

# Lenia and Expanded Universe

Bert Wang-Chak Chan

Hong Kong  
albert.chak@gmail.com

## Abstract

We report experimental extensions of Lenia, a continuous cellular automata family capable of producing lifelike self-organizing autonomous patterns. The rule of Lenia was generalized into higher dimensions, multiple kernels, and multiple channels. The final architecture approaches what can be seen as a recurrent convolutional neural network. Using semi-automatic search e.g. genetic algorithm, we discovered new phenomena like polyhedral symmetries, individuality, self-replication, emission, growth by ingestion, and saw the emergence of “virtual eukaryotes” that possess internal division of labor and type differentiation. We discuss the results in the contexts of biology, artificial life, and artificial intelligence.

## Introduction

The study of cellular automata (CA) is one of the major branches in artificial life and complex systems research. CAs were invented by John von Neumann and Stanislaw Ulam (Von Neumann, 1951; Ulam, 1962), then popularized by John H. Conway’s Game of Life (GoL) (Gardner, 1970) and Stephen Wolfram’s elementary cellular automata (ECA) (Wolfram, 1983). On the one hand, research on CAs led to proofs of Turing completeness and therefore the capability for universal computation in CAs, e.g. GoL and ECA Rule 110 (Rendell, 2002; Cook, 2004). On the other hand, CAs were utilized to model complex systems, generate patterns, and produce computer art.

One line of investigation involves attempts to construct long-range or continuous CAs, search for and study self-organizing autonomous patterns, or *solitons*. These attempts include CAPOW (Rucker, 1999), Larger-than-Life (Evans, 2001), RealLife (Pivato, 2007), SmoothLife (Rafler, 2011a), Lenia (Chan, 2019), and extended Lenia discussed in this paper. They generalize GoL into continuous space using arbitrary long range neighborhoods, into continuous time using arbitrary small incremental updates, and into continuous states using real numbers.

The algorithm of Lenia is as follows (see Figure 1).

1. Take a 2D array (world  $\mathbf{A}$ ) of real values between 0 and 1, initialize with an initial pattern  $\mathbf{A}^0$ .

2. Calculate weighted sums of  $\mathbf{A}$  with a predefined array (kernel  $\mathbf{K}$ ), which is equivalent to calculate the convolution  $\mathbf{K} * \mathbf{A}$ ; the kernel  $\mathbf{K}$  has radius  $R$ , forming a ring or multiple concentric rings (parameter  $\beta$  = list of peak value of each ring).
3. Apply a growth mapping function  $G$  to the weighted sums; the growth mapping  $G$  is any unimodal function (parameters  $\mu$  = growth center,  $\sigma$  = growth width).
4. Add a small portion  $dt$  of the values back to the array  $\mathbf{A}$ .
5. Finally clip the states of  $\mathbf{A}$  to between 0 and 1.
6. Repeat steps 2-5 for each time-step.

In formula:

$$\mathbf{A}^{t+dt} = [\mathbf{A}^t + dt G(\mathbf{K} * \mathbf{A}^t)]_0^1 \quad (1)$$

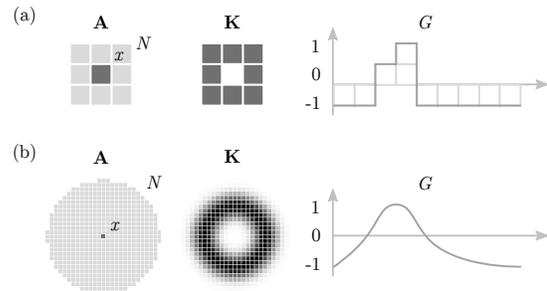


Figure 1: Rules of GoL and Lenia. (a) In GoL, a site  $x$  in the world  $\mathbf{A}$  has 8 surrounding sites as its Moore neighborhood  $N$ . Calculate the weighted sum of  $N$  with kernel  $\mathbf{K}$  (all weights 1), apply a mapping function  $G$  (survival = 0, birth = +1, death = -1), add the value back to the site  $x$  and clip it to 0 or 1, repeat. (b) In Lenia, the rule is similar, but generalized to the continuous domain - infinitesimal sites  $x$  with real values, circular neighborhood  $N$ , ring-like kernel  $\mathbf{K}$ , smooth mapping  $G$ , and incremental update by factor  $dt$ .

In such a continuous CA system, many self-organizing, autonomous solitons were discovered with diverse structures

and behaviors. Structures include symmetries like bilateral, radial and rotational symmetries, linear polymerized long-chains, and irregular structures. Behaviors include regular modes of locomotion like stationary, directional, rotating, gyrating, and irregular behaviors like chaotic movements, metamorphosis (shape-shifting), and particle collisions.

The current on-going work is aimed to answer the following open questions raised in the original Lenia paper (Chan, 2019):

9. Do self-replicating and pattern-emitting lifeforms exist in Lenia?
10. Do lifeforms exist in other variants of Lenia (e.g. 3D)?

We answer “Yes” to both questions. By exploring variants and generalizations of Lenia, we discovered new types of solitons with a wide range of unseen behaviors including self-replication and pattern emission. The current work also aims towards answering Lenia’s relationship with Turing completeness (question 6), open-ended evolution (question 7), and other implications in artificial life and artificial intelligence.

### Related Works

SmoothLife (Rafler, 2011a), an earlier independent discovery similar to Lenia, was the first to report solitons (called “smooth gliders”) in a continuous 2D CA.

Extensions to Lenia rules were inspired by numerous works about CAs in the literature and in code repositories. There were various attempts in taking existing 2D CAs and other artificial life systems into higher dimensions (Bays, 1987; Imai et al., 2010; Rafler, 2011b; Sayama, 2012; Hutton, 2012). Duplication of components in existing CA rules were demonstrated to produce very different dynamics, e.g. Multiple Neighborhoods CA (MNCA) (Rampe, 2018b,a), multiple layer CA “Conway’s Ecosystem” (Sherrill, 2019). There were also efforts to blur the boundary between CA and neural networks and brought amazing breakthroughs, e.g. Neural CA (Mordvintsev et al., 2020).

The results of the current work can be compared with other artificial life models, especially particle systems with multiple species of particles, e.g. Swarm Chemistry (Sayama, 2009), Primordial Particle Systems (Schmickl et al., 2016), Clusters (Ventrella, 2017), developed from the pioneering Boids (Reynolds, 1987). These models are able to generate cell-like structures of various styles.

### Methods

Inspired by the related works, we experimented with 3 major extensions to the original Lenia, namely higher dimensions, multiple kernels, multiple channels, and any combinations thereof. We updated the existing open-source software, designed semi-automatic algorithms to search for new patterns and solitons, and performed qualitative analysis on the results.

### Rule Extensions

**Higher dimensions** The 2D arrays in Lenia were upgraded to 3 or higher dimensions, and the algorithms used in the software were subsequently generalized to deal with multidimensional arrays. The number of dimensions is denoted as  $d$ . Experiments of 3D Lenia have been carried out before but without success in finding interesting patterns. With the utilization of GPU parallel computing and better searching algorithms, stable solitons have been found.

**Multiple kernels** The original Lenia involves one kernel  $\mathbf{K}$  with radius  $R$ , one growth mapping  $G$ , and one increment factor  $dt$ . Now multiply the rule with multiple kernels  $\mathbf{K}_k$ , each with relative radius  $r_k R$ , and corresponding growth mapping  $G_k$ . Weighted average of the results by factors  $h_k/h$  ( $h$  is the sum of  $h_k$ ) is taken. The number of kernels is denoted as  $n_k$ . This extension was inspired by MNCA (Rampe, 2018b,a) that produces highly irregular and dynamic patterns.

**Multiple channels** Lenia and most CAs have only one world array  $\mathbf{A}$ , so we experimented with “parallel worlds” or multiple channels  $\mathbf{A}_i$ . In addition to the kernels feeding back to each channel, there are also cross-channel kernels for the channels to interact with each other. Denote the number of channels as  $c$ , the number of self-interacting kernels per channel as  $k_s$ , and the number of cross-channel kernels per channel pair as  $k_x$ , then the total number of kernels  $n_k = k_s c + k_x c(c-1)$ . This was inspired by multi-layer CA (Sherrill, 2019) and Neural CA (Mordvintsev et al., 2020).

**Combinations** The above extensions (and potentially others) can be further combined to produce unique results, e.g. 3D 3-channel 3-self-kernel. The original Lenia becomes a special case, i.e. 2D 1-channel 1-kernel Lenia.

The algorithm of extended Lenia is summarized as follows (see Figure 2).

1. Create multiple channels of world  $\mathbf{A}_i (i = 1 \dots c)$ , each channel a  $d$ -dimensional array of real values between 0 and 1; initialize each channel with initial pattern  $\mathbf{A}_i^0$ .
2. Define multiple  $d$ -dimensional arrays of kernels  $\mathbf{K}_k (k = 1 \dots n_k)$ , each with relative radius  $r_k R$ , parameter  $\beta_k$ , source channel  $i$ , destination channel  $j$ , and corresponding growth mapping  $G_k$  with parameters  $\mu_k$  and  $\sigma_k$ .
3. For each kernel  $\mathbf{K}_k$ , calculate weighted sums with its source channel  $\mathbf{A}_i$ , i.e. convolution  $\mathbf{K}_k * \mathbf{A}_i$ .
4. Apply growth mapping  $G_k$  to the weighted sums.
5. Add a small relative portion  $dt \cdot h_k/h$  of the values to destination channel  $\mathbf{A}_j$ .
6. Repeat steps 3-5 for every kernel  $\mathbf{K}_k$ .

7. Finally clip the states of each channel  $\mathbf{A}_i$  to between 0 and 1.
8. Repeat steps 3-7 for each time-step.

In formula:

$$\mathbf{A}_j^{t+dt} = \left[ \mathbf{A}_j^t + dt \sum_{i,k} \frac{h_k}{h} G_k(\mathbf{K}_k * \mathbf{A}_i^t) \right]_0^1 \quad (2)$$

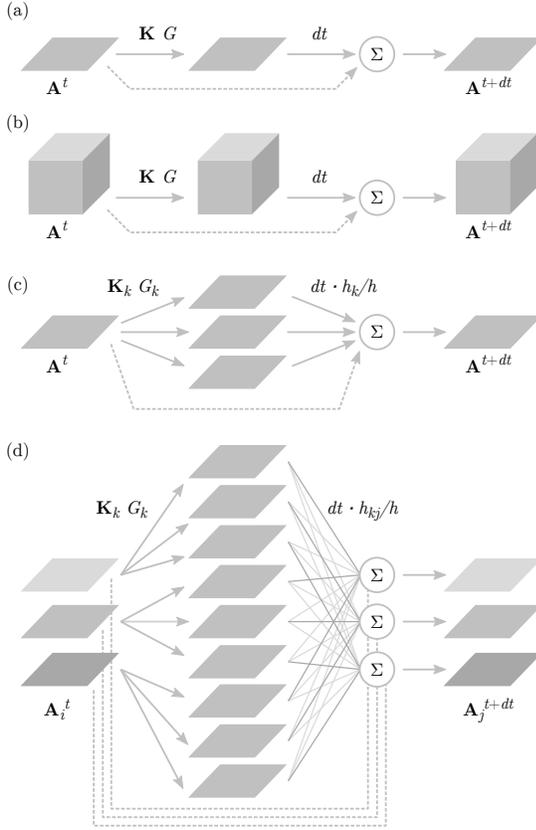


Figure 2: Extended Lenia rules. (a) Original 2D Lenia: world  $\mathbf{A}$  at time  $t$  passes through convolution with kernel  $\mathbf{K}$ , growth mapping  $G$ , and incremental update  $\Sigma$  to next time step  $t + dt$ . (b) Higher dimensions with  $d$ -dimensional arrays. (c) Multiple kernels, where multiple  $\mathbf{K}_k$  and  $G_k$  feed into  $\Sigma$  by factors  $h_k$ . (d) Multiple channels, where separate channels of world  $\mathbf{A}_i$  pass through  $\mathbf{K}_k$  and  $G_k$ , feed into multiple  $\Sigma$  that update channel  $\mathbf{A}_j$ . The architecture approaches a recurrent convolutional neural network.

### Genotypes, Phenotypes, and Search Space

The search space of extended Lenia consists of all possible genotypes and phenotypes. A *genotype* here is a particular combination of rule parameter values, a *phenotype* is a particular configuration of the world arrays. A pattern (or a soliton) is jointly specified by its genotype and phenotype.

Consider a moderately complex rule of 3D 3-channel 3-self-kernel, with all kernels composed of 3 concentric rings, and a soliton size of  $20 \times 20 \times 20$  sites. In this case, the genotype is in the form  $(r, h, \beta^3, \mu, \sigma)^{15}$ , that is 105 parameter values, and the phenotype consists of 3 channels of 3-dimensional arrays, amounting to 24000 site values.

### Search Algorithms

We want to search for interesting patterns or solitons given the new rules. However, the rules create higher degrees of freedom, hence summon the curse of dimensionality. The size of the search space now grows exponentially, manual parameter search and pattern manipulations become difficult if not impossible. We employed several semi-automatic search algorithms with an interactive user interface to tackle this problem and help exploring the search space.

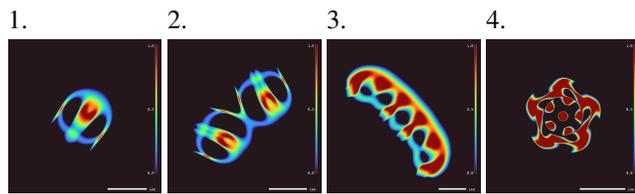
The algorithms pick genotypes and phenotypes according to some criteria in the search space, and automatically filter them by survival, i.e. to check that the solitons will not come to vanish or occupy the whole grid after running the CA for a period of time. The results are then selected by the human-in-loop for novelty, visual appeal, or prospects for further study, and used in further rounds of semi-automatic search.

**Global search** The algorithm generates random genotypes and phenotypes from the global search space. The ranges of random values can be tuned to narrow down the search. Once interesting patterns or solitons are found, they can be fed to other algorithms.

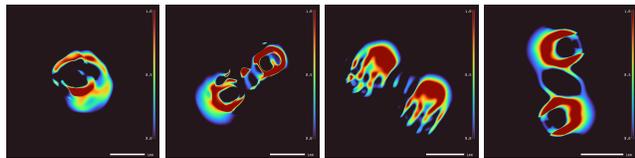
**Depth-first search** Starting with an initial soliton, the algorithm adds small random deviations to one or all values in its genotype, and tests if the phenotype survives. If it does, record the survived phenotype, repeat the process using this new genotype and phenotype as the starting point. This method allows deeper explorations of the search space.

**Breadth-first search** This algorithm is similar to depth-first search, but using the initial genotype and phenotype as the starting point in every search. This method is able to explore variations of one particular interesting soliton.

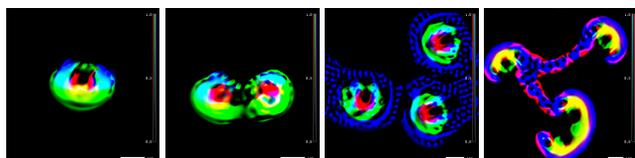
**Genetic algorithm** First set an fitness function and optimization goal (e.g. faster moving speed, higher mass oscillation). Starting from an initial soliton in a pool of samples, the genetic algorithm aggregates the pool using two genetic operators, (1) mutation: pick a random sample from the pool and randomly mutate its genotype; (2) recombination: pick two random samples, create a new sample by randomly mixing their channels and associated parameters. After checking for survival, calculate the fitness value of the new sample, add it to the pool, and sort the pool by fitness. Finally the samples with top fitnesses are recorded as results.



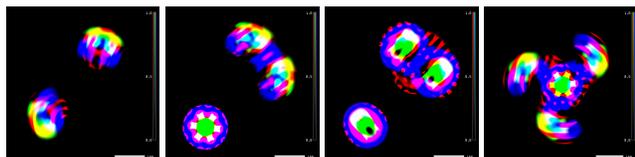
(a) Original Lenia: 1. *Orbium*; 2. *Orbium* individuals in elastic collision; 3. long-chain *Pentaptera*; 4. rotating *Asterium* with 5-fold rotational symmetry.



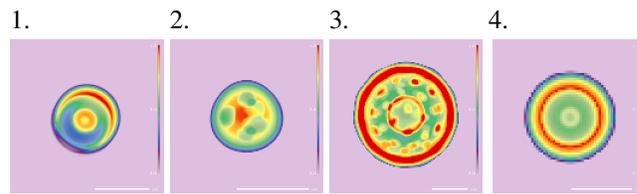
(b) Multi-kernel Lenia: 1. the first replicator discovered; 2. right after its self-replication; 3. solitons in parallel pair; 4. solitons in elastic collision, repulsive forces hinted by electricity-like lines.



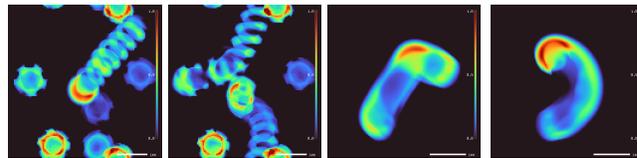
(c) Multi-channel Lenial: 1. aggregated soliton with cell-like structures; 2. right after its self-replication; 3. sea of emitted particles; 4. dendrite-like emissions from replicating solitons.



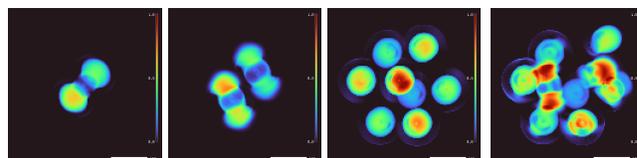
(d) "Aquarium" phenotypes: 1-3. (left to right) gyrating, slightly oblique; stationary, parallel pair; slow-moving, parallel slow-moving; 4. a few solitons in a stable, dynamic formation.



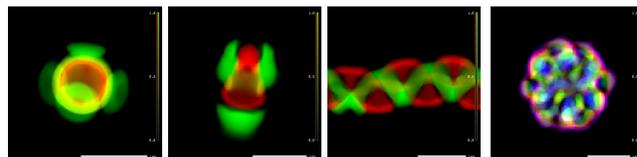
(e) Higher dimensions Lenia: 1. moving sphere; 2. rotating sphere with bubbles in trigonal bipyramidal arrangement; 3. pulsating sphere with dots; 4. pulsating 4D hypersphere, showing a 3D slice.



(f) 3D multi-kernel Lenia: 1. moving "Snake" and static "food dots"; 2. Snake grows while ingesting 3 dots (now spans across the screen); 3-4. a mutant of Snake performing elegant dance.



(g) Exponential growth: 1-3. replicator under three rounds of binary fission, repulsive forces visible as negative spheres; 4. Offsprings migrate out for further replication.



(h) 3D multi-channel Lenia: 1. tetrapod; 2. moving soliton with red nucleus and green pseudopods; 3. double helix pattern; 4. rainbow ball.

Figure 3: Sample solitons. Scale bar at lower right represents kernel radius  $R$ .

## Software

The interactive software for Lenia, now open source in GitHub, was updated with the above rule extensions and search algorithms.

For visualization of higher dimensions, the 3D world is flattened to 2D using a depth map, which can show the internal structures of 3D objects with transparency. For dimensions higher than 3, one 3D slice of the array is displayed.

The default color palette used for single-channel visualization was changed from Jet to Turbo (Mikhailov, 2019) for better perceptual uniformity. For higher dimensions, Paul Tol's Rainbow palette (Tol, 2018) is recommended to show 3D internal structures. For multiple channels, the first three channels are displayed in red, green and blue (RGB).

## Results

With the help of semi-automatic algorithms, we discovered a number of new structures and behaviors in the extended rules. Unlike the original Lenia, where most solitons are well defined and moderately symmetric, solitons found in the extended rules either possess even higher symmetries (in higher dimensions), or become highly chaotic yet highly self-organized and persistent (with multiple kernels or channels). See Figure 3 for samples (include the original Lenia for reference).

### Rule Specific Observations

**Higher dimensions** In higher dimensions, stable solitons are hard to find, and the found ones are highly stable. Their external shapes are almost always spherical, and their inter-

nal structures can be complex and highly symmetrical. In some cases, bubbles (inner voids) are arranged as vertices of Platonic solids or regular polyhedra, e.g. tetrahedron, octahedron, triangular bipyramid, and icosahedron. Most solitons are motionless, a few of them are oscillating, rotating, or directional moving.

Higher dimensional structures are not too chaotic even with multi-kernel or multi-channel extensions, which are supposed to introduce a lot of instability.

**Multiple kernels** As demonstrated by MNCA, multiple kernels could introduce instability and interesting dynamics into the complex system. Overall chaoticity of the CA increases, but given the right parameters, the system can achieve even higher degrees of self-organization and persistence. There we discovered new or more common behaviors - individuality, self-replication, emission, growth, etc.

**Multiple channels** In a multi-channel world, each channel develops patterns according to its own rule, and at the same time, these patterns co-develop and influence each other through channel-channel interactions. Different channels of a soliton could exhibit something like a division of labor, e.g. some channels act as outer flexible shells (membranes), some form central masses (nuclei), together they form cell-like structures. In a special case, a particular type of “Aquarium” genotype could produce an array of phenotypes, come with different behaviors and complex interactions.

### Common Phenomena

We summarize common soliton behaviors and phenomena that can be seen across rules. Refer to Figure 4 for schematic illustrations.

**Locomotion** In the original Lenia, solitons engage in various kinds of locomotory behaviors, like stationary, directional, rotating, gyrating, oscillating, alternating, drifting, and chaotic movements. In extended Lenia, these movements are still observed, but rotation becomes very rare, possibly because there are fewer cases of rotational symmetry. With multi-kernel and multi-channel, chaotic movements and metamorphosis (shape-shifting) become more prevalent than regular behaviors. Conversely, in 3 or higher dimensions, solitons become predominantly stationary.

**Individuality** Among the soliton species in the original Lenia, only the *Orbidae* family (out of 18 families) engages in some forms of elastic or inelastic collisions - when two *Orbidum* individuals collide, they often reflect each other and survive, or occasionally stick together to form a composite soliton *Synorbium*. For other species, solitons in collision simply lose self-organization and die out. Thus *Orbidum* possesses some kind of *individuality*, in that each soliton is able to maintain its own boundary or “personal space” and avoid mixing its contents with others.

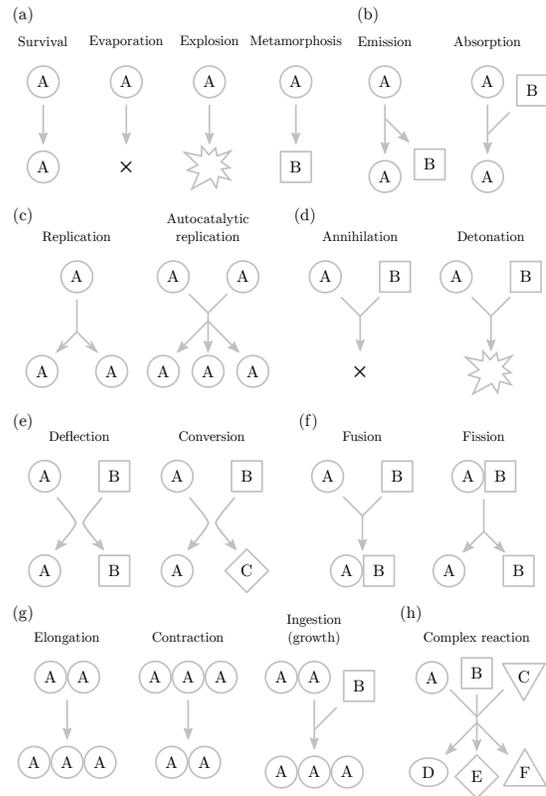


Figure 4: Behaviors and interactions of solitons in extended Lenia. Categories: (a) single soliton developments, (b) simple reactions, (c) reproduction, (d) mutual destruction, (e) elastic collisions, (f) inelastic collisions, (g) long-chain reactions, (h) complex reactions.

In multi-kernel or multi-channel rules, *Orbidum*-like individuality becomes a common phenomenon. Numerous types of solitons manage to maintain self-organization upon collision, thus are able to involve in complex particle interactions. It is possible that some of their kernels or channels act as repelling forces that separate individuals from each other.

**Self-replication** An important milestone in the study of Lenia is the discovery of *self-replication*. It is conspicuously missing in the original Lenia, but turns out to be not rare in extended rules. The mechanism is usually one soliton develops into two partitions of similar structures, each develops into a full soliton, drifts away, and is capable of further division. In highly reproductive cases, new individuals can develop out of debris. In multi-channel rule, self-replication is usually initiated by division in one channel, then other channels follow suit. Self-replication is closely related to individuality - newly replicated parts need to repel and separate from each other to complete the process.

There is also *autocatalytic replication*. In some cases, self-replication does not or only seldom happens when the

density of solitons is low. But when the density rises (e.g. from the very slow reproduction), congregation of solitons will force self-replication to happen, kicks start a wave of autocatalysis and causes exponential growth.

Reproducing solitons occupy all available space sooner or later. But if those solitons also vanish with a death rate not far from the birth rate, it may maintain a “healthy” population of regenerating solitons.

**Growth by ingestion** We found this curious phenomenon only in one setting (the “3D Snake” genotype) of 3D multi-kernel rule. In the Snake world, there is one type of static spherical solitons, “food dots”, and one type of dynamic helical solitons, “snakes”. A snake keeps contracting or extending linearly at one or both ends, giving an illusion of a moving snake. When its extending end reaches one food dot, it merges with that “inanimate” dot (*ingestion*), turns it into part of the “living” soliton, and slightly elongates (*growth*). The snake also slightly changes direction towards dots within reach, giving an illusion of the snake pursuing food.<sup>1</sup>

This growth behavior may be related to the elongation and contraction of long-chain species (Pterifera) in the original Lenia. It is probably an exceptional and isolated case, but remarkable that it is even possible to happen.

**Emission** In GoL, an important category of patterns that enables universal computation is the “guns” - stationary patterns that emit moving solitons. There are other categories: “puffer trains” (moving emit stationary), “rakes” (moving emit moving), and complex tertiary emissions. *Pattern emission* is sometimes found in extended Lenia, but is usually irregular and of the “puffer train” type. We aim to find more regular, reliable emitters in Lenia, especially of the “gun” type, in order to pursue Turing completeness (Berlekamp et al., 2018), or some kind of analog computation.

**Division of labor** In multi-kernel and multi-channel rules, various channels and kernels engage in different behaviors yet influence each other. As discussed above, some kernels or channels may form patterns that exert repulsion and define the scope of the pattern, some may facilitate binary fission, some engage in pattern emission; some may provide stability and some others provide motility.

Dynamic or static patterns from different channels combine into an *aggregated soliton*. For the aggregated soliton to survive and prosper, its channels must coordinate and cooperate with each other. It acts as a single unit, engages in diverse complex behaviors, and evolves as a whole.

**Differentiation** We found a special range of “Aquarium” genotypes in multi-channel rule, where one genotype pro-

<sup>1</sup>Upon seeing in action, one may be reminded of the “Snake” mini-game in Nokia mobile phones, except that the Snake world here is not pre-programmed and snake control is not provided.

duces multiple phenotypes of aggregated solitons, each having own stable structure and behavior.

The collection may include solitons with directional (*rectus*), oblique (*limus*), gyrating (*gyrans*), stationary (*lithos*), slower or faster moving (*tardus* or *tachus*), parallel / antiparallel pairing (*para-* / *anti-*) phenotypes, and possibly more. Each of the phenotypes is usually quite stable and well defined, but can switch to another phenotype in specific occasions, e.g. upon collision or after self-replication.

This is a desirable emergent property in Lenia, since it enables heterogeneous soliton-soliton interactions for the first time. Complex interactions and reactions, together with self-replication, may lead to higher-level structures and collective behaviors, like building up tissue-like megastructures.

## Discussion

### Relations to Biology

The original Lenia, and other models like SmoothLife (Rafler, 2011a), have shown that continuous CAs are able to produce patterns with appearance and dynamics comparable to real world biology. With more discoveries in extended Lenia, we can add more comparisons between artificial life and biological life.

**Origin of Life** The gradual emergence of several important phenomena in Lenia is reminiscent of the origin of life.

Cell individuality and self-replication are among the hallmarks of life on Earth, each has abiotic origins. Individuality originated from lipid membranes that were formed spontaneously by hydrophobic molecules in the primordial soup, separate the outside world from an area where specific chemical reactions can occur, and protect such an area from physical attacks and chemical insults (Haldane, 1929). Self-replication possibly came from the RNA World, where RNA molecules self-assemble and self-replicate out from amino acid building blocks (Joyce, 1989).

Division of labor inside eukaryotic cells, i.e. the cells of all animals, plants and fungi, stemmed from endosymbiosis of more basic lifeforms, i.e. bacteria, archaea, and possibly viruses (Mereschkowsky, 1905; Sagan, 1967). Mitochondria originated from an ancient unification of  $\alpha$ -proteobacteria with archaea. The bacteria provided aerobic energy metabolism, and the archaea provided the cytoplasm and membrane. Chloroplasts originated from further endosymbiosis with cyanobacteria, equipped algae and plant cells with photosynthesis. The nuclei of the eukaryotic cell may have originated from DNA viruses (Bell, 2001). These organelles, together with the cell body, perform various functions separately and also cooperate closely.

Here in extended Lenia, similar processes of individuality, self-replication, and division of labor have emerged from the more and more generalized CA rules. Is it possible that these processes, and maybe others, are essential in creating more

Lenia	Cellular level	Molecular level
Site	Cell	Molecule
Kernel	Cell signaling	Chemical reaction
Single-channel soliton	Simple multi-cellular life	Prokaryote, virus
Multi-channel soliton	Complex multi-cellular life	Eukaryotic cell
Division of labor	Organs	Organelles
Center	Heart / brain	Nucleus
Individuality	Body, skin	Cytoplasm, membrane
Motility	Limb	Pseudopod
Emission	Signal	Cytokine
Differentiation	Polymorphism	Cell type

Table 1: Comparisons of self-organization levels in Lenia to biology.

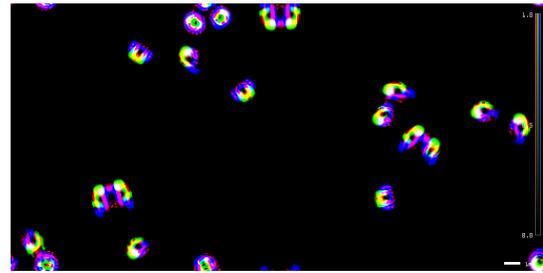
and more complex evolvable systems in both the real world and the virtual world.

**Organization hierarchy** If we compare the levels of organization in Lenia to the hierarchy of biological structures - from atoms to organisms to ecosystems, we could come up with more than one interpretations (Table 1).

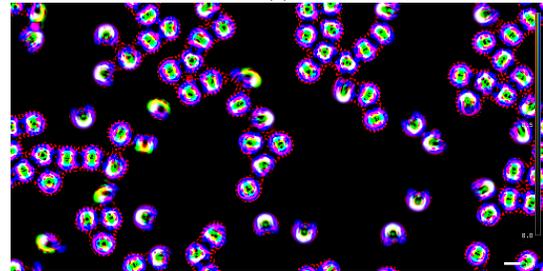
The straightforward take, as implied in the name “cellular automata”, is to interpret a site in CA as a biological “cell” (or a “concentration of cells” in continuous CAs). A neighborhood or kernel would be something like a cell signaling pathway, affecting surrounding cells with a certain effect. In this analogy, single-channel solitons are like simple multicellular organisms without organs (e.g. sponges, jellyfish, fungi, kelps, slime molds), and multi-channel solitons are like complex multicellular organisms (e.g. bilaterian animals, higher plants), with division of labor among organs.

In a more interesting interpretation, a site can be thought of as a “molecule” (or a “concentration of molecules” in continuous case). Consequently a kernel would be a type of molecular force or chemical reaction, influencing surrounding molecules according to distance and concentration. Single-channel solitons, including those in the original Lenia, would resemble simple microscopic lifeforms (e.g. bacteria, archaea, viruses), possess self-organization, self-replication, symmetry, individuality, motility, etc. Multi-channel solitons, especially of the “Aquarium” genotypes, would resemble eukaryotic cells, with internal division of labor among organelles, and differentiation among cell types.

**Virtual cells** These multi-channel solitons no longer need different genotypes to realize different behaviors, all they need are subtle changes in the division of labor and coordination of internal parts, express themselves as different phe-



(a)



(b)

Figure 5: “Virtual eukaryotes” in action. (a) Solitons of “Aquarium” set similar to Figure 3(d), but with a highly reproductive gyrating phenotype, start to reproduce, differentiate, migrate, interact and react with each other. (b) A few tissue-like colonies gradually formed, akin to what happens in multicellularity.

notypes. The kinds of division of labor observed include:

- Some channels form a pattern like a “nucleus”, usually at the center of an entity. Other channels develop patterns around the nucleus. Whenever the nucleus moves, self-replicates, or dies out, other channels usually follow suit.
- Some channels form “cytoplasm” or “membrane” that defines a private area around the nucleus, keeps safe distances from other patterns by means of repulsive and attractive forces.
- Some channels may form movable parts like “pseudopods”, direct the movement of whole soliton when the pseudopod is at the periphery, or stay stationary when it is kept inside the cytoplasm.
- Some channels may form “tails” behind the soliton (perhaps not for propulsion).
- Some channels may emit signal-like small particles like “cytokines”, significance uncertain.

In this regard, these complex solitons could be dubbed “virtual eukaryotes” or “virtual stem cells” (Figure 5). They are by far the most lifelike patterns in the Lenia family of continuous CAs.

Altogether, a community of “virtual eukaryotes” engages in diverse emergent behaviors and complex interactions

thanks to their own high level of self-organization, and it is not impossible that they will later be shown to produce another level of emergence and self-organization.

### Relations to Other Systems in Artificial Life

Particle systems (PS), like Swarm Chemistry (Sayama, 2009), Primordial Particle Systems (Schmickl et al., 2016), Clusters (Ventrella, 2017), have multiple species of particles engage in intra- and inter-species interactions. They produce results that are comparable to multi-channel Lenia. The particles in PSs self-organize into aggregated patterns (solitons), build cell-like structures like cytoplasm, membranes and nuclei, and engage in binary fission, etc. One difference is that solitons in these PSs do not possess strong individuality, hence almost always merge upon collision.

It may be difficult to compare CAs and PSs because of a few fundamental differences in their rulesets - PSs calculate the vector movements of every particle, and maintain a conservation of mass, while CAs only keep track of scalar states and the total mass is not conserved. To deal with this discrepancy, one may interpret the scalar states in CAs as concentrations of virtual molecules across a grid (see Molecular level column in Table 1), and the molecules can be constructed, destroyed or migrated with rates according to the CA rule. The relationship between CAs and PSs would be like that of the macroscopic view of thermodynamics vs the microscopic view of Newtonian physics.

### Relations to Artificial Intelligence

There are efforts to employ methodologies from artificial intelligence to search for new artificial life patterns. Reinke et al. (2019) used curiosity-based algorithm IMGEP (Baranes and Oudeyer, 2013) and neural networks like CPPN and VAE to explore the search space of the original Lenia, with success in increasing the diversity in pattern search. Interactive evolutionary computation (IEC) (Takagi, 2001) and genetic algorithms (GA) were also used in semi-automatic discovery of new patterns (Chan, 2019).

On the other hand, a number of researchers have noticed the close relation between CAs and neural networks (NN) (Wulff and Hertz, 1992; Gilpin, 2018). Mordvintsev et al. (2020) designed Neural CA, a CA-NN hybrid that can be trained to generate and regenerate (also playfully interpolate) predefined patterns. They suggested that the Neural CA could be named “Recurrent Residual Convolutional Networks with ‘per-pixel’ Dropout”.

The architecture of our multi-channel Lenia also approaches a “Recurrent Residual Convolutional Network” (see Figure 2(d)). The “recurrent”, “convolutional”, and “residual” attributes come from the repetitive updates, the convolution kernels, and the contributions from world states, respectively. The growth mapping is analogous to an activation function. The incremental update part vaguely resembles a fully connected layer in NN.

**Comparing Lenia and Neural CA** Lenia relies on tuning the parameters of kernels and growth mappings to “train” the model into generating self-organizing patterns, while the incremental update part has limited flexibility. Neural CA, on the other hand, is fixed in the convolutional kernels and activation functions, but heavily parameterized in the fully connected layers. Lenia is aimed at exploring novel patterns, helped by evolutionary, genetic and exploratory algorithms; Neural CA is aimed at generating predefined patterns, results are optimized by gradient descent.

Despite the differences, Lenia and Neural CA do one thing in common - exploit the self-organizing, emergence-inducing, and regenerating powers of CAs. Neural CA also exploits the learnable nature of its NN architecture, and it remains unknown whether the Lenia model can be made learnable to achieve other goals.

### Future Works

The following future works are proposed:

- Automatic identify and count soliton individuals. This would allow the software to detect individuality, self-replication, birth rate and death rate, soliton interactions, etc., and hence select for these attributes using genetic algorithms.
- Using “virtual eukaryotes” as elements, study the possibility of the next level of emergence and self-organization, and compare the results to multicellularity, cell differentiation, cell signaling in biology.
- Develop Lenia into trainable Recurrent Residual Convolutional Networks or GANs for whatever purpose.

### Supplementary Info

The open-source software of Lenia in Python is available at: <https://github.com/Chakazul/Lenia>

### Acknowledgements

This work is dedicated to the late John H. Conway, inventor of the Game of Life, and the late Richard K. Guy, discoverer of the “glider”, the first soliton in GoL.

I would like to thank Pierre-Yves Oudeyer and the Inria Flowers team Chris Reinke, Mayalen Etcheverry, Clement Moulin-Frier for intellectual exchanges; Will Cavendish, Clément Hongler, Gloria Capano, Takaya Arita, Nick Ky-parissas, Michael Simkin, Michael Klachko, John Sherrill, Alex Mordvintsev, Craig Reynolds for valuable discussions and inspirations; Hector Zenil, Josh Bongard, Dennis Allison for academic opportunities; David Ha, Lana Sinapayen, Sam Kriegman for continued supports in my road as an independent researcher.

## References

- Baranes, A. and Oudeyer, P. Y. (2013). Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robotics and Autonomous Systems*, 61(1):49–73.
- Bays, C. (1987). Candidates for the Game of Life in three dimensions. *Complex Systems*, 1(3):373–400.
- Bell, P. J. L. (2001). Viral eukaryogenesis: Was the ancestor of the nucleus a complex DNA virus? *Journal of Molecular Evolution*, 53(3):251–256.
- Berlekamp, E. R., Conway, J. H., and Guy, R. K. (2018). *Winning Ways for Your Mathematical Plays, Volume 2*. AK Peters/CRC Press.
- Chan, B. W.-C. (2019). Lenia: Biology of artificial life. *Complex Systems*, 28(3):251–286. arXiv preprint.
- Cook, M. (2004). Universality in elementary cellular automata. *Complex Systems*, 15(1):1–40.
- Evans, K. M. (2001). Larger than Life: digital creatures in a family of two-dimensional cellular automata. In *Discrete Mathematics and Theoretical Computer Science Proceedings vol. AA*, pages 177–192.
- Gardner, M. (1970). Mathematical games: the fantastic combinations of John Conway’s new solitaire game “life”. *Scientific American*, 223(4):120–123.
- Gilpin, W. (2018). Cellular automata as convolutional neural networks. *Physical Review E*, 100(3):032402.
- Haldane, J. B. S. (1929). The origin of life. *Rationalist Annual*, 148:3–10.
- Hutton, T. J. (2012). A 3D glider in the U-Skate World (reaction-diffusion). Retrieved from <https://www.youtube.com/watch?v=WYZVfFOaRgA>. [Video].
- Imai, K., Masamori, Y., Iwamoto, C., and Morita, K. (2010). On designing gliders in three-dimensional Larger than Life cellular automata. In *Natural Computing*, pages 184–190. Springer, Tokyo.
- Joyce, G. F. (1989). RNA evolution and the origins of life. *Nature*, 338(6212):217–224.
- Mereschkowsky, C. (1905). Über Natur und Ursprung der Chromatophoren im Pflanzenreiche [on the nature and origin of chromatophores in the plant kingdom]. *Biologisches Centralblatt*, 25(18):593–604.
- Mikhailov, A. (2019). Turbo, an improved rainbow colormap for visualization. Retrieved from <https://ai.googleblog.com/2019/08/turbo-improved-rainbow-colormap-for.html>. [Blog post].
- Mordvintsev, A., Randazzo, E., Niklasson, E., and Levin, M. (2020). Growing neural cellular automata. *Distill*. <https://distill.pub/2020/growing-ca>.
- Pivato, M. (2007). RealLife: The continuum limit of Larger than Life cellular automata. *Theoretical Computer Science*, 372(1):46–68.
- Rafler, S. (2011a). Generalization of Conway’s “Game of Life” to a continuous domain - SmoothLife. arXiv preprint.
- Rafler, S. (2011b). SmoothLife 3D with FFT glider. Retrieved from <https://www.youtube.com/watch?v=7RsEyu80l-Q>. [Video].
- Rampe, J. (2018a). More explorations with multiple neighborhoods cellular automata. Retrieved from <https://softologyblog.wordpress.com/2018/03/31/more-explorations-with-multiple-neighborhood-cellular-automata/>. [Blog post].
- Rampe, J. (2018b). Multiple neighborhoods cellular automata. Retrieved from <https://softologyblog.wordpress.com/2018/03/09/multiple-neighborhoods-cellular-automata/>. [Blog post].
- Reinke, C., Etcheverry, M., and Oudeyer, P. Y. (2019). Intrinsically motivated exploration for automated discovery of patterns in morphogenetic systems. arXiv preprint.
- Rendell, P. (2002). Turing universality of the Game of Life. In *Collision-Based Computing*, pages 513–539. Springer, London.
- Reynolds, C. W. (1987). Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics*, 21(4):25–34.
- Rucker, R. (1999). Continuous-valued cellular automata in two dimensions.
- Sagan, L. (1967). On the origin of mitosing cells. *Journal of Theoretical Biology*, 14(3):225–274, IN1–IN6.
- Sayama, H. (2009). Swarm chemistry. *Artificial Life*, 15(1):105–114.
- Sayama, H. (2012). Morphologies of self-organizing swarms in 3D Swarm Chemistry. In *Proceedings of the 2012 Genetic and Evolutionary Computation Conference (GECCO 2012)*, pages 577–584. ACM.
- Schmickl, T., Stefanec, M., and Crailsheim, K. (2016). How a life-like system emerges from a simple particle motion law. *Scientific Reports*, 6:37969.
- Sherrill, J. (2019). Conway’s ecosystem. Retrieved from <https://www.shadertoy.com/view/wsl3DN>. [Website].
- Takagi, H. (2001). Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation. *Proceedings of the IEEE*, 89(9):1275–1296.
- Tol, P. (2018). Paul Tol’s notes - sequential color schemes. Retrieved from <https://personal.sron.nl/~pault/#sec:sequential>. [Blog post].
- Ulam, S. (1962). On some mathematical problems connected with patterns of growth of figures. In *Proceedings of Symposia in Applied Mathematics (Vol. 14)*, pages 215–224.
- Ventrella, J. (2017). Clusters. Retrieved from <http://www.ventrella.com/Clusters/>. [Website].
- Von Neumann, J. (1951). The general and logical theory of automata. *Cerebral Mechanisms in Behavior*, 1(41):1–2.
- Wolfram, S. (1983). Statistical mechanics of cellular automata. *Reviews of Modern Physics*, 55(3):601.
- Wulff, N. H. and Hertz, J. A. (1992). Learning cellular automaton dynamics with neural networks. In *Proceedings of the 5th International Conference on Neural Information Processing Systems*, pages 631–638. Morgan Kaufmann Publishers Inc.