GOAL-GUIDED NEURAL CELLULAR AUTOMATA: LEARNING TO CONTROL SELF-ORGANISING SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Inspired by cellular growth and self-organization, Neural Cellular Automata (NCAs) have been capable of "growing" artificial cells into images, 3D structures, and even functional machines. NCAs are flexible and robust computational systems but – similarly to many other self-organizing systems — inherently uncontrollable during and after their growth process. We present an approach to control these type of systems called *Goal-Guided Neural Cellular Automata* (GoalNCA), which leverages goal encodings to control cell behavior dynamically at every step of cellular growth. This approach enables the NCA to continually change behavior, and in some cases, generalize its behavior to unseen scenarios. We also demonstrate the robustness of the NCA with its ability to preserve task performance, even when only a portion of cells receive goal information.

1 INTRODUCTION

Cellular growth and morphogenesis has resulted in an incredibly diverse collection of multi-cellular organisms on Earth. A single cell starts off a chain of replication and local interactions that eventually result in a fully grown organism. Some of these organisms have learned to control and manipulate characteristics of their cells dynamically, resulting in diverse behaviors, such as the ability to change skin color, texture, and even shape (Figure 1). Inspired by the flexibility and adaptability of these biological systems, we present *Goal-Guided Neural Cellular Automata* (GoalNCA), an artificial self-organizing system that is able to not only grow artificial cells, but also guide their behaviors dynamically.

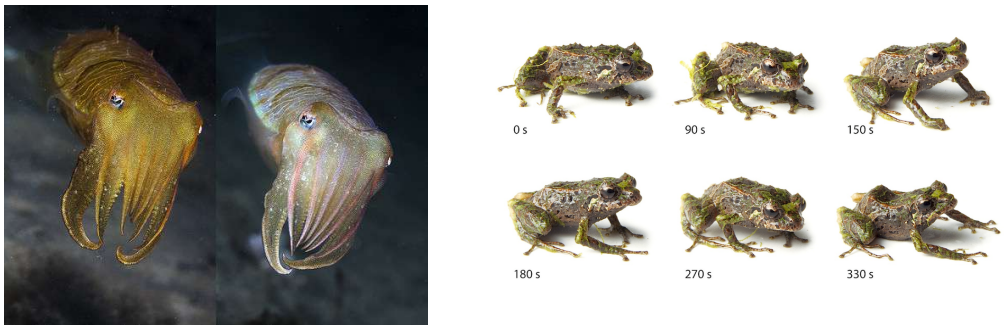

Figure 1: Left: Cuttlefish that can change its color of its cells to blend in with its environment. Right: *Pristimantis mutabilis*, or mutable rainfrog, can change its skin texture from smooth to spiky. Images from Gentle (2015).

1.1 COMPUTATIONAL MODEL: NEURAL CELLULAR AUTOMATA

Cellular Automata (CAs), inspired by biological cellular growth and self-organization, are systems that apply local rules repeatedly on artificial cells that are typically represented as a 1 or a 0 in a grid. Conway's Game of Life, a simple CA system with only 4 discrete local rules, is Turing

Complete, able to represent many complex behaviors. Neural Cellular Automata (NCAs) attempt to learn the local rules using compact neural networks and cell states, guided by an objective function (Mordvintsev et al., 2020; Nichele et al., 2017). Each cell, now represented as a vector, is composed of target channels, a living channel that determines whether a cell should be alive or dead, and hidden channels which serve as a sort of cell memory, similar to the hidden state in an RNN (Sherstinsky, 2018). This formulation has been shown to be effective at learning a variety of tasks, such as growing self-classifying 2D images, game levels, and regenerative functional 3D machines (Randazzo et al., 2020; Earle et al., 2022; Sudhakaran et al., 2021).

Typically, an NCA starts with a small set of initial "live" cells, or "seeds", that eventually grow into a full artificial organism over time. This seed can be a static vector (with the hidden channels being set to 1), or they can be learned in order to grow a diverse set of objects from a single NCA network (Palm et al., 2022; Frans, 2021)). However, controlling a self-organizing system composed of many independent components can be challenging. As Kelly (2009) writes:

"Guiding a swarm system can only be done as a shepherd would drive a herd: by applying force at crucial leverage points, and by subverting the natural tendencies of the system to new ends.".

For example, NCAs can lack control over cell behaviour because the system's stability and behaviors rely on repeated operations on initial seeds. These cells are not trained to have dynamic behaviors during and after growth, so naively nudging a cell in a direction can cause a chain reaction that results in the collapse of the system. Our proposed variation, GoalNCA, uses encoded goal vectors to perturb and guide these cells throughout growth, giving the ability to change cell behaviors dynamically. The insights gained on NCAs (e.g. the system can be guided also if only a portion of the cells receive the goal information) could be useful to making self-organizing systems in general more controllable and open interesting future research directions.

2 Related Work: Guiding Self-Organizing Systems

Recently, ideas from collective intelligence and self-organizing systems have started to capture the interest of the larger machine learning community (Risi, 2021; Ha & Tang, 2021). However, self-organizing systems in general have shown to be inherently difficult to control Kelly (2009); Gershenson (2007). In this section we review the two most related approaches to the one introduced in this paper, which both deal with the control of self-organization in cellular automata.

The first one is **Lenia**, an extension of classical CA, moving a discrete grid-like space to a continuous one (Chan, 2018). A computational system classified as artificial life —or virtual creature—, Lenia exhibits a series of remarkable properties reminiscent of life forms. Because of this, it constitutes an interesting abstract model to study the emergence of complex morphogenetic patterns *in-silico*. Lenia has recently been extended for goal guided locomotion (Hamon et al., 2022), using NCAs and curriculum learning with sampled goals to grow creatures that can traverse through obstacles (Forestier et al., 2020). Our approach is similar in that we also grow creatures that are able to perform a variety of goals, but we explicitly incorporate goal encodings directly into the cells, and perturb the cell states to change behaviors dynamically.

The second one is **Adversarial Perturbations in Neural Cellular Automata**. The idea of changing and guiding behaviors of NCA systems has been explored before in Randazzo et al. (2021), where different methods of adversarial attacks on NCA states were evaluated. One particularly effective method was multiplying trainable symmetric perturbation matrices with NCA cell states. The authors showed that this method can change the cells behaviors, guided by a particular target (such as changing the colors of the cells). To train these perturbation matrices, the authors first freeze a pretrained NCA's weights, and then "fine tune" the perturbation matrices towards goals. This setup has the advantage of training smaller parameters efficiently and re-using the existing trained NCA. In our work, instead of freezing the NCA weights and training perturbation matrices, we train a goal encoder concurrently with the NCA in order to change cell behavior throughout the growth process. We also opted for a simpler, more computationally efficient approach, where we instead add goal encoded perturbation vectors.

3 GOAL-GUIDED NEURAL CELLULAR AUTOMATATA

Our system and network architecture is similar to that presented in Mordvintsev et al. (2020). The main contribution of the presented approach is a goal encoder which adds updates to every living cell’s hidden states at each step in the growth cycle. Surprisingly, this simple addition is effective in guiding cell behavior towards tasks such as morphing images and learning locomotion dynamics.

3.1 MODEL DETAILS AND TRAINING PROCEDURE

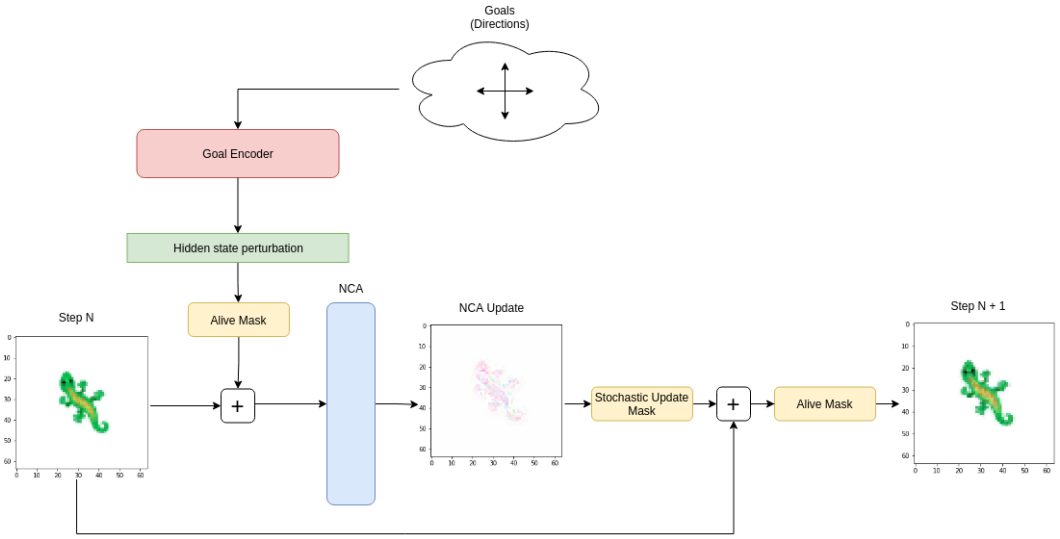

Figure 2: Single NCA update step in the controllable locomotion experiment. A one-hot encoded direction is passed into a goal encoder to compute hidden state perturbations for live cells, which get added to previous cell states to compute the next NCA state

Our GoalNCA model consists of two major components: A simple 3 layer MLP goal encoder and a typical NCA, similar to the one proposed in Mordvintsev et al. (2020). Our NCA consists of a learnable 3×3 single convolution, followed by 3 1×1 convolutional blocks. A single update step of our NCA is described in Figure 2. We first take the cell states at step n, perturb all live cells’ hidden states by a encoded goal vector, and calculate updates via forward pass through our convolutional layers. Before adding each update, we multiply it by a boolean stochastic mask, where each cell has a probability of being updated (which is referred to as $cell_fire_rate$). We then multiply an “alive mask”, which utilizes the cells’ living channels (3rd channel in the cell vector), where if all neighborhood cells have values < 0.1, the cell is considered dead and multiplied by 0.0.

Our training procedure, detailed in Algorithm 1, is also similar to that in Mordvintsev et al. (2020), with the addition of goal sampling. At the start of training, we populate the sample pool with initial seeds (single alive cell in the center with ones for its hidden values and living channel), which acts as a replay memory to ensure that the NCA is able to preserve the stability of its cells over many iterations. At every training step, we uniformly sample and encode goals and NCA states from the sample pool, replacing 2 of the states with an initial seed. The NCA states are updated for a random number of steps in the range $[min_steps, max_steps]$, guided by the goal encodings. Next, we update the NCA and goal encoder via MSE loss between the new NCA states’ RGBA channels and the corresponding sampled goals’ target images. Finally, we replace the older NCA states in the sample pool with the new NCA states.

Algorithm 1 : GoalNCA

initialize sample pool S with N initial seeds
initial collection of goals G
initialize NCA f_θ with parameters θ
initialize goal encoder e_ϕ with parameters ϕ
initialize min_steps, max_steps
for training iterations $t = 1...T$ **do**
  sample NCA states from sample pool b_t
  sample goals and targets g_t, $target_t$
  encode $e_t = e_\phi(g_t)$
  $o_t = b_t$
  for $num_steps \sim [min_steps, max_steps]$ **do**
    $o_t = f_\theta(o_t, e_t)$
  end for
  $\theta \leftarrow \theta$ - $\nabla_\theta L(o_t, target_t)$
  $\phi \leftarrow \phi$ - $\nabla_\phi L(o_t, target_t)$
  replace b_t with o_t in S
end for

4 Experiments

For all experiments, we use Adam Optimizer (Kingma & Ba, 2017) with 1e-3 learning rate. We also normalize the gradients and clip NCA states between [-10, 10]. We use 64 channels for our 1×1 convolutional layers and for our 3×3 convolutional layer we have $num_cell_channels * 3$ output channels. We also trained for 100k training iterations on an NVIDIA 2080ti GPU. Additional hyperparameter details shown in Table 1. We plan to release our code publicly on Github with instructions on how to reproduce our results.

Experiment Hyperparameters							
Experiment	No. NCA params	Goal Type	Encoder	No. Hidden Cell States	Cell Fire Rate	Batch Size	Pool Size
Morphing Emoji, Image Encoder	10,167	one-hot vector, size 10	Linear, 1888 params	16	0.5	8	1024
Controllable Locomotion	10,167	one-hot vector, size 5	Linear, 160 params	32	1.0	24	256

Table 1: Hyperparameter Details

4.1 Morphing Emojis

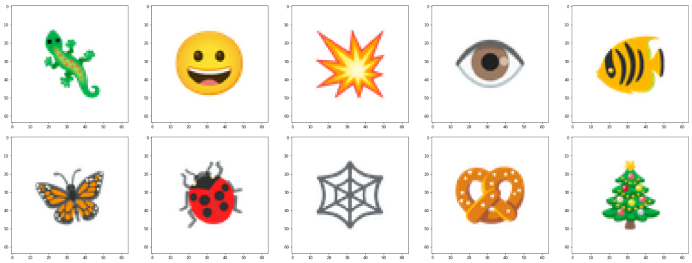

Figure 3: Emoji dataset used in the Morphing Emojis experiment. Each image is 4×64×64 (RGBA channels) and is scaled between [0,1].

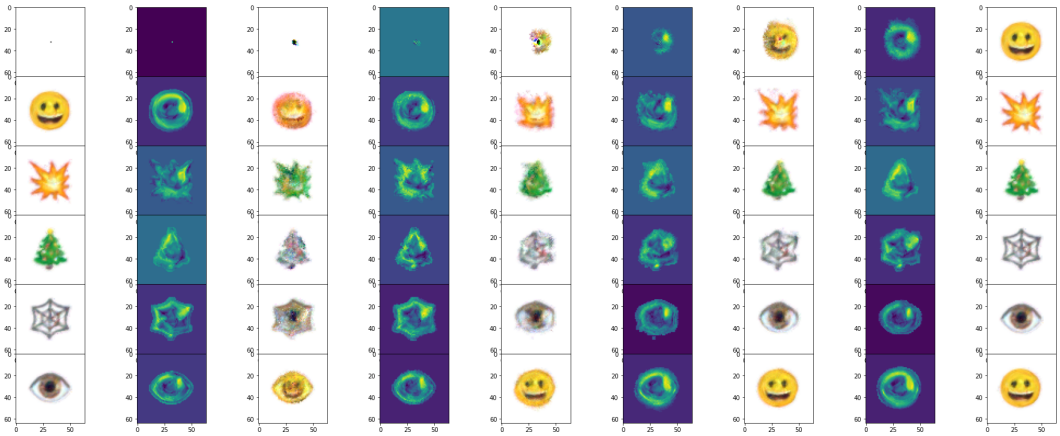

Figure 4: Visualization of emoji morphing and hidden states over time. Emojis in column 1 are the previous rows’ final grown emojis. Update steps are in intervals [8, 16, 16, 64], guided by a goal encoding.

Inspired by organisms that can change their shape and colors, we aim to learn a GoalNCA that is able to manipulate cells to morph into different emojis, shown in Figure 3.

For our goal encoder, we chose a simple 3 layer MLP that takes in a one-hot encoded vector, corresponding to an emoji. We also experimented with a convolutional goal encoder that takes in the actual emoji, which resulted in more detailed grown emojis. However, this also introduced over 100k additional parameters.

We train using $min_steps = 96$ and $max_steps = 192$, but because of GPU limitations, we instead grow samples for a random number of steps between $[48, 96]$ twice and perform NCA updates for each. We also set “cell fire rate” to 0.5, denoting that a cell has a 50% chance of receiving an update at every growth step. We found that this stabilizes training and adds robustness to the system.

The GoalNCA learns to grow cells that are able to transform from any emoji in the dataset to another while preserving the system, essentially learning a stable mapping between emojis. The system can also interpolate between emojis, revealing a mixture of characteristics when cells are morphing between them (Figure 4).

4.2 Controllable Locomotion

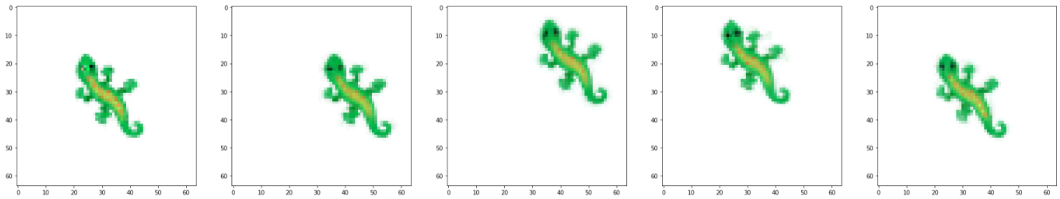

Figure 5: Discrete Movements controlled by NCA from left to right: (1) Salamander grown for 96 steps; (2) salamander moving right for 96 steps; (3) salamander moving up for 96 steps; (4) salamander moving left for 96 steps; (5) salamander moving back down for 96 steps.

Cellular Automata systems, such as Lenia (Hamon et al., 2022; Chan, 2018) and Game of Life (Conway et al., 1970), can create diverse artificial creatures that can move while also keeping consistent shape. In this experiment, we aim to extend this ability by learning a system that can not only grow and move artificial creatures, but control and learn general locomotion dynamics.

We choose the salamander emoji as our target and use 5 different directions (stay in place, up, down, left, right) as our goals. We represent the directions as one-hot encoded vectors following a linear goal encoder. In our implementation, we use the Pytorch Embeddings module (Paszke et al. (2019))

for simplicity. The target for each direction is the salamander shifted by $num_steps/8$ number of pixels in the particular direction, meaning that 8 NCA growth steps should result a shift of one pixel. In order to achieve smooth movement, at every training step we sample random directions and use a series of 4 targets that are gradually farther away. This allows for the NCA to learn natural movement while also being able to retain the salamander’s shape (see Figure 5).

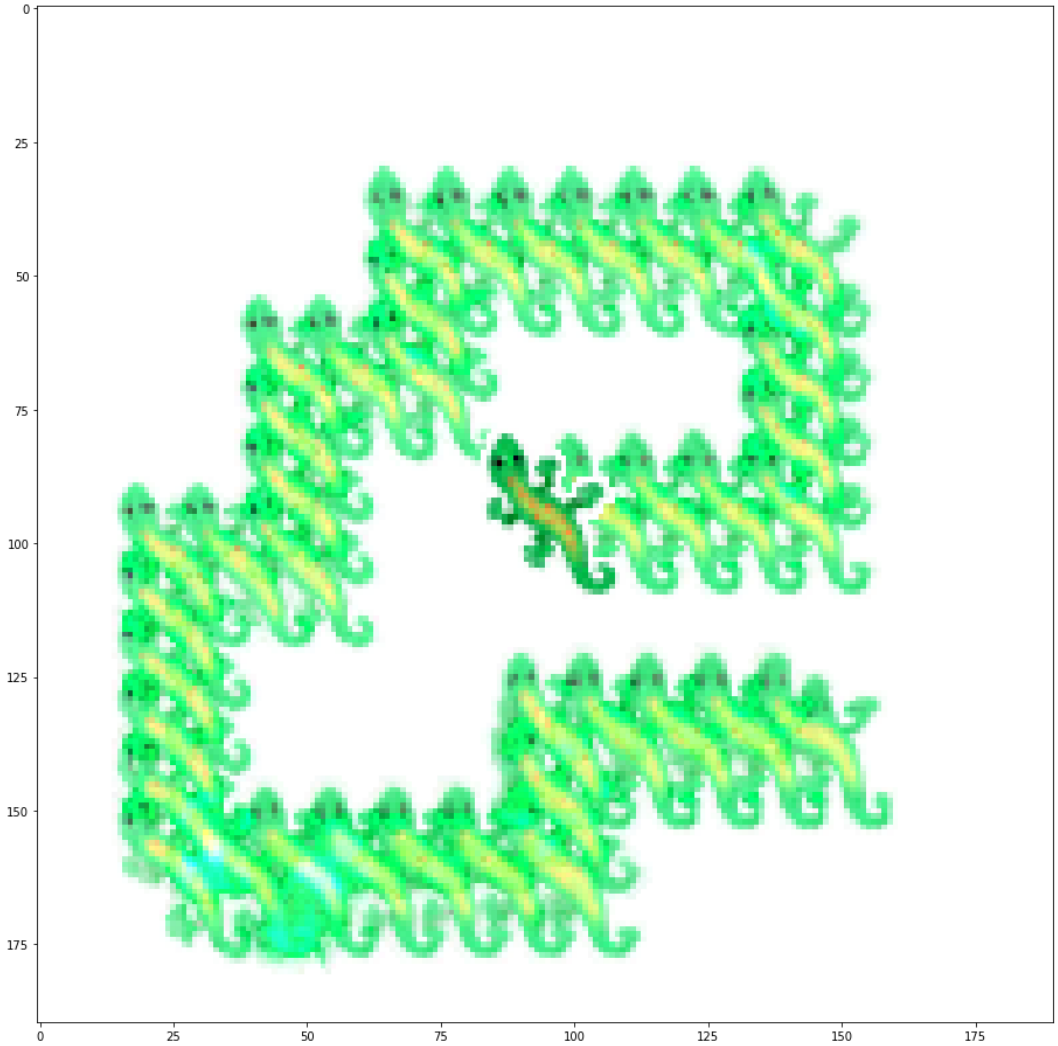

Figure 6: Example unseen trajectory in a larger grid than what was used in training (this one is 192×192 while during training the grid was only 64×64). Our NCA grows a salamander in the center, and moves in a trajectory that has not been seen during training, following human input. The cells are able to constantly switch directions while keeping their consistency for thousands of steps.

The GoalNCA learns the fundamentals of movement and is able to guide the cells to move freely in diverse trajectories that had not been observed during training (Figure 6). We observe that it is able to do this remarkably well, as it’s able to move in an erratic fashion for 1000s of steps without collapsing the system. Also, because NCAs operate solely on local information, we can expand the grid size infinitely and thus discover an infinite number of trajectories.

4.3 Robustness to Goal Dropout

Self-organizing systems, leveraging local communication, are often inherently robust to adversarial modifications, such as observation dropout (Tang & Ha, 2021) and damage (Mordvintsev et al., 2020; Palm et al., 2022). We find that in our controllable locomotion experiment, in which all cells

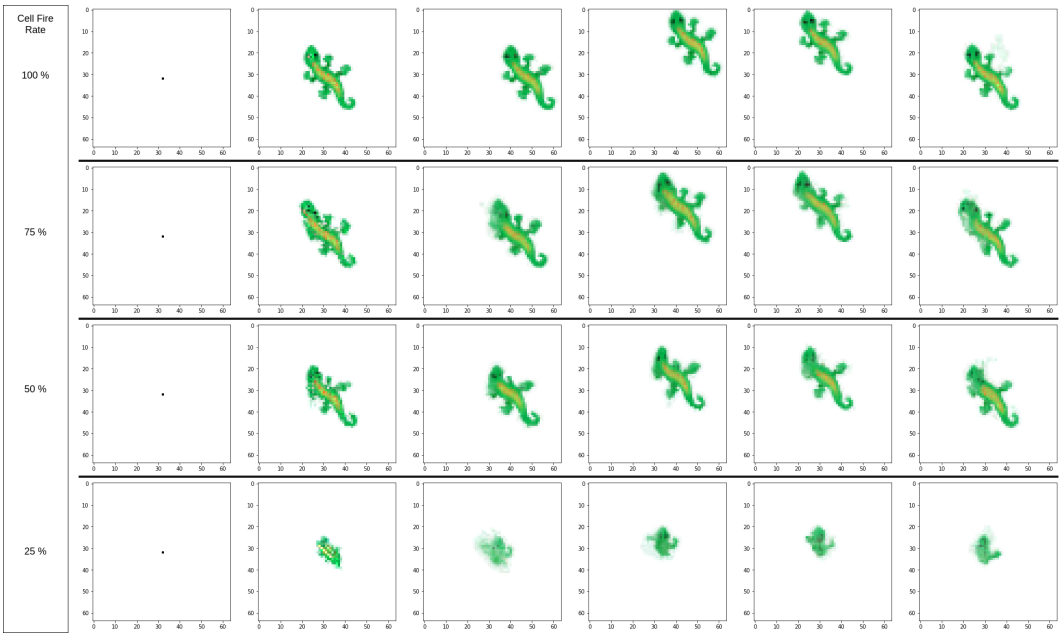

Figure 7: Movements compared across different cell fire rates. Note that cell fire rate during training is set to 100%. Columns: (1) initial seed with only one live cell in the center. (2) Grown for 96 steps, guided by center goal embedding (goal is to grow and keep the salamander stationary). (3) Guided by right direction goal for 128 steps, and then guided by center goal for 96 steps. (4) Guided by up direction goal for 128 steps, and then guided by center goal for 96 steps. (5) Guided by left direction goal for 128 steps, and then guided by center goal for 96 steps. (6) Guided by down direction goal for 128 steps, and then guided by center goal for 96 steps. At this point the salamander should be back in the center of the grid.

received updates during training (cell fire rate = 1.0), the system is robust even if only a portion of cells receive an update during evaluation. In the controllable locomotion experiment, we found that if cells only have a 75% chance of receiving updates per timestep, the grown salamander is still able to move and retain its rough shape. At 25% - 50%, even though the salamander shape isn't completely preserved, the cells still manage to move slightly in the goal directions. This strengthens the case for developing dynamic behaviors in self-organizing systems, which can enable them to thrive in imperfect environments with missing information.

5 Discussion and Future work

NCAs and self-organizing systems are incredibly powerful and can be robust to damage, but they lack flexibility and control over their cells after the growth process is complete. Here we show that by simply adding goal encoded perturbations to hidden cell states, our GoalNCA approach is able to guide cells towards diverse and robust behaviors throughout the growth cycle. We also notice that, as a byproduct of the training procedure, the GoalNCA ends up learning a goal embedding space, able to group similar goals closer together and keep others farther apart (Figure 8).

The GoalNCA was able to effectively guide cells towards desired behaviors while only requiring an additional small linear goal encoder ($<$ 2k params) for both experiments. In the morphing image experiment, we also tested a convolutional encoder with over 100k params, which was able to grow and morph sharper images with the tradeoff of a large increase in parameters. One limitation with the goal encoders presented is that they are not generative models, in that we cannot learn a goal distribution. This approach can be incredibly powerful, as we would be able to sample goals and diverse behaviors. This would be similar to the Variational Neural Cellular Automata (VNCA) introduced in Palm et al. (2022), which samples seed states in order to grow diverse images.

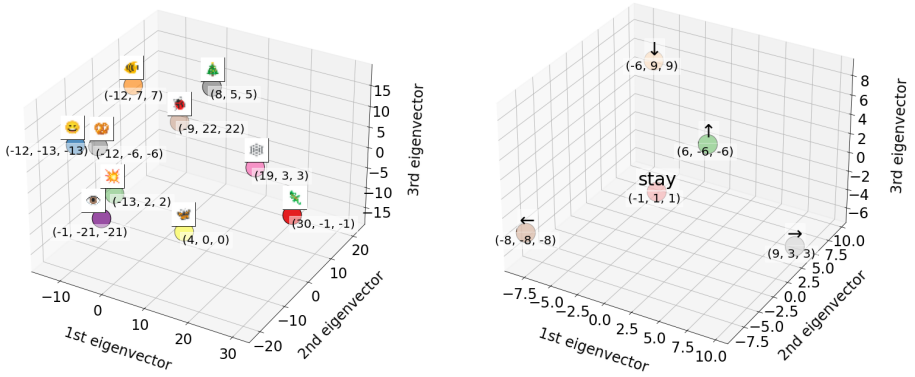

Figure 8: Left: PCA embedded morphing image encodings. Similar emojis are grouped together (smiley and pretzel), while those with distinct shapes are farther away from the rest (spider web, salamander). Right: PCA embedded directions encodings. Each direction has distinct behaviors associated with it, so they are far apart from eachother.

Our goal sampling method, essential in order for the GoalNCA to learn behaviors associated with goals, is naive and can be optimized. At every training step, we sample goals uniformly with replacement. Because we used a small goal space, our system was exposed to enough samples of each goal during training. However, in cases of large goal spaces, it is not guaranteed that goals will be observed equally and some may even be overlooked entirely. In the future, we hope to improve our goal sampling methods with ideas from curriculum learning. One possibility could be Intrinsically Motivated Goal Exploration Processes (IMGEP) (Forestier et al., 2020), where goals are dynamically chosen based on an exploration / exploitation procedure, allowing for efficient exploration and learning with large goal spaces.

In the future, we plan to extend our approach to more complex tasks that require incorporating dynamic feedback in self-organizing systems. We are particularly interested in guiding cells that actually react to their environment by leveraging goals that represent environmental states. This could prove useful when designing self-organizing controllers, which need to be able to incorporate environment information.

REFERENCES

Bert Wang-Chak Chan. Lenia - Biology of Artificial Life. *arXiv*, Dec 2018. URL https://arxiv.org/abs/1812.05433v3.

John Conway et al. The game of life. *Scientific American*, 223(4):4, 1970.

Sam Earle, Justin Snider, Matthew C. Fontaine, Stefanos Nikolaidis, and Julian Togelius. Illuminating diverse neural cellular automata for level generation, 2022.

Sébastien Forestier, Rémy Portelas, Yoan Mollard, and Pierre-Yves Oudeyer. Intrinsically motivated goal exploration processes with automatic curriculum learning, 2020.

Kevin Frans. Stampca: Growing emoji with conditional neural cellular automata. *kvfrans.com*, 2021. URL http://kvfrans.com/stampca-conditional-neural-cellular-automata/.

Louise Gentle. Some shape-shifting animals that can morph to fool others, 2015. URL https://theconversation.com/some-shape-shifting-animals-that-can-morph-to-fool-others-39616.

Carlos Gershenson. *Design and control of self-organizing systems*. CopIt Arxives, 2007.

David Ha and Yujin Tang. Collective intelligence for deep learning: A survey of recent developments. *arXiv preprint arXiv:2111.14377*, 2021.

Gautier Hamon, Mayalen Etcheverry, Bert Wang-Chak Chan, Clément Moulin-Frier, and Pierre-Yves Oudeyer. Learning sensorimotor agency in cellular automata, January 2022. URL https://hal.inria.fr/hal-03519319.

Kevin Kelly. *Out of control: The new biology of machines, social systems, and the economic world*. Hachette UK, 2009.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

Alexander Mordvintsev, Ettore Randazzo, Eyvind Niklasson, and Michael Levin. Growing neural cellular automata. *Distill*, 2020. doi: 10.23915/distill.00023. https://distill.pub/2020/growing-ca.

Stefano Nichele, Mathias Berild Ose, Sebastian Risi, and Gunnar Tufte. Ca-neat: evolved compositional pattern producing networks for cellular automata morphogenesis and replication. *IEEE Transactions on Cognitive and Developmental Systems*, 10(3):687–700, 2017.

Rasmus Berg Palm, Miguel González Duque, Shyam Sudhakaran, and Sebastian Risi. Variational neural cellular automata. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=7fFO4cMBx_9.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.

Ettore Randazzo, Alexander Mordvintsev, Eyvind Niklasson, Michael Levin, and Sam Greydanus. Self-classifying mnist digits. *Distill*, 2020. doi: 10.23915/distill.00027.002. https://distill.pub/2020/selforg/mnist.

Ettore Randazzo, Alexander Mordvintsev, Eyvind Niklasson, and Michael Levin. Adversarial reprogramming of neural cellular automata. *Distill*, 2021. doi: 10.23915/distill.00027.004. https://distill.pub/selforg/2021/adversarial.

Sebastian Risi. The future of artificial intelligence is self-organizing and self-assembling. *sebastianrisi.com*, 2021. URL https://sebastianrisi.com/self_assembling_ai.

Alex Sherstinsky. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *CoRR*, abs/1808.03314, 2018. URL http://arxiv.org/abs/1808.03314.

Shyam Sudhakaran, Djordje Grbic, Siyan Li, Adam Katona, Elias Najarro, Claire Glanois, and Sebastian Risi. Growing 3d artefacts and functional machines with neural cellular automata. *CoRR*, abs/2103.08737, 2021. URL https://arxiv.org/abs/2103.08737.

Yujin Tang and David Ha. The sensory neuron as a transformer: Permutation-invariant neural networks for reinforcement learning, 2021.