Capturing Emerging Complexity in Lenia

Sanyam Jain;
Aarati Shrestha;
Stefano Nichele.

Dept of Computer Science and Communication

Østfold University College & OsloMet University

Table of Contents

1 Complexity and Open-Endedness
- Complexity
- Open-Endedness
- Evolvability

2 Lenia
- Continuous CA
- Lenia Update Rule
- Kernel and Growth Function

3 Emerging Complexity and Behaviour
- EvoLenia: Compression based
- EvoLenia: Variation based
- EvoLenia: AEVoT
- EvoLenia: Rest details

4 Results
- Results VoT (Not a good fitness!)
- Results AE (Not a good fitness!)
- Results AEVoT (Better than both)
- Results AEVoT (for known Kernel)

5 Challenges and Future Improvements
- Challenges, Learnings and Conclusion
- Future Scope of Improvement

6 References and Offline material

WIVACE 2023 Venice | 6-8 September   OSLOMET   Høgskolen i Østfold

Complexity and Open-Endedness

Complexification

- End Goal: How to solve a task?[Clune, 2019];
- Where Manual AI fails? [Stanley et al., 2017];
- Novelty, Task based and Hybrid approaches [Stanley et al., 2017];
- Open-endedness and Complexification goes together [Randazzo and Mordvintsev, 2023];

Why open-endedness?

- Endless variation implies complexification;
- Fundamental questions: initial conditions, selection pressure, etc;
- Endless variation + Sensitivity to initial conditions + Selection pressure = Complexification [Randazzo and Mordvintsev, 2023]
  OR
  Open-endedness allows AI systems to continue to learn and improve over time, adapting to changing environments and evolving to meet new challenges

Why Evolvability? [Randazzo and Mordvintsev, 2023]

- Heritable Genetics and Selectable Phenotype with variation;
- Without Evolvability there would be no discovery, no new behaviour;
- Dynamical task landscapes, adaptive mutations, novelty search;

Lenia [Chan, 2018]

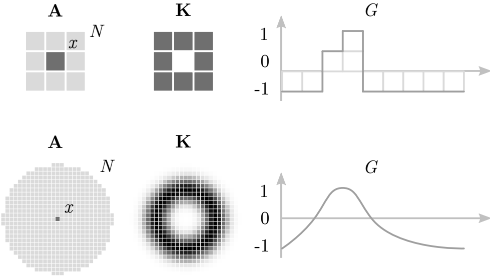

Figure: Discrete CA vs Continuous CA

Lenia Update Rule

Update function

The Lenia update rule is given by:

$$A_{t+1} = [A_t + \Delta t\, G(K * A_t)]$$

- A_t: Current State at t
- Δt: Step size
- G: Growth Function (for eg. Gaussian)
- K: Neighborhood Kernel
- $*$: Convolution operation

Kernel and Growth Function

Kernel Function

- Weighted importance to neighboring pixels and gradually reduced importance to distant pixels.
- Calculate the distances of each coordinate from the center and apply a mask to filter out values outside the desired radius.

Growth Function

- By adjusting the μ and σ parameters of the G, each cell's growth or decay can be controlled by taking input as Nh. sum array.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Emerging Complexity and Behaviour in Lenia using standard Genetic Algorithm

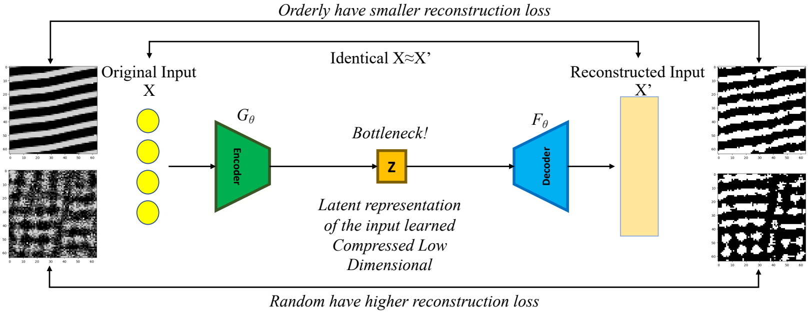

Figure: Compression based: AutoEncoder [Cisneros et al., 2019]

Fitness VoT

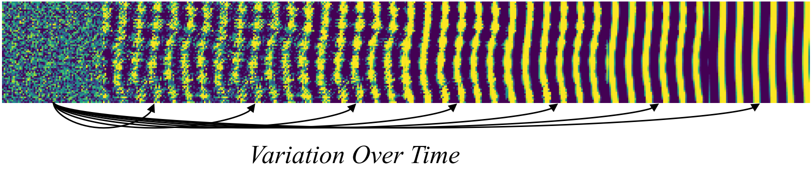

Figure: Variation based: Variation over Time

Algorithm 3. Auto-Encoder based Variation over Time (AEVoT)

Input : Input frames of Lenia patterns

Output: Population standard deviation (pstd) over list of alive cells count of each frame

1 **Begin**

2 Reconstruct the original frames using an auto-encoder (AE):; **For** *each input frame* f **do**

3 $f_{AE} = \mathrm{AE}(f)$; Calculate the number of alive cells in the reconstructed frame f_{AE} using a threshold:; $alive_f = count(p \geq threshold)$; Store the number of alive cells $alivef$ in a list;

4 Calculate the population standard deviation (pstd) of the list of alive cells counts:; $mean = \frac{1}{n} \sum i = 1^n alivei$; $variance = \frac{1}{n} \sum i = 1^n (alive_i - mean)^2$; $pstd = \sqrt{variance}$; **return** *Population standard deviation (pstd) over list of alive cells count of each frame*;

Figure: AEVoT Based: Combined Approach

Selection, Crossover, and Mutation

- Roulette Wheel Selection
- No Crossover
- Mutation by perturbation

Results from different experiments for AE, VoT and AEVoT

VoT based Experiments

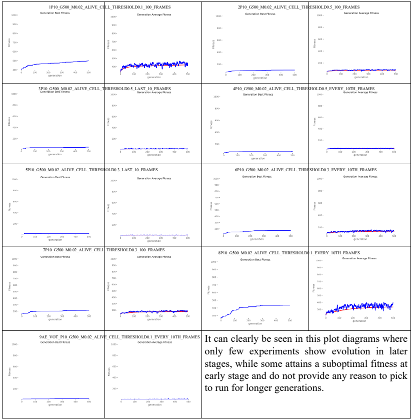

Figure: VoT experiments

AE based Experiments

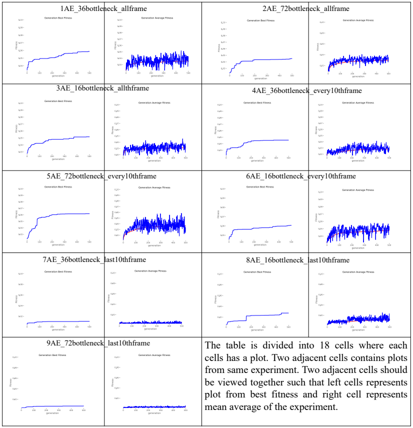

Figure: AE experiments

AEVoT based Experiments

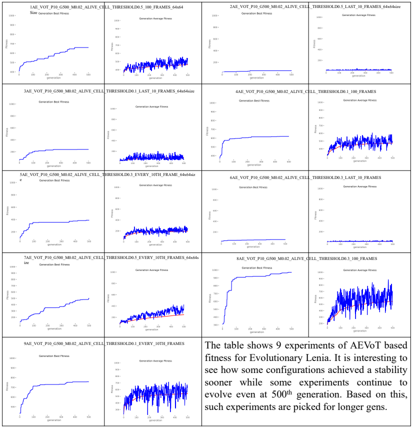

Figure: AEVoT experiments

WIVACE 2023 Venice | 6-8 September OSLOMET Høgskolen i Østfold

AEVoT based Experiments: Known Kernel

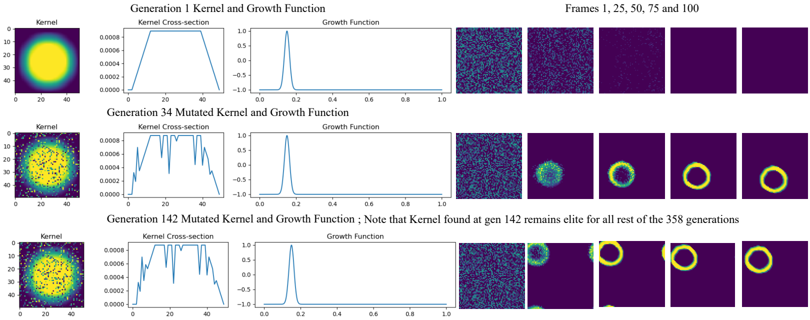

Figure: AEVoT experiment for known kernel: Adaptable mutations

Challenges, Learnings and Conclusion

- Automated Discovery, Emergent Agency, Open-Endedness [Chan, 2018];
- A strong hypothesis and supporting proofs for chosen parameters, yields good results (We discovered a ring forming bacteria);
- Evolution is time-taking!

Future Scope of Improvement

- Mutating Known Kernels.
- Particle Lenia, Flow Lenia, Sensorimotor Lenia
- Using JAX.

Major References

Chan, B. W.-C. (2018).
Lenia-biology of artificial life.
arXiv preprint arXiv:1812.05433.

Cisneros, H., Sivic, J., and Mikolov, T. (2019).
Evolving structures in complex systems.
In *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 230–237. IEEE.

Clune, J. (2019).
Ai-gas: Ai-generating algorithms, an alternate paradigm for producing general artificial intelligence.
arXiv preprint arXiv:1905.10985.

Randazzo, E. and Mordvintsev, A. (2023).
Biomaker ca: a biome maker project using cellular automata.
arXiv preprint arXiv:2307.09320.

Stanley, K. O., Lehman, J., and Soros, L. (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.

Play with it:

https://s4nyam.github.io/evolenia/