

Fractal Automata

Using Fractal Neural Networks to Play SimCity 1 and Conway's Game of Life at Variable Scales

Sam Earle

November 13, 2019

Outline

- 1 Introduction
- 2 Environments
 - Micropolis
 - Game of Life
 - Power Puzzle
- 3 Methods
- 4 Results
- 5 Fractal Expansion
- 6 Idea for a Scalable, Pathfinding Cellular Automaton
- 7 Conclusion

Goal: to have an A.I. play SimCity.

Approach:

- observation: 2D image of map, channels of tilestates
 - action: 2D image of map, channels of (tile-specific) actions
 - agent observes map, takes one tile-specific action, city sim ticks, repeat
 - agent is rewarded for population (or some combination of city-wide metrics) once per step

Desiderata:

- an agent should be able to play the game on various map-sizes
 - an observation in one corner of the map can affect an action taken in the opposite corner Non-athropomorphic agents invite new metaphors.

What does 'expert play' look like in the context of an open-ended simulation?

- to transfer gracefully between interesting states
 - to achieve simulation-states with varying global characteristics by effective combinations of local actions, e.g.:
 - *Roller Coaster Tycoon's* scenarios (num. guests, park/roller-coaster rating, cash flow, etc.)
 - player-generated goals in *The Sims* (e.g., new couple in town conspires to drown entire neighbourhood in ladderless pool; player engaging in high-level games not represented directly by the software itself)

In addition to an overall score, we have other global metrics like cash flow, population by zone-type, traffic, crime rate, citizen happiness, land value, etc.

Goal: to have an A.I. play SimCity, or, to extend SimCity with A.I.

- to extend the human player's abilities and knowledge
 - abilities: e.g., 'build a low-density residential neighbourhood in the 2nd quadrant of the map'
 - knowledge: i.e., indirectly, as the player witnesses the layout-strategies used to achieve various goals, gaining new insight into simulation logic

Auxiliary goal: develop RL-agent extension for Geographical Information Software (GIS)

- Planners specify global (or local) constraints, agent drafts multiple possible city layouts.

Micropolis

The player builds power plants, zones, services, and infrastructure on a 2D grid, to attract residential, commercial and industrial development. Traffic results. The player has to manage the emergent global dynamics of localized forces to keep the city alive.

Introduction
oooo

Environments
○●oooooooooooo

Methods
ooo

Results
oooo

Fractal Expansion
oo

CA
oo

Conclusion
oooo

Simulation Quirks

Circuitous commutes.

- In SC4, visible traffic is produced by a collection of CA's, and "only loosely" influenced by pathfinding algorithms. From the guide to the Network Addon Mod (NAM) mod for SimCity4, which extends the traffic simulator:

To a very large extent, then, the traffic simulator is actually a desirability generator.

...the destination finder has been tuned aggressively by Maxis in ways that we cannot access in order to minimize how much the pathfinder must run, since finding paths is one of the most expensive operations (in terms of CPU time and memory) in SC4.

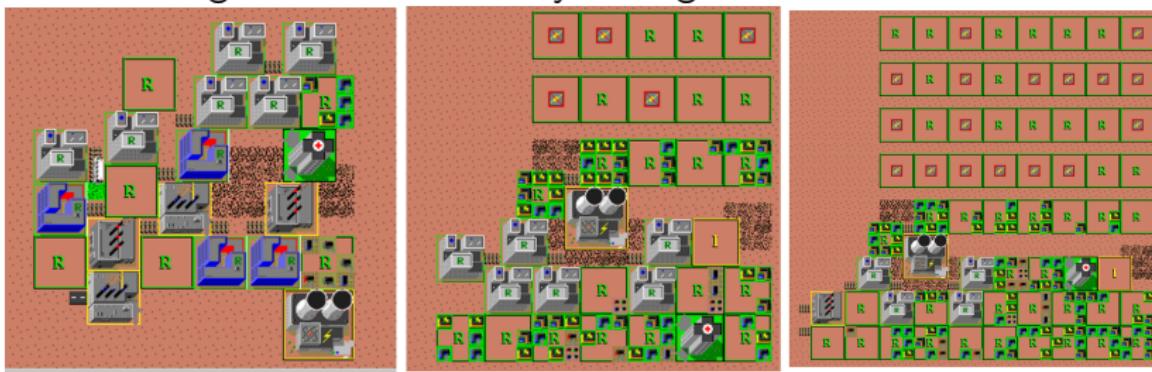
- Idea: implement traffic simulator as stack of pathfinding CA and send to GPU for parallel processing.
 - Train randomly initialized stack of CA as NN, in pytorch, to imitate NAM-like, CPU-based traffic sim.

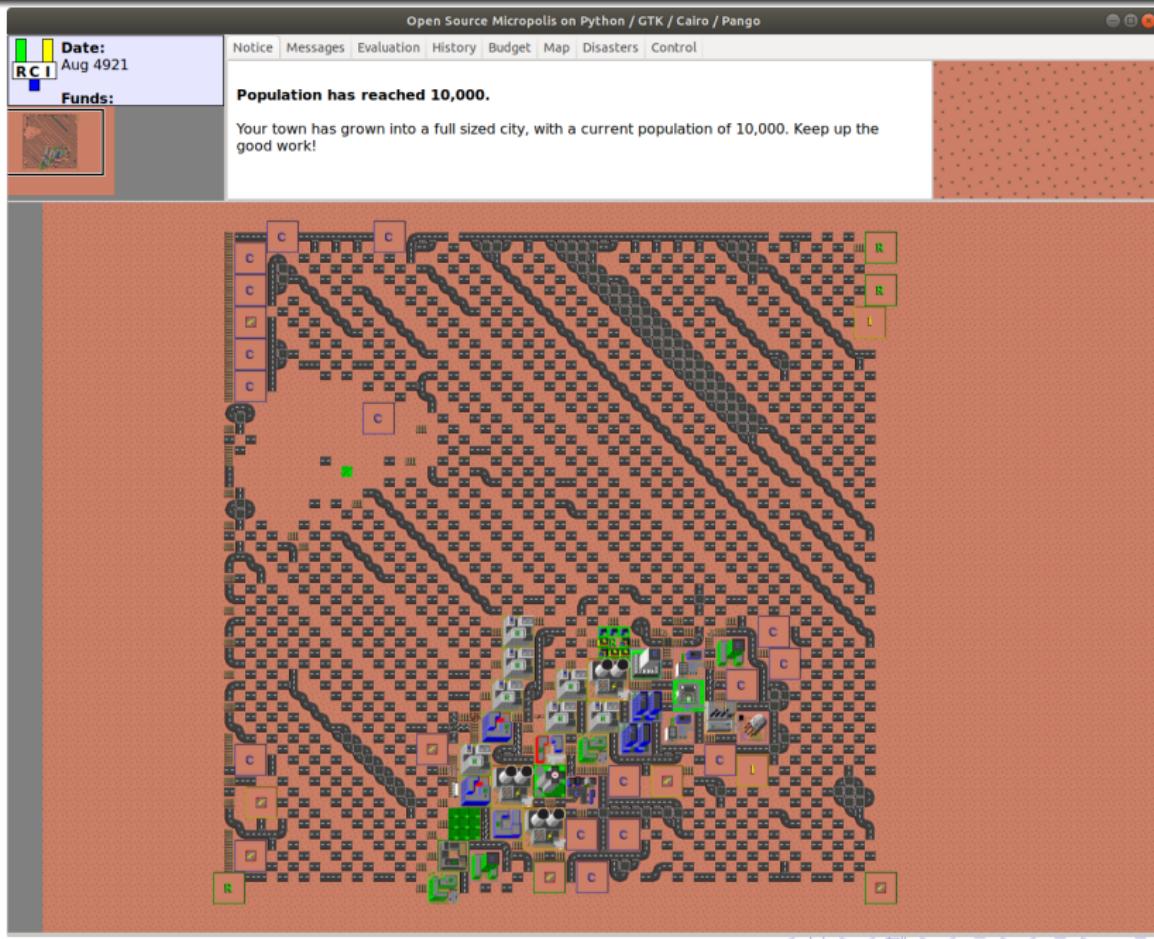
Agent learns to avoid traffic, which is repellent and unnecessary, by limiting connectivity of networks.



Above: agent-inspired utopian hack Right:
(NIMBY)-human and agent collaboration

Local strategies scale more easily than global ones.





First observation: spatial logic is preserved at multiple scales.

- traffic flows between neighbourhoods, cities, and regions; so do power, trash, and goods
 - model with repeated convolutions
- specialized subregions exert forces of desirability and demand at multiple scales
 - model with convolutions applied to variable-size maps.

Consider possible subtask of predicting power flow over grid.

- Use a CA: "if I am not an obstacle, and I am adjacent to a powered cell, then I am a powered cell"
 - Can easily implement this with a single repeated convolution and a (hard) tanh activation.

Batty and Longley, 1994, show that fractals can be used to generate plausible urban plans, and that real cities have fractal dimension.

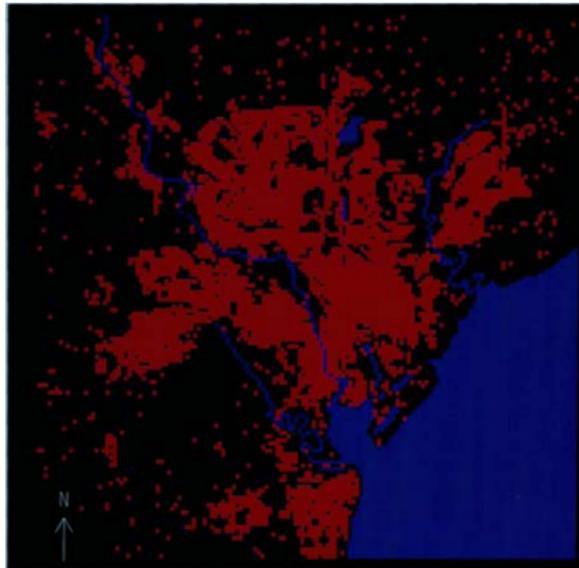
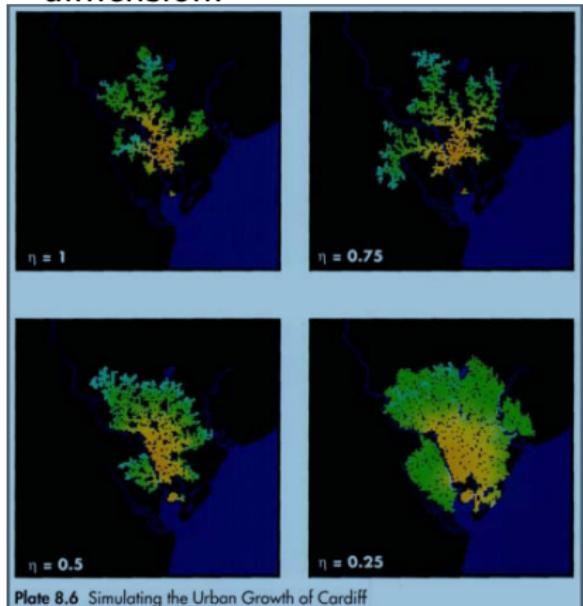
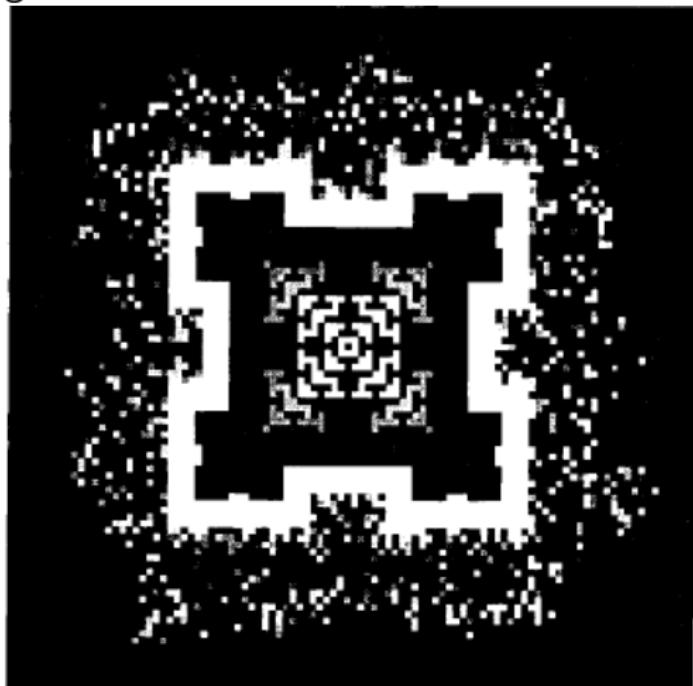


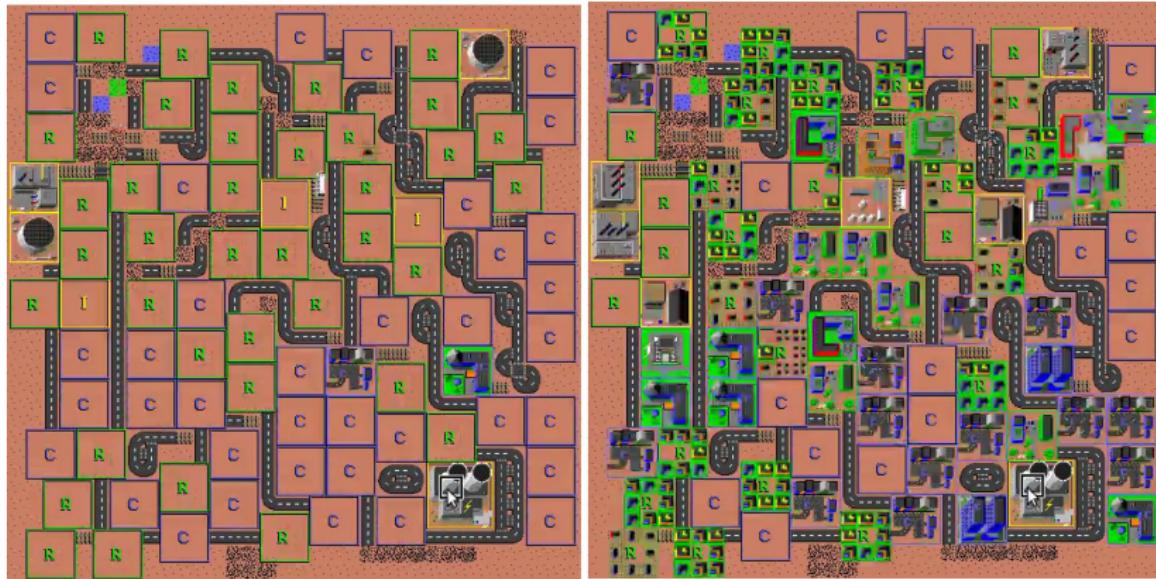
Plate 8.6 Simulating the Urban Growth of Cardiff



Batty, 1997, shows that Cellular Automata can be used to create idealized and stochastic fractal urban geometries.



Local zoning strategies lead to emergent neighbourhoods at various scales.



Conway's Game of Life

A 2D binary grid is (uniformly) randomly initialized. Based on (spatially-invariant) rules about neighbouring cells, the state of each cell at $t + 1$ is computed in parallel. A rich variety of global dynamics may result. For RL:

- After each tick of the simulation, a player/agent can activate a single cell.
- Agent is rewarded for net active-cell count (i.e., population)

Introduction
oooo

Environments
oooooooooooo●oooo

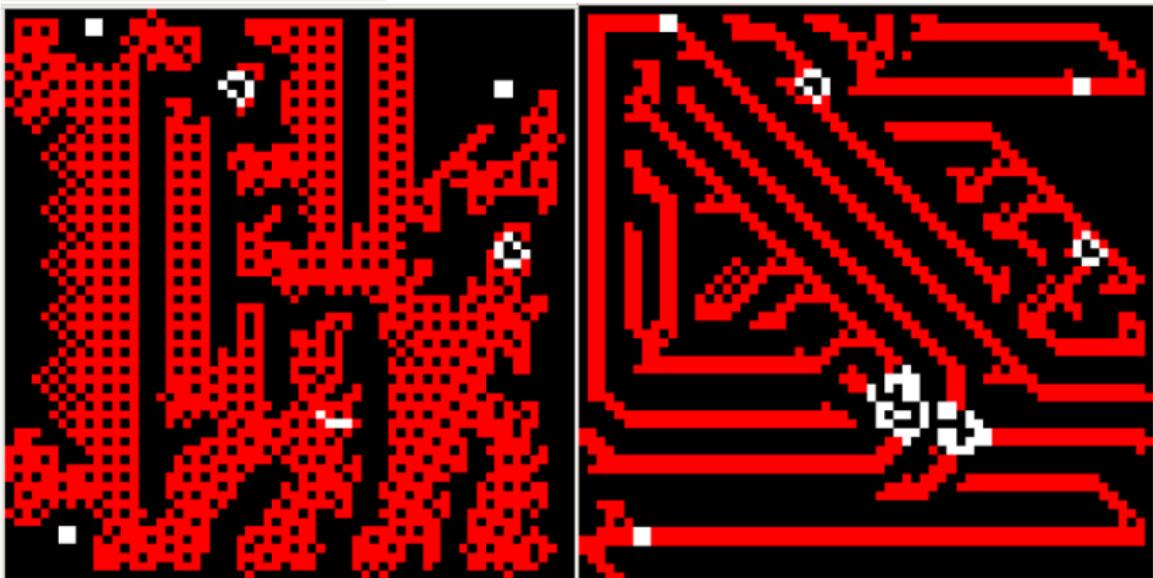
Methods
ooo

Results
oooo

Fractal Expansion
oo

CA
oo

Conclusion
oooo



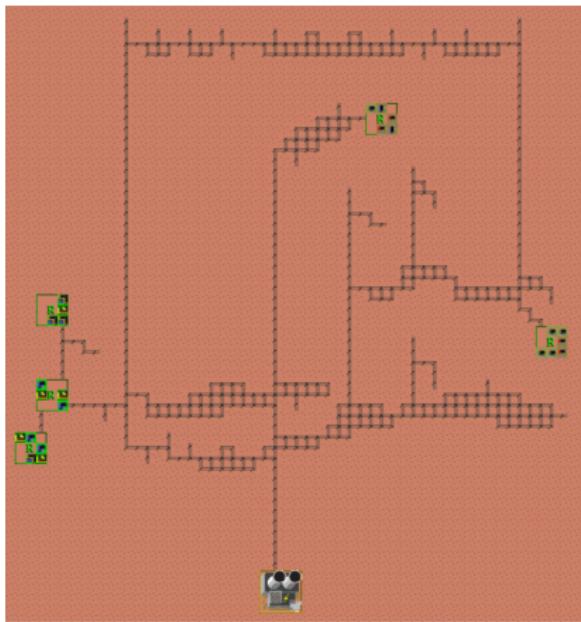
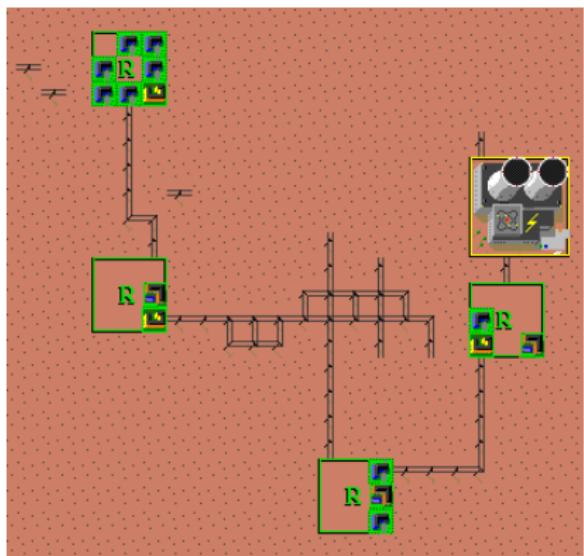
Game of Life: Qualitative Observations

- All agents tend to build branching, stable structure, usually only once the simulation has reached a near-static state.
 - Hard to tell if the agents actively quell chaos, or simply wait for dust to settle.
 - In rare cases, though, agents that learn to build stable structures on small maps, have a different experience on larger ones: a cloud of oscillating chaos gradually fills the board, then stabilizes.
- Agents sometimes get caught in oscillators, or participate in glider-like structures.

Power Puzzle

A random micropolis map is initialized, with one power plant and multiple residential zones. The agent must deliver power to each, as quickly as possible (to achieve maximum population reward per episode).

- Per-step reward demands *time-sensitive* spanning tree, which adds nodes to tree using shortest available path, s.t. (num. powered zones) / (num. timesteps) is maximized.



Introduction
oooo

Environments
oooooooooooooo●

Methods
ooo

Results
oooo

Fractal Expansion
oo

CA
oo

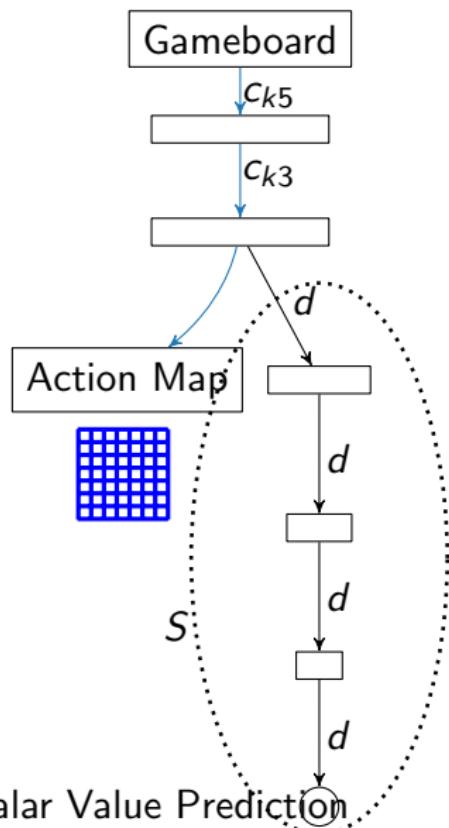
Conclusion
oooo

Learning Algorithm

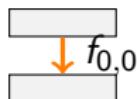
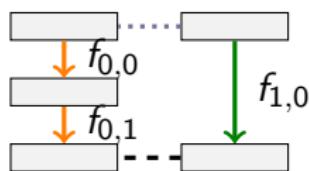
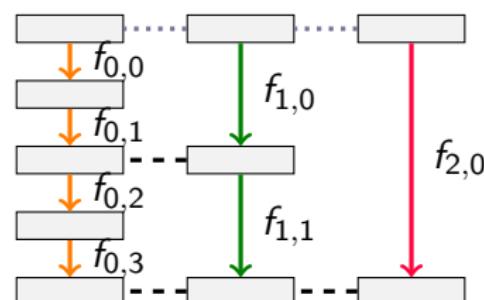
In Advantage Actor Critic (A2C), the agent must produce an action and a scalar value-prediction at each step. The value prediction is trained to predict the ‘return’ of a given map-state, which is a function of the reward it generates, while the action prediction is trained to optimize the predicted value of the resulting map-state.

- To allow 2D input and output of variable dimensions, stick to convolutional layers.
- To generate the value-prediction, recursively apply a single strided convolution shrinking the hidden 2D activation as necessary.

Network Architecture



In order to calculate a scalar value prediction regardless of map size, replace the fully-connected layer in Vinyals, 2017 with a (variably) repeated, strided convolution.

F^1  F^2  F^3 

→ Convolution f

(32 × 32) Activation Map

... Join Operation (el.-wise averaging)

..... Equality

intra-column weight-sharing:

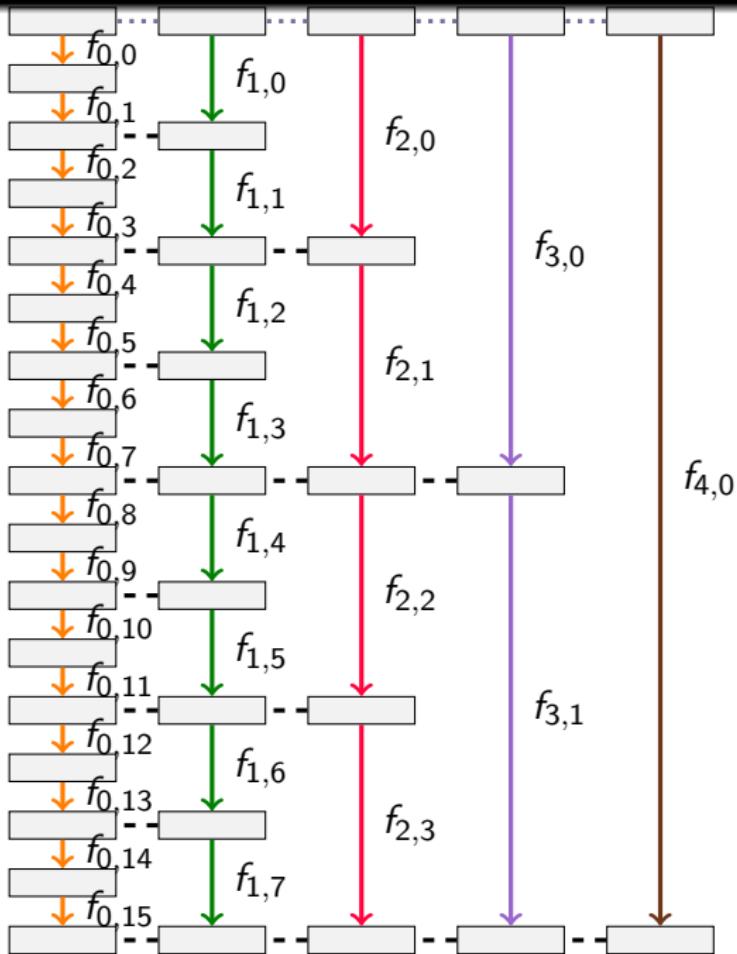
$$\forall i, j, j'.$$

$$f_{i,j} = f_{i,j'}$$

inter-column weight-sharing:

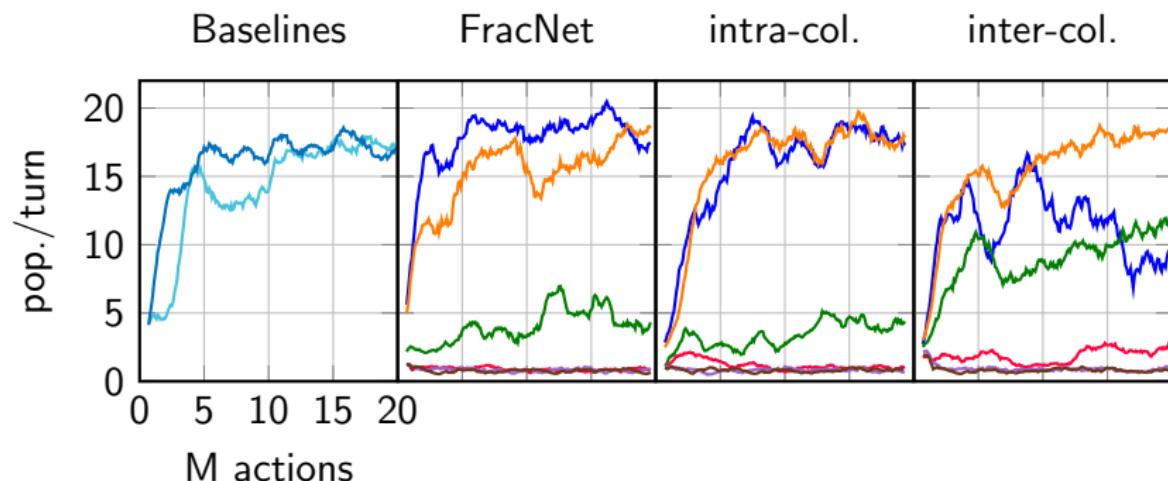
$$\forall i, i', j, j'.$$

$$f_{i,j} \equiv f_{i',j'}$$



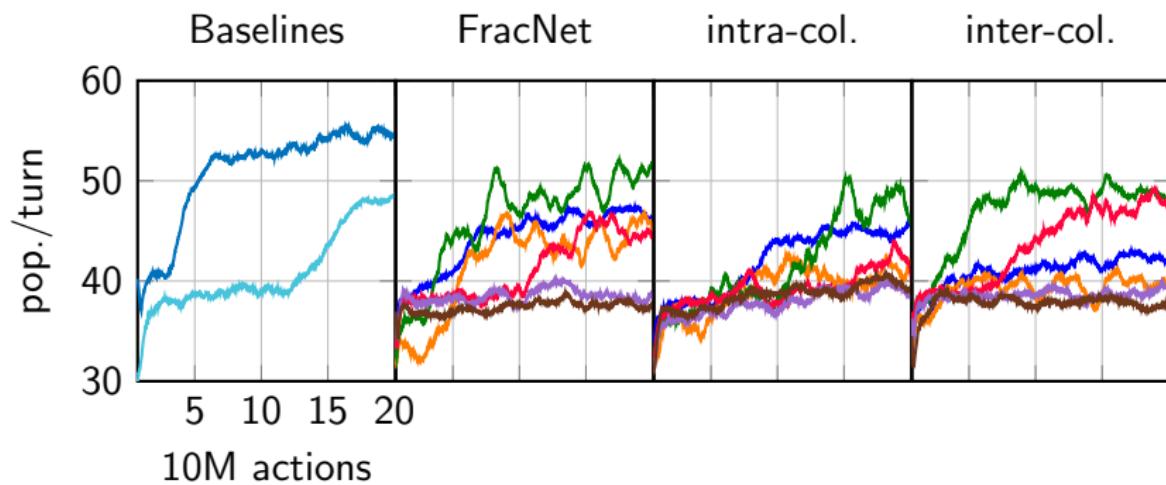
With drop path,
each column is
encouraged to learn
the task
independently.
With full
weight-sharing, a
single set of weights

Results: Power-Puzzle



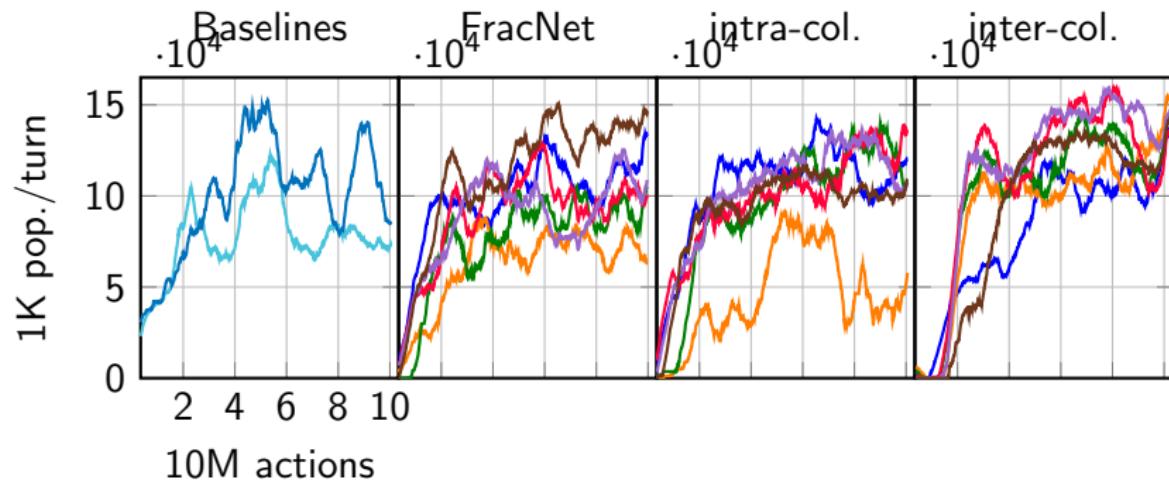
- recursive value function accelerates learning
- larger receptive fields help
- a set of weights that is forced to encode a deeper network is better at encoding a shallower one

Results: Game Of Life



- 8-layer-deep column dominates
- learned strategy seems largely local, and does not scale well to deeper nets

Results: Micropolis



- Viable strategies can be encoded by both shallow and deep recursive nets; and in the same set of weights.

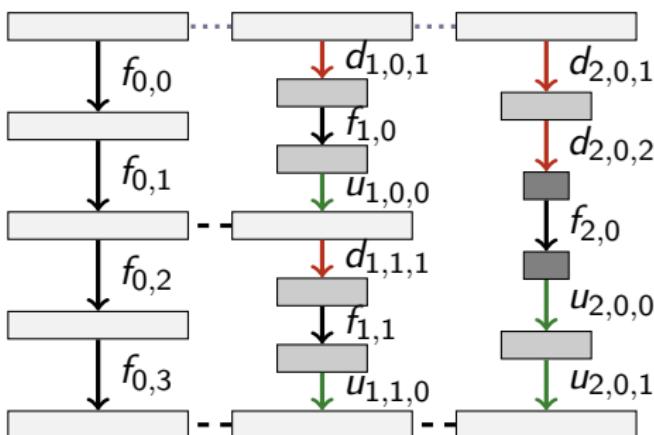
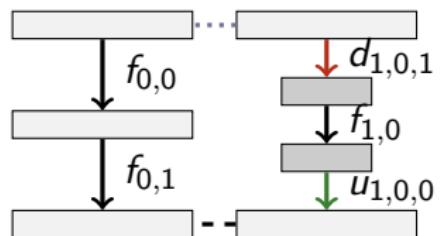
Caveats of the Results

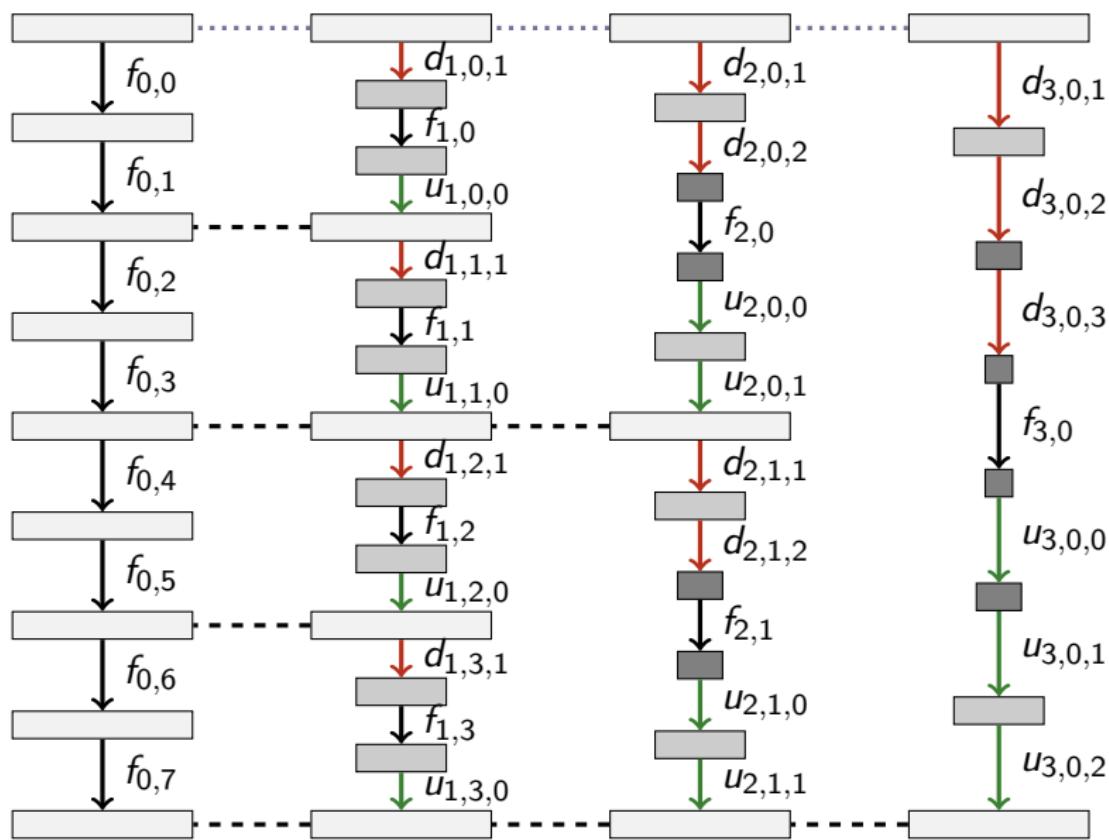
In the above experiments:

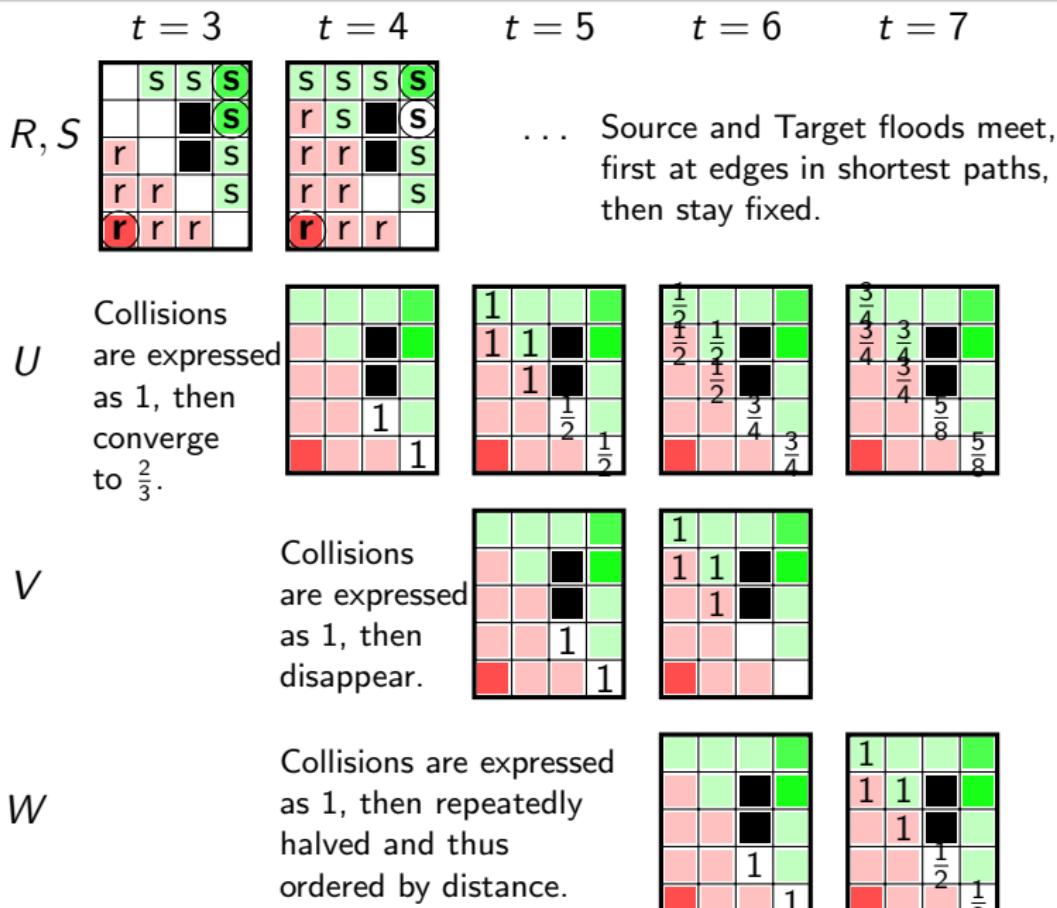
- Depth \iff Receptive-Field Size, but which matter(s)?
 - Does the agent use increased depth to incorporate distant info., or to refine local computation?
 - Power-Puzzle incentivizes global info.
 - Can we design a problem that is locally difficult—that demands depth (i.e., multiple non-linearities) but not receptive-field-size?
- Expanding a trained FNN does not help it do 0-shot learning on larger maps.
 - We probably need to expand and contract the FNN (and/or the map-size) during training.

Dimensionality-Reducing Fractal Expansion

Dimensionality-reducing columns are cheaper in terms of compute, while maintaining the large receptive field required to solve networking and related problems.





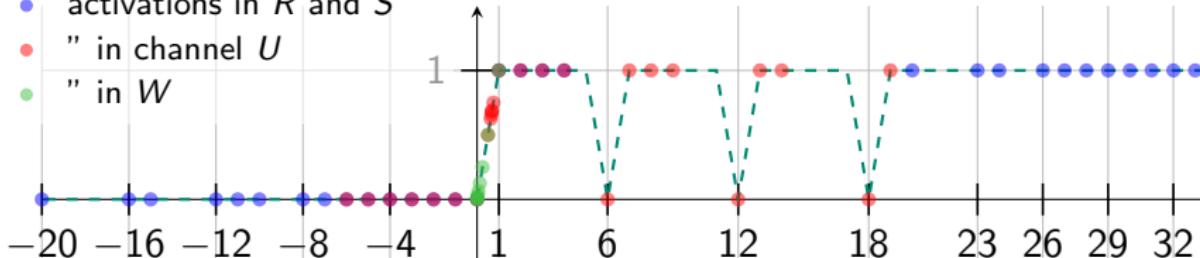


- - - possible nonlinearity $g(a)$

- activations in R and S

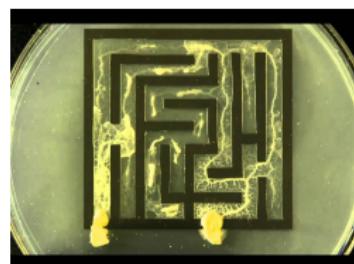
- " in channel U

- " in W



Like the "physarum" slime mould (which has been modeled as a CA), the CA pathfinding algorithm finds shortest paths by sending out floods of activation from sources and targets. Once the floods meet:

- In "physarum," the spread continues for some time, before longer paths are pruned.
- In our CA, the timestep of first contact between the floods is recorded at each tile.



In the POET (Paired Open-Ended Trailblazer) algorithm, the environment is mutated incrementally in order to challenge the agent to develop a diverse set of gameplay strategies.

- Our next step: use POET to mutate a target vector (or reward-weight vector) to encourage the agent to become adept in developing cities with different overall character.
 - The player will be able to control this target vector in real-time, modifying the agent's behaviour at a high-level.

Recent work has leveraged intrinsic curiosity to train RL agents to generate “life” in Lenia, a continuous-valued adaptation of Conway’s Game of Life.

- A CPPN generates starting configurations.
- A VAE attempts to predict the resulting activity of the simulation.
 - Its hidden activation-space is treated as the CPPN’s goal-space.

Algorithm:

- The CPPN is evolved to elicit the target vector in the VAE
- The target is sampled uniformly from a sufficiently large hypercube in goal-space

Note that the agent provides only the initial state (rather than acting on each tick), and affects all tiles at once (rather than one at a time).

Proposal: Agent and Environment emerge from same substrate, and co-evolve, driven by intrinsic curiosity.

Summary

- New RL environments: SimCity, (step-wise interactive) Game of Life, and a minimal but challenging pathfinding problem.
 - Inherently open-ended (SC and GoL). Interactive.
 - Variable map-sizes.
- Weight-sharing in FNNs allows for 0-shot transfer to larger map-sizes with the possibility for global policies.
- A new fractal expansion rule saves compute when scaling up, while encouraging dimensionality reduction.
 - Conjecture: in addition to their form and function, these FNNs possess fractal representations of the map (and produce fractal structures upon it over time).
- A NN implementation of a CA pathfinding algorithm.
 - A possible path toward a general mapping from CA to NN.
 - A possible paradigm for the co-evolution of game-player and game-mechanics.