Digital Ecosystems: Interactive Multi-Agent Neural Cellular Automata

Luke Darlow[1]

[1]Sakana AI, Tokyo, Japan
luke@sakana.ai

Abstract

Neural cellular automata (NCAs) are well-studied for single-species morphogenesis and texture generation. Multi-agent settings, where several differentiable species compete on a shared grid via *online* gradient descent, have received far less attention; to our knowledge, only the Petri Dish NCA (PD-NCA) framework (Zhang et al., 2025) trains multiple species this way, and no prior work has made such a system directly interactive. **This work centres on interactivity.** We present an open-source, browser-based platform that supports a *checkpoint-branch-explore* workflow: observe a live digital ecosystem, save promising states, adjust parameters, and compare divergent outcomes from the same starting point. Because species are trained online, parameter changes propagate through the learning loop—not just the forward dynamics—so interactive steering is qualitatively richer than in rule-based ALife tools. We introduce algorithmic updates to PD-NCA including a tunable soft growth gate, spatial concentration, local win-rate feedback, and emergency respawn. Five case studies exercise the platform across regime transitions, emergent cooperation, optimiser comparison, and environmental construction. Static figures necessarily omit the live dynamics; the interactive platform is the primary contribution.

Introduction

Neural Cellular Automata have demonstrated single-species morphogenesis (Mordvintsev et al., 2020), texture synthesis (Niklasson et al., 2021), and self-repair. Recent work has extended NCAs to multi-agent competition on shared grids (Plantec et al., 2023; Barbieux and Canaan, 2024; Randazzo et al., 2023; Zhang et al., 2025), but researchers still typically study these dynamics (attractors, transients, path-dependent regime transitions) as offline batch experiments. Such workflows miss transient phenomena and prevent investigation of unexpected behaviour as it appears.

In this paper we present an open-source, browser-based platform that makes a multi-agent NCA 'digital ecosystem' directly manipulable. Building on PD-NCA (Zhang et al., 2025), the platform supports a **checkpoint-branch-explore** workflow: observe dynamics in real time, save promising states, adjust parameters, and compare divergent outcomes from the same starting point. The workflow is inspired by Picbreeder (Secretan et al., 2011), where collaborative branching discovered artefacts that automated search could not find; we apply the same idea to dynamic ALife ecosystems. Our contributions are: (1) an interactive web platform with a real-time timeline dashboard and checkpoint slots; (2) algorithmic improvements to PD-NCA, including a tunable growth gate, spatial concentration, local win-rate feedback, and emergency respawn; and (3) five case studies—spanning regime transitions, emergent cooperation, optimiser comparison, and environmental construction—that exercise the platform's checkpoint-branch-explore workflow in distinct ways. The platform runs entirely in the browser via TensorFlow.js (Smilkov et al., 2019) and WebGL, with no installation, at `https://pub.sakana.ai/digital-ecosystem`.

Background

Neural cellular automata. A neural cellular automaton (NCA) is a 2D grid in which every cell carries a small state vector and a shared neural network repeatedly maps each cell's local neighbourhood to a state update; iterating produces emergent global structure from local rules. Mordvintsev et al. (2020) showed such networks can be trained end-to-end to grow target patterns and recover from damage, with extensions to texture synthesis (Niklasson et al., 2021), goal-conditioned control (Sudhakaran et al., 2022; Grasso and Bongard, 2022), and high-resolution rendering via implicit decoders (Pajouheshgar et al., 2025).

Multi-agent and ecosystem NCAs. Lenia (Chan, 2019) extended CAs to a continuous substrate supporting emergent species; Flow-Lenia (Plantec et al., 2023) added parameter localisation so multiple species share a mass-conserving grid. Coralai (Barbieux and Canaan, 2024) evolves ecosystems of embodied NCAs with competition and symbiosis, Biomaker CA (Randazzo et al., 2023) explores multi-agent grid worlds under metabolic constraints, and Agüera y Arcas et al. (2024) show that self-replicating programs emerge spontaneously from simple computational substrates without an explicit fitness landscape. These systems use evolutionary or rule-based dynamics; none trains multi-

ple competing species by gradient descent during simulation. Zhang et al. (2025) introduced Petri Dish NCA (PD-NCA), which does: N neural species compete for territory via attack/defence vectors with *online* backpropagation. Ours is the first work to make such a system directly interactive in real time: because species are trained online, parameter changes propagate through the learning loop, making interactive steering qualitatively richer than in rule-based ALife tools.

Interactive and open-ended ALife tools. NetLogo (Wilensky, 1999), ALIEN (Heinemann, 2024), and the Lenia browser demos provide accessible ALife exploration environments. Interactive evolutionary computation (Takagi, 2001) and Picbreeder (Secretan et al., 2011) showed that human-guided branching discovers artefacts inaccessible to automated search; recent automated open-ended search (Kumar et al., 2025; Packard et al., 2019) explores without human input. Existing interactive tools generally lack the ability to checkpoint state, branch into alternative trajectories, and share reproducible recipes for an *online-trained* multi-agent ecosystem, which our platform provides.

Method Updates from PD-NCA

PD-NCA in brief. In PD-NCA (Zhang et al., 2025), N neural species share a 2D grid alongside a *sun* background. At each pixel every species emits an *attack* vector $\mathbf{a}_{\text{att},i}$ and a *defence* vector $\mathbf{a}_{\text{def},i}$; the dot product $\langle \mathbf{a}_{\text{att},i}, \mathbf{a}_{\text{def},j} \rangle$ is summed over defenders j to give each attacker i a total score, a softmax yields per-species competition weights, and all species are trained *online* by back-propagating a per-species growth loss through the simulation step. Each cell carries $[\mathbf{a} \,|\, \mathbf{a}_{\text{att}} \,|\, \mathbf{a}_{\text{def}} \,|\, \mathbf{h}]$ with $\mathbf{a} \in \mathbb{R}^{N+1}$ the aliveness vector for all N species plus the sun; a cell is alive for species i when a_i exceeds a *survival threshold* θ (default 0.5; PD-NCA used 0.4). We also treat the sun as a *learnable* competitor optimised jointly with the species, rather than a static environment.

Presence gating, walls, and sun penalty. Three spatial constraints share one implementation: a per-cell additive penalty before the softmax. *Presence gating*—a species only competes inside its dilated territory (a 3×3 max-pool of a_i); cells where $m_i<0.05$ receive -10^9. *Walls* take -10^9 for all species. An optional *sun handicap* reduces sun scores near species frontiers (default off). PD-NCA achieved grow-from-territory via an aliveness mask; we re-implement it as an additive penalty so walls, presence, and sun handicap share the same code path.

Growth gate. After the softmax winner is chosen, each species' new aliveness at every cell is further modulated by a *soft* presence mask with a tunable steepness. Writing $\tilde{a}_i(x, y)$ for the 3×3 max-pooled aliveness of species i at (x, y) and θ for the survival threshold, the growth gate is

$$g_i(x, y) \;=\; \sigma\big(k_{\text{gate}}\,(\tilde{a}_i(x, y) - \theta)\big), \tag{1}$$

and the species' new alpha channel becomes $a_i^{t+1} = w_i \cdot g_i$ with w_i the softmax competition weight. The default k_{gate}=20 gives a sigmoid 10%–90% transition zone of $\pm 2.2/k_{\text{gate}} \approx \pm 0.11$, effectively a step function producing the bistable territorial regime of the other case studies. In PD-NCA this steepness was hard-coded; exposing it as an interactive knob unlocks a distinct dynamical regime (see the growth-gate case study below).

Concentration. PD-NCA distributes attack/defence energy uniformly across a species' territory, preventing local focus on a hard boundary. We add a sliding-window softmax over local energy $E_i = \|\mathbf{a}_i\|_2 + \|\mathbf{d}_i\|_2$ in a $k\times k$ window (k=4, τ_c=0.4), $c_i(x, y) = \exp(E_i/\tau_c) / \sum_{\mathcal{N}_k} \exp(E_i/\tau_c)$, focusing competitive force on locally high-energy cells. An optional baseline b floors low-energy cells via $c_i' = b + (1-b)c_i$.

Win-rate feedback and stochastic updates. A per-cell EMA tracks maximum local aliveness, $r \leftarrow \alpha\, r + (1-\alpha)\, \max_i a_i$ (α=0.95), modulating an update probability $p_{\text{update}} = p_0(1 + s(r-0.5))$ clamped to $[0.1, 1]$ with p_0=0.56. This is a positive-feedback mechanism: cells with high aliveness (r high) update more frequently, reinforcing established territories, while weakly-held cells (r low) update less often, damping oscillatory boundary dynamics and breaking synchronous-update artefacts.

Loss and respawn. We minimise a soft-min over per-species mean aliveness with an entropy bonus, $\mathcal{L} = k^{-1} \log \sum_i \exp(-k\beta\, \text{asinh}(\bar{a}_i/\beta)) - w_d H(\mathbf{p}) / \log N$ (k=8, β=0.4, w_d=0.4). The soft-min focuses gradient on the weakest species, asinh compresses growth so small-population doublings are rewarded as much as large-population doublings, and the entropy bonus pushes toward uniform populations. Minority species also receive a competition-time score boost $\Delta_i = 10 w_d \max(0, 1/N - p_i)$ and a frontier preservation bonus below the occupancy floor. A hard *emergency respawn* fires when a species effectively dies ($\sum a_i < 1$), injecting five seed cells at random non-wall locations—a key delta from PD-NCA, which permitted terminal extinction.

Architecture and optimisation. We replace PD-NCA's convolutional blocks with MobileNetV2-style InvertedResidual blocks (Sandler et al., 2018), configurable in depth (0–10), width (8–256), and kernel size ($\{1, 3, 5, 7\}$). A shared trunk feeds a per-species grouped 1×1 decoder. For efficiency we use a *one-step lag*: inference runs on the full grid, then gradients are computed on a random 10%-area rectangular patch under the pre-update weights, saving a redundant forward pass at negligible visual cost.

Interactive Exploration Platform

The platform realises the **checkpoint-branch-explore** workflow described in the Introduction: observe, hypothesise, checkpoint, perturb, compare, revert. It runs entirely in the browser via TensorFlow.js, WebGL, and the SwissGL

rendering library (Mordvintsev, 2023), with no installation (Figure 1).

Parameter control. The interface organises 40+ adjustable parameters into collapsible folders; all take effect immediately without restarting the simulation. Drawing tools provide direct environmental manipulation: walls block growth and carve spatial niches, while individual species can be painted or erased to probe invasion and recolonisation.

Timeline. A multi-row Canvas2D dashboard renders a stacked area chart of species populations, with sparklines for normalised Shannon diversity and training loss. Checkpoint diamonds and parameter-change triangles overlay the timeline, so environmental edits can be correlated with population dynamics; hover tooltips and scroll-to-zoom expose the exact step at which a species begins to decline.

Checkpoints. We save and restore *complete* simulation state: the grid tensor, all network weights, optimiser buffers, metrics history, and every configuration value. Five user slots plus a rolling auto-save (every 500 steps) persist in IndexedDB across sessions. Restoring a checkpoint and then making different parameter choices yields an implicit controlled experiment with identical initial conditions. Automatic NaN detection rolls back to the most recent checkpoint, so numerical instability does not destroy progress.

Determinism caveat. Checkpoint restoration is exact, but the WebGL simulation step is not bitwise reproducible: two reruns from the same checkpoint over 94 additional steps disagree on $\sim 26\%$ of cell ownerships, concentrated at contested boundaries. By "controlled experiment" we mean identical initial conditions with qualitatively comparable trajectories; the effects reported below are well above this noise floor.

Exploration Case Studies

We exercise the platform through five case studies, each illustrating a different style of interactive exploration. All experiments use five species; grid size is 128×128 except 256×256 for the growth-gate study and 200×200 for biogeography. The studies progress from tuning a single simulator parameter (growth gate) through sweeping and scheduling training hyperparameters (extreme temperature, cooperation), to branching alternative optimiser configurations from a shared checkpoint, to constructing and modifying the spatial environment itself (biogeography). Two parameters recur across several studies: the *competition temperature* τ (softmax temperature over inter-species scores; low τ is winner-take-all, high τ permissive) and the *survival threshold* (minimum aliveness for a cell to count as alive).

The Growth Gate as a Learned Edge of Chaos

Most parameters accessible through the platform control training (learning rates, loss weights) or the environment (walls, seeding). The growth-gate steepness k_{gate} is different: it modifies the simulator's forward pass directly, reshaping the alive mask that determines which cells participate in each update. The default $k_{\text{gate}}=20$ makes the alive-channel field bistable, and the system behaves as a discrete-territory simulator. Relaxing the gate trades this locality guarantee for a distinct dynamical regime.

A learned Langton-λ. Reducing k_{gate} softens the gate so that stable *intermediate* alive values become possible, and the alive-channel field crosses into a regime resembling "Class IV" cellular automata (Wolfram, 1984). k_{gate} plays the role of Langton's λ parameter (Langton, 1990; Packard, 1988), but inside a *learning* system where species adapt to whatever value the user sets, so the sweet spot is reachable interactively rather than by exhaustive sweep (Agüera y Arcas et al., 2024).

Internal structure at the edge of chaos. At $k_{\text{gate}}=20$ the gate is effectively a step function and the alive field resolves into flat coloured tiles. Dropping k_{gate} into the edge band ($\sim$4.6–4.9) widens the sigmoid transition zone so many cells carry stable intermediate aliveness, the "texture" visible in Figure 2: a spatially-extended continuum of partial aliveness realised through the alpha channel, not binary territory. Once partial-aliveness cells exist, the per-cell update loop becomes dynamic and the alive field of each species continuously rearranges within its own territory.

At the edge, species exhibit distinct phenotypes within a single run: some hold attack-focused interiors with churning mini-domains where pockets of a rival appear and dissolve, while others maintain defence-focused territories with stable cores and negotiating boundaries. Sub-populations of one species persist as embedded enclaves inside another. The middle row of Figure 2 shows this texture regenerating autonomously over steps 733–757 without any parameter changes, confirming that the edge dynamics are intrinsic rather than transient. (The live platform at `https://pub.sakana.ai/digital-ecosystem` includes a pre-trained "Load Showcase" checkpoint that starts in this regime, allowing readers to explore the edge-of-chaos dynamics interactively.) These behaviours are qualitative observations from interactive exploration: we were able to steer the system into each of these states by hand, but we do not claim quantitative evidence for discrete regime boundaries.

Protocol and path dependence. Cold-starting at $k_{\text{gate}}\lesssim5$ collapses before any species can establish presence. The working protocol is to start at $k_{\text{gate}}=17.5$ (Figure 2, top-left) and step the slider down once the ecosystem has settled. The capture run used SGD with $\tau=1.01$, $\theta=0.51$, $\text{lr}=7.45\times10^{-3}$ on a 256×256 grid. Moves of 0.1–0.3 in k_{gate} push the system between regimes; the bottom row of Figure 2 walks three orthogonal perturbations from $k_{\text{gate}}=4.8$, showing that raising τ, raising k_{gate}, or relowering it each produce visibly different outcomes.

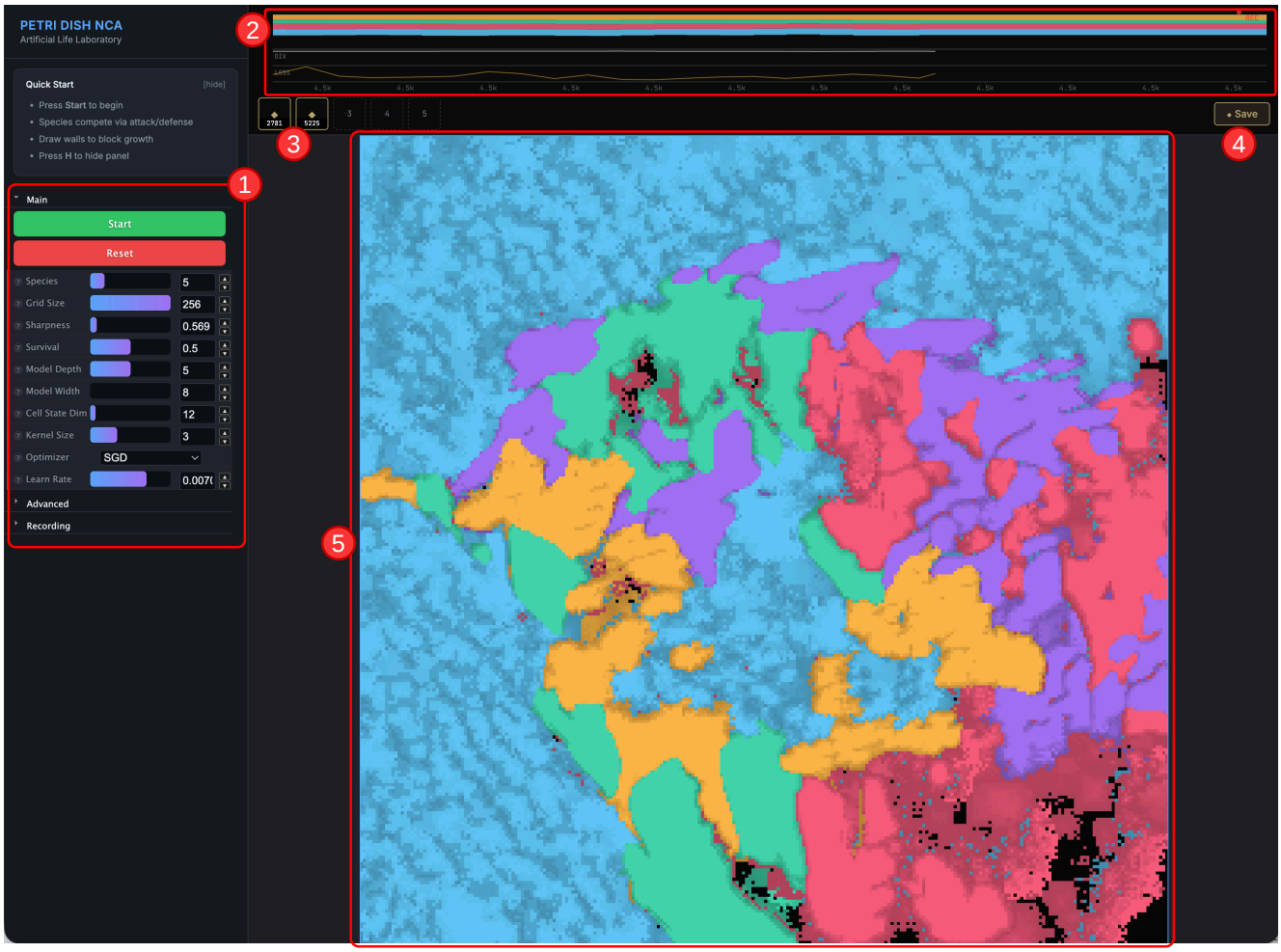


Figure 1: The interactive platform. **(1)** Control panel with 40+ adjustable parameters. **(2)** Timeline: population strata, diversity, loss, checkpoint/parameter-change markers. **(3)** Checkpoint tray for branching. **(4)** Save button. **(5)** Simulation canvas. The checkpoint tray and timeline markers together enable the checkpoint-branch-explore workflow. Live at `https://pub.sakana.ai/digital-ecosystem`.

Caveats. This is a single interactive run; within-run evidence (autonomous succession, re-excitation) supports the regime structure, but statistically rigorous replication remains future work. We do not claim novelty for edge-of-chaos dynamics in CAs; the contribution is that k_{gate} is a differentiable knob whose sweet spot is reachable interactively inside a learning multi-agent NCA.

Extreme Competition Temperature: a Flicker-Mixing Attractor

Setting $\tau = 0.1$ pushes the inter-species softmax close to one-hot (the GUI slider is labelled "Sharpness" but its value is τ; low τ is the winner-take-all extreme). Because species scores are nearly equal early in training, the per-cell winner flips every step as small fluctuations resolve differently. The survival threshold decides whether this flicker is visible or absorbed (Figure 3): at $\text{thr} \leq 0.1$ the system enters a *flicker-mixing* state with pixel-scale spatial mixing; at $\text{thr}=0.5$–0.6 larger-scale structure emerges; at $\text{thr}=0.8$ the stricter gate filters out the per-step flicker and recognisable fine-grained territories form, reaching a stable configuration that persists for thousands of steps. This appears to be a persistent regime the shared architecture cannot escape: near-tied scores resolve into rapidly-alternating winners, and only a strict gate hides the minority-winner moments long enough to yield visible territories.

The threshold as a spatial filter. The survival threshold acts as a temporal filter on the per-step winner signal. At low threshold, every cell passes the alive gate regardless of which species won the most recent step, so the raw flicker is directly visible. At high threshold, a cell must sustain local dominance across its 3×3 neighbourhood to remain alive;

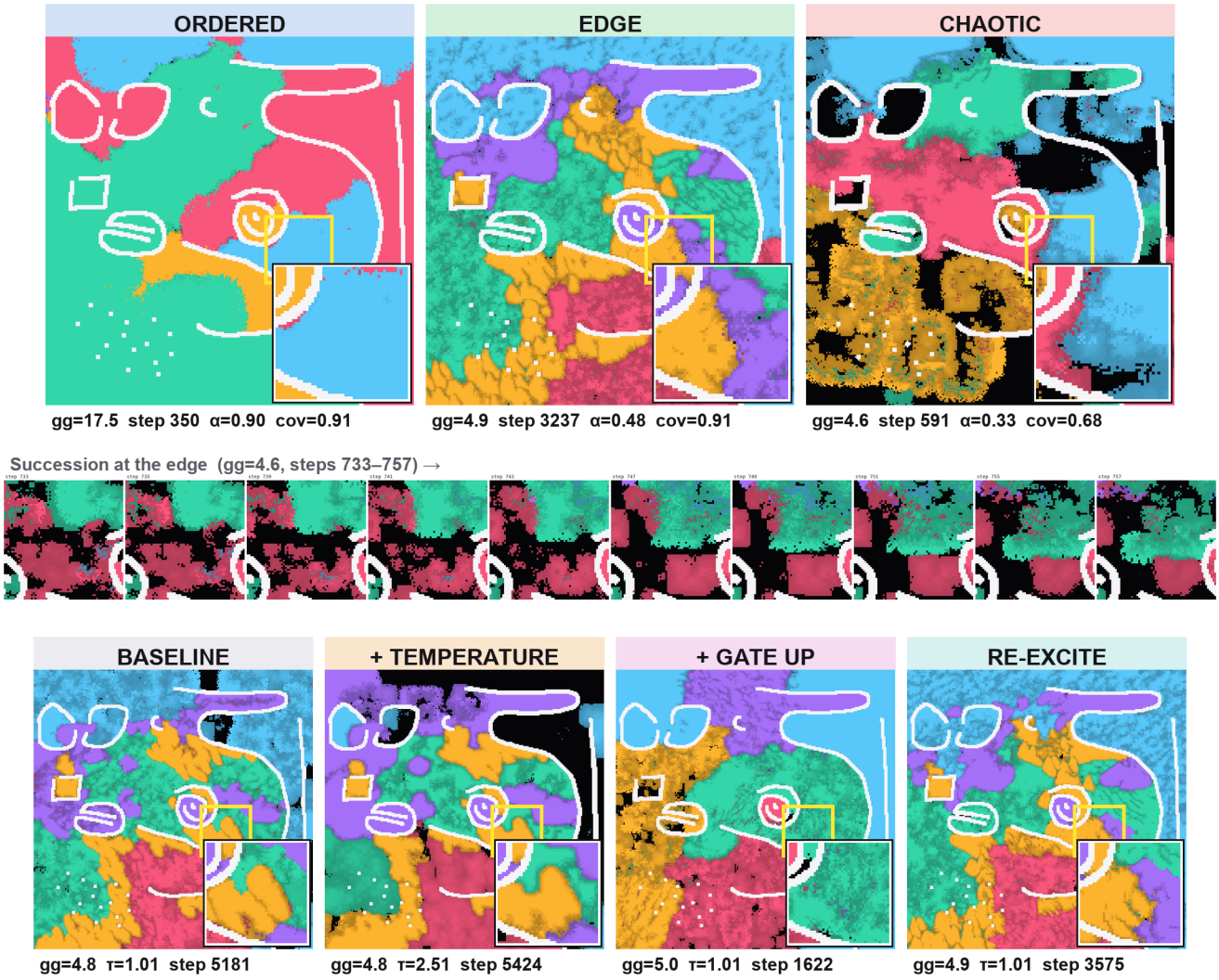


Figure 2: **Growth-gate steepness k_{gate} as an edge-of-chaos control. Top:** *Ordered* (k_{gate}=17.5), *Edge* (k_{gate}=4.9), *Chaotic* (k_{gate}=4.6). **Middle:** succession at k_{gate}=4.6 (steps 733–757); texture regenerates autonomously. **Bottom:** perturbations from k_{gate}=4.8: raising τ, raising k_{gate}, and re-lowering each produce different outcomes. Zoom insets share coordinates within each row.

transient flips kill weakly-held cells, so only cells with consistent local advantage survive (Figure 3c). Low threshold shows the instantaneous competition; high threshold shows only its durable outcome.

Relation to other case studies. The flicker-mixing regime is the degenerate extreme of the competition continuum: no stable spatial structure exists at any scale. This contrasts with the cooperative interleaving in the next case study, where species share territory via fine-grained dithering that persists for thousands of steps. Together with the growth-gate steepness, the survival threshold and competition temperature form a three-dimensional control surface whose regimes the platform lets users navigate in real time.

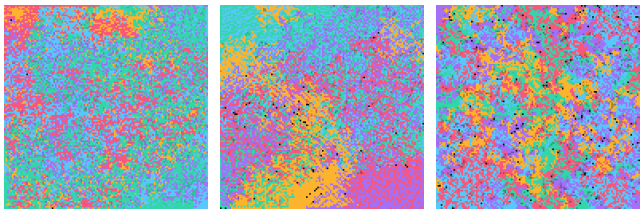
Emergent Cooperation via Threshold Cycling

Cooperative spatial sharing emerges from competitive objectives augmented by population-balancing loss terms, via a three-stage trajectory through parameter space (Figure 4).

Stage 1: Permissive mixing (steps 0–500, $\text{thr} = 0.2$). All five species coexist at close quarters without clean boundaries, with rapid early power shifts (Figures 4a–4b).

Stage 2: Territorial crystallisation (steps 500–1250, threshold ramped to 0.51). Boundaries sharpen through 0.37 (Figure 4c) and by $\text{thr} = 0.51$ (Figure 4d) the system settles into a conventional competitive equilibrium with clean territorial borders.

Stage 3: Cooperative relaxation (threshold reduced to



(a) thr = 0.1 (b) thr = 0.5 (c) thr = 0.8

Figure 3: Three survival-threshold regimes at extreme competition temperature (τ=0.1). (a) Pixel-scale flicker-mixing with no spatial coherence. (b) Diffuse blobs emerge but boundaries remain mixed. (c) Fine-grained territories stabilise once the gate filters out per-step winner flips.

0.39, Figures 4e–4f). Rather than reverting to Stage-1 mixing, the previously clean boundaries break into *interleaved sharing patterns*: specific species pairs develop checkerboard and dithered co-occupation of shared zones. Cooperation emerges *selectively* between species that were territorial neighbours during Stage 2; non-neighbours remain separated, suggesting the crystallised "social structure" seeds the relaxed state.

Cold-start control. A fresh simulation at $\tau = 1.0$, $\mathrm{thr} = 0.39$ for 2,000 steps (bottom row, panels g–l) also reaches cooperation by step $\sim$1,500, so the dynamics appear to be an attractor of the regime itself. However, the cycled run concentrates cooperation into one dominant dyad while the cold-start distributes it across multiple pairs. This observation lacks statistical support and we phrase it as "cycling appears to bias which species cooperate" rather than a general path-dependence claim.

Optimiser Choice and Learning Rate

We compare three optimisers (plain SGD, SGD with momentum, Adam) each at a representative low and high learning rate, all branched from the same checkpoint at (τ=1, thr=0.39) (Figure 5). The values are the extremes of each optimiser's viable range, chosen to make differences visible.

Plain SGD produces similar dynamics at both learning rates: slow monotonic growth into clean territorial blobs, with all five species reaching a stable equilibrium of $\sim$2,500–5,000 cells. The seed determines which species wins more than the learning rate does.

SGD+Momentum produces coherent rotations of dominance, driven by the momentum buffer carrying updates long enough to push one species into runaway expansion before it crashes. At high LR one species is permanently locked out, the only branch in our sweep that loses a species outright.

Adam produces wave-like repainting at low LR (species 4 swings $\sim$9$\times$ in amplitude over $\sim$200 steps) and *dominance bursts* at high LR (a single species briefly captures $\sim$75% of the grid before collapsing). The respawn mechanism prevents terminal extinction.

Why the three regimes differ. SGD updates are small and self-correcting, so no species overshoots. Momentum accumulates gradient direction: when species B overtakes A, B's momentum still points at A's old strategy, causing overshoot and enabling C—a slow precession through strategy space. Adam's adaptive LR divides by recent gradient variance; during stable periods variance drops, so the effective LR creeps up until a perturbation triggers a disproportionately large update, producing the burst-quiescence cycle. Switching optimiser is one click and a checkpoint reload; the difference is visible within $\sim$50 steps.

Biogeographic Construction and In Situ Perturbation

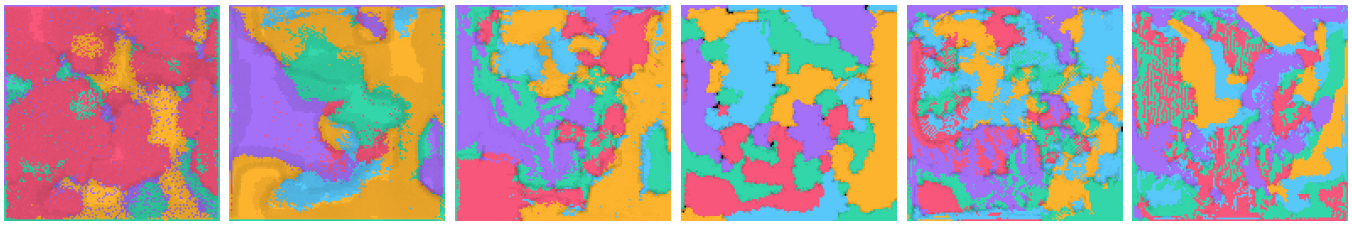
This case study perturbs the *environment* rather than parameters (Figure 6). We painted an intricate wall field on a 200$\times$200 grid and seeded five species by hand, then ran SGD with τ=0.99, thr=0.43, lr =1.3$\times 10^{-3}$.

Initial colonisation (a–c). Species grow from their seed positions and fill the available space within $\sim$350 steps, with boundaries settling along wall geometry.

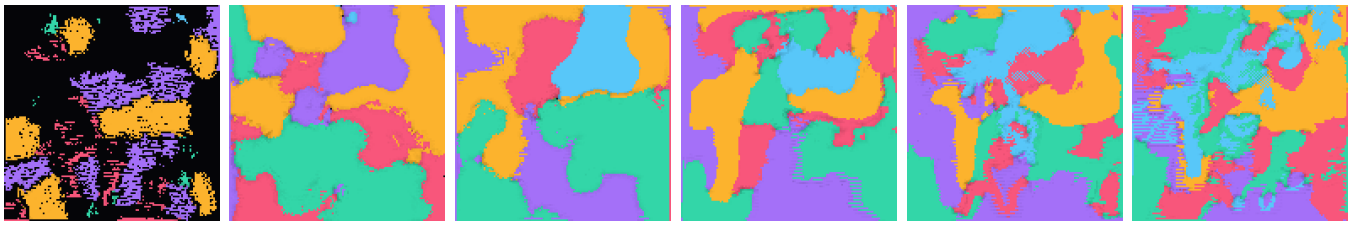
First cut and invasion (d–f). At step $\sim$550 we erased a section of wall near the centre-bottom (panel d). The nearest species raced into the opened space and temporarily dominated it (panels e–f), expanding aggressively through the gap.

Fight-back and wall-hugging equilibrium (g–l). Over the next $\sim$400 steps the population-balancing loss slowed the invaders and encouraged the displaced species to push back (panels g–h), eventually reaching a new equilibrium (panel i). In this settled state, species exhibit a notable learned behaviour: they *hug* walls, growing along them and using them as defensible safe zones that anchor their territory. Species borders follow wall geometry rather than cutting across open space, and enclaves persist for over a thousand steps inside protective alcoves (panels k–l). This wall-hugging is consistent across runs that include walls. Part of the effect is mechanical: presence gating means species can only grow into adjacent cells, so walls naturally anchor territory. But the persistence and specificity of wall-aligned boundaries suggest the NCA networks additionally learn to exploit environmental structure as part of their competitive strategy.

Second cut and regrowth (m–r). At step $\sim$2,100 we slashed two large eraser strokes across the grid (panel m), clearing species and walls. One species dominates the gap initially (n–p), but the balancing loss slows it and competitors grow back within $\sim$70 steps (q–r), demonstrating that the ecosystem's competitive dynamics are robust to large environmental perturbations.



(a) Cyc. 102, thr 0.2 (b) Cyc. 499, thr 0.2 (c) Cyc. 829, thr 0.37 (d) Cyc. 1256, thr 0.51 (e) Cyc. 1526, thr 0.39 (f) Cyc. 2676, thr 0.39



(g) Cold 26 (h) Cold 251 (i) Cold 501 (j) Cold 999 (k) Cold 1499 (l) Cold 1999

Figure 4: Emergent cooperation at τ=1.0. **Top (cycled, a–f):** threshold ramp from 0.2 to 0.51, then relaxation to 0.39; cooperative interleaving emerges at previously crystallised boundaries. **Bottom (cold-start, g–l):** fresh start at thr=0.39; cooperation emerges by step $\sim$1,500 but is distributed across multiple pairs rather than concentrated in one dyad.

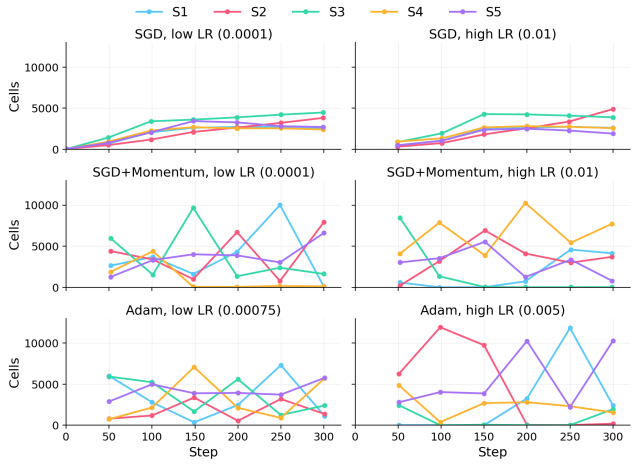


Figure 5: Per-species cell counts for six optimiser×LR branches from one checkpoint (τ=1, thr=0.39). SGD (top): stable equilibria. SGD+Momentum (middle): rotating dominance at low LR, species lockout at high LR. Adam (bottom): 9×-amplitude waves at low LR, dominance bursts ($\sim$75% grid capture) at high LR.

Discussion

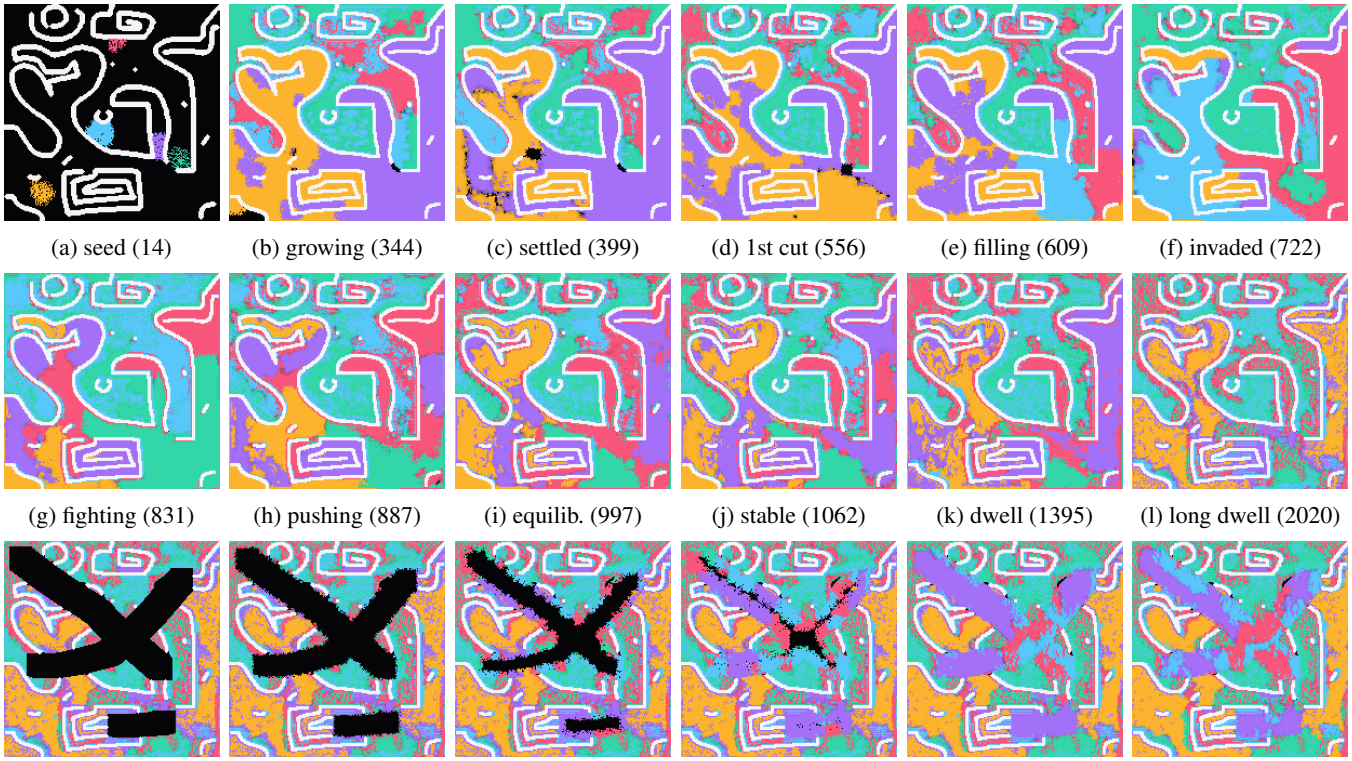
Interactive exploration vs. batch sweeps. Several phenomena reported here (threshold cycling, the growth-gate slider walk, the biogeography eraser cut) emerged from observation during a live session, not from a predefined experimental design. Batch sweeps map equilibria across parameter space but miss path-dependent transients that require sequential changes from a specific dynamical state. Interactive exploration is most valuable in the hypothesis-generation phase; automated sweeps can then quantify robustness across seeds. The two approaches are complementary, and the checkpoint export system is designed to bridge them: a researcher can explore interactively, identify a phenomenon, export the checkpoint, and hand it to a scripted sweep for replication.

The growth gate as a design principle. The growth-gate steepness controls whether the system sits in a frozen, critical, or turbulent regime via a single scalar that tunes the nonlinearity of the alive/dead boundary. Any differentiable ALife system with a threshold-based viability criterion likely has an analogous control point; exposing it as an interactive knob lets users navigate between regimes in real time.

Respawn and coexistence. The emergency respawn reseeds a species when its total aliveness drops below 1, so the dynamics here are quasi-stationary coexistence under an immigration floor, not emergent stability. Disabling respawn would reveal whether any observed equilibria are self-sustaining.

Phenomena beyond these case studies. The parameter space contains many more phenomena than five studies can sample. Holding the survival threshold at 0.6 produces a rigid equilibrium; raising τ>3 at that threshold collapses species into isolated migrating blobs. These qualitative observations illustrate why interactive exploration is valuable: the parameter space is too large and too nonlinear to cover with scripted sweeps alone.

Scope and limitations. The case studies vary only six of the platform's 40+ parameters; many axes (species count, architecture, loss shaping) remain at defaults. The observations reported here are qualitative and exploratory; rig-



(a) seed (14) (b) growing (344) (c) settled (399) (d) 1st cut (556) (e) filling (609) (f) invaded (722)

(g) fighting (831) (h) pushing (887) (i) equilib. (997) (j) stable (1062) (k) dwell (1395) (l) long dwell (2020)

(m) 2nd cut (2102) (n) +2 steps (o) +7 steps (p) +13 steps (q) +38 steps (r) +70 steps

Figure 6: Biogeographic construction over $\sim$2,200 steps. **Top (a–f):** colonisation, then wall erasure (d); nearest species race into the gap. **Middle (g–l):** fight-back, wall-hugging equilibrium persists $>$1,000 steps. **Bottom (m–r):** two eraser slashes open corridors; one species dominates briefly before the balancing loss restores equilibrium.

orous quantitative analysis with statistical significance testing across multiple seeds is needed before any causal claims can be drawn. WebGL non-determinism prevents bitwise replay ($\sim$26% cell disagreement after 94 steps from the same checkpoint, concentrated at contested boundaries). The sun, as a globally uniform learnable competitor, introduces spatially homogeneous coupling between all cells; the platform includes an optional *local sun* mode in which each grid cell carries an independent sun parameter vector, decoupling the environment spatially, but this mode is not exercised here. PD-NCA's pairwise scoring is $O(N^2)$ per pixel; a mean-field approximation is implemented but not exercised here, and scaling past five species is an open direction.

Conclusion

We have presented an interactive, browser-based platform for real-time exploration of multi-agent neural cellular automata trained by online gradient descent. The checkpoint-branch-explore workflow enables a form of experimental investigation that offline batch sweeps cannot replicate: researchers observe transient phenomena as they occur, form hypotheses, and test them by perturbing a live system from a saved state. The growth-gate case study demonstrates this most directly, showing that a single differentiable parameter can push the alive-channel field from bistable territories into an excitable edge-of-chaos regime that the platform lets researchers observe, checkpoint, and probe as it happens. The remaining case studies show that competition temperature, survival threshold, optimiser choice, and spatial geometry each produce distinct dynamical regimes accessible within the same platform. Two natural next directions are an automated controller that co-adjusts gate steepness, temperature, and threshold to keep the system at the edge of chaos as training proceeds, and a community gallery where users can share and branch from each other's checkpoints, bringing Picbreeder-style collaborative exploration to online-trained ALife ecosystems. The platform is available at `https://pub.sakana.ai/digital-ecosystem`.

Acknowledgements

We acknowledge the Frontier Intelligence Group at Sakana AI for their creative thoughts, ideas, and testing of the Digital Ecosystems demo. Without their excitement and passion, this work would not have made it to submission.

References

Agüera y Arcas, B., Alakuijala, J., Evans, J., Laurie, B., Mordvintsev, A., Niklasson, E., Randazzo, E., and Versari, L. (2024). Computational life: How well-formed, self-replicating programs emerge from simple interaction. *arXiv preprint arXiv:2406.19108*.

Barbieux, A. and Canaan, R. (2024). Coralai: Intrinsic evolution of embodied neural cellular automata ecosystems. *arXiv preprint arXiv:2406.09654*.

Chan, B. W.-C. (2019). Lenia: Biology of artificial life. *Complex Systems*, 28(3):251–286.

Grasso, C. and Bongard, J. (2022). Empowered neural cellular automata. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 108–111.

Heinemann, C. (2024). ALiEn: Artificial life environment. `https://github.com/chrxh/alien`. Open-source CUDA-based ALife simulator.

Kumar, A., Faldor, M., Zhang, J., Cully, A., and Clune, J. (2025). Automating the search for artificial life with foundation models. *Artificial Life*, 31(3):368–395.

Langton, C. G. (1990). Computation at the edge of chaos: Phase transitions and emergent computation. *Physica D: Nonlinear Phenomena*, 42(1–3):12–37.

Mordvintsev, A. (2023). SwissGL: Minimalist WebGL2 wrapper for GLSL shaders. `https://github.com/nicoptere/SwissGL`. Open-source library.

Mordvintsev, A., Randazzo, E., Niklasson, E., and Levin, M. (2020). Growing neural cellular automata. *Distill*.

Niklasson, E., Mordvintsev, A., Randazzo, E., and Levin, M. (2021). Self-organising textures. *Distill*.

Packard, N., Bedau, M. A., Channon, A., Ikegami, T., Rasmussen, S., Stanley, K. O., and Taylor, T. (2019). An overview of open-ended evolution: Editorial introduction to the open-ended evolution II special issue. *Artificial Life*, 25(2):93–103.

Packard, N. H. (1988). Adaptation toward the edge of chaos. In Kelso, J., Mandell, A., and Shlesinger, M., editors, *Dynamic Patterns in Complex Systems*, pages 293–301. World Scientific.

Pajouheshgar, E., Xu, Y., Abbasi, A., Mordvintsev, A., Jakob, W., and Süsstrunk, S. (2025). Neural cellular automata: From cells to pixels. *arXiv preprint arXiv:2506.22899*.

Plantec, E., Hamon, G., Etcheverry, M., Oudeyer, P.-Y., Moulin-Frier, C., and Chan, B. W.-C. (2023). Flow-Lenia: Towards open-ended evolution in cellular automata through mass conservation and parameter localization. In *Proceedings of the Artificial Life Conference*. MIT Press.

Randazzo, E., Mordvintsev, A., Niklasson, E., and Levin, M. (2023). Biomaker CA: a biome maker project using cellular automata. *arXiv preprint arXiv:2307.09320*.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520.

Secretan, J., Beato, N., D'Ambrosio, D. B., Rodriguez, A., Campbell, A., Folsom-Kovarik, J. T., and Stanley, K. O. (2011). Picbreeder: A case study in collaborative evolutionary exploration of design space. In *Evolutionary Computation*, volume 19, pages 373–403.

Smilkov, D., Thorat, N., Assogba, Y., Nicholson, C., Kreeger, N., Yu, P., Cai, S., Nielsen, E., Soegel, D., Bileschi, S., et al. (2019). TensorFlow.js: Machine learning for the web and beyond. In *Proceedings of Machine Learning and Systems*.

Sudhakaran, S., Najarro, E., and Risi, S. (2022). Goal-Guided neural cellular automata: Learning to control self-organising systems. In *ICLR 2022 Workshop "From Cells to Societies"*.

Takagi, H. (2001). Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation. *Proceedings of the IEEE*, 89(9):1275–1296.

Wilensky, U. (1999). NetLogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University, `http://ccl.northwestern.edu/netlogo/`.

Wolfram, S. (1984). Universality and complexity in cellular automata. *Physica D: Nonlinear Phenomena*, 10(1–2):1–35.

Zhang, I., Risi, S., and Darlow, L. (2025). Petri dish neural cellular automata. `https://pub.sakana.ai/pdnca`. Sakana AI technical report.