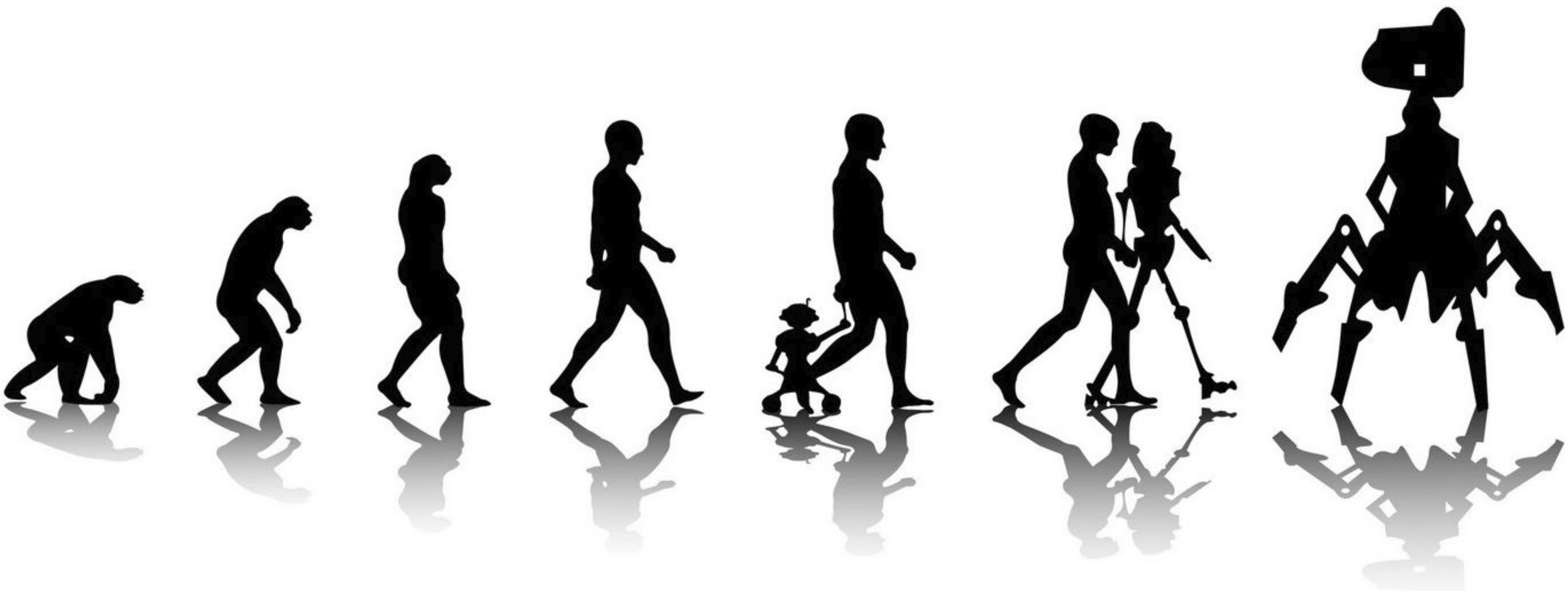


Modern Robotics: Evolutionary Robotics
COSC 4560 / COSC 5560

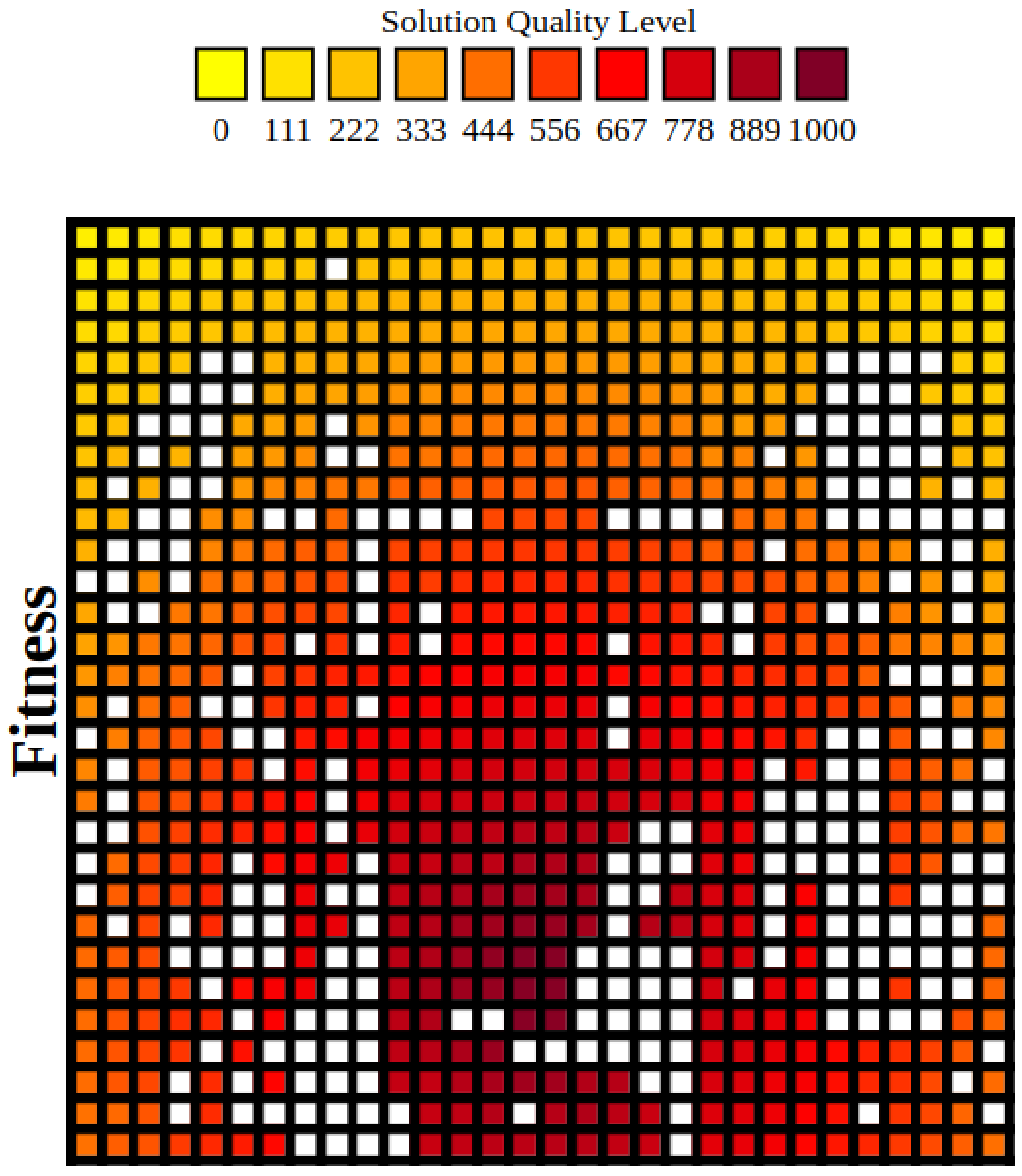
Professor Cheney
3/7/18

Quality Diversity with MAP-Elites

Illuminating search spaces by mapping elites

Jean-Baptiste Mouret[1] and Jeff Clune[2]

[1]Université Pierre et Marie Curie-Paris 6, CNRS UMR 7222, France
[2]University of Wyoming, USA


Solution Quality Level
0
111
222
333
444
556
667
778
889
1000
Fitness

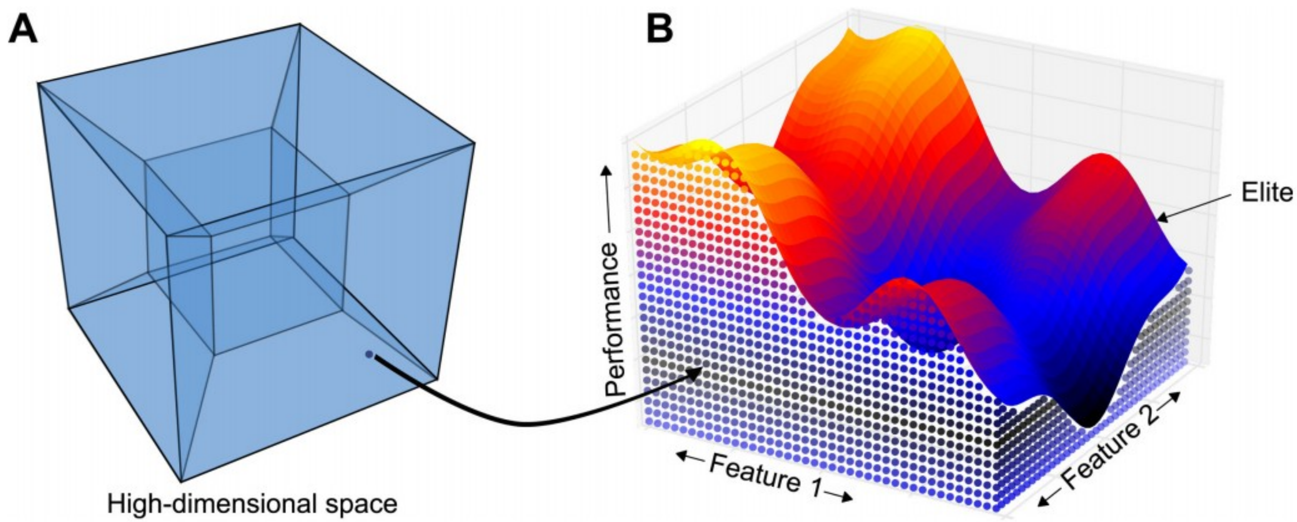


Fig. 1. The MAP-Elites algorithm searches in a high-dimensional space to find the highest-performing solution at each point in a low-dimensional feature space, where the user gets to choose dimensions of variation of interest that define the low dimensional space. We call this type of algorithm an “illumination algorithm”, because it illuminates the fitness potential of each area of the feature space, including tradeoffs between performance and the features of interest. For example, MAP-Elites could search in the space of all possible robot designs (a very high dimensional space) to find the fastest robot (a performance criterion) for each combination of height and weight.

procedure MAP-ELITES ALGORITHM (SIMPLE, DEFAULT VERSION)

```
    (𝒫 ← ∅, 𝒳 ← ∅)                        ▷ Create an empty, N-dimensional map of elites: {solutions 𝒳 and their performances 𝒫}
    for iter = 1 → I do                    ▷ Repeat for I iterations.
        if iter < G then                   ▷ Initialize by generating G random solutions
            x' ← random_solution()
        else                               ▷ All subsequent solutions are generated from elites in the map
            x ← random_selection(𝒳)         ▷ Randomly select an elite x from the map 𝒳
            x' ← random_variation(x)       ▷ Create x', a randomly modified copy of x (via mutation and/or crossover)
        b' ← feature_descriptor(x')        ▷ Simulate the candidate solution x' and record its feature descriptor b'
        p' ← performance(x')               ▷ Record the performance p' of x'
        if 𝒫(b') = ∅ or 𝒫(b') < p' then    ▷ If the appropriate cell is empty or its occupants's performance is ≤ p', then
            𝒫(b') ← p'                     ▷ store the performance of x' in the map of elites according to its feature descriptor b'
            𝒳(b') ← x'                     ▷ store the solution x' in the map of elites according to its feature descriptor b'
    return feature-performance map (𝒫 and 𝒳)
```

Fig. 2. A pseudocode description of the simple, default version of MAP-Elites.

Criteria for Measuring the Algorithms

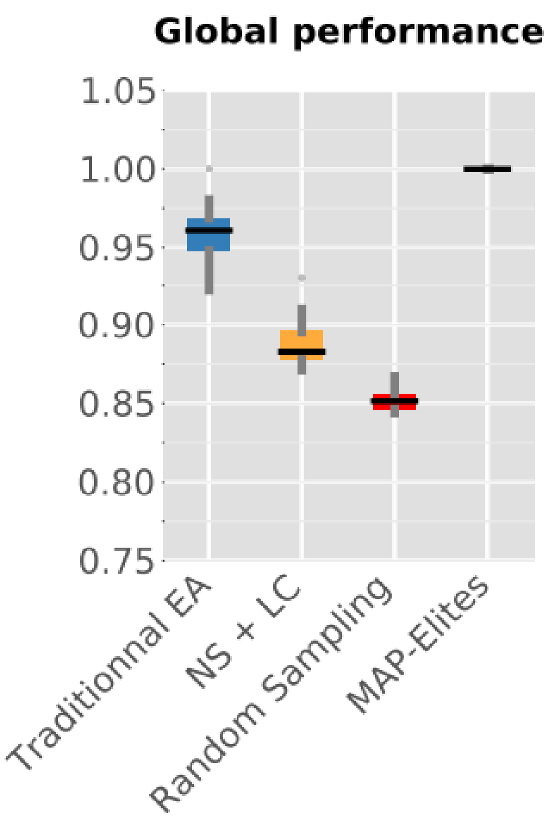
Global Performance: For each run, the single highest-performing solution found by that algorithm anywhere in the search space divided by the highest performance possible in that domain.

Global reliability: For each run, the average across all cells of the highest-performing solution the algorithm found for each cell (0 if it did not produce a solution in that cell) divided by the best known performance for that cell as found by any run of any algorithm.

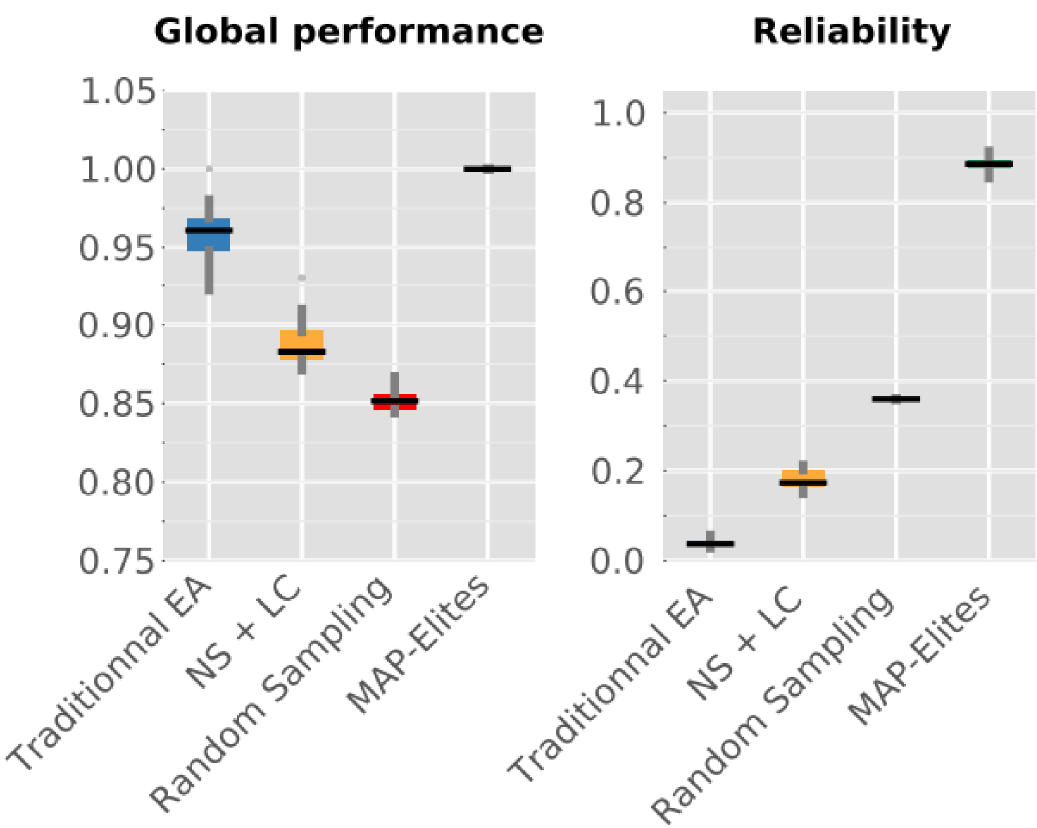
Precision (opt-in reliability): For each run, if (and only if) a run creates a solution in a cell, the average across all such cells of the highest performing solution produced for that cell divided by the highest performing solution any algorithm found for that cell.

Coverage: Measures how many cells of the feature space a run of an algorithm is able to fill of the total number that are possible to fill.

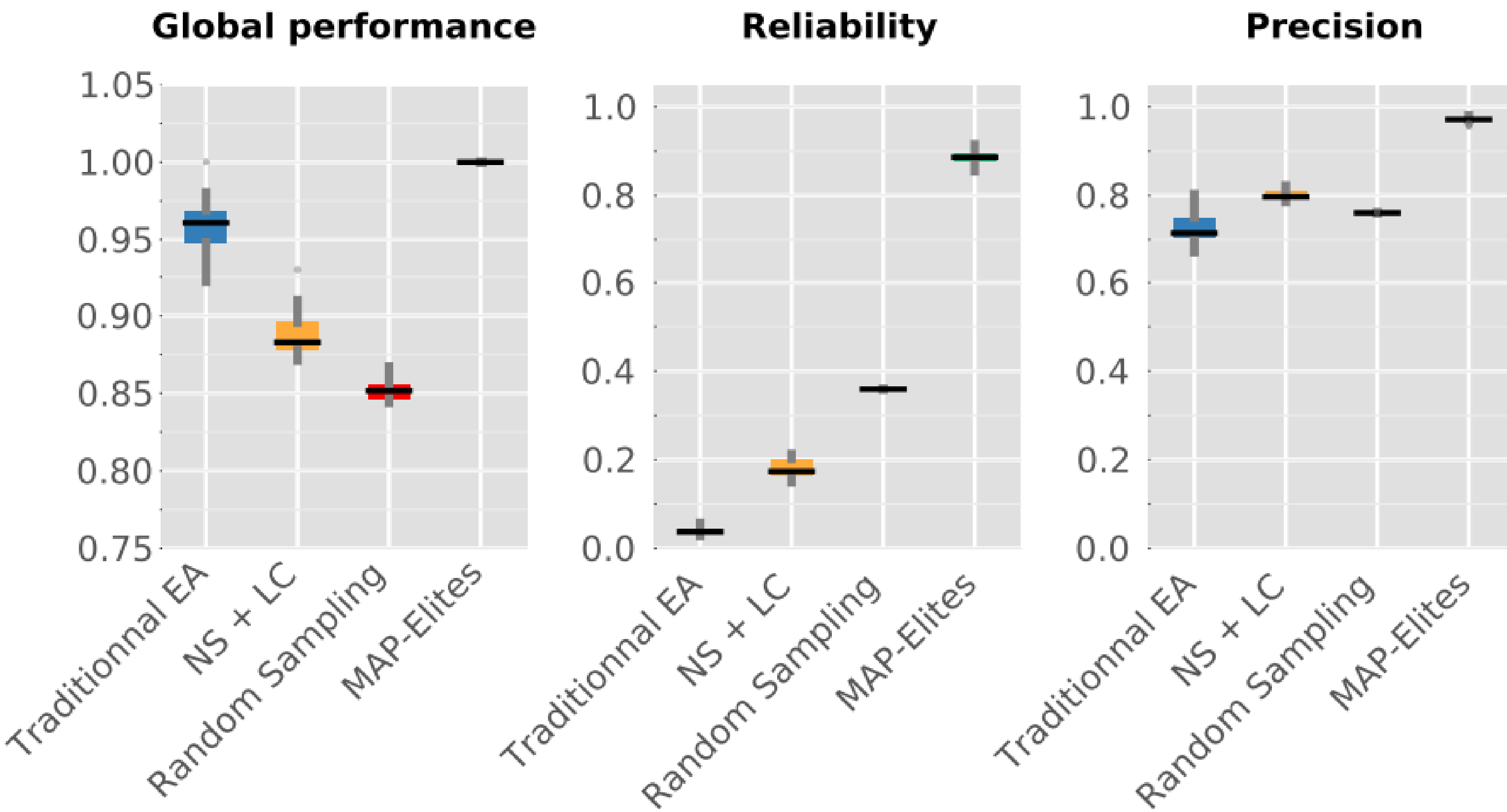
Search space 1: neural networks



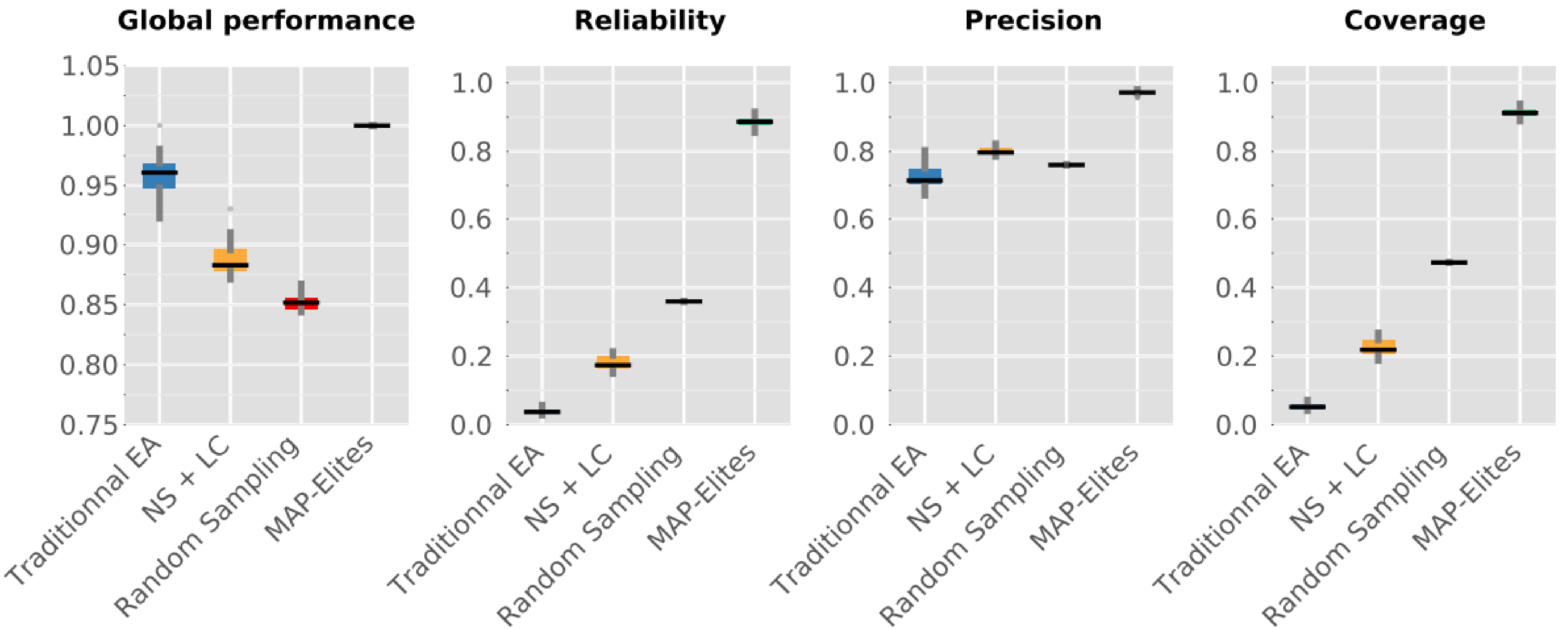
Global performance
1.05
1.00
0.95
0.90
0.85
0.80
0.75
Traditionnal EA
NS + LC
Random Sampling
MAP-Elites

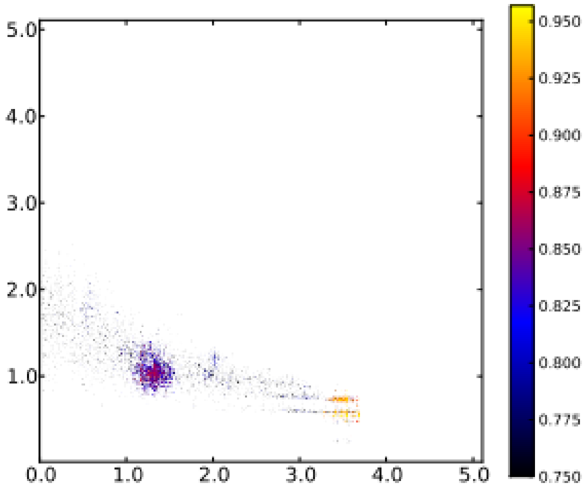


Global performance
Reliability
1.05
1.00
0.95
0.90
0.85
0.80
0.75
1.0
0.8
0.6
0.4
0.2
0.0
Traditionnal EA
NS + LC
Random Sampling
MAP-Elites

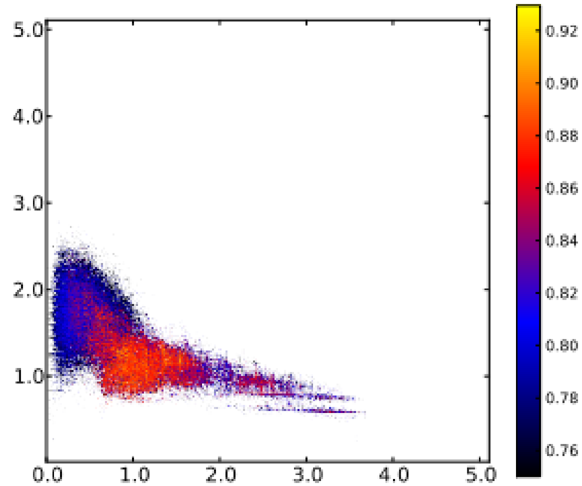


Global performance
Reliability
Precision
1.05
1.00
0.95
0.90
0.85
0.80
0.75
1.0
0.8
0.6
0.4
0.2
0.0
Traditionnal EA
NS + LC
Random Sampling
MAP-Elites

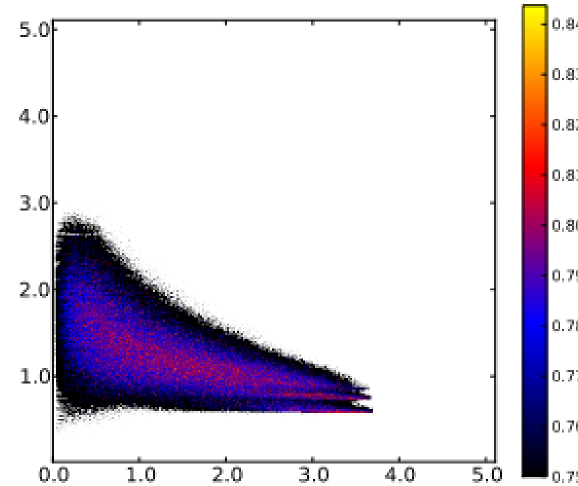

Global performance
Reliability
Precision
Coverage
1.05
1.00
0.95
0.90
0.85
0.80
0.75
1.0
0.8
0.6
0.4
0.2
0.0
Traditionnal EA
NS + LC
Random Sampling
MAP-Elites



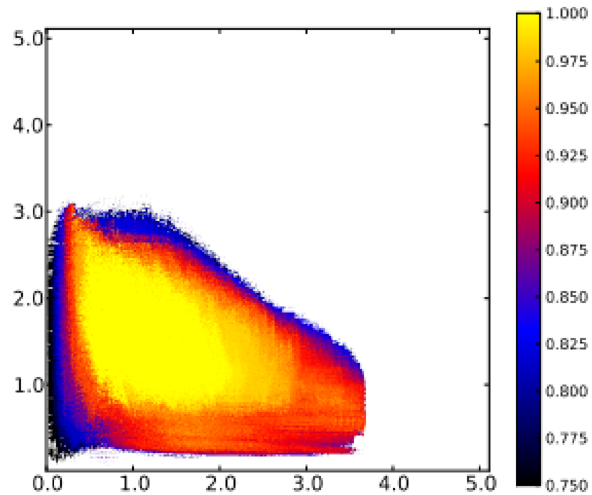
(b) Traditional EA



(c) Novelty Search + Local Competition

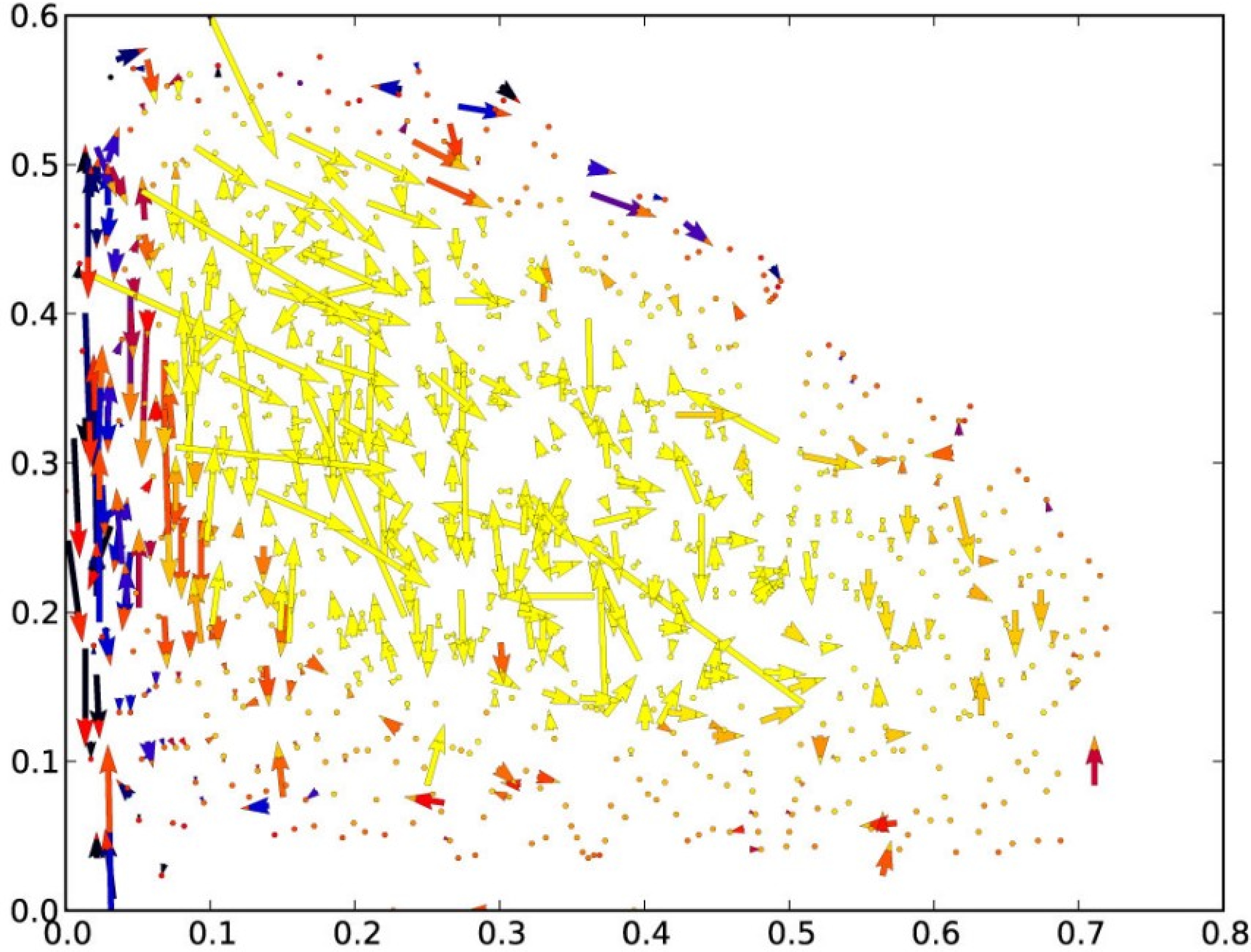


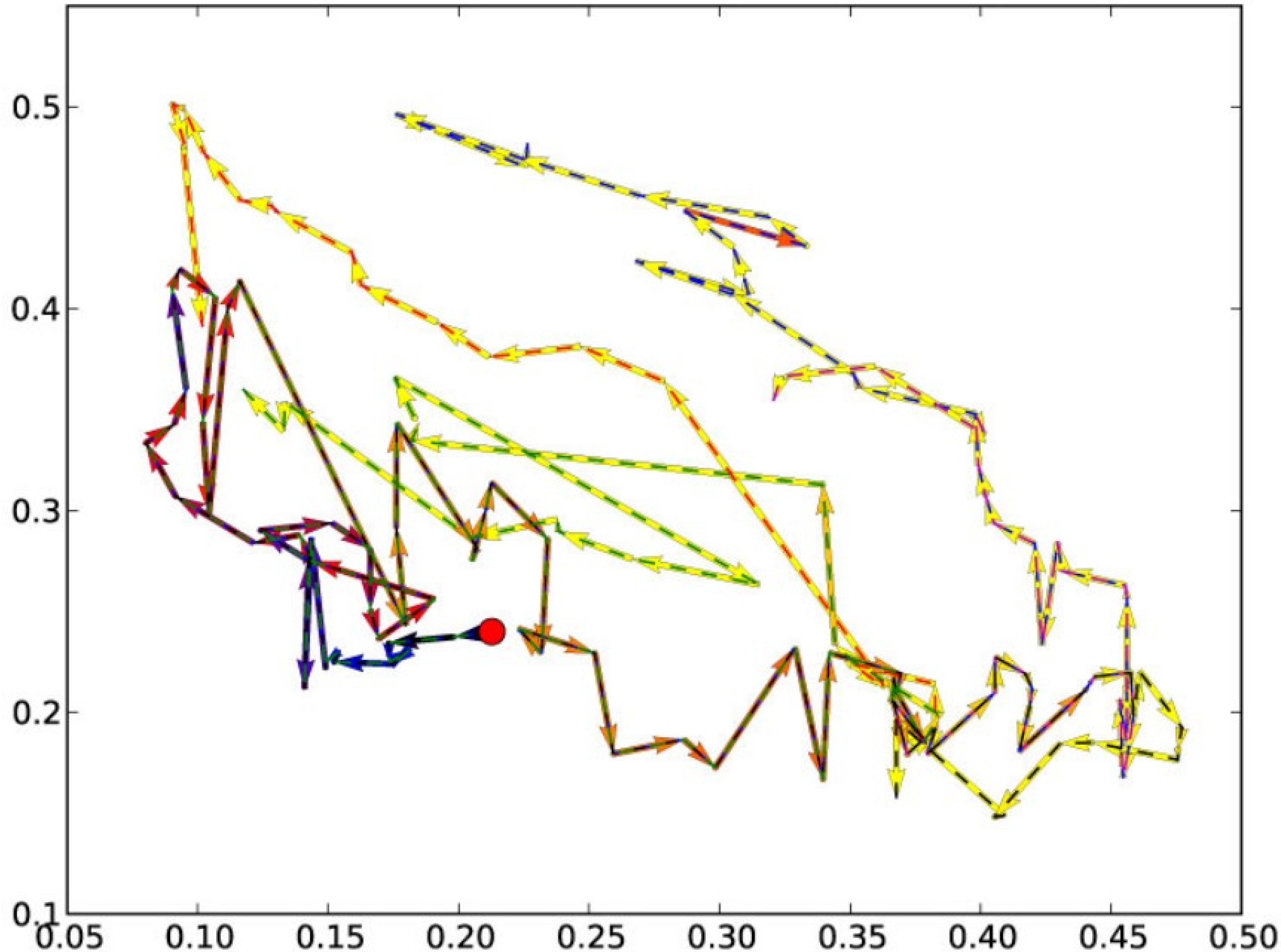
(d) Random Sampling



(e) MAP-Elites

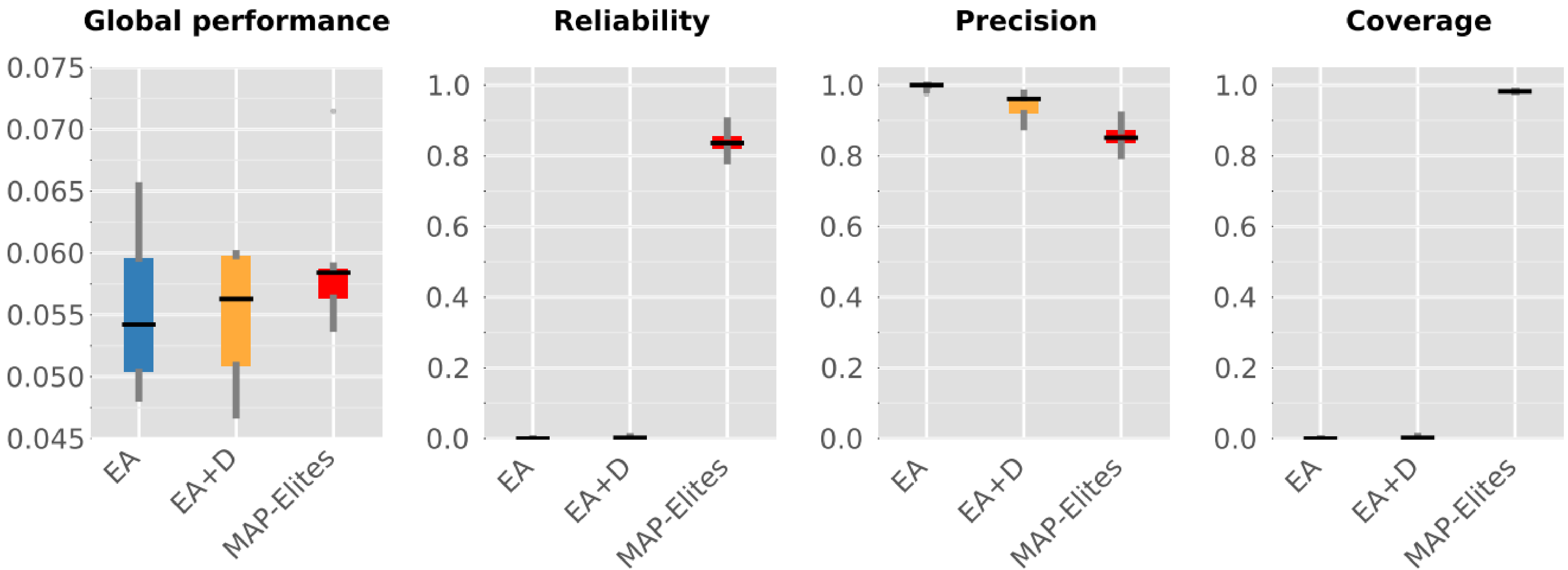
the x-axis is connection cost, the y-axis is modularity


0.6
0.5
0.4
0.3
0.2
0.1
0.0
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8

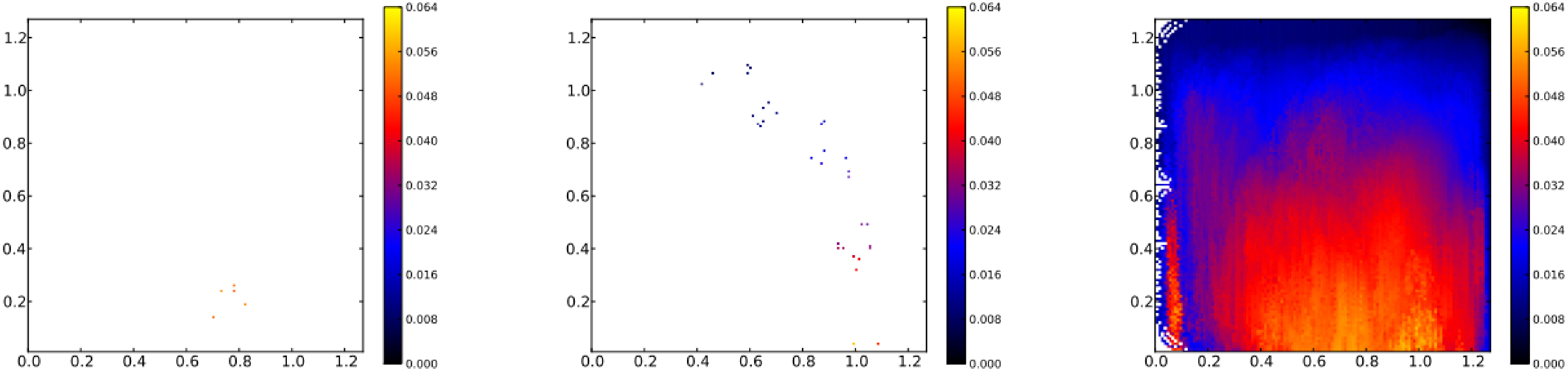


0.5
0.4
0.3
0.2
0.1
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45
0.50

Simulated soft, locomoting robot morphologies



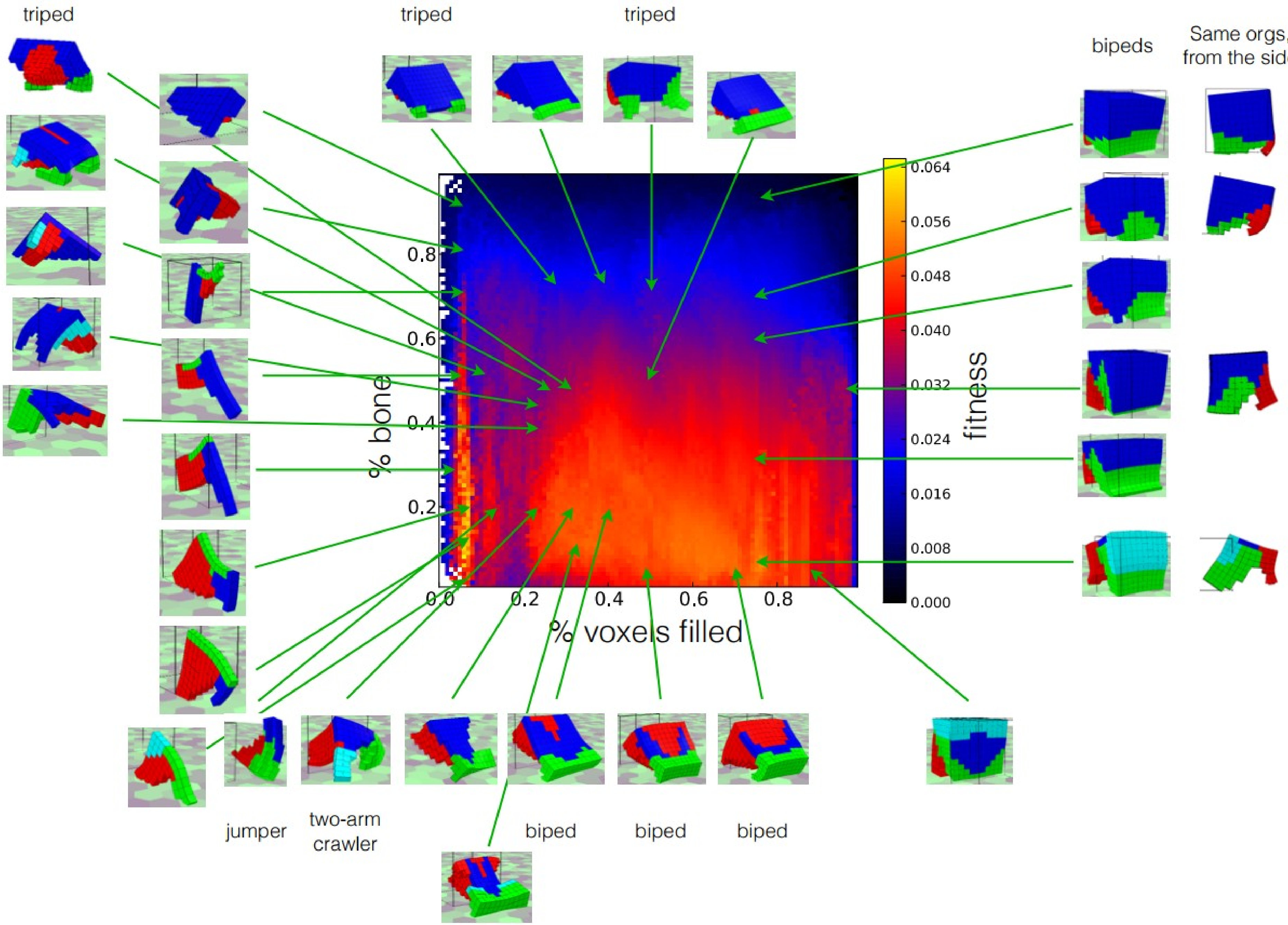
Global performance
Reliability
Precision
Coverage
0.075
0.070
0.065
0.060
0.055
0.050
0.045
1.0
0.8
0.6
0.4
0.2
0.0
EA
EA+D
MAP-Elites



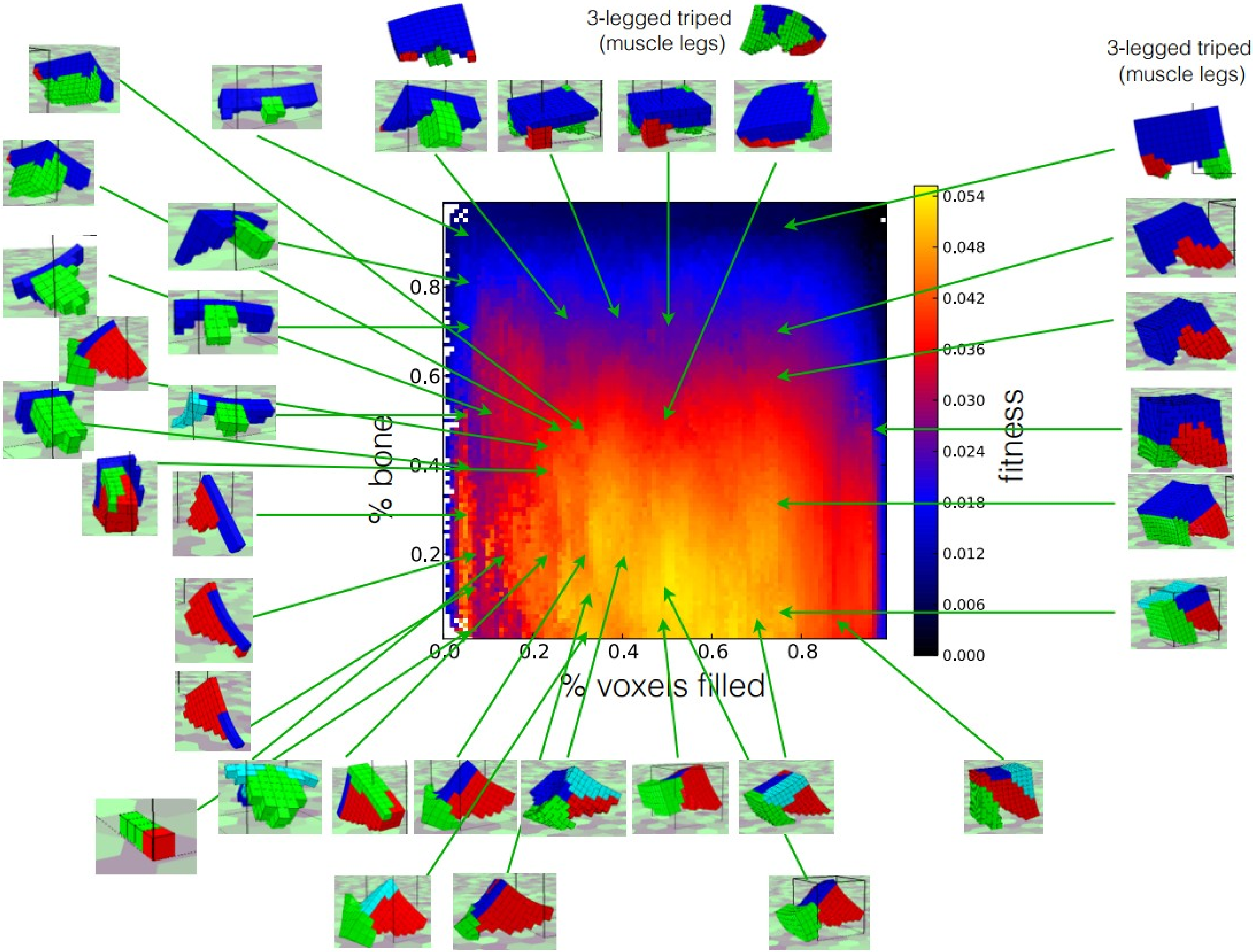
(a) EA

(b) EA + D

(c) MAP-Elites



triped
triped
triped
bipeds
Same orgs, from the side
fitness
0.064
0.056
0.048
0.040
0.032
0.024
0.016
0.008
0.000
% bone
0.8
0.6
0.4
0.2
0.0
0.2
0.4
0.6
0.8
% voxels filled
jumper
two-arm crawler
biped
biped
biped



3-legged triped
(muscle legs)
3-legged triped
(muscle legs)
% bone
% voxels filled
fitness
0.0
0.2
0.4
0.6
0.8
0.000
0.006
0.012
0.018
0.024
0.030
0.036
0.042
0.048
0.054

Robots that can adapt like animals

Antoine Cully,[1,2] Jeff Clune,[6] Danesh Tarapore,[1,2] Jean-Baptiste Mouret[1−5,*]

INSIGHT
Machine intelligence

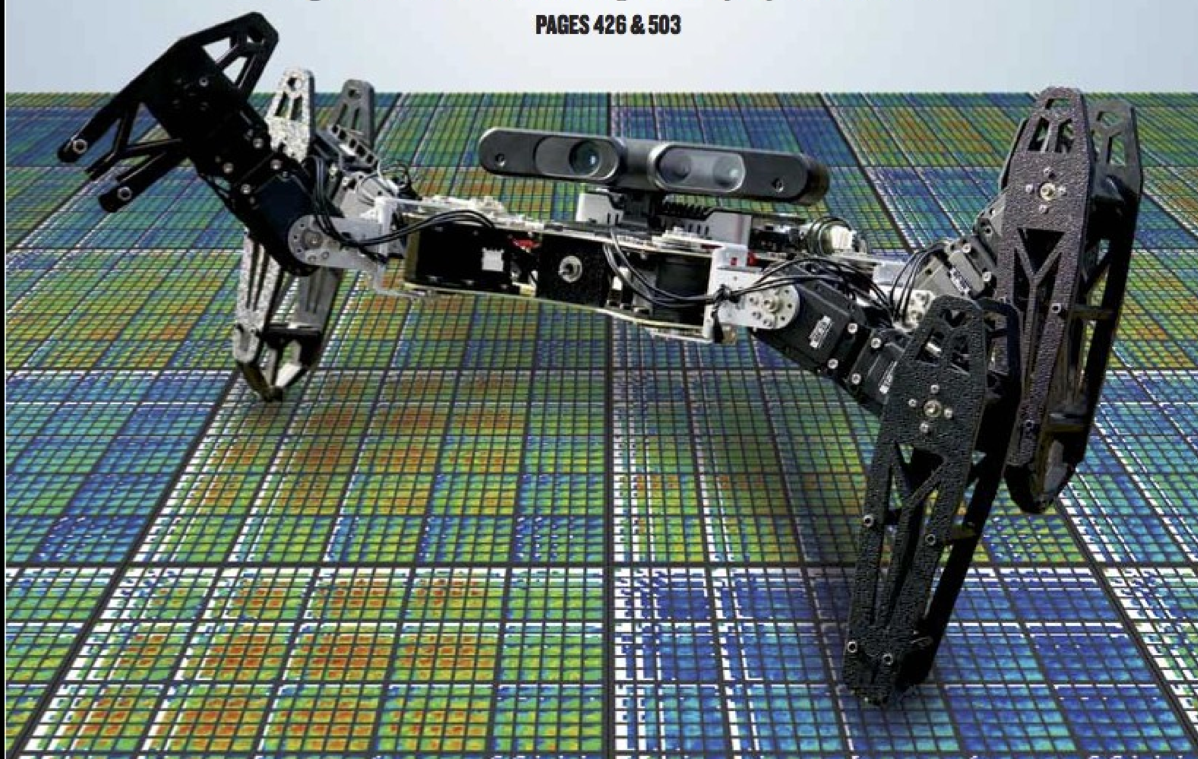
nature

THE INTERNATIONAL WEEKLY JOURNAL OF SCIENCE

Back on its feet

Using an intelligent trial-and-error learning algorithm this robot adapts to injury in minutes

PAGES 426 & 503



COGNITION

WHY FISH NEED TO BE CLEVER

Social behaviours need plenty of brainpower

PAGE 412

ARTIFICIAL INTELLIGENCE

LIVING WITH ROBOTS

AI researchers' ethics prescriptions

PAGE 415

HUMAN EVOLUTION

ANOTHER FACE IN THE CROWD

A new hominin from Ethiopia's middle Pliocene

PAGES 432 & 483

NATURE.COM/NATURE
28 May 2015 £10
Vol. 521, No. 7553
22
9 770028 083095

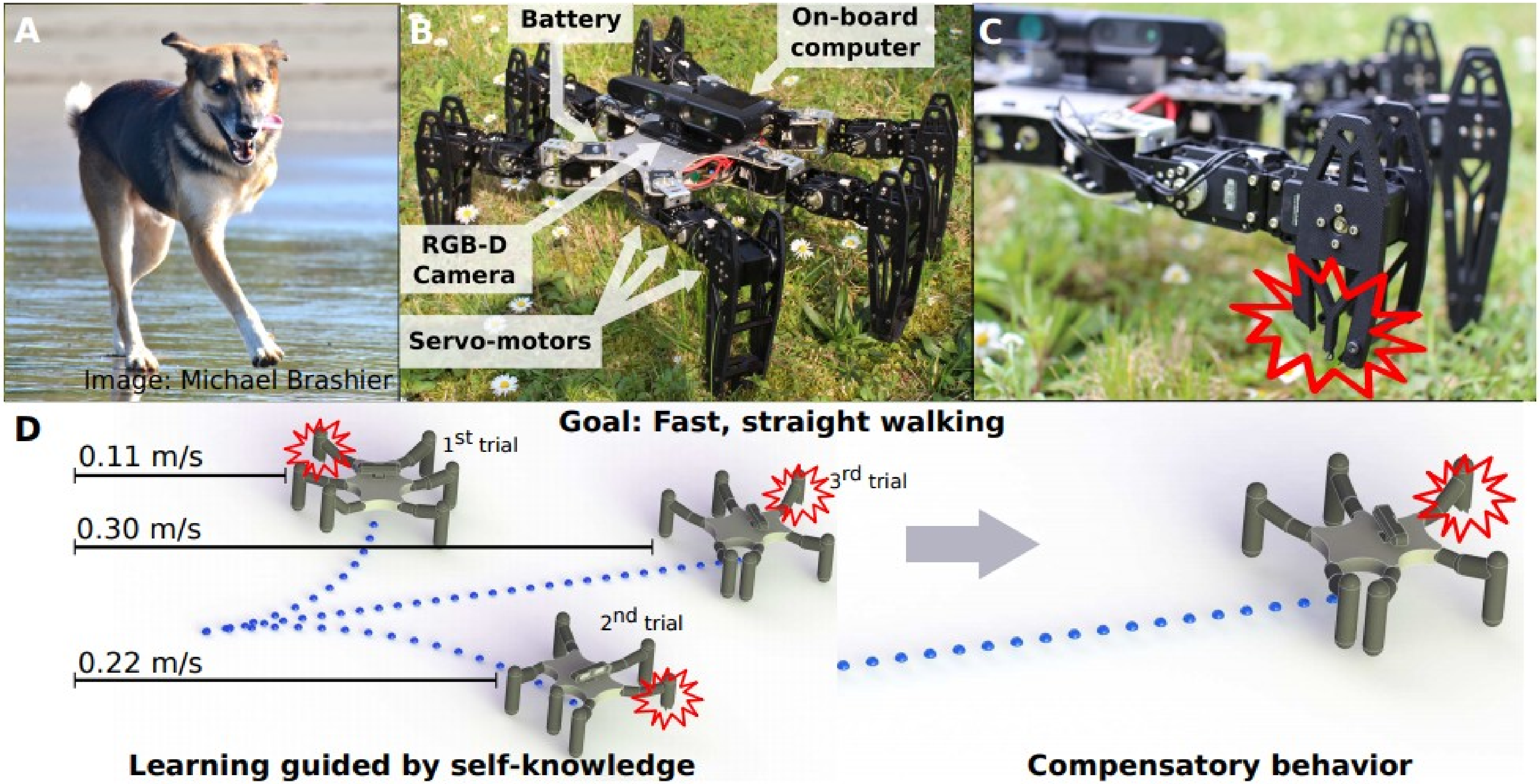
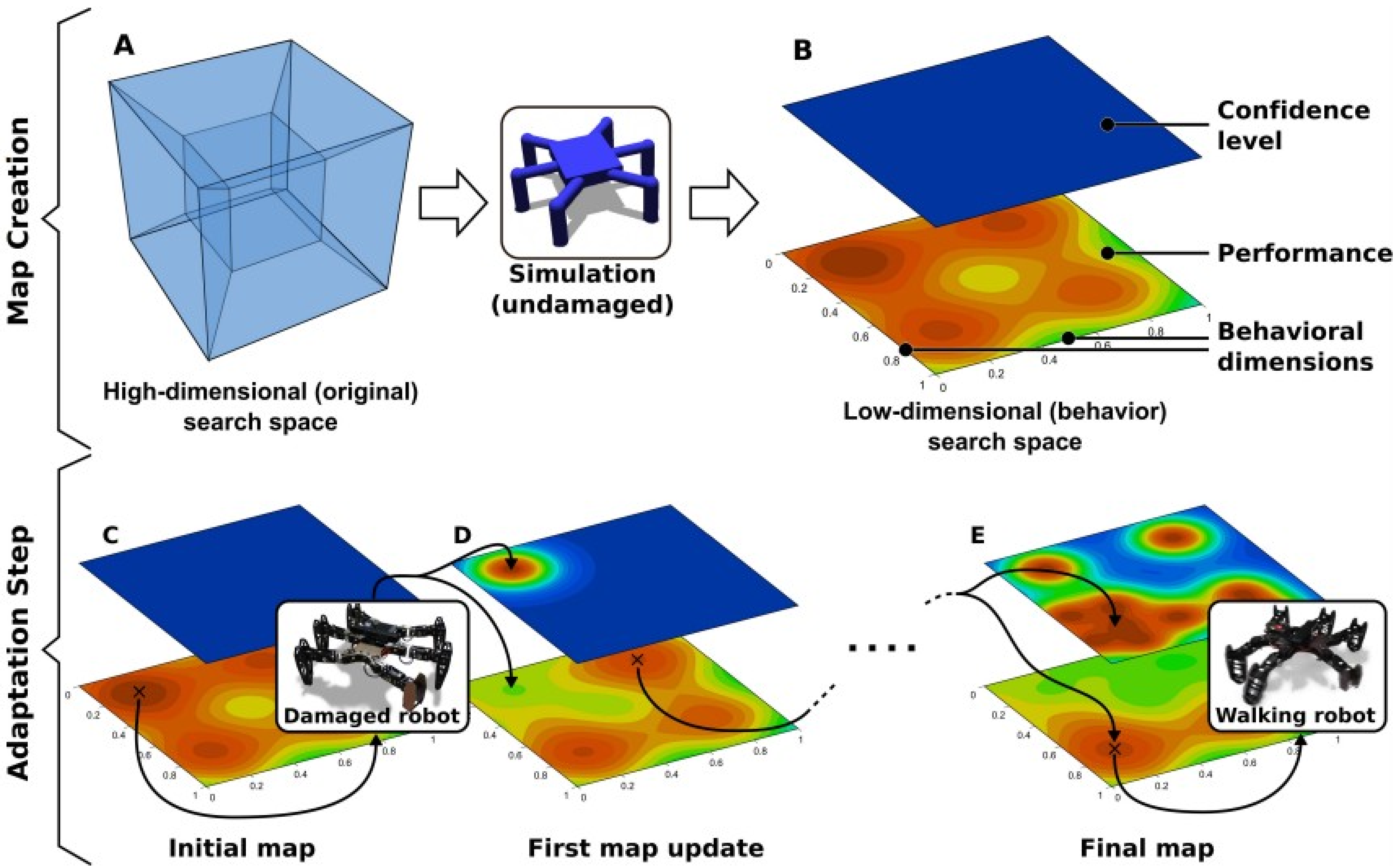
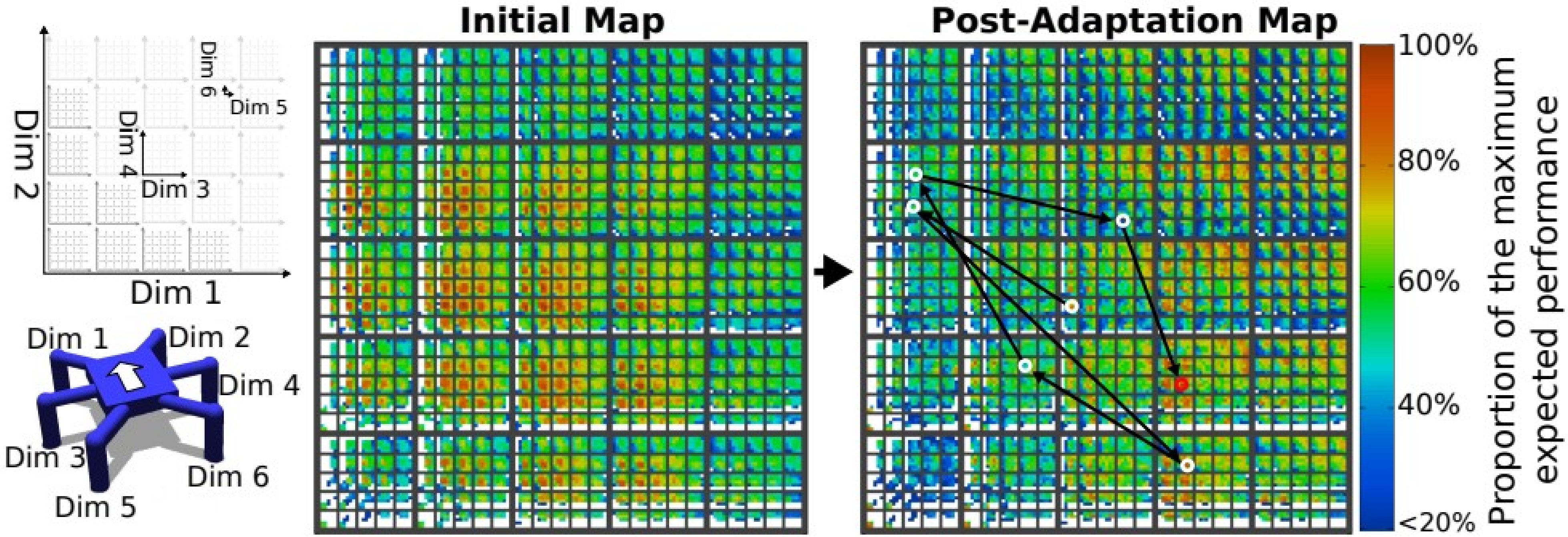


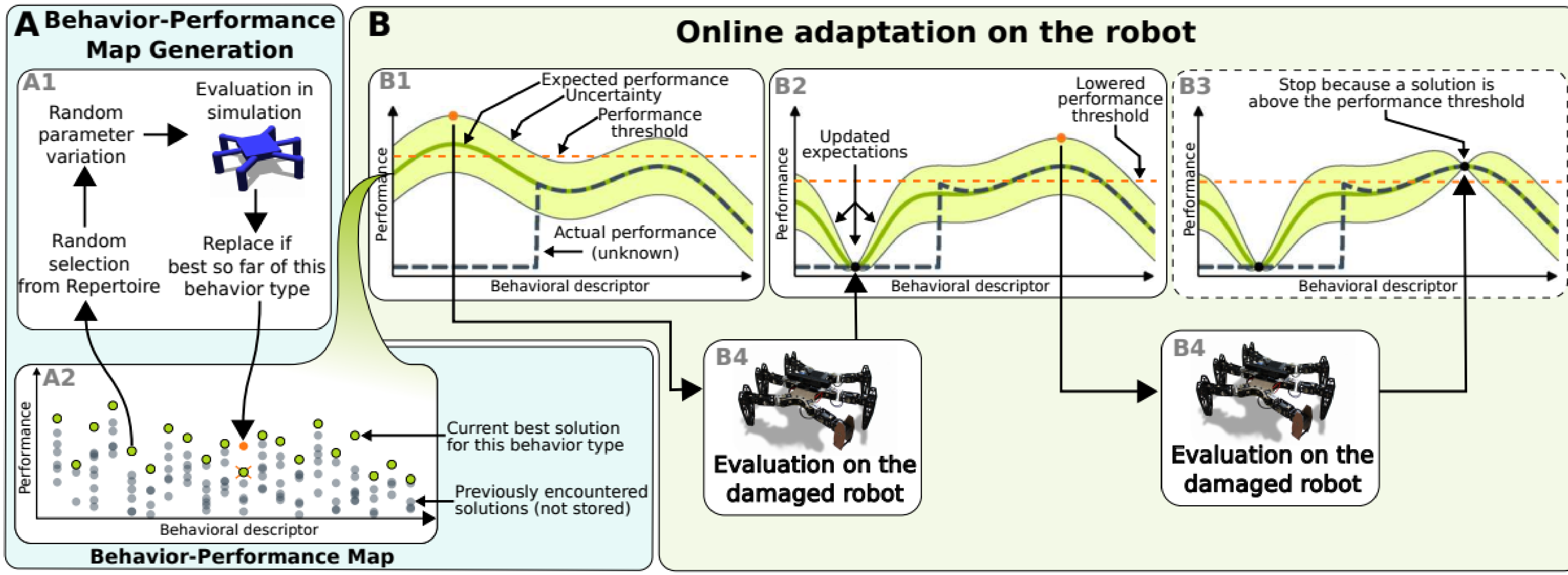
Figure 1 | With Intelligent Trial and Error, robots, like animals, can quickly adapt to recover from damage. **(A)** Most animals can find a compensatory behavior after an injury. Without relying on *predefined* compensatory behaviors, they learn how to avoid behaviors that are painful or no longer effective. **(B)** An undamaged, hexapod robot. **(C)** One type of damage the hexapod may have to cope with (broken leg). **(D)** After damage occurs, in this case making the robot unable to walk straight, damage recovery via Intelligent Trial and Error begins. The robot tests different types of behaviors from an automatically generated map of the behavior-performance space. After each test, the robot updates its predictions of which behaviors will perform well despite the damage. This way, the robot rapidly discovers an effective compensatory behavior.



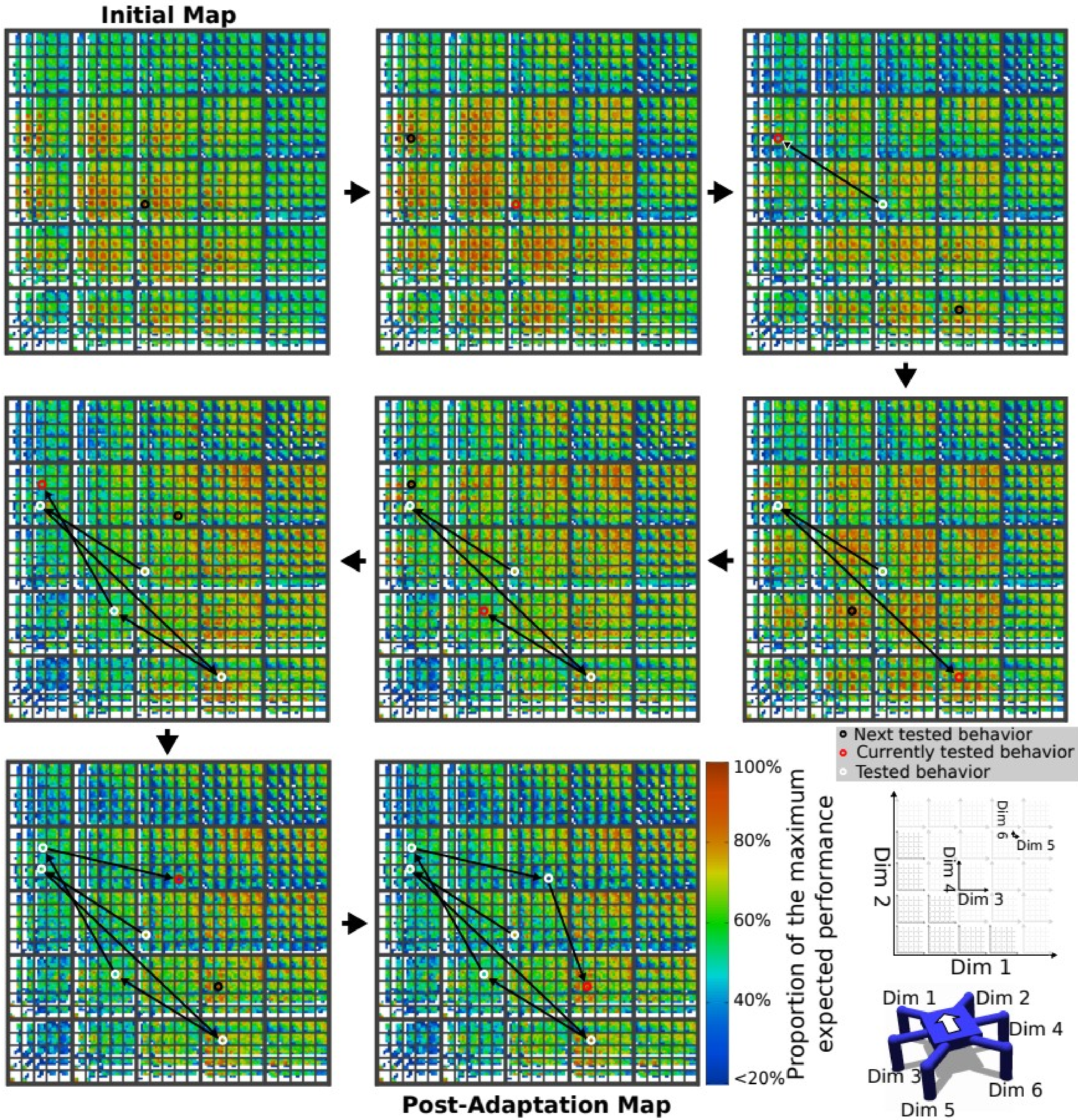
Map Creation
A
High-dimensional (original) search space
Simulation (undamaged)
B
Confidence level
Performance
Behavioral dimensions
Low-dimensional (behavior) search space
Adaptation Step
C
D
E
Damaged robot
Walking robot
Initial map
First map update
Final map



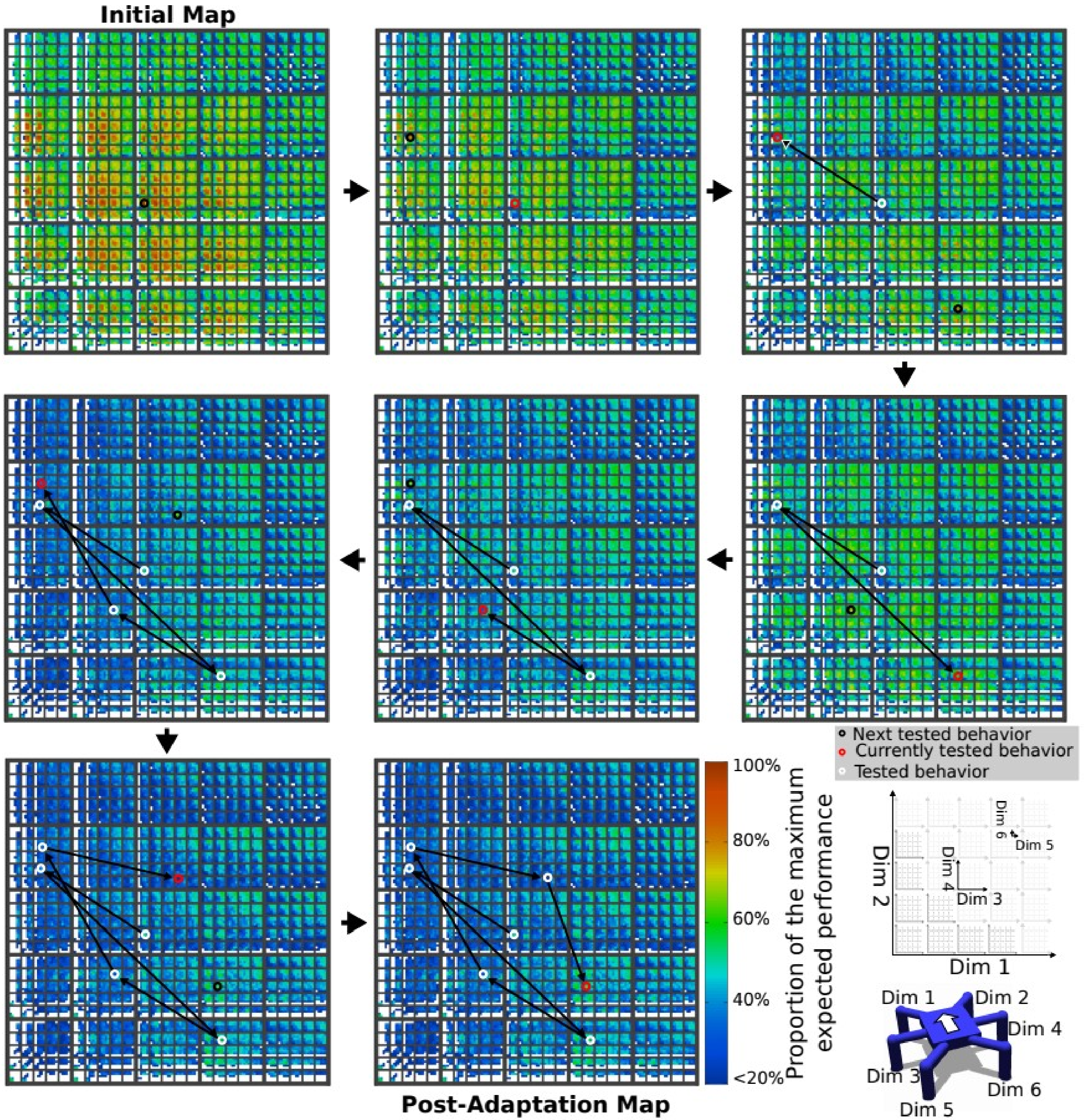
Initial Map
Post-Adaptation Map
Dim 1
Dim 2
Dim 3
Dim 4
Dim 5
Dim 6
Dim 1
Dim 2
Dim 3
Dim 4
Dim 5
Dim 6
100%
80%
60%
40%
<20%
Proportion of the maximum expected performance



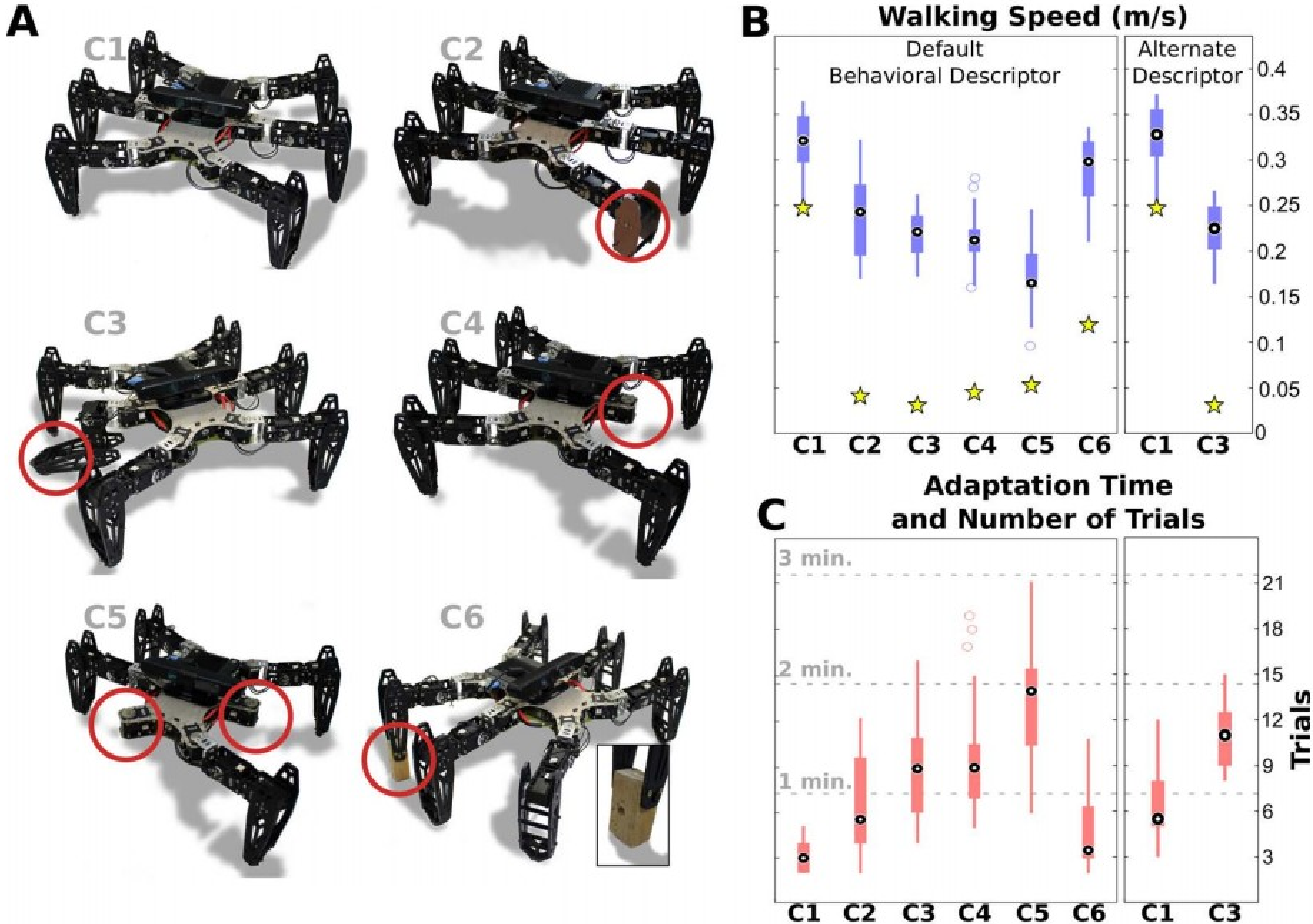
Extended Data Figure 1 | An overview of the Intelligent Trial and Error Algorithm. (A) Behavior-performance map creation. After being initialized with random controllers, the behavioral map (A2), which stores the highest-performing controller found so far of each behavior type, is improved by repeating the process depicted in (A1) until newly generated controllers are rarely good enough to be added to the map (here, after 40 million evaluations). This step, which occurs in simulation, is computationally expensive, but only needs to be performed once per robot (or robot design) prior to deployment. In our experiments, creating one map involved 40 million iterations of (A1), which lasted roughly two weeks on one multi-core computer (Supplementary Methods, section "Running time"). **(B) Adaptation.** (B1) Each behavior from the behavior-performance map has an expected performance based on its performance in simulation (dark green line) and an estimate of uncertainty regarding this predicted performance (light green band). The actual performance on the now-damaged robot (black dashed line) is unknown to the algorithm. A behavior is selected to try on the damaged robot. This selection is made by balancing exploitation—trying behaviors expected to perform well—and exploration—trying behaviors whose performance is uncertain (Methods, section "acquisition function"). Because all points initially have equal, maximal uncertainty, the first point chosen is that with the highest expected performance. Once this behavior is tested on the physical robot (B4), the performance predicted for that behavior is set to its actual performance, the uncertainty regarding that prediction is lowered, and the predictions for, and uncertainties about, nearby controllers are also updated (according to a Gaussian process model, see Methods, section "kernel function"), the results of which can be seen in (B2). The process is then repeated until performance on the damaged robot is 90% or greater of the maximum expected performance for any behavior (B3). This performance threshold (orange dashed line) lowers as the maximum expected performance (the highest point on the dark green line) is lowered, which occurs when physical tests on the robot underperform expectations, as occurred in (B2).

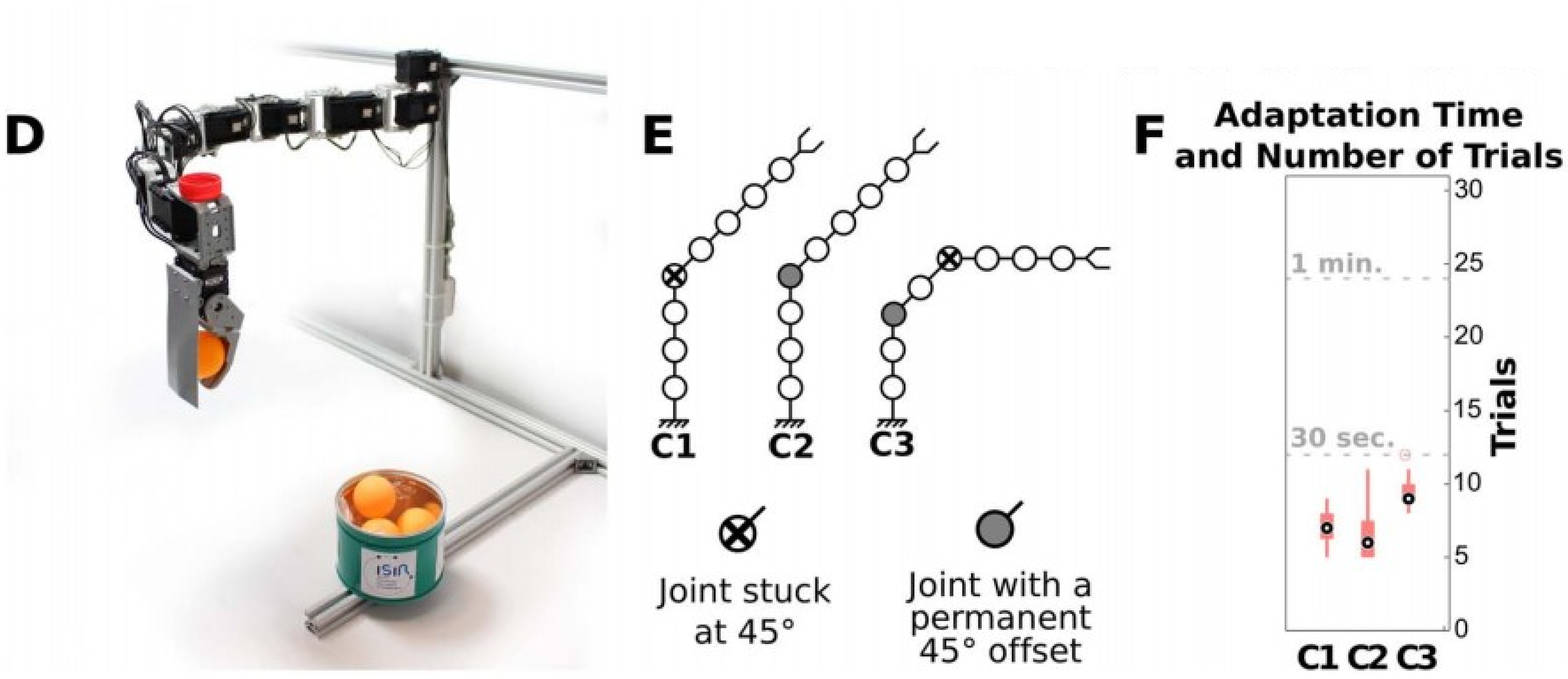


Initial Map
Next tested behavior
Currently tested behavior
Tested behavior
100%
80%
60%
40%
<20%
Proportion of the maximum expected performance
Dim 1
Dim 2
Dim 3
Dim 4
Dim 5
Dim 6
Post-Adaptation Map



Initial Map
Post-Adaptation Map
Next tested behavior
Currently tested behavior
Tested behavior
100%
80%
60%
40%
<20%
Proportion of the maximum expected performance
Dim 1
Dim 2
Dim 3
Dim 4
Dim 5
Dim 6


A
C1
C2
C3
C4
C5
C6
B
Walking Speed (m/s)
Default
Behavioral Descriptor
Alternate
Descriptor
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
C1 C2 C3 C4 C5 C6 C1 C3
C
Adaptation Time
and Number of Trials
3 min.
2 min.
1 min.
21
18
15
12
9
6
3
Trials
C1 C2 C3 C4 C5 C6 C1 C3



D
E
C1
C2
C3
Joint stuck at 45°
Joint with a permanent 45° offset
F
Adaptation Time and Number of Trials
1 min.
30 sec.
Trials
30
25
20
15
10
5
0
C1 C2 C3

Robots that can adapt like animals

Nature, 2015

which describes damage recovery via Intelligent Trial and Error

Antoine Cully
UPMC/CNRS
(France)

Jeff Clune
University of Wyoming
(USA)

Danesh Tarapore
UPMC/CNRS
(France)

Jean-Baptiste Mouret
UPMC/CNRS/Inria/UL
(France)

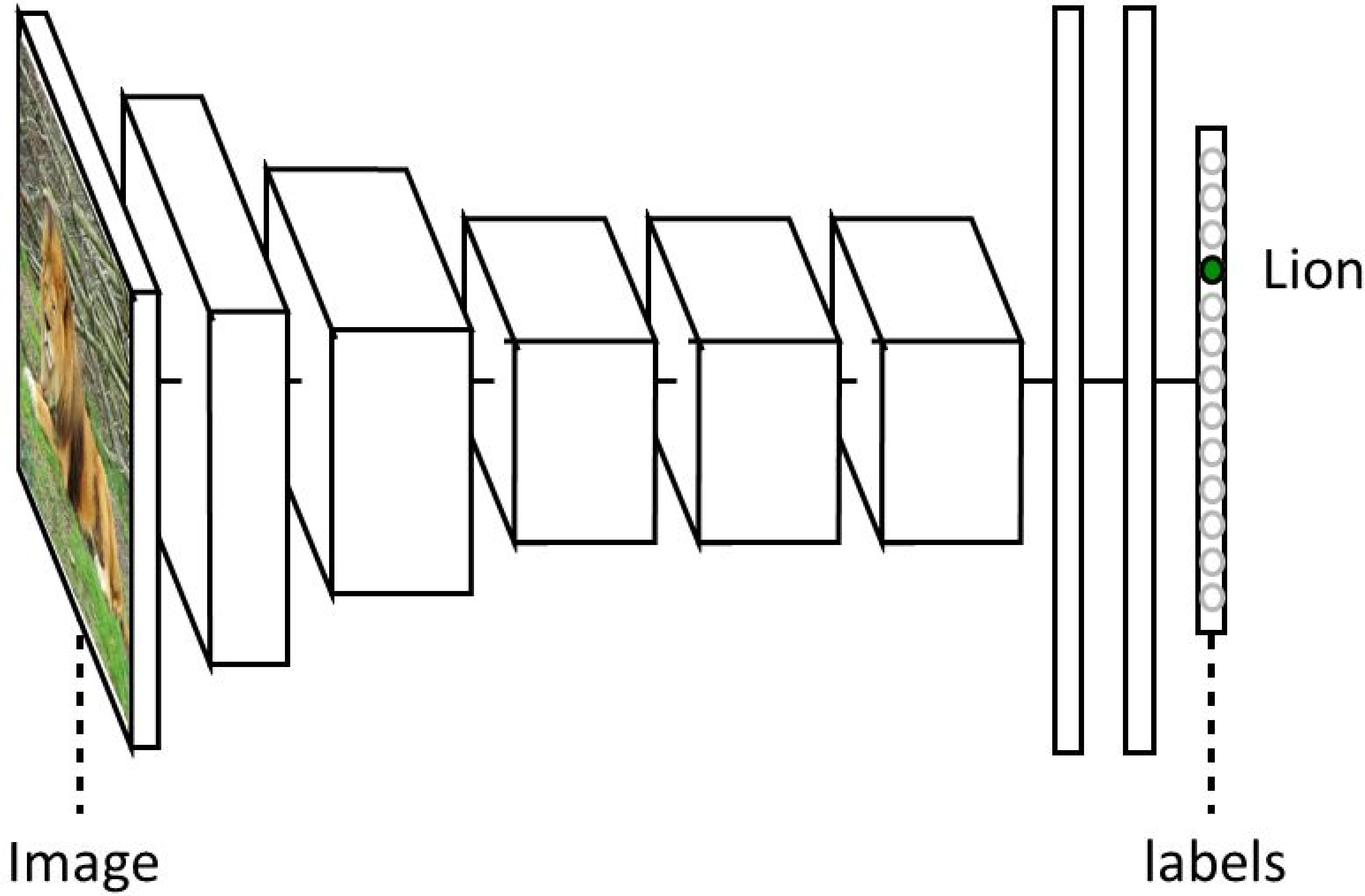
Innovation Engines: Automated Creativity and Improved Stochastic Optimization via Deep Learning

Anh Nguyen
University of Wyoming
anguyen8@uwyo.edu

Jason Yosinski
Cornell University
yosinski@cs.cornell.edu

Jeff Clune
University of Wyoming
jeffclune@uwyo.edu

Convolutional Neural Networks: AlexNet



Krizhevsky, Sutskever, Hinton — NIPS 2012

slide credit Jason Yosinski

IMAGENET

SEARCH

14,197,122 images, 21841 synsets indexed

Home | Explore
About | Download

Not logged in. Login | Signup

Structure, construction

A thing constructed; a complex entity constructed of many parts; "the structure consisted of a series of arches"; "she wore her hair in an amazing construction of whirls and ribbons"

1190 pictures

96.04% Popularity Percentile

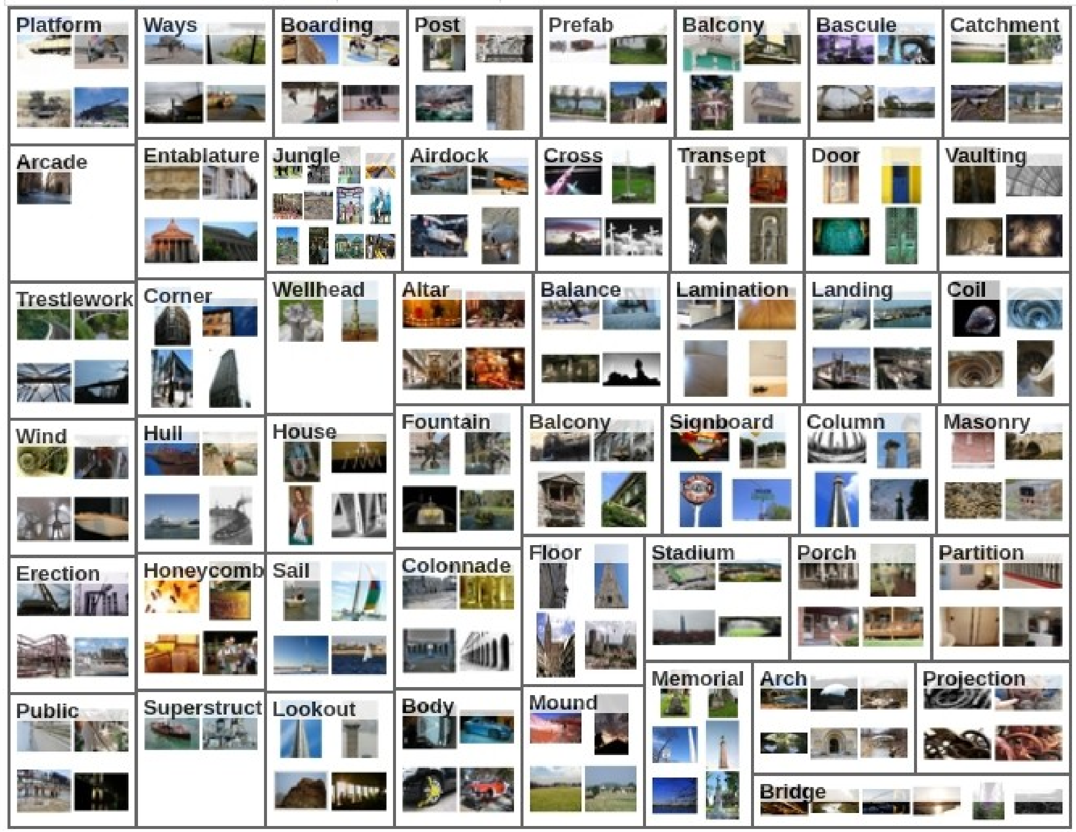
Wordnet IDs

Numbers in brackets: (the number of synsets in the subtree).

- ImageNet 2011 Fall Release (32326)
  - plant, flora, plant life (4486)
  - geological formation, formation (17
  - natural object (1112)
  - sport, athletics (176)
  - artifact, artefact (10504)
    - instrumentality, instrumentation
    - structure, construction (1405)
      - airdock, hangar, repair shed
      - altar (1)
      - arcade, colonnade (1)
      - arch (31)
      - area (344)
      - balcony (4)
      - balcony (2)
      - bascule (0)
      - boarding (2)
      - body (2)
      - bridge, span (17)
      - building, edifice (267)
      - building complex, complex (
      - catchment (0)
      - coil, spiral, volute, whorl, heli
      - colonnade (1)
      - column, pillar (2)
      - corner, quoin (0)
      - cross (0)
      - deathtrap (0)
      - defensive structure, defense
      - door (0)

Treemap Visualization | Images of the Synset | Downloads

ImageNet 2011 Fall Release 〉 Artifact, artefact 〉 Structure, construction



ImageNet Challenge

IMAGENET

- 1,000 object classes (categories).
- Images:
  - 1.2 M train
  - 100k test.

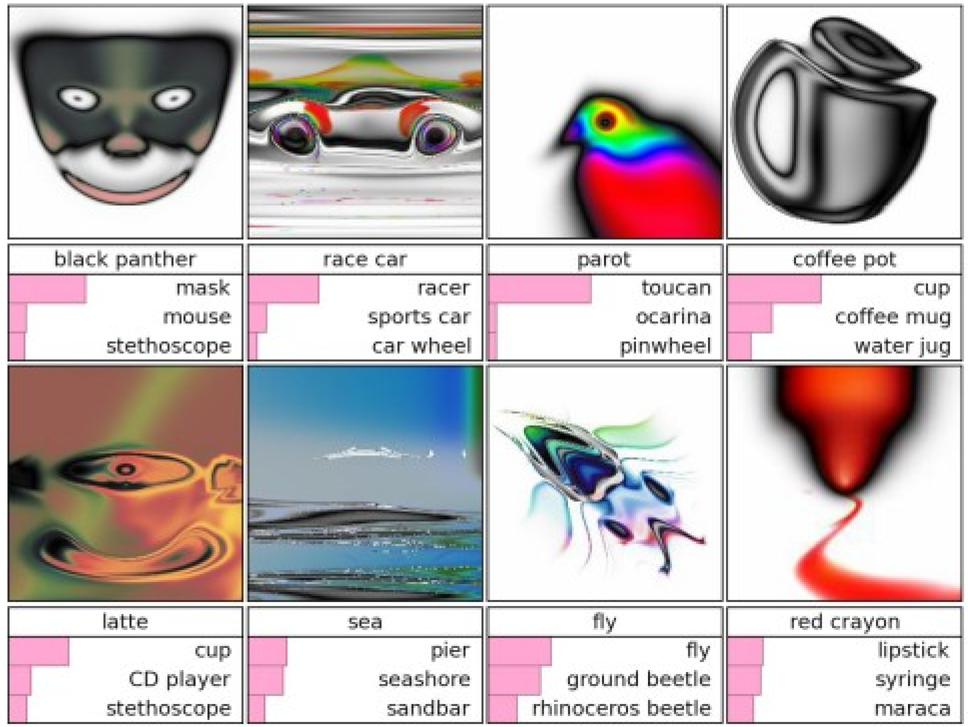


Figure 3: CPPN-encoded images evolved and named (centered text) by Picbreeder.org users. The DNN's top three classifications and associated confidence (size of the pink bar) are shown. The DNN's classifications often relate to the human breeder's label, showing that DNNs can recognize CPPN-encoded, evolved images. Adapted from [21].

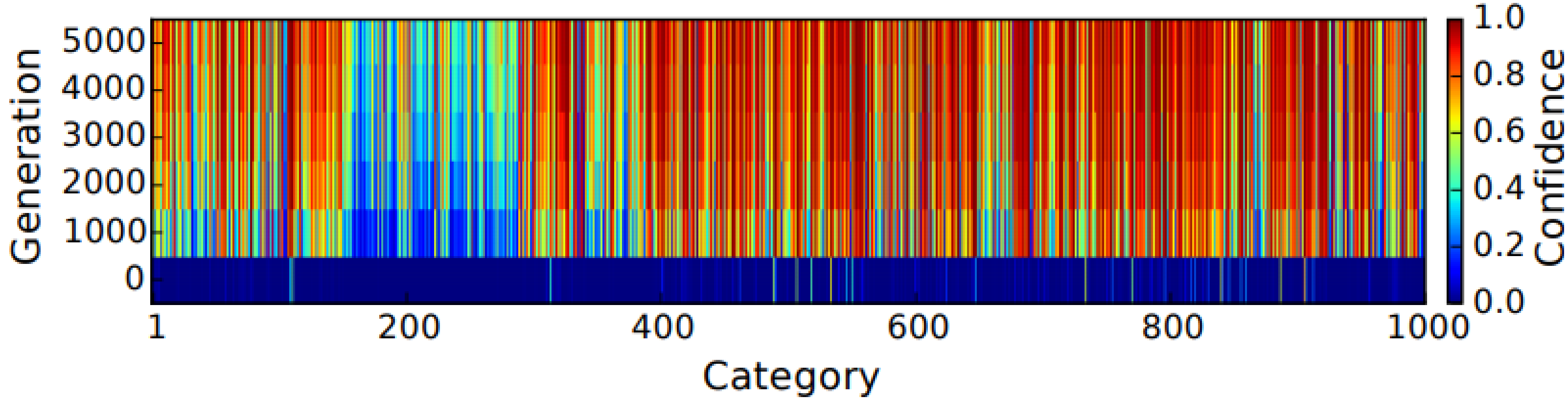


Figure 4: The MAP-Elites evolutionary algorithm produces images that the DNN declares with high confidence to belong to most ImageNet classes. Colors represent median confidence scores from 10 runs.

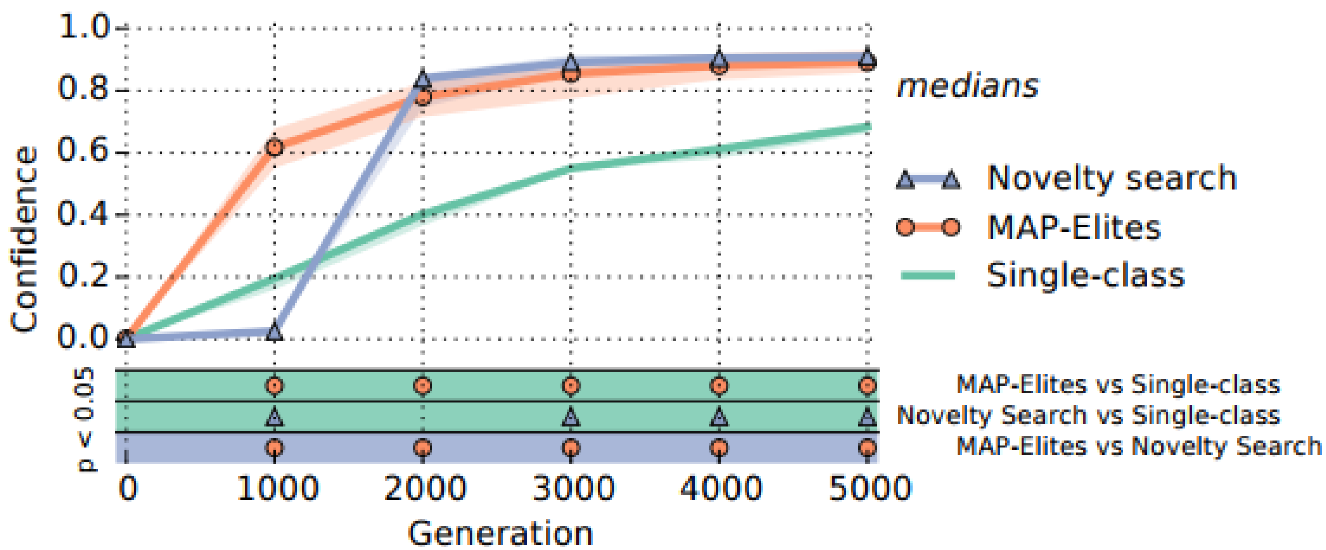


Figure 8: Innovation Engines built with MAP-Elites or Novelty Search perform similarly to each other, and both significantly outperform a single-class evolutionary algorithm. Solid lines show median performance and shaded areas indicate the 95% bootstrapped confidence interval of the median. The bottom three rows show statistical significance.

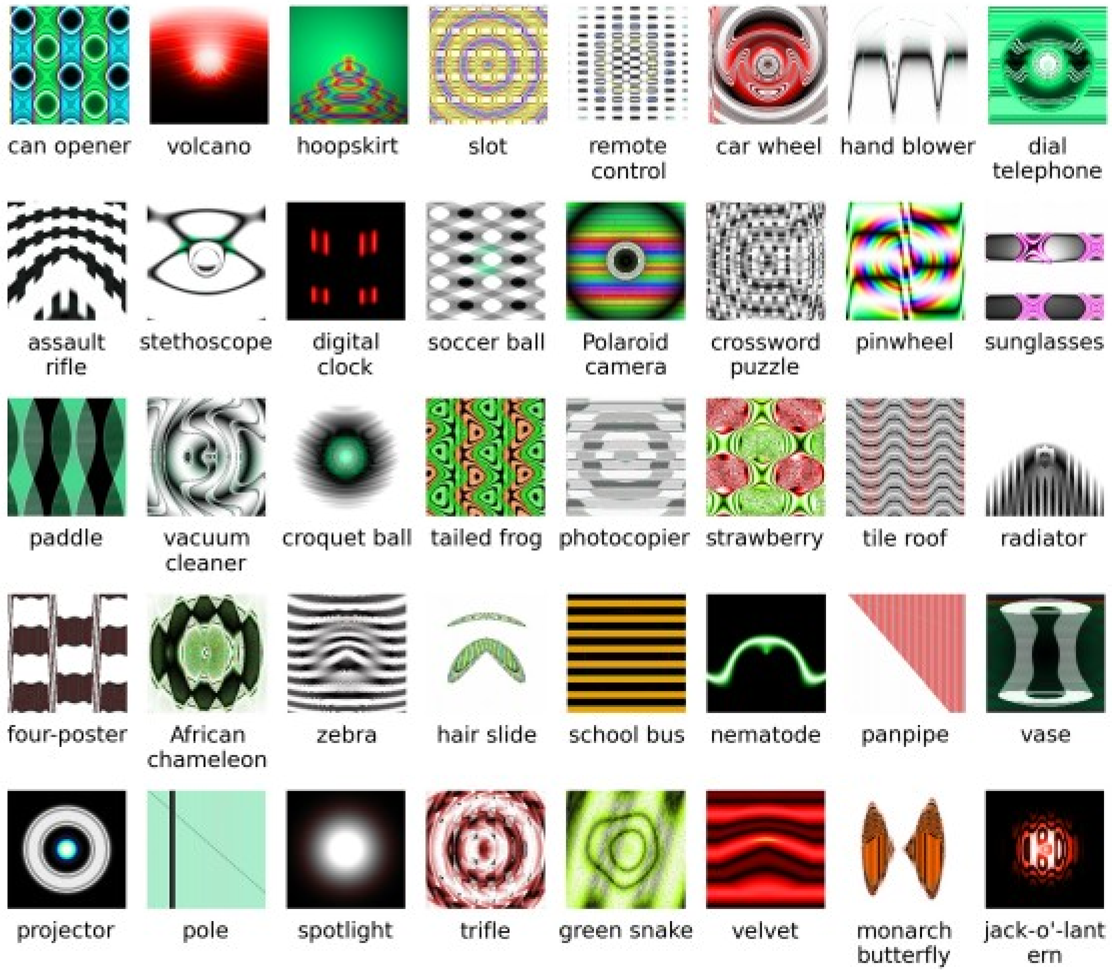


Figure 5: Innovation Engines in the image domain generate a tremendous diversity of interesting images. Shown are images selected to showcase diversity from 10 evolutionary runs. The diversity results from the pressure to match 1000 different ImageNet classes. In this and subsequent figures, the DNN's top label for each evolved image is shown below it.

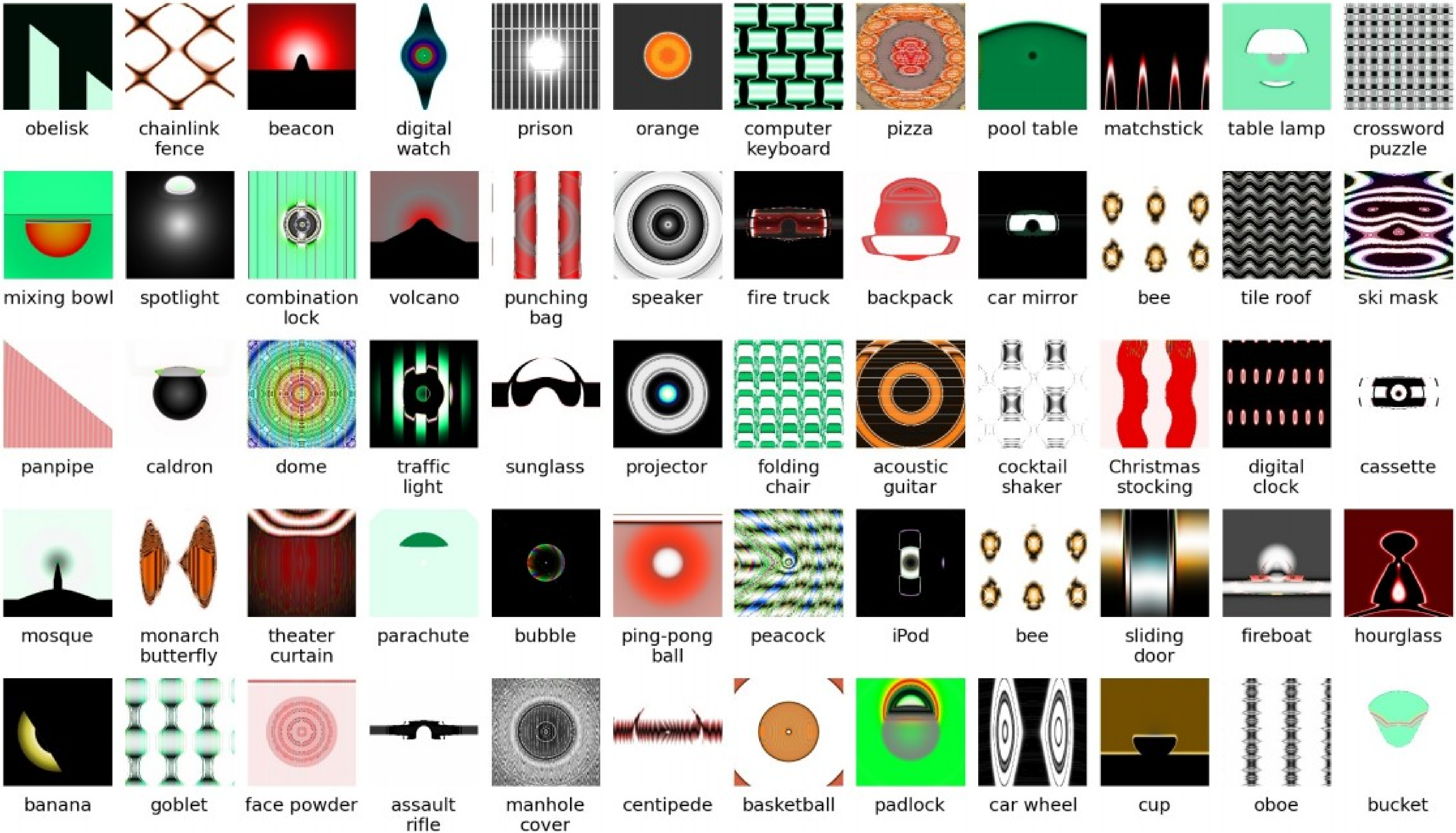
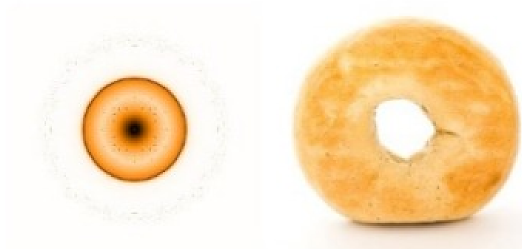


Figure 7: Innovation Engines are capable of producing images that are not only given high confidence scores by a deep neural network, but are also qualitatively interesting and recognizable. To show the most interesting images we observed evolve, we selected images from both the 10 main experiment runs and 10 preliminary experiments with slightly different parameters.

Matchstick

Television

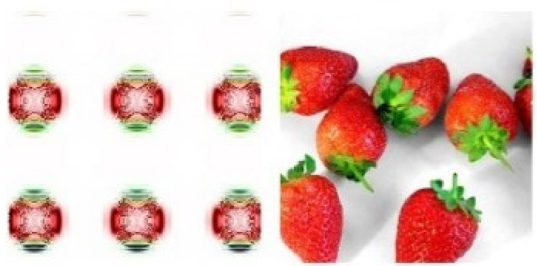


Bagel

Prison

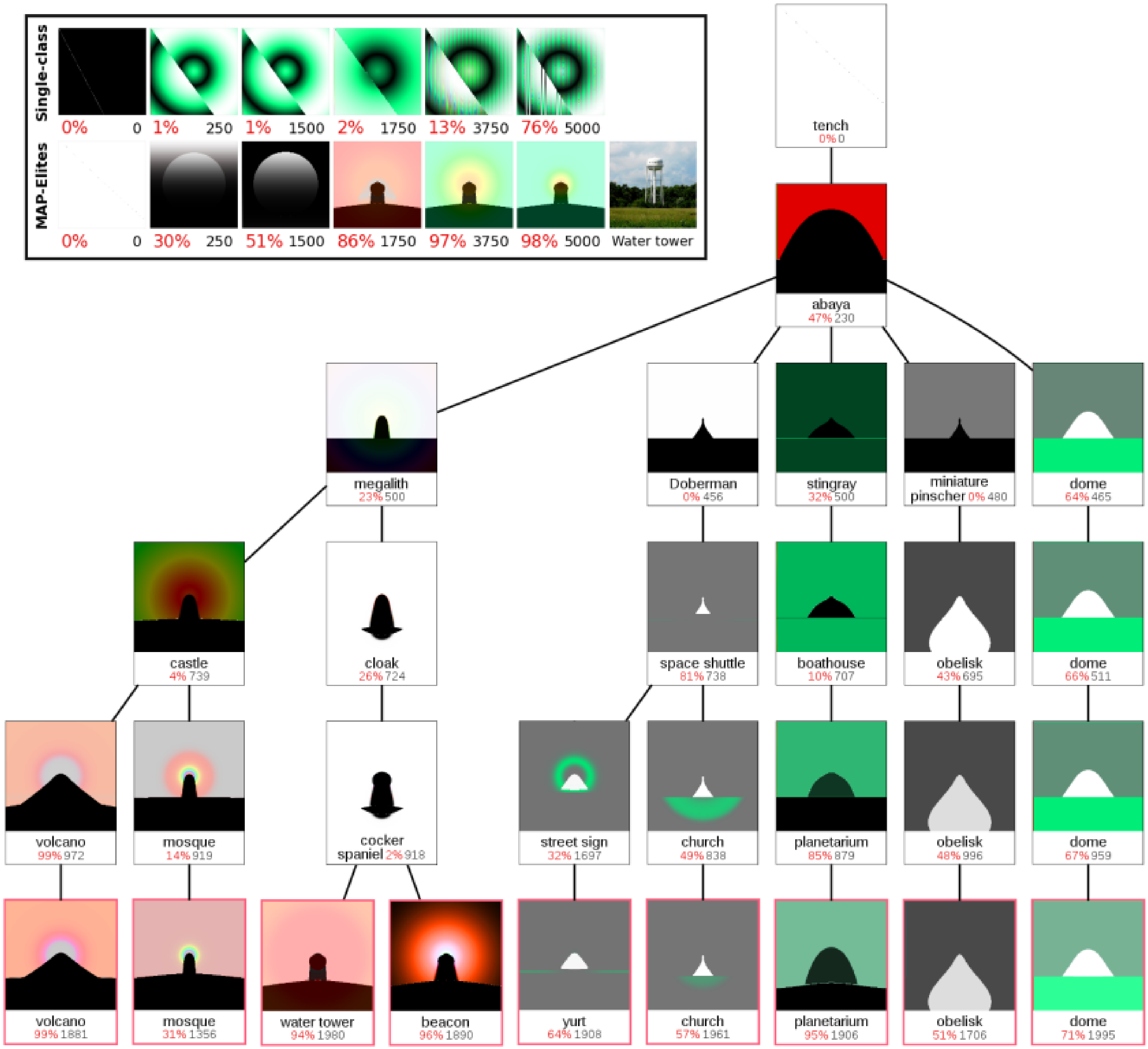
Chainlink fence

Tile roof



Strawberry

Sunglasses


Single-class
0% 0
1% 250
1% 1500
2% 1750
13% 3750
76% 5000
MAP-Elites
0% 0
30% 250
51% 1500
86% 1750
97% 3750
98% 5000
Water tower
tench 0% 0
abaya 47% 230
megalith 23% 500
Doberman 0% 456
stingray 32% 500
miniature pinscher 0% 480
dome 64% 465
castle 4% 739
cloak 26% 724
space shuttle 81% 738
boathouse 10% 707
obelisk 43% 695
dome 66% 511
volcano 99% 972
mosque 14% 919
cocker spaniel 2% 918
street sign 32% 1697
church 49% 838
planetarium 85% 879
obelisk 48% 996
dome 67% 959
volcano 99% 1881
mosque 31% 1356
water tower 94% 1980
beacon 96% 1890
yurt 64% 1908
church 57% 1961
planetarium 95% 1906
obelisk 51% 1706
dome 71% 1995