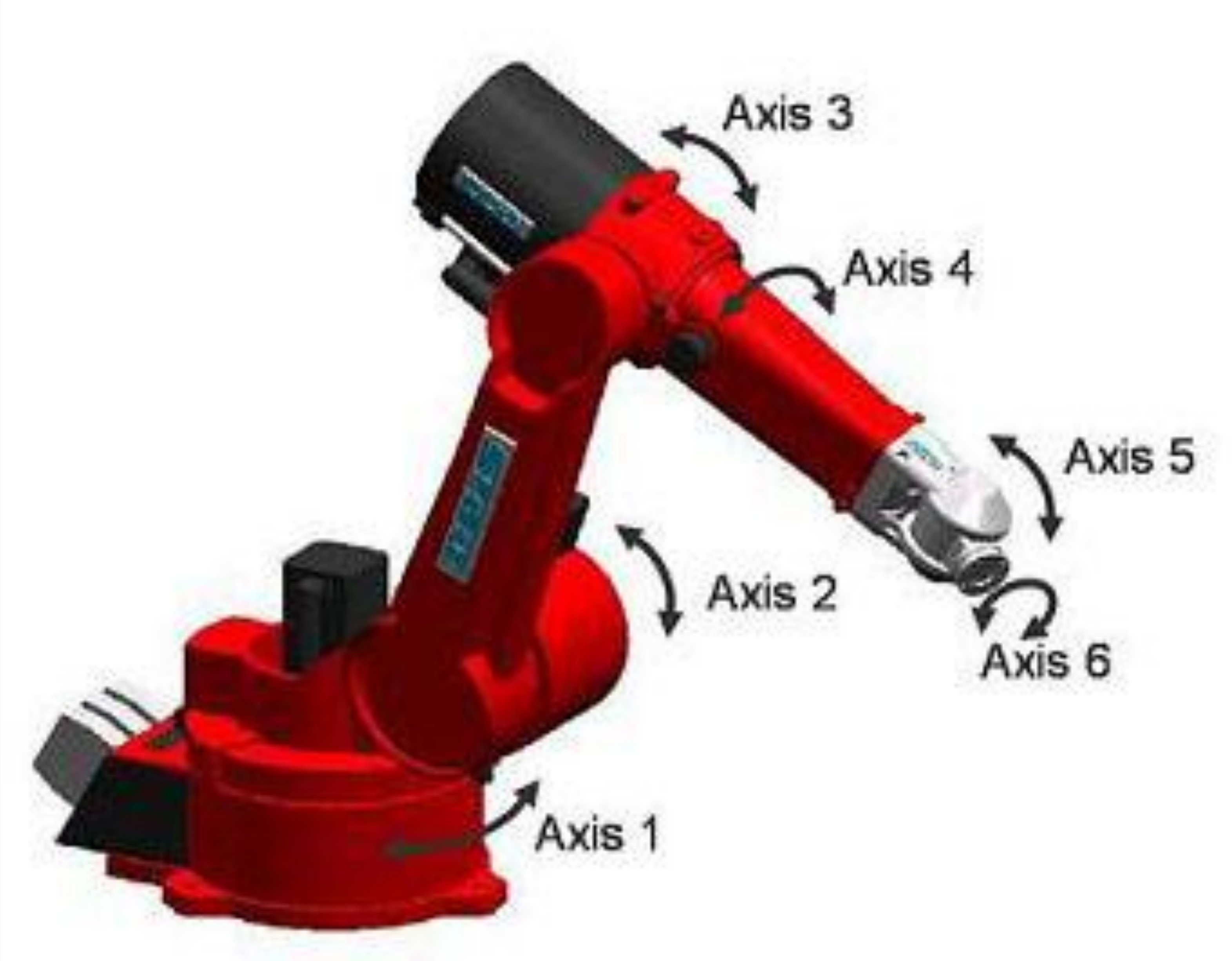
Model of “Self” can be reused In new tasks

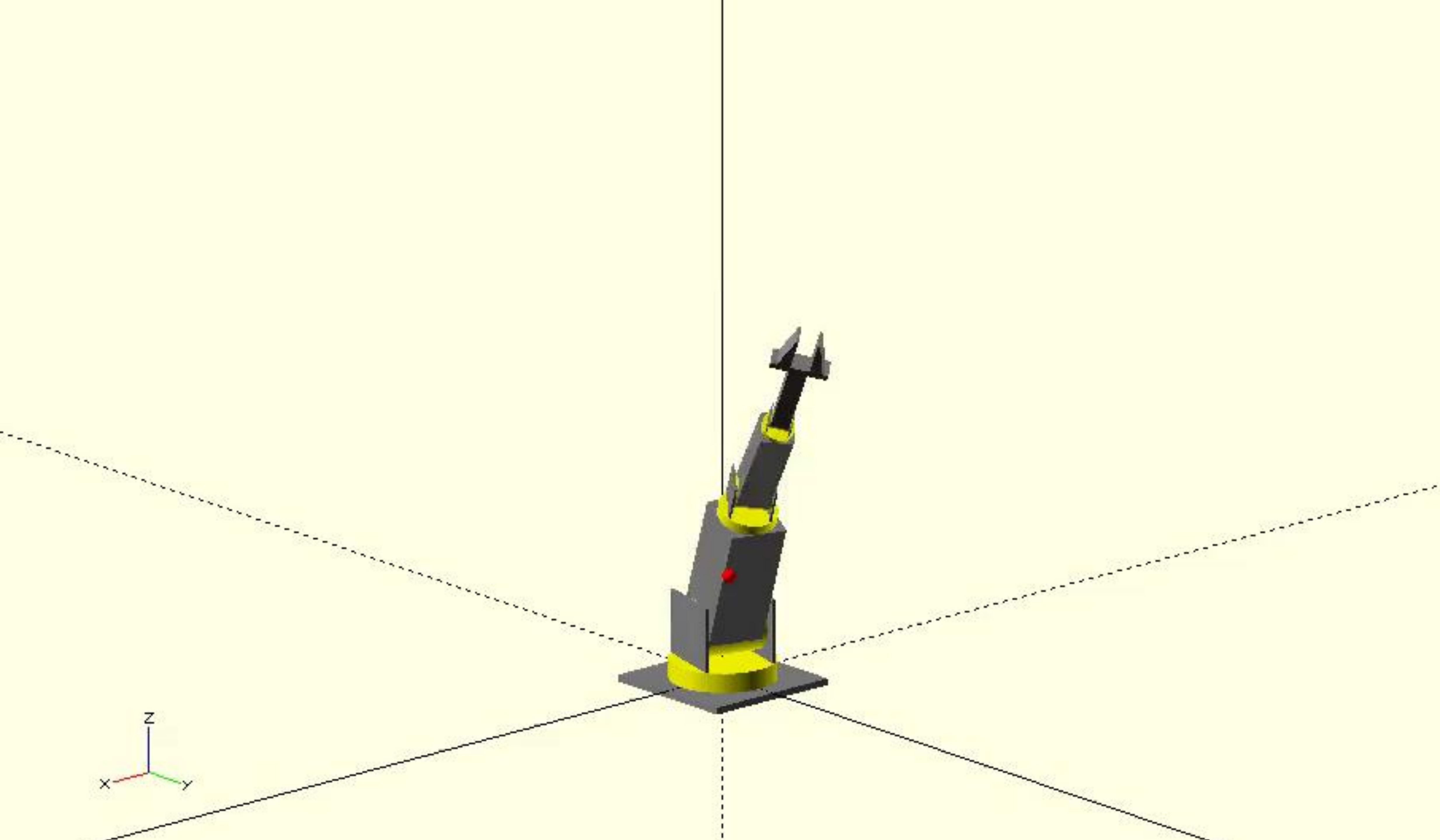
Task can be reused in a modified “self”

Damage Detection

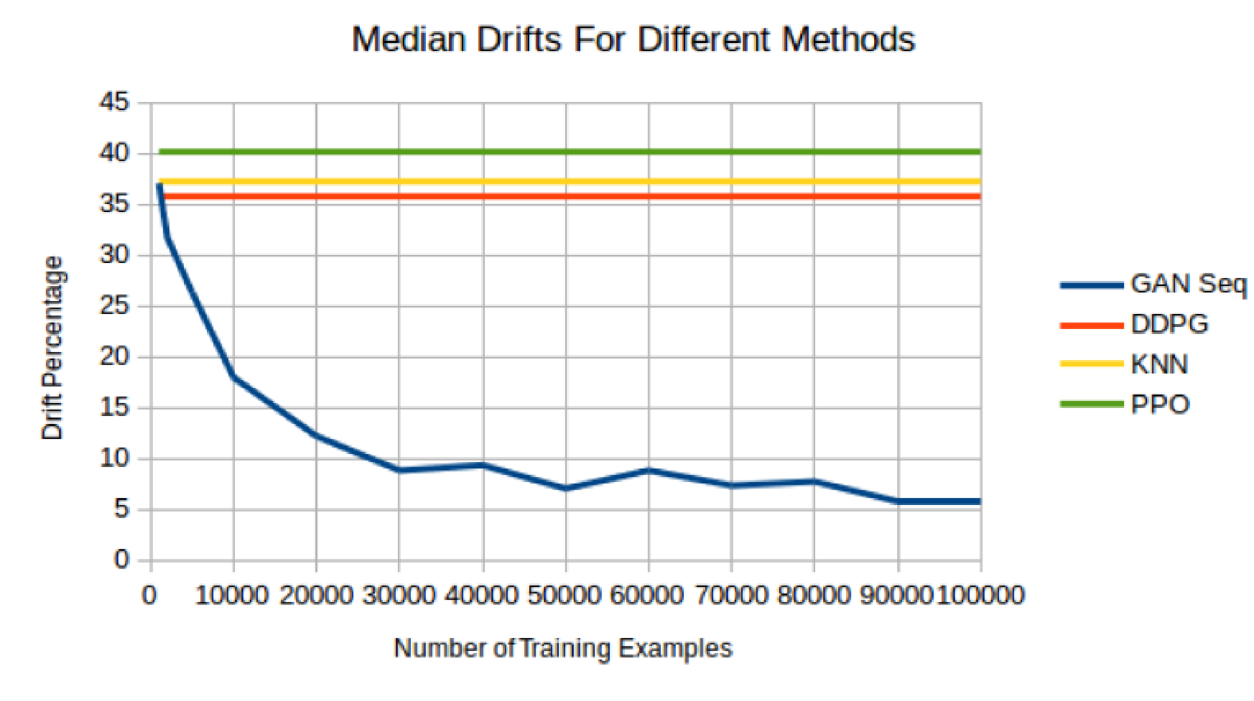

Self-Model synthesis
?
Exploratory Action synthesis
Target Behavior synthesis


First cycle (of 16)


Axis 3
Axis 4
Axis 5
Axis 2
Axis 6
Axis 1



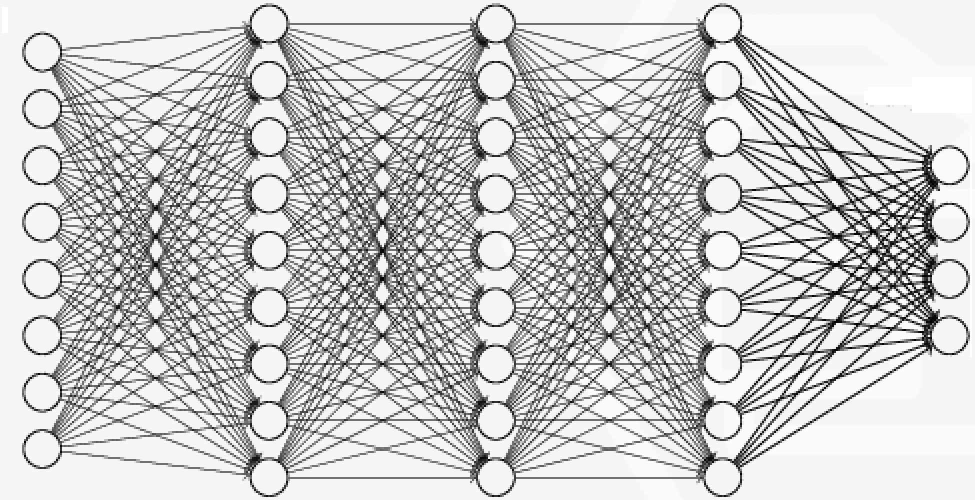
Robert Kwiatkowski

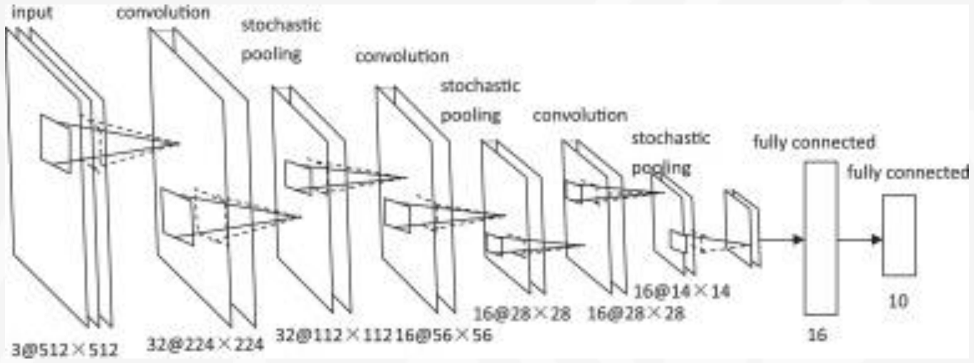


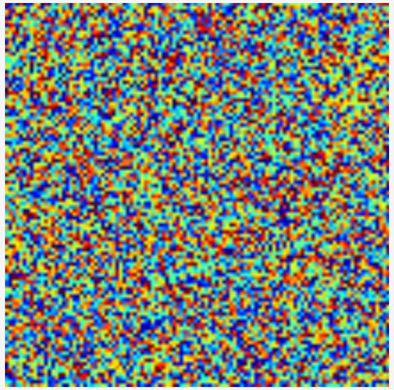
Experiment (Open)	2DoF	4DoF	6DoF
Baseline	1.80%	5.72%	11.7%
Noise	4.11%	9.10%	11.0%
Self-Collision	0.864%	4.17%	6.05%
Noise+Self-Collision		8.50%	9.05%
Experiment (Closed)	**2DoF**	**4DoF**	**6DoF**
Baseline	0.270%	0.887%	1.86%
Noise	1.19%	2.03%	2.26%
Self-Collision	0.098%	0.831%	1.77%
Noise+Self-Collision		1.79%	2.21%

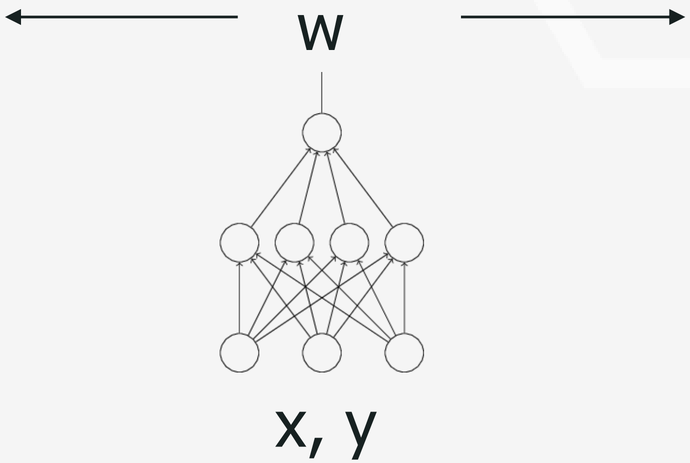
Robert Kwiatkowski, Hod Lipson, (2018) A Self-Modeling Framework for 2D and 3D Articulated Arms, Submitted, IROS

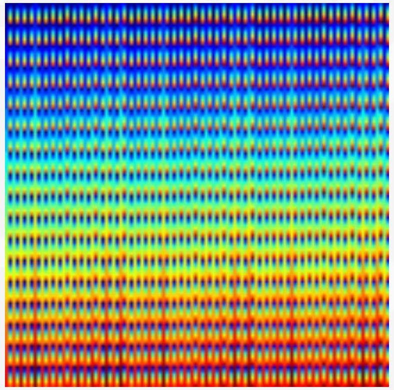
ADAPTING NN ARCHITECTURES











Arbitrary Weight Sharing

Needs Computational Infrastructure

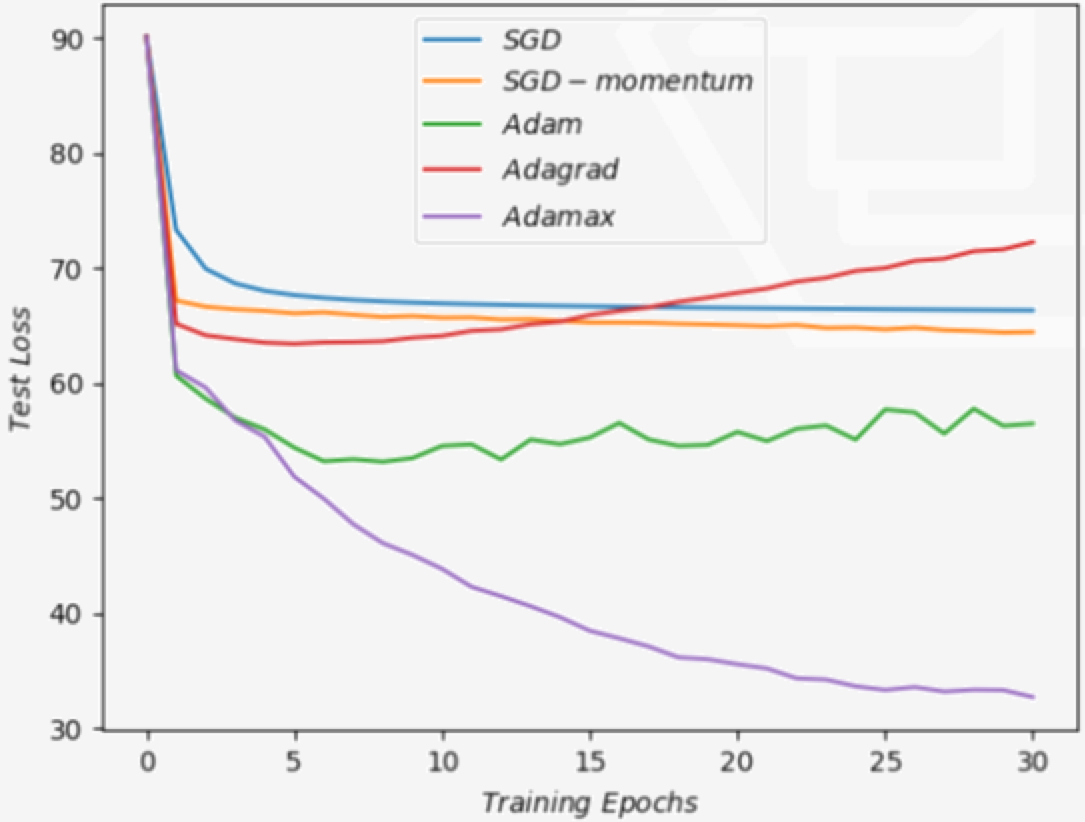
Hyper Networks (Ken Stanley, 201

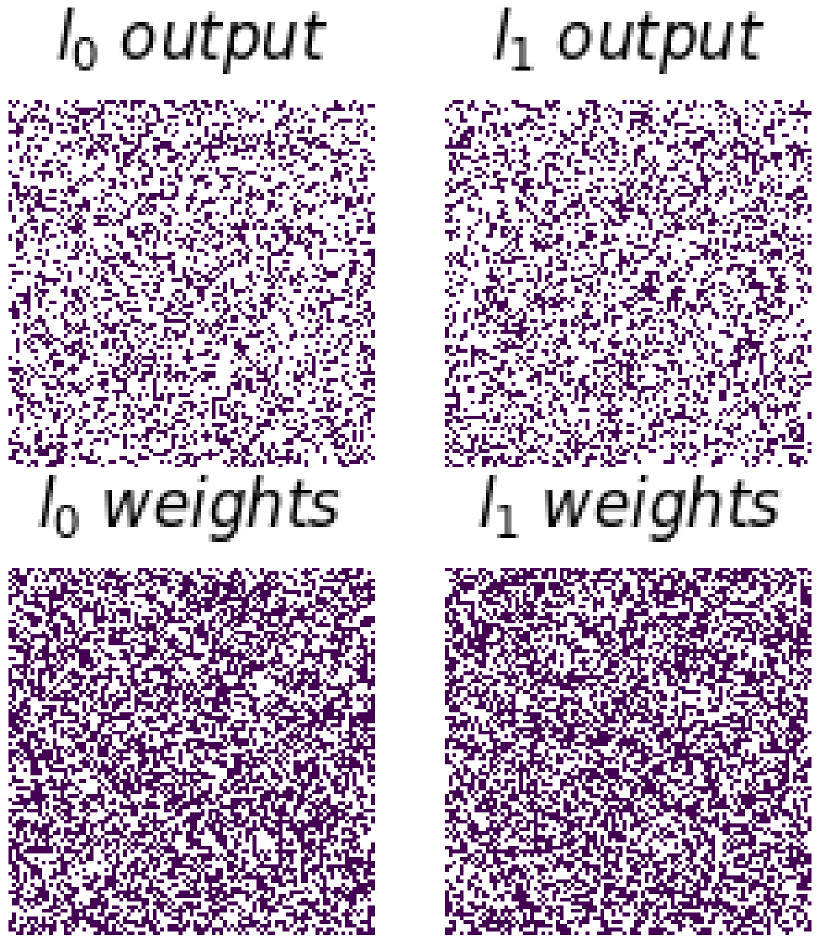
QUINES – SELF REPLICATORS

<u>A Python Quine:</u>

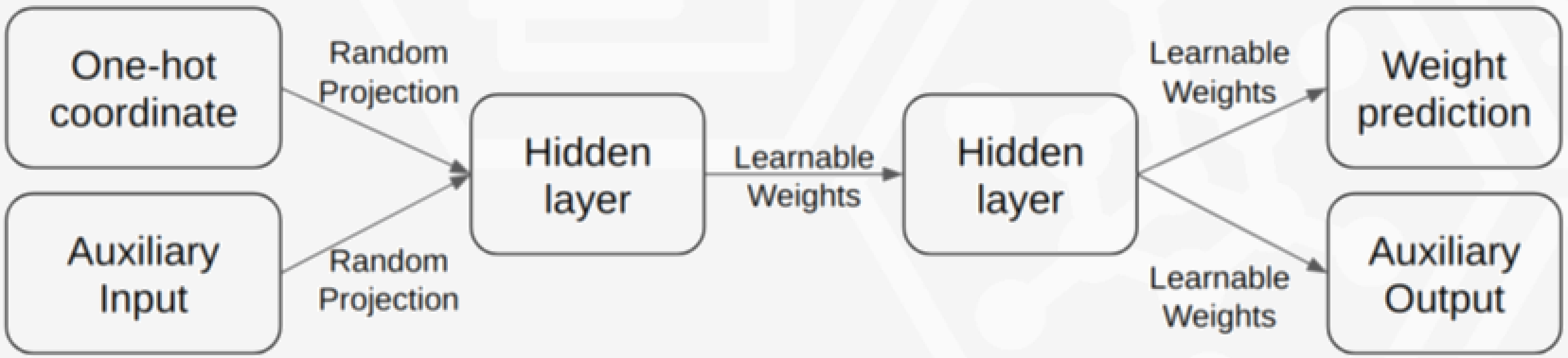
```
s = ’s = %r\nprint(s%%s)’
print(s%s)
```

Is there a *Neural Network Quine*?

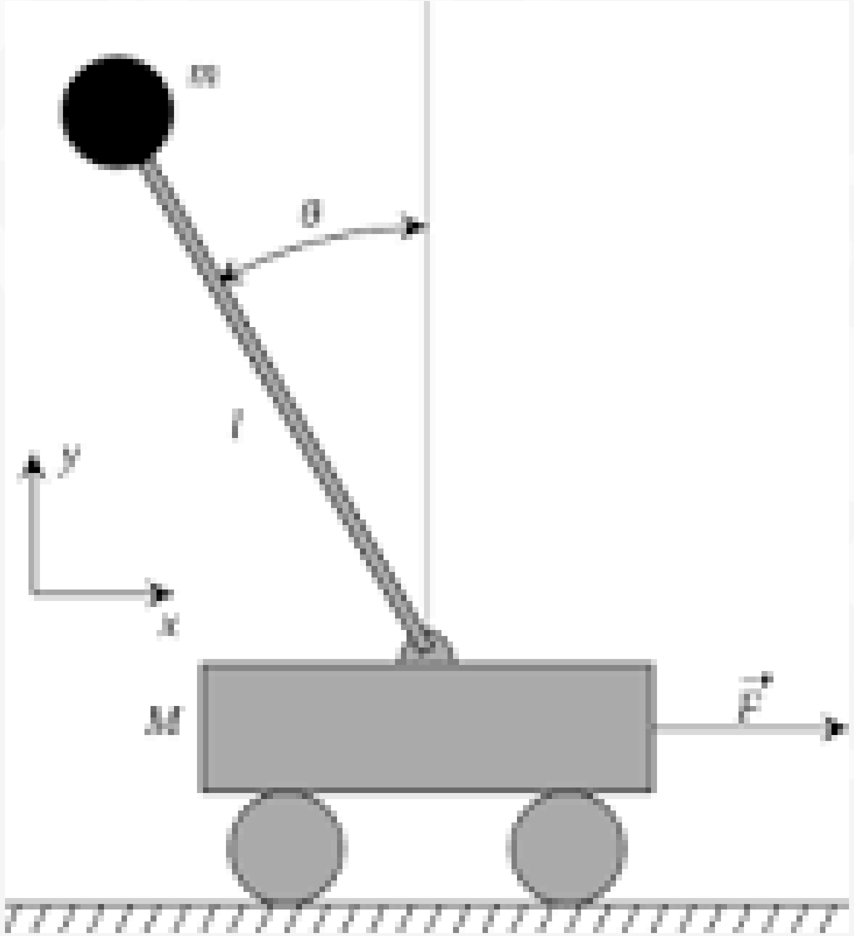




Epoch 10 Test Loss $L_{SR} = 0.86$



Chang, Lipson "A Neural Network Quine", (ALIFE) 2018

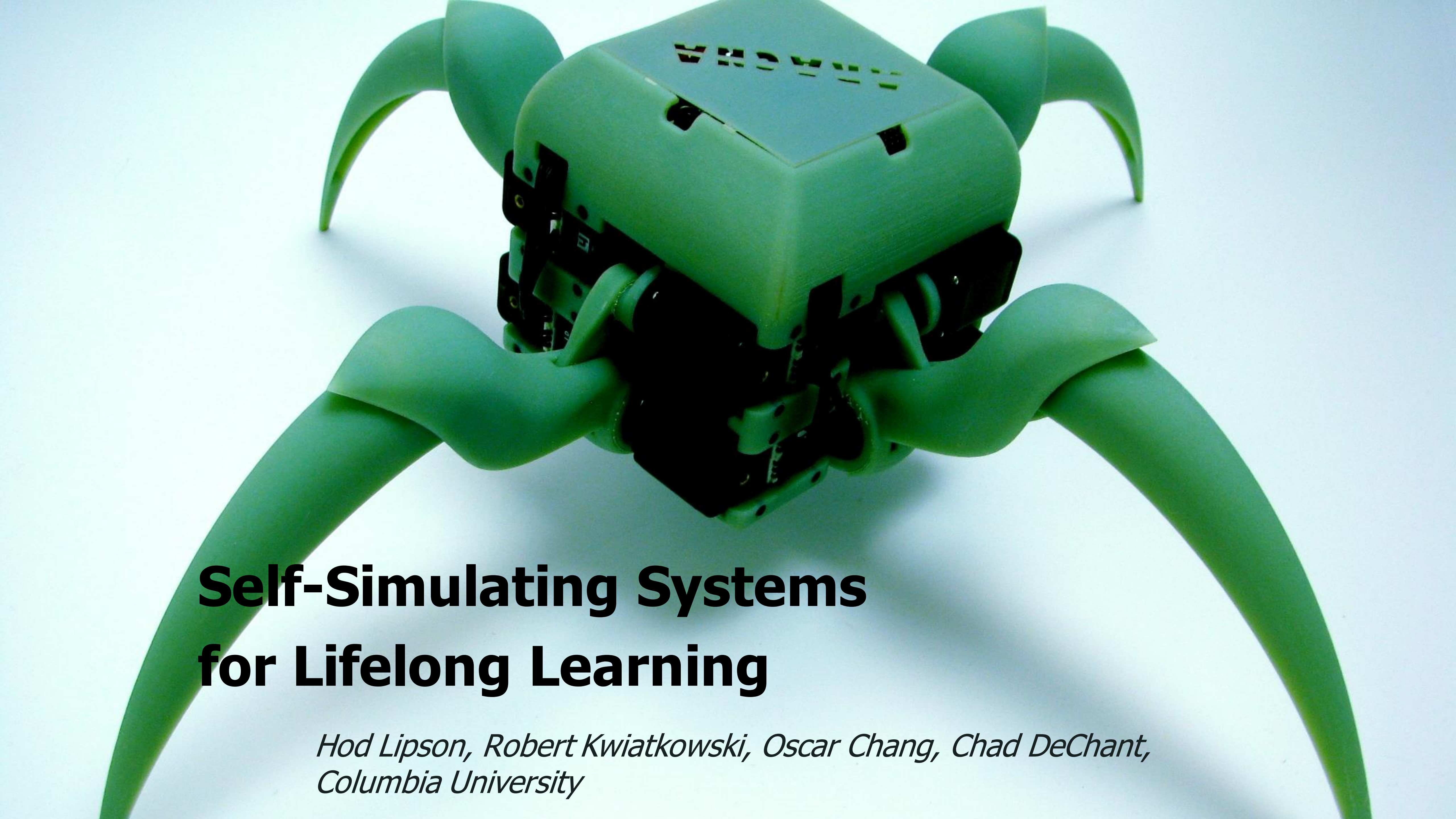


m
θ
l
y
x
M
F

Auto-Generated NN for cart-pole balancing

Bucket	Trainset Size	(Mean, Std.) of Average Steps in Trainset	(Mean, Std.) of Average Steps in Generated Samples
1 – 50 steps	687	37.1, 8.7	17.3, 17.8
51 – 100 steps	539	68.4, 14.9	37.7, 35.4
101–150 steps	219	126.7, 16.0	75.6, 58.0
151–200 steps	555	186.7, 15.1	124.0, 58.0

Chang, Lipson "Learning a Generative Model For Neural Networks", (Alife) 2018



Self-Simulating Systems for Lifelong Learning

Hod Lipson, Robert Kwiatkowski, Oscar Chang, Chad DeChant,
Columbia University

Microsystems Technology Office

ERI ELECTRONICS RESURGENCE INITIATIVE

SUMMIT

2018 | SAN FRANCISCO, CA | **JULY 23-25**