Novel Generation in Continuous Cellular Automaton

Exploring Metrics to Capture Intuitively Novel Generation in Agent Based Systems

Harkirat Gill

Northwestern University

INTRODUCTION

The field of artificial life has produced a number of interesting and sophisticated systems over the years, but a fundamental difference between the existing evolutionary systems and natural evolution is that artificial systems tend to converge and become limited in their diversity and complexity whereas natural evolution continuously generates new and increasingly complex phenomenon.

The possibility of one day understanding the principles of open-ended evolution (OEE) well enough to implement them in different domains could be very valuable to researchers in a number of fields. Biologists would be able to rapidly perform experiments and gain better insights into evolutionary insights. Evolutionary algorithms could be applied to a wider range of problems and overcome certain complexity limitations. In artificial life research, the ability to build and experiment with such systems would open up the possibility of better understanding the underlying mechanics of natural life (E. L. Dolson et al. 2019).

In order to be able to build open-ended systems, a better understanding of what criteria an open ended system should meet and clear metrics for measuring a systems success in meeting such criteria is needed. There is growing consensus that an open-ended system must have certain dynamics including the continual generation of novel states as well as the potential for unbounded growth in complexity and diversity. However, if there exist other essential dynamics, how such dynamics are measured, whether satisfying the criteria by such measures would be sufficient to declare a system to be open-ended are still open questions, and if any criteria even exists or open-endedness exists on a continuum, (E. L. Dolson et al. 2019). Furthermore, many systems claiming to have passed tests of open-endedness fail may not exactly match intuitions about what would be expected of open-ended generation.

Adams et al. argue that, for a system to exhibit OEE, it must be composed of interacting subsystems analogous to a natural organism and its environment. They argue that a subsystem can be considered open-ended if and only if it is unbounded and innovative and they provide precise definitions of both unbounded and innovative. E. L. Dolson et al. propose five hallmarks of OEE in a hierarchical ordering as well as precise definitions that they argue are necessary, but maybe not sufficient to generate open-ended behavior: change potential, novelty potential, complexity potential, diversity potential, and transition potential. Beyond simply creating measures of novelty, the ability to be able to easily apply them across a wide variety of agent based systems is another important consideration.

Of the different hallmarks of OEE, novelty is one of the most agreed upon and a core part of what would align with the intuitive sense of a system being open-ended. Thus the focus of this paper is on finding simple and intuitive measures of novelty that can flexibly be applied to a range of complex systems. Specifically, the goals here are (1) find simple metrics for novelty that align with what is intuitively considered novel which can flexibly be applied to different systems and (2) see if a system can be maximized for novelty based on these metrics generate states that would be intuitively considered novel. Experiments in both evaluating and maximizing these metrics are conducted within a 1D continuous cellular automaton (CA) system due to their simplicity and potential for complex generation. This is described in more detail in the design section.

Novelty

So what are some existing measures of novelty? A number of systems use the amount of time an evolutionary system runs before arriving at a "dead" state and the emergence of new species as simple measures for how much novelty a particular system is capable of (E. L. Dolson et al. 2019). E. L. Dolson et al. measure novelty in a multi agent system by tracking the lineage of different agents and counting the number of new agents that have a lineage of a certain length to filter out mutations that may not last long. Adams et al. defines a state in a subsystem as innovative only if, without outside influence to the subsystem, the state would not have been possible. Adams et al. point out that no finite system that is isolated from outside influence can exhibit OEE because any finite system must eventually repeat some state within its Poincaré recurrence time, the time before all possible states can be visited. At least in the case of CA being explored here, once a state repeats, it will stay in a loop as the same rule applied on the same state will repeat the same states. They say that another way to address limitation this is to add a level of stochasticity (Adams et al. 2017). This paper is concerned with exploring robust measures of novelty and so even though, because the system these measures are explored on is finite, robust metrics should be able to be easily applied to such a system which responds to outside influences and is capable of satisfying Adams et al.'s definition of innovative.

The metrics described here, require the system on which novelty needs to be measured to fit certain criteria. E. L. Dolson et al. requires the system to either have a shadow world where evolutionary pressures are turned off or be able to track agent lineages. Adams et al. requires that counterfactual histories be considered to measure the effect of an outside systems influence on a subsystems innovation potential. Part of the aim of the first goal for this paper is to explore metrics which can flexibly be applied to different to a number of agent based systems for which it may be difficult or impractical to meet the criteria of the other works described.

DESIGN

Cellular Automaton

Cellular automaton are a mathematical models where cells in N dimensional space act autonomously in determining their next state based on their own state and their neighbors' states. Even simple transition rules are capable of generating quite sophisticated outputs (Wolfram 2002). Due to their balance of simplicity and complexity, they are commonly used to explore complex phenomenon. There are also a number of recent works on measuring open-endedness in 1D binary CA systems (Sughimura et al. 2013, Adams et al. 2017, Andras 2021). Using 1D cellular has the added advantage of being able to easily analyze the spatial and temporal aspects of novelty easily in one image. Because of this, they are used in this paper to explore different novelty metrics. However, instead of binary CA, that only take values 0 or 1, continuous CA which take values anywhere between 0 and 1 are used in this paper. This is because they are capable of far more sophistication and diversity than simple 1D binary CA and this is needed to better evaluate the effectiveness of different novelty metrics.

In order to be able to evaluate the effectiveness of various novelty metrics in this cellular automaton environment, a method of randomly generating different rules providing a proper coverage of possible patterns is needed. In 1D binary CA, this is simple. Because there are only eight possible permutations of states a cell and its neighbors can be in, a binary value needs to be defined for each, giving $2^8 = 256$ possible rules. In continuous CA, there are an infinite number of states a cell and its two neighbors can be in. A simple method of generating rules used here is to generate a 3D matrix size RxRxR with random values between 0 and 1. In order to generate the next value of a cell, the current cell value and its two neighbor values can be used as coordinates and the value at those coordinates is the next state of the cell.

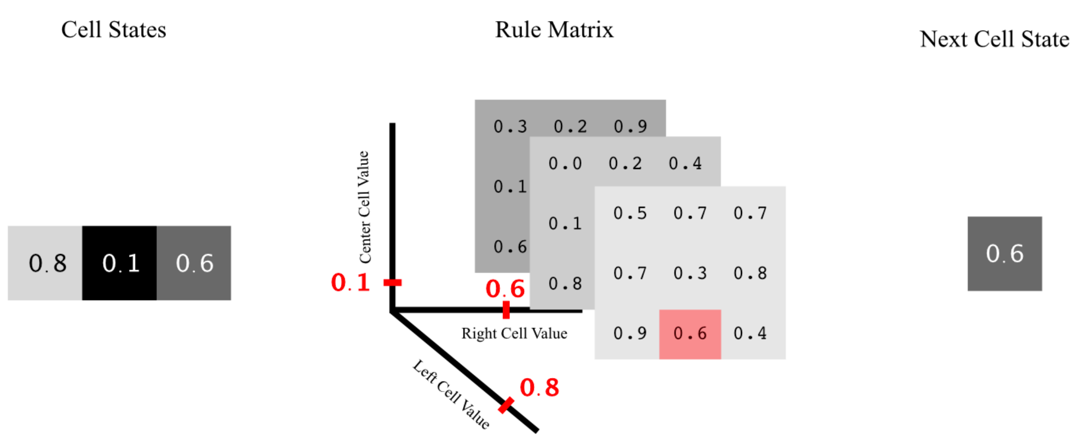

Figure 1: Generating the next cell state given its current state and neighbors current state using the rule matrix.

With this simple rule matrix method, each cell can determine its next value quite easily. The size of the matrix, R (resolution), allows for control of complexity of the output. The larger the R, the more intricate the output of the CA will generally be. While this CA is still discrete in a sense, sampling a new rule will allow it to take new values anywhere between 0 and 1 each time with no set restriction on how the values are discretized.

Sample Rule Generations

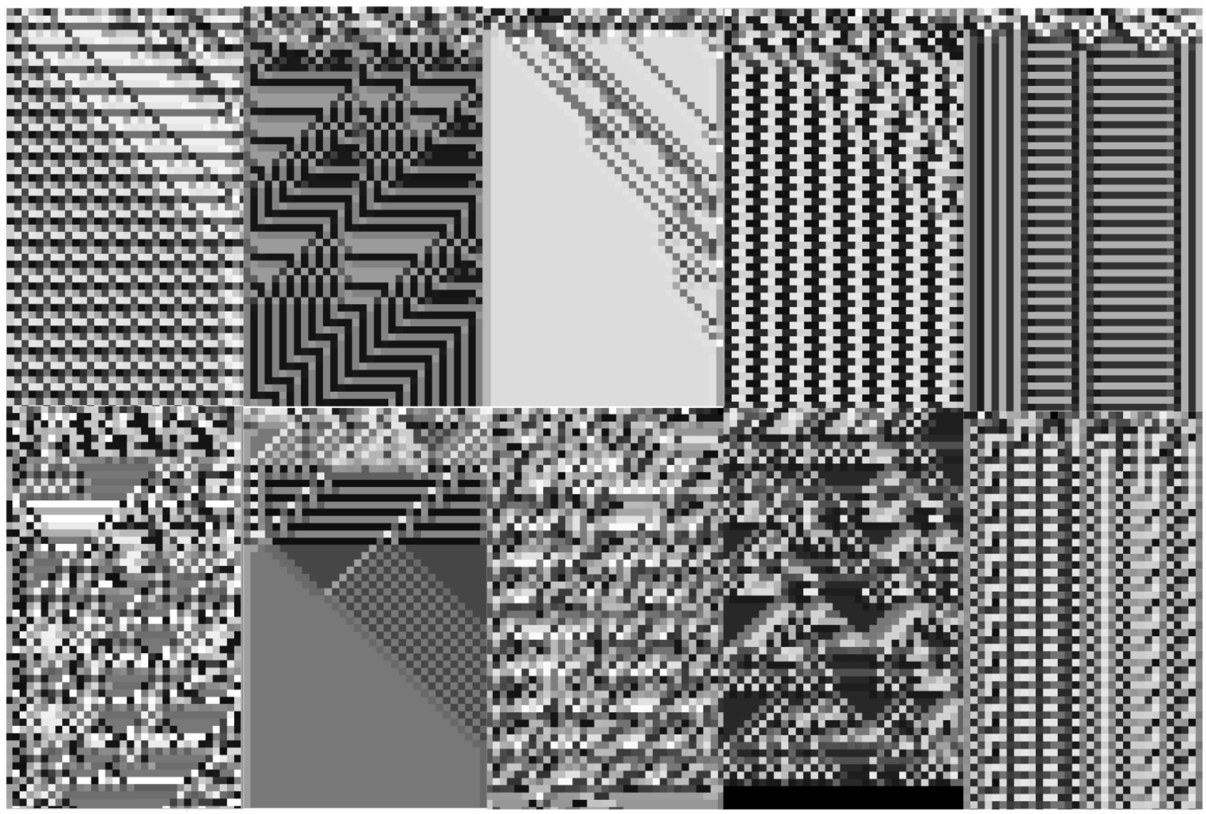

Figure 2: Sample rule generations using randomly generated rule matrices with R=3.

For R=1 or often R=2, the CA doesn't sustain itself for very long, but for R > 5, the most outputs look like noise. For most of the following experiments, there is a preference towards relatively simpler CA (R=2, R=3) as they contain a more balanced mix of orderly and disorderly states.

Measuring Novelty

For this section, we explore three different metrics for capturing novelty: steps to repeat, sparsity, and entropy change.

Steps to Repeat: The first and most basic novelty metric is the number of steps to a CA performs before falling into a loop. The motivation behind this is that the simplest definition of a novel state would just be whether that state has been generated before. Once a state repeats, every state after that would also be a repeat because the rules of the CA remain the same. This or a similar metric capturing the number of steps taken until a dead state is reached is commonly used to capture the degree of evolutionary activity in a system (E. L. Dolson et al. 2019). This simple metric gives quite a high score for rules with more interesting outputs and a low score for those that appear more basic. These are sample outputs of rules that performed well and poorly according to this metric.

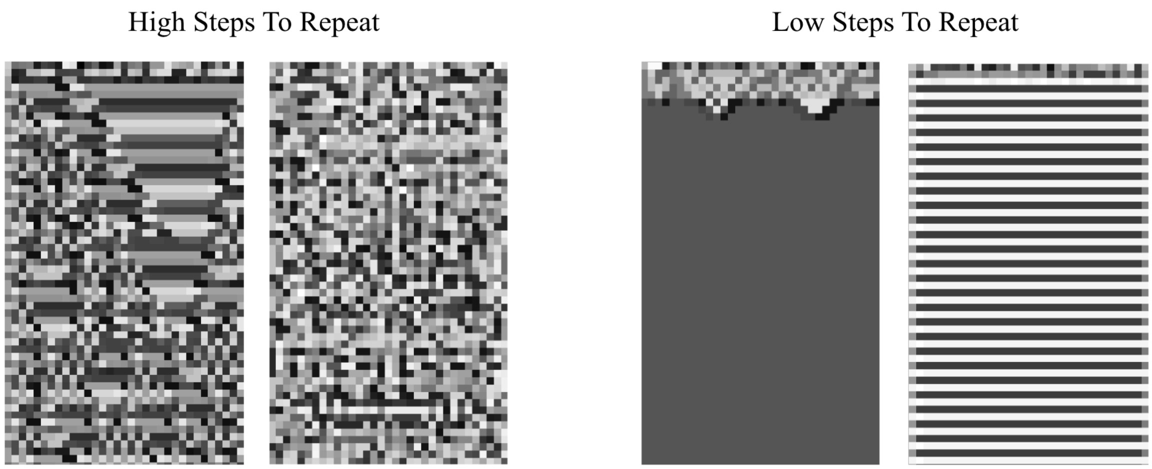

Figure 3: Sample generations of rules with high and low steps to repeat.

One aspect of novelty that this metric fails to capture is how different two states are. If N generated states are all different, but each cell is close in value to its previous states and in the next state, the cell values change significantly, there should be a way to give that state a higher novelty score.

Sparsity: This metric is one commonly used in novelty search algorithms and naturally captures the issue with the steps to repeat approach (Lehman et al. 2012). Sparsity is the average distance between the k closest previous states to the current state. Shown below is the equation for sparsity where m_i is th with closest state in the state history to state.

$$sparsity = \frac{1}{k}\sum_{i=0}^{k} dist(state, m_i)$$

Sparsity measures how new and unexplored a certain point is in the state space which naturally captures much of what is intuitively considered novel. The continuous version of this metric simply weights the distance between a pair of states so smaller distances have a larger weight and larger distances have a near zero weight.

$$sparsity = \frac{1}{N}\sum_{i=0}^{N} sigmoid\left(-2 \cdot dist\left(state, state_i\right)\right) \cdot dist\left(state, state_i\right)$$

These are sample outputs of rules that performed well and poorly according to sparsity.

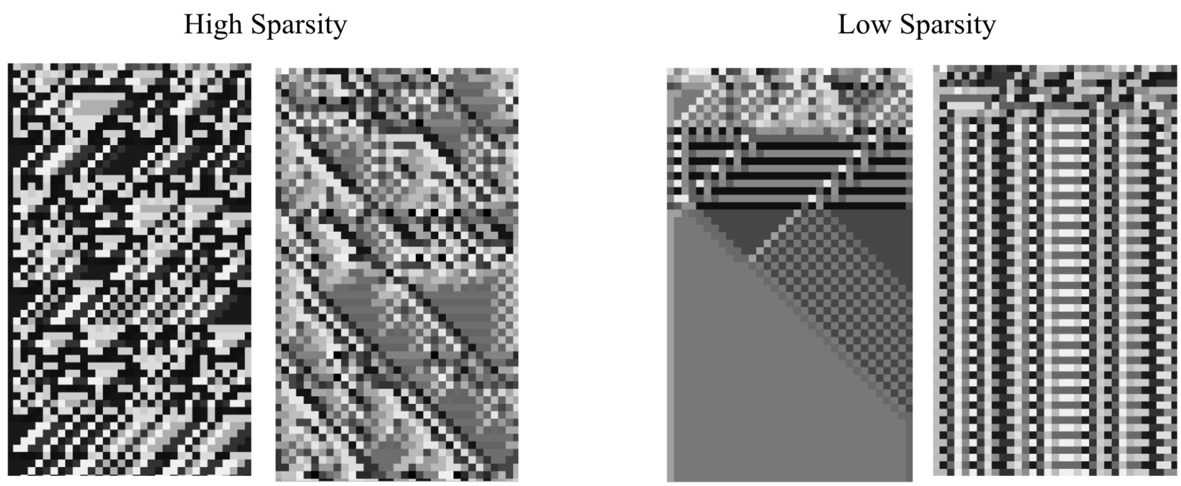

Figure 4: Sample generations of rules with high and low sparsity.

Both the steps to repeat and sparsity metrics still miss one other important aspect of novelty. That is, two states may be different and on opposite sides of the state space, but both appear to be noise. A rule that constantly generates noise would be very novel by both metrics, but in a sense, all states are the same since they're all noise.

Entropy Change: This metric is meant to capture the fact that a basic state transitioning into a noisy state is more novel than a noisy state transitioning into a noisy state. The entropy of a state captures how noisy it is on a scale from 0 to 1. The change in entropy thus captures how much the system moves between order and disorder.

The continuous version of this metric is the spread change. The spread is simply the standard deviation of the distance between a cell value and the average cell value.

$$spread = sd\Big(dist\big(state, \mathrm{mean}(state\ history)\big)\Big)$$

This measure correlates well with entropy change, but is on a different scale and so is shown on a different plot in NetLogo. The following are sample generations from rules where entropy change spiked and ones where it remained constant.

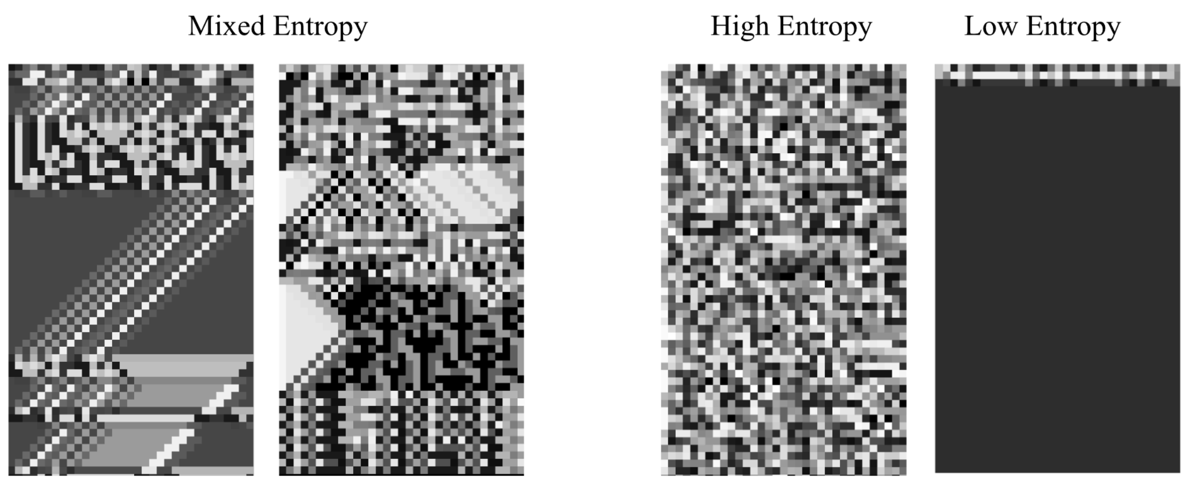

Figure 3: Sample generations of rules with high and low entropy change.

Other Metrics: A number of other metrics were explored in addition to the ones described above. Before sparsity, a metric measuring the average difference between slices of sample generations was explored to simply measure how different each state was from each other state on average. Another metric was the entropy of entropy values, motivated by the fact that, a system that explored many entropy states instead of just fluctuating between very high and very low entropy states should be considered more novel. These three metrics were specifically selected because they were the simplest, least redundant, and most cohesive in what aspects of novelty they captured. One advantage sparsity has over entropy change is that it drops to zeros when the CA starts repeating. Entropy change drops, but continues to fluctuate slightly. In theory it is possible for entropy change to continue fluctuation a lot in a repetitive state, but this was not actually observed. The reason why steps to repeat is kept even though sparsity captures when states repeat is because steps to repeat is a very understandable and consistent across different rules.

Measures on Sample Rule Generations

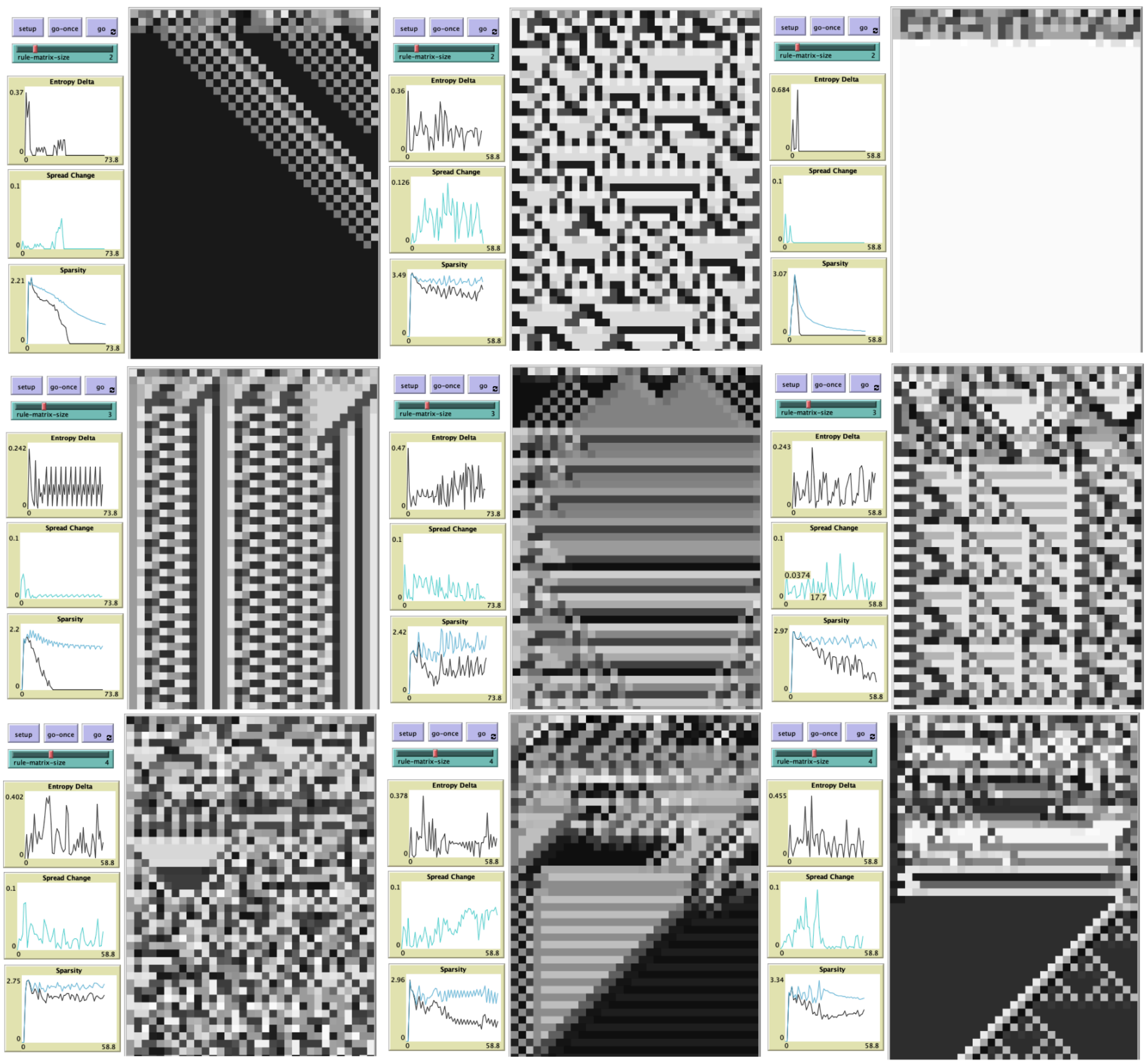

Figure 6: Novelty measures on different sample generations (The first plot shows entropy change, the second is the spread change, and the third shows sparsity and continuous sparsity as they are on the same scale)

Running these metrics on some sample generations from random rules yielded some interesting characteristics of the metrics. For example, a bit before the output of a rule is about to enter into a loop, the sparsity metric starts quickly and consistently declining. This decline happens before the output qualitatively looks like it is about to enter a loop and so the ability to capture that is very interesting. Another interesting characteristic of sparsity and entropy change is that when the output enters a new "phase" where the newest generated states follow a very different pattern from the old ones, entropy change spikes and sparsity rises consistently and falls consistently when the new phase goes on for a while which matches an intuitive sense of a novel pattern losing it's novelty after some time. One other interesting point about entropy change is that even though it was created with the intention of increasing when states go from noisy to stable or vice versa, it also increases for some patterns which don't appear noisy but continuously produce different and intricate shapes like the one on the top center in figure 6.

Maximizing Novelty (400 Level Component)

The objective of maximizing novelty has two objects. First, if the metrics are sufficient to actually learn rules which have maximal novelty, it could suggest that having such metrics for other hallmarks of open-endedness could allow for directly optimizing a system for OEE. An idea which sounds contradictory, but may be feasible with robust metrics. Second, the outputs of the rules optimized for novelty should appear naturally novel, if not, it would suggest the metrics are not robust and approximate novelty rather than capturing its essence.

To perform these, experiments, the generated samples of the learned rules are first quantitatively compared with a baseline which is a sample of randomly generated rules for R=2 and R=3 and then qualitatively compared to see if intuitions about novelty align with the sample generations.

Neural Network

A convolutional neural network (CNN) was used to train a neural network to take an entire state of W cells and generate the next state. The first convolutional layer looks at a cell and its two neighbors and the last layer outputs a single value between 0 and 1 for a single cell. Because of the local connectivity of a CNN, the same weights are used to generate the states for other cells, making it a valid way of generating states for a cellular automaton. This use of a CNN for CA was described in (Mordvintsev et al. 2020).

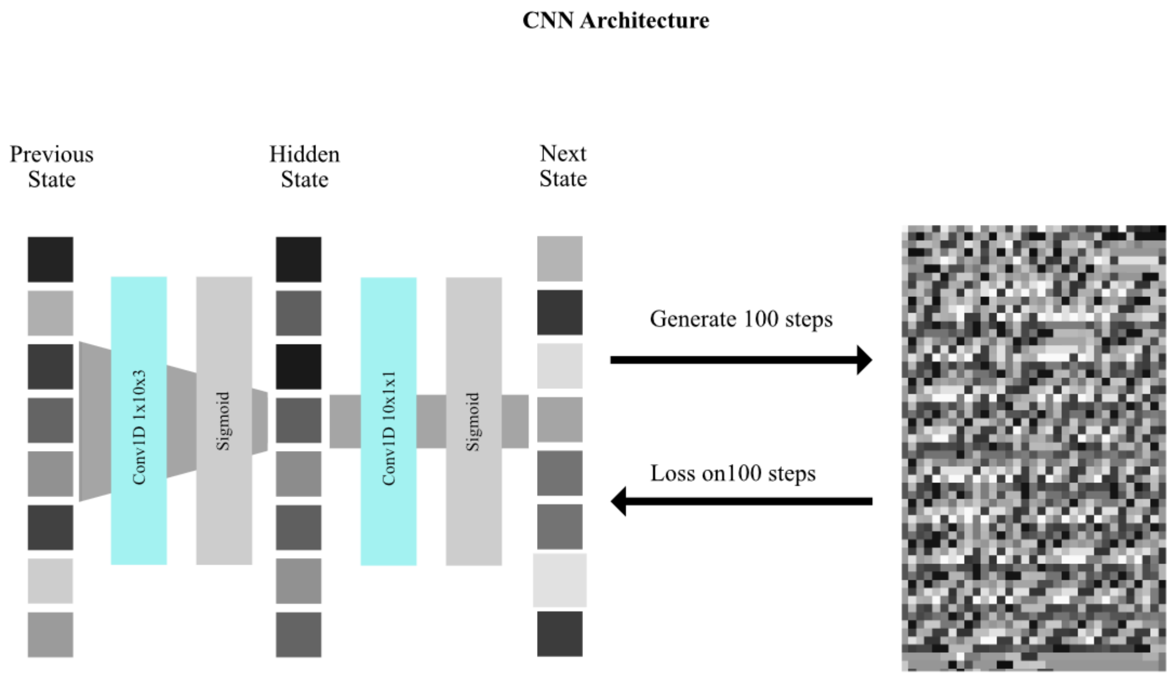

Figure 7: CNN architecture to generate next cell states given current states.

Before using the continuous novelty metrics as an objective for this model, it was first trained to reproduce basic CA rules to see if it was capable of being able to output sophisticated patterns and the training worked properly. The output of the CNN over multiple steps was compared with the output of the rule over the same number of steps and using mean squared error between the two as loss, the CNN could be updated in the direction of the rule. This worked well, even when trained to fit to the more complex binary CA rules, it was able to reproduce the rule almost perfectly, learning to only output values very close to 0 or 1. An interesting point about this though was that the loss initially dropped slowly and then stagnated for a while until it rapidly dropped to near zero suggesting the loss landscape of just this one rule was quite rugged.

In order to train this network to maximize the novelty metrics, the network generated a number of states and the negative of each of the novelty metrics was computed over them as the loss. The training approach was to compute this loss over multiple batch generations and experimented with 1,000 - 10,000 epochs for each metric.

Genetic Algorithm

Next, a genetic algorithm was used which can directly optimize for the discrete metrics is a genetic algorithm. An individual in this population is a rule and the rule matrix are the genes. Mutation was done by adding a random value between (-MUTATION_DEGREE, MUTATION DEGREE) at the specified mutation rate. Here, the initial population was 300 rules and the fitness was computed using each of the novelty metrics over 100 generation steps for R=2 and 200 generation steps for R=3. The other parameters including mutation rate and mutation degree were kept fixed. More detailed experimentation was not done with the mutation parameters because the rules turned out to be very sensitive to mutations. Very high fitness rules often performed very poorly with slight mutations. Thus mutation rate and mutation degree were kept at 0.01 and 0.1 respectively.

RESULTS

Neural Network

For each metric the CNN was trained on, the loss dropped consistently in the beginning, but quickly went flat and became very noisy with some minor fluctuations. The outputs were completely dead patterns. The same training objective was tried for a simple training objective, the mean difference between generated states. Even here the CNN fell into a local minimum of cycling every 2 states between 0 and 1. With a number of modifications of the loss function, the output remained very basic, varying very little each time. These are examples of the outputs.

CNN Rule Sample Generations

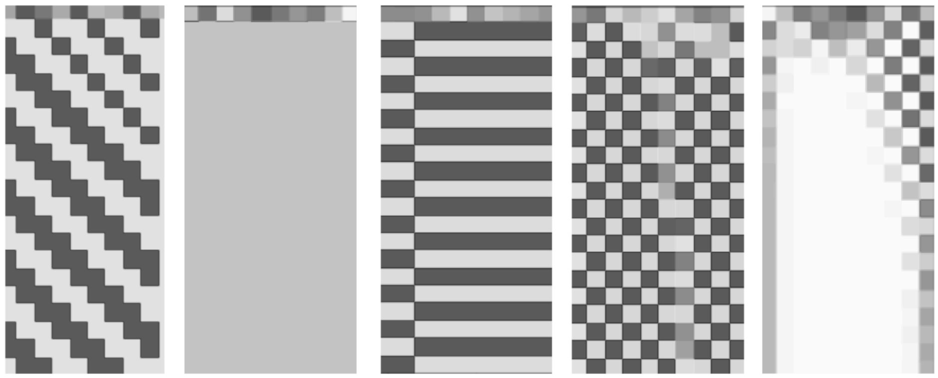

Figure 8: Trained outputs of the CNN all fell into local minimum producing very basic patterns.

Other minor modifications to the loss function, larger networks with larger and more layers were tried to overcome this falling into local minimum as well as smaller batch size to increase the volatility of the training trajectory with no success. The loss followed the same pattern. The difference between using the continuous sparsity metric and the continuous version of the entropy change metric was that sparsity showed a more consistent downward trend whereas using the continuous version of the entropy change metric as a loss fluctuated more and generally did not decrease very much.

Genetic algorithm

For R=2 and R=3, the genetic algorithms were run for 20 and 50 generations respectively. Throughout training, the average fitness improved rapidly in the beginning and very slowly after. The individual in the population with the best fitness only improved slightly in the beginning and seemed to hit a barrier and stayed at that fitness level after that.

These are the result of the genetic algorithm learning rules for R=2 and R=3 using each metric as the fitness. The results show the mean and standard deviation of each novelty measure for the final population. The baseline scores are calculated on 300 randomly sampled rules for both R=2 and R=3.

Rules (R = 2)	Steps to Repeat	Total Sparsity	Total Entropy Change
Randomly sampled (Baseline)	$\mu = 19.22$, $\sigma = 16.06$	$\mu = 22.50$, $\sigma = 17.37$	$\mu = 0.98$, $\sigma = 0.88$
GA (Steps to repeat)	$\mu = 69.57$, $\sigma = 2.47$	$\mu = 59.50$, $\sigma = 17.16$	$\mu = 0.85$, $\sigma = 0.36$
GA (Sparsity	$\mu = 60.97$, $\sigma = 8.31$	$\mu = 86.85$, $\sigma = 10.97$	$\mu = 0.92$, $\sigma = 0.18$
GA (Entropy change)	$\mu = 20.88$, $\sigma = 6.31$	$\mu = 25.41$, $\sigma = 5.23$	$\mu = 1.60$, $\sigma = 0.28$

Rules (R = 3)	Steps to Repeat	Total Sparsity	Total Entropy Change
Randomly sampled (Baseline)	$\mu = 53.99$, $\sigma = 30.50$	$\mu = 54.09$, $\sigma = 23.63$	$\mu = 1.13$, $\sigma = 0.411$
GA (Steps to repeat)	$\mu = 244.99$, $\sigma = 34.96$	$\mu = 209.07$, $\sigma = 21.29$	$\mu = 3.58$, $\sigma = 0.40$
GA (Sparsity	$\mu = 75.34$, $\sigma = 14.40$	$\mu = 107.59$, $\sigma = 12.99$	$\mu = 0.85$, $\sigma = 0.14$
GA (Entropy change)	$\mu = 19.56$, $\sigma = 5.83$	$\mu = 20.16$, $\sigma = 5.13$	$\mu = 2.75$, $\sigma = 0.12$

Figure 9: Results of the genetic algorithm optimized for each metric.

The results show that optimizing the genetic algorithm on steps to repeat and sparsity yield the best performance whereas optimizing on entropy change sometimes does worse than baseline on the other measures. Even the ones optimized on steps to repeat and sparsity appear to sometimes do worse on entropy change than the baseline.

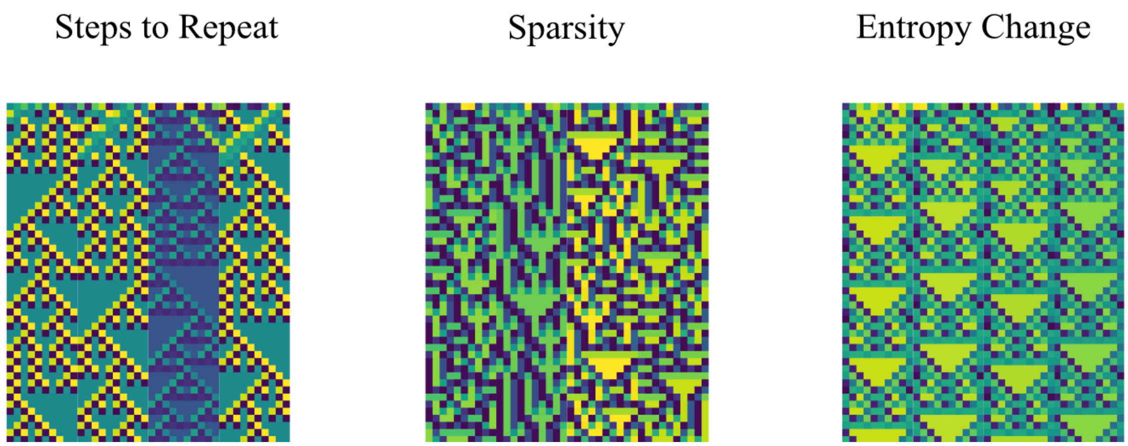

Figure 10: Four sample generations of genetic algorithms trained on each metric (R=2)

DISCUSSION

Results Analysis

It appears that, for the CNN, as mentioned in the case of the test trying to reproduce a rule with the CNN, the loss landscape is very complex and it is too easy for the network to fall into a very basic local minimum. It may have dropped rapidly after finding its way out of the local minimum, but this was not observed over the course of 4000 epochs for a number of runs. Perhaps a more cleverly designed loss function and significantly longer training time would allow the CNN to overcome these limitations, but if the underlying loss landscape is too complex for a proper measure of novelty, it may just be difficult for gradient based optimization using this architecture.

By looking at some of the sample generations of the genetic algorithms trained on each novelty metric, we can see that the mutation method yields relatively little diversity in the generation as each. While the ones trained on steps to repeat and sparsity appear more novel than the one trained on entropy change, the offspring of the top performing rules yield relatively little difference in their dynamics which likely means that the genetic algorithm is mainly improving by simple creating slight variations of the top performing rule and not actually learning to form a rule to maximize novelty. The rules with the highest fitness at the beginning of training improved very slowly and quickly stopped improving. This was less the case with R=3, but still held. This again supports the idea that the genetic algorithm was mainly just creating slight variations of a good performing rule that was randomly found. The main limitation that resulted in this was the fact that the outputs were very sensitive to mutations in the rule and slight changes to the rule matrix often resulted in significant drops in fitness, but the opposite was not observed. This may suggest that a property of the rule matrix is that slight local changes to the rule matrix in the space of all possible matrices don't significantly change its dynamics. Thus, one potential future direction would be to try a different method of randomly generating rules.

Regarding the genetic algorithm results, optimizing on steps to repeat performed the best in terms of the other metrics perhaps because it computed a measure of novelty over the entire sample generation of the CA and did not compute intermediate values before the CA was finished which may have suggested an output is very novel in the short term, but perhaps not in the long term. Another potential reason is that a rule probably needs to have a high sparsity and entropy change to survive for more steps, but perhaps does not need to survive for many steps in order to achieve a high sparsity or entropy change. While the genetic algorithms optimized on entropy change performed the worst, the fact that only R=2 and R=3 may have biased the results. For higher R values, the sample generations of the CA rules would be noisy and relatively less novel. This is where entropy change would more naturally suggest low novelty and steps to repeats would suggest high novelty. This is an important extension to the experiments which should be run if there was more time to continue this project.

Future Directions

There were a number of metrics, methods of optimizing novelty, and ways to improve the current experiments which could not be done due to time restraints for this project, but would be very interesting ideas to explore. First, instead of training the CNN directly on novelty measures, another neural network could be used to approximate novelty measures and the original CNN could use this approximation to optimize the generations for novelty using an approximate and simpler loss landscape. Another interesting future direction would be to explore these novelty measures on different types of systems and compare how much more flexibly they can be applied to different kinds of systems that those proposed by Adams et al. and E. L. Dolson et al.

CONCLUSION

In this paper, we explored three different novelty metrics aimed at capturing what we intuitively consider novel and are simple enough to apply in different agent based systems. In the qualitative analysis, we see that the metrics captured interesting and uninteresting aspects of the 1D continuous CA. We then attempted to maximize the novelty of the 1D CA system using a CNN and genetic algorithms. The CNN was unsuccessful as the loss landscape appeared to be very complex and easy for the model to get stuck in local minimum. The genetic algorithms trained on each metric had some limitations as well with regards to increasing the novelty of already novel states due to the sensitivity of mutations, but we saw that the learned rules are quite novel by each metric and qualitatively compared to the randomly sampled rules.

REFERENCES

1. Dolson, Emily L. et al. “The MODES Toolbox: Measurements of Open-Ended Dynamics in Evolving Systems.” Artificial Life 25 (2019): 50-73.
2. Adams, A. et al. “Formal Definitions of Unbounded Evolution and Innovation Reveal Universal Mechanisms for Open-Ended Evolution in Dynamical Systems.” Scientific Reports 7 (2017): n. pag.
3. Sughimura, Noritsugu et al. “Non-uniform Cellular Automata based on Open-ended Rule Evolution.” Artificial Life and Robotics 19 (2014): 120-126.
4. Wolfram S (2002) A new kind of science. Wolfram Media Inc, Champaign
5. András, P.. “Open-ended evolution in cellular automata worlds.” ECAL (2017).