morphogen: a browser-native laboratory for cellular automata and agent-based models

Enes Öz
https://github.com/en970/morphogen
https://en970.github.io/morphogen

July 2026

Abstract

Cellular automata are usually presented on the web as pictures. This is a laboratory: ten models drawn from the primary literature, with the parameters their authors used, simulation cores written in C and compiled to WebAssembly, and quantitative observables computed and plotted while the model runs. Every run is a pure function of a seed and a parameter vector, both of which live in the address bar, so a link reproduces a run cell for cell rather than merely depicting one. The models are not validated against themselves; they are validated against numbers other people measured. A nutrient-starved colony box-counts to a fractal dimension approaching the Witten–Sander value of 1.71; the same colony, fed, fills in as a disc of dimension 2.0. Rule 90 from a single live cell reproduces the Sierpiński triangle at $D = \log 3/\log 2 = 1.585$. Sugarscape drives a Gini coefficient from equal endowments to ≈ 0.4. Schelling's households, each content to be outnumbered two to one, segregate a city to ≈ 0.77 against a chance baseline of 0.50. These are assertions in a test suite that runs on every commit. We also report what did not work, and why the failures were instructive: in particular, an early version of the colony model produced a diffuse cloud of wandering cells that box-counted to a perfectly respectable $D \approx 1.71$ while being, physically, nothing like a colony — a cautionary result about the sufficiency of a single scalar observable.

1 Introduction

The interesting thing about a cellular automaton is not that a simple rule can produce a complicated picture. It is that a rule with no notion of the global object it is building — no representation of a colony, a spiral, a city, or a species — nevertheless builds one, and builds a specific one, whose properties can be measured and compared against a laboratory.

That comparison is the part that is usually missing. There is no shortage of Game-of-Life implementations on the web, and several of them are beautiful. Very few of them measure anything, and almost none of them let you take a configuration you found and hand it to somebody else exactly as you found it. Without measurement, a simulation is an illustration of a claim rather than a test of one; without reproducibility, it is not an instrument at all.

This paper describes **morphogen**, which attempts both. It contains ten models, chosen so that they form a single argument rather than a menu: rules, then chaos, then chemistry, then colonies, then ecosystems, then societies, then criticality. The question they are arranged around is deliberately naive, and it is the one that makes cellular automata worth taking seriously in the first place:

> *How does something alive — a bacterial colony, an ecosystem, a society — find food, survive, adapt, and organise itself, when nothing in it can see further than its*

own neighbours?

Contributions.

1. A portable C99 simulation core with a common model interface, compiled both natively (for testing and profiling) and to WebAssembly (for the browser), with an observable bus that every model streams into.
2. Bit-exact reproducibility: a seed and a parameter vector determine a run completely, and both are carried in the URL fragment.
3. Validation against the primary literature, as executable assertions (§5), rather than against the implementation's own prior output.
4. A rendering method that treats the display as a printing press rather than a screen: each field is a halftone separation at its own screen angle (§2.4).
5. An in-browser sweep engine: a worker pool that runs a model headlessly across a parameter grid, so that the phase diagrams in this paper can be regenerated by a reader with nothing but a browser.
6. Negative results (§6), including one that we think is generally useful: a fractal dimension in the right range is weak evidence that a growth model is right.

2 Architecture

2.1 The core is not a web project

`src/core/` is portable C99 and has no dependency on Emscripten, on JavaScript, or on the existence of a browser. It compiles with the system compiler and runs on the command line. This is the single most useful structural decision in the project: it means the models can be unit-tested, profiled with native tools, swept over parameter ranges at full speed, and checked against the literature in continuous integration without a browser anywhere in the loop. WebAssembly is one target among several, and the figures in this paper are produced by another.

Every model implements one interface: a parameter table (identifier, range, step, default, and a one-line explanation), an initialiser, a step function, an observable vector, and up to four *ink* channels. The shell — the renderer, the parameter panel, the plots — knows nothing else about any model. The parameter table is also the only place in the project where a parameter's range is written down; the control panel in the browser is generated from it at run time, so the interface cannot drift out of step with the science.

2.2 Determinism

Reproducibility is not a nice property here, it is the product. The following are enforced:

- PCG32 [12] for all sequential randomness, seeded per logical stream, so that adding a random draw in one place does not perturb the numbers everywhere else.
- A counter-based hash (splitmix64) where a rule is defined asynchronously and the visiting order must not matter.
- No wall-clock time in any update. The speed control changes how many generations are computed per animation frame; it never changes what happens in one.

- Agent-list compaction after a step rather than swap-removal during one. Swap-removing mid-iteration silently reorders the agents that have not yet acted, which makes the run a function of the death pattern as well as the seed.

Consequently a run is identified by `(model, seed, parameters, generation)`, all of which are serialised into the URL fragment. Copying the link shares the run, not a picture of it.

2.3 The boundary

The simulation state lives in the WebAssembly heap and is never copied out of it. C writes the ink channels into a buffer; JavaScript takes a typed-array view directly onto those bytes and hands the view to `texSubImage2D`. The only copy in the pipeline is the driver's upload to video memory. The boundary is crossed three times per frame — step, render, read observables — and never per cell; the exports are raw WebAssembly symbols rather than `ccall` or Embind wrappers.

Memory growth is disabled (`-sALLOW_MEMORY_GROWTH=0`) with a generous initial heap. This is deliberate and worth stating, because the alternative fails in a way that is very hard to diagnose: a `memory.grow` detaches every typed-array view held into linear memory, including the one the renderer uploads from, and the symptom is a blank canvas with no error reported anywhere. Allocating once removes the failure mode rather than working around it.

2.4 Rendering: the display as a press

Several of the models carry more than one field — three species in cyclic competition; bacteria, nutrient and a chemorepellent in the colony — and overlaying them is a real problem, not a cosmetic one. Printing solved it a century and a half ago. Each field is rendered as its own halftone screen: a lattice of dots whose radius encodes the local value, laid down at its own screen angle, so that the screens interleave into a rosette instead of colliding into a moiré. Dot radius goes as $\sqrt{v}$ rather than v, because a dot's optical density is proportional to its area; a linear mapping crushes the midtones. A single-field model prints at zero degrees, so a glider still looks like a glider.

The whole pass is one fragment shader, and the simulation's ink buffer is its only input.

3 The models

The ten are listed in Table 1. They are not a menu; they are arranged as an argument, running from the simplest rules that can be written down to systems that regulate themselves. Each is implemented from its source paper rather than from a secondhand description, with that paper's parameters, boundary conditions and update order — which matters more than it sounds, and §6 gives three cases where it mattered a great deal.

4 Results

4.1 A colony's shape is set by its food

The colony model is a nutrient field on a no-flux lattice (a Petri dish is closed) coupled to cells that consume it, bank the energy, divide when they have enough, and sporulate into an

Model	Source	What it demonstrates
Bacterial colony	[2, 7]	Food alone decides whether growth is a fractal or a disc.
Gray–Scott	[13]	Two chemicals, no biology, and spots that divide.
Lenia	[3]	A self-maintaining, self-propelled creature in a continuous field.
Cyclic competition	[14, 9]	Biodiversity as a phase transition in mobility.
Sugarscape	[6]	A fair world producing an unfair distribution.
Segregation	[15]	Mild preferences composing into extreme outcomes.
Life-like	[8]	The whole 2^{18} B/S rulespace in one branchless kernel.
Elementary CA	[18, 4]	Eight bits of rule; one of them is Turing complete.
Langton's ant	[10]	Chaos for 10^4 steps, then, unprompted, a highway.
Forest fire	[5, 1]	Self-organised criticality, and what destroys it.

Table 1: The ten models.

inert scaffold when they cannot cover their metabolic cost. The cells are static: the colony is a structure, not a swarm. Growth happens only where the thin living rind touches food.

Bacteria eat, so a colony digs a depletion halo around itself. A bump on the growth front protrudes into fresher nutrient than the flat parts do, eats better, grows faster, and becomes a bigger bump: the Mullins–Sekerka instability, the same mathematics as a snowflake or a lightning strike. Starve the colony and the instability wins and the front advances in fingers. Feed it and diffusion refills the halo as fast as the cells empty it, no bump gains an advantage, and the colony fills in.

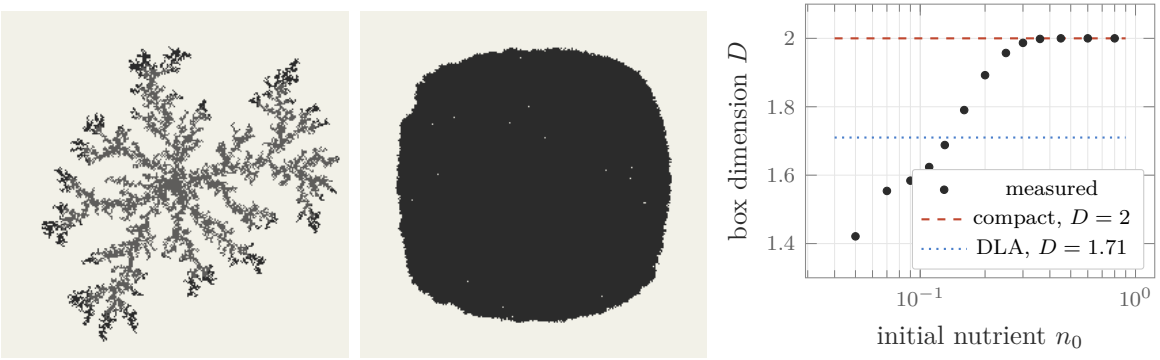
Figure 1 shows both limits and the transition between them, measured by box counting.

The two leftmost points of the curve should be read with care. Below $n_0 \approx 0.07$ the colony never exceeds a few thousand cells, and box counting on a cluster that small is biased downwards by the finite lattice: the apparent $D = 1.42$ at $n_0 = 0.05$ reflects the size of the object as much as its geometry. The regime in which the estimate is trustworthy — where the cluster spans enough octaves for a scaling range to exist — is roughly $0.10 \leq n_0 \leq 0.45$, and it is across that range that the dimension moves from fractal to compact.

It is worth being precise about the cause, because "the colony branches because it is hungry" is not quite right. Branching requires the instability, and the instability requires that the nutrient field around the colony relax into a quasi-static harmonic field, so that tips see far more of it than hollows do. That in turn requires a long *diffusion length*: nutrient must travel a good distance between one division and the next. Figure 2 holds the food fixed and varies only the diffusion length, and the morphology moves anyway. Starve a colony but let it eat only what is immediately in front of it, and it does not branch — it starves compactly.

4.2 Mild preferences, segregated cities

Schelling's households look at their eight neighbours and move if fewer than a fraction τ of them are of their own kind. A randomly mixed city of two equal groups scores 0.5 on the segregation index by chance. The system halts as soon as every household is content, so τ does not merely influence the outcome — it *sets* it, and the city segregates exactly as far as it must and no further.



(a) $n_0 = 0.10$: starved **(b)** $n_0 = 0.45$: fed, at gen. 250 **(c)** D against food

Figure 1: The colony's morphology is set by one environmental number. Nothing about the cells is different between (a) and (b). D falls monotonically as food is withdrawn, crossing the diffusion-limited-aggregation value near $n_0 \approx 0.14$ and reaching $D = 2.00$ above $n_0 \approx 0.40$. The fed colony (b) is caught at generation 250, while its growth front still exists: left alone it reaches the walls of the dish by generation 450 and the picture becomes a solid black square, which is true and shows nothing. What matters in (b) is that the front is smooth and convex — there are no fingers to see. Witten and Sander derived $D = 1.71$ for DLA from an argument about random walkers with no biology in it [17]; Fujikawa and Matsushita measured $D \approx 1.73$ on a plate of *Bacillus subtilis* [7].

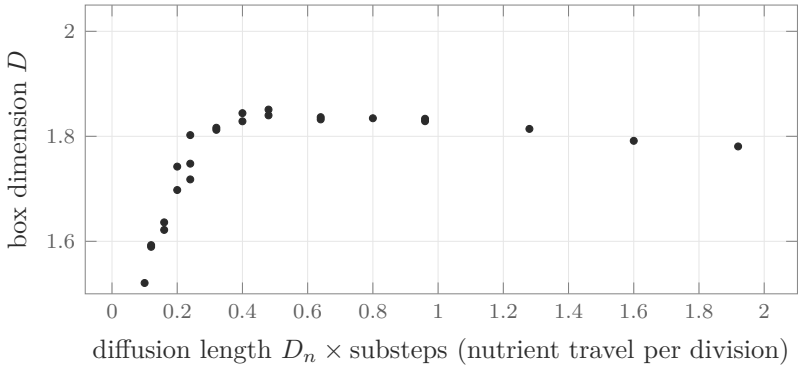


Figure 2: Food held fixed at $n_0 = 0.16$; only the diffusion length varies. The colony is equally hungry throughout. Branching is not caused by hunger; it is caused by hunger *plus* a diffusion field long enough for a tip to exploit.

The result is the gap between the two axes of Figure 3. At $\tau = 0.35$ — a population in which everyone is perfectly content to be outnumbered nearly two to one, which is more tolerant than any society that has ever existed — the city still settles at a segregation index of 0.771 ± 0.002, against a chance baseline of 0.50. The curve has already left the baseline by $\tau = 0.15$ and is at 0.93 by $\tau = 0.50$. There is no threshold of intolerance below which the city is safe.

4.3 A fair world and an unfair distribution

Sugarscape's rules are scrupulously fair: agents inherit nothing, cheat nobody, and draw their vision, metabolism and starting endowment from the same distributions. Figure 4 shows the Gini coefficient of the wealth distribution over time, from an initial condition in which everybody has roughly the same. Two things emerge that nobody specified: a carrying capacity, and inequality.

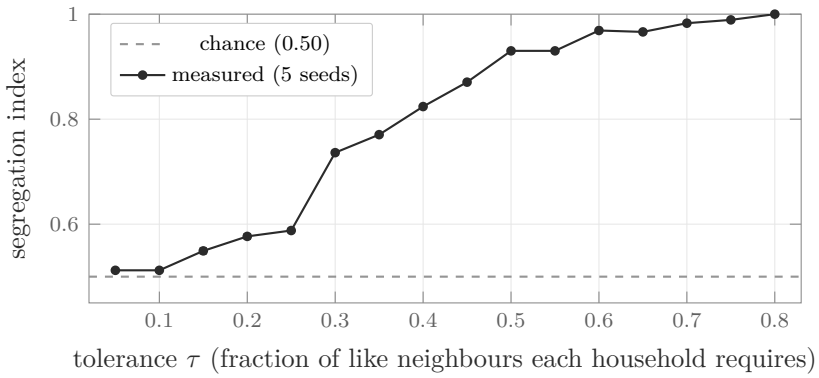
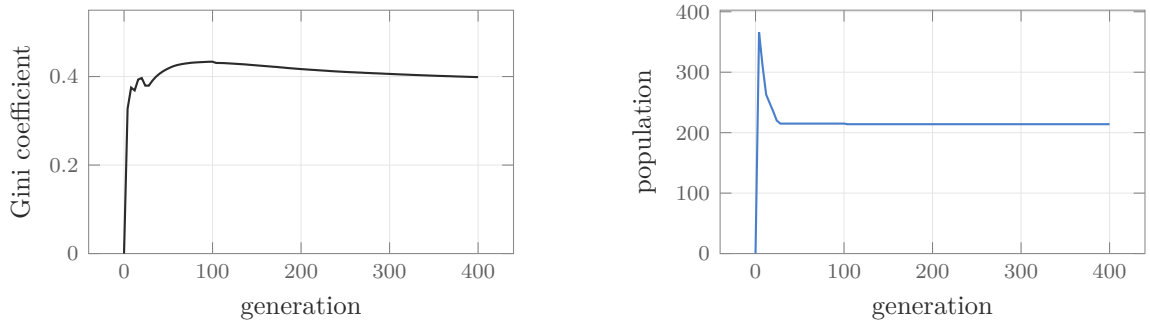


Figure 3: Segregation against tolerance, after 500 rounds on a 128^2 lattice with 15% vacancy. The curve leaves the chance baseline long before $\tau = 0.5$. Nobody in this model prefers a segregated city; the city segregates regardless.



(a) inequality **(b)** carrying capacity

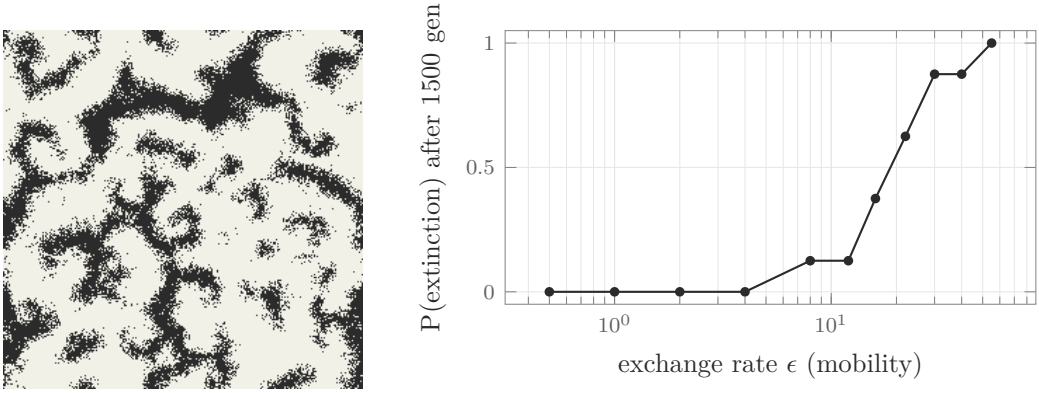
Figure 4: Sugarscape, from an initial population of 400 with endowments drawn uniformly from $[5, 25]$. (a) The Gini coefficient of the wealth distribution climbs away from equality within a few tens of generations, peaks at 0.434, and settles at 0.40 — which is roughly the Gini of a real economy. Nothing in the rules distributes anything unfairly. (b) The population overshoots, crashes, and settles at 214 agents: a carrying capacity that is a property of the landscape and of nothing anybody decided.

4.4 Biodiversity as a phase transition

Three species in a cycle where none is strongest. In a well-mixed system this ends in a monoculture. On a lattice it need not — and whether it does is decided by a single number, the rate at which neighbours exchange places. Reichenbach, Mobilia and Frey showed that below a critical mobility the system organises into interlocking spiral waves and all three species persist indefinitely; above it, the spiral wavelength (which grows as $\sqrt{M}$) exceeds the system size, the pattern washes out, and biodiversity collapses [14]. Figure 5 reproduces the transition: on a 128^2 lattice, all three species survive every run up to $\epsilon = 4$, and by $\epsilon = 55$ every run has collapsed to a monoculture, with the crossover near $\epsilon \approx 18$.

4.5 Classifying all 256 elementary rules

Wolfram sorted the elementary rules into four classes by eye [18]. Wuensche's input entropy does it mechanically and, more importantly, *quantitatively* [19]: at each step, take the Shannon entropy of the histogram of how many cells matched each of the eight neighbourhood patterns. Rules that freeze drive it low and flat; rules that boil hold it high and flat; the interesting rules sit in between with a *large variance*, because the entropy jerks around as coherent structures form, collide and die. The variance of the input entropy is a glider detector.



(a) $\epsilon = 2$: coexistence

(b) Extinction probability against mobility (8 seeds per point)

Figure 5: One slider takes the system from a living ecosystem to a monoculture. The three strains are not a metaphor: Kerr *et al.* played this game with real *E. coli*, and got the same answer — in a flask one strain took over, on a plate all three persisted [9].

Figure 6 places all 256 rules in that plane, and the statistic does most of what it claims. The fully chaotic rules cluster at the right-hand edge: rule 30 and rule 90 both sit at a mean entropy of $\approx$2.98 bits (of a possible 3) with a variance of order 1×10^{-4}, which is to say they boil steadily and never stop boiling. Rules with intermediate entropy ($2.7 < H < 2.95$) have a median variance of 4.1×10^{-4}, five times that of the fully chaotic band (8×10^{-5}), and it is in this intermediate band that the two rules Wolfram singles out as class IV are found: rule 110, which Cook proved Turing complete [4], at $H = 2.907$, Var $= 4.3 \times 10^{-4}$, and rule 54 at $H = 2.908$, Var $= 3.4 \times 10^{-3}$.

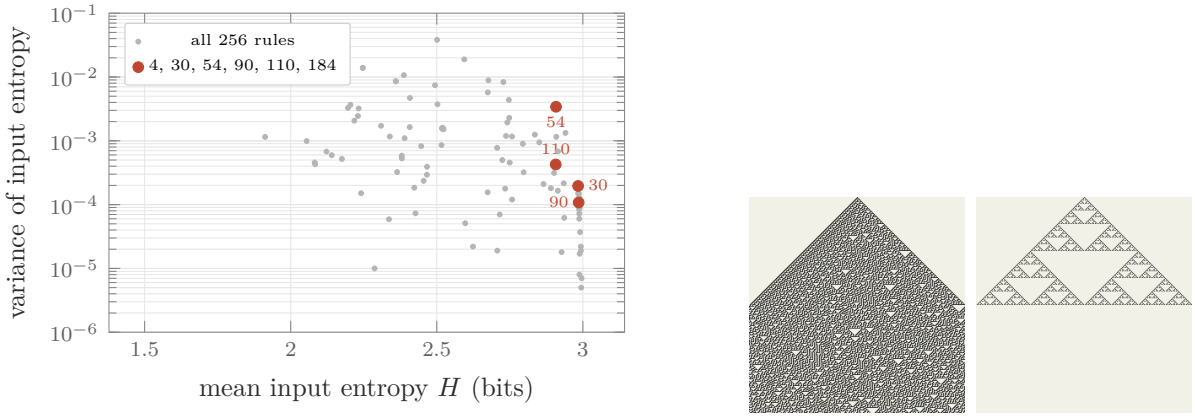
But we should be honest about what the statistic does *not* do, because our own figure shows it. The very highest variances in the plane do not belong to rule 110 at all; they belong to rules 1, 5 and 127, whose entropy simply oscillates with period two because the whole lattice inverts every step. A variance is a variance: it cannot tell an entropy that jumps because gliders are colliding from an entropy that jumps because the pattern is blinking. What rescues the classifier is the *other* axis — those rules sit at low mean entropy (1.5–1.7 bits) and are nowhere near the complex band — so the two coordinates together separate the classes even though neither does alone. It is a good illustration of why one measures two things and plots them against each other rather than reporting a single number.

4.6 Lenia: the niche is a filament

Lenia [3] dissolves every discrete feature of Conway's Life: the state is real-valued, the neighbourhood is a smooth ring-shaped kernel, the birth and survival sets become a smooth growth function, and the time step is small.

$$A_{t+\Delta t} = \left[\, A_t + \tfrac{1}{T}\, G(K * A_t) \,\right]_0^1, \qquad G(u) = 2\exp\!\left(-\frac{(u-\mu)^2}{2\sigma^2}\right) - 1. \tag{1}$$

What survives the dissolution is a creature. *Orbium unicaudatus* holds its shape, moves under its own power in a direction of its own choosing, and repairs itself when damaged — and it dies if the parameters leave a narrow region of the (μ, σ) plane. Figure 7 maps that region. It is a filament. The claim that life sits at an "edge" between order and chaos is usually made rhetorically; here it is a measurement, and the edge is about six thousandths of σ wide.



(a) all 256 rules **(b)** rule 30 **(c)** rule 90

Figure 6: Automatic classification of the whole elementary rulespace (log variance axis). Ordered rules collapse to the bottom left; the fully chaotic rules — rule 30 and rule 90 among them — pile up at the right-hand edge with a steady, high entropy and almost no variance in it. The class IV rules sit to their left, at intermediate entropy and higher variance — rule 54 by an order of magnitude, rule 110 only by a factor of two, which is a thinner margin than the method's reputation suggests. The outliers at the top left are not complex; they are rules whose lattice inverts every step, so their entropy oscillates with period two — which is why the classifier needs both coordinates and not just the variance. Rule 90 (c), from a single live cell, is the Sierpiński triangle, and box-counts to $D = 1.585 = \log 3 / \log 2$.

4.7 Self-organised criticality, and what destroys it

The Drossel–Schwabl forest fire [5] needs no parameter tuned to a critical value: given a wide separation between the growth rate p and the lightning rate f, the forest drives *itself* to a critical density and stays there, and fires arrive with no characteristic size. Figure 8 shows the fire-size distribution in that regime and in one where the separation of timescales has been destroyed. The power law is not a property of the rules. It is a property of the rules *plus* the separation of timescales, and it disappears when the separation does.

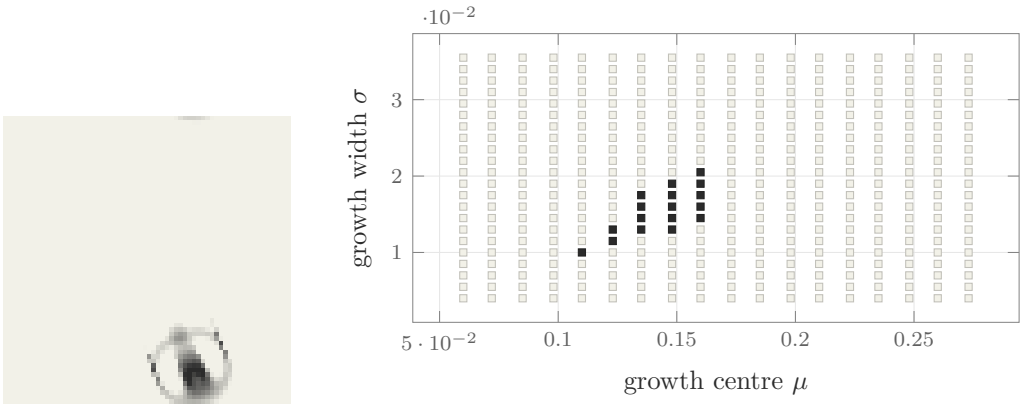
The two regimes are separated most sharply not by the number of fires but by their extremes. Over 20 000 generations on a 256^2 lattice, the critical regime ($p = 0.02$, $f = 2 \times 10^{-5}$) produced 11 008 fires, the largest of which consumed 221 702 trees — many times the number standing at any one moment, since the fire outran the regrowth across the entire lattice. The driven regime ($f = 1 \times 10^{-3}$) produced forty times as many fires, 442 025 of them, and the largest consumed 1320. Frequent lightning does not make the forest more flammable. It makes it *safe*, by never letting it become connected enough to burn properly — and in doing so it abolishes the scale-free behaviour entirely.

5 Validation

The models are not checked against their own previous output. They are checked against numbers that other people measured, several of them with actual bacteria. These are assertions in `tests/main.c`, they run natively, and continuous integration runs them on every push. When one of them drifts, it is a bug, not a tolerance to be widened.

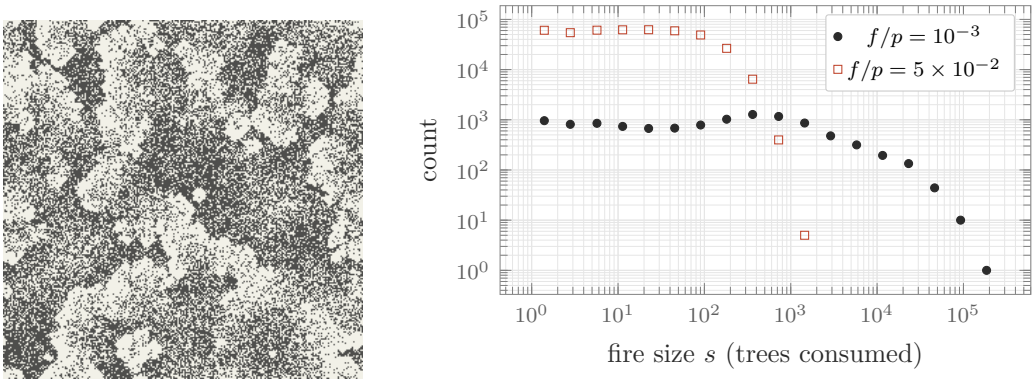
6 Negative results

The failures were more instructive than most of the successes, and three of them generalise.



(a) Orbium **(b)** where Orbium survives 400 steps

Figure 7: Dark squares: the creature is still there, still bounded, still itself, after 400 steps. Light: it either evaporated or bloomed into structureless mush that fills the world. Both are deaths. Life occupies the filament between them.



(a) $f/p = 10^{-3}$ **(b)** fire-size distribution

Figure 8: Fire sizes, binned by octave. The near-horizontal black distribution is the signature, not a contradiction of it: with logarithmic bins, a power law $P(s) \sim s^{-\tau}$ with $\tau \approx 1.15$ gives a count per octave going as $s^{1-\tau} = s^{-0.15}$, which is almost flat. Fires of every size occur with comparable frequency across five decades, until the finite lattice cuts the tail off. The red distribution, with lightning fifty times more frequent, has a scale: it stops dead at a few hundred trees.

A fractal dimension in the right range is weak evidence. The first version of the colony model let its cells wander freely inside a lubricated envelope, in the spirit of Ben-Jacob's communicating walkers. It produced a diffuse cloud of live cells, and that cloud box-counted to $D \approx 1.71$ — squarely on the Witten–Sander value, and squarely wrong. A scattered point cloud is fractal too. The test passed; the physics was absent. The model only started producing colonies when the cells were made *static*, so that the structure was a scaffold of spent cells with growth confined to a thin living rind. We report this because the scalar was doing exactly what a scalar does: it summarised, and in summarising it discarded the distinction between a colony and a gas. We now look at the images as well, and we recommend that anybody validating a growth model against D alone does the same.

Surface diffusion is not surface tension without a coordination bias. We attempted a second morphological axis — agar hardness, following Matsushita — implemented as surface diffusion: a newborn cell takes a few steps along the colony's surface before settling. It did not smooth the interface. It merely dumped daughters somewhere arbitrary, half of them in a starved crevice where they died, so the colony grew more slowly and no rounder. Adding

Model	Quantity	Expected	Source
Colony (starved)	box dimension	fractal, $\to 1.7$	[17, 7]
Colony (fed)	box dimension	2.0 (compact)	—
ECA rule 90	box dimension	$\log 3/\log 2 = 1.585$	[18]
Life, random soup	ash density	≈ 0.03	—
Life, glider	cells after 64 gen	5 (and translated)	[8]
Sugarscape	Gini from equal starts	0.4–0.5	[6]
Schelling, $\tau = 0.35$	segregation index	≈ 0.77 (chance: 0.50)	[15]
Schelling, $\tau = 0.50$	segregation index	> 0.85	[15]
Lenia, Orbium	mass after 400 steps	bounded, nonzero	[3]
Lenia, $\sigma = 0.005$	mass after 300 steps	zero (dies)	[3]
Cyclic competition	species at high mobility	< 3 (extinction)	[14]
Forest fire	tree density	self-organises, $0.2 < \rho < 0.75$	[5]
All models	re-run from same seed	bit-identical	—

Table 2: The literature, as a test suite.

a preference for sites of high coordination (more occupied neighbours, as an adatom prefers a kink site) made the sliding smooth the interface, but at the cost of pulling daughters *inward*, which stalled the growth front entirely. We removed the axis rather than ship a control that does not do what its label says. The paper and the interface now claim only the axis we can demonstrate: the diffusion length (Figure 2).

The obvious way to count a fire destroys the power law. In the forest fire, treating every cell burning anywhere on the map at a given moment as one fire — closing the books when the last ember goes out — seems harmless. It is fatal. In precisely the regime the model exists to study, two independent fires are almost always burning somewhere at once, so the map is never cold, so the counter never closes, and 1500 generations report a single enormous fire. The power law, which is the entire point, vanishes. Each ignition must open a numbered fire, and its number must be inherited by every tree it spreads to.

A note on Lenia's kernel. Chan's paper presents the exponential kernel core $\exp(4 - 1/(r(1-r)))$ and a Gaussian growth function, and that is the form every secondhand description repeats. The reference implementation defaults to the *polynomial* $(4r(1-r))^4$ and a polynomial growth function, and every creature in the species list — Orbium included — was found under the polynomial. Loaded under the exponential core, Orbium does not glide; it wobbles and dies. We ship both, with the polynomial as the default, because that is the physics the animals actually live in.

7 Limitations and further work

Most of the figures in this paper can now be regenerated by the reader, in the browser, without a toolchain: a pool of Web Workers — each with its own WebAssembly instance, since `SharedArrayBuffer` is unavailable on GitHub Pages — runs the model headlessly across a parameter grid, and each model carries a *reproduce the figure* button that sweeps its own source paper's figure. Schelling's segregation curve (Figure 3) takes about four seconds; Orbium's survival map (Figure 7), which is 168 independent runs of a convolutional field, takes about forty. The resolutions are lower than the ones printed here, and the sweeps start from whatever the reader has on the sliders rather than from the defaults, so they explore the neighbourhood of the run in front of them. This was the architecture's whole purpose: a core with no browser

dependency can be stepped by anything, including nothing.

What remains: the sweeps report a single scalar per point, so a phase diagram knows nothing about *how* a run failed, only that it did — and §6 is an argument about why that is a dangerous way to look at a growth model. The honest fix is to keep the pictures alongside the numbers, and at present the browser only shows the numbers.

Lenia's convolution is direct rather than FFT-based, which caps the practical kernel radius. Cross-origin isolation is unavailable on GitHub Pages, so `SharedArrayBuffer` and therefore WebAssembly threads are off the table; the simulation runs on the main thread under a wall-clock budget. Neither is fundamental.

8 Conclusion

A cellular automaton is worth taking seriously when you can measure it, argue with it, and hand it to somebody else exactly as you found it. The models here are small — the whole simulation core is a few thousand lines of C, and it compiles to 64 KB of WebAssembly — but they reproduce, from local rules and nothing else, a fractal dimension that somebody obtained from a Petri dish, a wealth distribution that nobody designed, a segregated city that nobody wanted, and a creature that repairs itself.

References

[1] Bak, P., Tang, C. & Wiesenfeld, K. Self-organized criticality: An explanation of $1/f$ noise. *Phys. Rev. Lett.* **59**, 381–384 (1987).

[2] Ben-Jacob, E., Schochet, O., Tenenbaum, A., Cohen, I., Czirók, A. & Vicsek, T. Generic modelling of cooperative growth patterns in bacterial colonies. *Nature* **368**, 46–49 (1994).

[3] Chan, B. W.-C. Lenia — Biology of Artificial Life. *Complex Systems* **28**(3), 251–286 (2019). arXiv:1812.05433.

[4] Cook, M. Universality in Elementary Cellular Automata. *Complex Systems* **15**, 1–40 (2004).

[5] Drossel, B. & Schwabl, F. Self-organized critical forest-fire model. *Phys. Rev. Lett.* **69**, 1629–1632 (1992).

[6] Epstein, J. M. & Axtell, R. *Growing Artificial Societies: Social Science from the Bottom Up.* Brookings Institution Press / MIT Press (1996).

[7] Fujikawa, H. & Matsushita, M. Fractal growth of *Bacillus subtilis* on agar plates. *J. Phys. Soc. Jpn.* **58**, 3875–3878 (1989).

[8] Gardner, M. Mathematical Games: The fantastic combinations of John Conway's new solitaire game "life". *Scientific American* **223**(4), 120–123 (1970).

[9] Kerr, B., Riley, M. A., Feldman, M. W. & Bohannan, B. J. M. Local dispersal promotes biodiversity in a real-life game of rock–paper–scissors. *Nature* **418**, 171–174 (2002).

[10] Langton, C. G. Studying artificial life with cellular automata. *Physica D* **22**, 120–149 (1986).

[11] Matsushita, M. *et al.* Interface growth and pattern formation in bacterial colonies. *Physica A* **249**, 517–524 (1998).

[12] O'Neill, M. E. PCG: A family of simple fast space-efficient statistically good algorithms for random number generation. Harvey Mudd College, HMC-CS-2014-0905 (2014).

[13] Pearson, J. E. Complex patterns in a simple system. *Science* **261**, 189–192 (1993).

[14] Reichenbach, T., Mobilia, M. & Frey, E. Mobility promotes and jeopardizes biodiversity in rock–paper–scissors games. *Nature* **448**, 1046–1049 (2007).

[15] Schelling, T. C. Dynamic models of segregation. *J. Mathematical Sociology* **1**(2), 143–186 (1971).

[16] Turing, A. M. The chemical basis of morphogenesis. *Phil. Trans. R. Soc. B* **237**, 37–72 (1952).

[17] Witten, T. A. & Sander, L. M. Diffusion-limited aggregation, a kinetic critical phenomenon. *Phys. Rev. Lett.* **47**, 1400–1403 (1981).

[18] Wolfram, S. Statistical mechanics of cellular automata. *Rev. Mod. Phys.* **55**, 601–644 (1983).

[19] Wuensche, A. Classifying cellular automata automatically: finding gliders, filtering, and relating space-time patterns, attractor basins, and the Z parameter. *Complexity* **4**(3), 47–66 (1999).