AssemblyCA: A Benchmark of Open-Endedness for Discrete Cellular Automata

Keith Y. Patarroyo
Complex Chemistry Labs
Glasgow University
Glasgow, UK
keith.patarroyo@glasgow.ac.uk

Abhishek Sharma
Complex Chemistry Labs
Glasgow University
Glasgow, UK
abhishek.sharma@glasgow.ac.uk

Sara I. Walker
School of Earth and Space Exploration
Arizona State University
Tempe, AZ, USA
sara.i.walker@asu.edu

Leroy Cronin
Complex Chemistry Labs
Glasgow University
Glasgow, UK
lee.cronin@glasgow.ac.uk

Abstract

We introduce AssemblyCA, a framework for utilizing cellular automata(CA) designed to benchmark the potential of open-ended processes. The benchmark quantifies the open-endedness of a system composed of resources, agents interacting with CAs, and a set of generated artifacts. We quantify the amount of open-endedness by taking the generated artifacts or objects and analyzing them using the tools of assembly theory(AT). Assembly theory can be used to identify selection in systems that produce objects that can be decomposable into atomic units, where these objects can exist in high copy numbers. By combining an assembly space measure with the copy number of an object we can quantify the complexity of objects that have a historical contingency. Moreover, this framework allows us to accurately quantify the indefinite generation of novel, diverse, and complex objects, the signature of open-endedness. We benchmark different measures from the assembly space with standard diversity and complexity measures that lack historical contingency. Finally, the open-endedness of three different systems is quantified by performing an undirected exploration in two-dimensional life-like CA, a cultural exploration provided by human experimenters, and an algorithmic exploration by a set of programmed agents. A web version of this paper which shows animations for the different CAs evolution is available in https://assemblyca.github.io/.

1 Introduction

The construction and exploration of open-ended worlds has received increased attention in the AI community in the recent years [30, 18, 36, 35, 12, 45]. Within these environments one is able to not only explore the agent capabilities, but properties like multi-agent behaviour, self-organization and long-time horizons. Moreover, with the recent introduction of LLM agents, the construction of agents that can acquire an increasing amount of skills inside an open-ended world is now a reality [41, 40].

While the field of open-endedness has a rich conceptual literature, there is no underlying benchmark to characterize whether a set of agents are undergoing open-ended evolution [33]. Additionally, the actual measures of open-endedness as diversity or complexity ignore the historical contingency of the object and they lack a conceptual framework upon being interpreted. None of these measures are suited for dynamic patterns like cellular automata(CA). As the community delve deeper into the study

2nd Workshop on Agent Learning in Open-Endedness (ALOE) at NeurIPS 2023.

of potentially evolving agents [23], we hypothesised that it maybe be useful to develop a well-defined benchmark to facilitate robust advancements in research exploring open-ended systems.

In this paper we present *AssemblyCA*, a new benchmark aiming to fill this gap. AssemblyCA consists of a framework that analyses the products of a system consisting of agents, resources and discrete CA. The amount of open-endedness of this environment will be quantified by two measures that quantify the historical contingency of an object, the *complexity* and the *copy number* of a given product or *object*. The conceptual foundation of these two quantities comes from the Assembly Theory(AT) framework, [27, 32]. AT captures *historical contingency* by exploring the fact that highly complex objects need to have lower complexity sub-parts exist in the system, otherwise they cannot exist. Therefore there is a *causal* property to the object formation.

In Section 2 we give an overview of the current approaches present in the field of open-endedness and discuss their current limitations. We then discuss the work done regarding neural networks and CA, and their relation to our benchmark. In Section 3 we give the basics of AT to describe the specifics of our benchmark, we then explore it with some test CA patterns and compare it with other open-endedness measures. In Section 4, we first present two different exploration case studies to understand the open-endedness signature in a baseline of evolving and non-evolving systems. Finally we analyse the exploration of a system of heuristic agents and analyse their open-endedness using our framework.

2 Related Work

Most similar to our work is the literature relating to open-endedness environments in videogames [9, 10, 21, 17, 8]. The Game of Life is used as an open-ended environment in [9, 10], however the measure of complexity used is JPEG compression thereby ignoring the historical contingency of dynamical patterns of CAs. Kepes et al. [21] introduce an artificial chemistry environment that uses the pathway complexity [27] as a measure of the different objects or molecules that the agent generates. Grbic et al. [17] similarly focuses on the generation of artifacts in Minecraft as an open-ended environment, but does not characterize their complexity. Davis [8] just like this work attempts to generate patterns for life-like cellular automata as human experimenters have done for the last five decades in an open-ended fashion. Additionally Davis [8] mostly focuses on the introduction of this open-environment, but the current work focuses on the complexity quantification and a rigorous assertion of open-endedness.

A rigorous conception and quantification of open-endedness has been much of the focus of previous research in the topic [33, 39, 2]. Taylor [39] generates a hierarchy of different stages of open-endedness, however in our current work there is no differentiation between stages since new objects are generated in each stage. Adams et al. [2] tries to capture the idea of open-endedness in cellular automata by introducing an organism and an environment, but the framework is not able to analyse large automata with long time-scales. Song [33] makes a survey of the field of open-endedness concluding a mix of conceptual definitions and a lack of a sophisticated measure for open-ended environments. For cellular automata you can use jpeg compression [9], algorithmic specified complexity [11] or entropy [22], but these measures or other [24] do not quantify the historical contingency of complex objects in CAs. Our work improves on this element by defining new measures that capture historical contingency and can quantify dynamical patterns in CAs.

The use of cellular automata(CA) [20, 43], in the neural network literature has received some attention in the last few years [12, 40, 9, 10, 17, 8, 34, 37]. While some of the work has been focused on the use of neural networks for predicting discrete cellular automata behaviour [34], in this work we focus in the open-endedness and the generation of new objects and the possible use of neural nets to solve this problem. Fan et al. [12], Wang et al. [40], Earle [9], Earle et al. [10], Grbic et al. [17], Davis [8] use neural nets to explore an open-ended environment by means of different learning architectures, Earle [9], Earle et al. [10], large-language models Fan et al. [12], Wang et al. [40], and evolutionary algorithms Grbic et al. [17]. The task of generating objects in an open-ended environment has also used neural nets for object construction [17, 8, 37], where the main task of the neural net is to find new objects in these environments.

In the intersection between AI and Artificial Life, the investigation of open-ended environments in continuous CAs has received lot's of attention in the last years [5, 28, 6, 14]. In these approaches the rules of the system are optimized in order to obtain an automata that can produce novelty as the system evolves in time. Similar ideas have been explored in discrete CAs [44, 29], where rules are searched to generate a system capable of open-endedness. However in this work we take a different approach and the rule is fixed, but the search-space is explored by agents modifying the initial conditions of the CA. In this way, the agents are constantly discovering new objects like human experimenters.

The exploration of discrete CA patterns has a decade long history by human experimenters [1]. Practitioners of cellular automata search patterns by either random soup search [16, 13], by combining existing patterns together and by looking at particular properties of patterns via algorithmic search [20]. The self-sustaining nature of such patterns have been studied [4], but in this work like in Davis [8], we'll assume the categories of current searches and not worry about the nature of the object over time. For the large-scale simulation of life-like cellular automata, the algorithm Hash-Life [31, 15] is widely used and will also be of importance for our approximation of pathway complexity in large-scale patterns.

Designed to detect the emergence of selection in systems that precede early life and have no self-replication capacity, Assembly Theory(AT), [26, 27, 32, 19] quantifies object construction processes by measuring the shortest path to generate an object and the number of copies of such object. The framework of AT was initially formulated to quantify the complexity of chemical molecules and use this measure for generating agnostic biosignatures [26, 19]. Such original idea has been generalized to a mathematical framework for arbitrary objects [27] and to an explanatory framework [32] in order to quantify the emergence of selection by quantifying the historical contingency of an object. If a process generating objects systematically utilizes already created objects, then the system has undergone selection. This idea is the one we use in this work to characterize the open-endedness of our cellular automata system.

3 AssemblyCA benchmark specification

The main goal of the AssemblyCA framework is to rigorously formulate the simplest environment where one could potentially obtain an open-ended process. In this case consider a set of agents that have access to a bundle of resources to modify CA initial conditions and execute them to find objects. These agents might be able to execute a fixed size cellular automata, and insert black, white or higher level cells on this CA, or modify already found patterns in order to find new patterns in the output of the CA. The obtained objects by these agents are quantified by their assembly space and the number of copies found by each agent of such object. This methodology can be seen in Figure 1.

3.1 Assembly Theory

Given a forward dynamic process leading to object construction, the aim of AT [32] is to quantify the amount of selection is needed to build complex objects abundantly. This is possible given a pool of building blocks and lower complexity objects. For this a system that explores the object-space with an undirected process is characterized as not selective, when this occurs complex objects are produced in low number, Figure 2. On the other hand, a directed process is one that utilizes already build objects to generate complex objects in high numbers. From the distributions of copy number and complexity, selection can be discerned.

This selectivity signature is the minimal feature we require for a system to present open-ended evolution. The undirected-directed transition can be thought of the explore-exploit tradeoff in reinforcement learning [38], an undirected process is trying to explore as many novel objects as possible, but as a consequence they are of low complexity. On the other hand a directed process uses the objects found before to build new objects. However a purely directed process can generate high complexity objects in low quantity, so the selectivity transition is when there is high complexity objects in abundance. In other words a mixed explore-exploit strategy.

Object. In AT a fundamental concept is that of the *object*, and at its foundation it can be defined as a pattern that persists over time in a physical substrate, Figure 2. As an example consider a DNA strand as an object, the pattern is the sequence of nucleotides, and its chemistry is stable in time. With respect to CA, an object satisfies the intuitive properties of, finiteness, composability, distinguishability and constrained. For simplicity in CA the object is defined as smallest rectangular array of cells that contains the entire pattern, or bounding box. As an example consider the pattern

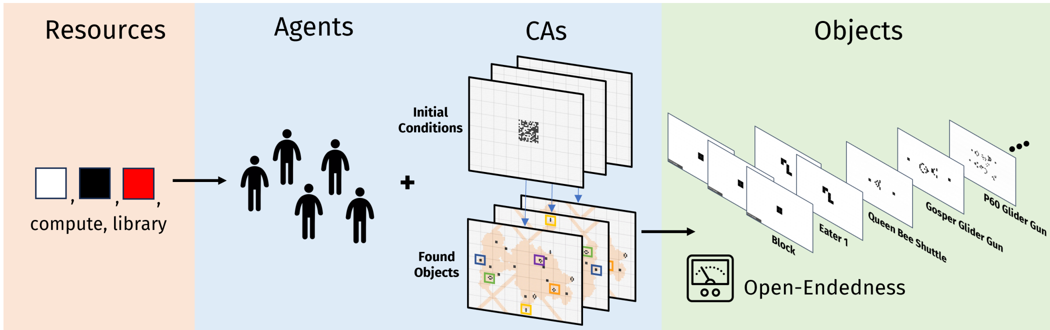

Figure 1: **AssemblyCA Framework**. Given system where a set of agents can interact with a CA world and they have access to a limited amount of states, computation, and a library of discovered patterns. The agents will generate as a byproduct a series of objects, or CA patterns by modifying the CA initial conditions. By analysing the output of objects of this system the amount of open-endedness of the system is quantified.

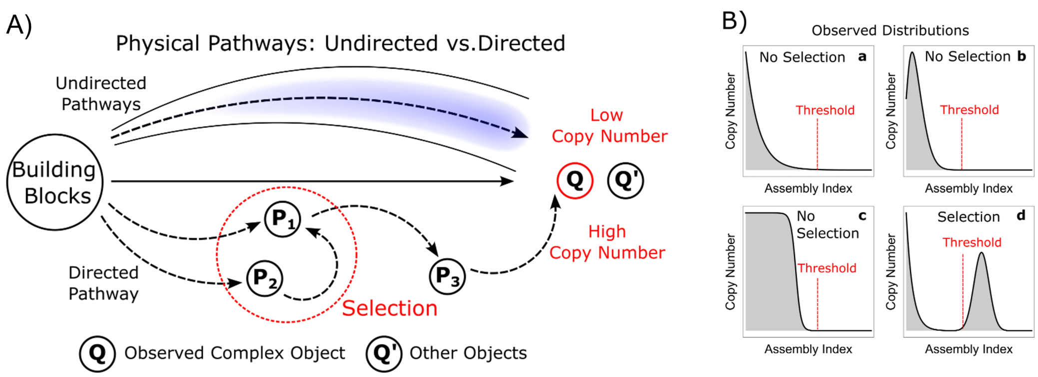

Figure 2: **Selection in Assembly Theory adapted from [32]**. **A)** Representation of pathways to construct objects with undirected and directed processes (selected) leading to the low and high copy numbers of the observed object. **B)** Distributions of copy number vs assembly index(complexity) for processes that undergo selection or not.

known as beacon in the Game of Life CA,Figure 3b. It is finite, it is composed of smaller patterns, it is distinguishable dynamically since it has a period of two configurations and it is constrained to be found in a CA of enough size. Given this properties we can clearly denote such object and its *copies* in a given substrate, Figure 3c.

Assembly Space Given an object, and a set of building blocks, we can find the shortest path to generate such object given a set of elementary operations [27]. If we have an object that is defined in a set of different configurations $\{S_i\}_{i=1}^{T}$, we take all the configurations and find a shortest pathway to build all at the same time, this is what we denote as assembly space $\Gamma(S_1, ..., S_T)$, Figure 3d.

The assembly space encodes the dynamical behavior of a CA object, and its historical contingency. The number of steps to build the object is what we'll denote as **assembly index** a_{CA}. By calculating the overlap between configurations, structural properties of the pattern can be quantified. In this consider the following two quantities, defined as **memory** and **assembly distance**,

$$M_{CA} = \frac{1}{T}\sum_{n=1}^{N_{frag}} r_n, \qquad D(S_1, S_2) = \bigcap_{\text{i}=1}^{2} \Gamma\left(S_i\right),$$

where T is the number of configurations, N_{frag} are the number of elements of $\Gamma(S_1, ..., S_T)$, r_i is the number of times a fragment i contributes for different configurations S_i. Also Note that for one configuration, $M_{CA} = A_{CA}$.

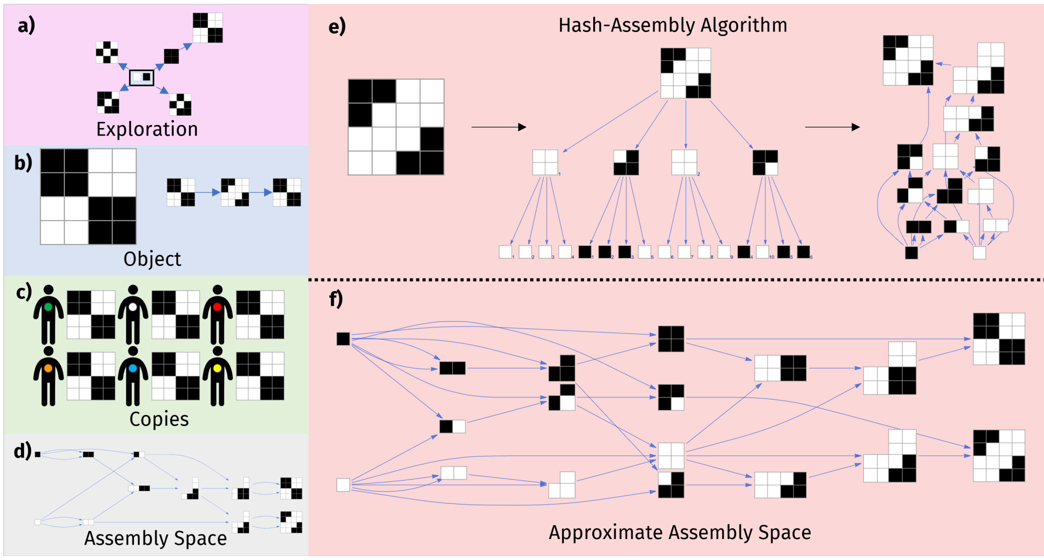

Figure 3: **Assembly Theory measures and approximations in CA**. **a)** Given a set of starting building blocks a process that explore patterns, in this case in the Game of Life CA. **b)** An object that oscillates every two generations is found as a result of the exploration. **c)** Different agents have found the object and have added it to each own library. **d)** The complexity of this object is calculated by finding the minimum way of building its two configurations together and calculating the properties of this space. **e)** Approximation algorithm for an arbitrary $2^n \times 2^n$ block, by doing a quad-decomposition and joining the common fragments one can find an assembly pathway. **f)** By joining the reconstructed pathways of all the configurations of the object one obtains an approximation of the assembly space.

The memory of a CA tries to capture the mean assembly index of a pattern $\{S_i\}_{i=1}^{T}$, by putting more weight on the fragments that are repeated in more configurations. Additionally, an object exhibits infinite growth, the number of configurations is infinite, however the number of repeated fragments grows linearly in the assembly space, therefore the memory remains finite, Figure 4.d. The assembly distance captures the common fragments between two configurations, and allows for a comparison of similarity between objects.

Hash-Assembly. The Assembly Space of a CA was shown for a small object in Figure 3d, however the general problem of finding an assembly index of a configuration is NP-Complete, see Appendix [25]. However the computation assembly spaces with a very large number of cells is highly desired. So we approximate each configuration using an algorithm inspired from the Hash-Life algorithm [31, 15]. First for simplicity we will assume that we have a $2^n \times 2^n$ grid, then we follow an analogous procedure as Hash-Life: 1. A quad-tree is built from the configuration, 2.The the common fragments are matched to generate the approximate space $\Gamma^*(S_1, ..., S_T)$, Figure 3e,f. In practice if we run a CA using the Hash-Life algorithm, the first step is computed at each time-step, so any Hash-Life implementation can be used as a step to compute an approximate Assembly space.

From this approximate space we can define analogously the hash-assembly index $a^*{}_{CA}$, memory $M_{CA}(\Gamma^*)$ and hash-assembly-distance D^*. Now with this approximation the memory of different patterns over an increasing number of configuration is computed until convergence. The memory behavior over time-step is computed for a still pattern, an oscillator, a spaceship and a linearly increasing pattern, Figure 4. The memory is able to be computed even for infinitely growing patterns, this is done by computing the memory after a large enough time, until it has reached an asymptotic value, in this way the total memory is approximated.

From this point onwards, we drop the hash from our set of quantities, and we refer to **memory** and **complexity** interchangeably unless we say otherwise.

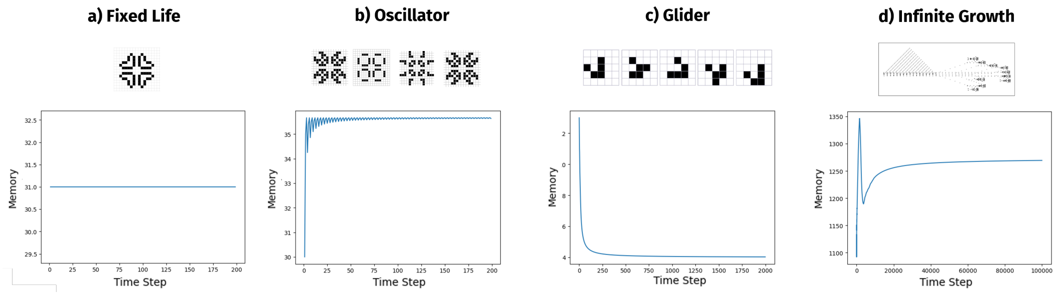

Figure 4: **Memory of different types of patterns**. **a)** The memory of a fixed life is constant and equal to the Hash-Assembly-Index. **b)** The memory of an oscillator of period 3increases until the pattern repeats and then remains roughly constant. **c)** The memory of a spaceship decreases until the pattern repeats and then remains constant(note that the pattern is moving over space). **d)** The memory of a quadratic growing pattern grows in a non-linear manner until it asymptotes to a fixed value.

3.2 Entropy Comparison

A common measure of complexity, both in CA and other dynamical systems is entropy. It is commonly argued that entropy is a measure of diversity or even complexity [33, 3]. In order to compare the relationship of entropy with the assembly index, two CAs are executed in one dimension, Figure 5. Both the entropy, and the mutual information [7] will be used as a benchmark for the time evolution of the automata.

We start with a simple configuration running Rule 110, Figure 5, the entropy increases until it arrives to a maximum, but the assembly index behaves in a way that summarizes structural information of the automata behaviour at a meso-scale. Furthermore the assembly index, unlike entropy does not average spatial information to a uniform distribution as time increases.

Next we consider an an engineered initial condition running Rule 110, Figure 5. In this case the entropy is maximal the whole time , but clearly there is a spatio-temporal evolution in the pattern that is structured at the meso-scale. The assembly index both quantifies when the two gliders are moving and when they merge into a regular/trivial structure. The analysis of a random initial condition of Rule 110 and a chaotic rule is also analyzed in appendix B.

The mutual information and the assembly distance between any two timesteps of the CA was also analysed in order to capture historical contingency. The mutual information can only quantify the relationship between two close timesteps but is unable to properly compare the structural behaviour of two configurations separated by several timesteps. Therefore it's concluded that the assembly index is able to capture spatio-temporal similarities between different configurations and in this way quantify the historical contingency of the CA time-evolution.

4 Experiments

We perform three experiments to showcase our benchmark, (1) an undirected process, (2) a open-ended process that is undergoing selection, and (3) an algorithmic process with programmed agents.

4.1 Undirected Process

In the context of our benchmark, an undirected process is one that explores the space to get a highly diverse set of objects, but they are not very complex. In the case of CA, this is a classical technique called soup-search [16, 13] where the initial conditions are set at random and objects are found when the CA has reached stabilization[20]. In the AssemblyCA framework this formulated as a set of agents that explore random CA initial conditions and find objects in the resulting stabilized pattern, Figure 1.

In this case we used the database Catagolue [16] for the Game of Life CA, that has run trillions of searches starting from asymmetric 16-by-16 soups and found quadrillion of objects classified as Still Lifes, Oscillators, Spaceships or Linearly Growing Patterns. This four categories are chosen since the

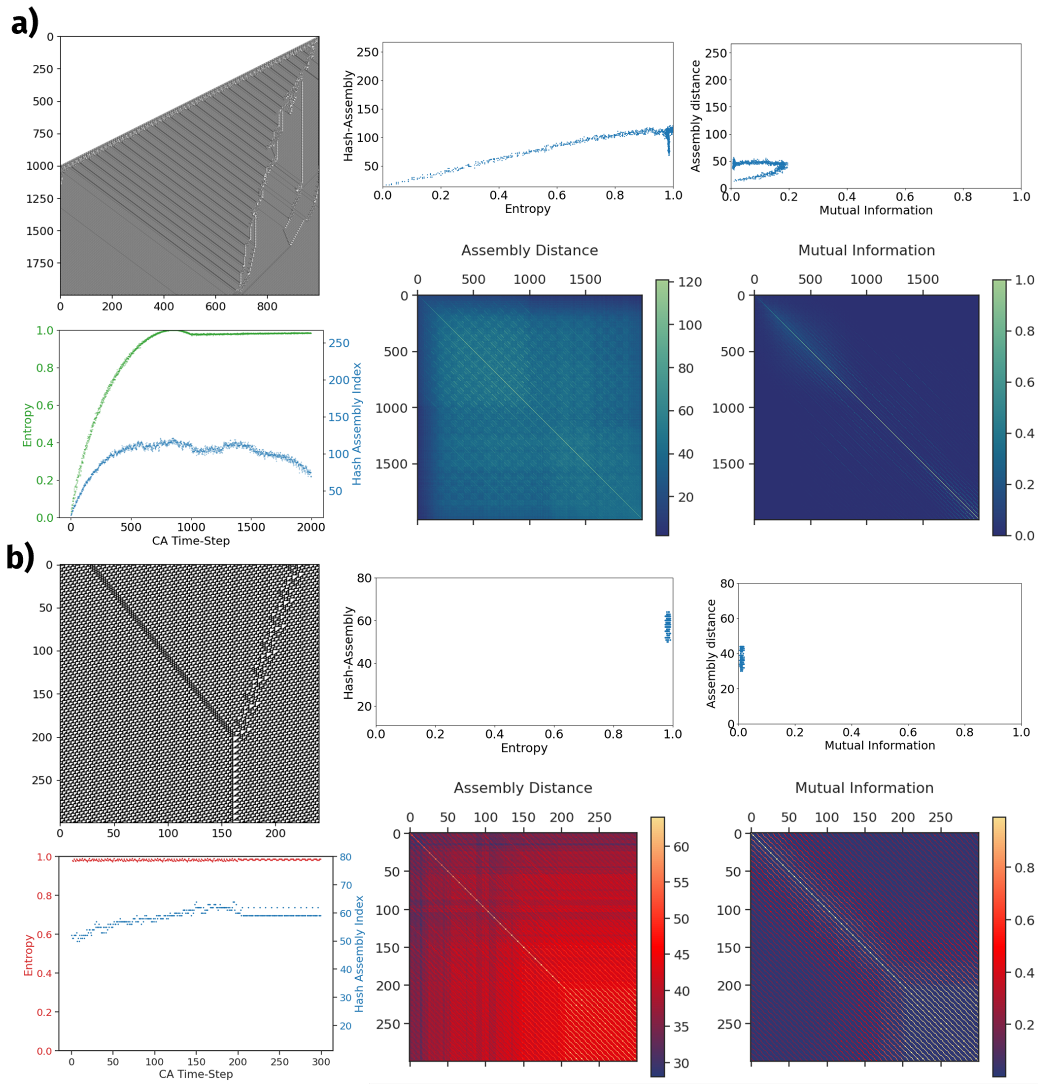

Figure 5: **Comparison between entropy and assembly. a**) Behaviour of a simple initial condition in the Rule 110 Cellular Automata. Comparison of the Assembly Index vs Entropy of the time evolution and the Mutual Information and the Assembly Distance between two arbitrary cells and subsequent cells. **b**) Behaviour of an engineered initial condition in the Rule 110 Cellular Automata. Comparison of the Assembly Index vs Entropy of the time evolution and the Mutual Information and the Assembly Distance between two arbitrary cells and subsequent cells.

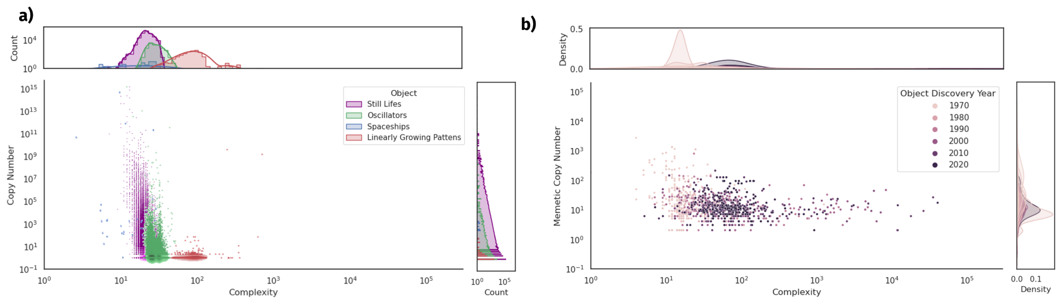

Figure 6: **Distributions of undirected and directed processes.** **a)** An undirected process of searching GoL patterns generates a large number of unique low-memory objects and a handful of unique high-memory objects. The GoL soup search discovered objects of low-memory in very high copy whereas the high-memory objects are found in very low copy. **b)** Distribution of complexity and copy number of objects found by the GoL enthusiast community. We can see that the community has been systematically finding high complexity objects in large quantities. This implies that the agents search procedure are undergoing selection, therefore it is an open-ended process.

behaviour of them is easy to check algorithmically. The number of times each object is found in the search is estimated as the copy number. For each object the memory is computed, Figure 6a.

For the undirected search the behaviour is clearly a high number of low complexity object and a low number of high complexity objects. It is also to be noted that some high complexity patterns can be found in large quantities, but these are found to be "thermodynamically" favourable patterns that do not require of any previous historical contingency to be generated. The word "thermodynamically" is an analogy to the formation of molecules that represent thermodynamic minima on an energy landscape under conditions where they are formed.

4.2 Open-Ended Process

In order to know if a process is achieving the benchmark, the objects from an actual evolving system are analysed. Consider the objects found by the community of Game of Life CA practitioners [1]. In this case we take the 50 year pattern collection of the LifeWiki database and compute their complexity with the methods described above. Since this patterns have been found by humans, the copy number of these patterns is estimated. In the database each of the objects has a named attached to it. With this on mind the LifeWiki is scrapped by the number of times an object is mentioned in the Wiki. This memetic copy number is what we consider as an approximation of the copy number of the patterns when they've been found by actual practitioners.

In this case the exploration is clearly undergoing selection. There is a distribution of objects that are found in very high copy and also are very complex, Figure 6b. Also the objects found are much more complex than the ones found in the undirected process, Figure 6a. This proves that the objects found have a historical contingency and empirically proves the well known fact that constructions by practitioners of Game of Life are built based on simpler constructions [20].

4.3 Algorithmic Process

For this process a set of heuristic agents is considered to probe the benchmark. These agents find objects as follows: 1.) They start with a library of already found patterns. 2.) They randomly select a pattern. 3.) The grow the pattern around its first neighbours. 4.) The resulting object is classified [16] and if found to be novel it is added to the library. 5.) If the object is used to find a new object or re-discovered by a different agent, its copy number increases. 6.)This process is repeated until desired, Figure 7a.

In this process we consider ten agents following the previous procedure. At each repetition the agents are able to pool from the library, and at the end of the process they all put their found objects in the library. At each new repetition the library is incremented with new objects. With this process the

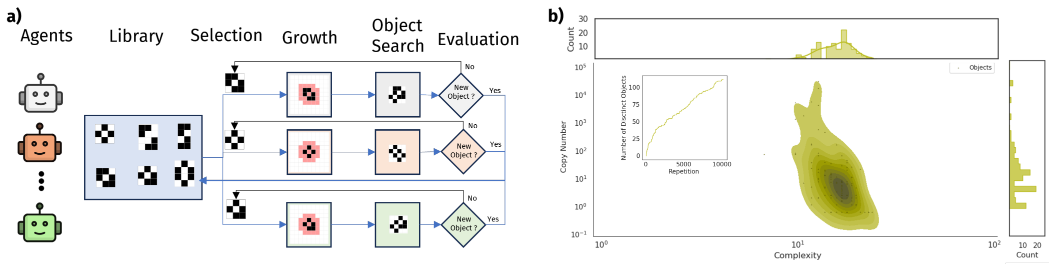

Figure 7: **Distributions of objects found by a group of agents**. **a)** A group of agents search for novel patterns given an starting library. The select a random object, modify it by means of growth and they search for new patterns from the resulting pattern. If a new pattern is found it is added to the library. **b)** The group of agents find more unique objects at a given complexity than a completely undirected process, but they stagnate at a relatively low complexity. This implies that the agents search procedure is not undergoing selection, therefore it is not open-ended process.

agents are able to explore more complex patterns than just looking at random initial conditions since the initial pattern exerts a contingency on what the agents can find, Figure 7b.

More sophisticated algorithmic agents can also be used to generate an increasing set of more complex objects. An LLM-agent [40] with an automatic curriculum and a skill library could be able to explore more patterns than our current approach. In fact our benchmark could be used as a way to characterize the capabilities of different language models assisting the agents. While this is subject of current investigation, the current results showcase one of the simplest agents able to discover more than just random search in a structured way.

5 Conclusions

In this work, a benchmark for open-endedness that analyses the products of a system consisting of agents, resources, and discrete cellular automata was introduced. The benchmark is capable of capturing the intuitive idea of unlimited complexity and novelty of an open-ended system. By analyzing the complexity and the copy number of a set of CA objects generated by our agents, we can characterize the amount of open-endedness of such system. The benchmark is compared with other measures of open-endedness, its complexity measure is compared with entropy, and it's found that the AssemblyCA complexity or memory is able to properly characterize the phenomenon of historical contingency, key for generating complex objects. Finally, three experiments were performed with the aim of showing systems with different degrees of open-endedness.

In the future, we intend to study in more detail some elements of the undirected-directed transition in the CA framework. These include the amount of selectivity in the agent's search, a robust definition of an object in CA, and the analysis of different CA rules and the transition between them. In addition, we plan a deeper study of the cultural behavior of the LifeWiki system. This is highly relevant for the understanding and quantification of open-endedness. Lastly, we plan to study more elaborate agents that can discover CA objects, such as LLM-based CA object explorers. This is highly relevant for a rigorous benchmark of the capabilities of different LLM-agents.

Acknowledgments

The authors would like to thank Yibin Jiang for the idea to generalize the Hash-Life algorithm for grids of arbitrary size.

References

[1] 2008. LifeWiki. `https://conwaylife.com/wiki` Available at `https://conwaylife.com/wiki`.

[2] Alyssa Adams, Hector Zenil, Paul CW Davies, and Sara Imari Walker. 2017. Formal definitions of unbounded evolution and innovation reveal universal mechanisms for open-ended evolution in dynamical systems. *Scientific reports* 7, 1 (2017), 997.

[3] Mark A Bedau, Emile Snyder, and Norman H Packard. 1998. A classification of long-term evolutionary dynamics. In *Artificial life vi: Proceedings of the sixth international conference on artificial life*, Vol. 6. MIT Press, 228.

[4] Randall D Beer. 2014. The cognitive domain of a glider in the game of life. *Artificial life* 20, 2 (2014), 183–206.

[5] Bert Wang-Chak Chan. 2019. Lenia: Biology of Artificial Life. *Complex Systems* 28, 3 (2019).

[6] Bert Wang-Chak Chan. 2023. Towards Large-Scale Simulations of Open-Ended Evolution in Continuous Cellular Automata. *arXiv preprint arXiv:2304.05639* (2023).

[7] Thomas M Cover. 1999. *Elements of information theory*. John Wiley & Sons.

[8] Q Tyrell Davis. 2021. Carle's game: An open-ended challenge in exploratory machine creativity. In *2021 IEEE Conference on Games (CoG)*. IEEE, 01–08.

[9] Sam Earle. 2020. Using fractal neural networks to play simcity 1 and conway's game of life at variable scales. *arXiv preprint arXiv:2002.03896* (2020).

[10] Sam Earle, Julian Togelius, and Lisa B Soros. 2021. Video games as a testbed for open-ended phenomena. In *2021 IEEE Conference on Games (CoG)*. IEEE, 1–9.

[11] Winston Ewert, William Dembski, and Robert J Marks. 2014. Algorithmic specified complexity in the game of life. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45, 4 (2014), 584–594.

[12] Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. 2022. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *Advances in Neural Information Processing Systems* 35 (2022), 18343–18362.

[13] A Flammenkamp. 1994. Natural grown oscillators out of random soup. `https://wwwhomes.uni-bielefeld.de/achim/oscill.html` Available at `https://wwwhomes.uni-bielefeld.de/achim/oscill.html`.

[14] Morgan Giraud. 2022. Leniax: efficient and differentiable Lenia simulator. `http://github.com/morgangiraud/leniax`

[15] R Wm Gosper. 1984. Exploiting regularities in large cellular spaces. *Physica D: Nonlinear Phenomena* 10, 1-2 (1984), 75–80.

[16] Adam P Goucher. 2015. Catagolue. `https://catagolue.hatsya.com/home` Available at `https://catagolue.hatsya.com/home`.

[17] Djordje Grbic, Rasmus Berg Palm, Elias Najarro, Claire Glanois, and Sebastian Risi. 2021. Evocraft: A new challenge for open-endedness. In *Applications of Evolutionary Computation: 24th International Conference, EvoApplications 2021, Held as Part of EvoStar 2021, Virtual Event, April 7–9, 2021, Proceedings 24*. Springer, 325–340.

[18] David Ha and Yujin Tang. 2022. Collective intelligence for deep learning: A survey of recent developments. *Collective Intelligence* 1, 1 (2022), 26339137221114874.

[19] Michael Jirasek, Abhishek Sharma, Jessica R Bame, Nicola Bell, Stuart M Marshall, Cole Mathis, Alasdair Macleod, Geoffrey JT Cooper, Marcel Swart, Rosa Mollfulleda, et al. 2023. Multimodal Techniques for Detecting Alien Life using Assembly Theory and Spectroscopy. *arXiv preprint arXiv:2302.13753* (2023).

[20] Nathaniel Johnston and Dave Greene. 2022. *Conway's Game of Life: Mathematics and Construction*. Nathaniel Johnston.

[21] Miklos Kepes, Nicholas Guttenberg, and L. B. Soros. 2022. ChemGrid: An Open-Ended Benchmark Domain for an Open-Ended Learner. In *2022 IEEE Conference on Games (CoG)*. 221–228. `https://doi.org/10.1109/CoG51982.2022.9893665`

[22] Chris G Langton. 1990. Computation at the edge of chaos: Phase transitions and emergent computation. *Physica D: nonlinear phenomena* 42, 1-3 (1990), 12–37.

[23] Joel Lehman, Jonathan Gordon, Shawn Jain, Kamal Ndousse, Cathy Yeh, and Kenneth O Stanley. 2022. Evolution through large models. *arXiv preprint arXiv:2206.08896* (2022).

[24] Kristian Lindgren and Mats G Nordahl. 1988. Complexity measures and cellular automata. *Complex Systems* 2, 4 (1988), 409–440.

[25] Yu Liu, Cole Mathis, Michał Dariusz Bajczyk, Stuart M Marshall, Liam Wilbraham, and Leroy Cronin. 2021. Exploring and mapping chemical space with molecular assembly trees. *Science advances* 7, 39 (2021), eabj2465.

[26] Stuart M Marshall, Cole Mathis, Emma Carrick, Graham Keenan, Geoffrey JT Cooper, Heather Graham, Matthew Craven, Piotr S Gromski, Douglas G Moore, Sara I Walker, et al. 2021. Identifying molecules as biosignatures with assembly theory and mass spectrometry. *Nature communications* 12, 1 (2021), 3033.

[27] Stuart M Marshall, Douglas G Moore, Alastair RG Murray, Sara I Walker, and Leroy Cronin. 2022. Formalising the pathways to life using assembly spaces. *Entropy* 24, 7 (2022), 884.

[28] Erwan Plantec, Gautier Hamon, Mayalen Etcheverry, Pierre-Yves Oudeyer, Clément Moulin-Frier, and Bert Wang-Chak Chan. 2022. Flow Lenia: Mass conservation for the study of virtual creatures in continuous cellular automata. *arXiv preprint arXiv:2212.07906* (2022).

[29] Ettore Randazzo and Alexander Mordvintsev. 2023. Biomaker CA: a Biome Maker project using Cellular Automata. *arXiv preprint arXiv:2307.09320* (2023).

[30] Sebastian Risi. 2021. The future of artificial intelligence is self-organizing and self-assembling. *Available at: sebastianrisi. com* (2021).

[31] Tomas Rokicki. 2018. Life Algorithms. (2018). `https://www.gathering4gardner.org/g4g13gift/math/RokickiTomas-GiftExchange-LifeAlgorithms-G4G13.pdf` Available at `https://www.gathering4gardner.org/g4g13gift/math/RokickiTomas-GiftExchange-LifeAlgorithms-G4G13.pdf`.

[32] Abhishek Sharma, Dániel Czégel, Michael Lachmann, Christopher P Kempes, Sara I Walker, and Leroy Cronin. 2023. Assembly theory explains and quantifies the emergence of selection and evolution. *Nature* (2023).

[33] Asiiah Song. 2022. A little taxonomy of open-endedness. In *ICLR Workshop on Agent Learning in Open-Endedness*.

[34] Jacob M Springer and Garrett T Kenyon. 2021. It's hard for neural networks to learn the game of life. In *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

[35] Kenneth O Stanley, Joel Lehman, and Lisa Soros. 2017. Open-endedness: The last grand challenge you've never heard of. *While open-endedness could be a force for discovering intelligence, it could also be a component of AI itself* (2017).

[36] Joseph Suarez, Yilun Du, Phillip Isola, and Igor Mordatch. 2019. Neural MMO: A massively multiagent game environment for training and evaluating intelligent agents. *arXiv preprint arXiv:1903.00784* (2019).

[37] Shyam Sudhakaran, Djordje Grbic, Siyan Li, Adam Katona, Elias Najarro, Claire Glanois, and Sebastian Risi. 2021. Growing 3d artefacts and functional machines with neural cellular automata. In *Artificial Life Conference Proceedings 33*, Vol. 2021. MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . , 108.

[38] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

[39] Tim Taylor. 2019. Evolutionary innovations and where to find them: Routes to open-ended evolution in natural and artificial systems. *Artificial life* 25, 2 (2019), 207–224.

[40] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291* (2023).

[41] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. 2023. A survey on large language model based autonomous agents. *arXiv preprint arXiv:2308.11432* (2023).

[42] Stephen Wolfram. 1986. Random sequence generation by cellular automata. *Advances in applied mathematics* 7, 2 (1986), 123–169.

[43] Stephen Wolfram et al. 2002. *A new kind of science*. Vol. 5. Wolfram media Champaign, IL.

[44] Bo Yang. 2023. Self-Replicating Hierarchical Structures Emerge in a Binary Cellular Automaton. *arXiv preprint arXiv:2305.19504* (2023).

[45] Haoqi Yuan, Chi Zhang, Hongcheng Wang, Feiyang Xie, Penglin Cai, Hao Dong, and Zongqing Lu. 2023. Plan4mc: Skill reinforcement learning and planning for open-world minecraft tasks. *arXiv preprint arXiv:2303.16563* (2023).

Appendix

A Glossary

assembly distance captures the common fragments between the assembly spaces of two configurations S_i, and S_j, and allows for a comparison of similarity between them. 4

assembly index is the number of steps from the shortest path to generate an object given a set of elementary operations. 4

assembly space is the union of all shortest pathways to build a set of different configurations $\{S_i\}_{i=1}^{T}$ all at the same time. 4

complexity is quantified as the minimum number of steps that are needed to construct an object given a set of elementary operations. 2, 3

configurations are an assignment of an state to every cell of a CA grid-space. 4

copy number is the number of times a distinct object is found in the library of patterns of different agents. 2, 3

directed process is a process with inflow of building blocks leading to object construction that generates complex objects in high numbers or copies. 3

fragment is an element of the assembly space of an object. 4

hash-assembly represents the approximation of the minimal construction process of a rectangular grid given by a binary-tree or quad-tree representation. 5

historical contingency references the fact that highly complex objects need to have lower complexity sub-parts exist in the system where they are assembled, otherwise they cannot exist. 2

memory is the mean assembly index of a pattern $\{S_i\}_{i=1}^{T}$, by putting more weight on the fragments that are repeated in more configurations. 4

object or **pattern** is defined as a structure that persists over time in a physical substrate. It satisfies the intuitive properties of, finiteness, composability, distinguishability and constrained. For a CA we think about it as the structure that persists over time and whose size is determined by the bounding box at each configuration time-step. 2, 3

pathway is an idealized construction process represented by a directed acyclic graph where each element is a step in the construction and each element expect the building blocks is build by two previously constructed objects. 4

selection its the process of hierarchical constraint of a combinatorial space in order to generate complex objects in high complexity. 3

states are the possible values a CA cell can take, e.g. black or white. 4

undirected process is a process with inflow of building blocks leading to object construction that generates complex objects in low numbers. 3

B Extended Entropy Comparison

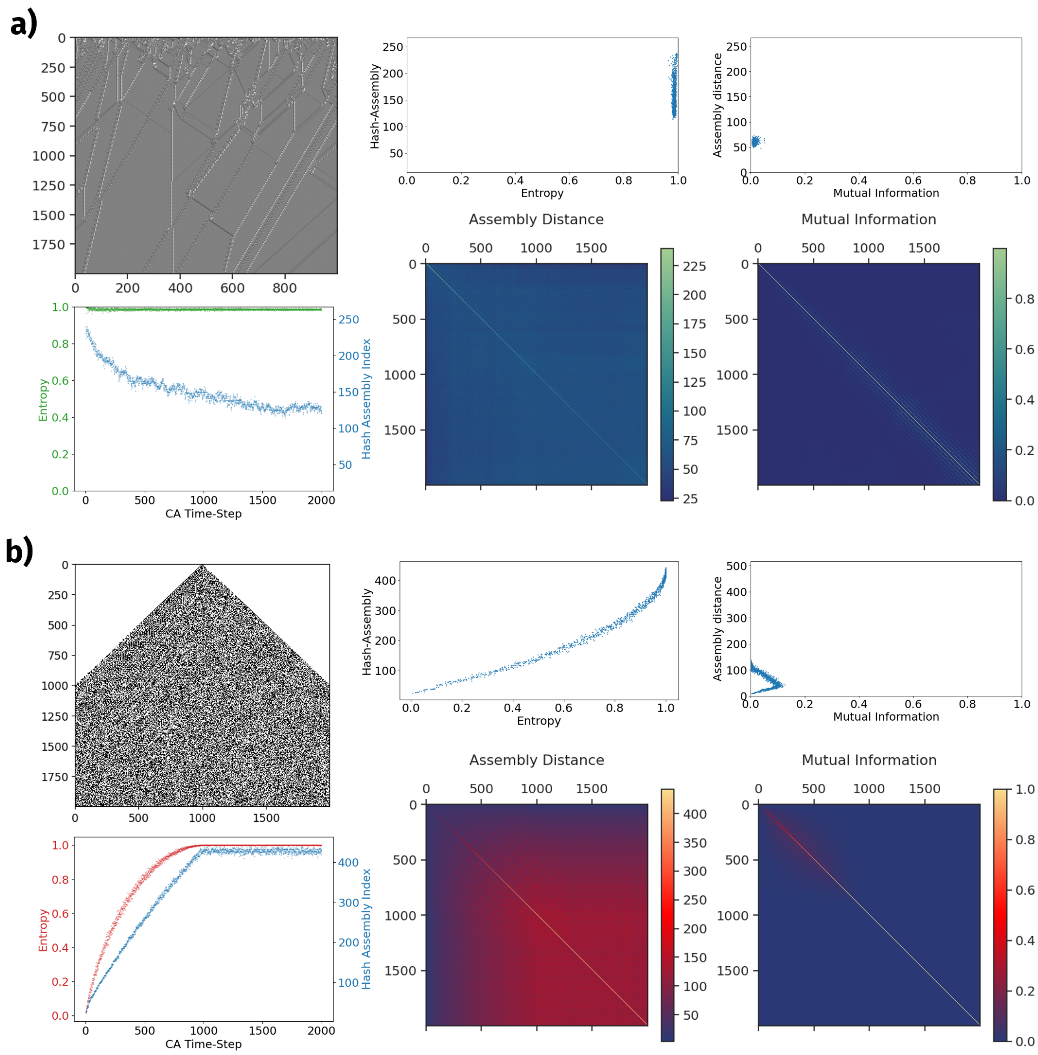

Figure 8: **Comparison between entropy and assembly. a**) Behaviour of a random initial condition in the Rule 110 Cellular Automata. Comparison of the Assembly Index vs Entropy of the time evolution and the Mutual Information and the Assembly Distance between two arbitrary cells and subsequent cells. **b**) The behavior of the Rule 30 Cellular Automata. Comparison of the Assembly Index vs Entropy of the time evolution and the Mutual Information and the Assembly Distance between two arbitrary cells and subsequent cells.

In the first case we have a random configuration running Rule 110, Figure 8, the entropy is maximal all the time, but the assembly index actually goes down as the CA evolves in time. This implies that the CA starts random but, due to the rules the configurations get more and more structured. Again this implies that the assembly index is able to capture spatio-temporal structural information of the evolution of the CA.

In the second case we run the elementary automata Rule 30, Figure 8. This automata generate pseudo-randomness as it output [42]. In this case, both entropy and assembly index increase over time until they arrive to a maximum. Here the advantage of assembly index is not clear, since there is no temporal and spatial structure to the data. This suggests that class III automata that have patterns that exhibit chaotic growth are not amenable to the AssemblyCA benchmark.