

Adversarial Environment Evolution for Robust Pathfinding Neural Cellular Automata

Sam Earle and Julian Togelius
New York University, USA
sam.earle@nyu.edu

Abstract

Pathfinding makes up an important sub-component of a broad range of complex tasks in AI, such as robot path planning, transport routing, and game playing. While classical algorithms can efficiently compute shortest paths, neural networks with learned weights could be better suited to adapting these sub-routines to more complex and intractable tasks. Neural Cellular Automata (NCAs) are a particularly promising candidate for learning to solve pathfinding problems given that their purely convolutional architecture amounts to a series of distributed, recursive computations over a grid. This kind of computation is aligned with classical approaches to pathfinding such as Dijkstra’s algorithm. NCAs can also be seen as modeling morphogenetic processes, and by this token exhibit robustness against spatially local perturbations and can scale naturally to variable grid-sizes. We train NCAs to solve pathfinding problems using supervised learning and introduce an adversarial training scheme in which data points are mutated during training in order to bias the dataset toward more challenging problems. We find that this approach leads to pathfinding NCAs capable of increased generalization on larger grid-sizes. Our work opens the door to scaling NCAs to more complex problems by automatically evolving curricula during training.¹

Motivation

Pathfinding is a crucial sub-routine in many important applications. On a 2D grid, the shortest path problem is useful for robot path planning [Wang et al., 2011] or in transportation routing [Fu et al., 2006]. Long paths (and quantities such as the diameter) are relevant for estimating photovoltaic properties of procedurally-generated microstructures for solar panels [Stenzel et al., 2016, Lee et al., 2021], or the complexity of grid-based video game levels [Khalifa et al., 2020].

Classical algorithms to solve pathfinding and related problems include the Bellman-Ford algorithm [Bellman, 1958, Ford Jr, 1956] for finding the shortest path from a single source node to other nodes, and Breadth-First Search (BFS) [Moore, 1959, Merrill et al., 2012], which models

the connected nodes using a Dijkstra map [Dijkstra et al., 1959]. Often, these algorithms involve “flooding” values across space (in a potentially distributed manner) in order to measure distance and connectivity.

Neural networks are increasingly used to solve complex problems involving pathfinding subroutines. Therefore, modeling classical pathfinding algorithms in “neurally plausible” ways could be advantageous for holistically solving these more complex problems.

Meanwhile, Neural Cellular Automata (NCAs), which involve recursive, distributed computations over a grid (or graph) structure, have gained a cult following among the Artificial Life community thanks to their simplicity (often involving a relatively minuscule number of learned parameters) and the intuitive, interpretable way in which they model the morphogenesis or “growth” of artifacts across space and time. Such artifacts include images [Mordvintsev et al., 2020], textures [Mordvintsev et al., 2021], warehouses [Zhang et al., 2023], video game levels [Earle et al., 2021], player actions [Variengien et al., 2021] and even neural network controllers [Najarro et al., 2022].

The distributed nature of NCAs make them a natural fit for pathfinding problems. We use such problems as a testbed, and show that NCAs can indeed be trained to solve such problems with high degrees of accuracy. More generally, we ask how we might further improve the robustness of NCAs when solving problems over grids. We propose adversarially evolving environments (here, binary mazes), treating the NCA’s loss on a given environment as the environment’s fitness. We show that this approach leads to substantially improved performance on held-out test environments including larger grid-sizes, suggesting that online adversarial environment evolution pushes NCAs to learn more robust policies.

Pathfinding NCAs

Our implementation of Neural Cellular Automata (NCAs) follows that of Mordvintsev et al. [2021]. The NCA consists of one convolutional layer with a 3×3 kernel and padding of 1, followed by a ReLU activation. In our dataset, mazes are represented as 2D one-hot arrays (with separate chan-

¹Code is available at <https://github.com/smearle/pathfinding-nca>

task	env. evolution	n. hid chan	train			test		
			n. params	16x16		16x16		32x32
				sol. length	pct. complete	accuracies	pct. complete	accuracies
Shortest path	False	96	86,400	9.02 ± 0.00	78.53 ± 43.92	77.09 ± 43.10	76.84 ± 42.97	65.99 ± 37.70
		128	152,064	9.02 ± 0.00	99.37 ± 0.24	97.44 ± 0.13	97.61 ± 0.20	90.06 ± 2.93
	True	96	86,400	21.39 ± 6.99	74.06 ± 41.44	79.18 ± 44.26	79.21 ± 44.28	78.12 ± 43.67
		128	152,064	23.75 ± 1.36	92.92 ± 3.79	98.88 ± 0.36	99.34 ± 0.11	97.17 ± 2.15
Diameter	False	96	84,672	24.09 ± 0.00	71.58 ± 40.13	66.15 ± 36.99	55.66 ± 31.15	-5.20 ± 41.28
		128	149,760	24.09 ± 0.00	74.04 ± 41.39	67.20 ± 37.57	58.32 ± 32.60	27.47 ± 52.59
	True	96	84,672	26.06 ± 1.11	33.94 ± 19.12	70.47 ± 39.40	52.90 ± 29.61	60.01 ± 33.55
		128	149,760	27.43 ± 0.15	49.85 ± 4.07	90.02 ± 0.44	75.79 ± 1.57	30.02 ± 64.80

Table 1: **Online adversarial data evolution:** Adversarial evolution of mazes during training increases NCAs’ generalization ability (particularly to larger 32×32 grids), as well as the mean shortest-path/diameter length of mazes in the dataset.

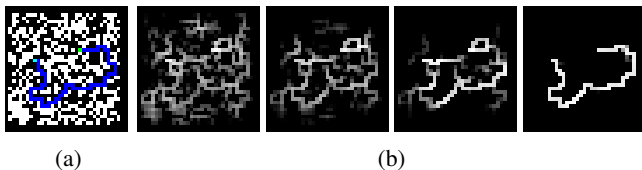


Figure 1: **Learned pathfinding behavior.** a) An example maze with shortest path, where blue, green, dark blue, and black respectively represent source, target, path, and wall tiles. b) A learned model computes the shortest path on an out-of-distribution example. It first activates all traversable tiles, then strengthens activations between source and target, while gradually pruning away the rest.

nels for empty tiles, walls, and sources and targets where applicable). These mazes are concatenated with a number of 0-initialized hidden channels, and the NCA is repeatedly applied to these grids 64 times, evolving hidden activations, after which point the first hidden channel is taken as the model’s prediction of a path over the grid. Cross-entropy loss is then computed between this prediction and a target path represented as a binary array.

The training dataset comprises 10,000 16×16 mazes, which are generated by randomly placing empty, wall, source, and target tiles until the target is reachable by the source. The resulting paths are relatively short, ≈ 9 tiles on average. We additionally test models on 32×32 mazes, which are generated as above, with a mean path length of ≈ 13 tiles. To generate target paths in the shortest path problem, we compute the shortest path between source and target nodes using BFS. For the diameter problem, we compute shortest paths between all pairs of empty tiles and take the greatest such path as the diameter of the grid.

For the sake of evaluation, accuracy (inverse loss) is normalized against all-zero output, which would achieve $\approx 97\%$ accuracy on the dataset. Accuracy can thus be negative

when, e.g., a model predicts a path comprising a majority of non-overlapping tiles relative to the true path. Finally, we record whether, after rounding, the output perfectly matches the target path, i.e. the percentage of target paths perfectly completed, denoted as *pct. complete* in tables. The behavior of a learned pathfinding NCA is pictured in 1.

Adversarial Environment Evolution

During training, we evolve grids to maximize a fitness value corresponding to the learned models’ loss relative to the target path. When the model’s loss falls below a certain threshold, we randomly select among the fittest environments, randomly flip empty/wall tiles with 10% probability, and (in the shortest path domain) randomly select a new source tile with 5%, and move the target tile to a random empty cell within the connected region occupied by the source tile.

We evaluate the model’s performance on these new mazes and use them to replace any less-fit environments in the current training set. In Table 1, we see that such adversarial environment evolution leads to increased generalization on held out test mazes, particularly when these mazes are larger (32×32) than those in the training set (16×16). Additionally, this process tends to increase the complexity of the target paths in the training set.

Conclusion

Neural Cellular Automata can be trained to solve pathfinding problems over grids, modeling solutions via morphogenesis. But generating a balanced dataset with full coverage of the problem space is challenging. We show that this dataset can be evolved during training, adversarially with respect to the learned model, leading to NCAs that generalize better to out-of-distribution environments. Future work may extend our approach to other more complex problems over grids such as puzzle game playing, generating curricula of increasingly complex level layouts and solutions for training robust NCA-based player controllers.

References

- Huijuan Wang, Yuan Yu, and Quanbo Yuan. Application of dijkstra algorithm in robot path-planning. In *2011 Second International Conference on Mechanic Automation and Control Engineering*, pages 1067–1069, 2011. doi: 10.1109/MACE.2011.5987118.
- L. Fu, D. Sun, and L.R. Rilett. Heuristic shortest path algorithms for transportation applications: State of the art. *Computers and Operations Research*, 33(11):3324–3343, 2006. ISSN 0305-0548. doi: <https://doi.org/10.1016/j.cor.2005.03.027>. URL <https://www.sciencedirect.com/science/article/pii/S030505480500122X>. Part Special Issue: Operations Research and Data Mining.
- Ole Stenzel, Omar Pecho, Lorenz Holzer, Matthias Neumann, and Volker Schmidt. Predicting effective conductivities based on geometric microstructure characteristics. *AIChE Journal*, 62(5):1834–1843, 2016.
- Xian Yeow Lee, Joshua R Waite, Chih-Hsuan Yang, Balaji Sessa Sarath Pokuri, Ameya Joshi, Aditya Balu, Chinmay Hegde, Baskar Ganapathysubramanian, and Soumik Sarkar. Fast inverse design of microstructures via generative invariance networks. *Nature Computational Science*, 1(3):229–238, 2021.
- Ahmed Khalifa, Philip Bontrager, Sam Earle, and Julian Togelius. Pcgrl: Procedural content generation via reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 16, pages 95–101, 2020.
- Richard Bellman. On a routing problem. *Quarterly of applied mathematics*, 16(1):87–90, 1958.
- Lester R Ford Jr. Network flow theory. Technical report, Rand Corp Santa Monica Ca, 1956.
- Edward F Moore. The shortest path through a maze. In *Proc. Int. Symp. Switching Theory, 1959*, pages 285–292, 1959.
- Duane Merrill, Michael Garland, and Andrew Grimshaw. Scalable gpu graph traversal. *ACM Sigplan Notices*, 47(8):117–128, 2012.
- Edsger W Dijkstra et al. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.
- Alexander Mordvintsev, Ettore Randazzo, Eyvind Niklasson, and Michael Levin. Growing neural cellular automata. *Distill*, 5(2):e23, 2020.
- Alexander Mordvintsev, Eyvind Niklasson, and Ettore Randazzo. Texture generation with neural cellular automata. *arXiv preprint arXiv:2105.07299*, 2021.
- Yulun Zhang, Matthew Fontaine, Varun Bhatt, Stefanos Nikolaidis, and Jiaoyang Li. Arbitrarily scalable environment generators via neural cellular automata. *Advances in Neural Information Processing Systems*, 36:57212–57225, 2023.
- Sam Earle, Justin Snider, Matthew C Fontaine, Stefanos Nikolaidis, and Julian Togelius. Illuminating diverse neural cellular automata for level generation. *arXiv preprint arXiv:2109.05489*, 2021.
- Alexandre Variengien, Stefano Nichele, Tom Glover, and Sidney Pontes-Filho. Towards self-organized control: Using neural cellular automata to robustly control a cart-pole agent. *arXiv preprint arXiv:2106.15240*, 2021.
- Elias Najarro, Shyam Sudhakaran, Claire Glanois, and Sebastian Risi. Hypernca: Growing developmental networks with neural cellular automata. *arXiv preprint arXiv:2204.11674*, 2022.