On the Entanglement between Evolvability and Fitness: an Experimental Study on Voxel-based Soft Robots

Andrea Ferigo[1], L. B. Soros[2], Eric Medvet[3], and Giovanni Iacca[1]

[1]Department of Information Engineering and Computer Science, University of Trento, Italy
[2]Cross Labs, Cross Compass Ltd., Kyoto, Japan
[3]Department of Engineering and Architecture, University of Trieste, Italy
andrea.ferigo@unitn.it, lisa.soros@cross-compass.com, emedvet@units.it, giovanni.iacca@unitn.it

Abstract

The concept of evolvability, that is the capacity to produce heritable and adaptive phenotypic variation, is crucial in the current understanding of evolution. However, while its meaning is intuitive, there is no consensus on how to quantitatively measure it. As a consequence, in evolutionary robotics, it is hard to evaluate the interplay between evolvability and fitness and its dependency on key factors like the evolutionary algorithm (EA) or the representation of the individuals. Here, we propose to use MAP-Elites, a well-established Quality Diversity EA, as a support structure for measuring evolvability and for highlighting its interplay with fitness. We map the solutions generated during the evolutionary process to a MAP-Elites-like grid and then visualize their fitness and evolvability as maps. This procedures does not affect the EA execution and can hence be applied to any EA: it only requires to have two descriptors for the solutions that can be used to meaningfully characterize them. We apply this general methodology to the case of Voxel-based Soft Robots (VSR), a kind of modular robots with a body composed of uniform elements whose volume is individually varied by the robot brain. Namely, we optimize the robots for the task of locomotion using evolutionary computation. We consider four representations, i.e., ways of transforming a genotype into a robot, two for the brain only and two for both body and brain of the VSR, and two EAs (MAP-Elites and a simple evolutionary strategy) and examine the evolvability and fitness maps. The experiments suggest that our methodology permits us to discover interesting patterns in the maps: fitness maps appear to depend more on the representation of the solution, whereas evolvability maps appear to depend more on the EA. As an aside, we find that MAP-Elites is particularly effective in the simultaneous evolution of the body and the brain of Voxel-based Soft Robots.

1 Introduction

Evolution in nature has created a diversity of viable ways of living occupying vastly different niches (plants, mammals, etc.). Yet, despite the rich diversity observed in nature, much of this evolution depends on variation on common ancestors. For example, all breeds of domesticated dog (Canis familiaris), from poodles to Great Danes, are hypothesized to have descended from the grey wolf (Canis lupus) (Coppinger and Smith, 1983) or some other wild canid (Koler-Matznick, 2002)[1].

What is it about these common ancestors that makes them so likely to further evolve into high-quality descendants very distinct from themselves? Capturing this property, called *evolvability*, and replicating it inside an algorithmic process is a challenge for advancing our theoretical understanding of natural and artificial evolutionary systems. It is also likely that figuring out how to discover highly evolvable individuals will have positive practical results for the purposes of optimization by virtue of avoiding premature convergence (Squillero and Tonda, 2016).

This paper explores the concept of evolvability in the latter sense (as a tool for achieving high performance in an engineering context). In particular, we use two fundamentally different Evolutionary Algorithms (EAs): a simple form of Evolution Strategies (ES) (Beyer and Schwefel, 2002), and the Quality Diversity algorithm MAP-Elites (Cully et al., 2015). We apply them to discover highly evolvable morphologies and controllers for Voxel-based Soft Robots (VSRs). We choose these EAs as they represent two different ways to conduct evolutionary search: one mainly aimed at exploitation (ES), the other one mainly aimed at exploration (MAP-Elites). On the other hand, we choose VSRs for this study mainly due to their modularity, which makes them particular expressive and suitable for constituting an autonomous robotic ecosystem (e.g., for space exploration applications (Methenitis et al., 2015)) in which evolvability would be a key feature.

We review the concept of evolvability with a focus on its applications in artificial evolutionary systems. We introduce a method for keeping track of the evolvability and fitness of solutions generated during the execution of potentially any EA. We use a grid-like structure for storing the most relevant solutions in a way that enables a convenient visualization of their fitness and evolvability at the end of the

[1] Interestingly, when Darwin wrote about evolution, he concluded that the diversity of dogs necessarily must result from interbreeding of many kinds of wild dogs (Darwin, 1875). However, molecular dating techniques in recent years suggest that this conclusion is likely wrong (Wayne et al., 1991).

evolutionary process: this methodology only requires two descriptors for characterizing the solutions and does not interfere with the EA execution. Since fitness and evolvability of individual solutions are placed in a grid-like structure, the analysis of their interplay is facilitated. We apply this general methodology to the case of the optimization of VSRs (brain only and both body and brain) for the task of locomotion. We consider four different representations and combine them with the two EAs mentioned above and apply our methodology to discover insights about how the EA and the representation impact on the interplay between fitness and evolvability.

Results indicate that both the choices of the representation and the EA have a major impact not only on the quality of the solution, but also on how the search space is explored, which ultimately reflects in different observations of evolvability. More precisely, we observe that the fitness distribution in a phenotypic space determined by predefined descriptors (more on this below) is mostly determined by the adopted representation; on the contrary, the evolvability distribution over the same phenotypic space appears to be determined by the EA. Hence we conclude that fitness and evolvability are somehow "entangled" and that this entanglement is affected by multiple aspects of evolutionary systems.

The rest of the paper is structured as follows. In the next section, we introduce the background concepts and briefly summarize the related works. In Section 3, we describe the methods. Then, we present the results in Section 4. Finally, we give the conclusions in Section 5.

2 Background and related works

Evolvability is essentially a measure of potential for evolutionary innovation. However, measures of evolvability differ in what features for evolutionary innovation are salient (Pigliucci, 2008). Per one popular definition, "Evolvability is the ability of a biological system to produce phenotypic variation that is both heritable and adaptive" (Payne and Wagner, 2019; Nordmoen et al., 2021). It is important to note two distinct components of this definition: that there is variation (i.e., diversity) being passed from parent to offspring, and that this variation leads to positive effects on fitness. Interestingly and importantly, measures and studies from artificial life (a primary domain of interest for evolvability studies related to artificial evolution) regard evolvability purely as adaptation (Medvet et al., 2017; Veenstra et al., 2020; Liu et al., 2022; Tarapore and Mouret, 2015), or evolvability as diversification (Mengistu et al., 2016; Gajewski et al., 2019; Lehman and Stanley, 2011b, 2013; Lim et al., 2021; Carlo et al., 2021), but not both.

Searching directly for evolvability has become a recently popular trend. In Evolvability Search (Mengistu et al., 2016), the fitness function of a traditional EA rewards high evolvability (in this diversity-oriented interpretation, it is the number of distinct behaviors in the set of offspring generated by an individual) instead of rewarding maximizing a domain-specific objective. This algorithm is shown to outperform both greedy optimization and novelty search (Lehman and Stanley, 2011a). The subsequent Evolvability Evolution Strategy (E-ES) (Gajewski et al., 2019) introduces improvements in terms of computational expense to scale to deep neural networks. Quality Evolvability ES (QE-ES) (Katona et al., 2021) builds on both of these algorithms, simultaneously optimizing for both evolvability (as diversity) and fitness with non-dominated sorting, as in NSGA-II (Deb et al., 2002). Note that unlike Quality Diversity algorithms (Pugh et al., 2016), which seek to discover a diverse population of high-performing individuals, the goal in Quality Evolvability is to discover a single individual with diverse offspring. The work reported in this current paper will leverage the Quality Diversity algorithm MAP-Elites (see Section 3) to find diverse populations of highly evolvable individuals.

3 Methods

In the following, we present the main methods and tools we used in our study: the VSRs, the EAs used for evolving them, the descriptors adopted for characterizing evolved VSRs, and the evolvability metric.

Voxel-based Soft Robots

First presented by Hiller and Lipson (2011), VSRs are a type of modular soft robot composed of cubic elements that can vary their volume according to a control signal that is generated by a controller. In this work, we use the 2D (yet, physically plausible) VSR simulator presented by Medvet et al. (2020a). In fact, using a two-dimensional model allows to reduce the numerical complexity of the robot simulations, without losing the potential variety of robot shapes and behaviors.

From a conceptual level, a VSRs can be seen as a composition of two components: the *body*, i.e., a set of voxels arranged in a given shape, and the *brain*, i.e., a controller that produces the control signal for each voxel in the body. The details of these two components are reported below.

Body. The body of a VSR is defined by a number of deformable squares, called *voxels*, arranged (in our case) in a 2D grid (Figure 1). Pairs of adjacent voxels are glued together at their two common vertexes. During the simulation, each voxel changes its area as a result of the combined action of (a) external forces imposed by bodies in contact with the voxel, namely other voxels and the ground, and (b) an internal force that makes the voxel expand or contract. The internal force at time t is determined by a *control value* $c(t) \in [-1, 1]$, where -1 means maximum area expansion and 1 means maximum area contraction. The control value is itself determined, for each voxel of the body, by the brain of the VSR, described in the next section.

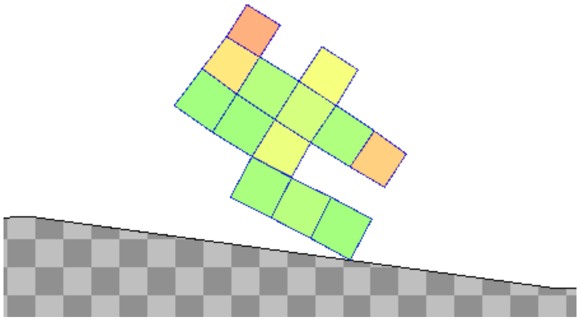


Figure 1: An example VSR with a body enclosed in a 4×5 grid; namely, one of those obtained in our experiments (see Section 4 and Figure 5a). The color of each voxel depends on its current area: green for expanded, red for contracted, yellow for equal to the rest area.

We rely on 2D-VSR-Sim (Medvet et al., 2020a) for simulating the VSR body. In brief, in 2D-VSR-Sim voxels are modeled as an assembly of four masses (at the vertexes of the square) and multiple spring-damper systems connecting them. The internal force is modeled as an instantaneous change of the resting length of the spring-damper systems.

Clearly, the body of a VSR, together with its brain, plays a crucial role in making the robot more or less effective for a given task. It follows that the body, i.e., how to organize voxels in the 2D grid, can be optimized.

Brain. The brain generates the control signal for each voxel and hence acts as the controller for the robot. In this work, we use an open-loop controller where the control signal depends only on the current time t. Namely, we use a sinusoidal controller for which $c(t) = \sin(2\pi f t + \phi)$. Despite appearing trivial, this form of open-loop controller has been used for VSRs to achieve effective behaviors in different environments (Corucci et al., 2018) as well as for investigating complex adaptation dynamics (Kriegman et al., 2018).

The values of the frequency f and phase ϕ can be different among the voxels in the body. All together, they constitute the parameters of the controller and may be optimized for obtaining a desired behavior.

Evolutionary Algorithms

To optimize VSRs, we use two different EAs: a traditional fitness-driven EA, namely a simple form of Evolution Strategies (ES) (Beyer and Schwefel, 2002), and a Quality Diversity algorithm, namely MAP-Elites (ME) (Cully et al., 2015).

We decide to employ these two EAs because they support diversity, a substrate for evolvability, in radically different ways. While in ES the entire offspring is generated from a prototype individual deriving from a small subset of the best parents, in ME the offspring is generated by randomly sampling parents from the entire population, such that worse, but diverse parents can reproduce too.

Next, we briefly summarize the salient elements of the two algorithms. We denote by G the search space, i.e., the space where individuals are defined. In this study, we have $G = \mathbb{R}^p$, hence we denote individuals as numerical vectors $\boldsymbol{g} \in \mathbb{R}^p$; we remark that some of the components, namely, MAP-Elites, are defined for more general search spaces. We also assume that the fitness of an individual, i.e., its quality $f(\boldsymbol{g})$, can be evaluated as a single numerical value, i.e., with a fitness function $f : G \to \mathbb{R}$, for which it holds that the greater, the better.

Evolution Strategies (ES). In our simple form of ES, we iteratively evolve a fixed-size population of numerical vectors as individuals, i.e., $G = \mathbb{R}^p$, where p is the number of parameters to optimize.

Initially, we build the population by randomly generating n_pop individuals: namely, we build each individual $\boldsymbol{g}$ by sampling from the uniform distribution in $[-1, 1]$, i.e., $\boldsymbol{g} = (g_1, \dots, g_p)$ and $g_i \sim U(-1, 1)$. Then, we iterate the following steps (generations) until we have done n_eval fitness evaluations. First, we select the best $\frac{n_\text{pop}}{4}$ individuals as parents, i.e., those with the greatest fitness in the current population, and compute their element-wise mean $\boldsymbol{\mu} \in \mathbb{R}^p$. Then, we build $n_\text{pop} - 1$ offspring individuals, each one by adding to each element of $\boldsymbol{\mu}$ a Gaussian noise $\sim N(0, \sigma^2)$. Finally, we build the next population by taking the offspring and the best parent, i.e., we employ a form of elitism.

At the end of the evolution, ES outputs a single best solution being the individual with the largest fitness in the population at the last iteration of the algorithm.

MAP-Elites (ME). Multidimensional Archive of Phenotypic Elites (commonly known as MAP-Elites, here further abbreviated as ME), was originally introduced by Cully et al. (2015) for evolving robust behaviors in robots.

A key requirement in ME is the availability of some numerical *descriptors* of the solutions: the descriptors should be good at characterizing the solutions with respect to the problem being tackled, but, ideally, they should be orthogonal with respect to the fitness. Formally, we denote by $d : G \to \mathbb{R}^m$ the function for computing the descriptors $d(\boldsymbol{g}) = (d_1(\boldsymbol{g}), \dots, d_m(\boldsymbol{g}))$ of an individual $\boldsymbol{g}$. We assume that each descriptor $d_i(\boldsymbol{g})$ is defined in a bounded interval $D_i = [d_{i,\text{min}}, d_{i,\text{max}}]$. Given a number n_bin of bins, each individual can be mapped to the cell of an m-dimensional grid by considering, for each descriptor d_i, the index of the equal width bin of D_i in which the descriptor value falls. That is, we define the function $\boldsymbol{c} : G \to \mathbb{N}^m$ as $\boldsymbol{c}(\boldsymbol{g}) = (c_1(\boldsymbol{g}), \dots, c_m(\boldsymbol{g}))$ where $c_i(\boldsymbol{g}) = k \in \mathbb{N}$ such that $d_{i,\text{min}} + k\frac{|D_i|}{n_\text{bin}} \le d_i(\boldsymbol{g}) < d_{i,\text{min}} + (k+1)\frac{|D_i|}{n_\text{bin}}$ with $|D_i| = d_{i,\text{max}} - d_{i,\text{min}}$. We say that $\boldsymbol{c}(\boldsymbol{g})$ are the coordinates of $\boldsymbol{g}$ in the descriptor grid.

Differently from ES, ME does not evolve a fixed-size population of individuals: the population, called here *archive*, can increase in size during the evolution, up to $m^{n_{\text{bin}}}$ individuals, yet never decreases. At the beginning of the evolution, we populate the initially empty archive A by repeating the following steps n_{init} times: first, we randomly generate a new individual $\boldsymbol{g}$, then, we add $\boldsymbol{g}$ to A if no other individual $\boldsymbol{g}'$ exists in A at the same coordinates, i.e., such that $\boldsymbol{c}(\boldsymbol{g}') = \boldsymbol{c}(\boldsymbol{g})$, or, otherwise, if such $\boldsymbol{g}'$ exists and $f(\boldsymbol{g}) \geq f(\boldsymbol{g}')$. In the latter case, we remove $\boldsymbol{g}'$ from A.

After the initialization, we iterate the following steps (generations) until we have done n_{eval} fitness evaluations. We select n_{parent} individuals from A with uniform probability as parents. For each parent $\boldsymbol{g}$, we apply a genetic operator (mutation) $o : G \to G$ and obtain a child $\boldsymbol{g}' = o(\boldsymbol{g})$. Then we add $\boldsymbol{g}'$ to A as in the initialization procedure, i.e., if no other individual $\boldsymbol{g}''$ exists in A such that $\boldsymbol{c}(\boldsymbol{g}'') = \boldsymbol{c}(\boldsymbol{g}')$ or, otherwise, if such $\boldsymbol{g}''$ exists and $f(\boldsymbol{g}') \geq f(\boldsymbol{g}'')$. In the latter case, we remove $\boldsymbol{g}''$ from A.

At the end of the process, the algorithm does not return a single best individual, but the entire archive A (from which, in principle, one may choose as single best solution the one with the highest fitness value).

Since in this study we work with numerical vectors as individuals, also in ME we use as mutation the Gaussian mutation that we adopt in ES, i.e., $o(\boldsymbol{g}) = \boldsymbol{g} + \boldsymbol{\alpha}$, where $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_p)$ and $\alpha_i \sim N(0, \sigma^2)$. Accordingly, we build each of the n_{init} initial individuals by sampling the uniform distribution in $[-1, 1]$, as in ES.

Evolving VSRs with ES and ME

We want to evolve VSRs for the task of locomotion, i.e., moving along a surface as fast as possible, using the EAs described above. For this purpose, we need to define the solution representation, i.e., how to map a numerical vector $\boldsymbol{g} \in \mathbb{R}^p$ to a VSR, and the fitness function that quantifies the degree to which a VSR is doing locomotion. Moreover, for ME we also need to define the descriptors, i.e., some quantitative measures suitable for characterizing VSRs doing locomotion. In the following, we describe the choices adopted in this study for the fitness function, the representation, and the descriptors.

Fitness function for locomotion. Given a VSR, we perform a simulation lasting $60\,\text{s}$ (simulated time) where the VSR is initially placed right above an terrain. We take as fitness of the VSR its average velocity v_x along the x-axis measured by considering its center of mass position at $t = 0\,\text{s}$ and at $t = 60\,\text{s}$.

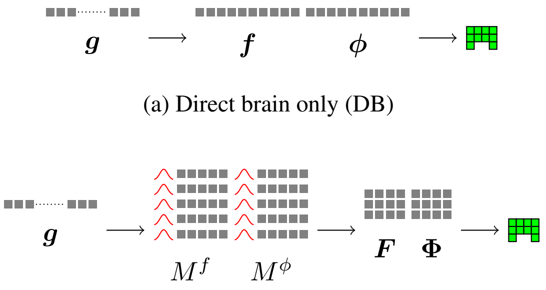
To make the task slightly more challenging, we consider a hilly terrain, instead of a flat, even terrain. The height of the terrain varies randomly along the x-axis: we use a single randomly generated terrain for all the experiments.

VSR representations. We consider four different ways of mapping numerical vectors to VSRs. They result from the combination of two options for two axes: direct vs. indirect representation; body and brain vs. brain only.

The direct representation for brain only optimization (DB) works as follows. Given a body consisting of n voxels, we map a vector $\boldsymbol{g} \in \mathbb{R}^p$, with $p = 2n$, to a VSR with the given body and equipped with the sinusoidal controller where, for each i-th voxel, the frequency is the i-th element of the first half ($\boldsymbol{f}$) of $\boldsymbol{g}$ and the phase is the i-th element of the second half ($\boldsymbol{\phi}$) of $\boldsymbol{g}$, with $\boldsymbol{g} = [\boldsymbol{f}\ \boldsymbol{\phi}]$. Since we use, for this representation, a 10-voxel body, that we call "biped", with two voxels as "legs" and a "trunk" of 4×2 voxels 🟩, it follows that for this representation we have $p = 20$. We schematize this representation in Figure 2a.

The indirect representation for brain only optimization (IB) is based on the concept of Gaussian Mixture Model (GMM) (Lindsay, 1995) and has already been used for 2D VSRs by Medvet et al. (2020b): it works as follows. Let n_{GMM} be the number of bi-variate Gaussian models in the mixture and let $w \times h$ the size of a 2D grid enclosing the VSR body—i.e., in the case of the biped, $w = 4$, $h = 3$. Each bi-variate Gaussian is described by five parameters: μ_x, μ_y, σ_x, σ_y, and β. We first map an individual $\boldsymbol{g} \in \mathbb{R}^p$, with $p = 2 \cdot 5 n_{\text{GMM}}$ to two sets of n_{GMM} bi-variate Gaussian models, one for the frequency, $M^f = \left\{\left(\mu_x^{f,i}, \mu_y^{f,i}, \sigma_x^{f,i}, \sigma_y^{f,i}, \beta^{f,i}\right)\right\}_i$, and one for the phase, $M^\phi = \left\{\left(\mu_x^{\phi,i}, \mu_y^{\phi,i}, \sigma_x^{\phi,i}, \sigma_y^{\phi,i}, \beta^{\phi,i}\right)\right\}_i$, of a sinusoidal controller; when mapping to σ_x and σ_y, we take the absolute value of the corresponding elements of $\boldsymbol{g}$. Then, we build a VSR with the given body and with a sinusoidal controller where frequencies and phases are determined as follows. For a voxel at position x, y (in the $w \times h$ 2D grid), we set the frequency to $f = \boldsymbol{F}_{x,y} = \operatorname{mix}(x', y'; M^f) = \sum_{i=1}^{n_{\text{GMM}}} \frac{\alpha^{f,i}}{2\pi\sigma_x^{f,i}\sigma_y^{f,i}} \exp\left(-\frac{1}{2}\left(\frac{\left(x'-\mu_x^{f,i}\right)^2}{\sigma_x^{f,i}} + \frac{\left(y'-\mu_y^{f,i}\right)^2}{\sigma_y^{f,i}}\right)\right)$, where $x' = \frac{x}{w}$ and $y' = \frac{y}{h}$. Similarly, we set the phase to $\phi = \boldsymbol{\Phi}_{x,y} = \operatorname{mix}(x', y'; M^\phi)$. In our experiments, we set $n_{\text{GMM}} = 5$ and apply this representation to the biped, hence $p = 50$. We schematize this representation in Figure 2b.

The direct representation for body and brain (DB^2) works as follows. Let n_{size} be the side of a square enclosing the largest representable body, i.e., a square of $n_{\text{size}} \times n_{\text{size}}$ voxels. We first take the vector $\boldsymbol{g} \in \mathbb{R}^p$, with $p = 3n_{\text{size}}^2$ and reshape it to three matrices $\boldsymbol{B}, \boldsymbol{F}, \boldsymbol{\Phi}$, each defined in $\mathbb{R}^{n_{\text{size}} \times n_{\text{size}}}$. We transform $\boldsymbol{B}$ to a Boolean matrix $\boldsymbol{B}' = \{\text{T}, \text{F}\}^{n_{\text{size}} \times n_{\text{size}}}$ where $B'_{x,y}$ is set to true if and only if $b_{x,y}$ is greater or equal than the median value of $\boldsymbol{B}$. Then, we build the body by considering the largest connected component of $\boldsymbol{B}'$ elements set to true and putting a voxel in the square at the coordinates of each element of such set. Finally, we build a sinusoidal controller for the body where the frequency and phase for each voxel at coordinates x, y are taken from the corresponding elements of $\boldsymbol{F}$ and $\boldsymbol{\Phi}$. In



(a) Direct brain only (DB)

(b) Indirect brain only (IB)

Figure 2: Processing steps of the DB and IB representations for the brain only, starting from the numerical vector $\boldsymbol{g}$ to the final VSR. Grids of gray squares represent vectors (e.g., ▪▪▪ $\in \mathbb{R}^3$) or matrices (e.g., ▦ $\in \mathbb{R}^{3\times 2}$). ∧▪▪▪▪▪ represents a bi-variate Gaussian model with its 5 parameters.

our experiments we set $n_\text{size} = 10$, hence $p = 300$. We schematize this representation in Figure 3a.

Finally, the indirect representation for body and brain (IB^2), based on GMM too, works as follows. Let n_size be the side of a square enclosing the largest representable body and let n_GMM be the number of bi-variate Gaussian models in the mixture. We first map a $\boldsymbol{g} \in \mathbb{R}^p$, with $p = 3 \cdot 5 n_\text{GMM}$ to three sets of n_GMM bi-variate Gaussian models, M^b, M^f, and M^ϕ, as for the IB case. We build a matrix $\boldsymbol{B} \in \mathbb{R}^{n_\text{size} \times n_\text{size}}$ by setting each element $B_{x,y} = \text{mix}(x', y'; M^B)$, with $x' = \frac{x}{n_\text{size}}$ and $y' = \frac{y}{n_\text{size}}$. Then, we build a body from $\boldsymbol{B}$ through an intermediate Boolean matrix $\boldsymbol{B}'$, as in the DB^2 case, but with 0.5 as threshold instead of the median of $\boldsymbol{B}$. Finally, we build a sinusoidal controller for the body where the frequency and phase for each voxel at coordinates x, y are set to $f = \text{mix}(x', y'; M^f)$ and $\phi = \text{mix}(x', y'; M^\phi)$. In our experiments we set $n_\text{size} = 10$ and $n_\text{GMM} = 5$, thus $p = 75$. We schematize this representation in Figure 3b.

VSR descriptors. We consider two sets of two descriptors: one characterizes the VSR body only, and the other characterizes the behavior, hence implicitly both together characterize the body and the brain. We use the former when using ME with the DB^2 and IB^2 representations, hence when evolving body and brain, and the latter when using ME with the DB and IB representations, hence when evolving the brain only.

As body descriptors, we simply consider the width and height of the smallest 2D grid enclosing the VSR body. Note that in our experiments, both descriptors are defined in $[1, 10] \in \mathbb{N}$, since $n_\text{size} = 10$ for DB^2, IB^2 and bodies are at most 10×10 large. We denote these descriptors by w and h respectively.

For the behavior descriptors, we consider the temporal pattern according to which the VSR touches the ground when doing locomotion. In detail, given a simulation of t_f seconds of the VSR, we proceed as follows. First, we build two binary signals $\tau_\text{back}, \tau_\text{front} : [0, t_f] \to \{0, 1\}$. At each t, $\tau_\text{back}(t)$ is 1 if at least one point of the leftmost half of the VSR was in contact, at t, with the ground, and 0 otherwise; with point of the leftmost half we mean a point whose x-coordinate is lower than the x-coordinate of the center of mass of the VSR. Similarly, $\tau_\text{forward}(t)$ considers the rightmost half of the VSR. Then, we compute the Fast-Fourier Transform for both signals and, for each one, we compute the amount of energy in the band 0 Hz–2 Hz and in the band 0 Hz–5 Hz. Finally, we define the descriptors ρ_back and ρ_front as the rate between the 0 Hz–2 Hz and 0 Hz–5 Hz energies for the τ_back and τ_front signals; both descriptors are defined in $[0, 1]$. Intuitively, the lower the gait pace, the greater the value of the descriptors.

Measuring and visualizing evolvability

As discussed earlier, evolvability is a characteristic of an evolutionary system that describes how much it is able to generate different and better performing individuals.

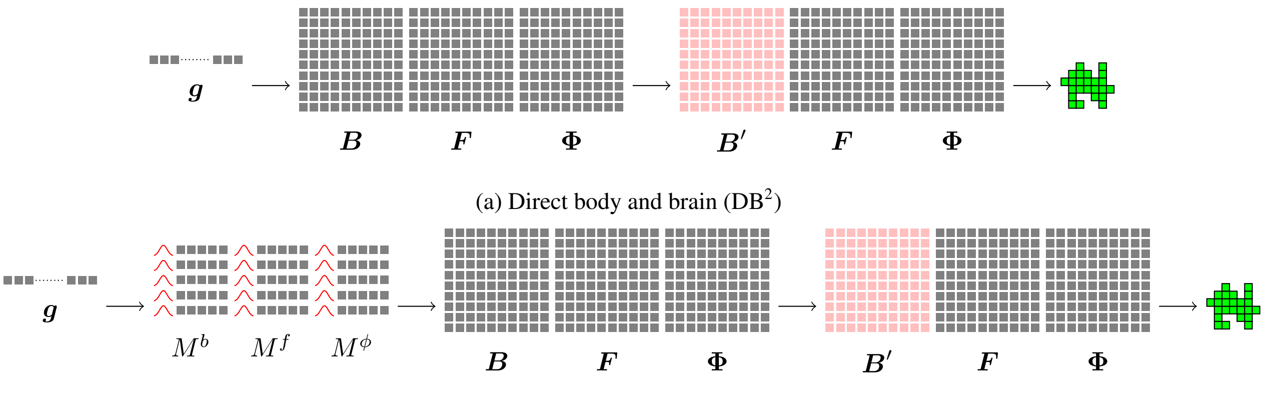
In this study, we define evolvability as a measure of an individual at a given iteration during the execution of an iterative EA. Formally, we define the evolvability $e(\boldsymbol{g}, i)$ of an individual $\boldsymbol{g}$ at iteration i as:

$$e(\boldsymbol{g}, i) = \frac{1}{|C_{\boldsymbol{g},i}|} \sum_{\boldsymbol{g}' \in C_{\boldsymbol{g},i}} f(\boldsymbol{g}') - f(\boldsymbol{g}), \tag{1}$$

where $C_{\boldsymbol{g},i}$ is the multiset of all the individuals generated from $\boldsymbol{g}$, i.e., its children, up to iteration i and $f(\boldsymbol{g})$ is the fitness of the individual $\boldsymbol{g}$.

While for ME the notion of children of an individual is trivial, since each (non-initial) individual has exactly one parent, in ES we assume that all the $n_\text{pop} - 1$ individuals generated at a given iteration are children of all the $\frac{n_\text{pop}}{4}$ parents chosen at that iteration.

For providing an aggregate view of the evolvability of an entire EA execution based on the individual measure of Equation (1), and with the aim of balancing the trade-off between detail and compactness of that view, we proceed as follows. During the execution of the EA, given some descriptors $d_1, \dots, d_m$, each defined as $d_i : G \to [d_{i,\text{min}}, d_{i,\text{max}}]$, and a number n_bin of bins, we maintain an initially empty archive A' with the same update policy of ME: whenever a new individual $\boldsymbol{g}$ is generated in the EA, it is added to A' if no other individuals exist in A at the same coordinates of $\boldsymbol{g}$ and it replaces, if any, the existing individual at those coordinates. At the end of the evolution, we analyze the individuals in A' by looking at their fitness and evolvability. Namely, if we use two descriptors, we can plot the values of $f(\boldsymbol{g})$ and $e(\boldsymbol{g}, i_\text{last})$ of each $\boldsymbol{g} \in A'$ in the form of two color maps, i_last being the last iteration in the EA execution. We refer to these plots as *fitness and evolvability maps*, respectively.



(a) Direct body and brain (DB^2)

(b) Indirect body and brain (IB^2)

Figure 3: Processing steps of the DB^2 and IB^2 representations for body and brain, starting from the numerical vector $\boldsymbol{g}$ to the final VSR. The visual syntax is the same of Figure 2; moreover, grids of pink squares represent Boolean matrices (e.g., 🟥 $\in \{T, F\}^{2\times 2}$).

Note that this way of tracking the evolvability of an EA execution does not interfere with the EA execution. In particular, in the case of ME, A' and A might be different, if based on different descriptors, domains, or numbers of bins. Nevertheless, we use for A', while executing both ME and ES, with the same parameters used for A in ME.

4 Experiments and discussion

We performed several evolutionary runs with the aims of (a) understanding if and how different EAs and representations produce different fitness and evolvability maps and (b) discovering relationships between fitness and evolvability maps.

For each of the eight combinations of representation (DB, IB, DB^2, IB^2) and EA (ES, ME), we performed 10 evolutionary runs (with different random seeds) and the following parameters.

For both ES and ME, we set $\sigma^2 = 0.35$ and $n_\text{eval} = 25\,000$. For ES, we set $n_\text{pop} = 20$. For ME, we set $n_\text{parent} = 20$, $n_\text{bin} = 10$, and, as descriptors, w, h with DB^2 and IB^2 and $\rho_\text{back}, \rho_\text{front}$ with DB and IB.

For DB and IB, we used the biped body 🟩. For IB and IB^2, we set $n_\text{GMM} = 5$. For DB^2 and IB^2, we set $n_\text{size} = 10$. As a result, the dimension of the search space $\mathbb{R}^p$ was 20, 50, 300, and 75, respectively for DB, IB, DB^2, and IB^2.

Finally, for computing the fitness and evolvability maps out of A' for each EA execution, we used the same descriptors and n_bin value used in ES, i.e., $n_\text{bin} = 10$ and w, h with DB^2 and IB^2 and $\rho_\text{back}, \rho_\text{front}$ with DB and IB.

The code for the experiments is publicly available at `https://github.com/ndr09/VSRevo`.

Overview: VSRs fitness and search efficiency

As initial point, we discuss the outcome of the evolutionary runs in terms of the effectiveness of the evolved VSRs in the task of locomotion. Figure 4 shows the trend of the fitness: at each iteration of the EA for each of the 8 combinations, the figure shows the median ± the standard deviation, computed across the 10 executions, of the fitness v_x of the best individual at that iteration.

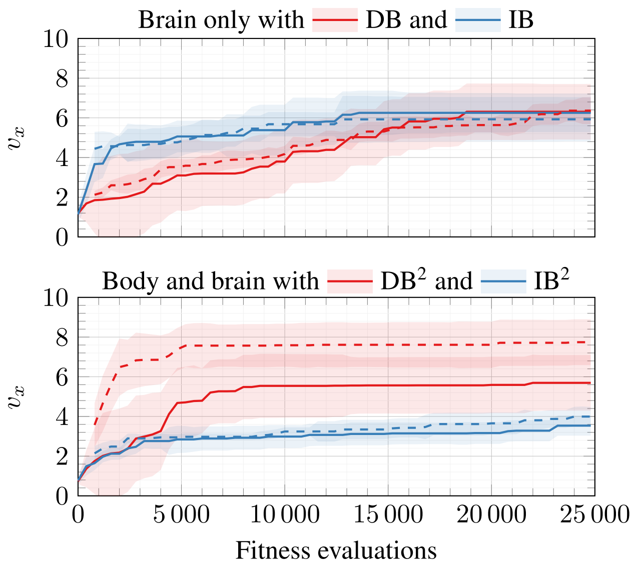


Figure 4: Fitness v_x (median ± standard deviation across the 10 runs) of the best individual in the during the evolution for the two EAs (—— ES and - - - ME) and the 4 representations, grouped in brain only (above) and body and brain (below).

First of all, we observe that all the approaches are able to produce effective solutions, achieving final best fitness values between 3, with the IB^2, and 8 with the DB^2+ME Values are in $\mathrm{m\,s^{-1}}$; as a reference, the side of each voxel is

	ES+DB^2	ME+DB^2	ES+IB^2	ME+IB^2
ES+DB^2	-	1.00	**0.0009**	**0.0030**
ME+DB^2	**0.016**	-	**0.0002**	**0.0002**
ES+IB^2	1.00	1.00	-	1.00
ME+IB^2	1.00	1.00	0.1818	-

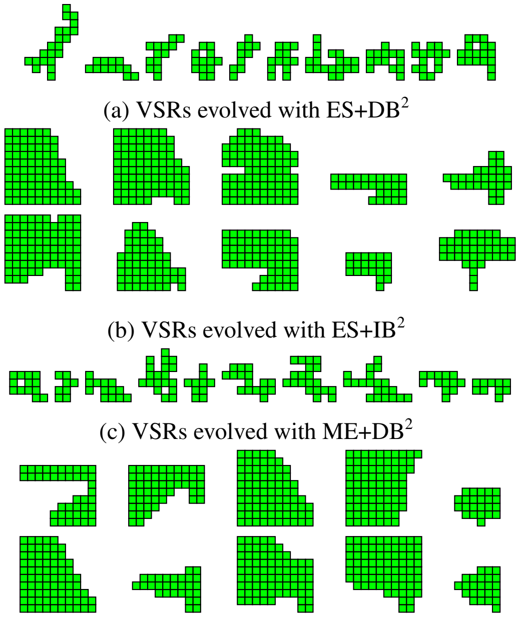
Table 1: Table of p-values obtained with the Mann-Whitney U test with Bonferroni correction on the final fitness reached in the brain and body evolution. The alternative hypothesis was set to check if the distribution in the row was greater than the one on the column. The entries indicated in bold rejects the null hypothesis with the corrected significance level of $\alpha = 0.003$.

$3\,\mathrm{m}$ long. In the other 5 cases, the final best fitness is ≈ 6.

Concerning the efficiency of the evolutionary optimization, we observe that the chosen value for n_{eval} appears to be large enough to let every combination converge to good solutions. Nevertheless, there are some differences among the combinations. The convergence happens earlier in the body and brain case, taking only ≈ 2000 and ≈ 7000 fitness evaluations for IB^2 and DB^2, respectively, vs. the larger values of the brain only case, $13\,000$ and $19\,000$ for IB and DB, respectively. In the brain only case, the indirect representation seems to enable a faster convergence than the direct one, despite the larger search space ($p = 50$ vs. 20).

While in the brain only case there are no apparent differences in the final best fitness, i.e., in the effectiveness of the evolutionary search, among the four cases, in the body and brain case the direct representation appears to be more effective than the indirect one, the former obtaining v_x values that are 2 to 2.5 times larger than the latter. The EA does not appear to play a role when coupled with IB^2, as confirmed by the Mann-Whitney U test (Table 1).

For explaining this performance gap, we observed the evolved VSRs for the body and brain case—see Figure 6 for an example of a robot behavior. We show all the $10 \cdot 4$ bodies evolved in the body and brain case in Figure 5. The most apparent difference between VSRs evolved with DB^2 and IB^2 is in the size, i.e., the number of voxels consituting the body. IB^2 are in general much larger: this is the combined effect of the different threshold being applied while building $\boldsymbol{B}'$ and the fact that IB^2, based on GMM, favors regular shape, and hence larger connected components, by design. Despite the fact that big and regular shapes have potentially more power to control the movement, Talamini et al. (2021) show that irregularity in the shape is an enabling factor for faster and more robust robots.


(a) VSRs evolved with ES+DB^2

(b) VSRs evolved with ES+IB^2

(c) VSRs evolved with ME+DB^2

(d) VSRs evolved with ME+IB^2

Figure 5: Body of the best robots at the end of the evolution for the DB^2 and IB^2 representations and the two EAs.

Fitness and evolvability maps

We here discuss the evolutionary runs in terms of the interplay between evolvability and fitness. Figures 7 and 8 show the fitness (top row, greenish colors) and evolvability (bottom row, red-violet colors) maps, one pair of maps for each combination of EA and representation. Each single map is obtained from 10 evolutionary runs: since individual runs can cover different portions of the descriptor grid, each cell in the overall map results from up to 10 corresponding cells of the individual maps. In particular, we use the median value of the cell value in the overall map.

The most apparent finding resulting from Figures 7 and 8 is that the maps for the evolvability are similar for the same EA regardless of the representation. On the other hand, the fitness maps differ for representation and are similar for the EA—the latter being more evident for the brain only case.

As regards the brain only case shown in Figure 7, we recall that the map is based on the behavioral descriptors ρ_{back} and ρ_{front} descriptors, i.e., they indicate how the robots move. We can see that faster robots have low ρ_{back} and ρ_{front} values: their energy is spent more in the $2\,\mathrm{Hz}$ to $5\,\mathrm{Hz}$ band than in the $0\,\mathrm{Hz}$ to $2\,\mathrm{Hz}$ band, i.e., they move they limbs at higher frequencies. Concerning the evolvability, the maps show that the evolvability is, generally, smaller than 0. This indicates that the initial heritable individuals are replaced in the map, as fitter individuals are generated. Moreover, we see a good match between the evolvability and fitness map: good fitness is related to low evolvability, which is expected, as it becomes harder, while the evolution progress, that high performing individuals produce better offspring.

In the body and brain optimization shown in Figure 8, maps are based on body descriptors w and h. The maps shows that almost all the cells are covered, meaning that all

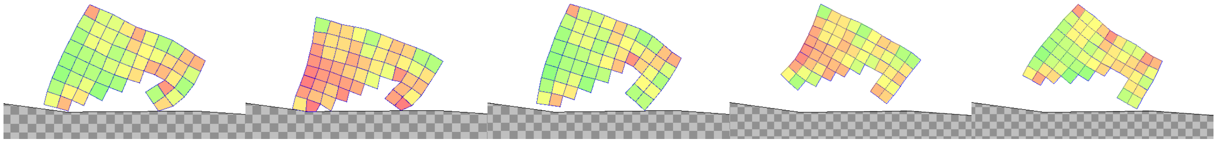


Figure 6: Frames of the simulation of one of the evolved VSRs (run 2 of ES+IB^2, see Figure 5b) doing locomotion. More videos of the behavior of evolved VSRs are available at https://youtu.be/nDUjq1VebsE.

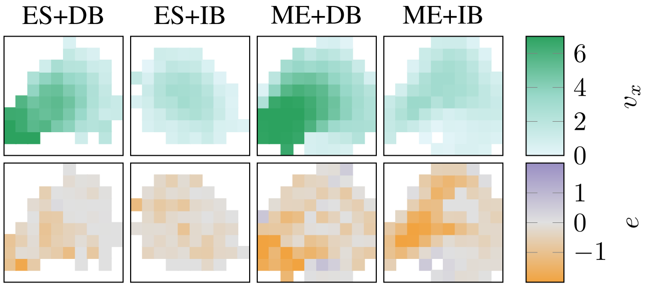


Figure 7: Fitness (top row) and evolvability (bottom row) maps for the DB and IB representations and the two EAs: ρ_{back} and ρ_{front} are used as descriptors.

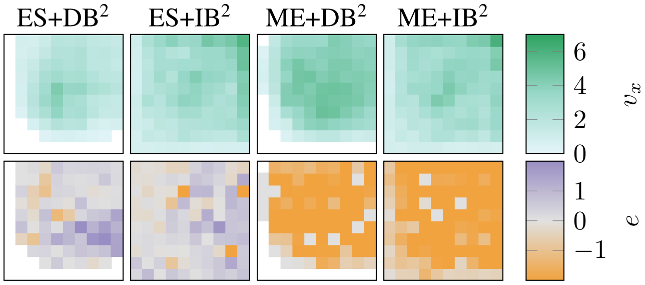


Figure 8: Fitness (top row) and evolvability (bottom row) maps for the DB^2 and IB^2 representations and the two EAs: w and h are used as descriptors.

the different sizes are found with the difference that DB^2 tends to avoid $1 \times n$ (and $n \times 1$) bodies. Differently from the previous case, here there are no regions in the grid with high fitness values. This is probably because it is harder to find good individual, while optimizing both the body and brain. For what concerns the evolvability map, we note that there are some grid portions with positive evolvability. This is likely because in those areas the individuals are selected just few times for reproduction, they give origin to high performing offspring, but are never replaced by better individuals.

Summarizing, the entanglement between the fitness and evolvability is related to the dynamics of the evolution. Negative values indicate that individuals in a cell generated less fit offspring, likely because they are themselves fairly fit. Positive values show that the individuals produced better children and were not replaced. Unfortunately, it is not possible to say if a cell in the map hosted an individual with good evolvability before being replaced by a fitter individual later in the evolution. This can be considered a limitation of this approach; the filling of the map is driven by the fitness: hence the historic information about the evolvability of individuals is lost when they are replaced. We plan to address this limitation as future work.

5 Conclusions and future works

We proposed the use of MAP-Elites as support structure for the calculation and visualization of evolvability. We tested this approach on a locomotion task carried out by Voxel-based Soft Robots, where we considered two different EAs (MAP-Elites and Evolution Strategies), with two genotypic representations (direct and indirect), applied to two optimization settings (robot brain only, and body and brain).

Our results show that, given a predefined phenotypic space (in the form of a descriptor grid, as in MAP-Elites), the distribution of evolvability over it (measured as the average difference in fitness between the offspring and their parents, considering only the offspring inserted in the grid during the evolutionary process), is mostly determined by the adopted EA. On the other hand, the distribution of fitness over the same phenotypic space depends, in our experiments, on the adopted representation. Overall, these findings suggest that evolvability is not an intrinsic property of the fitness landscape, or the genotypic representation, but rather it stems from multiple factors, including the EA being used and its capability to keep diversity and generate better solutions over time.

In future works, we will extend the analysis of evolvability to the case of sensor evolution and learning, which have been recently addressed in the context of VSRs in (Ferigo et al., 2021a, 2022) and (Ferigo et al., 2021b) respectively. Moreover, we will investigate how the proposed measure of evolvability correlate with some specific features of the fitness landscape, e.g., modality, as well as the behavior descriptors, which in turn may depend on the specific task (in the case of robots). Another possibility would be to embed a measure of evolvability into the evolutionary loop, for instance in the selection step. However, our preliminary results in this direction (not reported here for brevity) did not yield promising results. Finally, it would be interesting to test our proposed approach to calculate evolvability on other EAs, including e.g., novelty search (Lehman and Stanley, 2011a).

References

Beyer, H.-G. and Schwefel, H.-P. (2002). Evolution strategies–a comprehensive introduction. *Natural computing*, 1(1):3–52.

Carlo, M. D., Ferrante, E., Zeeuwe, D., Ellers, J., Meynen, G., and Eiben, A. E. (2021). Heritability in morphological robot evolution. *CoRR*, abs/2110.11187.

Coppinger, R. P. and Smith, C. K. (1983). The domestication of evolution. *Environmental Conservation*, 10(4):283–292.

Corucci, F., Cheney, N., Giorgio-Serchi, F., Bongard, J., and Laschi, C. (2018). Evolving soft locomotion in aquatic and terrestrial environments: effects of material properties and environmental transitions. *Soft robotics*, 5(4):475–495.

Cully, A., Clune, J., Tarapore, D., and Mouret, J.-B. (2015). Robots that can adapt like animals. *Nature*, 521(7553):503–507.

Darwin, C. (1875). *The Variation of Animals and Plants under Domestication*. John Murray, London, UK.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

Ferigo, A., Iacca, G., and Medvet, E. (2021a). Beyond body shape and brain: Evolving the sensory apparatus of voxel-based soft robots. In *EvoApplications 2021: Applications of Evolutionary Computation*, volume 12694, pages 210–226, Cham. Springer.

Ferigo, A., Iacca, G., Medvet, E., and Pigozzi, F. (2021b). Evolving hebbian learning rules in voxel-based soft robots. TechRxiv.

Ferigo, A., Medvet, E., and Iacca, G. (2022). Optimizing the sensory apparatus of voxel-based soft robots through evolution and babbling. *SN Computer Science*, 3(2):1–17.

Gajewski, A., Clune, J., Stanley, K. O., and Lehman, J. (2019). Evolvability es: scalable and direct optimization of evolvability. In *Genetic and Evolutionary Computation Conference*, pages 107–115, New York, NY, USA. ACM.

Hiller, J. and Lipson, H. (2011). Automatic design and manufacture of soft robots. *IEEE Transactions on Robotics*, 28(2):457–466.

Katona, A., Franks, D. W., and Walker, J. A. (2021). Quality evolvability es: Evolving individuals with a distribution of well performing and diverse offspring. arXiv:2103.10790.

Koler-Matznick, J. (2002). The origin of the dog revisited. *Anthrozoös*, 15(2):98–118.

Kriegman, S., Cheney, N., and Bongard, J. (2018). How morphological development can guide evolution. *Scientific reports*, 8(1):1–10.

Lehman, J. and Stanley, K. O. (2011a). Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation*, 19(2):189–223.

Lehman, J. and Stanley, K. O. (2011b). Improving evolvability through novelty search and self-adaptation. In *IEEE Congress of Evolutionary Computation*, pages 2693–2700, New York, NY, USA. IEEE.

Lehman, J. and Stanley, K. O. (2013). Evolvability is inevitable: Increasing evolvability without the pressure to adapt. *PloS one*, 8(4):e62186.

Lim, B., Grillotti, L., Bernasconi, L., and Cully, A. (2021). Dynamics-aware quality-diversity for efficient learning of skill repertoires. *CoRR*, abs/2109.08522.

Lindsay, B. G. (1995). Mixture models: theory, geometry and applications. In *NSF-CBMS regional conference series in probability and statistics*, pages i–163. JSTOR.

Liu, D., Virgolin, M., Alderliesten, T., and Bosman, P. A. N. (2022). Evolvability degeneration in multi-objective genetic programming for symbolic regression.

Medvet, E., Bartoli, A., De Lorenzo, A., and Seriani, S. (2020a). 2d-vsr-sim: A simulation tool for the optimization of 2-d voxel-based soft robots. *SoftwareX*, 12:100573.

Medvet, E., Bartoli, A., De Lorenzo, A., and Seriani, S. (2020b). Design, Validation, and Case Studies of 2D-VSR-Sim, an Optimization-friendly Simulator of 2-D Voxel-based Soft Robots. *arXiv*, pages arXiv–2001.

Medvet, E., Daolio, F., and Tagliapietra, D. (2017). Evolvability in grammatical evolution. In *Genetic and Evolutionary Computation Conference*, pages 977–984, New York, NY, USA. ACM.

Mengistu, H., Lehman, J., and Clune, J. (2016). Evolvability search: directly selecting for evolvability in order to study and produce it. In *Genetic and Evolutionary Computation Conference 2016*, pages 141–148, New York, NY, USA. ACM.

Methenitis, G., Hennes, D., Izzo, D., and Visser, A. (2015). Novelty search for soft robotic space exploration. In *Proceedings of the 2015 annual conference on Genetic and Evolutionary Computation*, pages 193–200.

Nordmoen, J., Veenstra, F., Ellefsen, K. O., and Glette, K. (2021). Map-elites enables powerful stepping stones and diversity for modular robotics. *Frontiers in Robotics and AI*, 8:1–17.

Payne, J. L. and Wagner, A. (2019). The causes of evolvability and their evolution. *Nature Reviews Genetics*, 20(1):24–38.

Pigliucci, M. (2008). Is evolvability evolvable? *Nature Reviews Genetics*, 9(1):75–82.

Pugh, J. K., Soros, L. B., and Stanley, K. O. (2016). Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI*, 3:40.

Squillero, G. and Tonda, A. (2016). Divergence of character and premature convergence: A survey of methodologies for promoting diversity in evolutionary optimization. *Information Sciences*, 329:782–799.

Talamini, J., Medvet, E., and Nichele, S. (2021). Criticality-driven evolution of adaptable morphologies of voxel-based soft-robots. *Frontiers in Robotics and AI*, 8:172.

Tarapore, D. and Mouret, J.-B. (2015). Evolvability signatures of generative encodings: Beyond standard performance benchmarks. *Information Sciences*, 313:43–61.

Veenstra, F., de Prado Salas, P. G., Stoy, K., Bongard, J., and Risi, S. (2020). Death and progress: How evolvability is influenced by intrinsic mortality. *Artificial life*, 26(1):90–111.

Wayne, R., Van Valkenburgh, B., and O'Brien, S. J. (1991). Molecular distance and divergence time in carnivores and primates. *Molecular Biology and Evolution*, 8(3):297–319.