Morphological Development at the Evolutionary Timescale: Robotic Developmental Evolution

Morphological Development at the Evolutionary Timescale: Robotic Developmental Evolution

Fabien C. Y. Benureau*
Okinawa Institute of Science and Technology
fabien.benureau@oist.jp

Jun Tani
Okinawa Institute of Science and Technology
jun.tani@oist.jp

Abstract Evolution and development operate at different timescales; generations for the one, a lifetime for the other. These two processes, the basis of much of life on earth, interact in many non-trivial ways, but their temporal hierarchy—evolution overarching development—is observed for most multicellular life forms. When designing robots, however, this tenet lifts: It becomes—however natural—a design choice. We propose to inverse this temporal hierarchy and design a developmental process happening at the phylogenetic timescale. Over a classic evolutionary search aimed at finding good gaits for tentacle 2D robots, we add a developmental process over the robots' morphologies. Within a generation, the morphology of the robots does not change. But from one generation to the next, the morphology develops. Much like we become bigger, stronger, and heavier as we age, our robots are bigger, stronger, and heavier with each passing generation. Our robots start with baby morphologies, and a few thousand generations later, end-up with adult ones. We show that this produces better and qualitatively different gaits than an evolutionary search with only adult robots, and that it prevents premature convergence by fostering exploration. In addition, we validate our method on voxel lattice 3D robots from the literature and compare it to a recent evolutionary developmental approach. Our method is conceptually simple, and it can be effective on small or large populations of robots, and intrinsic to the robot and its morphology, not the task or environment. Furthermore, by recasting the evolutionary search as a learning process, these results can be viewed in the context of developmental learning robotics.

1 Introduction

Generations evolve; individuals develop. Many of the evolutionary processes of a species, then, happen at a timescale an order of magnitude or more greater than the developmental ones; evolution is a long-term species-level process happening on top of many short-term individual developmental processes. This is the rhythm of our biological world.

Robots do not have to abide by such principles. Here, we propose to inverse the timescales and put a developmental process on top of an evolutionary one. We consider an evolutionary process with robots; the first-generation robots are small and weak and light. They are not the target

---

* Corresponding author.

morphology, only baby versions of it. They attempt to solve a task; the best performers survive and produce slightly mutated offspring of themselves for the next generation. Those second-generation offspring are a bit bigger, a bit stronger, and a bit heavier than the first generation. The second generation, in essence, *grows up* compared to the first one. Following a predefined developmental schedule happening at the phylogenetic timescale, each generation will be bigger and stronger and heavier than the previous one until reaching adult morphology. Our robots do not develop during fitness evaluation. Their morphology is determined at birth and remains constant while they are evaluated on the task. It is a developmental process *for generations*, rather than for an individual. Instead of *evolutionary developmental robotics*, this is *developmental evolutionary robotics*: *devo-evo-robo* rather than *evo-devo-robo*.

To understand some of the motivations behind this work, another perspective on it is useful. Above, we propose a new method for evolutionary robotics—adding development on top of evolution—and demonstrate that this approach can improve which behaviors are discovered and try to analyze why it does. Yet, another view is to consider the evolutionary search as a rudimentary trial-and-error learning process employed by a single robot. In that context, a generation of 20 robots becomes an epoch of 20 trials, and the 4,000 generations of the evolutionary search turn into our lone robot's lifetime: 4,000 epochs of learning. *Developmental evolution* becomes *developmental learning* (Cangelosi, 2015; Lungarella et al., 2003), and our initial question, *Can development improve the evolutionary process?*, becomes *Can growing up help a robot learn better?*, which may help, in turn, to shed light on the question: *Does growing up help us learn better?* Morphological computation (Pfeifer & Gómez, 2009) looks at how morphology participates in behavior. We are interested here in how morphology, or more precisely, morphological change, participates in behavior acquisition. Those underlying questions critically motivate our investigation.

The overwhelming majority of animals go through morphological growth at the beginning of their lives, yet the research on growing robots is scarce, and most of them mimic the indeterminate growth of plants (Corucci et al., 2017; Dottore et al., 2018). Most of the previous work on development in evolutionary robotics has focused on adding developmental processes to the fitness evaluation phase, i.e., *evo-devo* approaches (Bongard, 2011a; Corucci, 2016; Kriegman, Cheney, & Bongard, 2018; Kriegman et al., 2017, 2018). While this mimics biology at the phylogenetics timescale, computational and time constraints limit the extent and complexity of the developmental program, which must fit entirely in an evaluation period, usually a few dozens of simulation-seconds long. Vujovic et al. (2017) illustrates this: By considering real-world robot development within a generation, and having to construct and evaluate three different robot morphologies for each fitness evaluation, the number of generations, five, remains drastically limited. Furthermore, because even in simulation, fitness evaluations rarely exceed a few hundreds of seconds, this also forces development to happen at the same timescale as behavior, which may be problematic if modeling biology. By contrast, development slowly rolling out over generations has the potential to implement complex developmental paths, without adding any more computational cost per generation than a classical evolutionary algorithm, while allowing development to happen at an arbitrary slower timescale than a single behavior. To our knowledge, only Joshua Bongard has explored a *devo-evo* approach, studying a robot going through four different morphologies over the course of evolution, from anguilliform to legged adult, to conclude that it made the task more difficult, worsening performance, and proposing as a better-performing approach one where robots develop over the evaluation period (Bongard, 2011a).

Our approach is an incremental evolution approach (Doncieux & Mouret, 2014; Mouret & Doncieux, 2008): The evolutionary search starts with an altered version of the target task, and then, incrementally as the search progresses, is transformed into the target task. There are several methods in incremental evolution: staged evolution (splitting a difficult task into subtasks) (Mouret et al., 2006; Parker, 2001; Urzelai & Floreano, 1999; Urzelai et al., 1998); behavior decomposition (training sub-controllers); fitness shaping (modifying the fitness function to create smoother gradients) (Colby & Tumer, 2015; Nolfi, 1997); and environment complexification (making the environment/task harder or more complex as the search progresses) (Bongard, 2011b;

Gomez & Miikkulainen, 1997; Miras & Eiben, 2019a,b). Our approach falls into the environment complexification methods, except that, in contrast to most methods, we modify the robot itself—its morphology—rather than the rest of the environment.

Our approach distinguishes itself from existing incremental methods in two ways. First, most incremental approaches work by making the problem easier and working their way up to the target task: splitting a complex task into simpler subtasks, training subcontrollers, making the fitness function less rugged or less sparse, and progressively complexifying the environment—for instance, progressively increasing a prey speed when training a predator behavior (Gomez & Miikkulainen, 1997). By contrast, our approach is not to make the problem obviously easier. Indeed, much like human babies do not have an easier time walking than adults, our robots start with immature bodies that are less apt at walking than adults. We would expect a decrease in final, adult performance, as a significant part of learning happens on a different, and in some cases obviously inferior morphology. This is actually one of the main theoretical contributions of this article: Despite starting with immature bodies, our population of robots discover better behaviors. Morphological development here, rather than being a hindrance, helps.

Second, incremental evolution approaches have been shown to be effective, but they necessitate designing a tailored incremental program to modify the task or environment. This requirement for hand-tuned expert knowledge on a case-by-case basis severely limits their generality and introduces bias into the solution. Environmental and task complexification mirrors teaching in humans, where the best results are obtained by an attentive teacher tailoring a curriculum of lessons and exercises of increasing difficulty to the student, with a change of topics requiring the design of a new curriculum, usually from scratch. By contrast, the fundamental mechanisms of human development have changed little over the last thousands of years, while the environment, the tasks we engage in, and the skills we develop have undergone dramatic transformations. Morphological development in robots is modeled after the latter. It does not require any modification of the environment or the task; it is intrinsic to the robot itself and to its embodiment. This opens the possibility that there exists, much like in humans, developmental programs for a given robot that can be effective in a wide range of tasks and environments. While this possibility is a major motivation for our study, this is not something we establish here, and for the time being, we have to rely on hand-tuned developmental paths.

But there is an even stronger argument to be made there: As we observe that humans, after an extended developmental period, end up possessing generalization capabilities unmatched across the animal kingdom (Anderson, 2003; Shapiro, 2011; Smith & Gasser, 2005), we may formulate the hypothesis that development is not just able to adapt, but also *needed* in order to adapt to a range of different tasks, needed for the generalization capabilities of humans. Morphological development may one day participate in making robots, as Linda B. Smith says, *flexibly smart* (Smith & Gasser, 2005). It goes without saying that our results are limited and simplistic in most of their aspects, and prove little robotically and nothing biologically. Still, those questions and hypotheses underpin the work presented.

The developmental trajectories we explore in this article are inspired by morphological growth in the animal world: change in size, in mass, in muscle strength. In particular, we consider a simplified model of muscle development for our robots, where those three dimensions develop together under physical constraints. With this simple physically plausible muscle model, we show that robot generations that grow up discover better ways to move than robot generations that only feature adult robots. Then, we study how development affects exploration within the evolutionary process.

2 Methodology

We consider simple "starfish" soft-robots in a 2D world. The robots possess a hexagonal body, and six tentacles stem from it. The robots are subjected to gravity and laid on a flat floor. They are evaluated on their ability to move as far as possible along the floor in 60 s. An evolutionary search is performed over the gaits of the robots.

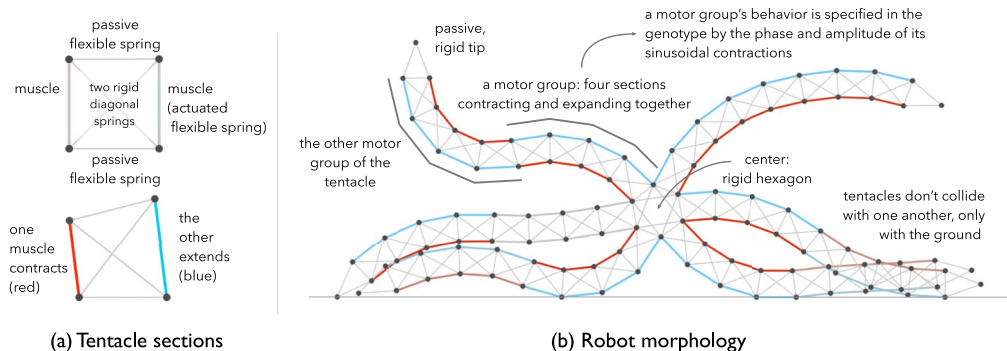(a) Tentacle sections   (b) Robot morphology

Figure 1. Our robots are made of point masses and springs. (a) A section is composed of two passive and rigid (high-stiffness) diagonal springs, two passive and flexible springs on the top and bottom, and two actuated springs, the *muscles*, on each side. The muscles work in pairs and act in an antagonistic manner: When one contracts by decreasing its resting length (in red), the other extends (in blue). The color represents the command sent to the muscle, i.e., the desired length of the muscle, not the actual length. (b) The sections are assembled into tentacles, with a passive triangular tip at the end. Sections are grouped into motor groups that actuate together, and attached to a central hexagonal body.

2.1 Robots

We consider a physics engine with only three basic elements: point masses, springs between point masses, and a flat floor that collides with the point masses. Each point mass experiences reaction forces and friction forces from the ground and gravity, as well as the forces of the springs connected to it. Given a spring linking two point masses of mass m_A and m_B, with resting length x_r, length x, stiffness k, and damping ratio ζ, we have:

$$m_r \ddot{x} = -k(x - x_r) - c\dot{x} \quad \text{with} \quad m_r = \frac{1}{\frac{1}{m_A} + \frac{1}{m_B}} \quad \text{and} \quad c = 2\zeta\sqrt{m_r k}$$

Our robots have passive springs for which the resting length is fixed, and actuated springs, where the resting length is modified by the motor commands of the robot: decreased for contraction and increased for extension. We will refer, in the rest of the text, to actuated springs as the artificial *muscles* of the robot.

Our robots follow a starfish pattern: a main body—a regular hexagon made of rigid springs— from which six tentacles stem (Figure 1). Each tentacle is composed of eight sections, plus a passive triangular tip. Each section is a square with rigid diagonal springs and passive flexible springs at the top and bottom. On each side of the section, which also is the side of the tentacle, a muscle— an actuated spring—is present. All the nodes of a given robot have the same mass, and all its muscles have the same stiffness. The stiffness of passive springs is the same in all robots. (For a detailed description, see the data contained in the Online Supplemental Material, which can be found at https://doi.org/10.1162/artl_a_00357.)

The two actuated side-springs work as antagonistic muscles: When the tentacle section receives a motor command, one muscle contracts while the other muscle extends. The motor command of a section is a scalar, received at each timestep; if a section of height h receives the motor command α, the target length—i.e., the resting length of the spring—of the left muscle becomes $(1 + \alpha)h$, while the target length of the right muscle becomes $(1 - \alpha)h$.

Sections are gathered into motor groups (Figure 1(b)). An eight-section tentacle is divided into two groups of four sections each. All the sections of the same group receive the same motor command, and therefore expand and contract simultaneously.

The gait of the starfish applies a sinusoidal actuation signal over each motor group. The period of the sinusoid is fixed (at 2π) and shared by all the motor groups, and each motor group has an independent phase (in $[-\pi, \pi]$) and amplitude (in $[0, 0.2]$). This allows encoding the gait of the

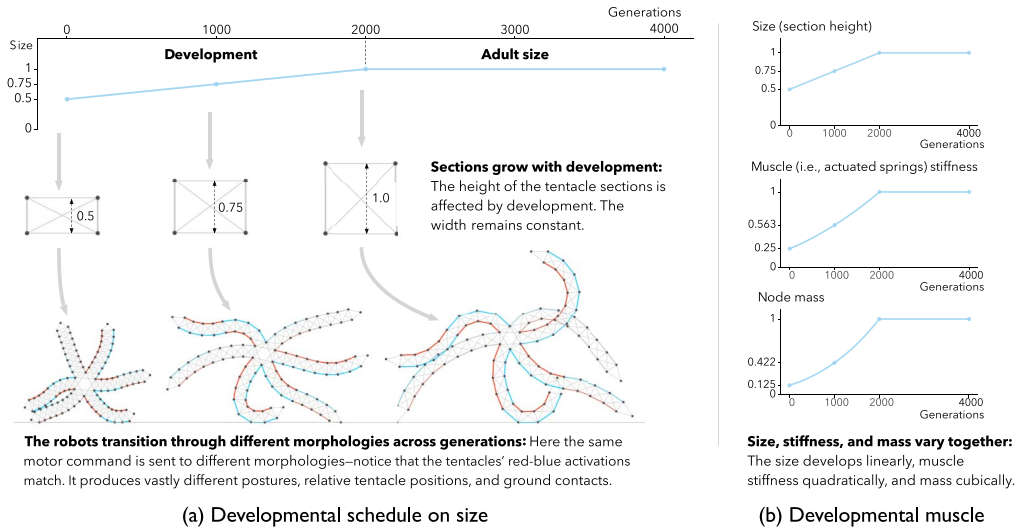(a) Developmental schedule on size

(b) Developmental muscle

Figure 2. The robots grow up according to a predefined schedule. (a) From generation 1 to 2,000, the height of the tentacle sections grows linearly from 0.5 in the first generation to 1.0, the adult size. The size remains the same for generation 2,000 to 4,000. (b) The developmental muscle model combines the development of three morphological characteristics of the muscle in parallel: size, muscle strength, and node mass.

starfish with two scalars per muscle group, and thus with 24 scalars in total for the starfish. Those 24 scalars form the genotype of our robots.

2.2 Task

The robots must move to the right as far as possible in 60 s; the fitness is the distance covered. The robots are dropped just above the ground. To avoid the robots taking advantage of the drop to bounce off the ground, which may produce chaotic fitness values, the robot settles for 9.42 (3π) s on the ground with no motor activation (0.0 actuation value sent to all motor groups). The robot then actuates until the 60 s mark, resulting in eight full actuation periods.

We employ a simple evolutionary strategy. At each generation, the five members with the best fitness become parents of the next generation, each creating three children through mutation. The parents survive too, creating a new population composed of the five parents and the 15 children.

Typically, in evolutionary robotics, the search is conducted from start to finish on the target—adult—morphology. We will call this *adult evolution*, and it will serve as control to evaluate the performance of *developmental evolution*, which we explain now.

2.3 Developmental Evolution

Developmental evolution considers populations of robots that develop across generations. During fitness evaluation, the morphology of the robots remains fixed and is the same for all members of the generation. But, from one generation to the next, the morphology changes slightly. This change is not under evolutionary control. It is predefined and goes according to a fixed schedule.

Figure 2(a) shows an example. In the first generation, the robots start at size 0.5: The height of the tentacle sections (including the tip) is half the height of the adult morphology. Each subsequent generation will see the size increment linearly (by 0.5/2,000 = 0.00025) until generation 2,000, when the adult morphology (size 1.0) is reached. For the next 2,000 generations, until generation 4,000—the end of the search—the robots will keep this adult morphology. In our evolutionary algorithm, the parents survive to the next generation. Therefore, their morphologies are adjusted to the new developmental values and they undergo fitness evaluation again.

Size is not the only morphological characteristic whose development we study. We analyze how developing muscle strength and the mass of the nodes of the robot affect the evolutionary process, including what happens when those three variables—size, strength, mass—develop together. For this, we create a crude developmental muscle model.

2.4 Developmental Muscle Model

In animals, muscle strength is a function of an array of factors. One of the most prominent ones is the cross-sectional area of the muscle, which has a proportional effect on strength (Sacks & Roy, 1982). If a muscle doubles in size in every proportion, its cross-sectional area quadruples, and therefore the strength is affected in a quadratic fashion. Meanwhile, the mass of the muscle, dependent on volume, is affected cubically.

We use these insights to create a simple developmental muscle model for our robots that combines the development of size, strength, and mass. In our model, when the size doubles, the muscle strength—i.e., spring stiffness—quadruples and the node mass is multiplied by eight. We express size, stiffness, and mass as coefficients of the adult values (see Online Supplemental Material for a description of the robots), and therefore the adult value for all characteristics is 1. For a given size coefficient s, the stiffness coefficient is s^2 and the mass coefficient s^3 (Figure 2(b)).

3 Experiments and Results

3.1 Developmental Muscle Model Experiment

We ran evolutionary searches with the developmental muscle model and contrasted them with the adult evolution, where the robots are adult from the first generation. Figure 3(a) shows two evolutionary runs of developmental evolution using the developmental muscle model, where the robots start with a size of 0.5 in the first generation, with the corresponding values of 0.25 for muscle stiffness and 0.125 for the mass of the nodes (see Movie S1 in the Online Supplemental Material for behavior throughout the generations for the first graph). They then gradually increment toward the adult values for the size, stiffness, and mass respectively for the next 2,000 generations. In the two examples, the adult evolution displays an archetypical learning curve, which progresses quickly at the beginning and then flattens. In contrast, the muscle model's developmental phase is characterized by sudden changes in fitness, both increases and collapses. Some of the behaviors, especially in the first graph, have higher fitness on young morphologies than the fitness found by the adult evolution. After the development ends, the fitness stabilizes around a significantly higher value than the adult evolution.

This difference in performance is statistically confirmed in Figure 3(b), where the average fitness over generations for 100 repetitions of the experiment is shown. (For a given condition, the 100 repetitions use the random seeds 1, 2, …, 100 and therefore the same initial population of genotypes, creating paired experiments across different conditions.) We observe a steady rise of fitness during development, with the average over 100 repetitions smoothing out the sudden collapses and increases of an individual run. At the end of development, the fitness is already significantly better than the adult evolution ($p < 0.001$, Wilcoxon signed-rank test). The average fitness then continues to increase after generation 2,000 for a few hundred generations at a larger rate than the adult evolution before stabilizing.

Figure 3(c) gives insights into the effect of different developmental trajectories by comparing different starting sizes for the developmental muscle model. Starting smaller, weaker, and lighter provides a sizeable and significant increase in performance compared to the adult evolution, with a peak observed for starting size 0.5. This increase is reduced when the starting values of the development get closer to the adult ones, but even a modest development—starting size 0.9—provides notable benefits to fitness. Starting bigger, stronger, and heavier is not as effective and does not bring significant benefits.
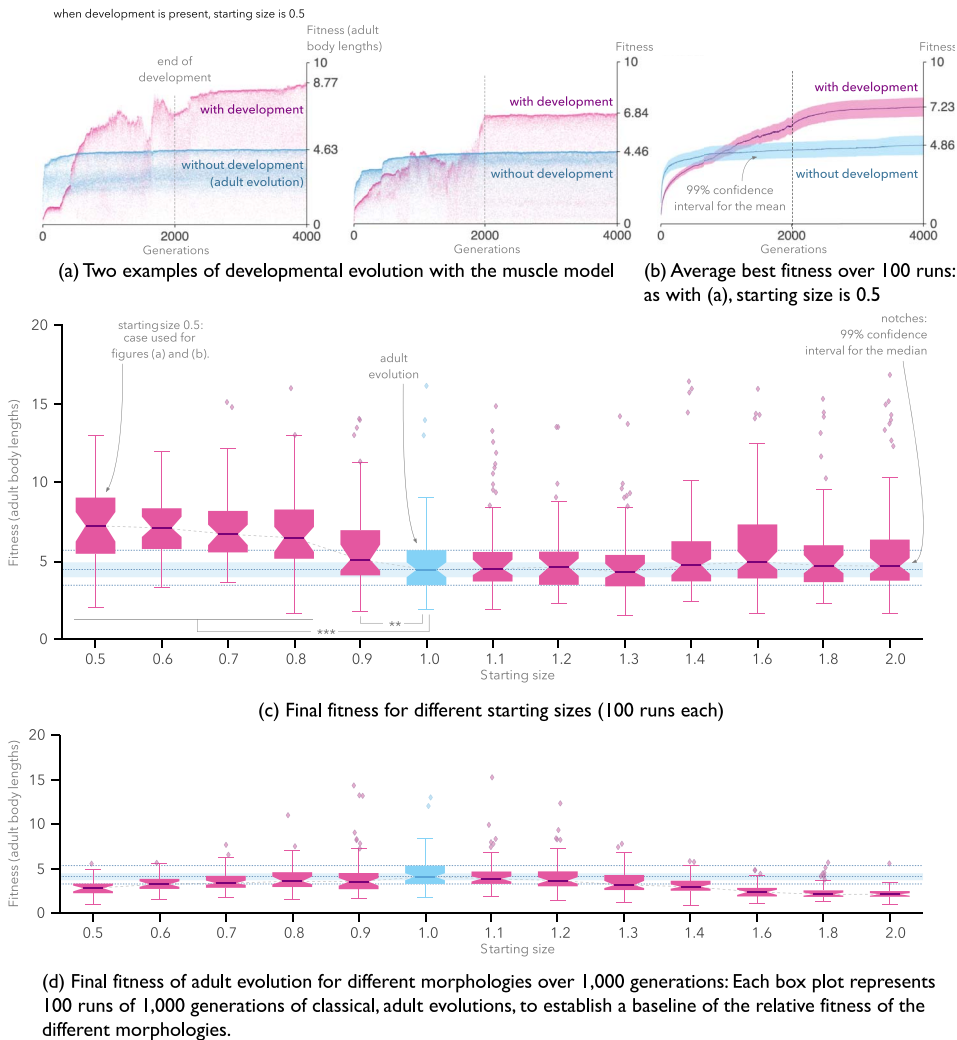
(a) Two examples of developmental evolution with the muscle model

(b) Average best fitness over 100 runs: as with (a), starting size is 0.5

(c) Final fitness for different starting sizes (100 runs each)

(d) Final fitness of adult evolution for different morphologies over 1,000 generations: Each box plot represents 100 runs of 1,000 generations of classical, adult evolutions, to establish a baseline of the relative fitness of the different morphologies.

Figure 3. The developmental muscle model outperforms adult evolution. (a) Two evolutionary runs of developmental evolution with the muscle model at starting size 0.5, out of the 100 runs performed, contrasted with the adult evolution performed with the same initial population. Each dot represents the fitness of a population member. Not shown: members with negative fitness. (b) Average final fitness over the 100 runs of the developmental muscle model with starting size 0.5 and the adult evolution. The final fitness is the mean of the 10 best fitnesses obtained in the last 50 generations. The shaded area is the 99% confidence interval of the mean of the final fitness over 100 runs. (c) Comparison of the distribution of the final fitness of the developmental muscle model for different starting sizes. Box plots are computed from 100 runs each, and the same set of 100 random seeds is used for each box plot, creating paired experiments with the same initial populations. Box plots show the first and third quantile, and the minimum and maximum value, or 1.5 × Interquartile Range, whichever is closer to the median; outliers are represented by diamonds. Notches represent the 99% confidence interval of the median. Significance stars are computed using a Wilcoxon signed-rank test after Shapiro-Wilk testing revealed that the differences between some of the paired fitnesses deviate significantly from normality. The significance threshold is set at 0.01. These conventions will be used for all box plots in the article. In blue, the adult evolution, equivalent to a starting size of 1.0, has its mean, confidence interval, and first and third quantile extended through the plot by a light shaded blue area and dashed horizontal lines. (d) Adult evolution over 1,000 generations over all the starting morphologies of (c). No development is involved.

Interestingly, none of the starting morphologies with the developmental model is better at solving the task than the adult morphology, as Figure 3(d) shows. Over 1,000 generations, all the starting morphologies are learned on a standard adult evolution, to establish a baseline of the fitness they can reach. The results show that development can provide higher fitness by passing through

morphologies that are worse at the task than the target one. Moreover, given that the fitness is already better at generation 1,999 in Figure 3(b), developmental evolution can provide better fitness on the adult morphology without ever directly experiencing it.

To understand the causes behind those dynamics, it is useful to look at the effect of developing the size, the muscle stiffness, and the mass separately. In each instance, the two other characteristics are fixed at adult values during the entire evolutionary process. The results are shown in Figure 4.

Perhaps unsurprisingly, having low mass or being strong at the beginning brings unequivocal performance increases. These results would be easily explainable if those characteristics—low mass or high stiffness—were permanent. But the performances displayed here are the ones of the adult morphology, after development has ended, since they are calculated from the mean of the 10 best individuals of generations 3,951–4,000. It is interesting to notice that the increased fitness of the low-mass or high-stiffness start have increased variability across different runs. What is hidden here is that we have two populations of behaviors: rolling and non-rolling behaviors. For our purposes here, we will define a rolling behavior as the robot's central body having done two or more complete revolutions on itself by the end of the evaluation. Typically, non-rolling behavior manifests as crawling or shuffling (see Movies S2 and S3 in the Online Supplemental Material for examples of non-rolling and rolling behavior respectively). Rolling is a highly beneficial behavior and allows reaching fitness scores that non-rolling behavior does not. This is shown with Figure 4(d), (e), and (f): The distributions of fitness of the rolling (in grey) and non-rolling (in color) behaviors are separated. We observe that the low-mass and high-fitness developments generate a lot of rolling behaviors compared to adult evolution, which is hardly surprising: It is easier to start a roll when light or strong. The evolutionary process is then able to refine and adapt the rolling behavior to the changing morphology, retaining them into adulthood.

Conversely, when handicapping the robot during development by a high mass or low stiffness, no significant fitness benefit is observed, and the number of rolling behaviors discovered is even lower than in adult evolution: Indeed, the robot has difficulties moving from its initial position when development starts, much less engaging in a highly dynamical behavior such as rolling. It is notable though that these handicapping developmental paths do not have a long-term detrimental effect: The final fitness is not significantly different from the control. Development, here, is robust and degrades gracefully.

Here, it is interesting to remark that the difference between morphological development and environmental scaffolding is sometimes thin. In particular, developmental evolution with a low starting mass is quite similar—although not equivalent—to environmental scaffolding with a low starting gravity. Figure S5 in the Online Supplemental Material confirms this; the performance impact of both interventions is remarkably similar. Of course, changing gravity outside of simulations is hardly ever a practical method.

Developing the size brings interesting results. Starting small generates the most benefits, and the benefits increase the smaller the robots start. Few rolling behaviors are discovered. Remarkably, when comparing the fitness of the non-rolling behaviors alone across experiments (Figure 4(d), (e), and (f)), the benefits of development for starting small or strong or light are similar. Starting big is helpful as well, with more rolling behaviors discovered. One hypothesis to explain the increase in rolling behavior is that longer tentacles tend to break (see Figure S7 in the online supplement), and thus are not as much an obstacle to rolling as they might seem. Here the importance of *discovering* the behavior seems paramount, and once discovered, maintenance of it by the evolutionary process through the morphological changes seems comparatively easier, even as tentacle breaks become less and less possible as the morphology nears the adult body.

Overall, we observe here two important dynamics. First, developmental evolution can generate better behaviors and help discover efficient behaviors that are hard to find if experiencing only the adult morphology. And second, different developmental paths lead to different behaviors: Starting small is highly beneficial, as it improves the behavior of the robots significantly, but it will not induce many rolling behaviors, whereas a high-stiffness or low-mass developmental path will.
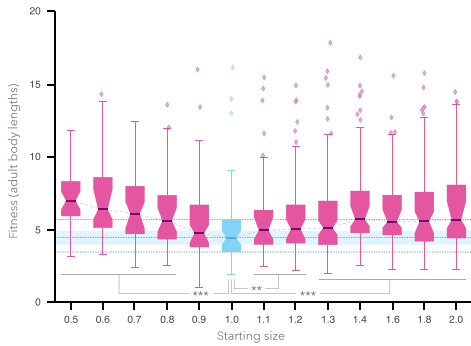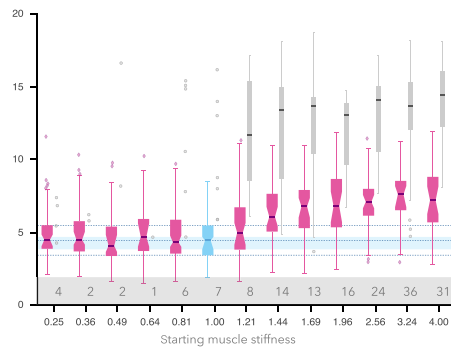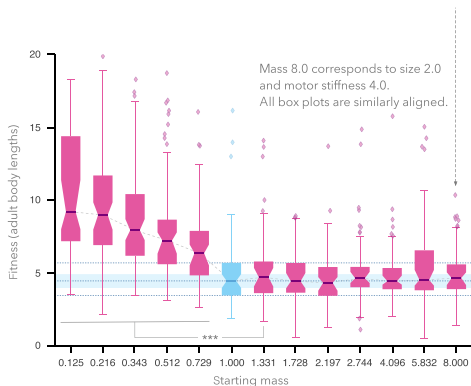
(a) Size development only
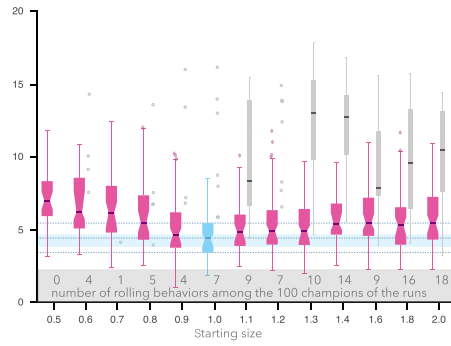
(d) Size development—No rolling: In (d), (e), (f), the colored, notched box plots show the distribution of fitness of the non-rolling behaviors. In grey, non-notched box plots and circle show one of the rolling behaviors.
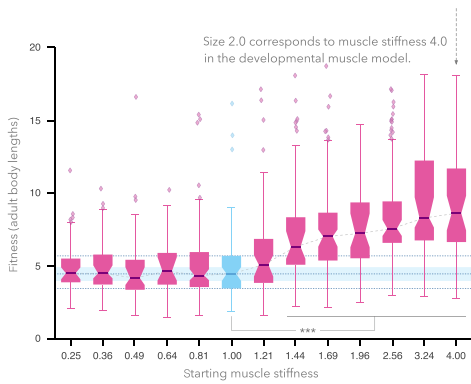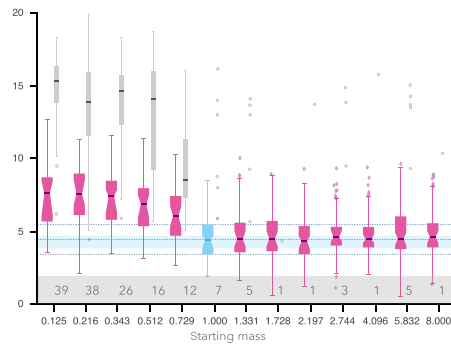
(b) Muscle stiffness development only

(e) Muscle stiffness development—No rolling: Fewer than 8 rolling behaviors, shown as individual samples. More than 8, shown as box plots.

(c) Mass development only

(f) Mass development—No rolling

Figure 4. Size, muscle stiffness, and mass development can all benefit the evolutionary process. The experiments in (a), (b), and (c) show the effect of the development that happened in Figure 3(c) when size, muscle stiffness, and mass develop on their own respectively, while the two other characteristics are fixed at adult values. Box plots conventions are the same as in Figure 3. (d), (e) and (f) show the same data as (a), (b), and (c), respectively, with the rolling (in grey) and non-rolling (in color) behaviors separated. The non-rolling behaviors are represented by box plots when there are 8 samples or more, and otherwise by grey circles (also used for outliers). Due to the small and varying sample size, the rolling box plots are not notched.

It remains to be explained, though, why non-rolling behaviors are improved as well by developmental evolution. To do that, it is useful to look at how the method impacts the exploration of the search space.

3.2 Development Fosters Exploration

For the first half of the evolutionary search, developmental evolution changes the morphology of the robots between every generation. Therefore, the fitness landscape changes with each generation during that time. Gaits that were competitive on one morphology are less so on the next, and are quickly superseded by new gaits; development, quite straightforwardly, prevents convergence. This can be observed in Figure 5(a), which shows a principal component analysis (PCA) of the champions of each generation (filled triangles), plus their ancestors (hollow triangles), for the adult evolution and the developmental muscle model with starting size 0.5. Developmental evolution travels much more than adult evolution across the search space and is especially mobile during the developmental period.

To quantify it, we define the *genealogy distance* between a member of the population and one of its ancestors as the sum of all the *mutation distances*—i.e., the euclidean distances—between successive members of the genealogy that goes from the ancestor to the member we are considering. In Figure 5(a), the genealogy distance between the initial root member and the elite of the 2,000th generation is 28.75 (over the 100 runs: 30.28 ± 0.74, mean with 99% confidence interval), versus 2.27 (6.66 ± 0.84) for the adult evolution. From generation 2,000 to the elite of generation 4,000, the genealogy distance is 3.99 (3.48 ± 0.59) versus 3.78 (1.81 ± 0.45) for the adult evolution.

This effect can be seen again in Figure 5(b), averaged over 100 repetitions of the experiments of Figure 5(a). Given the champion of the run, the graph shows the rolling average of the mutation distance between each of the successive generations of the champion's ancestors. The rolling average of the mutation distance is high during the developmental period and drops as soon as development ends, and therefore as soon as the fitness landscape stops changing. This increase is correlated to the magnitude of development. The developmental muscle model with starting size 0.5 (Figure 5(b-1)) has a higher rise in mutation distance than the one with starting size 0.9 (5(b-2)). For some of the most successful developmental evolutions, such as the one with starting mass 0.125 (5(b-3)), a significantly higher rate of mutation distance is maintained over the adult evolution for a long time after the stabilization into adult morphology. Conversely, nonbeneficial development, such as the low starting stiffness 0.49 (5(b-4)), still displays high mutation distance during development but quickly drops to adult evolution levels when development ends.

Another way to illustrate this is via Figure 5(c). From the same initial genotype, 100 runs of the adult evolution and the developmental model with starting size 0.5 were computed. Looking at a PCA of the genotype of the champion of each run, the pattern is clear: The adult evolution clusters around the initial search point, while the developmental runs went far away from the initial population, and all in the same general direction.

While development is indeed a way to prevent (premature) convergence by pushing the evolutionary search to move around a constantly evolving fitness landscape, it would not be useful if the location it arrived at by generation 2,000—at the end of development—was not any better than those randomly provided by the initial population or learned by adult evolution. The fitness results of Section 3.1 show that development can indeed be beneficial, but Figure 5(b-4) (starting stiffness 0.49) and the corresponding performance in Figure 4(b) equally demonstrate that moving around during development does not guarantee better performance. Still, when looking systematically at the relationship between exploration and fitness, some patterns are present.

It turns out that genealogy distance and fitness are correlated *within* a given developmental condition, as shown in Figure 5(d). Figure 5(d-1) exemplifies the correlation that can be found between fitness and genealogy distance, whether for the adult evolution or the developmental muscle model. Note that we do not compare developmental and adult evolution; the correlation is *within* the 100 runs of a given condition. The runs that travel more throughout the search space for a given
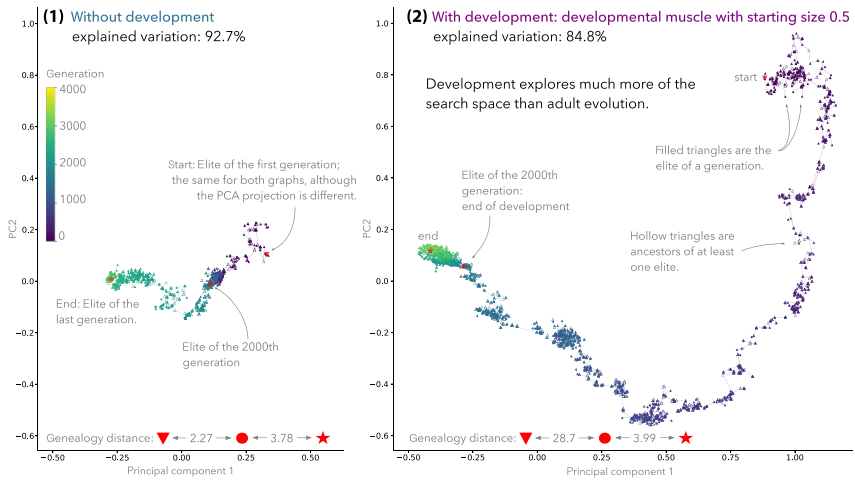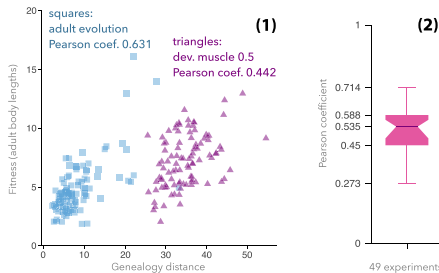
(a) Principal component analysis of the evolutionary search
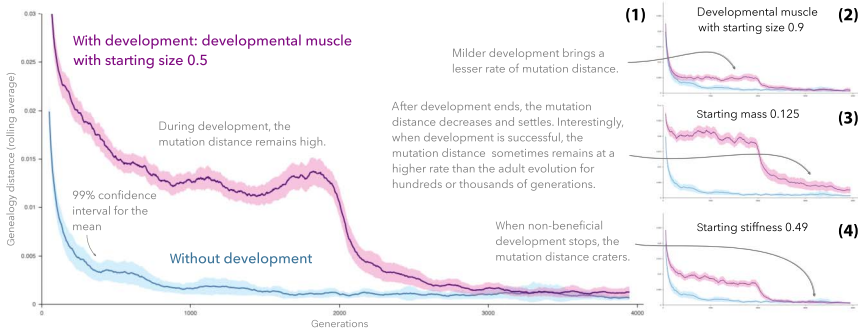
(b) Rolling average of the mutation distance
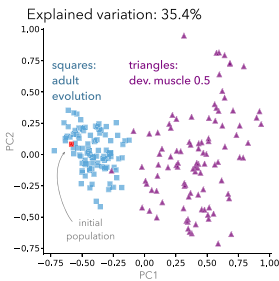
(c) PCA of champions of 100 runs

(d) Fitness and genealogy distance correlation

Figure 5. Development has a notable impact on exploration. (a) Principal component analysis (PCA) of the genotype vectors of two different runs: one with a developmental muscle model with a starting size 0.5 (a-2), and one with the adult evolution (a-1). The starting population was precomputed and the same in both cases, and was composed of only one member (red inverted triangle), selected as the one out of 15 random members with the best fitness on the adult morphology. The genealogy shows the member with the best fitness of each generation (filled triangles), alongside all their ancestors (unfilled triangles). Edges show parental relationships. The PCA was computed separately for each run, on all the depicted genotypes. (b) Rolling average with a 101-generation window of the mutation distance of each generation, for four different developmental cases: two with the muscle model, one with mass development only, and one with stiffness development only. (c) Given the same initial population, with only one member, PCA of the genotype of the champion of each run, for the adult evolution (squares) and the developmental muscle with starting size 0.5. (d) Genealogy distance correlates positively with fitness. In (d-1), the distribution of fitness in function of the genealogy distance of the elite at the end of the evolutionary search is displayed. The correlations are computed separately for control and the developmental muscle case with starting size 0.5. (d-2). Distribution of the fitness/genealogy distance correlation for all the experiments of Figure 3(c) (adult evolution + 12 starting sizes) and Figure 4 (12 conditions each for starting size, stiffness, and mass), all with a significant correlation ($p < 0.01$).

developmental schedule (or the adult evolution) tend to achieve higher fitness. Figure 5(d-2) explores how a different amount of exploration of the search space from one developmental condition (i.e., one experiment) to the next correlates with higher fitness. Across all experiments of Figure 3(c) and Figure 4 (49 experiments of 100 runs each), we correlate the fitness of the best individual across 100 runs (chosen within the last 50 generations) to its genealogy distance to its ancestor in the initial population. The median Pearson correlation coefficient is 0.535: Developmental conditions that travel more through the search space tend to generate better performance. This correlation is present within runs (Figure 5(d-1)) and across experiments (Figure 5(d-2)).

These results do not provide a conclusive answer to why developmental evolution improves the effectiveness of the non-rolling behaviors, but it strongly suggests one: a better exploration of the search space.

3.3 Population Size, Development Length, and Fast Development

Developmental evolution is effective for increasing fitness but requires going through the developmental period, where the behaviors are not learned—and therefore are a priori not effective—on the target adult morphology. One question is then: How fast can development occur? If we measure the length of development by the number of fitness evaluations spent on a morphology that is not the adult one, two ways to reduce it are straightforward: reduce the size of the population or reduce the number of developmental generations.

Figure 6(a) looks at the effect of population size on the impact of development. We compare the fitness distributions of different population sizes: 2 (1 parent/1 child), 3 (1/2), 5 (2/3), 7 (2/5), 13 (3/10), 20 (5/15, the basis for all other results), 40 (10/30), 60 (15/45), and 120 (30/90). As we can observe, development brings significant improvement at every population size studied, even for the extreme case of one parent/one child. And in the range of sizes 2–13, the fitness of developmental evolution grows faster than the fitness of adult evolution as the size of the population grows. Development both is robust to small population sizes and takes advantage of population increase. This is partly explained by development not acting *within* the population, as would a diversity-preserving algorithm, for instance, but *on* it, uniformly.

Figure 6(b) analyzes how the length of development affects the final fitness. Even a short developmental phase, 100 generations, brings significant benefits, which then increase as development goes on longer. The slight decrease for development length 3,000 and 4,000 can be explained by the evolutionary search ending at generation 4,000 in all cases: The settling period into the adult morphology is shortened for development length 3,000 and non-existent for 4,000. Development is therefore effective even when short—but not too short. Both long development and the presence of an extended adult phase are beneficial.

When trying to reduce as much as possible the development phase, we can take advantage of the dynamics of those two dimensions above: development length and population size. After trying different combinations of population size and development lengths, we found that a population of seven—two parents and five children—and a development length of 500 generations offers a good balance between developmental speed and fitness benefits, with a total of 3,500 fitness evaluations during development. Compared to the adult evolution with 20 members, after 7,000 fitness evaluations, i.e., 350 generations of the adult evolution, the fitness is significantly better (it is also significantly better than an adult evolution with seven members per generation; see Figure S6 in the online supplement). Interestingly, at 350 generations, the adult evolution has just started slowing down fitness improvement. So implementing development here provides few drawbacks in terms of time spent in the search.

3.4 Comparisons with Evo-Devo Approaches

3.4.1 2D Robots

Our approach in this work is to explore a biologically implausible modification of interaction between evolution and development. One of the main differences between our approach and an

(a) Effect of population size on development

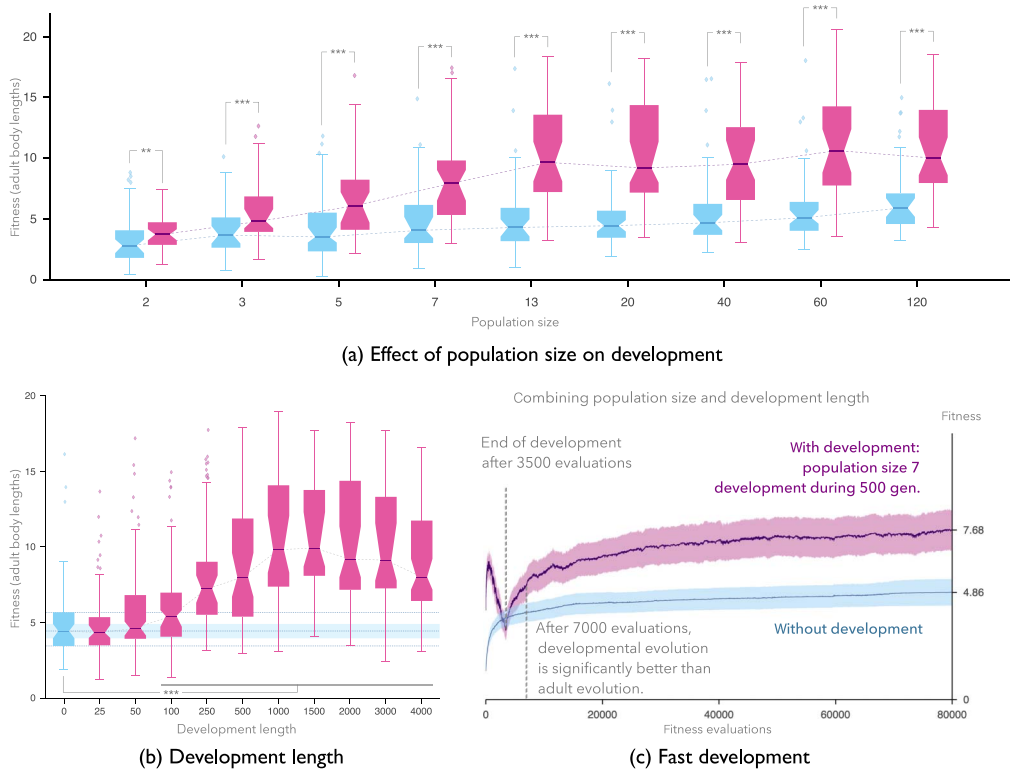(b) Development length

(c) Fast development

Figure 6. Development is effective at a low and high population size, and over short developmental periods. (a) Comparisons of the final fitness between the adult evolution and development with starting mass 0.125, for different population sizes. The number of generations is kept at 4,000 regardless of population size. (b) Final fitness observed for different development lengths of the development with starting mass 0.125, from 0 (adult evolution) to 4,000 (only development). The total number of generations is always 4,000, with the robots spending the remaining generations evolving with the adult morphology after development ends. (c) Average fitness over 100 runs. The adult evolution (with 20 members per generation) is contrasted with development with starting mass 0.125, with a population of 7 members (2 parents, 5 children) and a development length of 500. The *x* axis is the number of fitness evaluations to account for the different population sizes.

evo-devo approach is the relative position of the adaptation mechanism with regards to development. In our approach, development happens at the same timescale as adaption through evolutionary search. In an evo-devo approach, the development happens at a shorter timescale, during the evaluation period, with no adaptation happening. We tested the impact of an evo-devo approach on our robots.

In the evo-devo experiment, the evolutionary search still spans 4,000 generations, with no developmental evolution happening. At the start of a simulation, the robots start with child body size, and settle on the ground for 9.42 s. At 10 s, shortly after they started to actuate, they start to grow linearly in size, to reach adult size at the 40 s mark. They then continue to actuate with a fixed adult size until the 60 s mark. This developmental schedule never changes during the 4,000 generations. The population of behaviors of the last 50 generations of each run are then evaluated on a fixed adult morphology (adult robots during the whole 60 s simulation), and the resulting fitnesses are shown in Figure 7.

The evo-devo approach does not bring fitness increases over a pure adult evolution. Furthermore, it decreases fitness significantly for the robots that start much bigger than the adult size. This of course proves little, only that a devo-evo approach can work where an evo-devo one might not.
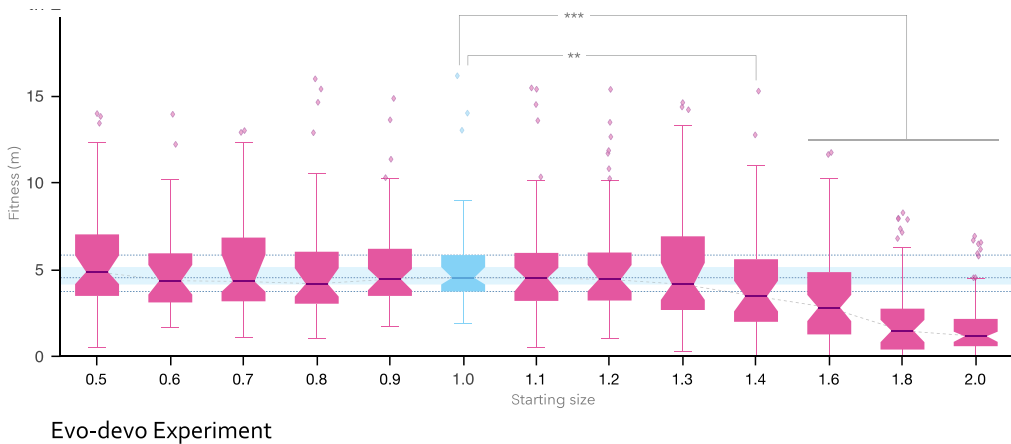
Figure 7. The evo-devo approach does not bring fitness benefits. The box plots represent the adult morphology performance after 4,000 generations of evo-devo evolutionary search. The blue box plot corresponds to the adult evolution, equivalent to an evo-devo starting size of 1.0.

3.4.2 3D Voxel Robots

In their 2018 article, Kriegman, Cheney, and Bongard propose an evolutionary development approach (*evo-devo*) where robots made of a 3D lattice of voxels (Hiller & Lipson, 2014) are allowed to develop during the evaluation period. On a locomotion task, the approach outperforms a regular evolutionary algorithm (*evo*), where no development happens. This task provides a good test bed to test our approach and how it generalizes to 3D environments and other robots.

For our approach—*devo-evo*—we modify the evo method by adding a mass-based developmental program on top of it. In the evo (and the evo-devo) approach, the voxels all have a constant mass of 1 g during the simulations; in devo-evo, we start the voxels of our robots, in the first generation, at 0.25 g, 25% of the adult mass. The mass then increases linearly with each generation, to reach 1 g at generation 2,000. The evolutionary search then continues until generation 10,000. We reproduced the results of the original article (Kriegman, Cheney, & Bongard, 2018, Figure 3.A) and compared them with the performance of the devo-evo approach.

The devo-evo approach achieves high fitness in early development (Figure 8) during the first 100 generations. The performance steadily falls as the robot mass increases. At generation 2,000, the robots have the same mass as in the other approaches, and the performance can be fairly compared with evo and evo-devo. The devo-evo approach exhibits significantly higher median performance than either of the other methods, and its final performance is also significantly higher than the other methods. Details about the method, additional results, and discussion are available in the Online Supplemental Material.

4 Discussion

Development—intrinsically—creates change. On its way to an adult form, the body goes through different iterations, each harboring the possibility to be particularly fit for a given purpose and to make some interesting behaviors easier to discover and perform. With this article, we harness this phenomenon in robots. But, rather than evolution having access only to the global fitness of the end-result of this developmental process, we make all the developmental steps directly accessible to evolution's selection capabilities.

This is illustrated nicely in this article by how difficult rolling behaviors can be to discover for 2D and 3D robots alike, and how a developing morphology creates easier access to them. Kriegman, Cheney, and Bongard (2018) noticed a similar effect when adding development during
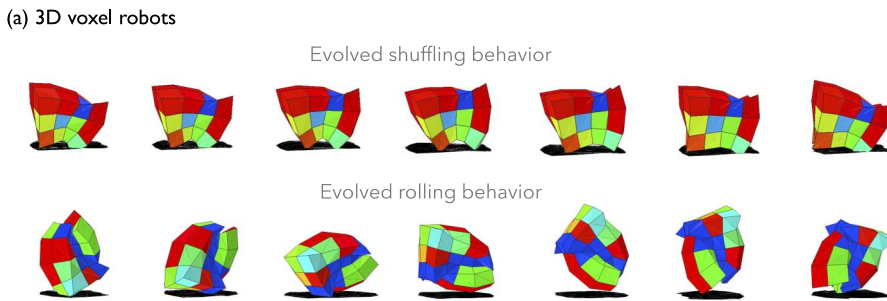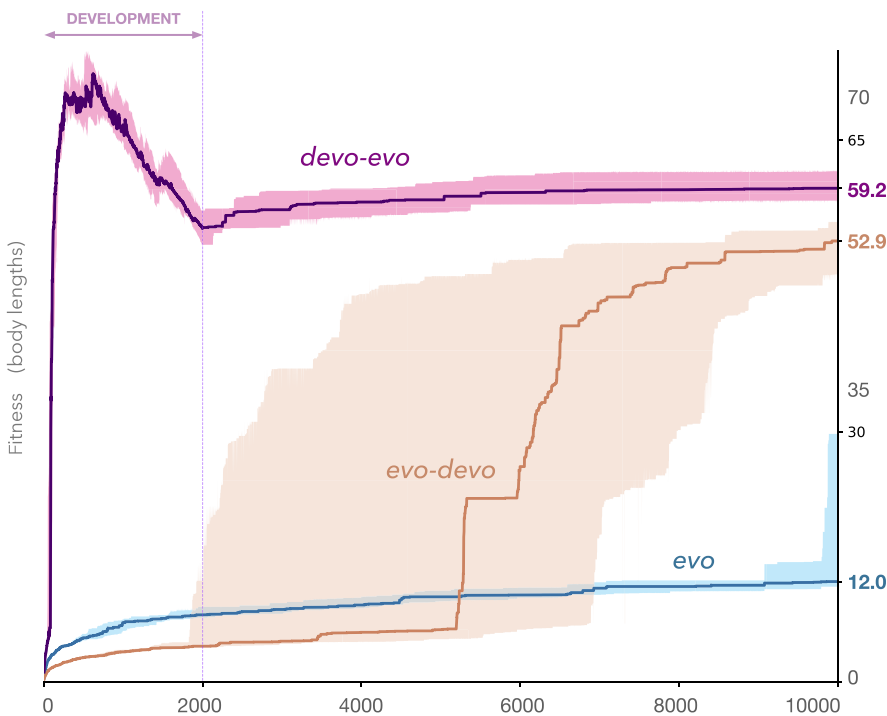
Figure 8. Devo-evo outperforms evo-devo. (a) A 4 × 4 × 3 lattice of voxels creates robots that can move by contracting the voxels. Each voxel contracts with the same sinusoidal signal, but each has its own phase. The resting size of the voxel is specific to each voxel as well, creating robots with different morphologies. Two evolved behaviors are shown: a shuffling gait at the top, and a rolling gait at the bottom. (b) Our devo-evo approach shows a typical developmental performance: The performance is high when the robots are light, and then decreases as the robots become heavier during the first 2,000 generations, to stabilize to high fitness that increases slowly after development ends. The graph shows the median performance of 30 runs, with the shaded area showing the 95% boostrapped confidence interval (20,000 resamplings), the same as in the original article by Kriegman, Cheney, and Bongard (2018).

fitness evaluation. Because in many real-world problems interesting behaviors represent a very small portion of the search space, sometimes without good gradients to help guide robots toward them, methods to help discover them are crucial for robots facing the complexity of the real world. In our work, we observed that once discovered through development, those high-fitness behaviors have a chance to be maintained by evolution into the adult form. Development is used here, quite straightforwardly, as a source of behavioral diversity that evolution can select from.

Many approaches in evolutionary robotics have aimed at fostering diversity. Some approaches place behavioral novelty directly under evolutionary control, using selective pressures (Doncieux &

Mouret, 2014; Risi et al., 2009), while others, like illumination algorithms (Cully et al., 2015; Mouret & Clune, 2015), modify the structure of the search to expose an ensemble of solutions across specific dimensions of behavioral diversity. Our devo-evo approach does not employ either of those mechanisms. Rather, morphological change is added on top of a classic evolutionary search, with no change to the evolutionary algorithm or the evaluation budget. The approach does not need memory archives to store elites or estimate novelty nor does it need to define novelty distance or behavioral dimensions along which a grid of MAP-Elites cells should be created. Rather, it relies on pure embodiment. It is worth considering here an extreme case: Even without adaptation capabilities, a robot executing the same motor activation and undergoing morphological development would produce behavioral diversity as its body grows (see Figure 2(a) for an example). Irrespective of the controller of the robot, morphological change produces behavioral change.

Morphological development also creates a constant source of perturbation, an imbalance in the evolutionary search that straightforwardly deals with two fundamental issues of evolutionary computation: the bootstrap problem and premature convergence (Bongard & Hornby, 2010; Eiben, 2015; Hornby, 2006; Mouret & Doncieux, 2009; Schmidt & Lipson, 2010). Premature convergence is prevented because the early ongoing morphological changes modify the fitness landscape with every generation and prevent the evolutionary search from settling too early: You cannot stay in a fitness landscape's valley if the valley disappears under you. The bootstrap problem—which happens when all the members of the initial population obtain minimal fitness, and hence stall the search—is mitigated by developmental evolution, since it will, throughout development, evaluate the population on a succession of slightly different fitness landscapes. If one of those fitness landscapes creates interesting behaviors, the evolutionary search can start. In this work, we have seen that even when the robots start with eight times the adult mass, and therefore can hardly move from the starting position (Figure S4.C in the online supplement)—a good setup for a bootstrapping problem—the fitness performance at the end of the evolutionary search is not statistically distinguishable from that of the adult evolution (Figure 4(c)). That developmental evolution was resilient to all the developmental paths we explored was an unexpected observation.

Another surprising result was that evolution was able to benefit from developmental paths that made the task intuitively harder (e.g., a smaller starting size). Making a robot stronger or lighter is often an effective way to boost performance. So it came without much surprise that the developmental paths that started that way (high muscle strength or low starting mass), and hence made the task initially easier, led to increased adult performance. This is a typical incremental evolutionary strategy. The results of the lower starting size experiments are harder to explain. It is indeed tempting to intuitively conclude that physical immaturity is a hindrance to skill acquisition only made necessary in animals by physiological constraints. This line of thinking, whether implicit or explicit, combined with considerations of practicality, robustness, and feasibility, has been embraced by robotics: Robots are born adult. This study, however modest its results, suggests that robots that grow up in size can bring qualitative performance benefits.

And the reason may lie in how developmental evolution affects exploration. The creation of behavioral diversity and the changing fitness landscape preventing early convergence both respectively facilitate and stimulate exploration. In the analysis, we have shown that increased exploration (characterized by increased genotyped distance between the first and last population) correlates positively with fitness within a developmental condition. The lower starting size experiments show this in another way: The smaller you start, the higher the adult performance. Starting smaller means a faster pace of change during development, and more morphologies explored.

Our study reaches a different conclusion than Bongard's (2011a) article. There are a lot of differences—robot (rigid), task (phototaxis), fitness criterion (time to success), many aspects of the method—that make a straightforward comparison difficult. One issue that Bongard reported was abrupt body changes (the evolutionary process goes through four discrete morphologies, rather than developing gradually), leading to a drop in fitness, in some cases to zero. This may explain why performance was affected negatively. Our developmental paths change morphology slowly, and this may make them robust to this. Future work is needed to investigate those issues in detail.

We have explored only a few developmental programs in this article. It may be that, just as humans are not necessarily good at hand-tuning good controllers for complex body shapes, the design of developmental programs as they become more complex and dynamic are beyond the skills of human designers. We may then let developmental programs evolve themselves, which may raise issues in how expensive that might be in computing resources. Nevertheless, the commonalities seen in the biological world in the development of species with widely different morphologies teases the possible discovery of general developmental principles and mechanisms that could be effectively applied to a wide range of robots.

As well as adapting to different robots, a developmental program would be useful if it can benefit a large range of tasks. One fundamental motivation for our approach is the idea that a well-designed development program for robots might adapt well to many different tasks and environments, much like what we observe in humans.

5 Conclusion

Developmental evolution gradually changes the morphology of populations of robots during an evolutionary search. In this study, we have seen that spending evolutionary time on morphologies different from the target one, potentially smaller or weaker ones, may bring significant improvement to fitness. This is achieved in two ways: by making some efficient behaviors easier to discover, and by preventing premature convergence through a continuous modification of the search space, therefore fostering its exploration. Because morphological development is intrinsic to the robot, its implementation does not depend on the task, environment, or fitness function. It has been shown to be robust to many different developmental paths, to work with small and large population sizes, on short and long developmental lengths, and with 2D and 3D robots. Interesting parallels exist between developmental evolution and developmental learning, and inspirations and techniques from both could lead to fruitful future work.

References

Anderson, M. L. (2003). Embodied cognition: A field guide. *Artificial Intelligence*, *149*(1), 91–130. https://doi.org/10.1016/s0004-3702(03)00054-7

Bongard, J. (C.) (2011a). Morphological change in machines accelerates the evolution of robust behavior. *Proceedings of the National Academy of Sciences*, *108*(4), 1234–1239. https://doi.org/10.1073/pnas.1015390108, PubMed: 21220304

Bongard, J. C. (2011b). Morphological and environmental scaffolding synergize when evolving robot controllers. In N. Krasnogor (Ed.), *GECCO '11: Proceedings of the 13th annual conference on genetic and evolutionary computation* (pp. 179–186). ACM. https://doi.org/10.1145/2001576.2001602

Bongard, J. C., & Hornby, G. S. (2010). Guarding against premature convergence while accelerating evolutionary search. In *GECCO '10: Proceedings of the 12th annual conference on genetic and evolutionary computation* (pp. 111–118). ACM. https://doi.org/10.1145/1830483.1830504

Cangelosi, A. (2015). *Developmental robotics: From babies to robots*. MIT Press.

Catto, E. (2011, March 21–25). *Soft constraints: Reinventing the spring* [Paper presentation]. Game Developer Conference, San Francisco, CA, United States. https://box2d.org/files/ErinCatto_SoftConstraints_GDC2011.pdf

Colby, M., & Tumer, K. (2015). Fitness function shaping in multiagent cooperative coevolutionary algorithms. *Autonomous Agents and Multi-Agent Systems*, *31*(2), 179–206. https://doi.org/10.1007/s10458-015-9318-0

Corucci, F. (2016). Evolutionary developmental soft robotics: Towards adaptive and intelligent soft machines following nature's approach to design. In C. Laschi, J. Rossiter, F. Iida, M. Cianchetti, & L. Margheri (Eds.), *Soft robotics: Trends, applications and challenges* (pp. 111–116). Springer. https://doi.org/10.1007/978-3-319-46460-2_14

Corucci, F., Cheney, N., Kriegman, S., Bongard, J., & Laschi, C. (2017). Evolutionary developmental soft robotics as a framework to study intelligence and adaptive behavior in animals and plants. *Frontiers in Robotics and AI*, *4*, Article 34. https://doi.org/10.3389/frobt.2017.00034

Cully, A., Clune, J., Tarapore, D., & Mouret, J.-B. (2015). Robots that can adapt like animals. *Nature*, *521*(7553), 503–507. https://doi.org/10.1038/nature14422, PubMed: 26017452

Doncieux, S., & Mouret, J.-B. (2014). Beyond black-box optimization: A review of selective pressures for evolutionary robotics. *Evolutionary Intelligence*, *7*(2), 71–93. https://doi.org/10.1007/s12065-014-0110-x

Dottore, E. D., Sadeghi, A., Mondini, A., Mattoli, V., & Mazzolai, B. (2018). Toward growing robots: A historical evolution from cellular to plant-inspired robotics. *Frontiers in Robotics and AI*, *5*, Article 16. https://doi.org/10.3389/frobt.2018.00016, PubMed: 33500903

Eiben, A. (2015). *Introduction to evolutionary computing*. Springer.

Gomez, F., & Miikkulainen, R. (1997). Incremental evolution of complex general behavior. *Adaptive Behavior*, *5*(3–4), 317–342. https://doi.org/10.1177/105971239700500305

Hiller, J., & Lipson, H. (2014). Dynamic simulation of soft multimaterial 3D-printed objects. *Soft Robotics*, *1*(1), 88–101. https://doi.org/10.1089/soro.2013.0010

Hornby, G. S. (2006). Alps: The age-layered population structure for reducing the problem of premature convergence. In *GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation* (pp. 815–822). ACM. https://doi.org/10.1145/1143997.1144142

Kriegman, S., Cheney, N., & Bongard, J. (2018). How morphological development can guide evolution. *Scientific Reports*, *8*(1), Article 13934. https://doi.org/10.1038/s41598-018-31868-7, PubMed: 30224743

Kriegman, S., Cheney, N., Corucci, F., & Bongard, J. C. (2017). A minimal developmental model can increase evolvability in soft robots. In *GECCO '17: Proceedings of the genetic and evolutionary computation conference* (pp. 131–138). ACM. https://doi.org/10.1145/3071178.3071296

Kriegman, S., Cheney, N., Corucci, F., & Bongard, J. C. (2018). Interoceptive robustness through environment-mediated morphological development. In H. Aguirre (Ed.), *GECCO '18: Proceedings of the genetic and evolutionary computation conference* (pp. 109–116). ACM. https://doi.org/10.1145/3205455.3205529

Lungarella, M., Metta, G., Pfeifer, R., & Sandini, G. (2003). Developmental robotics: A survey. *Connection Science*, *15*(4), 151–190. https://doi.org/10.1080/09540090310001655110

Miras, K., & Eiben, A. E. (2019a). Effects of environmental conditions on evolved robot morphologies and behavior. In M. López-Ibáñez (Ed.), *GECCO '19: Proceedings of the genetic and evolutionary computation conference* (pp. 125–132). ACM. https://doi.org/10.1145/3321707.3321811

Miras, K., & Eiben, A. E. (2019b). The impact of environmental history on evolved robot properties. In *ALIFE 2019: The 2019 conference on Artificial Life* (pp. 396–403). MIT Press. https://doi.org/10.1162/isal_a_00192

Mouret, J.-B., & Clune, J. (2015). *Illuminating search spaces by mapping elites*. ArXiv. http://arxiv.org/abs/1504.04909v1

Mouret, J.-B., & Doncieux, S. (2008). Incremental evolution of animats' behaviors as a multi-objective optimization. In M. Asada, J. C. T. Hallam, J. A. Meyer, & J. Tani (Eds.), *SAB 2008: From animals to animats 10* (pp. 210–219). Springer. https://doi.org/10.1007/978-3-540-69134-1_21

Mouret, J.-B., & Doncieux, S. (2009). Overcoming the bootstrap problem in evolutionary robotics using behavioral diversity. In *2009 IEEE congress on evolutionary computation* (pp. 1161–1168). IEEE. https://doi.org/10.1109/cec.2009.4983077

Mouret, J.-B., Doncieux, S., & Meyer, J.-A. (2006). Incremental evolution of target-following neuro-controllers for flapping-wing animats. In S. Nolfi, G. Baldassarre, R. Calabretta, J. T. Hallam, & D. Marocco (Eds.), *SAB'06: Proceedings of the 9th international conference on simulation of adaptive behavior: From animals to animats* (pp. 606–618). Springer. https://doi.org/10.1007/11840541_50

Nolfi, S. (1997). Evolving non-trivial behaviors on real robots: A garbage collecting robot. *Robotics and Autonomous Systems*, *22*(3–4), 187–198. https://doi.org/10.1016/s0921-8890(97)00038-9

Parker, G. B. (2001). The incremental evolution of gaits for hexapod robots. In *GECCO '01: Proceedings of the 3rd annual conference on genetic and evolutionary computation* (pp. 1114–1121). Morgan Kaufmann. https://dl.acm.org/doi/10.5555/2955239.2955440

Pfeifer, R., & Gómez, G. (2009). Morphological computation: Connecting brain, body, and environment. In B. Sendhoff, E. Körner, O. Sporns, H. Ritter, & K. Doya (Eds.), *Creating brain-like intelligence* (pp. 66–83). Springer. https://doi.org/10.1007/978-3-642-00616-6_5

Risi, S., Vanderbleek, S. D., Hughes, C. E., & Stanley, K. O. (2009). How novelty search escapes the deceptive trap of learning to learn. In *GECCO '09: Proceedings of the 11th annual conference on genetic and evolutionary computation* (pp. 153–160). ACM. https://doi.org/10.1145/1569901.1569923

Sacks, R. D., & Roy, R. R. (1982). Architecture of the hind limb muscles of cats: Functional significance. *Journal of Morphology*, *173*(2), 185–195. https://doi.org/10.1002/jmor.1051730206, PubMed: 7120421

Schmidt, M. D., & Lipson, H. (2010). Age-fitness pareto optimization. In *GECCO '10: Proceedings of the 12th annual conference on genetic and evolutionary computation* (pp. 543–544). ACM. https://doi.org/10.1145/1830483.1830584

Shapiro, L. (2011). *Embodied cognition*. Routledge.

Smith, L., & Gasser, M. (2005). The development of embodied cognition: Six lessons from babies. *Artificial Life*, *11*(1–2), 13–29. https://doi.org/10.1162/1064546053278973, PubMed: 15811218

Urzelai, J., & Floreano, D. (1999). Incremental evolution with minimal resources. In U. Rueckert & F. Mondada (Eds.), *First international Khepera workshop (IKW'1999)*. http://infoscience.epfl.ch/record/63903

Urzelai, J., Floreano, D., Dorigo, M., & Colombetti, M. (1998). Incremental robot shaping. *Connection Science*, *10*(3–4), 341–360. https://doi.org/10.1080/095400998116486

Vujovic, V., Rosendo, A., Brodbeck, L., & Iida, F. (2017). Evolutionary developmental robotics: Improving morphology and control of physical robots. *Artificial Life*, *23*(2), 169–185. https://doi.org/10.1162/artl_a_00228, PubMed: 28513207