Jeff Clune

Harris Associate Professor, Computer Science

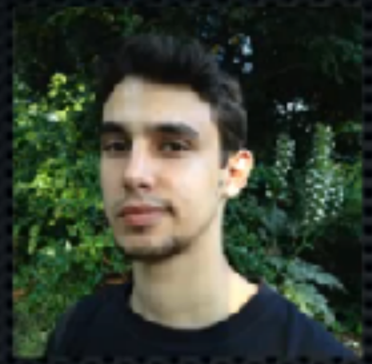
UNIVERSITY OF WYOMING

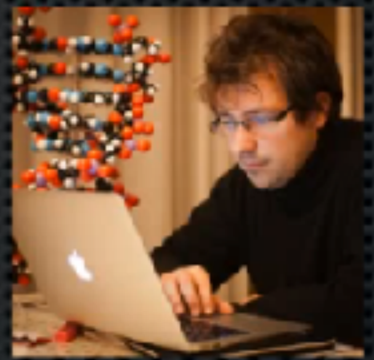
Senior Research Manager (Staff Scientist)

UBER AI Labs

Robots that adapt like animals

2015





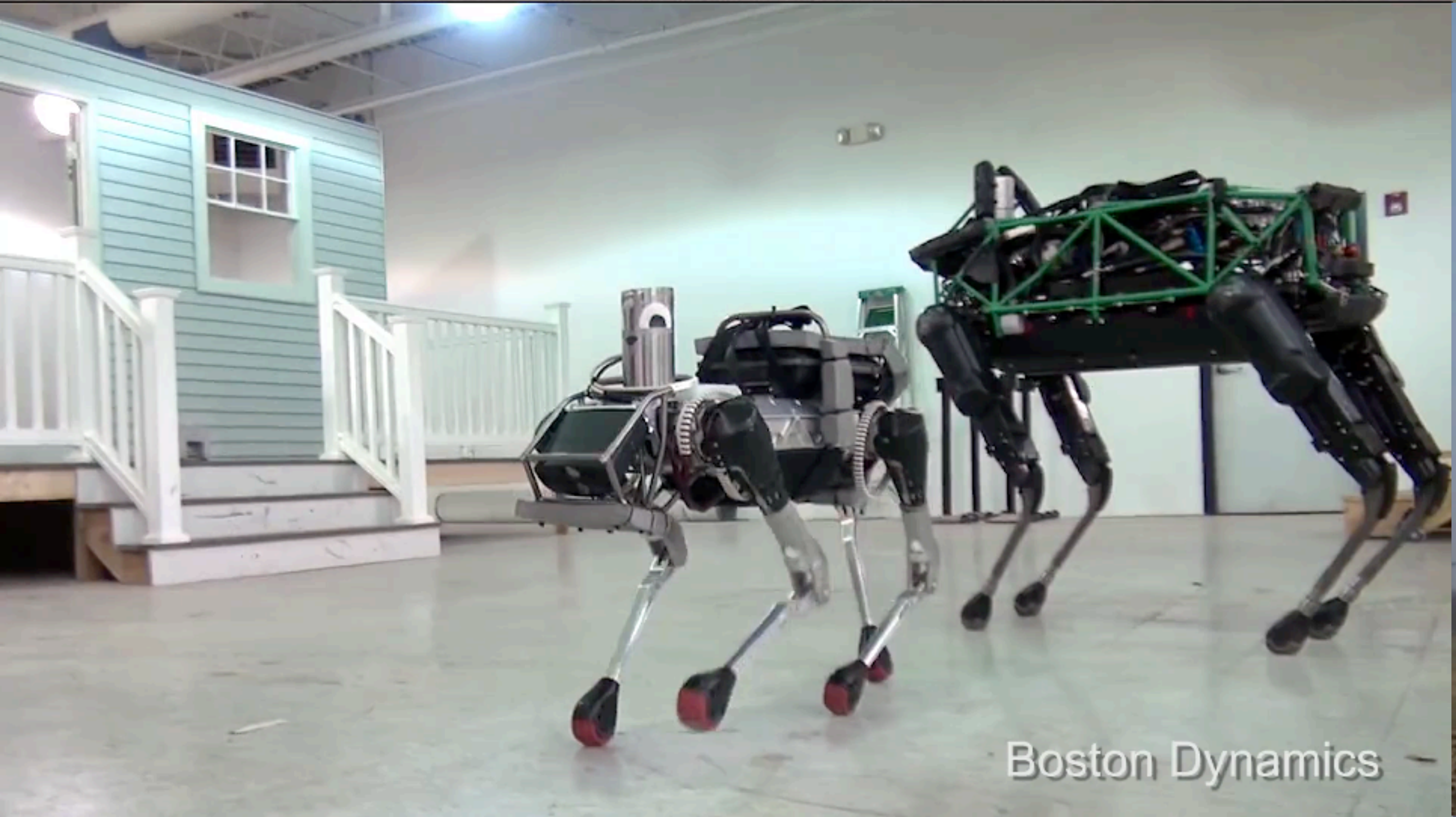
Antoine Cully
UPMC Université
France

Jeff Clune
University of Wyoming
USA

Danesh Tarapore
UPMC Université
France

Jean-Baptiste Mouret
UPMC Université
France


Boston Dynamics
Boston Dynamics

Boston Dynamics

Boston Dynamics



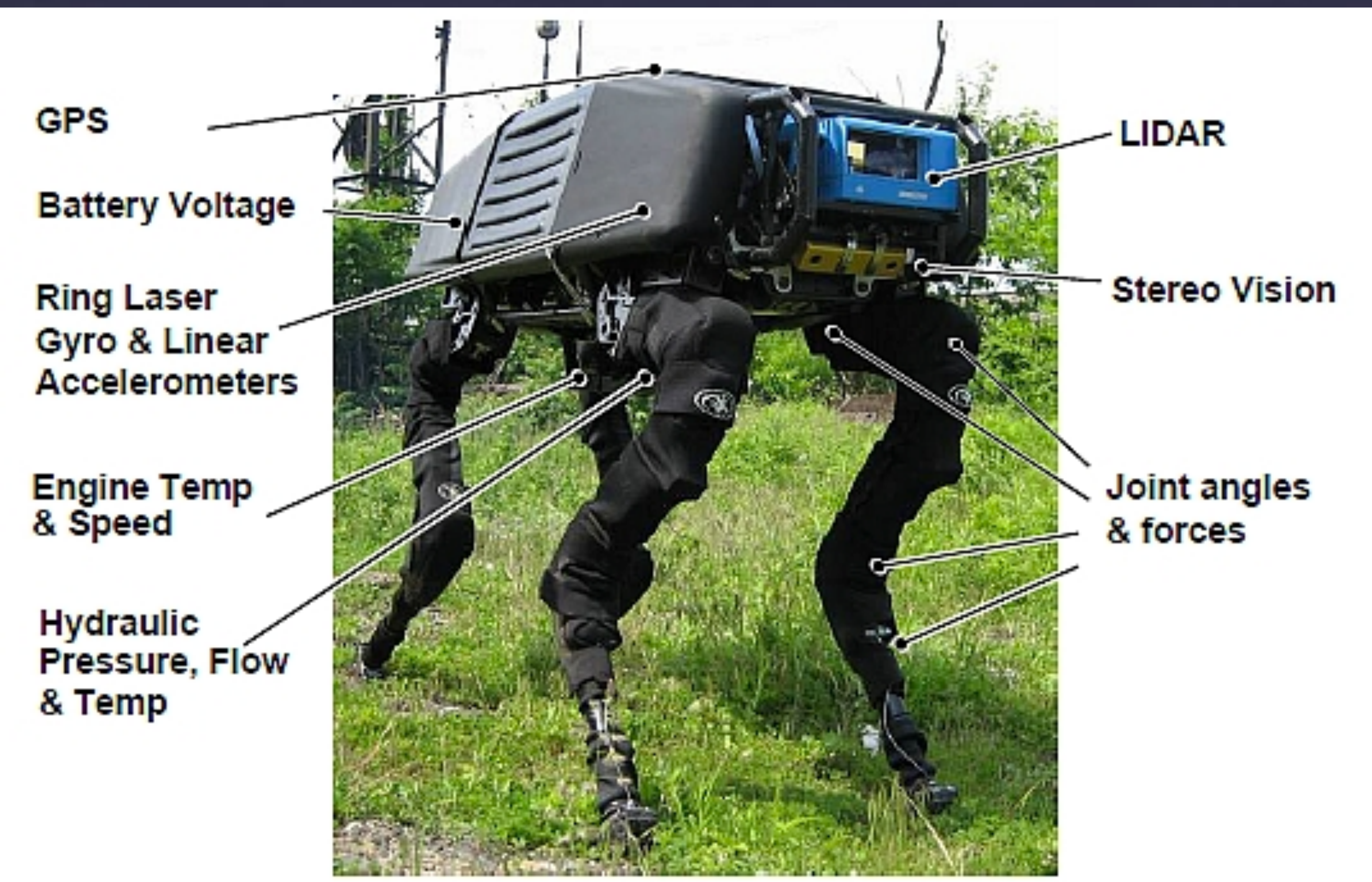


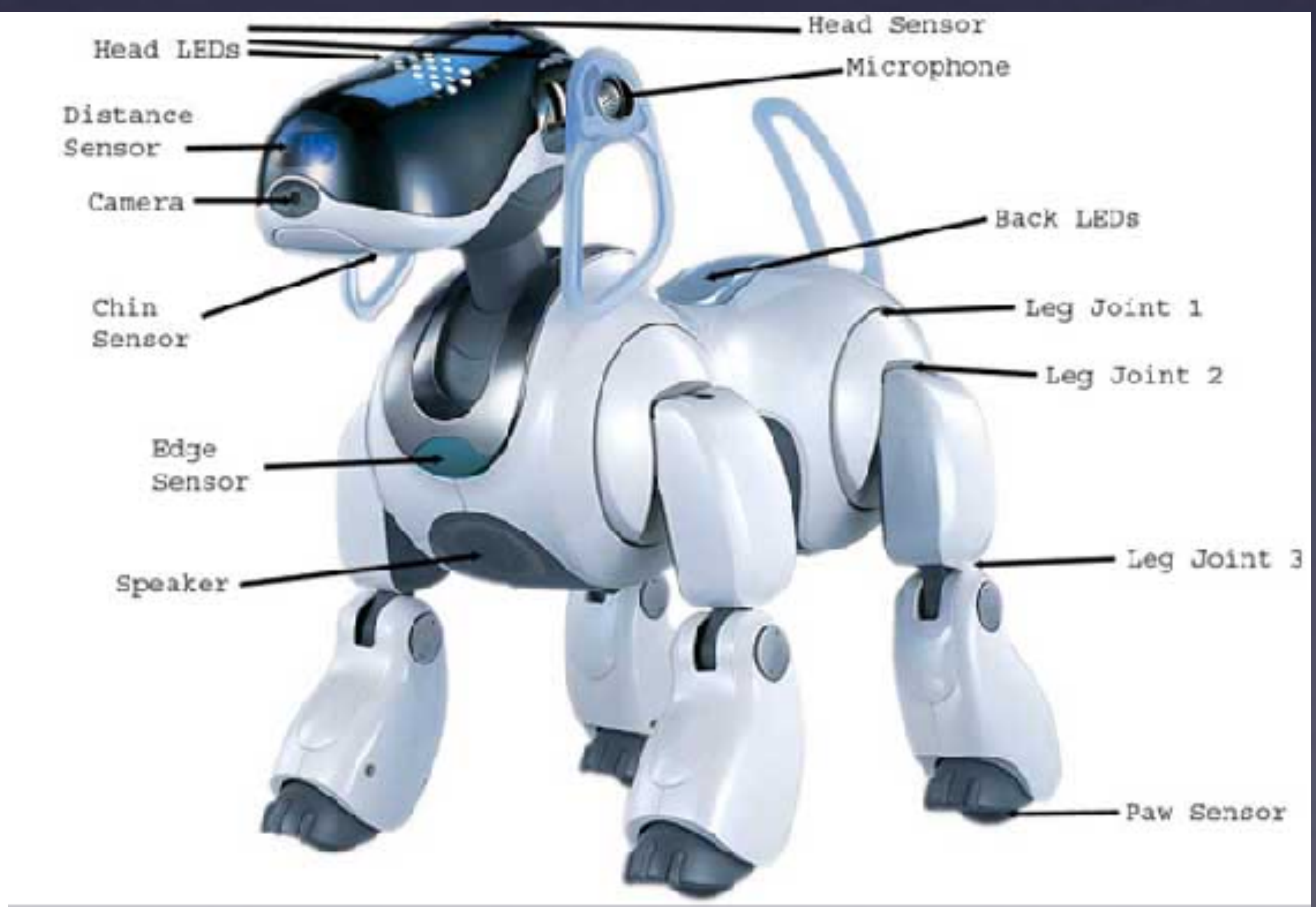

iRobot


Damage Recovery

Classic Approach to Damage Recovery

- Large suite of self-diagnosis sensors
- IF diagnosis is successful, choose pre-programmed response from large library
- Problems: expensive, error-prone, manual, doesn’t scale (impractical to have plan for each case)



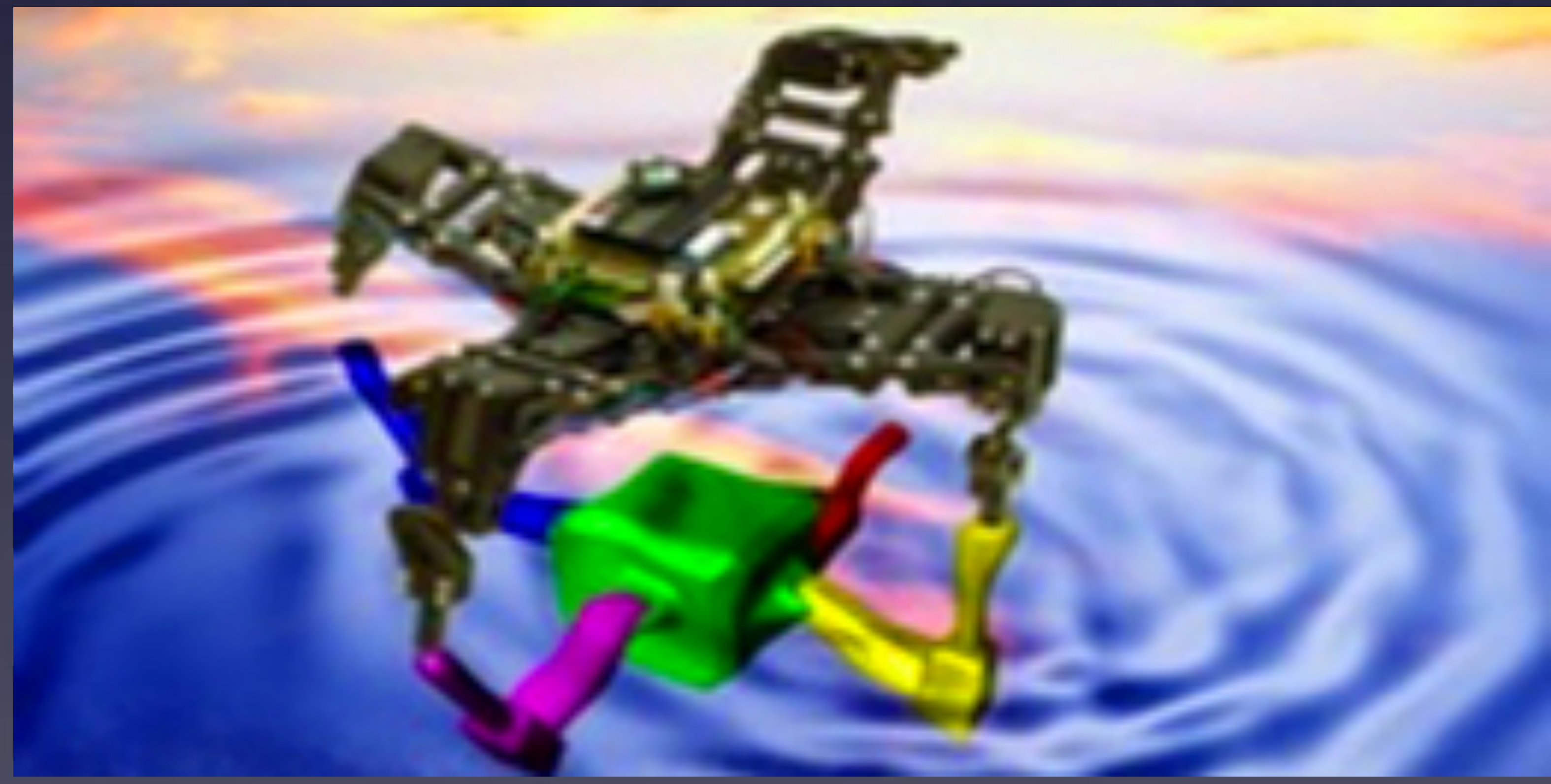


Modern, Learning-Based Approaches

- Simple robots (low-dimensional state & action spaces)
- Require lots of real-world trials

Yosinski et al. 2013

Kohl & Stone 2004



Bongard et al. 2006

Animals

- Have intuitions about different ways to move
- Conduct a few, intelligent tests
- Pick a behavior that works despite injury

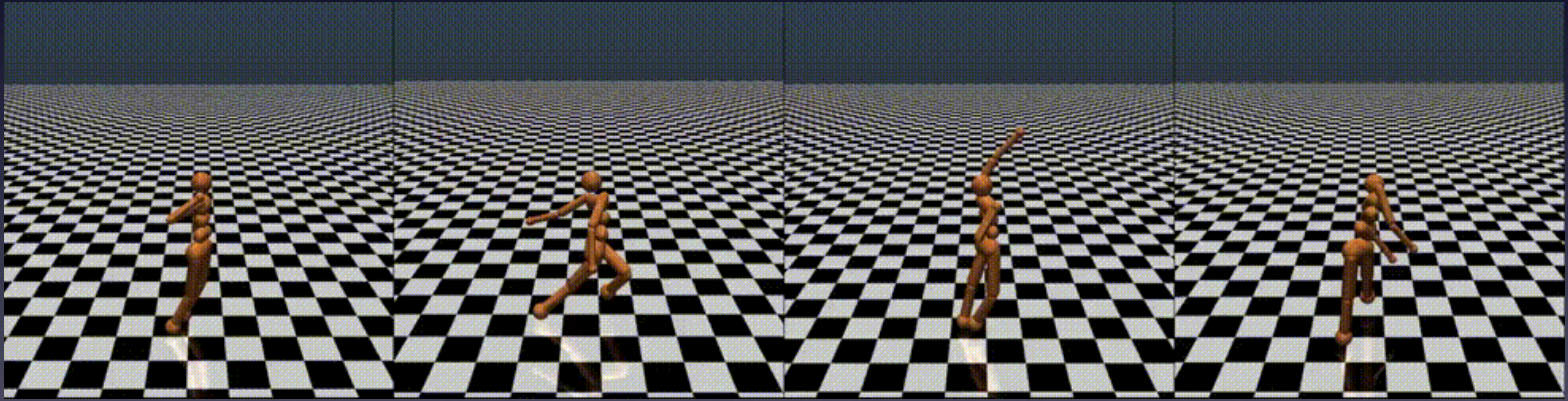
Robots that Adapt Like Animals

- Have intuitions about different ways to move
- Conduct a few, intelligent tests
- Pick a behavior that works despite injury

intuitions about different ways to move → few, intelligent tests → pick one that works despite injury

intuitions about
different ways to move

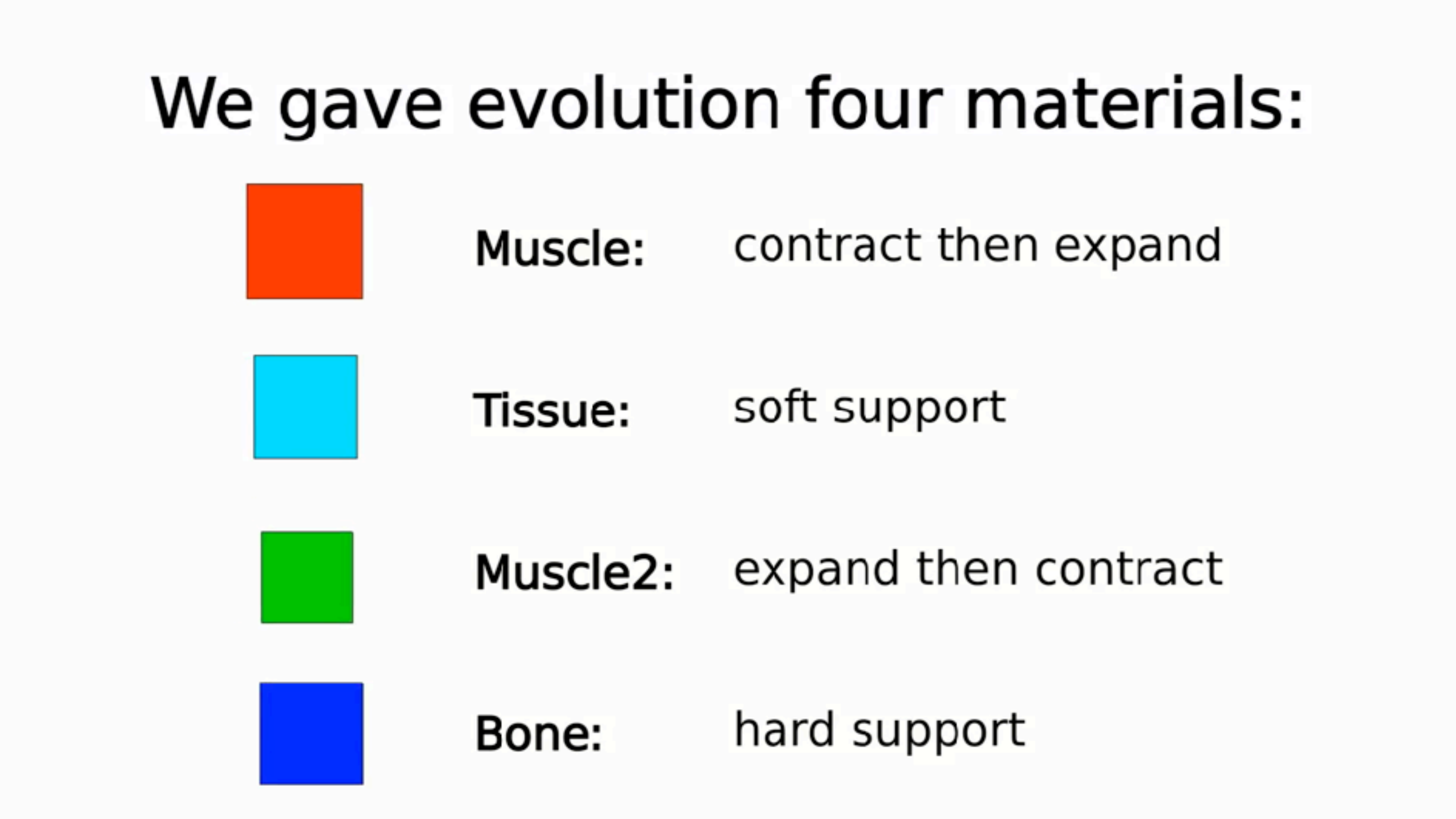
- Traditional machine learning methods produce little diversity

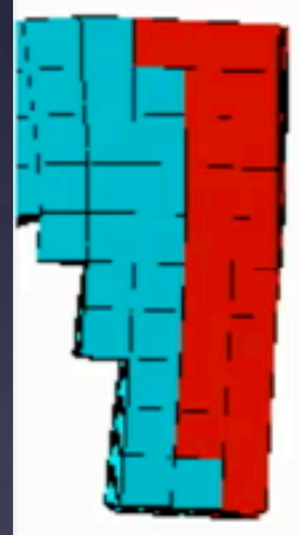


Salimans, Ho, Chen, Sidor, Sutskever 2017

intuitions about
different ways to move

- Traditional machine learning methods produce little diversity





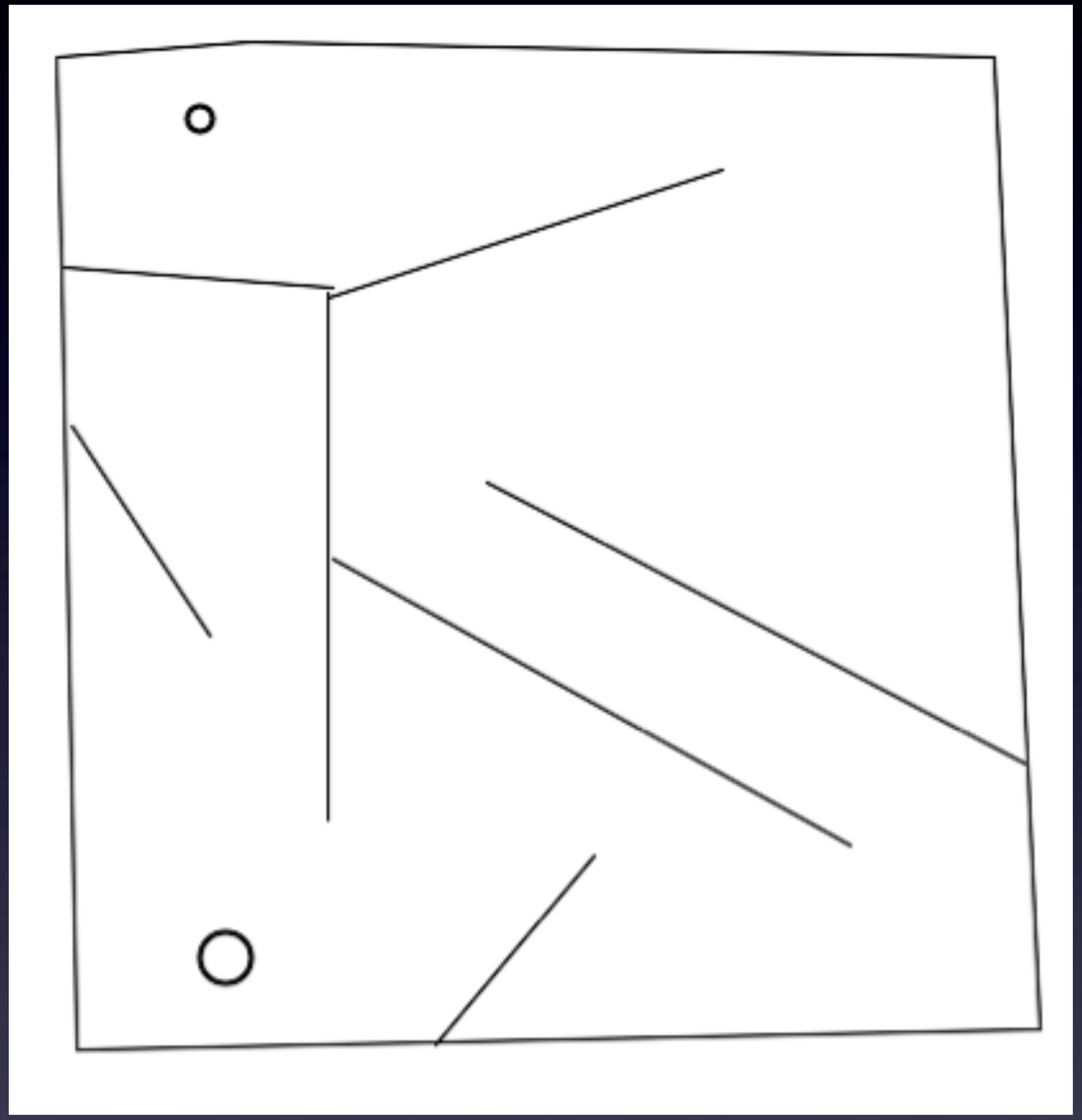
Cheney, MacCurdy, Clune, Lipson 2013

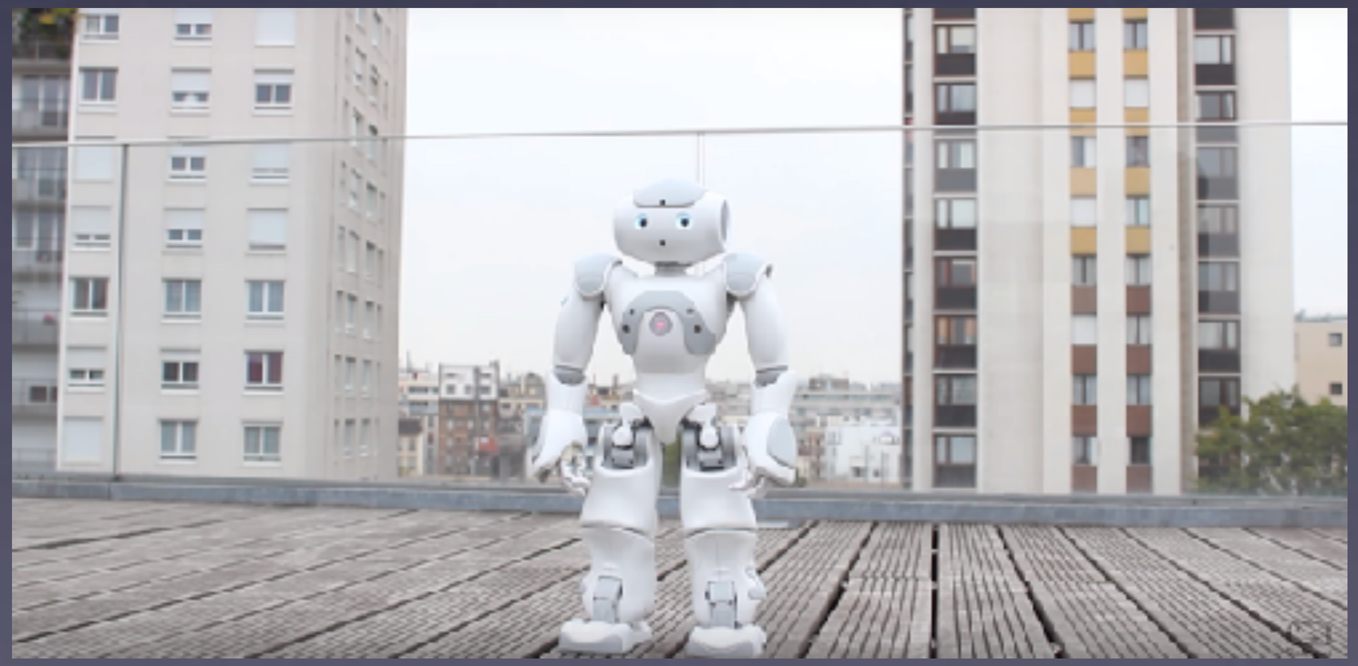
intuitions about
different ways to move

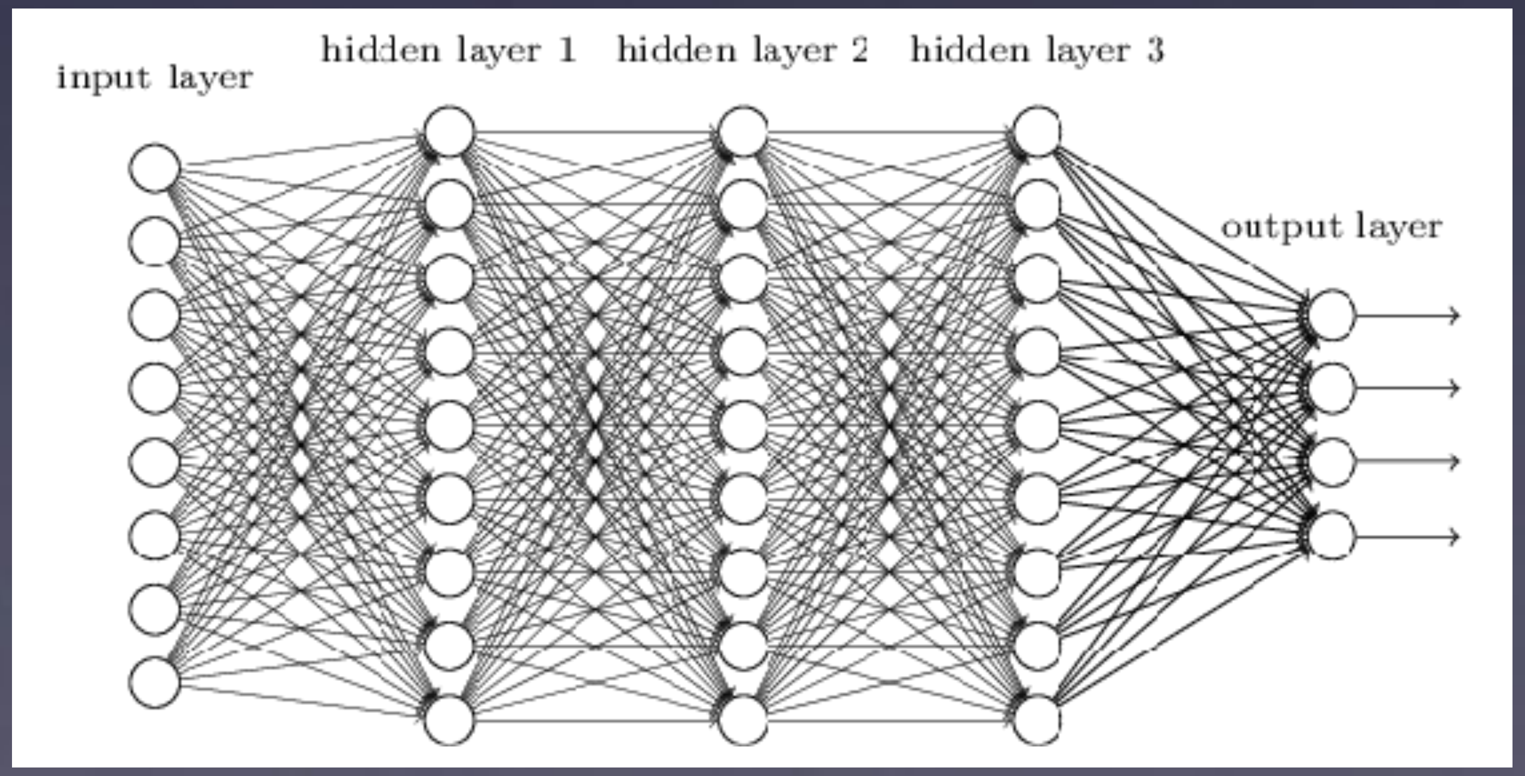
- Traditional machine learning methods produce little diversity
- Need an algorithm good at producing
  - a diverse set of high-performing agents (policies)
  - “Quality Diversity algorithms”

Promoting Diversity

- Old idea in optimization
  - but usually diversity in parameter space
    - might not produce new behaviors
    - deception remains
- Much better in behavior space
  - e.g. Lehman & Stanley 2011
  - imagine a robot in a city







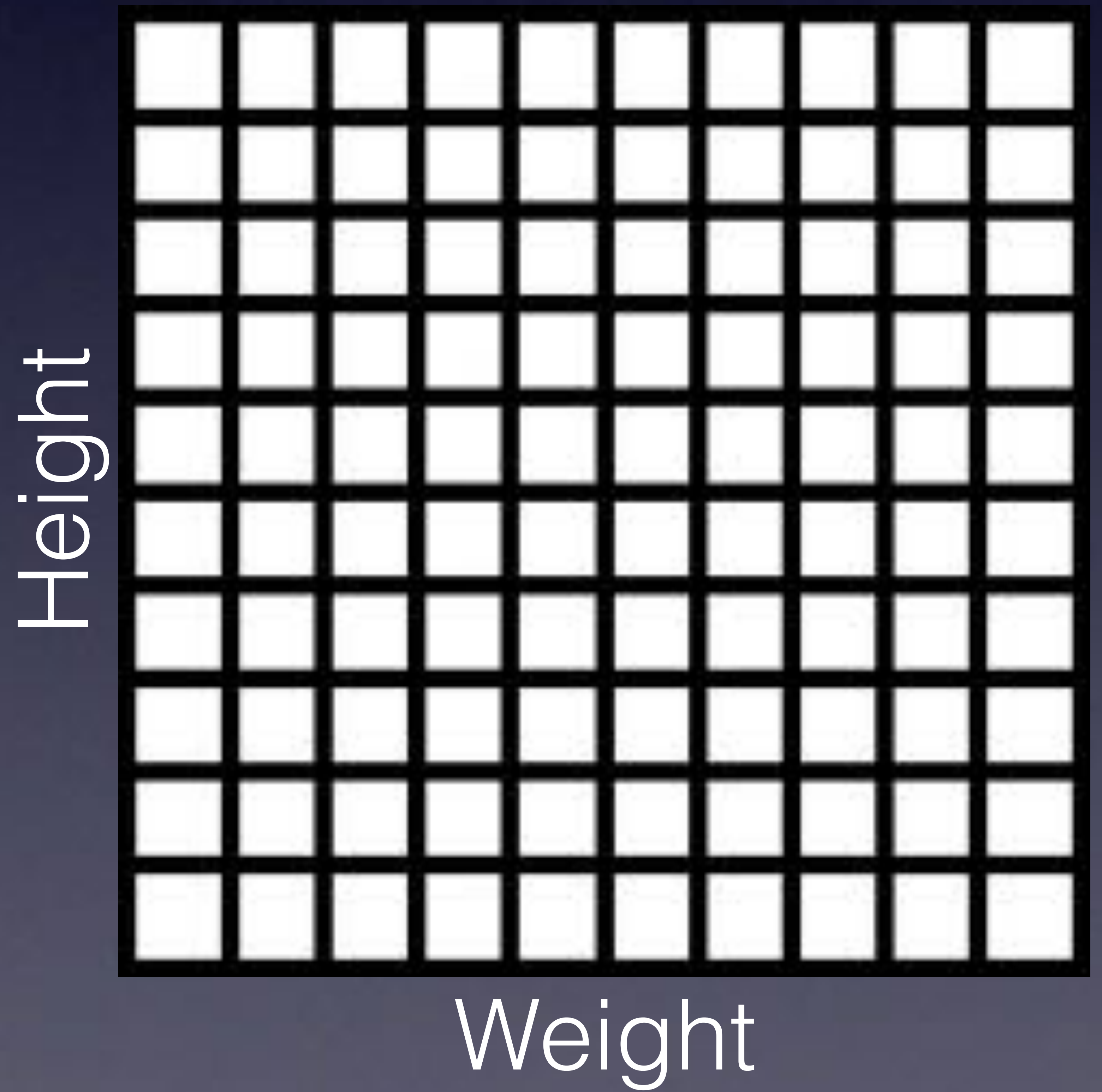
Challenge: Diversity & Performance

- Quality diversity algorithms
  - Novelty Search + Local Competition (Lehman & Stanley)
  - MAP-Elites (Mouret & Clune)

MAP-Elites

Mouret & Clune 2015, arXiv

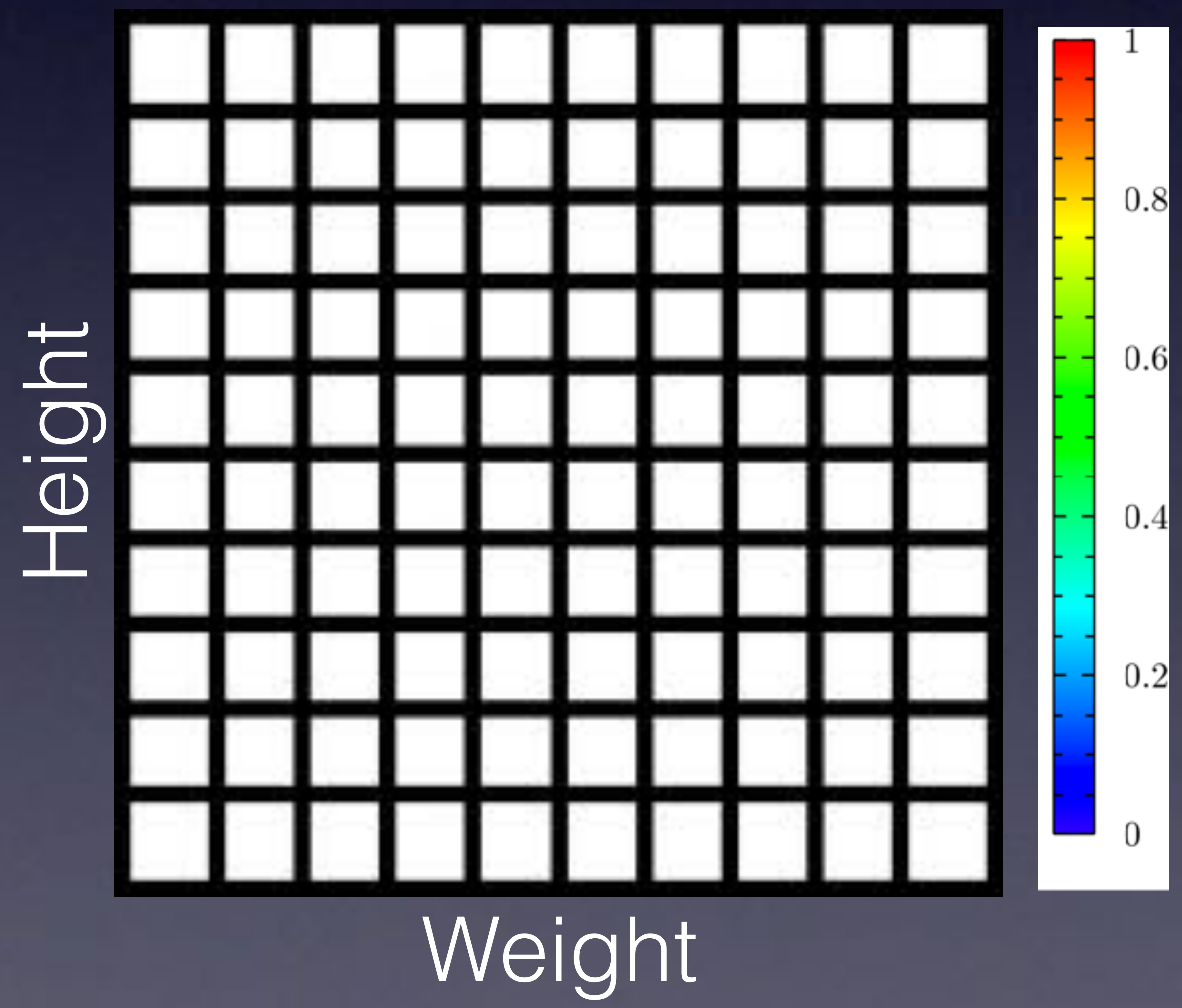
- Multi-dimensional Archive of Phenotypic Elites
  - Choose dimensions of interest in behavior space
  - Discretize
  - Mutate, locate, replace if better, repeat



MAP-Elites

Mouret & Clune 2015, arXiv

- Multi-dimensional Archive of Phenotypic Elites
  - Choose dimensions of interest in behavior space
  - Discretize
  - Mutate, locate, replace if better, repeat



MAP-Elites

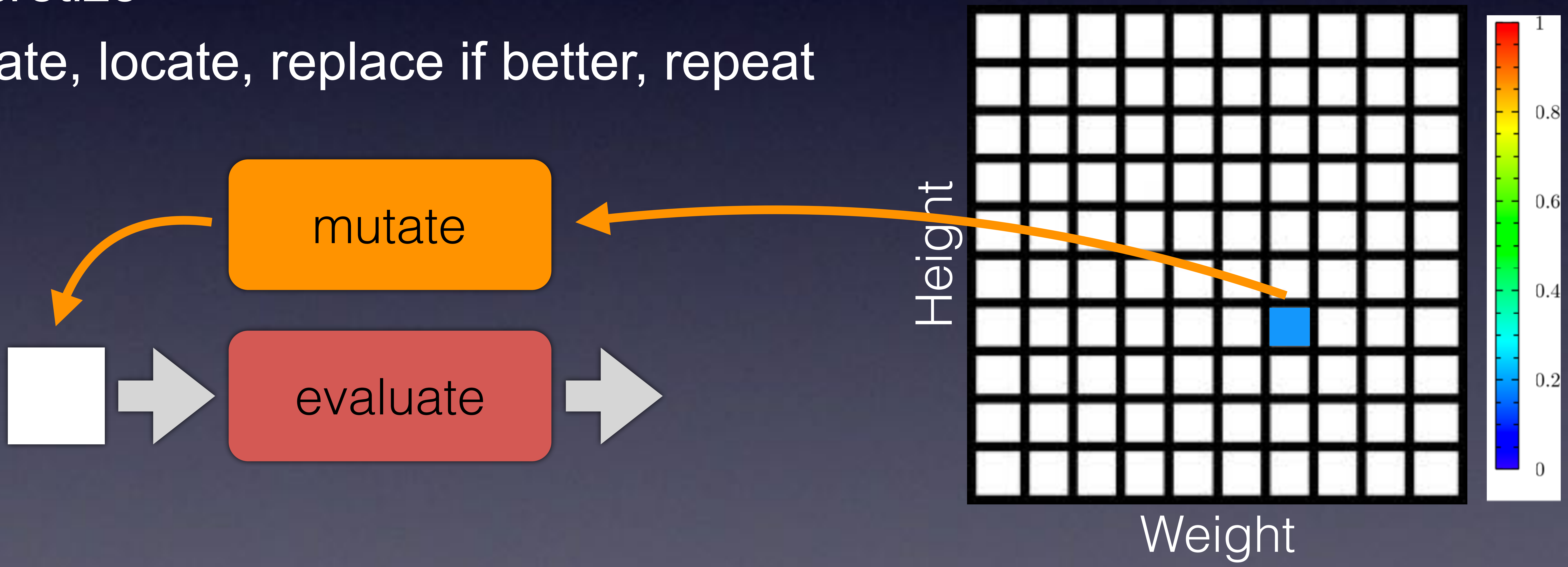
Mouret & Clune 2015, arXiv

- Multi-dimensional Archive of Phenotypic Elites
  - Choose dimensions of interest in behavior space
  - Discretize
  - Mutate, locate, replace if better, repeat

MAP-Elites

Mouret & Clune 2015, arXiv

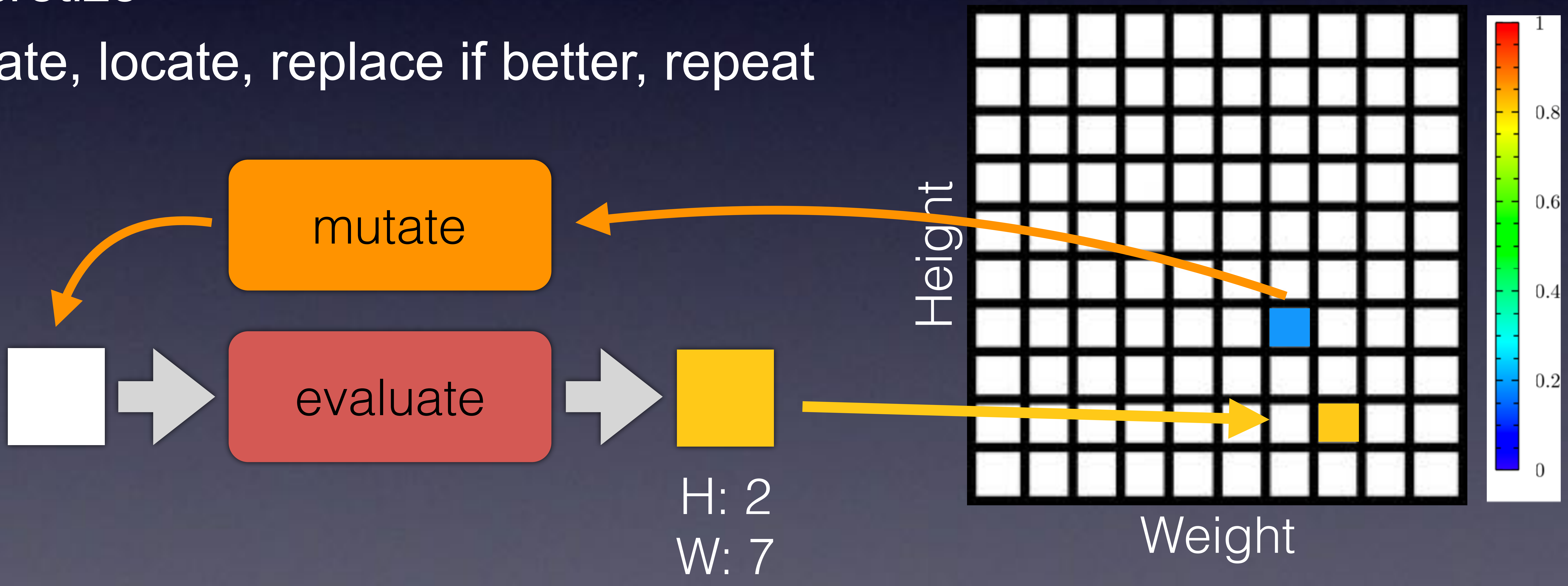
- Multi-dimensional Archive of Phenotypic Elites
  - Choose dimensions of interest in behavior space
  - Discretize
  - Mutate, locate, replace if better, repeat



MAP-Elites

Mouret & Clune 2015, arXiv

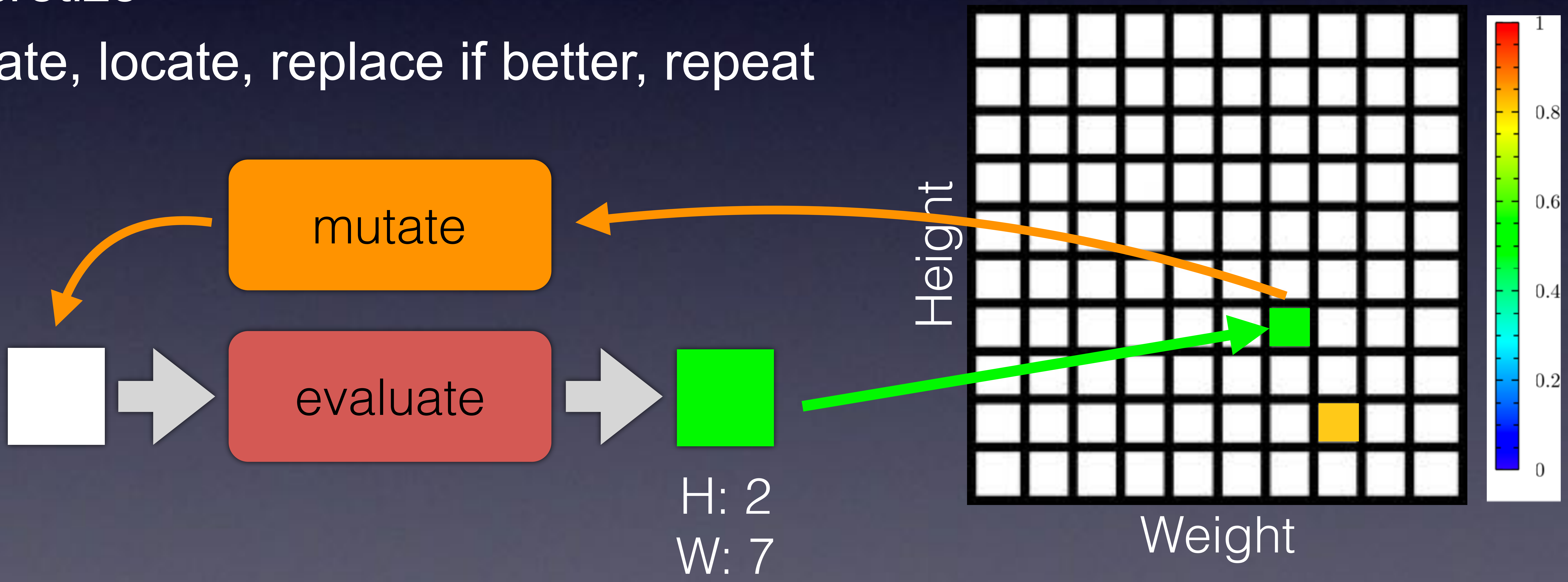
- Multi-dimensional Archive of Phenotypic Elites
  - Choose dimensions of interest in behavior space
  - Discretize
  - Mutate, locate, replace if better, repeat



MAP-Elites

Mouret & Clune 2015, arXiv

- Multi-dimensional Archive of Phenotypic Elites
  - Choose dimensions of interest in behavior space
  - Discretize
  - Mutate, locate, replace if better, repeat

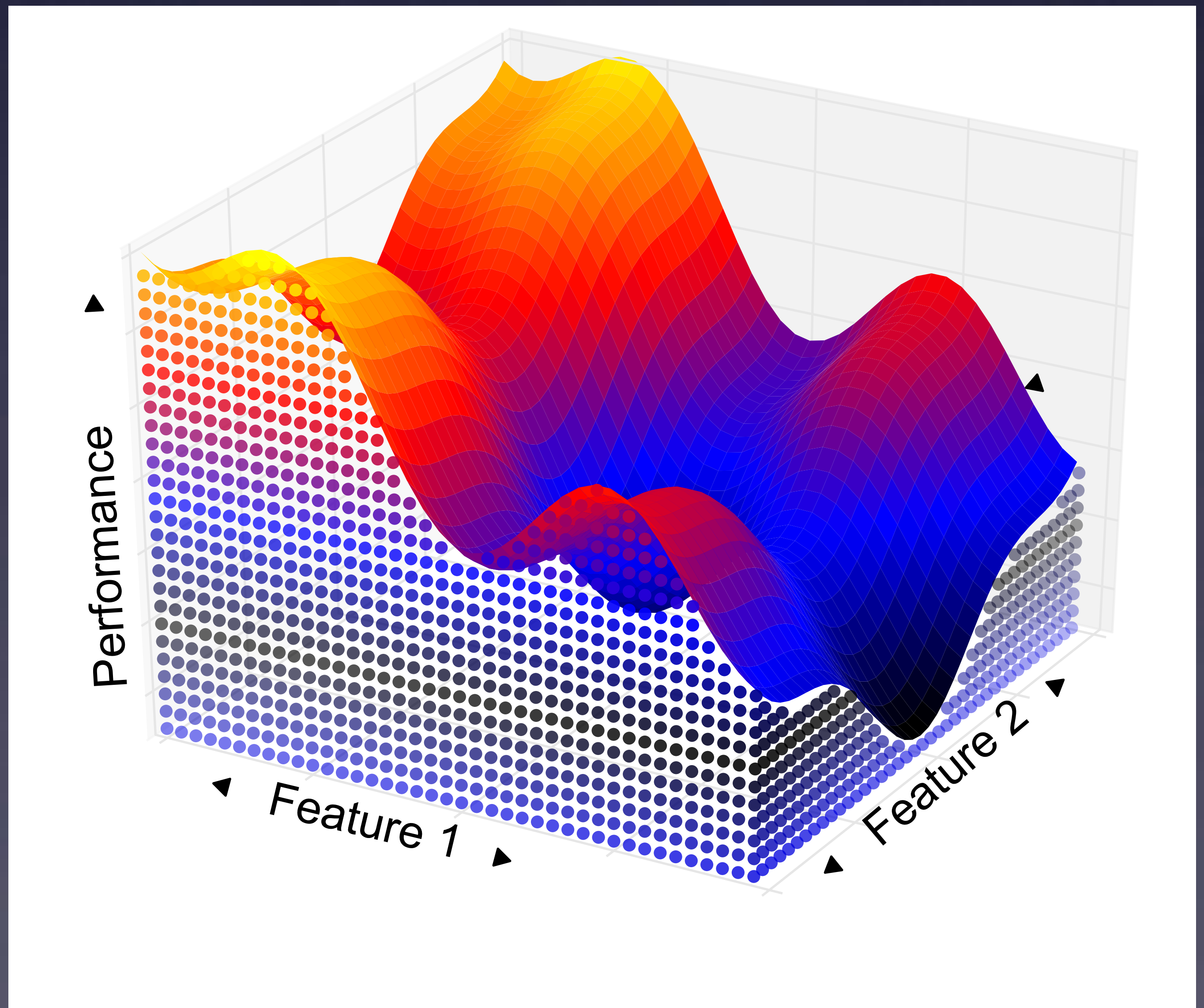


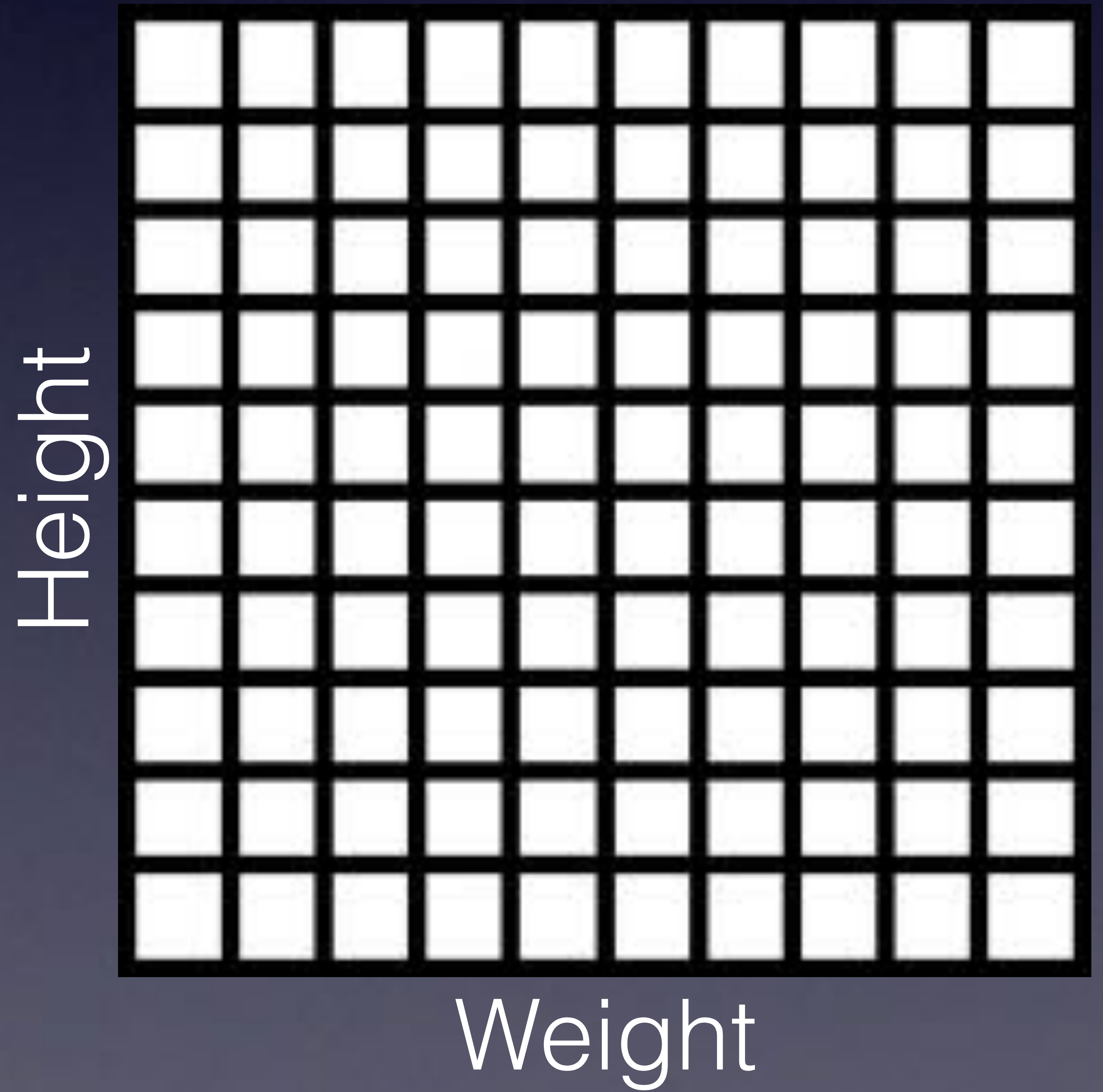
MAP-Elites

Mouret & Clune 2015, arXiv

- Multi-dimensional Archive of Phenotypic Elites
  - Choose dimensions of interest in behavior space
  - Discretize
  - Mutate, locate, replace if better, repeat

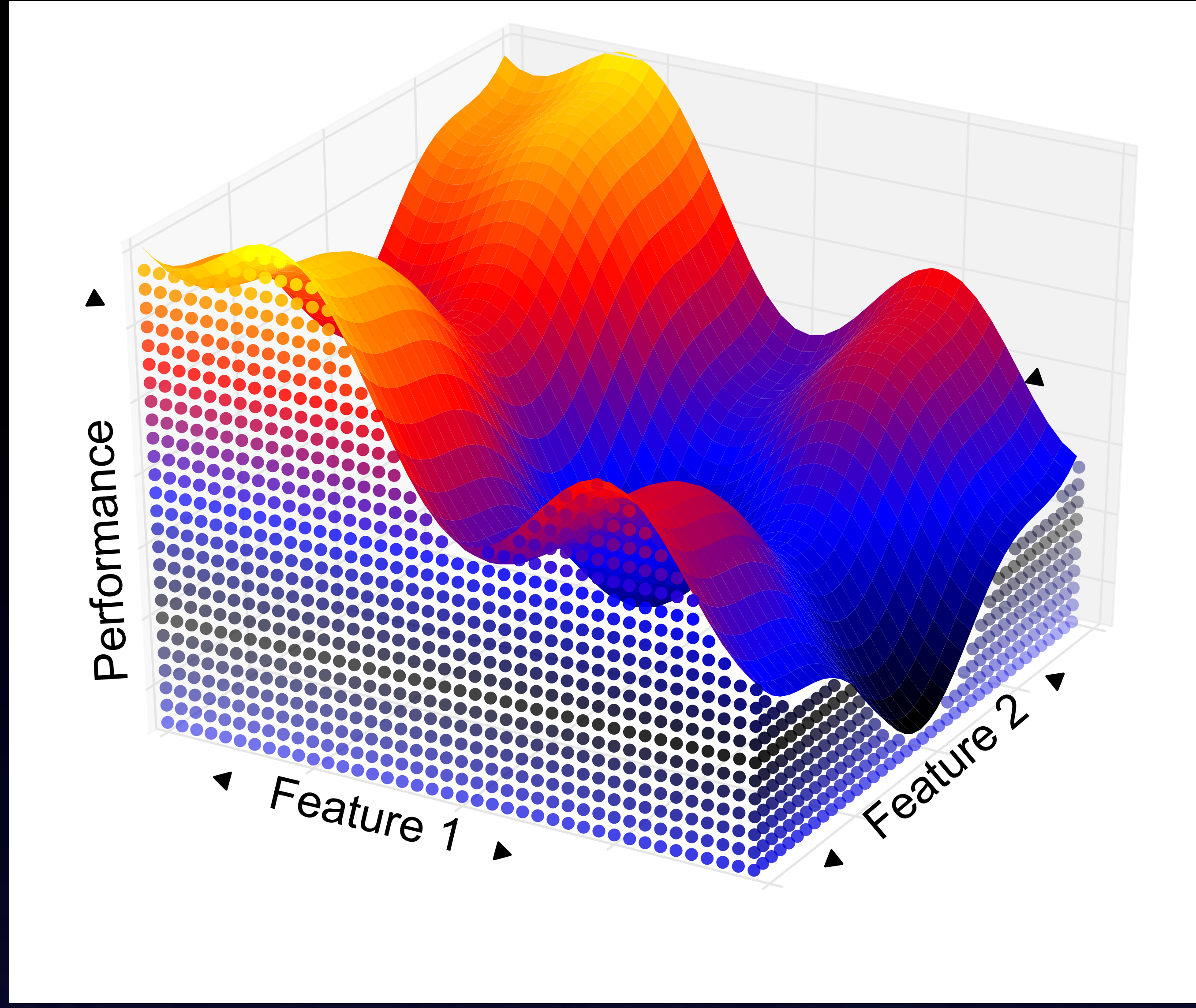
Set of diverse, high-quality solutions



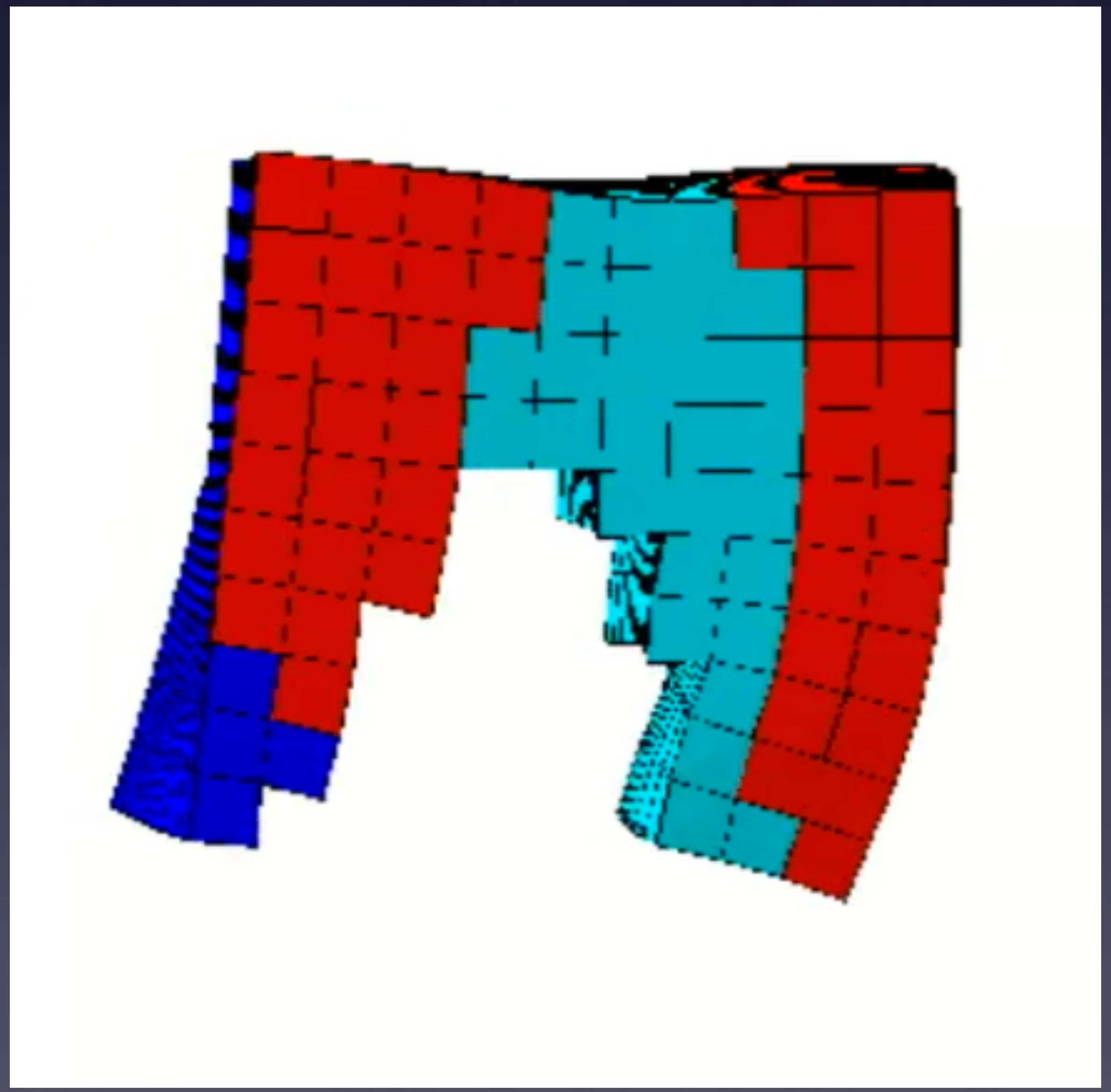


Soft Robots Problem

Mouret & Clune 2015, arXiv



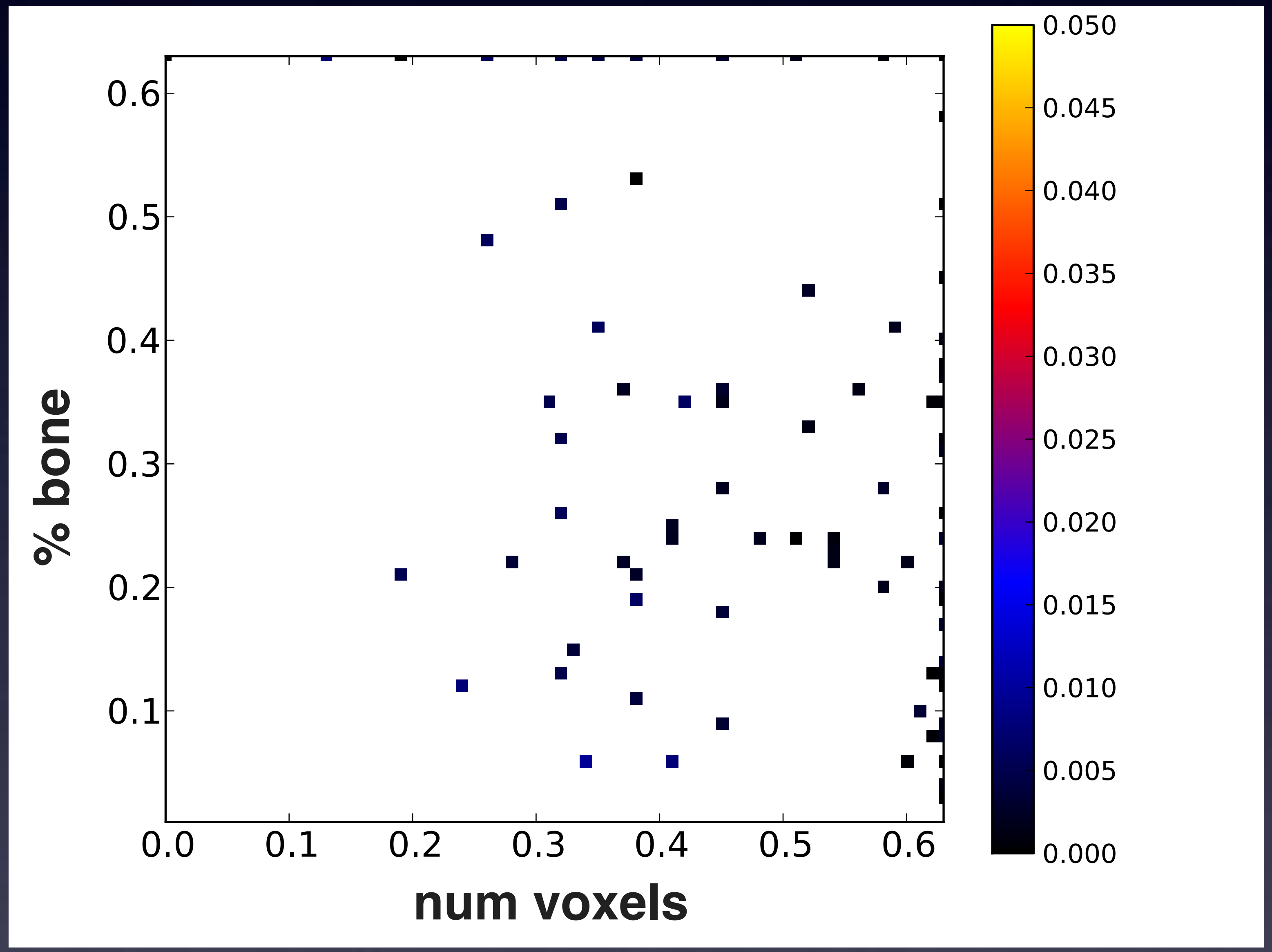
- Dimensions
  - number of voxels
  - % bone (dark blue)



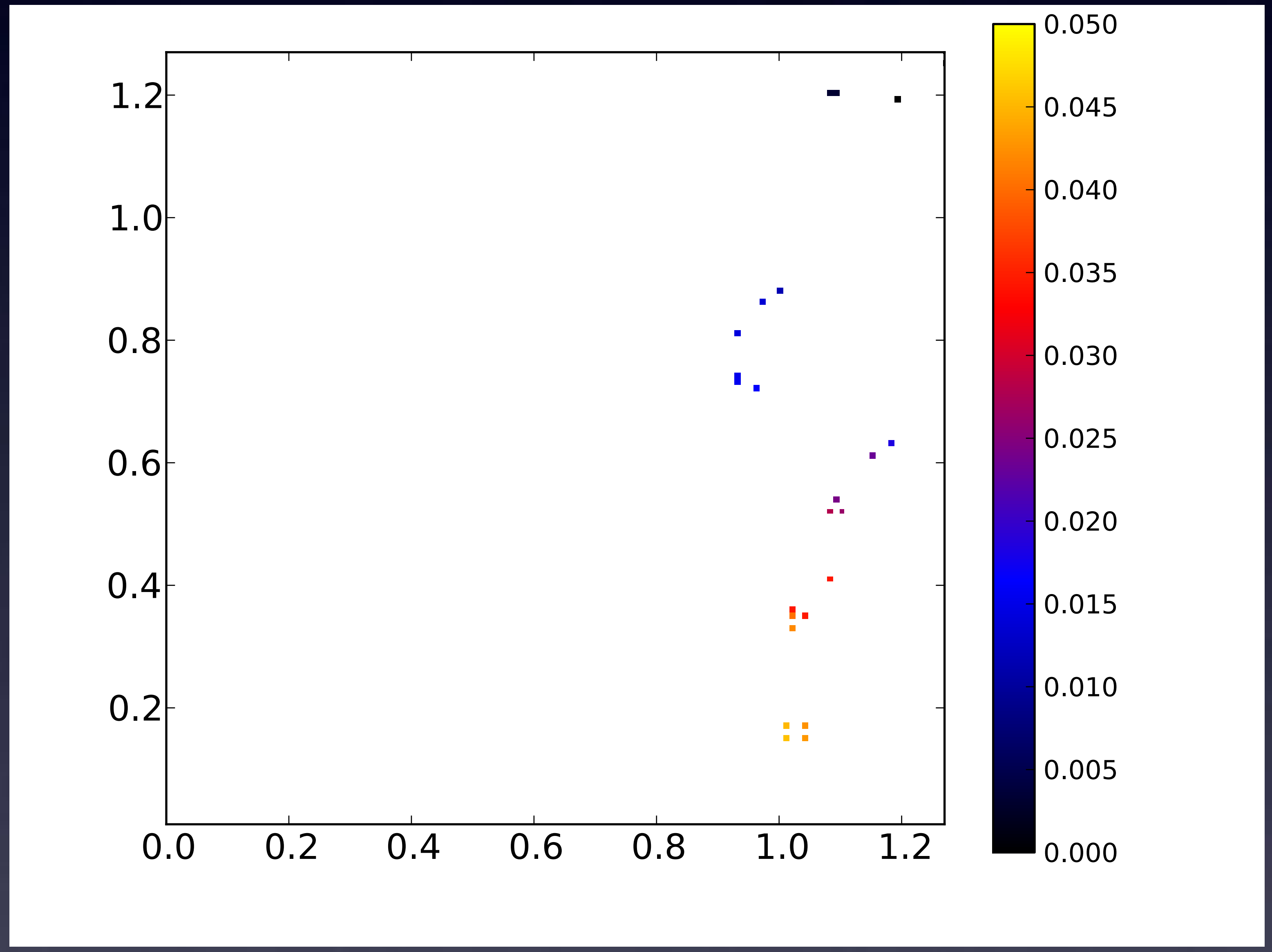
Soft Robots Problem

Mouret & Clune 2015, arXiv

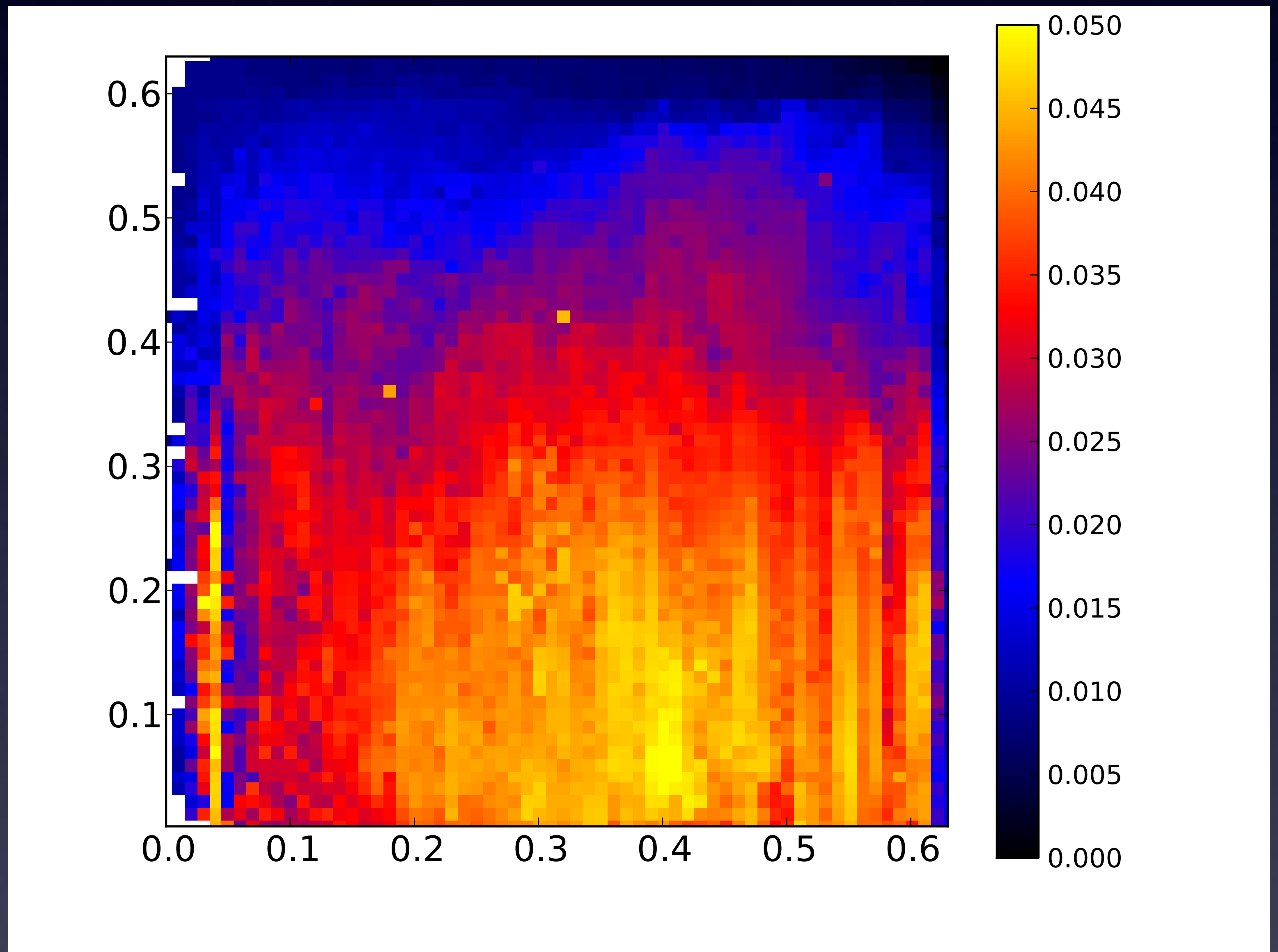
Classic Optimization



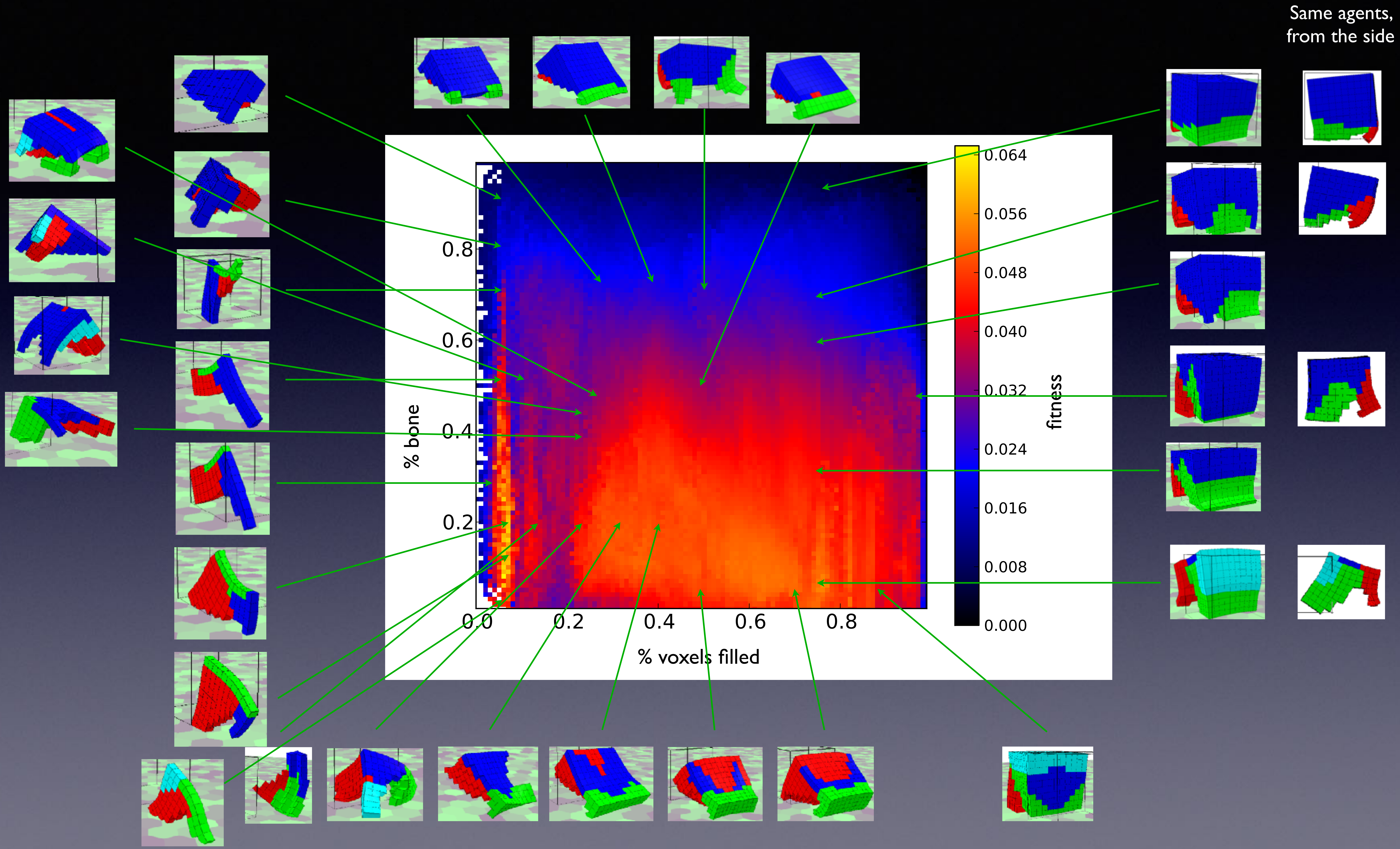
Classic + Diversity



MAP-Elites



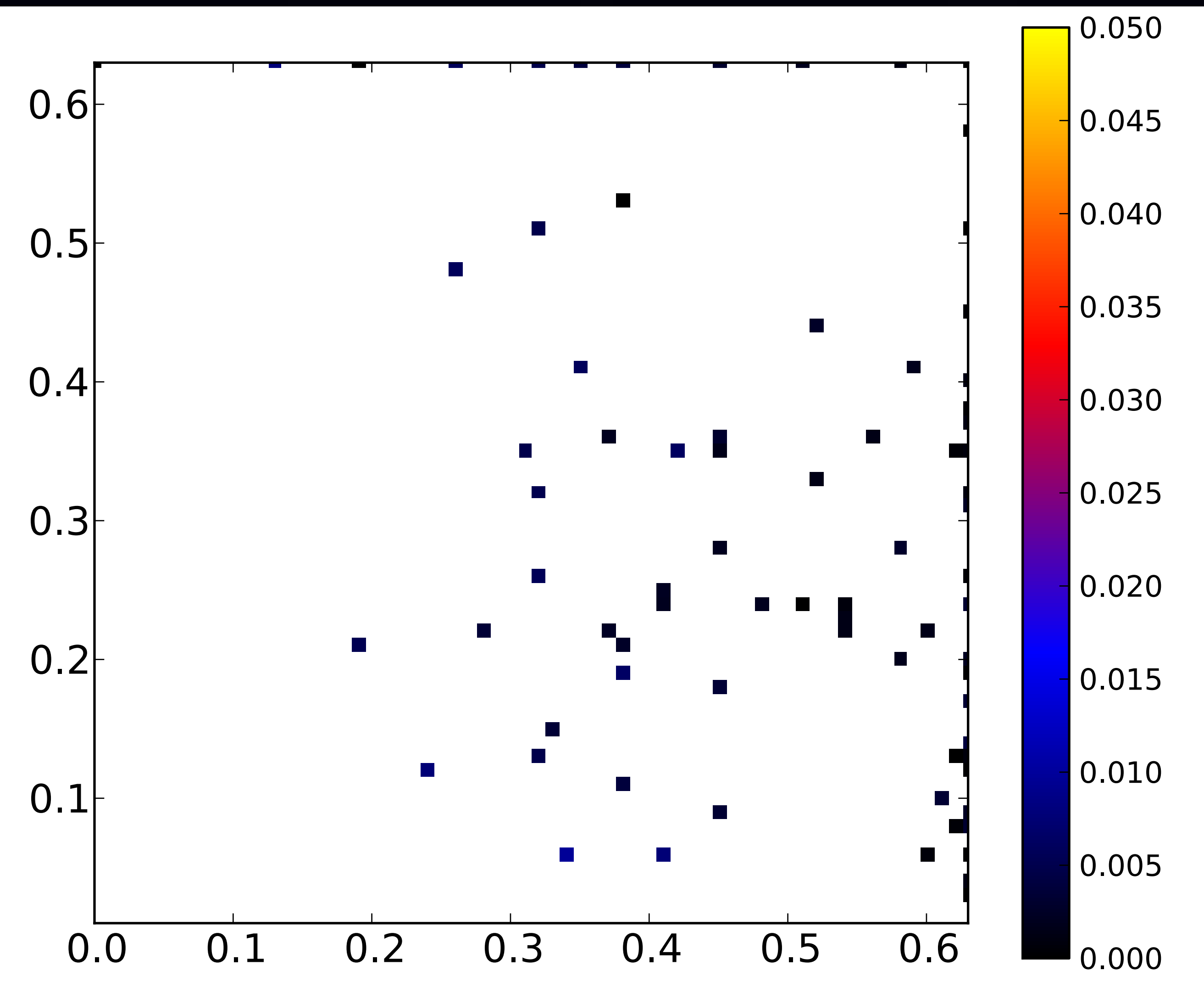
same # evals!

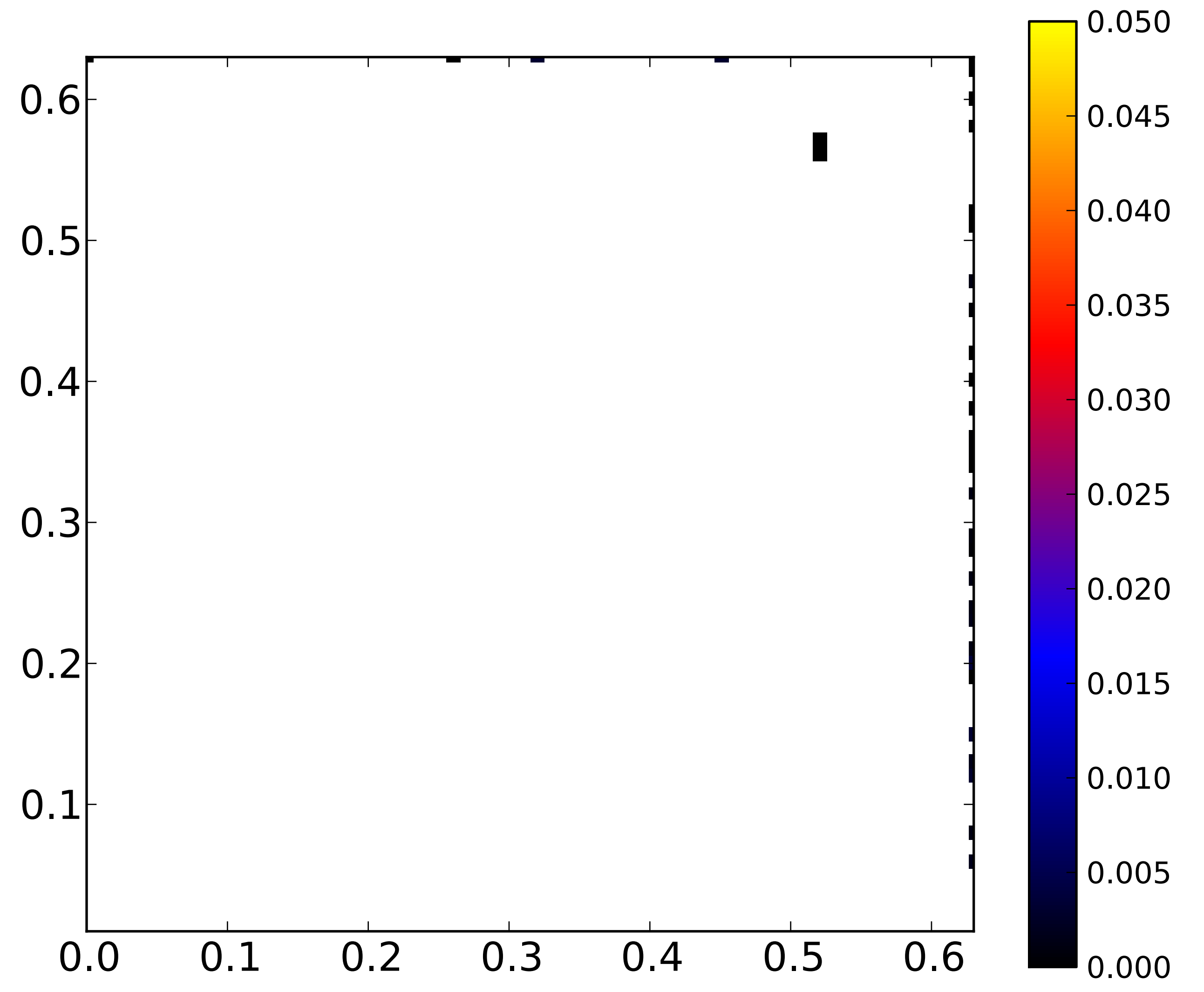


Same agents, from the side
% bone
% voxels filled
fitness
0.0
0.2
0.4
0.6
0.8
0.000
0.008
0.016
0.024
0.032
0.040
0.048
0.056
0.064

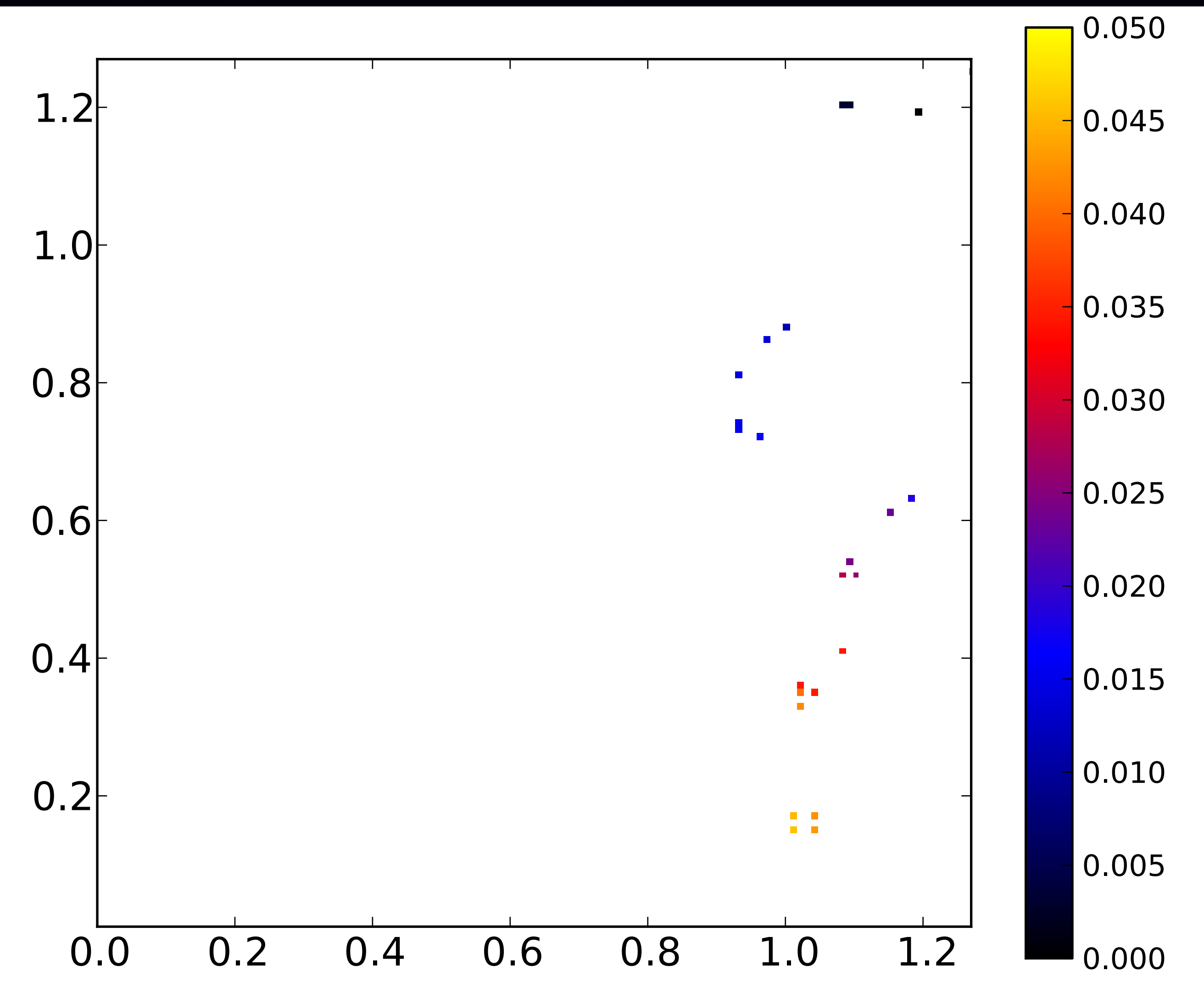
Different Runs: Soft Robot Problem

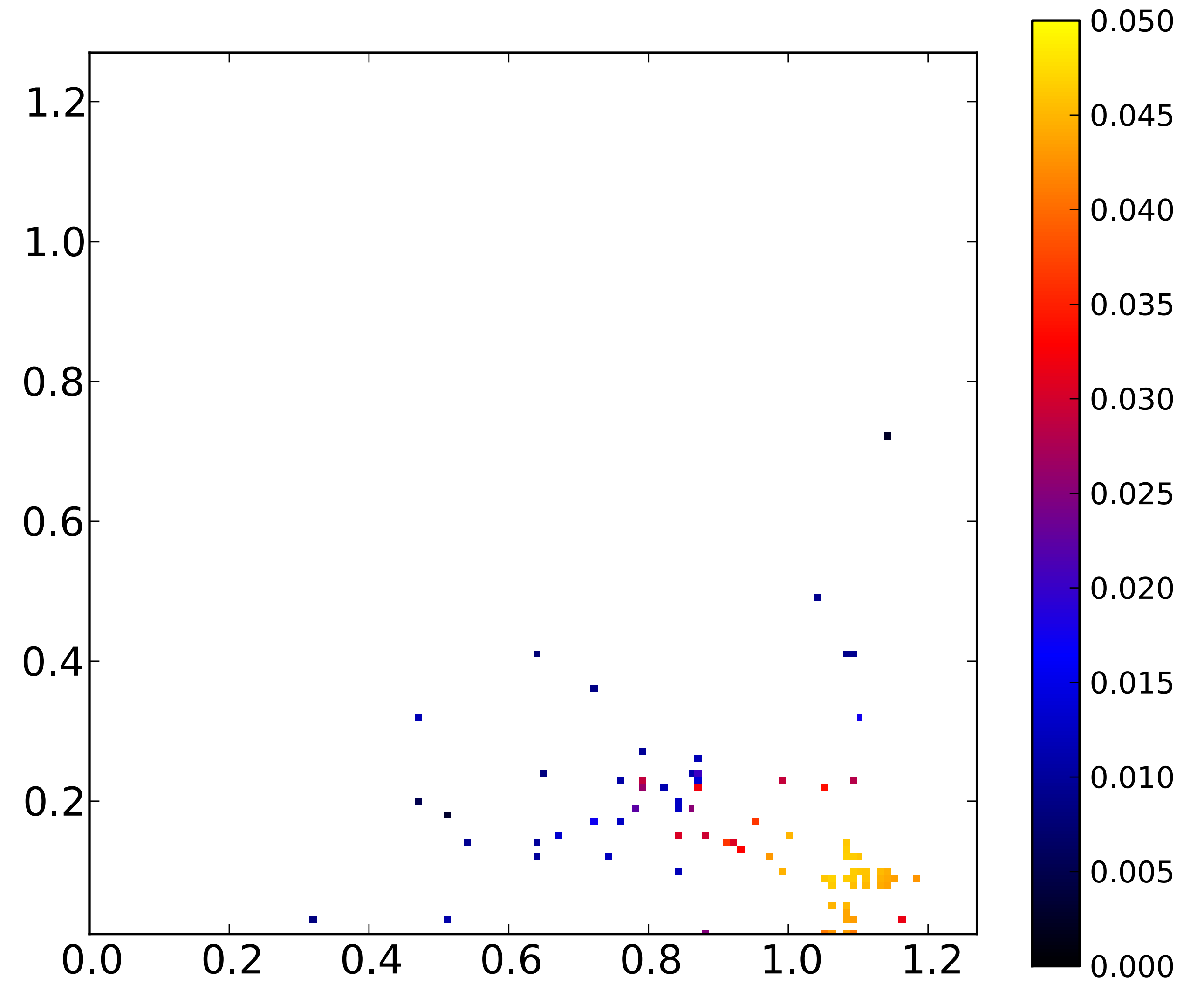
Classic Optimization



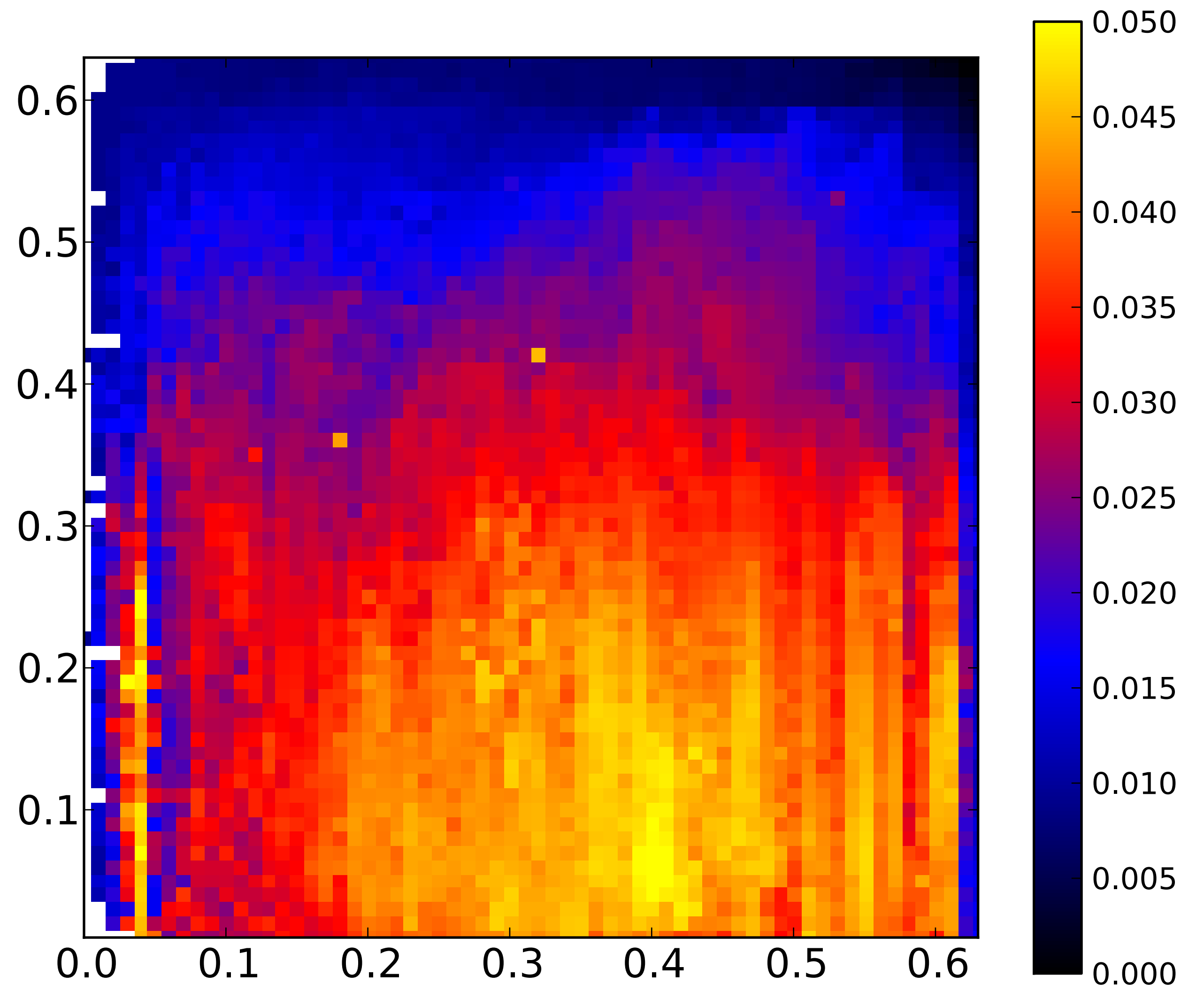


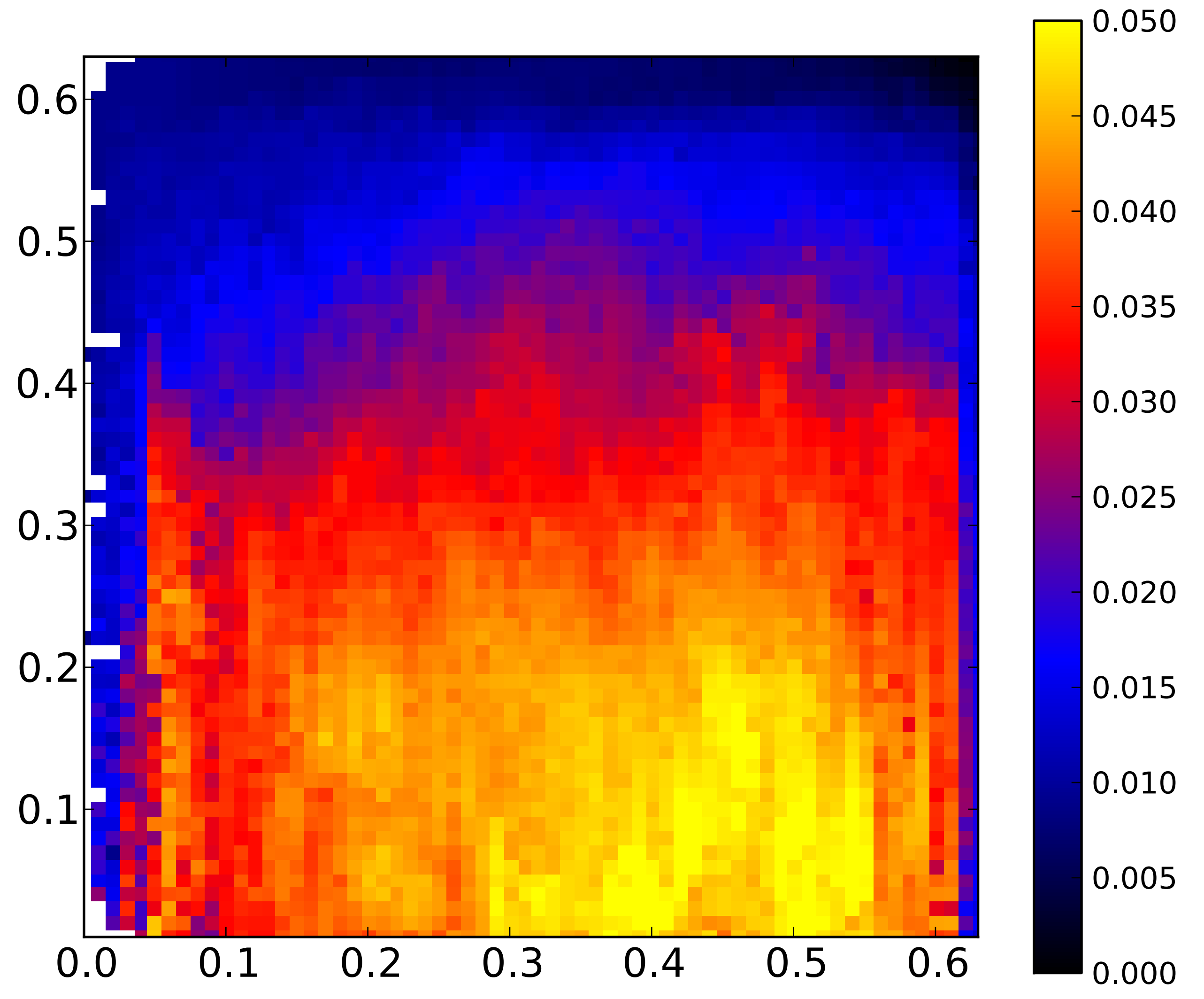
Classic + Diversity



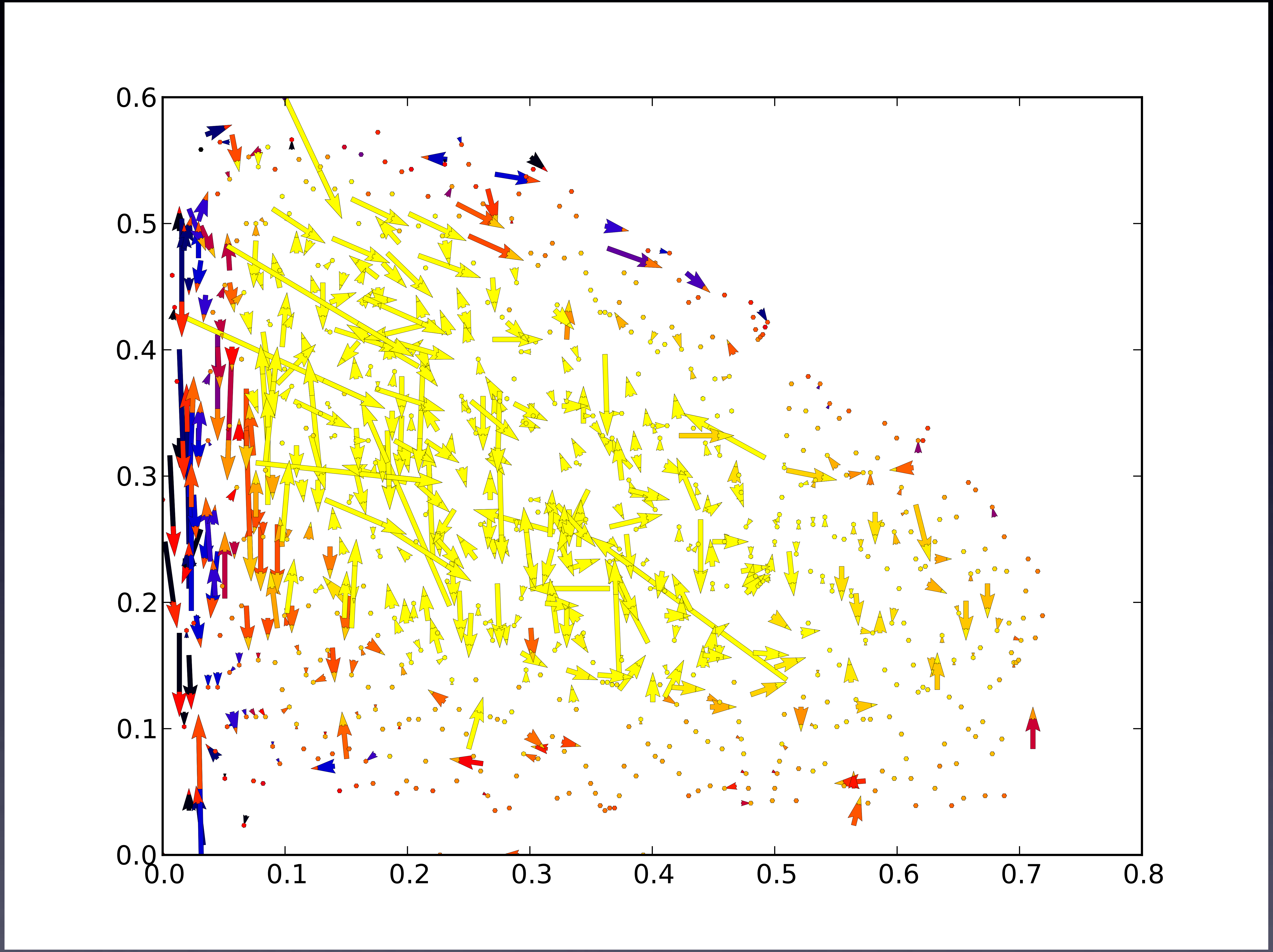


MAP-Elites



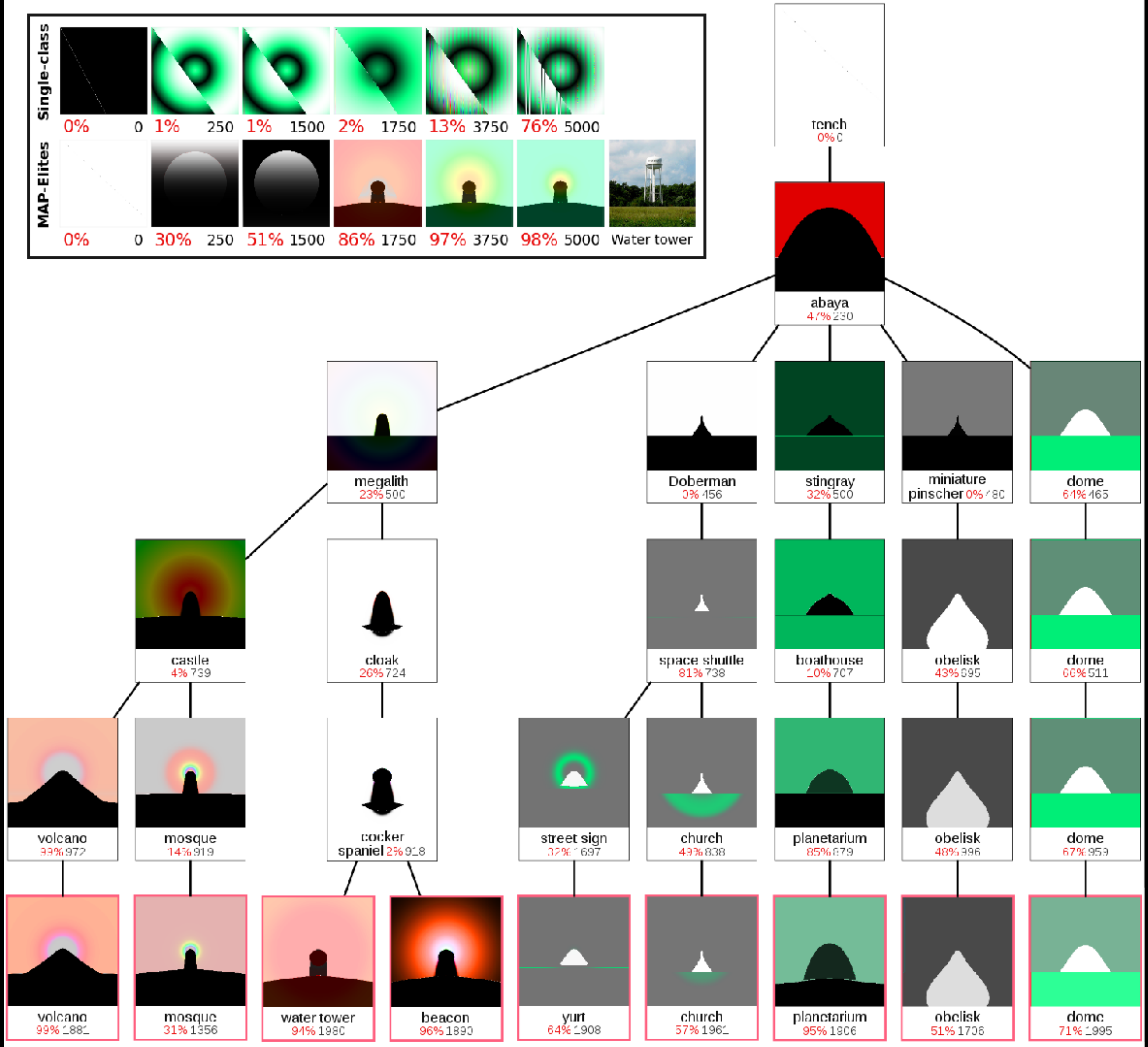


Goal Switching is Critical



retina problem

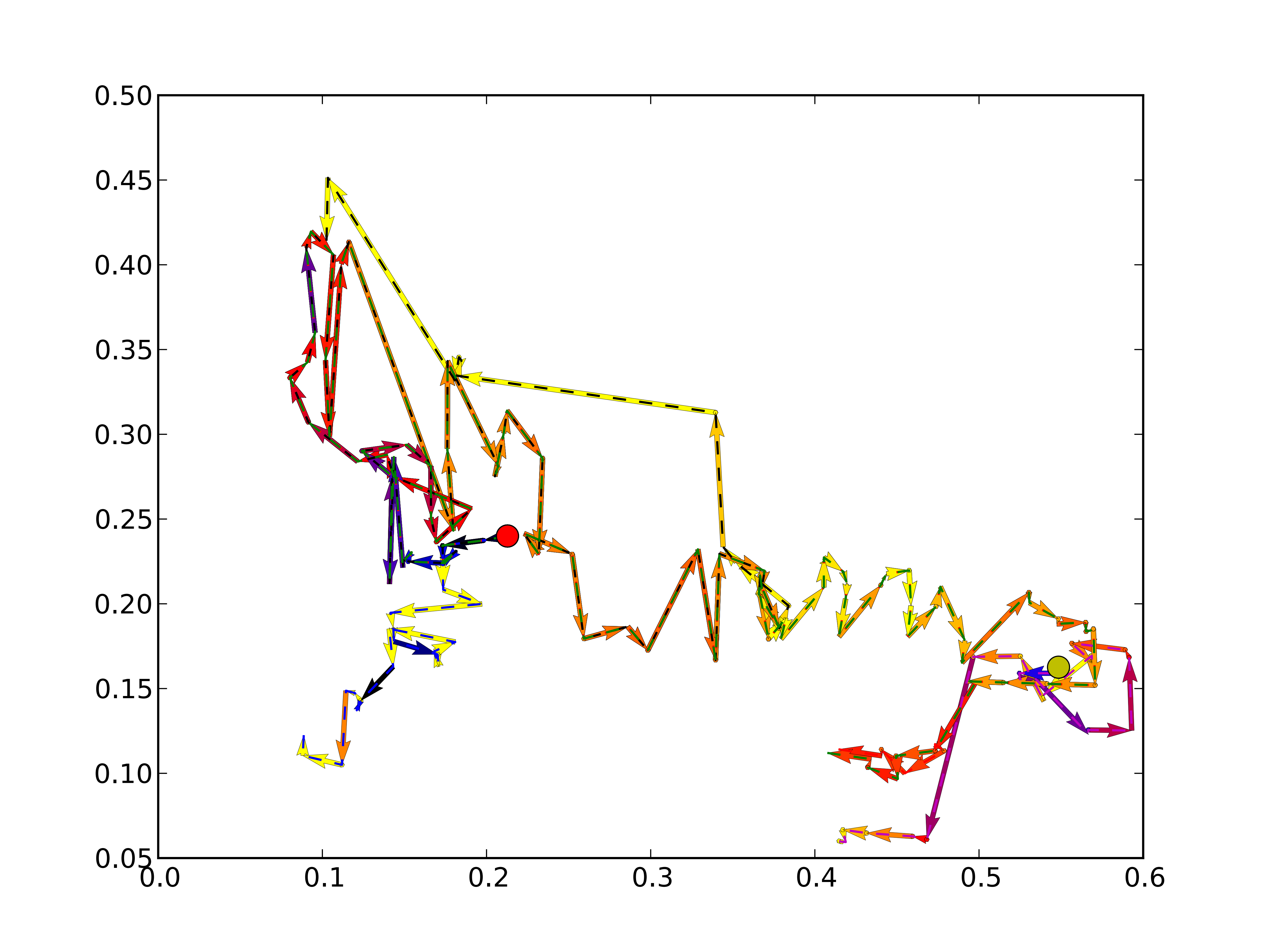
color = reward

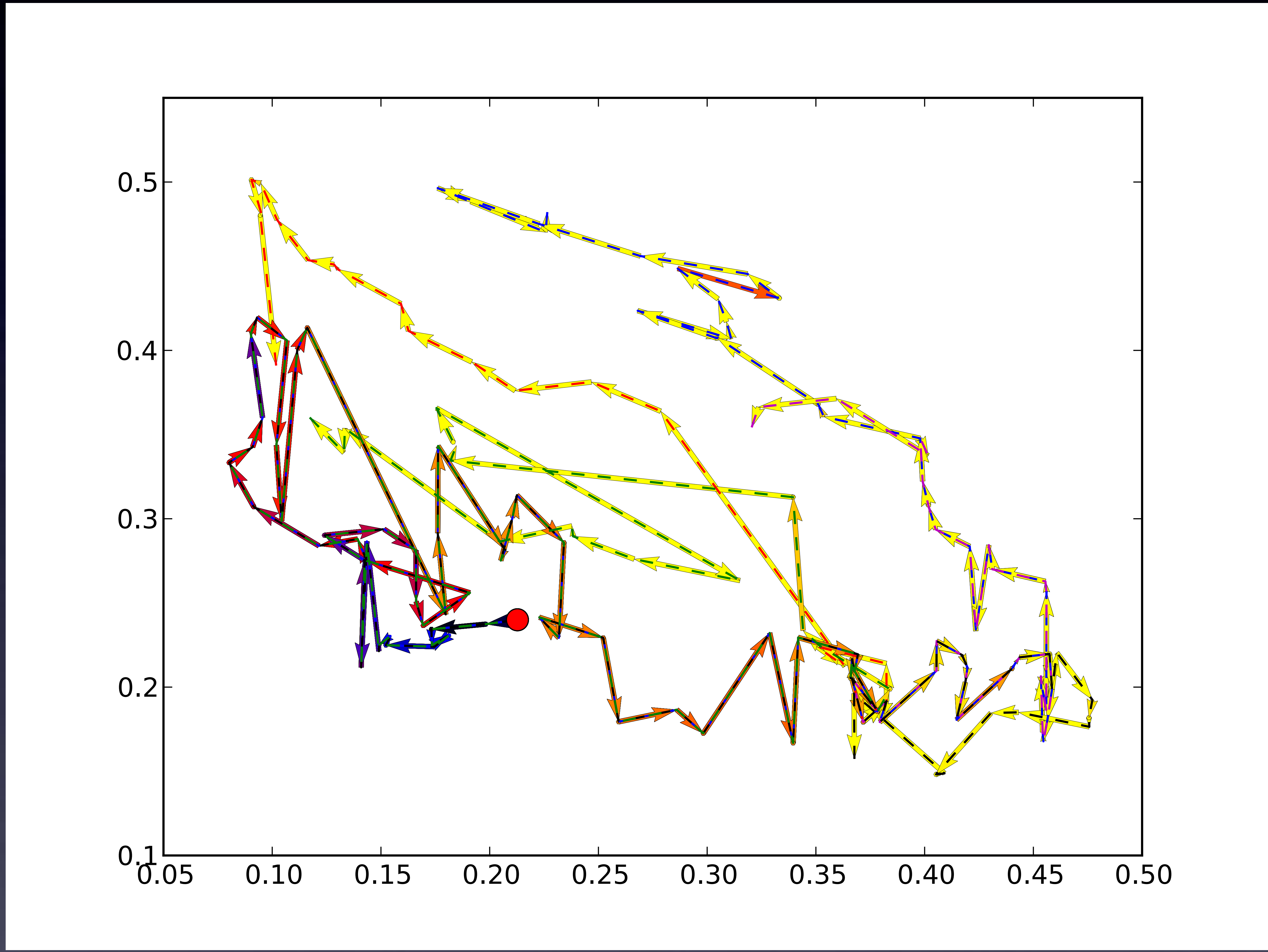


Innovation Engines

Nguyen, Yosinski, Clune (2016)

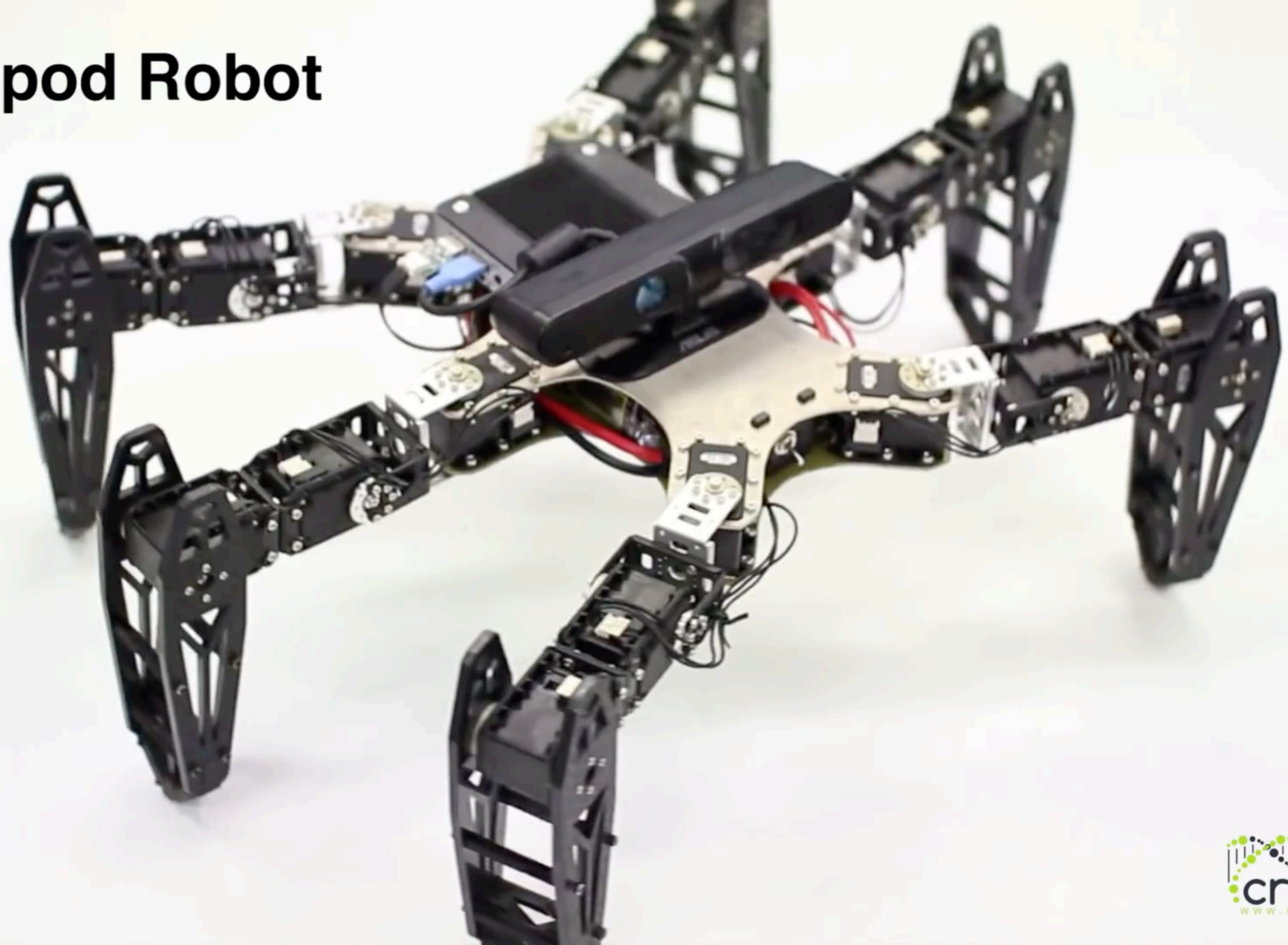
MAP-Elites Lineages of a Few Final Solutions





Circles are iteration 0, color = reward

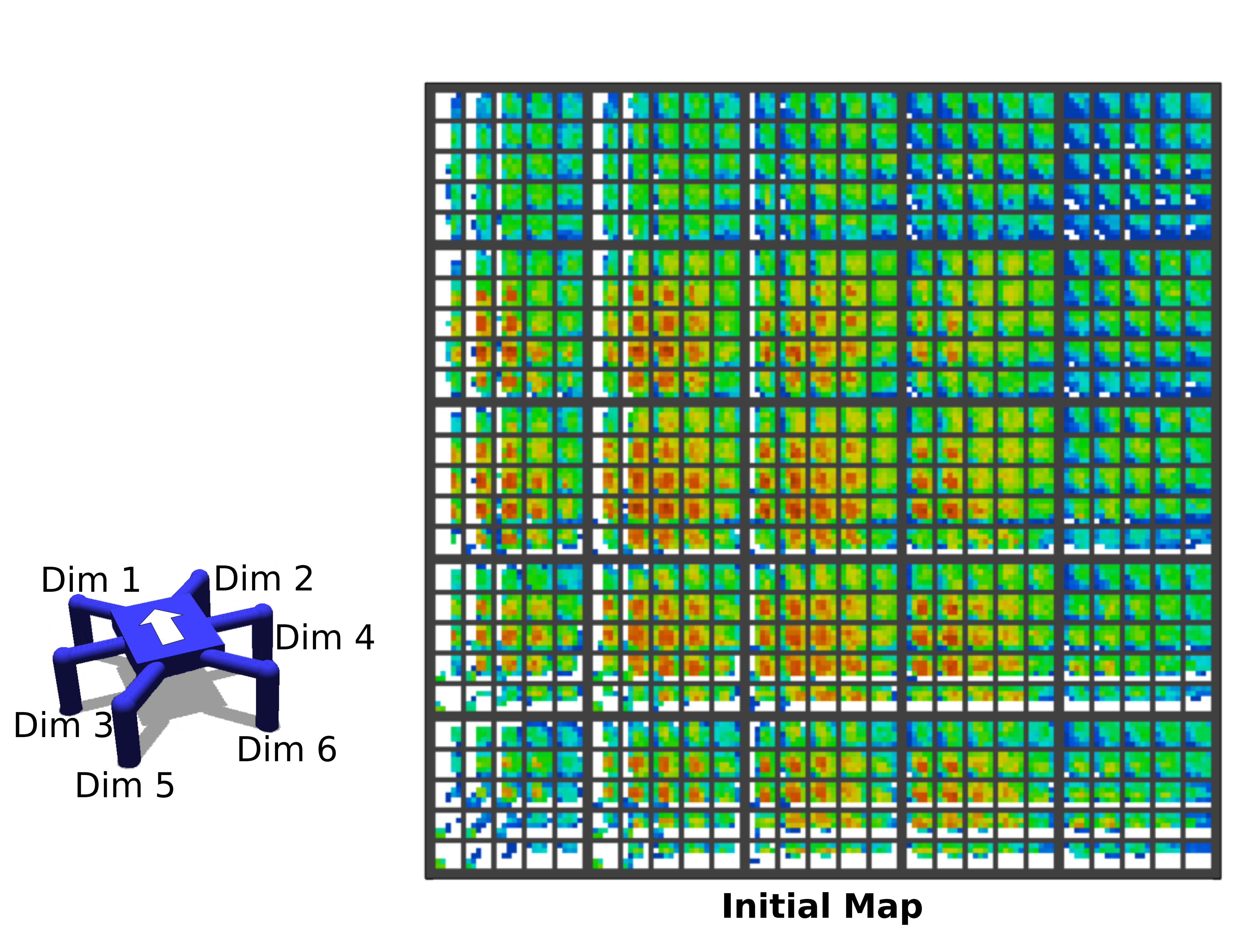
Hexapod Robot



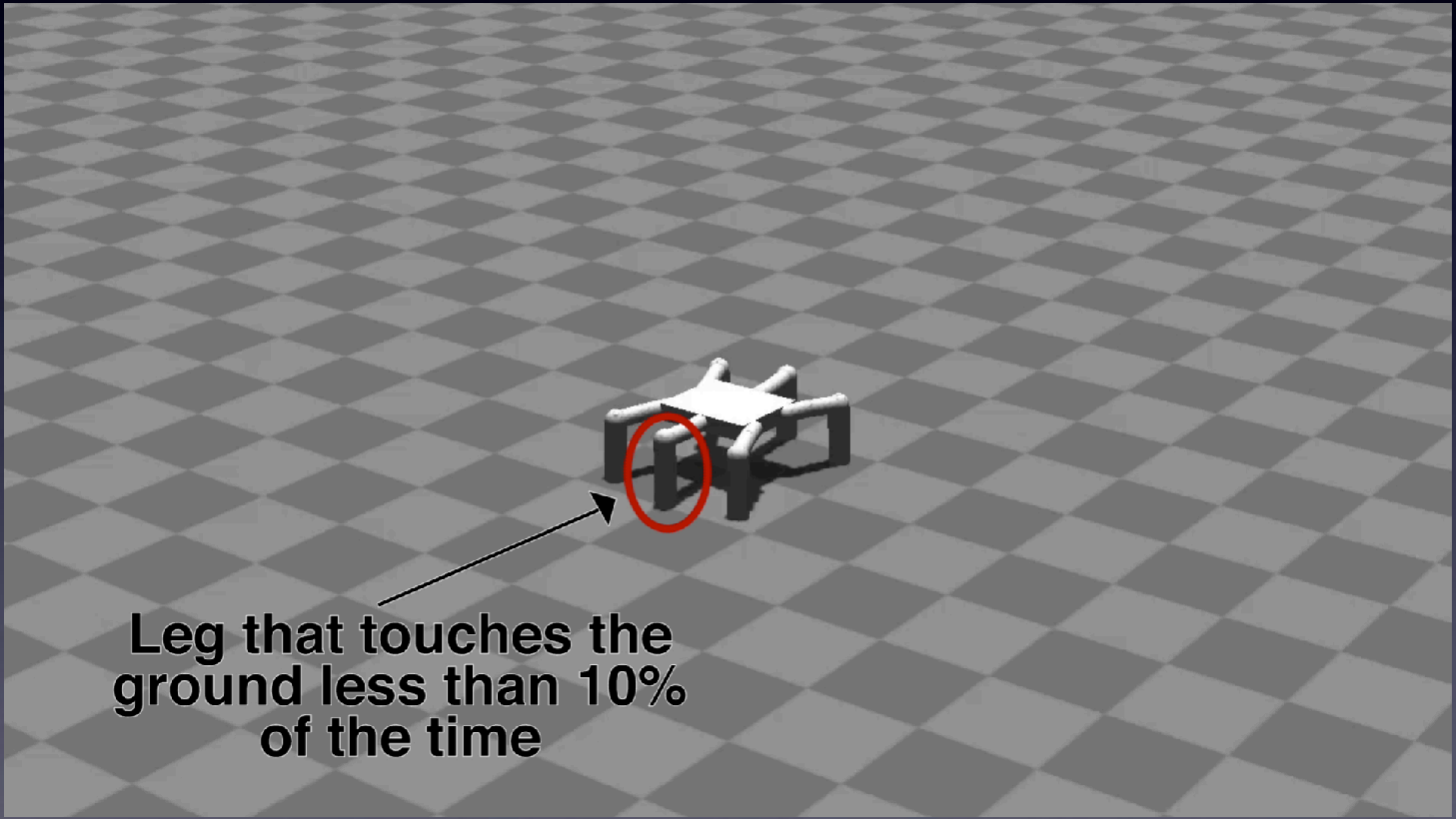
creadapt
www.creadapt.net

intuitions about different ways to move

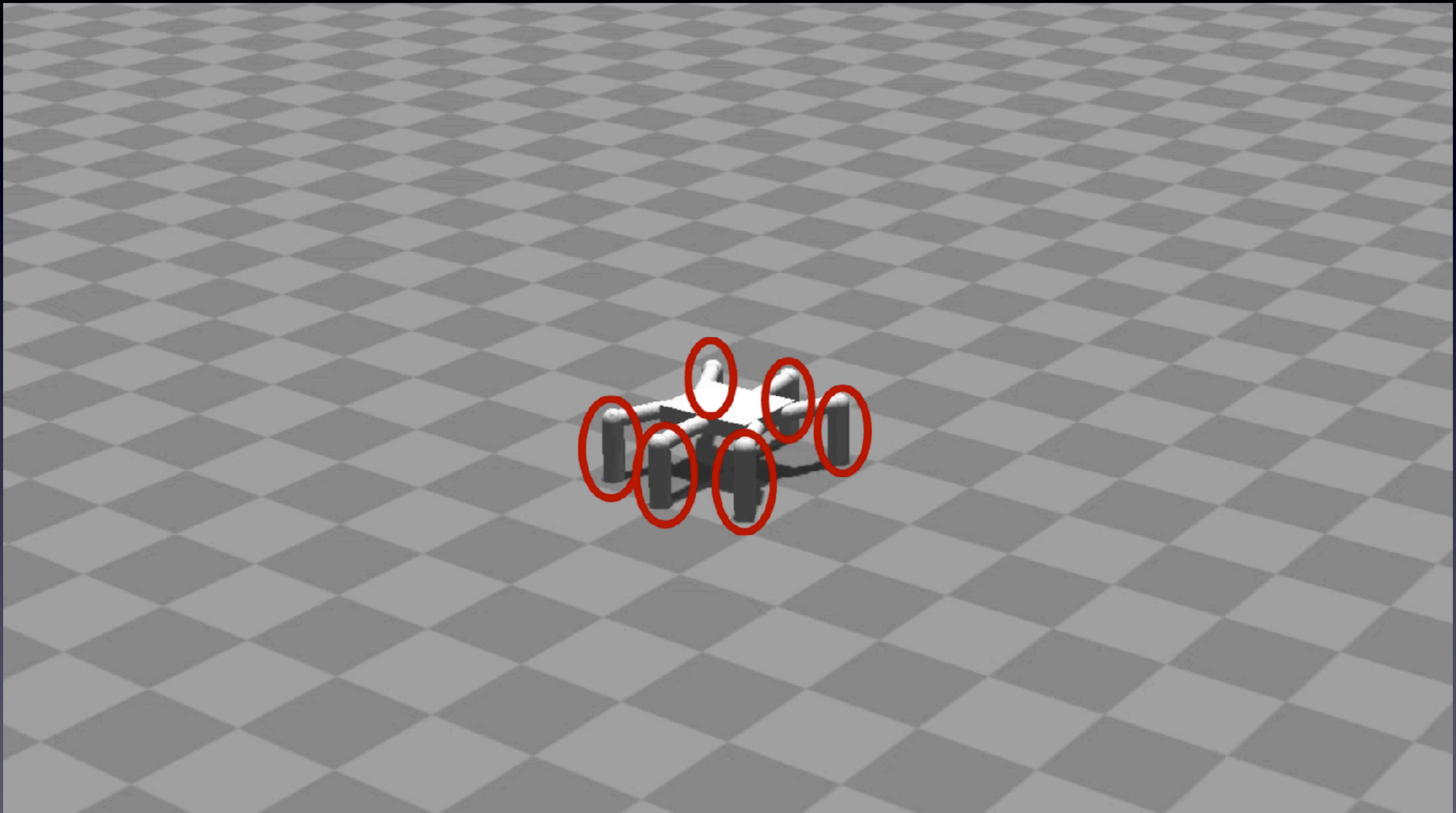
- Behavioral characterization
  - % of time each leg touches the ground (6-dimensional)
- Producing the map is expensive
  - 40 million evaluations per map (!)
  - But can be done once per robot pre-deployment
- Map has ~13,000 diverse, high-performing gaits



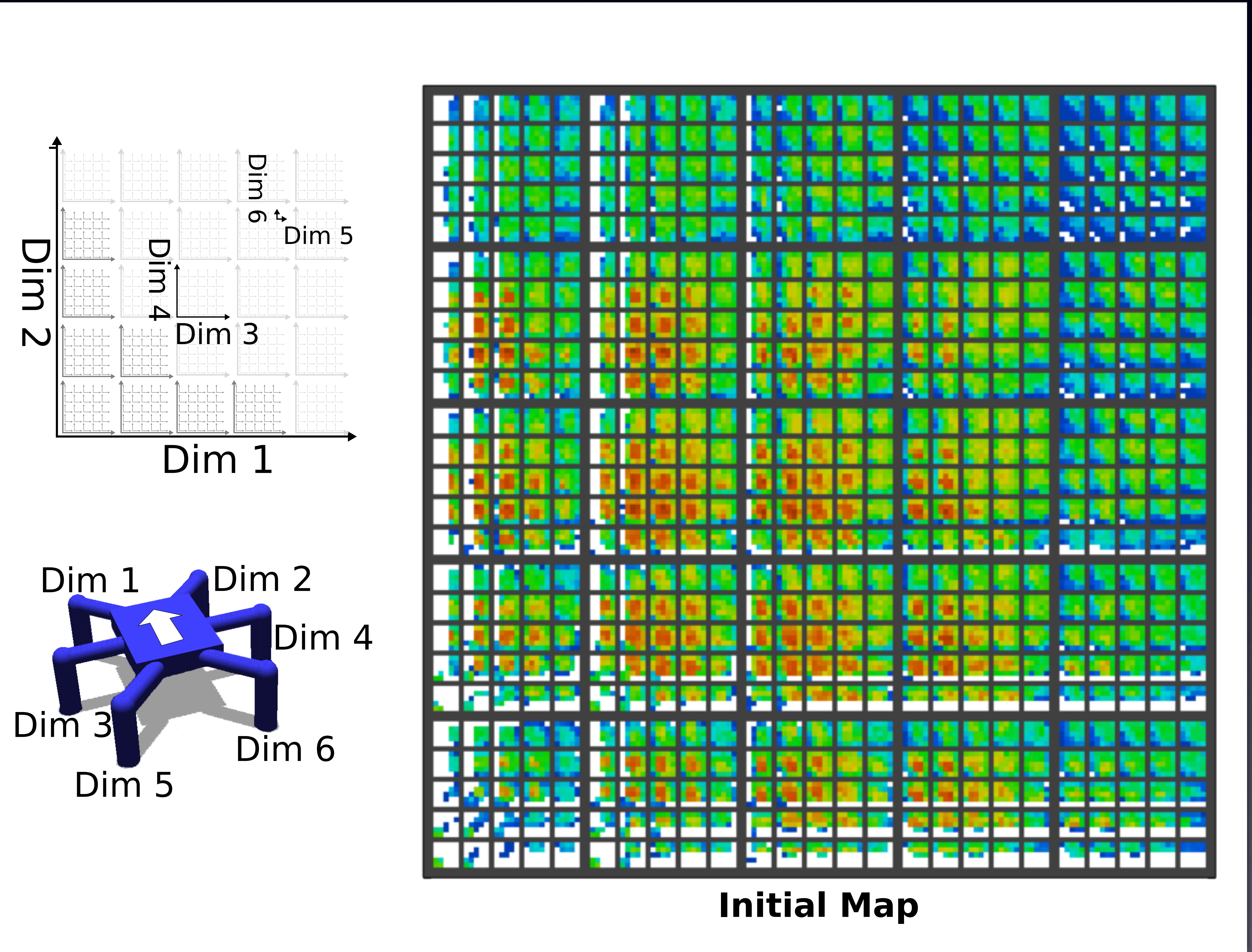
intuitions about
different ways to move



Corner Case: Feet never touch the ground

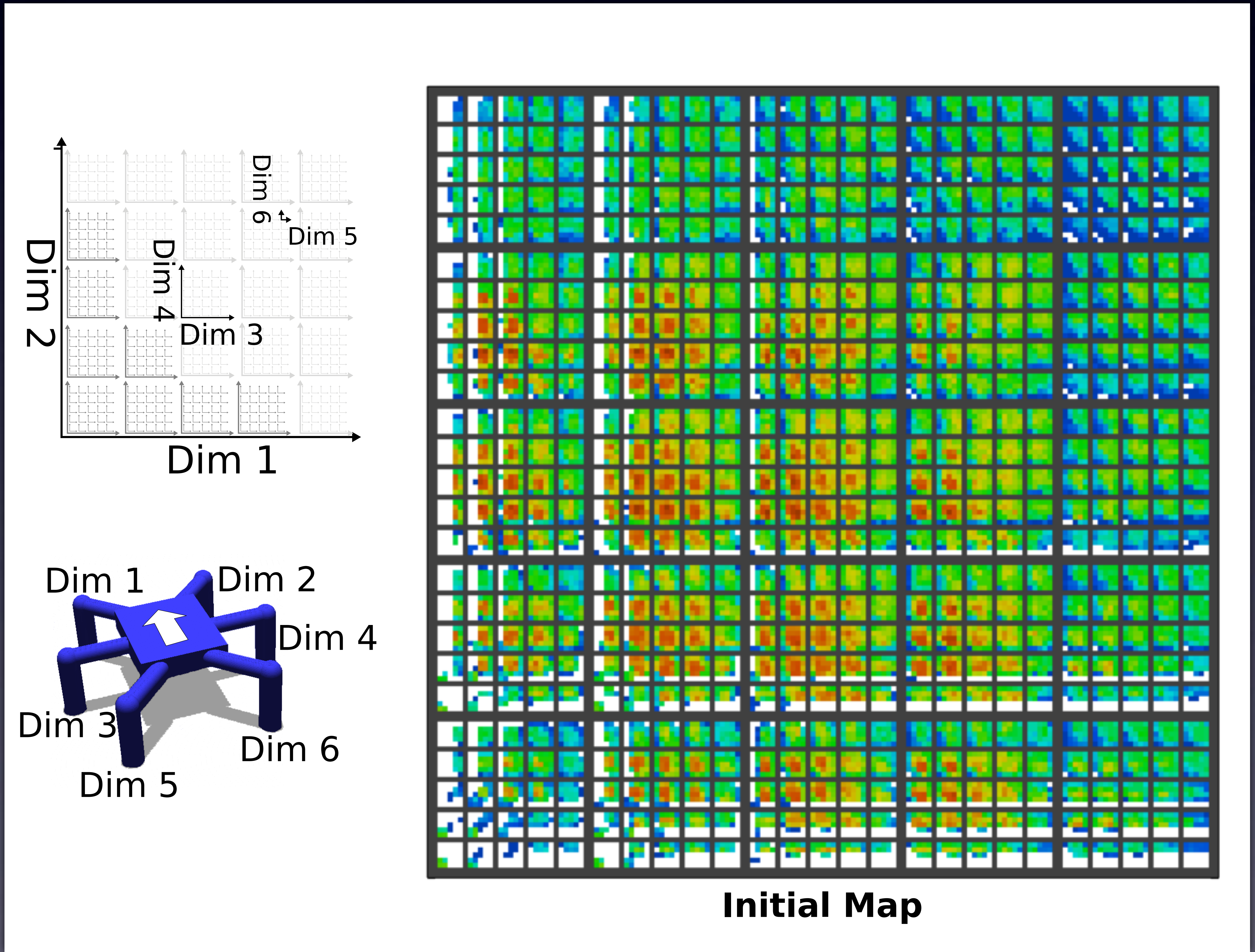


intuitions about different ways to move



On the simulated, undamaged robot

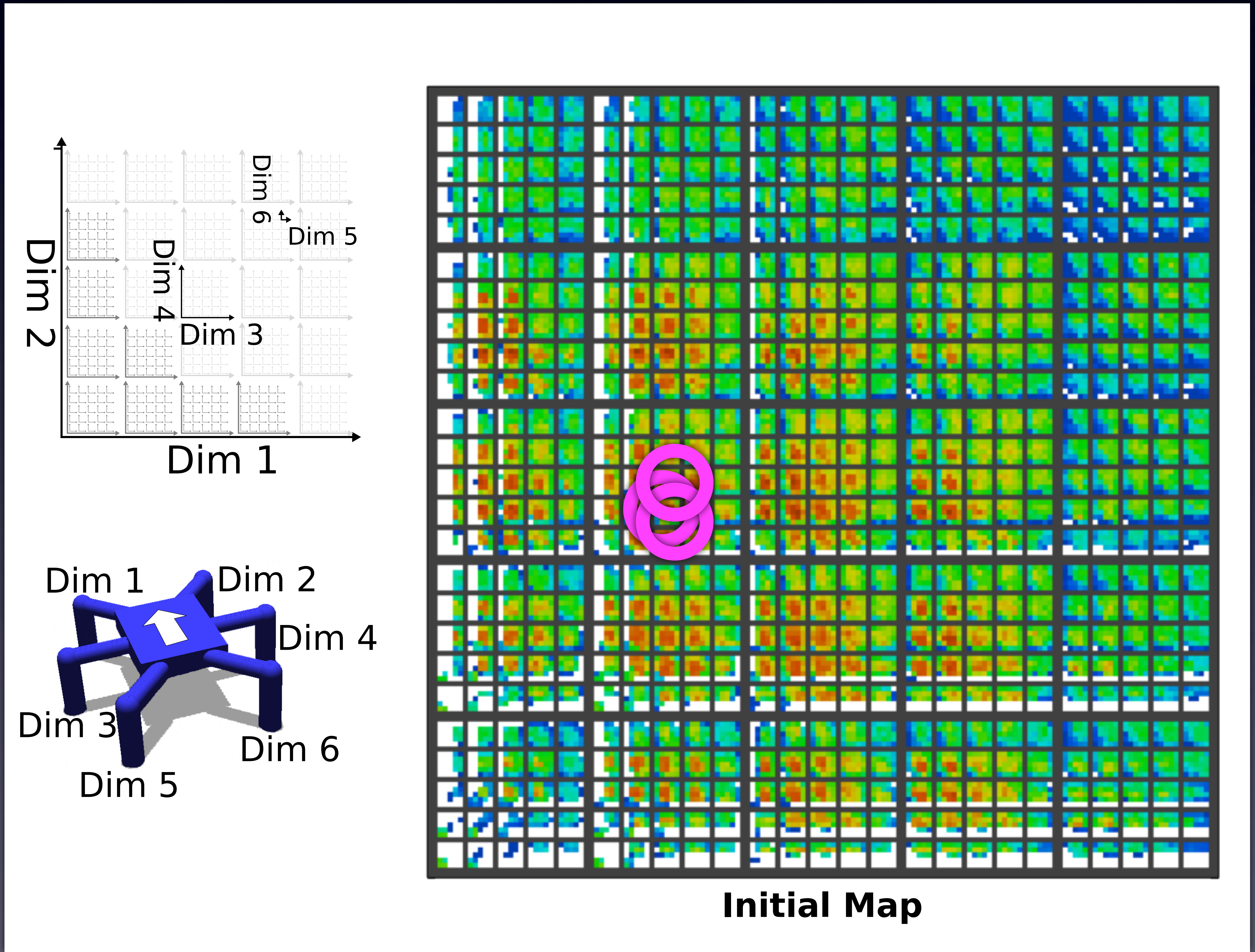
intuitions about different ways to move → few, intelligent tests



Which behaviors should we test?

intuitions about different ways to move

few, intelligent tests

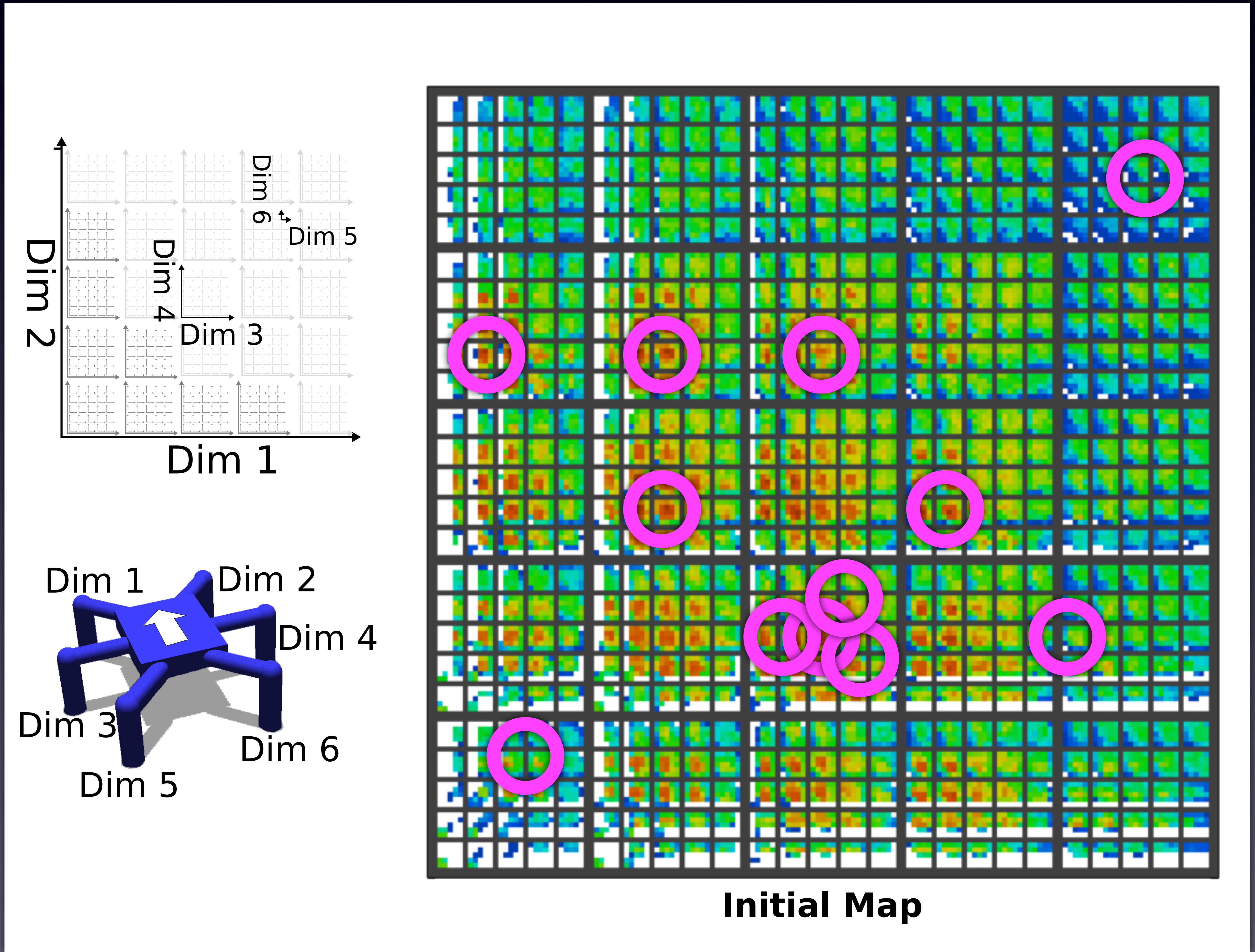


Could try top N:

But they are likely very similar.

intuitions about different ways to move

few, intelligent tests



Bayesian Optimization:

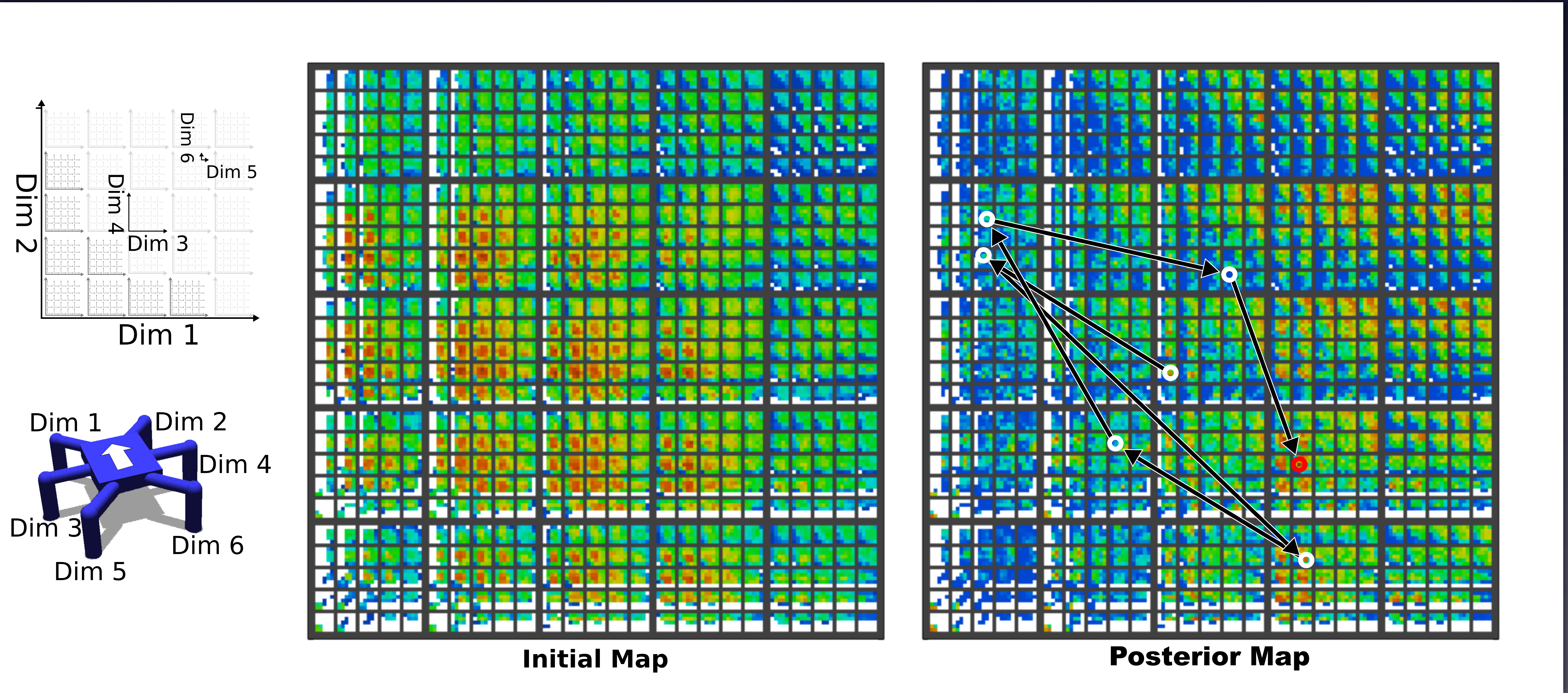
Tries different types solutions

Bayesian Optimization

Prior:
MAP-Elites Map

Posterior:
Map updated after
real-world tests

Stop when:
A real-world
behavior is >90% of
best untested point



One-dimensional Example

“Intelligent Trial & Error”

intuitions about different ways to move → few, intelligent tests → pick one that works despite injury

MAP-Elites Map

Bayesian Optimization w Map as Prior

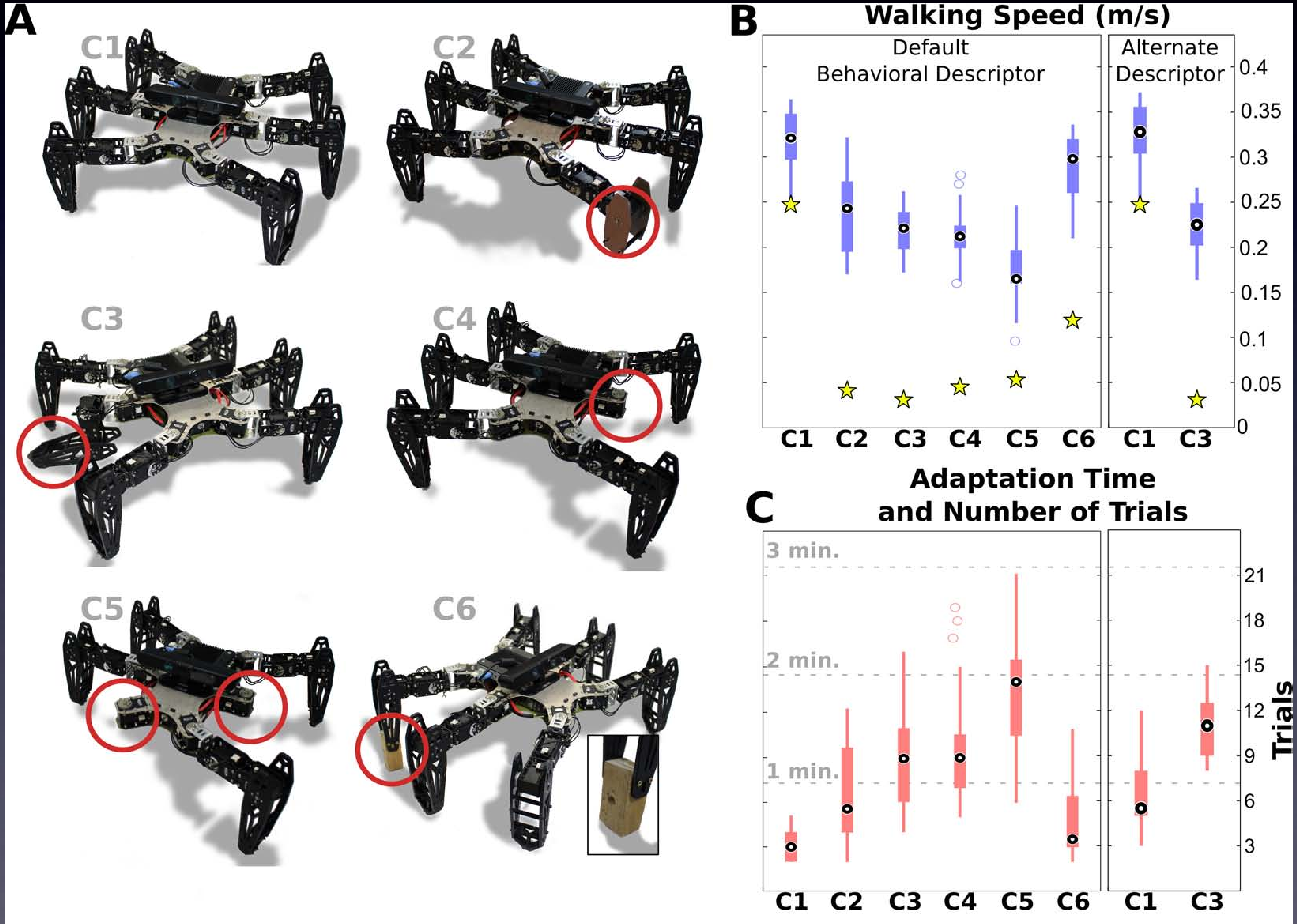
Found >90% of Best Possible

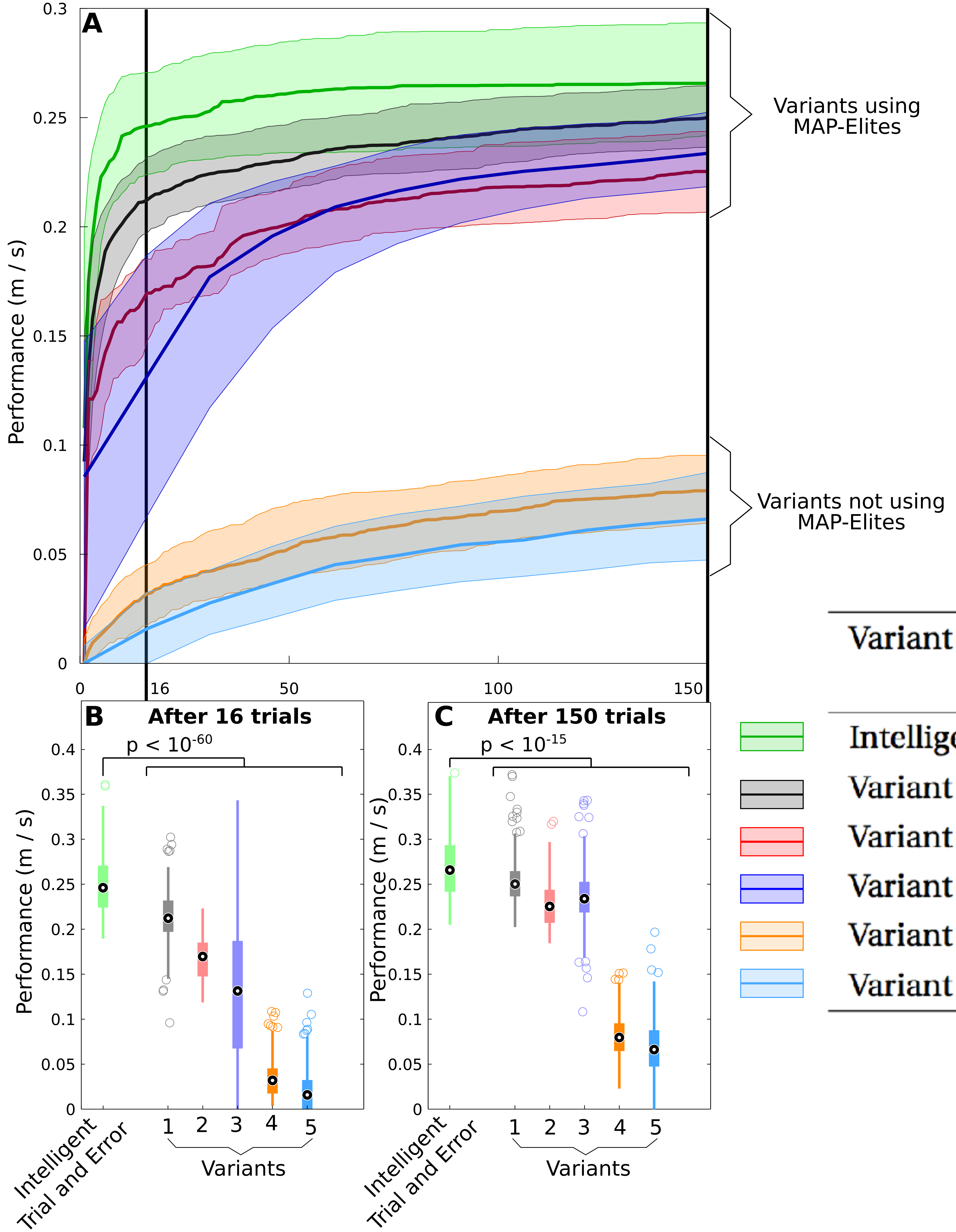
Undamaged robot controlled with classic tripod gait

00:00:00
Behavior-performance Map
Forward Speed (m/s)
0.13
Trajectory


X 0.25
Behavior-performance Map
Forward Speed (m/s)
0.24
Trajectory

Different Damage Conditions & Behavioral Descriptions





Variant	Behavioral repertoire creation	Priors on performance	Search algorithm
Intelligent Trial and Error	MAP-Elites	yes	Bayesian Optimization
Variant 1	MAP-Elites	none	random search
Variant 2	MAP-Elites	none	Bayesian optimization
Variant 3	MAP-Elites	none	policy gradient
Variant 4	none	none	Bayesian optimization
Variant 5	none	none	policy gradient

Undamaged robotic arm


00:00:01
Behavior-performance Map
Position Error (m)
0.24
Camera View

Different Environments

Deep Learning + IT&E

- Can swap in deep neural networks
  - deep reinforcement learning

Map-based Multi-Policy Reinforcement Learning: Enhancing Adaptability of Robots by Deep Reinforcement Learning

Ayaka Kume, Eiichi Matsumoto, Kuniyuki Takahashi, Wilson Ko and Jethro Tan

Abstract—**In order for robots to perform mission-critical tasks, it is essential that they are able to quickly adapt to changes in their environment as well as to injuries and or other bodily changes. Deep reinforcement learning has been shown to be successful in training robot control policies for operation in complex environments. However, existing methods typically employ only a single policy. This can limit the adaptability since a large environmental modification might require a completely different behavior compared to the learning environment. To solve this problem, we propose Map-based Multi-Policy Reinforcement Learning (MMPRL), which aims to search and store multiple policies that encode different behavioral features while maximizing the expected reward in advance of the environment change. Thanks to these policies, which are stored into a multi-dimensional discrete map according to its behavioral feature, adaptation can be performed within reasonable time without retraining the robot. An appropriate pre-trained policy from the map can be recalled using Bayesian optimization. Our experiments show that MMPRL enables robots to quickly adapt to large changes without requiring any prior knowledge on the type of injuries that could occur.**

A highlight of the learned behaviors can be found here: `https://youtu.be/QwInbilXNOE`.

I. Introduction

Humans and animals are well-versed in quickly adapting to changes in not only their surrounding environments, but also to changes to their own body, through previous experiences and information from their senses. Some example scenarios where such adaptation to environment changes takes place are walking in a highly crowded scene with a lot of other people and objects, walking on uneven terrain, or walking against a strong wind. On the other hand, examples of bodily changes could be wounds, incapability to use certain body parts due to task constraints, or when lifting or holding something heavy. In a future where robots are omnipresent and used in mission critical tasks, robots are not only expected to adapt to unfamiliar scenarios and disturbances autonomously, but also to recover from adversaries in order to continue and complete their tasks successfully. Furthermore, taking a long time to recover or adapt may result in mission failure, while external help might not be available or even desirable, for example in search-and-rescue missions. Therefore, robots need to be able to adapt to changes in both the environment and their own body state, within a limited amount of time.

Recently, deep reinforcement learning (DRL) has been shown to be successful in complex environments with both

All authors are associated with Preferred Networks, Inc., Tokyo, Japan. (e-mail: {kume, matsumoto, takahashi, wko, jettan}@preferred.jp)

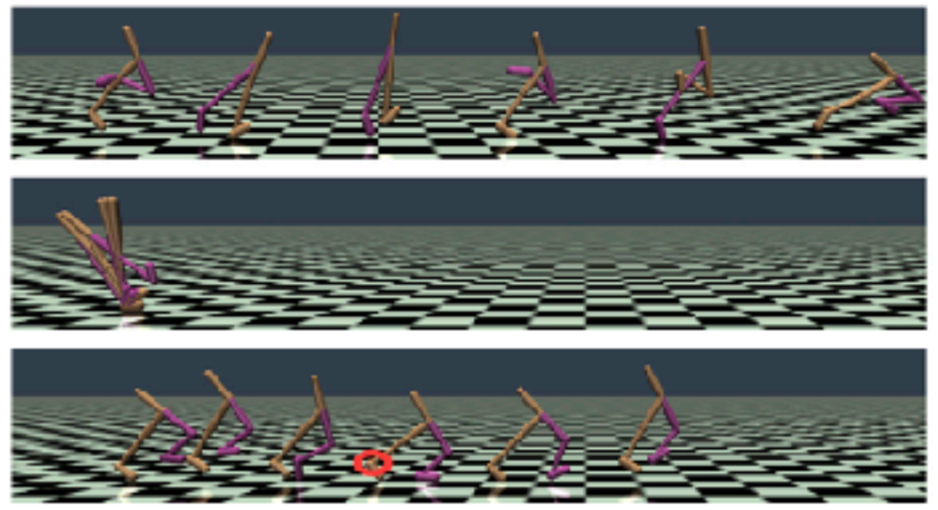


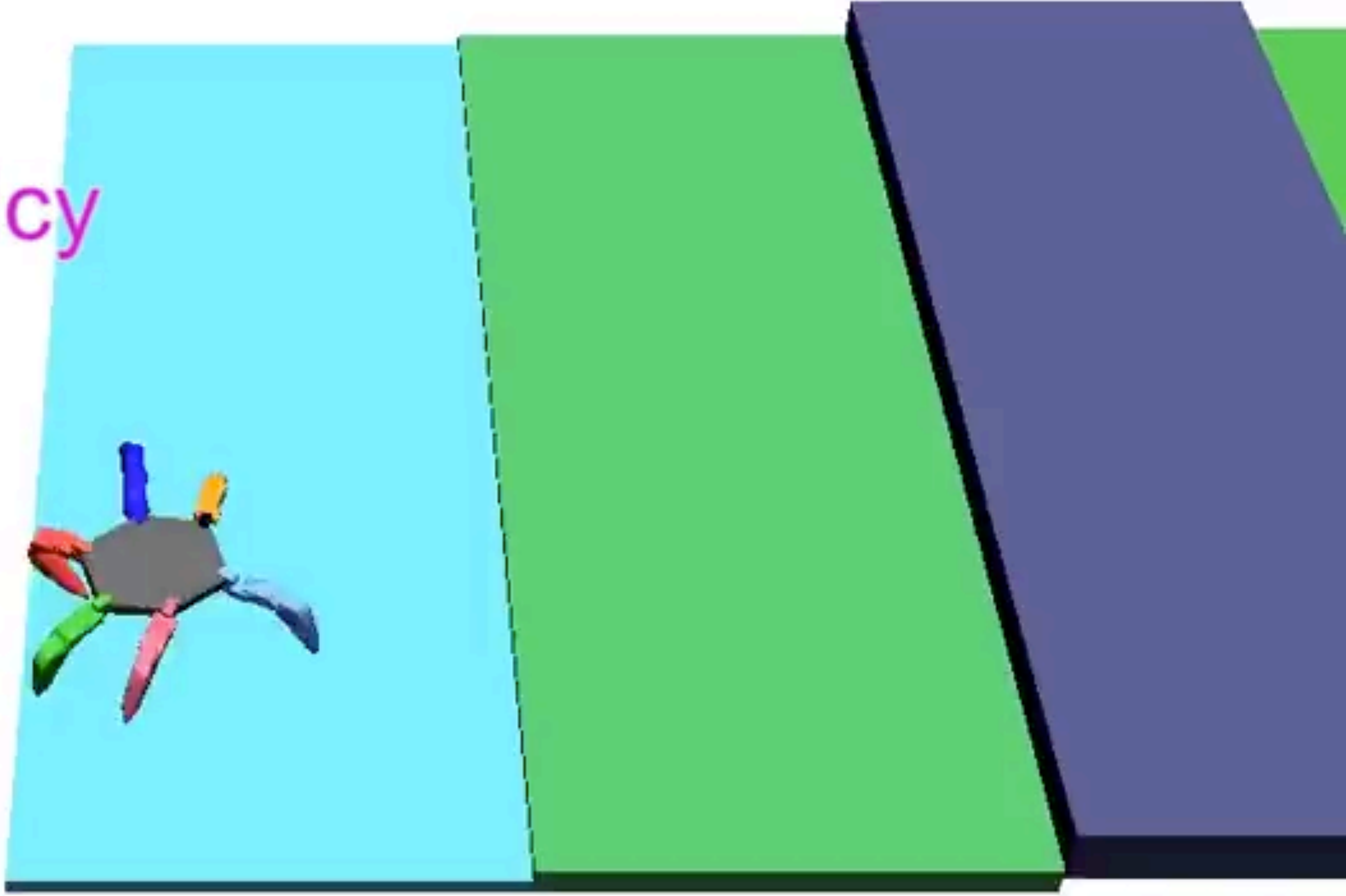
Fig. 1. Time lapse of the OpenAI Walker2D model walking for 360 time steps using a policy and succeeding while intact (top), failing due to a joint being limited (middle), and succeeding again post-adaptation despite the limited joint marked in red by selecting an appropriate policy using our proposed method (bottom).

high-dimensional action and state spaces [1], [2]. The success of these studies relies on a large number of samples in the orders of millions, so re-training the policy after the environment change is unrealistic. Some methods avoid re-training by increasing the robustness of an acquired policy and thus increasing adaptability. In robust adversarial RL, for example, an agent is trained to operate in the presence of a destabilizing adversary that applies disturbance forces to the system [3]. However, using only a single policy limits the adaptability of the robot to large modifications which requires completely different behaviors compared to its learning environment.

We propose Map-based Multi-Policy Reinforcement Learning (MMPRL), which trains many different policies by combining DRL and the idea of using a behavior-performance map [4]. MMPRL aims to search and store multiple possible policies which have different behavioral features while maximizing the expected reward in advance in order to adapt to the unknown environment change. For example, there are various ways for multi-legged robots to move forward: walking, jumping, running, side-walking, etc. In this example, only the fastest policy would survive when using ordinary RL, whereas MMPRL saves all of them as long as they have different behavioral features. These policies are stored into a multi-dimensional discrete map according to its behavioral feature. As a result, adaptation can be done within reasonable time without re-training the robot, but just by searching an appropriate pre-trained policy from the map using an efficient method like Bayesian optimization, see Figure 1. We show that, using MMPRL, robots are able to quickly adapt to large changes with little knowledge about what kind of accidents will happen.

Deep reinforcement learning is promising for robot control, but existing methods cannot deal with large unexpected changes

MMPRL
Stairs, initial policy



Other Applications of Quality Diversity Algorithms

Go-Explore

A new approach for hard-exploration problems

Adrien Ecoffet

Joost Huizinga

Joel Lehman

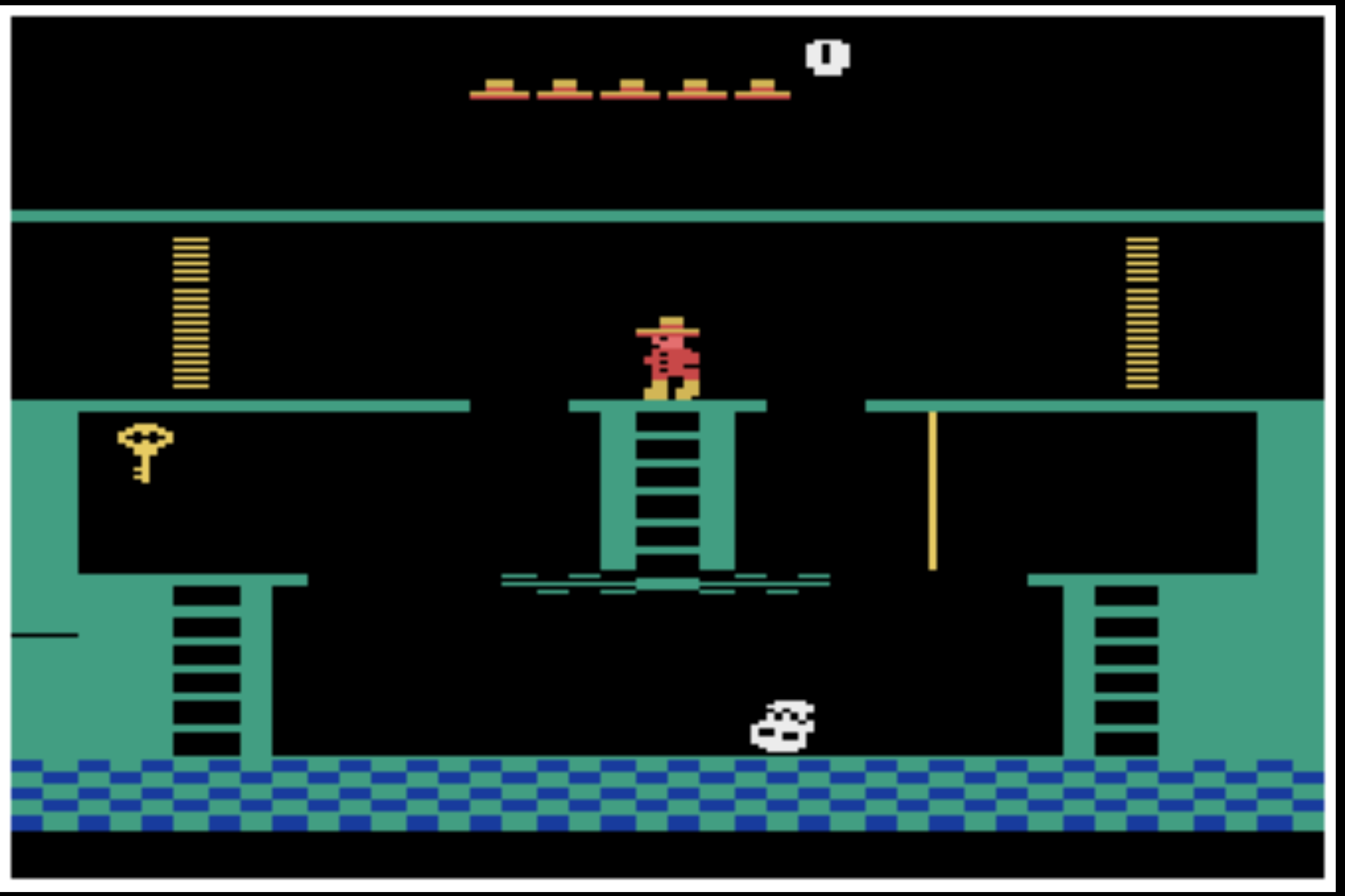
Ken Stanley*

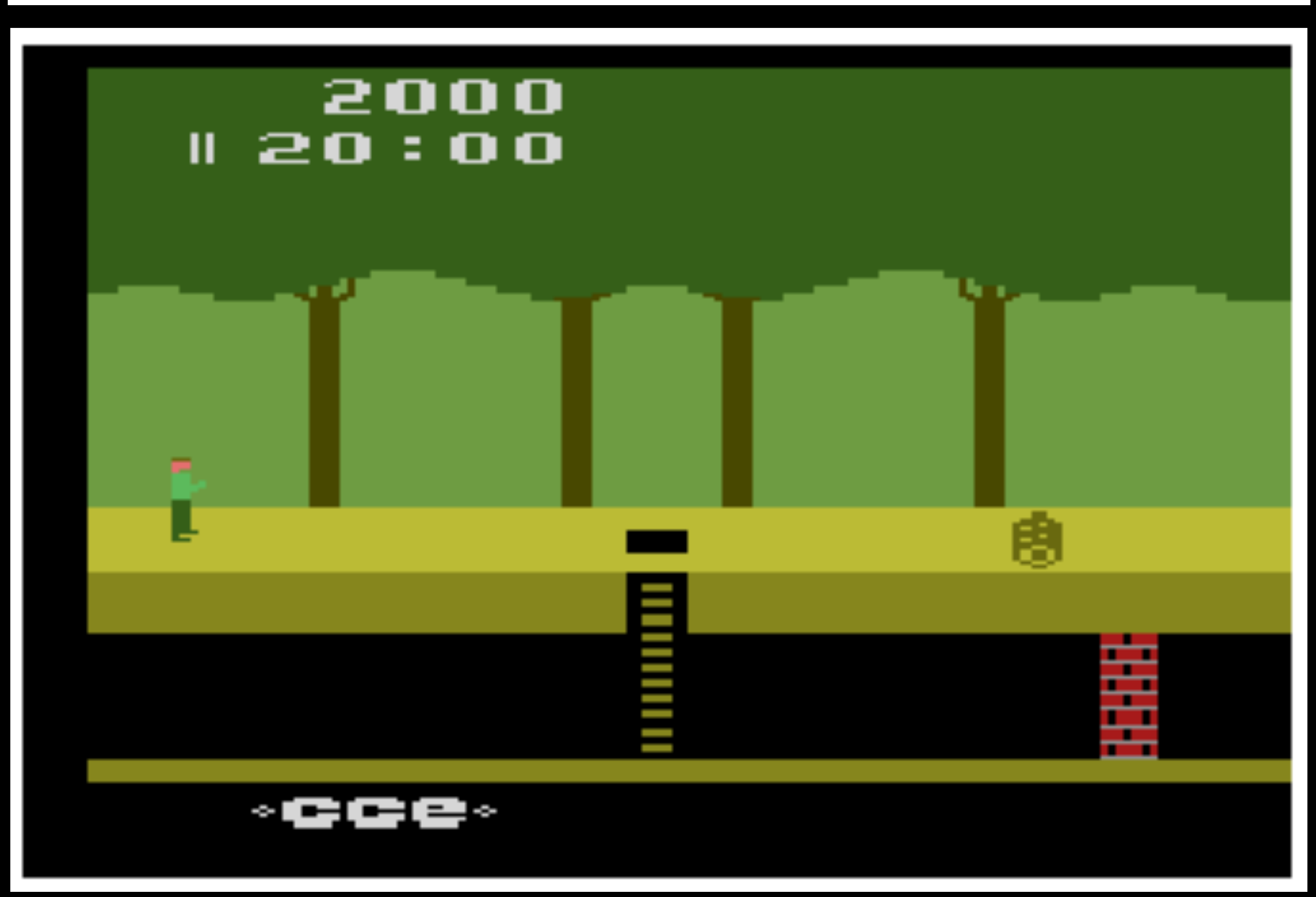
Jeff Clune*

UBER AI Labs

Grand Challenge in Deep RL
Effective Exploration

- Hard-exploration problems
  - Sparse-reward problems
    - rare feedback
    - Montezuma's Revenge
  - Deceptive problems
    - wrong feedback (wrt global optimum)

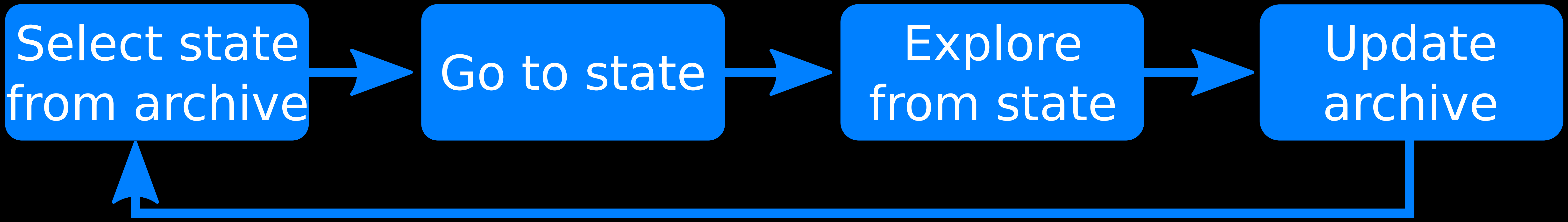




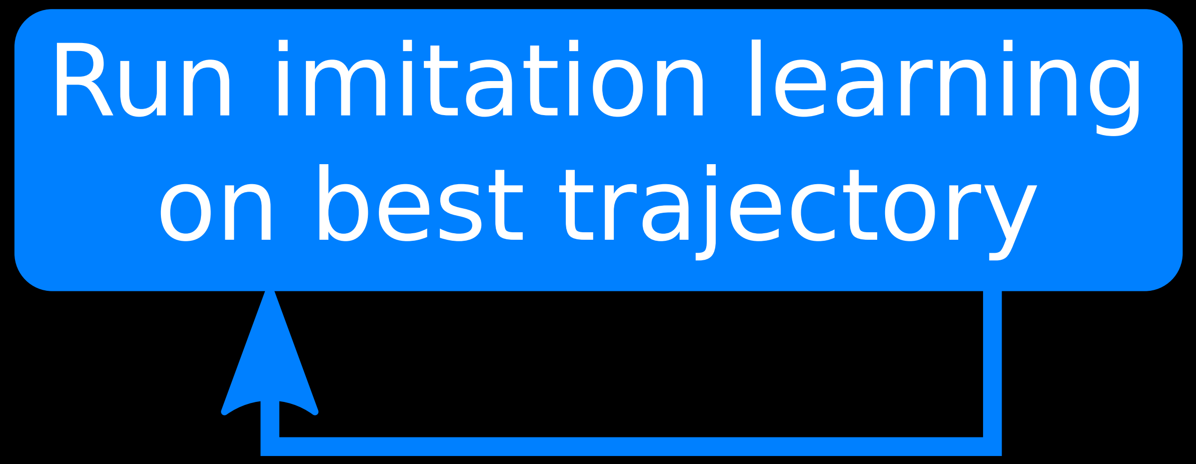
Go-Explore

Separates learning a solution into two phases

Phase 1: Explore Until Solved

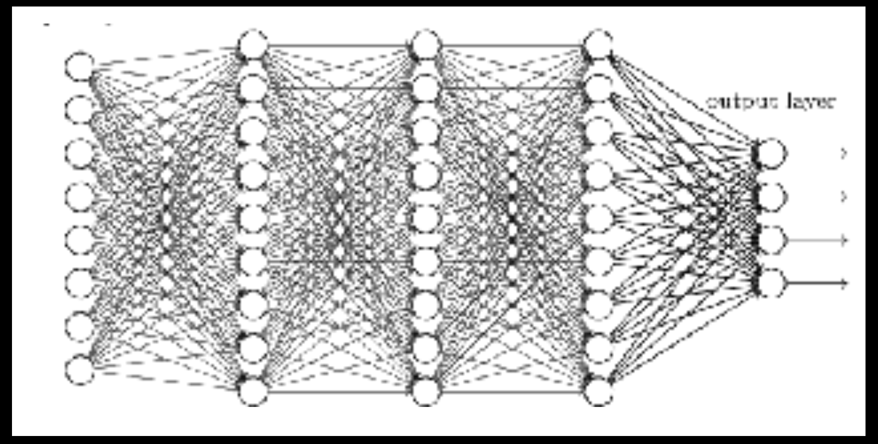


Phase 2: Robustify
(if necessary)



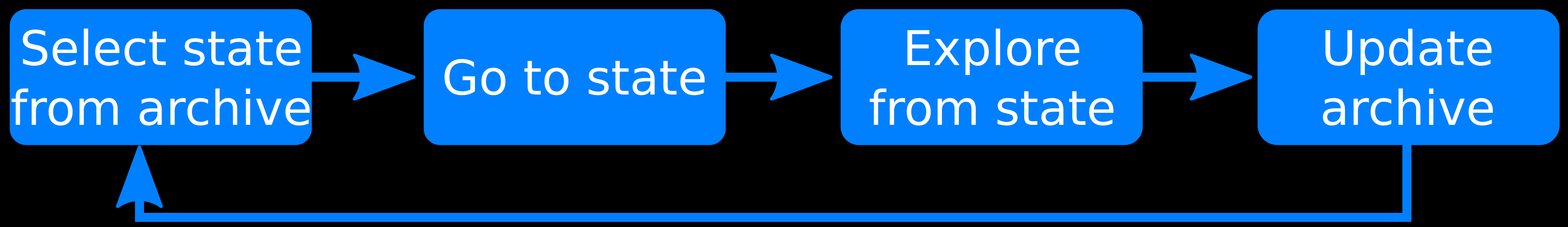
current work:
exploits deterministic training, no neural networks

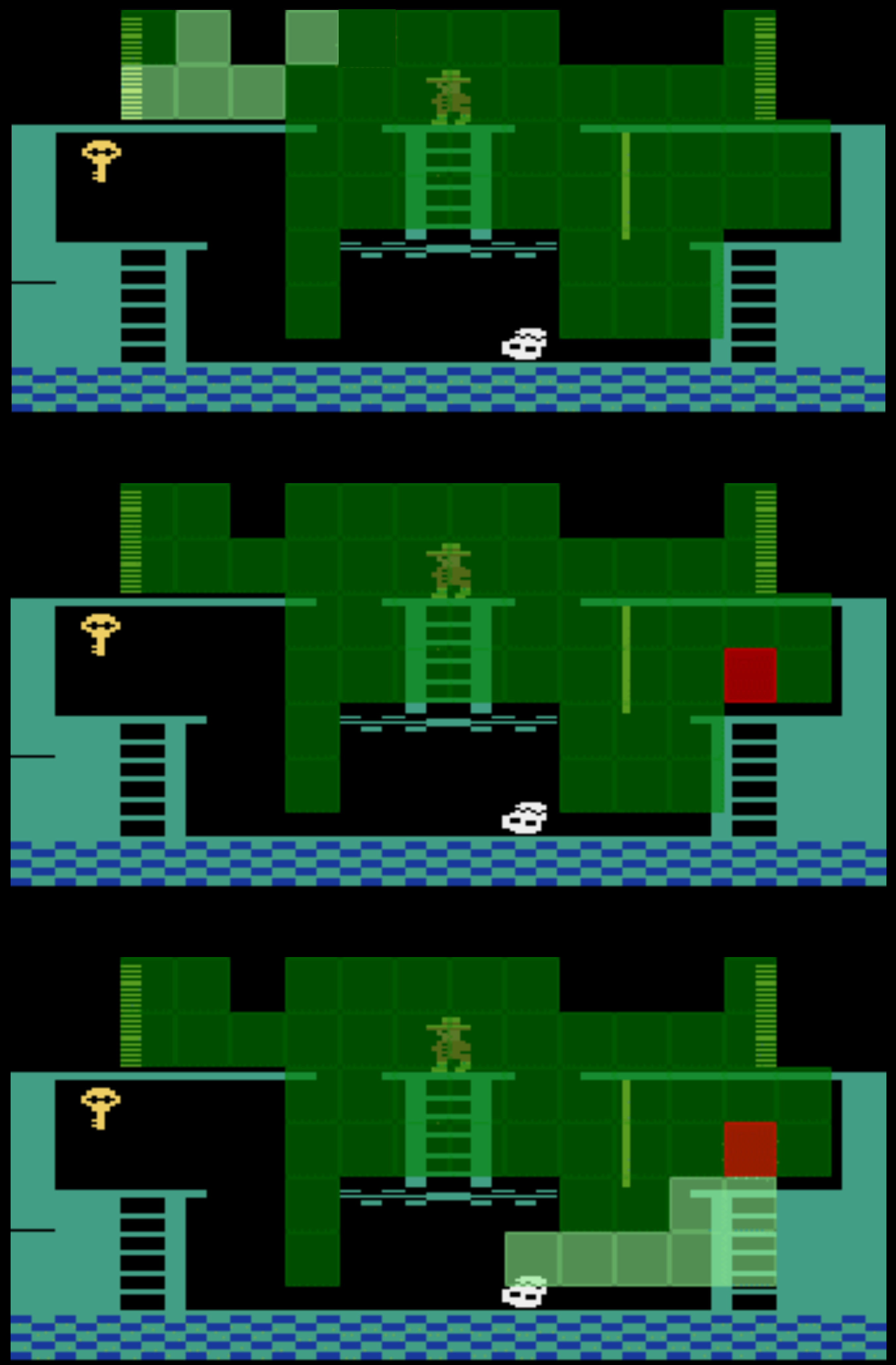
produces neural network
robust to stochasticity



Go-Explore: Phase 1

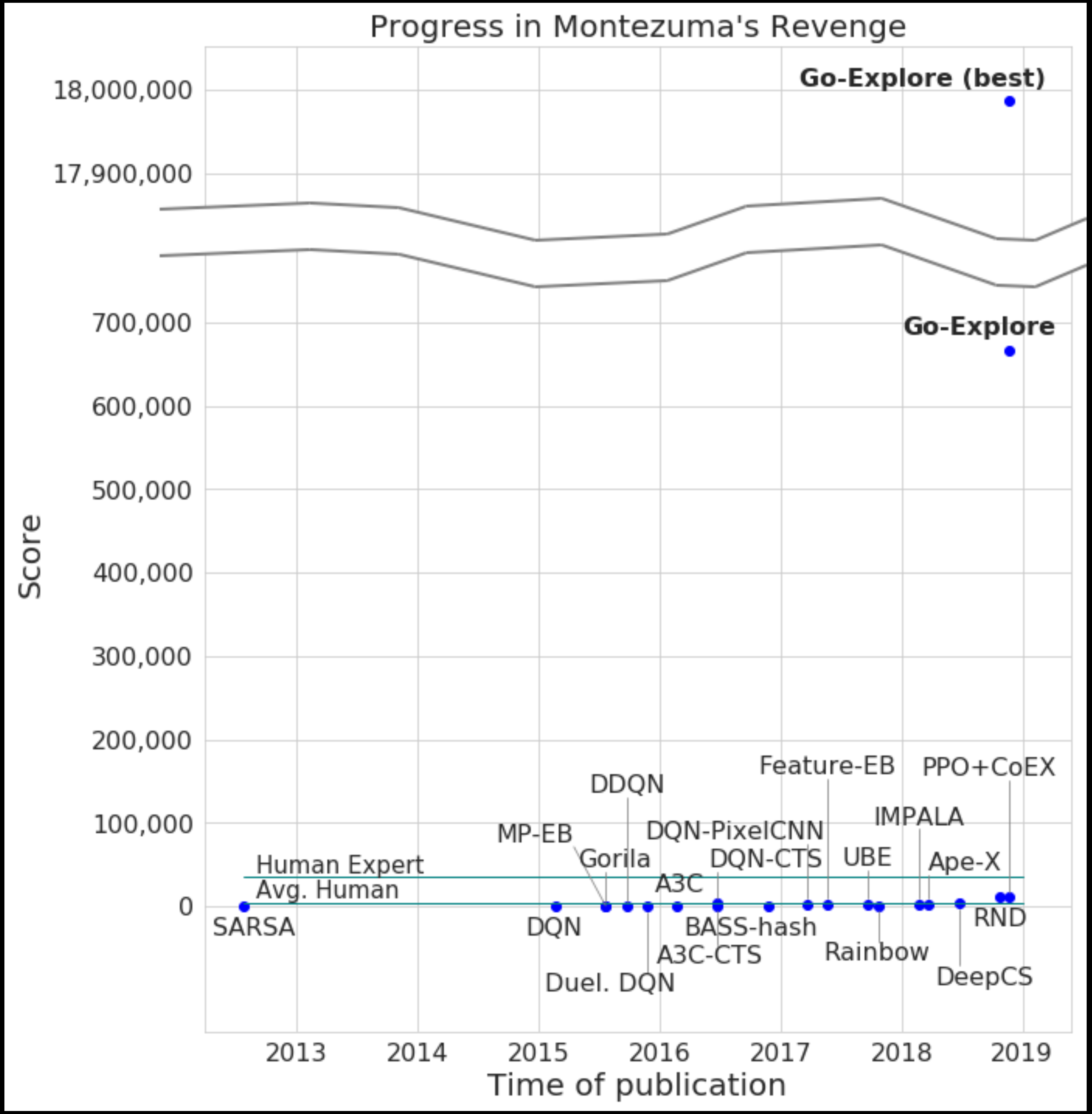
- Phase 1: explore until solved
  A. choose a state from archive
  B. Go back to it
  C. Explore from it
  D. add newly found states to archive
     - if better, replace old way of reaching state





An enhanced version of MAP-Elites

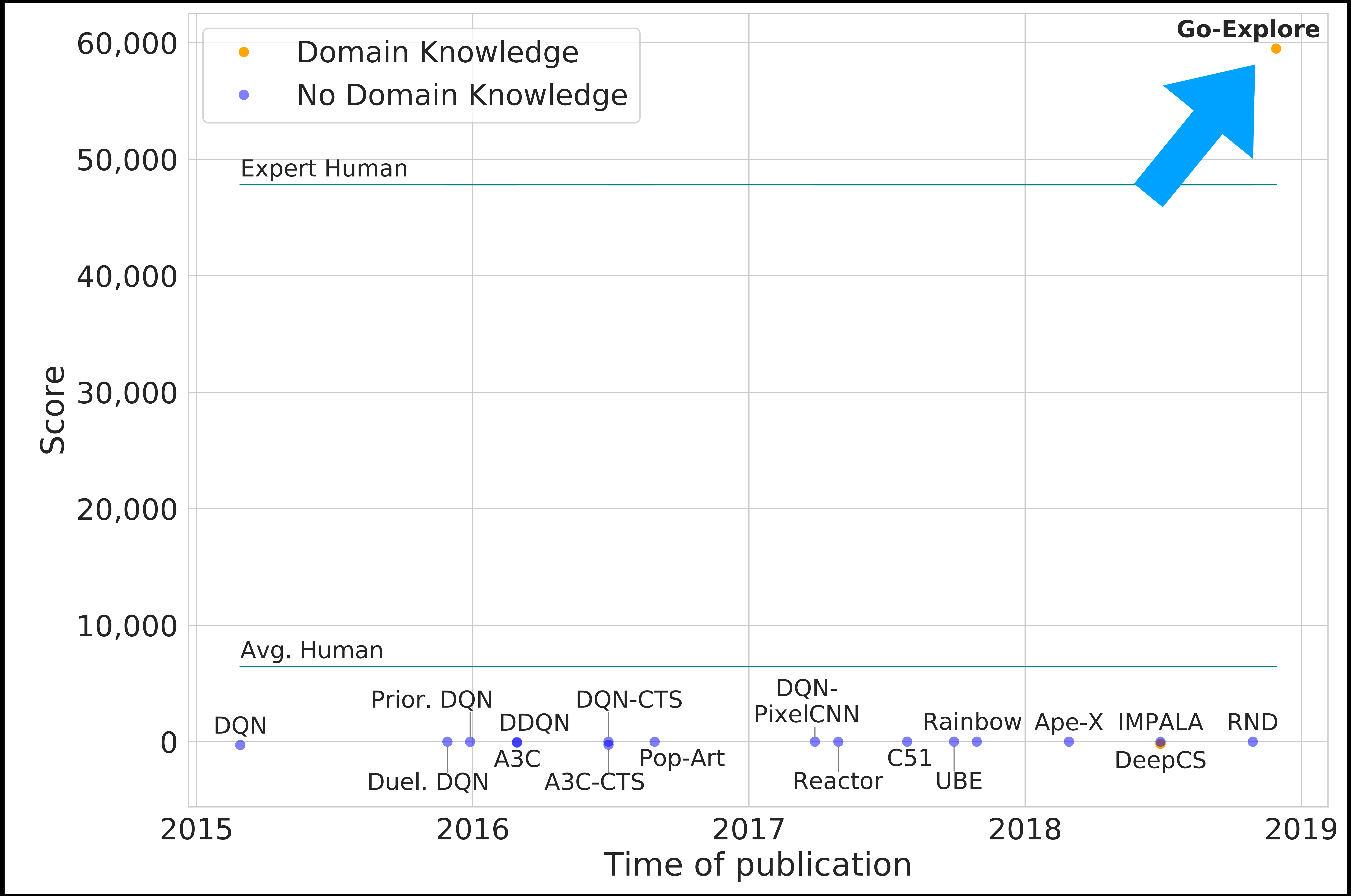
Montezuma's Revenge Results



- Average score: 660,000
- Best Go-Explore policy
  - scores ~18 million
  - solved 1,141 levels
- Beats human world record
  - 1,219,200

Note: exploits deterministic training (unlike Burda et al. 2018)

Pitfall Results



- no prior scores > 0
  - without:
    - fully deterministic test environment
    - or human demonstration
- average score: 59,000
- max: 107,000
- significantly advances state of the art

Robotics

- Solve hard problems in simulation
  - “Robot, find survivors”

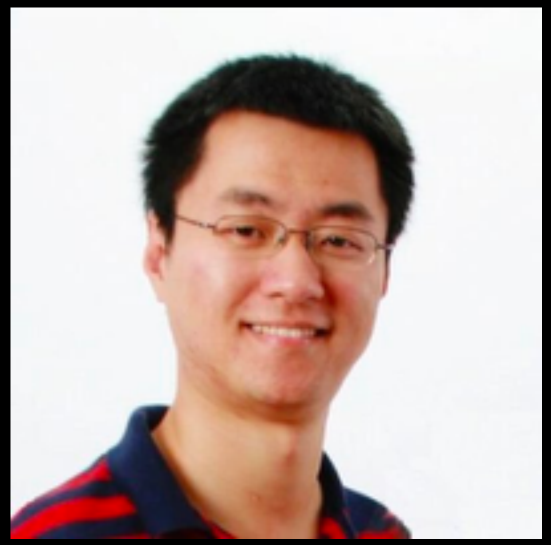
solve in deterministic simulator → robustify in stochastic simulator → transfer to reality → learn in reality (optional)

e.g. intelligent trial & error

Cully, Tarapore, Mouret, & Clune

Automatically generating training data and training environments

- Paired Open-Ended Trailblazer (POET)
- Generates Challenges and Solutions



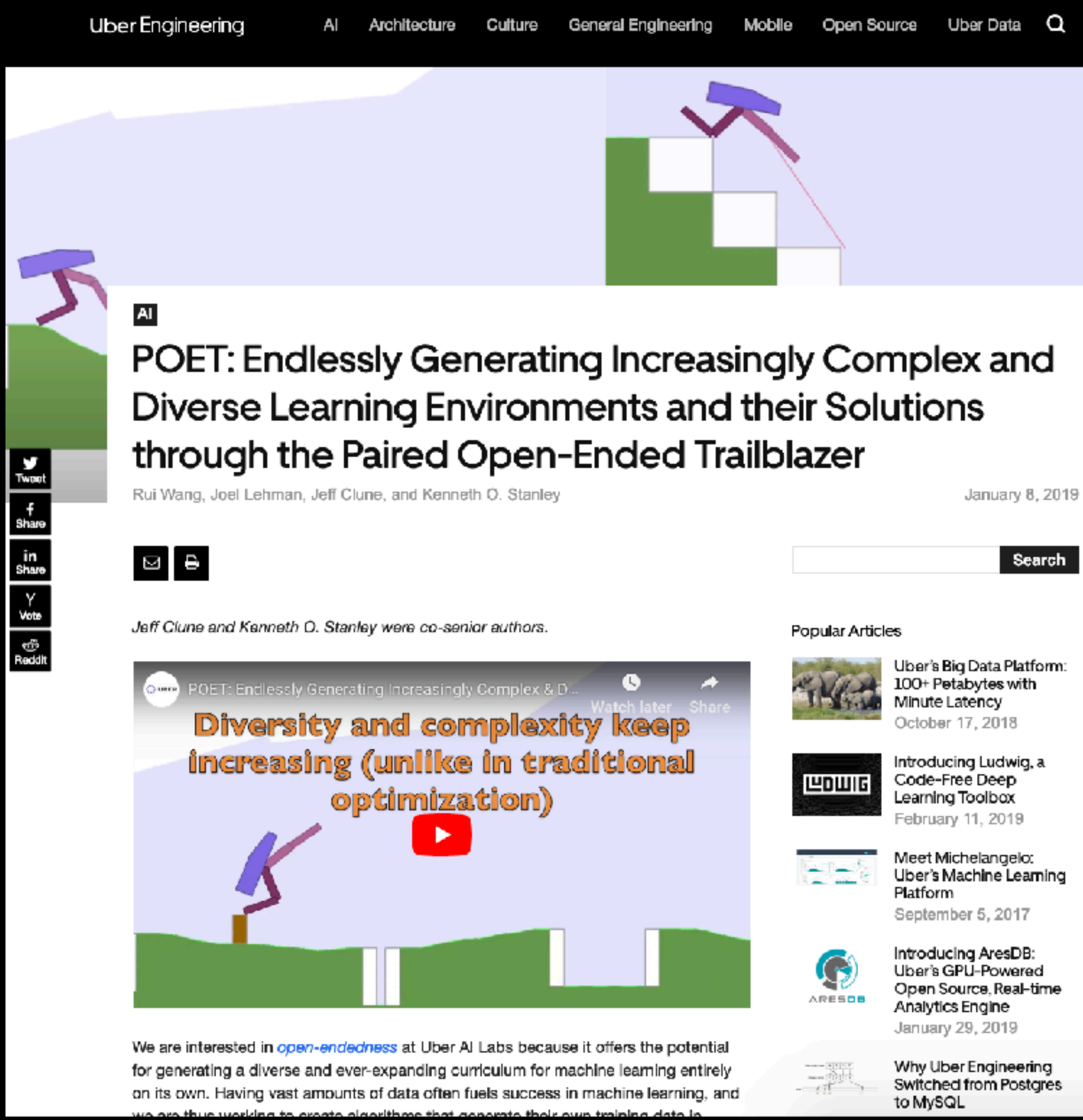
Rui Wang

Joel Lehman

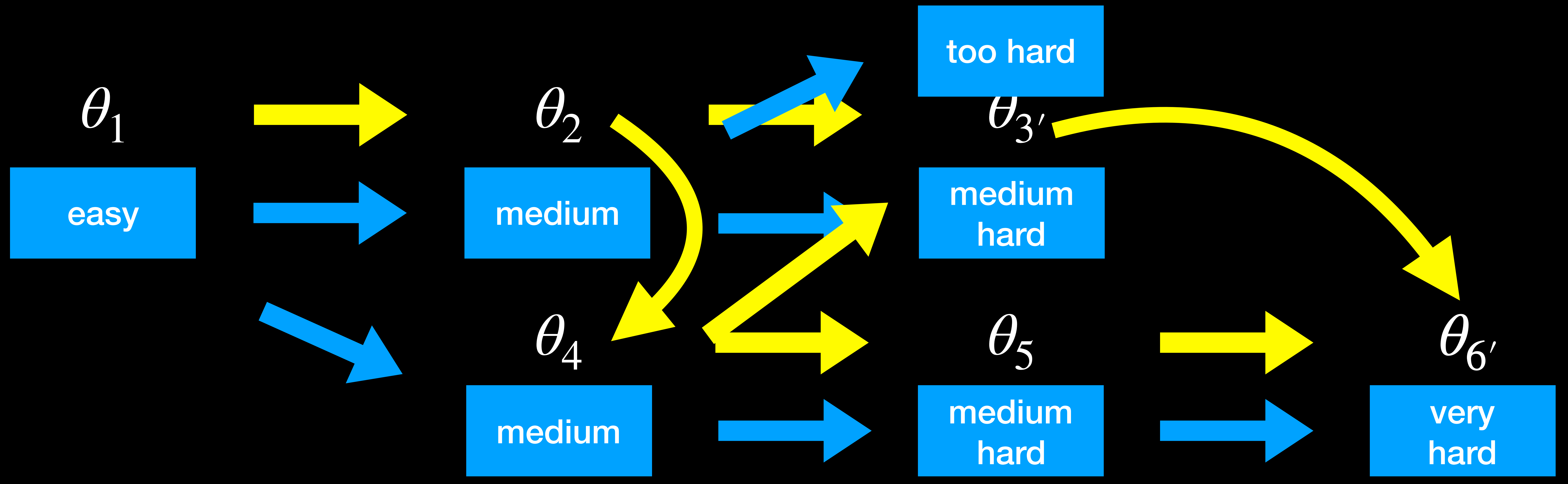
Jeff Clune*

Ken Stanley*

*Co-senior authors



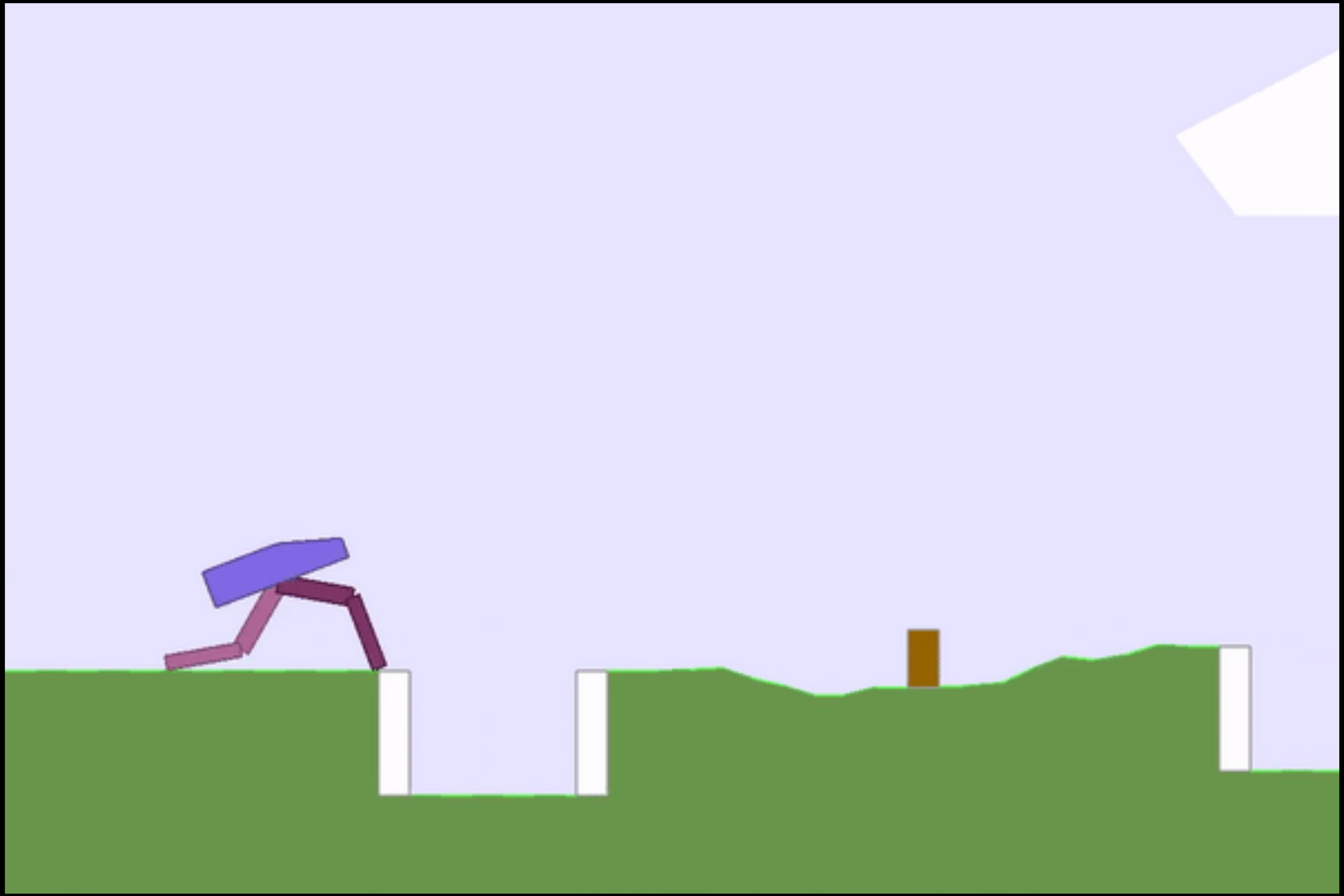
POET



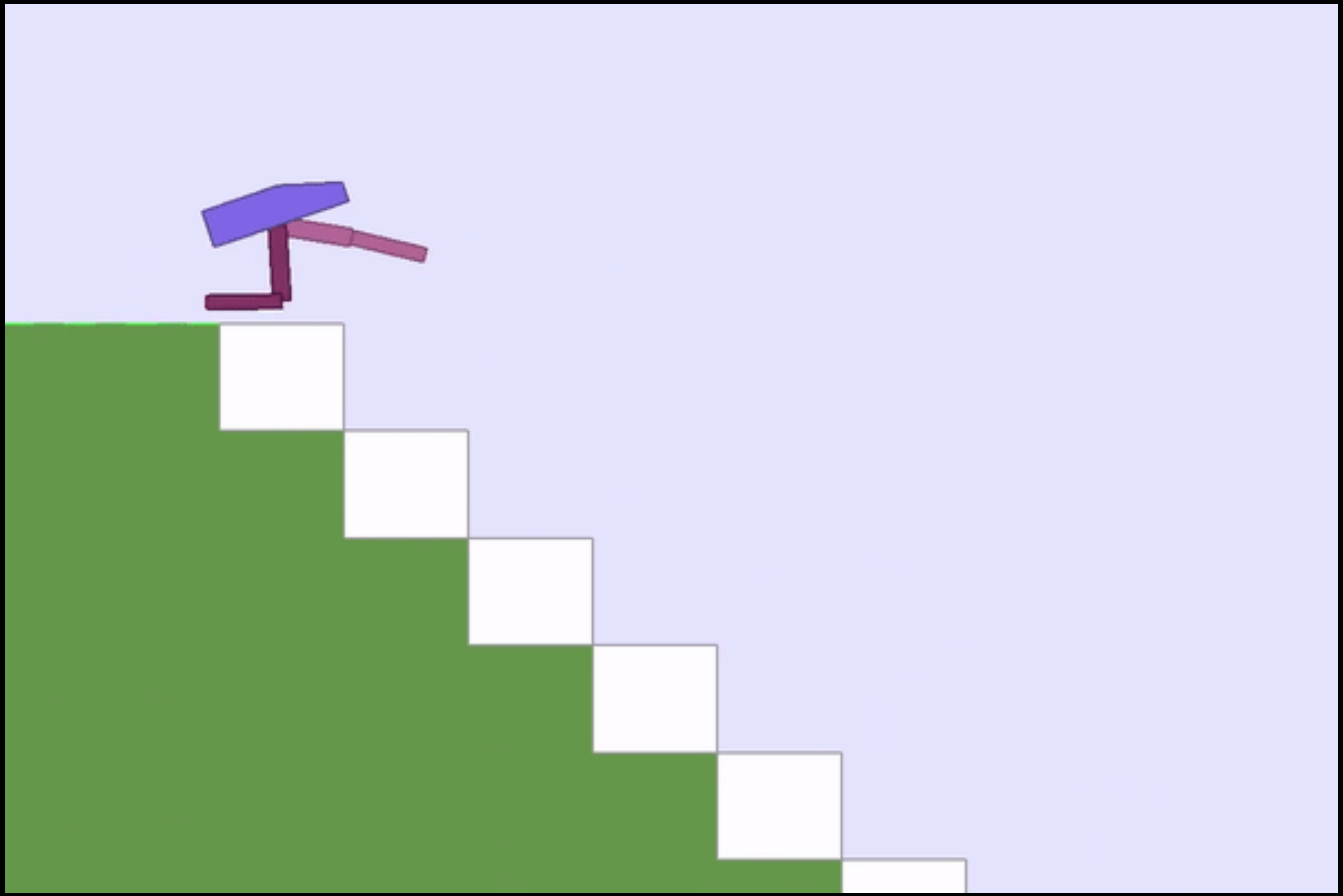
direct optimization fails

direct-path curriculum fails

POET



POET




298
207
304
311
349
309

Conclusions: Intelligent Trial & Error

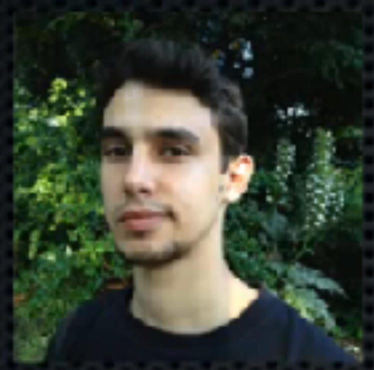
- State of the Art Robot Damage Recovery
  - adaptation, more broadly
- Adapts in < 2 minutes
- Combines
  - expensive creativity of optimization (e.g. deep RL), in simulation
  - with data efficiency of Bayesian optimization, in the real world
- Shows benefits of learning diverse, high-performing sets of policies: “Quality Diversity Algorithms”

intuitions about different ways to move
MAP-Elites

few, intelligent tests
Bayesian Optimization

pick one that works despite injury
found > X% of best

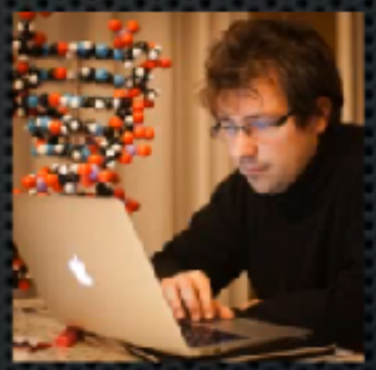
Thanks!



Antoine Cully
UPMC Université
France

Jeff Clune
University of Wyoming
USA

Danesh Tarapore
UPMC Université
France



Jean-Baptiste Mouret
UPMC Université
France

NSF
CAREER Award

UNIVERSITY OF WYOMING

UBER AI Labs