

Modeling Evolutionary Development with Indirect Encodings on Dynamic NK Fitness Landscapes

Jacob Ashworth^{1*}, Julian Fiorito^{1*} and Jason Yoder¹

¹Rose-Hulman Institute of Technology, Terre Haute, IN 47803
ashworjs@rose-hulman.edu, fioritjx@rose-hulman.edu

*These authors contributed equally to this work

Abstract

The study of evolutionary development (evo-devo) is frequently challenged by the scales of space and time complexity inherent to its study. This has led to the creation of abstract models to allow for the exploration of evo-devo in a manner that is both more computationally feasible and more general, without ties to the specific biological processes of a single organism. Our work expands upon these previous models by introducing an indirect encoding for developmental mechanisms, dynamic fitness landscapes, and a phenotypic structure that allows for the exploration of new interactions between the developmental and evolutionary processes. Introducing these changes allows us to conduct a more thorough study of factors impacting evo-devo. Our experimental results suggest a number of parallels to biological systems. These include representing the synergy of evolutionary and developmental processes, the evolution of adaptable features, and highly conserved regulatory genes. We also discuss the opportunities for exploration opened by this new model. These possibilities include the study of developmental exaptations and the robustness of developmental strategies.

Introduction

Development, evolution, and learning are indispensable to our current understanding of how populations are able to adapt to their environments. When exploring the interactions of such elements, development is often overlooked as a component of learning (Soltoggio et al., 2018), however, development fulfills a distinct role in adaptation. While evolution enables the adaptation of populations over generations and learning enables lifetime improvements based on experiences, development allows for the refinement of learning mechanisms and body plans to stages of the life cycle. As development serves to provide modulation of response dependent on age and other environmental factors, and modifies both the brain and body, treating it as nothing more than a component of learning may be an oversimplification.

The study of evo-devo is not only important for advancing scientific understanding, but also for advancing our technologies. Our understanding of learning, evolution, and their interactions has led to many innovations outside of biology. The principles of evolution have been effectively leveraged

in biologically inspired technologies, such as the antennae for NASA's Space Technology 5 spacecraft (Hornby et al., 2011). Learning-inspired algorithms are being used for many applications, from recommendation systems (Portugal et al., 2018) to medical image analysis (Shen et al., 2017). Furthermore, the interaction between learning and evolution has been effectively leveraged in research such as that of Khadka and Tumer (2018). However, there is a notable lack of technologies utilizing development. Improving our understanding of the interactions between evolution and development may enable such advances.

The study of evo-devo has traditionally been very difficult due to the time scales of evolution and the nature of development making it difficult to observe given our incomplete records. Thus, much of the current research relies upon modeling such interactions. Many models are structured around specific species, however, these models are difficult to generalize. This has led to the formation of more abstract models which avoid species-dependent factors. In our previous work (Ashworth et al., 2022), we designed one such model based on Kaufman and Levin's tunably rugged NK Fitness Landscape (NKFL). This previous model has several limitations which we built a new model to address. Under the old model, the developmental strategy is directly encoded, which only allows evolution to make small tweaks to the current strategy. The model presented in this paper uses a different set of developmental actions and an indirect encoding of the genotype to bypass this limitation. Another restriction of the previous model was how each trait in the phenotype (i.e., each bit in the bitstring) was developed without inheritance, which restricts the scope of evo-devo interactions that can be studied. Our introduction of different types of phenotypic relations addresses this issue. A final expansion upon the original model is the inclusion of experiments using both static and dynamic NK fitness landscapes, whereas the previous model dealt with only static landscapes (See Fig. 1).

Related Work

There have been many different approaches taken to model evolutionary development, some have been inspired by spe-

cific organisms, while others are more abstract and widely applicable. Some of the more biologically realistic evo-devo models in recent years have been based on model organisms such as *Caenorhabditis elegans* and *Drosophila melanogaster*, which were directly inspired by the known biological facts of these specific organisms (Frézal and Félix, 2015; Marco et al., 2003). Although meaningful progress has been made in this area, species such as *Nasonia vitripennis* have shown that evo-devo models of more complex organisms will include significant computational difficulties due to factors like gene knockdown (Sommer, 2009). This has led to attempts to build more abstract models (Frankenhuis and Walasek, 2020; Akhshabi et al., 2014) which trade some specificity to more directly study the process of evolutionary development. They can do this without being challenged by species-specific details or the complexity of computational biology in scales of time and space.

Some of these abstract models utilize the tunably rugged NKFL, originally popularized by Kauffman and Levin (1987). The NKFL model uses the parameter N to represent the length of a bitstring which represents the phenotype as in Todd et al. (2020). It uses the parameter K to regulate the number of interactions each bit has with its neighbors. Higher K values mean the landscape has more interdependencies, leading to more local optima, which make the landscapes harder to solve. The NKFL models the interdependence of phenotypic traits such that there is a well-defined concept of nearest neighbors. This has led to extensive use of NK landscapes for modeling adaptive walks and evolutionary development (Pitzer and Affenzeller, 2012; Ashworth et al., 2022). The NK model can be extended by various methods to form dynamic fitness landscapes, which change over a number of generations referred to as a *change cycle*. This allows for the exploration of how environments changing over evolutionary time impact the interactions of evolution and development. These methods add opportunities for new avenues of exploration without adding unneeded complexity. The dynamics of these fitness landscapes can be tuned to model specific types of change, and are parameterized to allow for adjustment of their intensity (Tinós and Yang, 2014).

Recently, Todd et al. (2020) developed a model using the NKFL to examine the tradeoffs between the duration of lifetime learning and evolution. This model led to interesting conclusions about the interplay between lifetime learning and both Darwinian and Lamarckian inheritance. Their results indicated that, with enough time being spent on lifetime learning, populations that experienced both learning and evolution events tended to have greater fitness than those groups which experienced only one or the other. However, this model made no attempt to explore the interactions between development and evolution.

Expanding upon this work, our previous model Ashworth et al. (2022) investigated how evolved developmental strategies with planned randomness could impact lifetime learning

rates. By changing the difficulty, or K -value, of the given landscape, we were able to demonstrate that evolved developmental strategies make use of randomness at later points in an agent's lifetime when the landscape exhibits more local optima. Though this model provides a helpful framework for the study of evo-devo in an abstract manner, its study is limited by several different factors. The core focus of this paper and model is increasing the number of ways in which the evolutionary and developmental processes can interact. The three key features of the new model that achieve this are the evolution of *initial traits*, an indirect encoding of development via the *developmental program*, and different trait interactions through *phenotypic regions*. We also introduce dynamic NKFLs into the model to study the evo-devo process on non-static problems. These expansions are discussed in greater detail in the following section (See Fig. 1).

Methodology

In order to capture the process of evolutionary development, our abstract model uses evolving populations of developing agents that are tasked with finding the global optima of an NK fitness landscape. The overall structure of the model is laid out in Fig. 1. The phenotype is represented by a bitstring, the fitness of which can be evaluated at any time by the landscape. The initial configuration of this phenotype is determined by the *initial traits* of the agent, part of its genotype. The traits are split up into four different regions which determine inheritance patterns, discussed further in the section on phenotypic regions. After the *initial traits* determine the initial configuration of the phenotype, the agent begins executing its *developmental strategy*, which changes its phenotype and thus its fitness over time. Once the *developmental strategy* is finished, the agent's life is over and its final fitness is recorded. The evolutionary process then uses the final fitnesses of a generation of agents to make a new generation, where the cycle repeats (See Fig. 1).

The interaction between the evolutionary and developmental processes is how this model simulates evolutionary development. The *developmental strategy* of the agents is modified through mutation in the evolutionary process, and the final fitnesses that drive evolutionary selection are greatly influenced by the *developmental strategy*. Similar ways of encapsulating the interaction between evolution and development have been used to study evo-devo in the past (Todd et al., 2020; Ashworth et al., 2022). We aim to expand upon the latter and introduce possibilities for different biological phenomena to be studied, such as developmental robustness or the occurrence of exaptations in the developmental process. For experimental purposes, we apply the abstract evo-devo model to an NK fitness landscape with three specific step types, though the specific fitness landscape and developmental steps can easily be changed while still keeping the structure of the abstract model intact as a faithful representation of evo-devo.

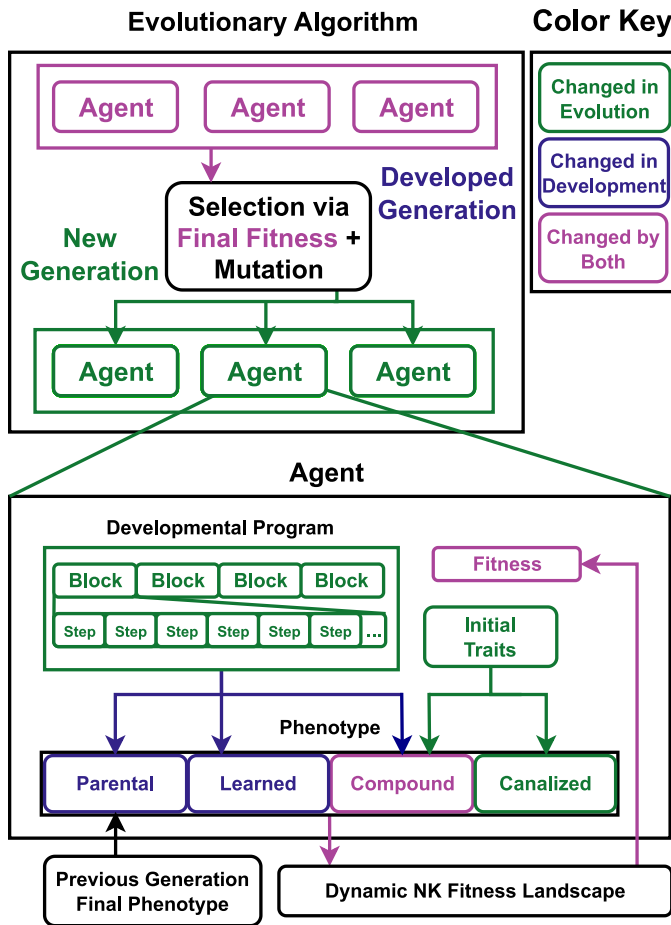


Figure 1: High-level overview of the components in our evolutionary development model. The three colors in this diagram represent during which process (evolution, development, or both) the data in different parts of the model are changed. Agents begin their life with a set of *initial traits*, a *developmental program*, and several *blocks*. These three components together make up the genotype of the agents. The *developmental strategy* is constructed using the *program* and *blocks* in a manner described by Fig. 2. During development, each step in the *developmental strategy* is executed, starting from the location determined by the *initial traits*. The location on the landscape is referred to as the phenotype, as it is the expression of the genotype. The steps of the *developmental strategy* each change one bit of the phenotype, making incremental changes over the agent’s lifetime. Once all steps in the *developmental program* are executed, the agent is considered *developed*, and we refer to the phenotype’s final state as the *final phenotype*. Evolution occurs on populations of *developed* agents. Evolution uses the final fitnesses of these agents (determined by the NKFL) to select those that are best-performing and mutates their *developmental programs*, *blocks*, and *initial traits* to create new generations of agents. Once a new generation is created, the cycle repeats.

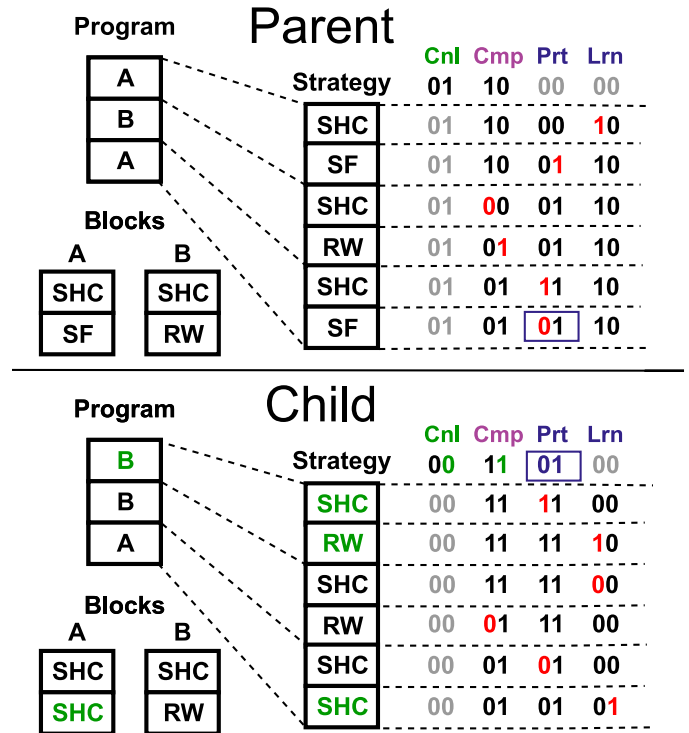


Figure 2: Simplified example of a parent and child *developmental program* and *developmental strategy* with *program* length three, *block* length two, and two total *blocks*. We abbreviate the four phenotypic regions (*Canalized*, *Compound*, *Parental*, *Learned*) as (Cnl/Cmp/Prt/Lrn) (See the phenotypic regions subsection for further explanation) and the three types of steps (Steepest Hill Climb, Random Walk, Steepest Fall) as (SHC/RW/SF). In both the parent and the child, the *developmental strategy* is constructed by concatenating copies of the *blocks* together in the order specified by the *developmental program*. The constructed *developmental strategy* is the developmental trajectory the individual will follow over their lifetime. The *initial traits* determine the starting state of the phenotype, and are the 8-bit string displayed directly beneath the phenotypic region labels. Two bits of the phenotype are assigned to each of the four phenotypic regions. Moving down through the steps of the *strategy*, iterative changes made to the phenotype are displayed in red. Each step of the *strategy* makes a single change to the phenotype in either the *compound*, *parental*, or *learned* portion. Evolutionary changes between the parent and child are displayed in green, while the child’s inheritance of its parent’s final phenotype is shown via the blue box. The different places where evolutionary changes occur are in the *blocks*, the *program*, and the *initial traits*. A change of a step in a *block* is called a *block mutation*, whereas a change in the *block* referenced by the *program* is called a *program mutation*.

Developmental Program

Our model uses an indirect encoding approach to the construction of the *developmental program*, described in Fig. 2. This indirect encoding takes the form of the *blocks* and *developmental program* which are used to define the *developmental strategy*, which is the list of steps to be executed. On the lower level of the *developmental program*, are several *blocks* which are short sequences of steps to be executed (akin to a finger in a body plan). The types of steps we use are steepest hill climbs (SHC), steepest falls (SF), and random walks (RW). We choose these three due to their simplicity, however, results generalized well with other types of developmental steps. Each of these steps (SHC, SF, RW) can evaluate all phenotypes within one bit flip from the current phenotype – these are called *neighbors*. SHC and SF select the *neighbors* with the highest and lowest fitness respectively, while RW selects a random *neighbor*. On the higher level of the *developmental program*, the *program* itself has an encoded ordering for which *blocks* are to be executed when (akin to instructions for a hand that is composed of five fingers). The *program* thus encodes which *blocks* are run and in which order. *Blocks* may be used more than once or not at all and in fact, the number of *blocks* can be different from the *program* length. When a *block* is not in use, it can be considered analogous to a non-coding gene.

Constructing our *developmental strategy* out of a *program* and *blocks* has many benefits. One benefit is that it handles the problem of an intractable search space created by our previous model from Ashworth et al. (2022). When the *developmental strategy* is encoded as a list of steps, the evolutionary search space grows exponentially as the length of the *strategy* increases. This greatly increases the time required for evolution to make progress. When the *developmental program* is introduced, the size of the evolutionary search space can be decreased without sacrificing *strategy* length by reducing the number of *blocks* available to the evolutionary algorithm. The desired *strategy* length can still be achieved, but since *blocks* will be reused several times by the *program*, the total evolutionary search space will shrink.

Another benefit of the *program* model is how *block mutations* and *program mutations* (see Fig. 2 for definition) allow the evolutionary algorithm to change the *developmental strategy* in different ways. A *program mutation* makes a potentially large local change in the *developmental strategy*, as all the steps executed at one part of the *program* are from an entirely different *block*. A *block mutation* on a *block* that is not referenced by the *developmental program* would have no impact on the *developmental strategy*, making it analogous to the modification of non-coding genes. However, if a *block mutation* occurs on a *block* that is referenced one or more times by the *program*, this would change a step executed for each reference of the *block*. These *block* and *program mutations* can be combined at specific rates to allow for the evolution of more performant *developmental strategies* across

a wide range of landscapes (see the developmental program subsection of the results).

Phenotypic Regions

We aim to address the inability of Ashworth et al. (2022) to explore the interactions between the evolution of the developmental pattern and the evolution of the *initial traits*. In addition, we seek to enable the model to explore the interaction of the development and evolution of independent traits. We treat the bits of the bitstring as different traits of the phenotype with various patterns of inheritance and development. We define the following as our types of traits: *canalized*, *compound*, *parental*, and *learned*. While an oversimplification of biological phenomena, this model enables us to explore phenomena that would otherwise be too difficult to extract from confounding factors.

In our model, *canalized* traits are those traits which are highly robust to environmental conditions, as with ABO blood types (Yamamoto, 1995), and so for the sake of simplicity are robust to developmental patterns and so do not change during the lifespan. These are simulated by inheritance of the parent's *initial traits* with some specified degree of generational mutation. The *compound* traits are those where the *initial traits* are inherited from that of the parent, but can still be changed by development. For example, some individuals may have genetics that predisposes them to be weaker than other individuals, however, they can still gain strength throughout their lifetime development. *Parental* traits are those which are inherited from the final phenotype of the parent, but can still develop over the child's lifespan. These traits parallel Lamarckian systems of inheritance, as the development of the parent effects both parent and offspring. In the real world, such effects can occur in a variety of ways, though a simple example could be parental effects (Crean and Bonduriansky, 2014). The *learned* traits are those that are robust to typical genetic variation, or which are otherwise primarily affected by developmental events. This is analogous to learning as learned traits often have too many contributing factors to be impacted directly by genetics and our model relies on hill climbing which is commonly used as an analogy for learning as in Todd et al. (2020). (2 demonstrates how the different regions are allowed to change.)

To incorporate these different trait types, the new model includes additional parameters to adjust the number of bits in the bitstring representations that are controlled in each manner. This is handled by producing 4 non-overlapping masks for the phenotype. These masks are then decomposed into masks controlling where inheritance from the *initial traits*, inheritance from the final phenotype, mutation, and development are able to be performed. Fig. 3 shows this process. These masks are bitstrings that contain additional binary information about the phenotype. In this case, the masks determine if specific types of inheritance or modification can occur. The masks must be randomized as the

Canalized: 00101000	From Initial: 01101111
Parental: 10010000	From Final: 10010000
Compound: 00000011	Mutated: 00101011
Learned: 01000100	Developed: 10010011

Figure 3: This figure shows how the ones of the non-overlapping masks representing each phenotypic trait (left) are decomposed into masks controlling what behavior each bit of the phenotype experiences (right). The ones are color-coded to show where each bit of the decomposed masks originates from. *Canalized* traits inherit from the parent’s *initial traits* and are mutated. *Parental* traits are inherited from the parent’s final phenotype without mutation, and can be modified by development throughout a lifetime. *Compound* traits are inherited from the parent’s *initial traits* with mutation, and are developed during the lifetime. *Learned* traits are inherited from the *initial traits* without mutation so the starting point never changes, but do undergo development.

NK Landscape (on which all dynamic landscapes are constructed) we use is local, meaning that traits are related to those immediately surrounding them in the bitstring. Thus, if the phenotypic traits were grouped consecutively on the mask, the interactions between different types of traits would be greatly reduced. These masks and their representative phenotypic regions, although an oversimplification, allow us to explore how such methods of inheritance and development impact the evolution of populations.

Dynamic Landscapes and Population Reruns

We wanted our model to be capable of modeling the evolution and development of agents on changing fitness landscapes. The goal was that this could provide some insight as to how the evolution of the *developmental programs* on such landscapes impacts their adaptability. Therefore, we designed our model to support dynamic fitness landscapes as described by Tinós and Yang (2014). To maintain consistency with the structure of NK landscapes we utilized three landscapes, *Xor*, *Even Odd*, and *Random*.

The *Xor* landscape applies a mask, which is modified each *change cycle* with a specified probability, to the phenotype using a Xor operation to change the observed fitness of the phenotype. This operation maintains neighborhood relations and is thus equivalent to randomly moving the *initial traits* of the agent (Tinós and Yang, 2014). This strategy may represent some change in the environment, resulting in changes to the utility of some traits.

The *Even Odd* landscape contains two NK-landscapes of equal N and K values, and swaps between the two every *change cycle*. This may be similar to insects and other organisms with life cycles shorter than seasonal changes. For such organisms, the phenotype required for success in one summer to the next is likely quite similar, however, it is radically

different from that which may be optimal in winter.

The final landscape was a *Random Landscape*, which generates a new random landscape each *change cycle* from some selection of K values. This models some environments which are rapidly changing and potentially getting more or less difficult for some organisms’ specific niches. An example of this might be a population living in an area that undergoes a natural disaster, resulting in a rapid change of environment. This dynamic landscape is less biologically feasible than the other dynamic landscapes as successive landscapes have very little relation to one another, however, it enables non-repeating landscapes over a range of K values. This is more valuable as a thought experiment which allows us to explore the extreme of what could be presented by a dynamic fitness landscape.

To test how a dynamic NKFL impacts the adaptability of their populations to new environments, we designed an experiment in which the final generation of one of the simulations was transferred to a variety of new NK landscapes. The maximum fitness that these populations are able to achieve when introduced to a new landscape indicates how well the *developmental programs* were able to adjust to new situations. We did notice that when fitness landscapes were changed, there was typically a spike in the diversity of *developmental programs*. As we only sample the best individual from any run, an increased level of diversity could lead to better performance through random chance rather than having more adaptable *developmental programs*. Thus, we further modified the experiments to remove this factor by only using the best-performing individual of the final generation.

Results

Program and Block Mutation Analysis

To understand the extent to which changes in the *developmental strategy’s* evolutionary process affect the final fitness of an organism, we created an experiment in which we varied the *strategy’s* mutation rates. The two mutation rates are the *block mutation* rate and the *program mutation* rate, as defined by Fig. 2. Fig. 4 shows a heatmap of the average final fitness of *strategies* evolved with varying mutation rates. The column where the *block mutation* rate is zero has been omitted, as its final fitness is a great decrease from those displayed, distorting the scale. Increasing the mutation rates beyond those displayed in our figure leads to a gradual decrease in final fitness, as the increased noise makes evolution increasingly difficult. As this is an analysis of the *developmental program*, the *initial traits* are not mutated during this experiment.

One important overall trend to notice is that it is optimal to use a combination of both *program* and *block* mutation rate. We can see that neither *program* nor *block* mutation rate is able to achieve optimal performance by itself. The best-performing *developmental strategies* were those that incorporated both mutation types. This optimal region ranges from *block mutation* rates of 1.5 – 3% and *program mutation* rates of 1 – 4%. We also notice that extending beyond this

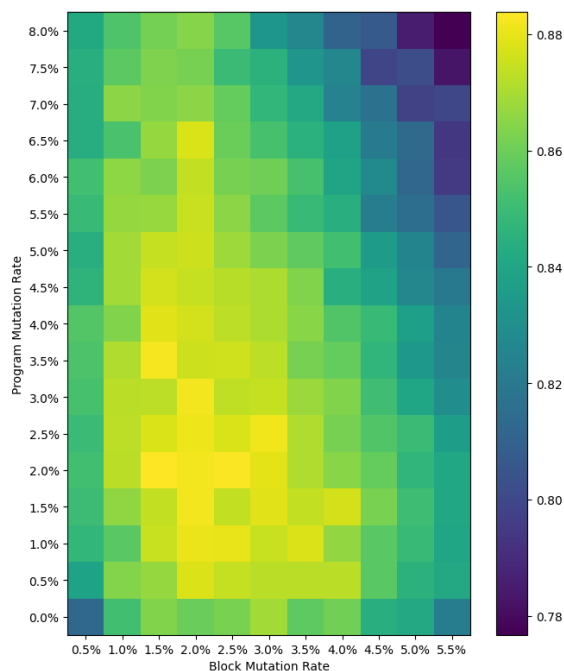


Figure 4: Heatmap of the mean final fitness of *developmental strategies* evolved on an $N=15$, $K=5$ static landscape. Evolution took place over 100 generations of 100 agents. *Program length* was five *blocks* long with *blocks* being ten steps each. Each mutation rate combination was run on 3000 different landscapes. Landscape fitnesses range from zero to one.

optimal region, increasing *program mutation* rate tends to have a less severe negative impact on final fitness than increasing *block mutation* rate. For instance, the row where the *program mutation* rate is 5% tends to achieve higher fitnesses than the column where the *block mutation* rate is 5%. This is especially obvious when analyzing the performance of *strategies* with an especially low *program mutation* rate (1.5% – 2%). In this scenario, increasing *program mutation* rate up to values even as high as 8% still achieves fitnesses in the 0.85-0.86 range. Increasing *block mutation* up to 8% in a similar manner, no matter the *program mutation* rate, achieves an average final fitness in the range of 0.78-0.8, a great decrease. This overall trend where higher *program mutation* rates are preferable to higher *block mutation* rates suggests ideas related to ‘teaching old genes new tricks’.

Relative Advantage of Phenotypic Regions

To explore how evolution and development can interact while affecting different traits, it was first necessary to establish a baseline of performance for each trait individually. The

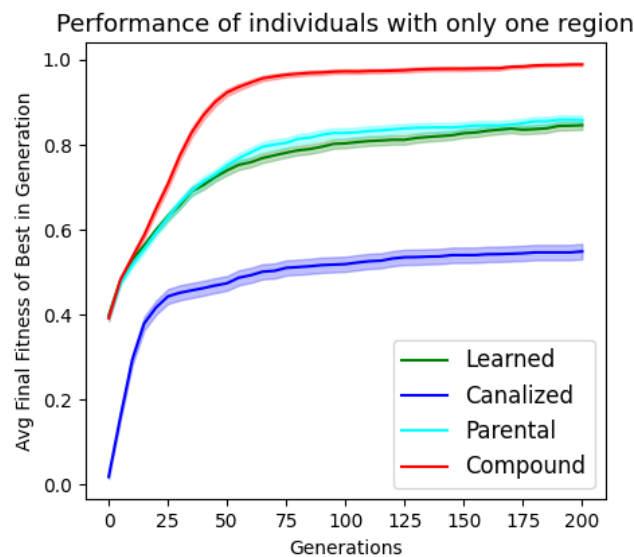


Figure 5: Performance of populations containing only one of each phenotypic region over the course of 200 generations on an $N=15$ $K=6$ static landscape. The population with only *canalized* phenotypic traits starts at a lower point than the others as it is unable to improve in fitness during a lifetime. The performance of populations with only *compound* traits is noticeably better than the others, which is consistent with the observations of Todd et al. (2020).

results of these tests can be seen in Fig. 5. It is clear that those individuals with an entirely *compound* phenotype outperform those with only *learned* or only *parental*. The worst performing individuals were those with an entirely *canalized* phenotype, which only changes over evolutionary time. These results are consistent with the results of the Todd et al. (2020) experiment, in which they varied the numbers of evolutionary and learning events. Our *canalized* traits are most similar to the evolution-only runs, the *parental* and *learned* traits are similar to the runs with only learning, while *compound* runs are close to the in-between ratios.

Using the new system of phenotypic regions, we also experimented with the quantities of *canalized* and *learned* traits in the phenotype. The results of this, shown in Fig. 6, demonstrate that even when the evolving and learning sites of the phenotype are separate, the synergy between the two processes is still able to occur.

Adaptability of Populations

To examine the adaptability of the developmental patterns from those populations which developed on different types of dynamic landscapes, we took the final generation of populations, with *learned* phenotypes, evolved on different types of landscapes and evaluated them on new landscapes with different K values. We found that changes in landscape

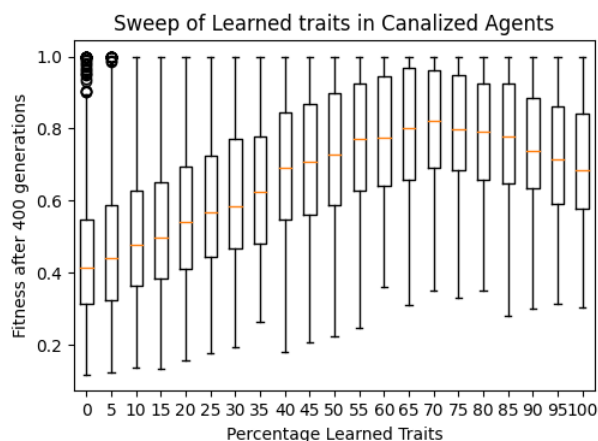


Figure 6: The distribution of fitnesses for best individuals after 400 generations on 200 runs. The percentage of *learned* phenotypic regions was varied in populations otherwise containing only *canalized* phenotypic regions. There is a noticeable peak at 70% phenotypic traits. This supports the concept of synergy between Darwinian inheritance of phenotypic traits and lifetime adaptations to the phenotype.

lead to increased diversity of developmental pattern in subsequent generations, to minimize the impacts of this the re-run generation was composed of clones of the best individual. The results of this experiment are shown in Fig. 7 and indicate that the best individuals evolved on dynamic landscapes were able to outperform those which evolved on static landscapes. Additionally, those dynamic landscapes which provided a greater diversity of local landscapes also demonstrated greater fitness values when introduced to new landscapes.

Discussion

One important high-level observation is how effective the evo-devo process as a whole is at solving NK landscapes. In situations where the evolution is run in an optimal configuration (*block* and *mutation* rates are tuned, and the *compound* phenotypic region comprises the entire phenotype) an agent that achieves the global fitness on a static $N=15$ $K=5$ landscape almost always emerges within the first 100 generations of evolution. This complicates the analysis of the evo-devo process because the period in which evolutionary progress occurs is limited. Once an agent is able to reliably achieve the globally optimal final fitness, the process stalls as it is impossible to improve upon such an agent. This prohibits the analysis of evo-devo properties that take a long time to manifest, so it becomes necessary to find ways to prolong the process. This can be accomplished is by increasing the N and K values significantly, but these come with drawbacks. Increasing the former leads to significant storage constraints as the landscape grows larger, and increasing the latter leads

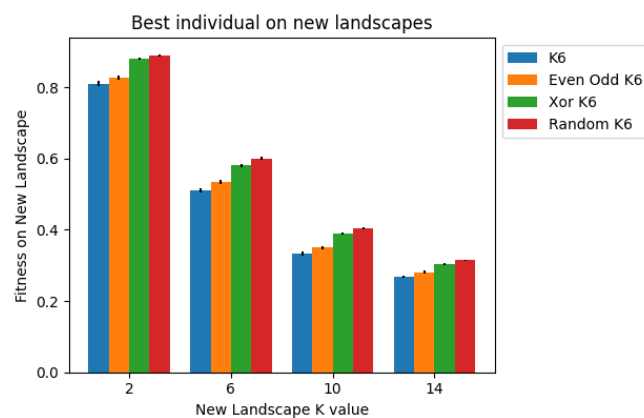


Figure 7: The average fitnesses of the best individuals after 400 generations on a variety of new $N=15$ landscapes. Each landscape on which evolution took place used exclusively $N=15$ $K=6$ landscapes as their basis, and the dynamic landscapes used a *change cycle* of 1 generation. Each bit of the *Xor* landscape's mask had a 10 percent chance of being flipped each *change cycle*.

to extremely chaotic landscapes that become impossible to explore. A more practical solution is provided by our introduction of dynamic landscapes - by having a changing landscape, we can ensure that the evo-devo process continually has ways to improve. This is a key role that the introduction of dynamic landscapes will play moving forward.

Adding phenotypic regions to the model allows an exploration of how the inheritance of traits impacts the evo-devo process. With our second experiment, in which all traits were of the same type (Fig. 5), we saw that populations with only *compound* traits were able to outperform the others. While it could be expected that *parental* traits would be superior, as the progress gained by development is not lost, the mutation of starting position present in *compound* traits enables escaping local optima. This result is consistent with the findings of Todd et al. (2020) who found that populations with some evolutionary events and some lifetime learning events outperformed those which only learned.

The third experiment using phenotypic regions explored the impact of adjusting the percentage of *learned* traits in an otherwise *canalized* organism. The results of this experiment, Fig. 6, indicate that a 70-30 split of traits was the most advantageous. Having some *canalized* traits is advantageous to the population, as it allows for random mutations between generations to move the *initial* traits out of the basins of local optima. However, as *canalized* traits are unable to develop, they are less beneficial than *learned* traits in general. The observed 70-30 split is near the point at which the advantage of escaping local optima is similar to the relative advantage of *learned* over *canalized* traits. This result is again consistent with Todd et al. (2020) in that systems using both

evolution and learning outperform those exclusively of either type. An interesting difference between the results is that their model varied a number of SHC between “evolutionary learning steps” while our experiment altered what percentage of the phenotype could be evolved. This difference shows that the synergy between learning and evolution is able to occur even when acting on different traits.

The results from running populations from different landscapes on novel landscapes indicate that developmental strategies evolved to allow for adaptability in highly dynamic landscapes. The *developmental strategy* was likely the reason for the adaptability, as all traits were *learned* and only the best individual was placed into the new environment. These two considerations minimize variation due to other factors, such as diversity in the population. These results support the theory put forward by Potts (2002) that evolution in changing environments encouraged the evolution of adaptable features, which allowed for the expansion of *Homo Sapiens* into many diverse environments.

Although our model is highly abstract, our results from exploring the rates of *program* and *block* mutations (see Fig. 4) show a parallel to the common description of evolution ‘teaching old genes new tricks’ (Carroll, 2012). Specifically, while it is important to have a minimum viable rate of each type of mutation, there is a trend that increasing the *program* mutation has a less negative influence on fitness compared with increasing the *block* mutation rate. This can be seen as consistent with findings such as the highly conserved regulatory genes which are re-used in arthropod segmentation (Fujimoto et al., 2008). In essence, the results demonstrate the ability to increasingly re-use and re-arrange particular *blocks* within a *developmental strategy* via higher *program mutation* rates.

Future Work

One intended area of further exploration is the concept of developmental exaptations. The idea of exaptations has been key in explaining the emergence of several different biological traits, such as the appearance of feathers on early birds or the antifreeze protein present in the antarctic family of notothenioid fish (Thanukos, 2009; Chen et al., 1997). It has been theorized that exaptations may play a role in how the developmental processes of individuals evolve over time (Chipman, 2021). This would manifest in some portion of a developmental process being duplicated, mutated, and then introduced into a new point in the organism’s developmental process to serve a different process than originally intended. Performing research in this area without an abstract model is extremely difficult as the sequence of events leading to the current developmental process of an organism is often unknown, so without an extremely long experimental timescale it becomes nearly impossible to detect an exaptation.

Our introduction of a *developmental program* allows us to model the key ideas necessary for an exaptation to occur. Information needs the ability to be duplicated, changed, and

then reintroduced to serve a new purpose for an exaptation to occur. This is easy to model with our *developmental program* if we introduce the duplication of *blocks*. After a *block* is duplicated, each copy is able to continue mutating on its own. When combined with the idea of non-coding genes that is inherently provided by the *developmental program*, we can have one of the copies mutate and be reintroduced in a modified manner, which provides all the components necessary for an exaptation to happen. Though this means that exaptations are possible within our model, it will take significant work to develop the mechanisms used to detect when and how exaptations happen. It may be possible to identify exaptations by detecting certain rare changes, as suggested by Chipman (2021). Such work could offer insight as to the conditions necessary and sufficient for the occurrence of exaptations.

Another possible area of exploration opened by this model is a deeper study into the robustness of *developmental programs*. Our experiment running evolved populations on new landscapes demonstrates that *developmental programs* evolved under changing conditions enable agents to adapt to new environments more effectively. This may indicate that *programs* evolved on several landscapes exhibit robustness and canalization in their *developmental program*, meaning that these *programs* are able to follow a similar developmental trajectory throughout their lifetime even when the landscape is perturbed, or possibly even when introduced to an entirely new landscape. Further analysis into this area may reveal how the conditions under which a developmental process evolves lead to a more or less robust strategy.

Conclusion

Studying the evolution of development is difficult, but in this paper, we have reported on advances addressing such challenges using an abstract model. Our new evo-devo model demonstrates results that align with biological observations, including synergy between evolutionary and developmental processes, the evolution of adaptable features, and highly conserved regulatory genes. The ability of our model to represent this range of biological ideas lends evidence to its predictive capacity. Many of our observations align with those of Todd et al. (2020) and substantiate our claims from Ashworth et al. (2022), particularly on the optimal balance between development and evolution. Furthermore, our model lays the groundwork for the exploration of future work such as developmental exaptations and developmental robustness. Our model enables the exploration of such phenomena at an abstract level so that work can be done to identify the mechanisms in areas that were previously difficult to study.

Acknowledgements

We would like to thank the ALIFE 2023 reviewers for constructive feedback, which greatly improved the final version of this paper. We would also like to thank Mitchel J Daniel for helpful feedback on an earlier draft of this paper.

References

- Akhshabi, S., Sarda, S., Dovrolis, C., and Yi, S. (2014). An explanatory evo-devo model for the developmental hourglass. *F1000Res*, 3:156.
- Ashworth, J., Lee, Y., Shen, J., Kim, E., Decker, Z., and Yoder, J. (2022). Evolution of developmental strategies in nk fitness landscapes. In *ALIFE 2022: The 2022 Conference on Artificial Life*. MIT Press.
- Carroll, S. B. (2012). *Endless forms most beautiful: The new science of evo devo and the making of the animal kingdom*. Quercus.
- Chen, L., DeVries, A. L., and Cheng, C.-H. C. (1997). Convergent evolution of antifreeze glycoproteins in antarctic notothenioid fish and arctic cod. *Proceedings of the National Academy of Sciences*, 94(8):3817–3822.
- Chipman, A. D. (2021). *Developmental Exaptation*, pages 29–38. Springer International Publishing, Cham.
- Crean, A. J. and Bonduriansky, R. (2014). What is a paternal effect? *Trends in Ecology 'I&' Evolution*, 29(10):554–559.
- Frankenhuis, W. E. and Walasek, N. (2020). Modeling the evolution of sensitive periods. *Developmental Cognitive Neuroscience*, 41:100715.
- Frézal, L. and Félix, M.-A. (2015). The natural history of model organisms: *C. elegans* outside the petri dish. *eLife*, 4:e05849.
- Fujimoto, K., Ishihara, S., and Kaneko, K. (2008). Network evolution of body plans. *PLOS ONE*, 3(7):1–13.
- Hornby, G. S., Lohn, J. D., and Linden, D. S. (2011). Computer-automated evolution of an x-band antenna for nasa's space technology 5 mission. *Evolutionary computation*, 19(1):1–23.
- Kauffman, S. and Levin, S. (1987). Towards a general theory of adaptive walks on rugged landscapes. *Journal of Theoretical Biology*, 128(1):11–45.
- Khadka, S. and Tumer, K. (2018). Evolutionary reinforcement learning. *arXiv preprint arXiv:1805.07917*, 223.
- Marco, R., Husson, D., Herranz, R., Mateos, J., and Medina, F. J. (2003). *Drosophila melanogaster* and the future of 'evo-devo' biology in space. challenges and problems in the path of an eventual colonization project outside the earth. *Adv Space Biol Med*, 9:41–81.
- Pitzer, E. and Affenzeller, M. (2012). A comprehensive survey on fitness landscape analysis. In *Studies in Computational Intelligence*, volume 378, pages 161–191.
- Portugal, I., Alencar, P., and Cowan, D. (2018). The use of machine learning algorithms in recommender systems: A systematic review. *Expert Systems with Applications*, 97:205–227.
- Potts, R. (2002). Complexity and adaptability in human evolution. *Probing human origins*, pages 33–57.
- Shen, D., Wu, G., and Suk, H.-I. (2017). Deep learning in medical image analysis. *Annual review of biomedical engineering*, 19:221–248.
- Soltoggio, A., Stanley, K. O., and Risi, S. (2018). Born to learn: the inspiration, progress, and future of evolved plastic artificial neural networks. *Neural Networks*, 108:48–67.
- Sommer, R. J. (2009). The future of evo-devo: Model systems and evolutionary theory.
- Thanukos, A. (2009). How the adaptation got its start. *Evolution: Education and Outreach*, 2:612–616.
- Tinós, R. and Yang, S. (2014). Analysis of fitness landscape modifications in evolutionary dynamic optimization. *Information Sciences*, 282:214–236.
- Todd, G., Candadai, M., and Izquierdo, E. J. (2020). Interaction Between Evolution and Learning in NK Fitness Landscapes. In *The 2020 Conference on Artificial Life*, pages 761–767, Cambridge, MA. MIT Press.
- Yamamoto, F.-i. (1995). Molecular genetics of the abo histo-blood group system. *Vox Sanguinis*, 69(1):1–7.