Pathfinding Neural Cellular Automata with Local Self-Attention

Felix Reimers[1,2], Sanyam Jain[1], Aarati Shrestha[1], and Stefano Nichele[1,3]

[1] Østfold University College, Halden, Norway
[2] Alfred-Wegener-Institut, Bremerhaven, Germany
[3] Oslo Metropolitan University, Oslo, Norway
{stefano.nichele}@hiof.no

Abstract. Current artificial intelligence systems are rather rigid and narrow, if compared to the adaptivity and the open-endedness of living organisms. Neural Cellular Automata (NCA) are an extension of traditional CA, where the transition rule is replaced by a neural network operating on local neighborhoods. NCA provide a platform for investigating more biologically plausible features of emergent intelligence. However, an open question is how can collections of cells in an NCA be trained to collectively explore an environment in search for energy sources and find suitable paths to collect them. In this work, we utilize an NCA equipped with a local self-attention mechanism trained with gradient descent for pathfinding. Our results show that NCA can be trained to achieve such task and collect energy sources, while being able to redistribute the available energy to neighboring alive cells. Ongoing work is exploring how those abilities may be incorporated in NCA to solve tasks with increased adaptivity and general intelligence.

Keywords: Neural Cellular Automata · Local Attention · Neural Network · Artificial Intelligence · Artificial Life.

1 Introduction

Cellular Automata (CA) have often been used to study how simple individual agents may self-organise with local information and give rise to an emergent collective behavior, in order to perform a global task. Recently, Neural Cellular Automata (NCA) have been proposed as an extension to traditional CA, where the transition rule is replaced by a neural network [1,2]. Such NCA models may be used as bottom-up machine learning tools [4] or, most notably, as substrates to explore how features of intelligence may emerge in artificial organisms [5,6]. One of the most essential tasks that even simple biological organisms, such as slime mold [7] or caenorhabditis elegans [8], can perform in order to survive is the exploration of their environment to reach energy sources.

In this work, we investigate if an agent, i.e., a collection of cells in a neural cellular automaton controlled by a uniform neural network using local information, can be trained to seek out energy sources and thus navigate around the grid to survive for a prolonged period of time. In order to succeed, an agent would need to be able to move and redistribute the available energy across neighboring cells, as well as the ability of pathfinding to detect new energy sources. As typically done in NCA models, we use multiple CA layers (channels) besides the standard visible CA channel (which represents the alive cells and their respective states). We include an energy channel, representing the amount of energy present in a specific location (either in the environment or in a living cell) which is consumed by cells to stay alive, and a chemical channel representing the concentration of chemicals in a site. The chemical channel works similarly to pheromones, which increase in the presence of a living organism or dissipate over time otherwise. We test our trained NCA to reach different concentrations of energy in the environment, and investigate their emergent behavior and survival.

The long term aim of this exploratory work is to investigate how artificial intelligence (AI) systems may become more open-ended [9], adaptive [10,11], and general, by exploiting some of the features of cellular automata and dynamical complex systems [12].

2 Background and Related Work

Classical CA are discrete computational models consisting of a grid of cells that evolve over time based on a set of predefined rules. CAs are widely used to study complex systems, where local interactions gives rise to an emergent behaviour at the global level. Extensions to classical CA have recently been introduced, such as Lenia (Continuous CA) [6] and Neural CA [1,2], that have proven to be a valuable tool for studying biologically-relevant dynamical processes, as pattern formation and morphogenesis.

In the work herein, we use the Neural CA proposed in [2] (with the additions outlined in the Section 3), where the authors introduce a differentiable framework for CA, i.e., trainable with gradient descent. Neural networks are used to control the state update of a 2D CA consisting of a visible layer and several hidden layers. Such Neural CA allows the growth and development of complex structures, demonstrating robustness to perturbations and regeneration. The use of linear convolution operations, combined with local cell updates, produces a highly complex multi-level update rule. Another Neural CA approach is presented in [1], where neuroevolution is used to train the neural update rules.

The work in [3] applies Neural CA (NCA) for bottom-up classification, exploring whether CA can achieve global agreement on the composition of handwritten digits by utilizing local message passing. This research addresses the problem of how cell collectives determine their anatomical structure and classify the large-scale morphology that they are part of. Another application domain includes the development of a control system for a cart-pole agent [16] via self-organisation. The work in [17] uses Heterogeneous CA update rules as testbed for an Ising model and spiking neural networks to model biological neural networks. In [4] a Neural CA using a Vision Transformer including self-attention is used to perform reconstruction of images. Very recent work [15] uses Isotropic Neural CA [2], allowing symmetry breaking. Such work focuses on enabling an inner sense of orientation in cellular systems, which parallels how real-world organisms develop organs that have a sense of directionality. In [13] a NCA framework that incorporates goal embeddings through iterative sampling is introduced, allowing for effective guidance of cells towards desired behaviors. Their approach demonstrates the ability to achieve diverse behaviors in a morphing image experiment, although it lacks generative capabilities to learn latent distributions. Finally, the work in [14] preliminarily explores how NCA agents may be able to reach energy sources in a similar fashion to slime molds.

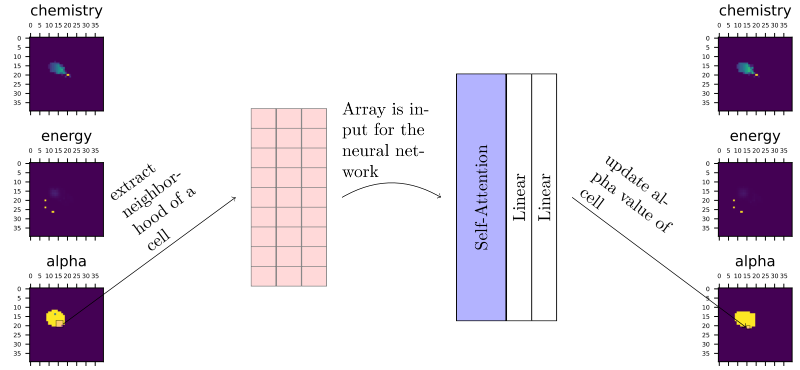
chemistry
energy
alpha
extract neighbor-hood of a cell
Array is in-put for the neural net-work
Self-Attention
Linear
Linear
update al-pha value of cell
chemistry
energy
alpha
0 5 10 15 20 25 30 35

Fig. 1: An update step of the cellular automaton by the neural network. On the left is the CA grid. From this the neighborhood of a single cell is extracted ($9 \times 3 = 27$). The resulting array propagates through the network and an update is computed, which is added to the alpha channel of the grid seen on the right. Before and after, the deterministic updates are performed on the chemistry and energy channels.

3 Proposed Method

In this study, we use a two-dimensional cellular automaton with continuous values (allowing for a differentiable update function). It has a discrete 40 by 40 grid with each cell having 3 channels. The channels are a chemistry channel, an energy channel, and the alpha channel (representing the alive cells). The cell's state is either updated by the output of an uniform neural network (for the alpha channel) or by fixed update rules (for the other channels). Our implementation supports additional hidden CA channels (currently not used in this work).[4]

The neural network consists of three layers, similar to the architecture in [2]. It operates locally: the channels of a single cell, and its eight neighboring cells, are the input to the neural network ($9 \times 3 = 27$ in total). The network computes an update for the alpha channel, which is then applied to the CA. Only the cells which are "alive", e.g. with an alpha channel over 0.1, and its neighbors are updated. The alpha channels are clipped to values between 0 and 1. The energy channel is initialized at the start of the training or testing. The values of the energy are updated deterministically, depending on the alpha channel: Energy, which is present in a cell with an alive alpha channel, is distributed within the neighborhood proportionally to their relative alpha channels, e.g. a cell with a high alpha value receives more energy from neighboring cells than a cell with a low alpha value. New energy deposits may be introduced during training. Otherwise, the cells will inevitably die after some iterations, as every update in an iteration costs a certain amount of energy. This introduces a notion of scarcity, similar to how it can be found in natural environments. Similarly, the chemistry channel is updated: On every alive cell, the chemistry value is increased, while on cells that are not alive the value is decreased until the initial value of 0 is reached. Those deterministic updates happen after the update of the alpha channel of the neural network.

The neural network is updating cells one at a time, but with the same topology and the same synaptic weights. This means, that there is only one agent operating on the grid, but for every update it can only use the locally available information. This reduces computational complexity, but also simulates either a single organism or a group of organisms which are similar in their genetic code but can still act independently based on their surrounding.

An example of the CA neural network and update step is shown in Figure 1.

3.1 NCA Architecture with Local Attention

The first layer is a self-attention layer. The motivation for including a self-attention layer is to enable learning which parts of the neighborhood are more important and should have more focus. As input it takes nine cells, the currently observed cell as well as its eight neighbors. All channels are used, which means, that the input to the neural network is an array of size 3 by 9. The layer has 633 trainable parameters. An important aspect of this is that the attention is applied only locally; cells which are not in the immediate neighborhood are disregarded. The self-attention mechanism is the canonical self-attention [18], widely used in Large Language Models.

Next comes two linear layers, the last layers output is used as an update to the alpha channel of the current cell. To keep those linear layers local, 1-D convolutional layers are used in the implementation (as done in the original

[4] The repository with the code is openly available at the following link: `https://github.com/Deskt0r/LocalAttentionNCA`

paper [2]). The first linear layer has 544 and the last linear layer has 33 trainable parameters. All layers are fully connected.

This procedure loops through all alive cells and all cells neighboring an alive cell (those can be "resurrected"). Each of those cells receives an update based on its neighbors. The implementation of the attention mechanism is canonical, the locality is achieved by padding the cellular grid and extracting patches of the grid according to the aliveness of the cells.

3.2 Training Routine

The neural network is trained by gradient-based optimization with error backpropagation, i.e., differentiable programming, as typically done in Deep Learning. To train the neural network, a pool based approach is used. Every sample in the pool is a grid which is initialized by setting the channels of a central cell such that it is alive and has a relatively large amount of energy. Furthermore, a second cell, randomly chosen in the vicinity of the center towards on of the four edges, is chosen for every batch as a target and also initialized with some energy. In total there are 8 batches with 4 samples.

For the loss, a mask is calculated which gives every cell a value corresponding to its distance to the target. It is important here to scale those values correctly, in order to give the cells an incentive to not just spread in all directions but in the desired one. The loss is the sum of the products of the alpha values with the values of the distance mask. In this work, there is no penalty term for the consumption of energy in the loss.

During the training phase, the network and the fixed rules update the cellular grid for a random number of iterations between 8 and 16 during each training step. Afterwards, the loss is calculated and backpropagated. This allows the calculation of a gradient. After a number of training steps, the target is changed. First only the location, but in a similar distance. Later on, the target is also put in a greater distance from the center. Furthermore, to avoid catastrophic forgetting, the sample of the batch with the highest loss may be reinitialized. The training runs for a total of 100 steps. This means that the network has to adapt to a changing environment over time, although that change is only partly random.

It could be noticed that depending on the random initialization of the weights of the network, the cells are prone to "die" before being able to be trained, especially when the network runs for a high amount of iterations before the weight's update. This is mitigated by re-initializing the grid in such cases. An example of the training loss for the CA model is shown in Figure 2.

As can be seen in the plot, the loss decreases over time, but is rather volatile. The two big spikes at iteration 15 and 20 can likely be explained by the introduction of targets further away, which poses a new challenge to the neural network that is has to overcome, i.e., a kind of curriculum learning.

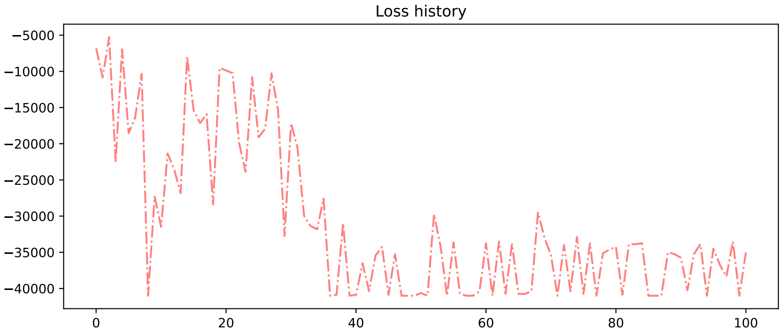

Fig. 2: The loss during the training of the model. Time on the x axis and loss value on the y axis.

4 Experiments

4.1 Experimental Setup

To test the performance of the network, first we executed cases where the target energy pocket is close to the center (Fig. 3 top), as it is in the initial steps of the training. Next, a target further away is picked (Fig. 3 bottom). To see how well the learned behavior generalizes, we also used two energy distribution patterns greatly differing from the ones during the training: In one case, the energy is distributed in a spiral (Fig. 5), in the other one there are two curved lines of energy that can be followed (Fig. 4). While we repeated the experiments several times confirming that the target behaviors are learned, the results have to be seen as qualitative rather then quantitative. Videos of the results can be seen at the following link: `https://github.com/Deskt0r/LocalAttentionNCA`.

5 Results and Discussion

Our tests indicate that for the "survival" of the cells, i.e. the amount of iterations in which the initialized or new cells have an alpha value above 0.1, the distance between the cells and energy deposits and the size of the initial energy available are of high importance, while the shape in which the energy is distributed seems less important.

In Fig. 3 it can be seen that when the next energy depot from the center is too far away, the cells are unavailable to reach it, even if there is enough energy initially available. This shows, that the network has not really learned to search for energy. When it is able to reach the deposit, the cells survive for the whole 100 iterations. In the latter case it dies out after 13 iterations. In Fig. 5 and Fig. 4, the cells are following the laid out pattern, even though they did not encounter such during training. It can not be seen on the single frames in the figures, but the cells show an oscillating behaviour in those experiments. Furthermore, when the cells reach the last energy deposit, the energy seems to spread through the alive cells. During the expansion on the other hand, the energy seems to spread just thickly enough to keep the cells alive.

On a more general level, being able to learn properties typically available in biological organisms such as environment exploration, pathfinding, energy usage, etc., in cellular systems such as Neural CA, may allow the development of more open-ended and general AI systems. Different tasks may be encoded in such systems in the form of environmental signals and energy sources, and the solution to the task would result from the self-organising process of the CA. This exploratory work fits well with the emerging area of self-organising AI [12].

6 Conclusion

Our results show that it is possible to train a neural network, such that the cells under its control can seek energy sources. Similar to a normal cellular automaton, cells receive an update based on their neighborhood, but the neural network with self-attention allows for significantly more complex behaviour.

Further work could include a more sophisticated training routine exposing the network to a larger variety of situations. This could include energy deposits randomly spawning on the grid. Furthermore, a channel for hidden information could be used, which would allow cells to communicate with each other or store information for later use. It would be interesting to see how complex behaviour of the cells can emerge with regards to searching for energy, splitting and joining up, as well as long term energy preservation.

Acknowledgments

We would like to thank the Helmholtz Information and Data Science Academy (HIDA) and the Norwegian Artificial Intelligence Research Consortium (NORA) for providing a stipend for Felix Reimers through a mobility collaboration agreement between HIDA and NORA. We would like to thank Østfold Univerisity College for additional financial support.

References

1. Nichele, S., Ose, M. B., Risi, S., and Tufte, G. (2017). *CA-NEAT: evolved compositional pattern producing networks for cellular automata morphogenesis and replication.* IEEE Transactions on Cognitive and Developmental Systems, 10(3), 687-700.
2. Mordvintsev, A., Randazzo, E., Niklasson, E., and Levin, M. (2020). *Growing neural cellular automata.* Distill, 5(2), e23.
3. Randazzo, E., Mordvintsev, A., Niklasson, E., Levin, M., & Greydanus, S. (2020). *Self-classifying mnist digits.* Distill, 5(8), e00027-002.
4. Tesfaldet, M., Nowrouzezahrai, D., and Pal, C. (2022). *Attention-based Neural Cellular Automata.* Advances in Neural Information Processing Systems, 35, 8174-8186.
5. Gregor, K., and Besse, F. (2021). *Self-organizing intelligent matter: A blueprint for an AI generating algorithm.* arXiv preprint arXiv:2101.07627.
6. Hamon, G., Etcheverry, M., Chan, B. W. C., Moulin-Frier, C., and Oudeyer, P. Y. (2022). *Learning sensorimotor agency in cellular automata.*
7. Beekman, M., and Latty, T. (2015). *Brainless but multi-headed: decision making by the acellular slime mould Physarum polycephalum.* Journal of molecular biology, 427(23), 3734-3743.
8. Qin, J., and Wheeler, A. R. (2007). *Maze exploration and learning in C. elegans.* Lab on a Chip, 7(2), 186-192.
9. Stanley, K. O. (2019). *Why open-endedness matters.* Artificial life, 25(3), 232-235.
10. Pontes-Filho, S., Walker, K., Najarro, E., Nichele, S., and Risi, S. (2022). *A single neural cellular automaton for body-brain co-evolution.* In Proceedings of the Genetic and Evolutionary Computation Conference Companion (pp. 148-151).
11. Nadizar, G., Medvet, E., Nichele, S., and Pontes-Filho, S. (2022). *Collective control of modular soft robots via embodied Spiking Neural Cellular Automata.* arXiv preprint arXiv:2204.02099.
12. Risi, S. (2021). *The future of artificial intelligence is self-organizing and self-assembling.* Available at: sebastianrisi.com.
13. Sudhakaran, S., Najarro, E., and Risi, S. (2022). *Goal-Guided Neural Cellular Automata: Learning to Control Self-Organising Systems.* arXiv preprint arXiv:2205.06806.
14. Barbieux, A., and Canaan, R. (2023). *EINCASM: Emergent Intelligence in Neural Cellular Automaton Slime Molds.* arXiv preprint arXiv:2305.13425.
15. Randazzo, E., Mordvintsev, A., and Fouts, C. (2023). *Growing Steerable Neural Cellular Automata.* arXiv preprint arXiv:2302.10197.
16. Variengien, A., Pontes-Filho, S., Glover, T. E., and Nichele, S. (2021). *Towards Self-organized Control: Using Neural Cellular Automata to Robustly Control a Cart-pole Agent.* Innovations in Machine Intelligence (IMI), vol. 1, pp. 1-14.
17. Khajehabdollahi, S., Giannakakis, E., Buendia, V., Martius, G., and Levina, A. (2023). *Locally adaptive cellular automata for goal-oriented self-organization.* arXiv preprint arXiv:2306.07067.
18. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... and Polosukhin, I. (2017). *Attention is all you need.* Advances in neural information processing systems, 30.

A Appendix

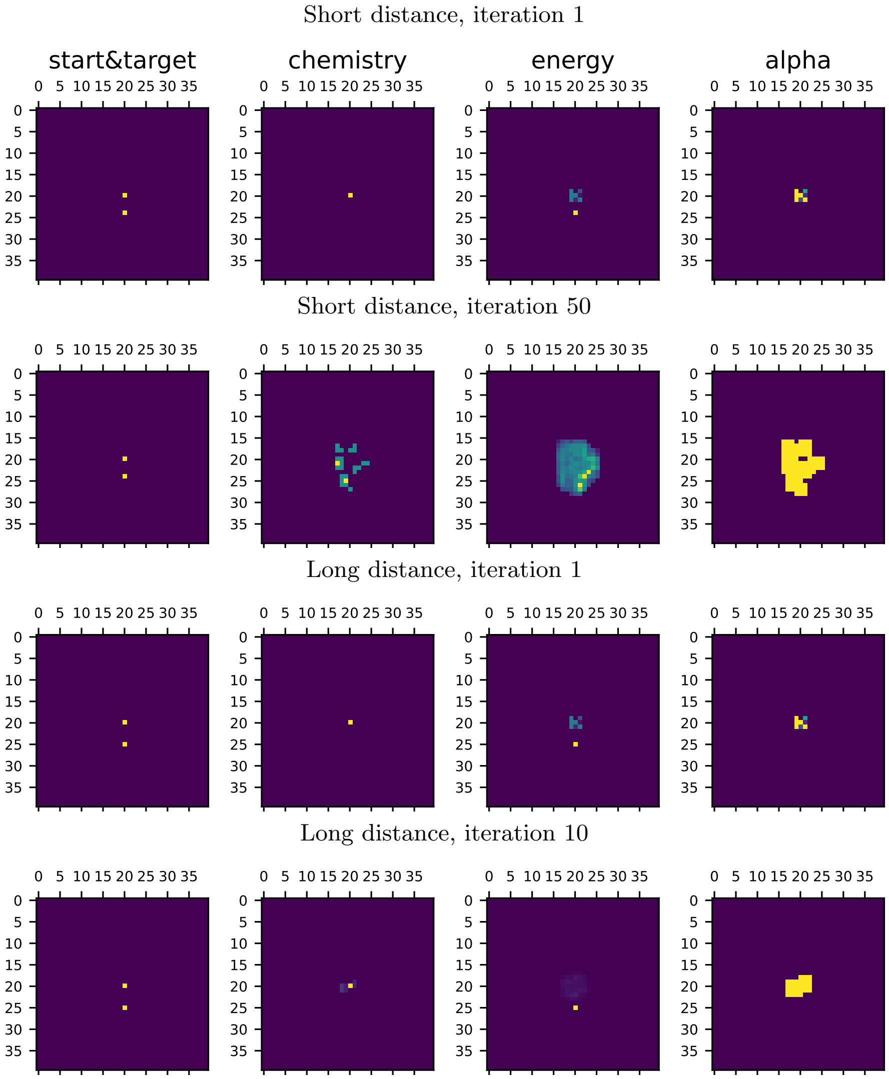

Fig. 3: *Short and long distance.* From left to right: seed of the grid with energy at the starting point and additional locations, chemistry channel, energy channel, alpha channel. All energy depots are initialized with a value of 5. The single additional energy deposit is located below the center. Short distance: 4 cells away; large distance: 5 cells away.

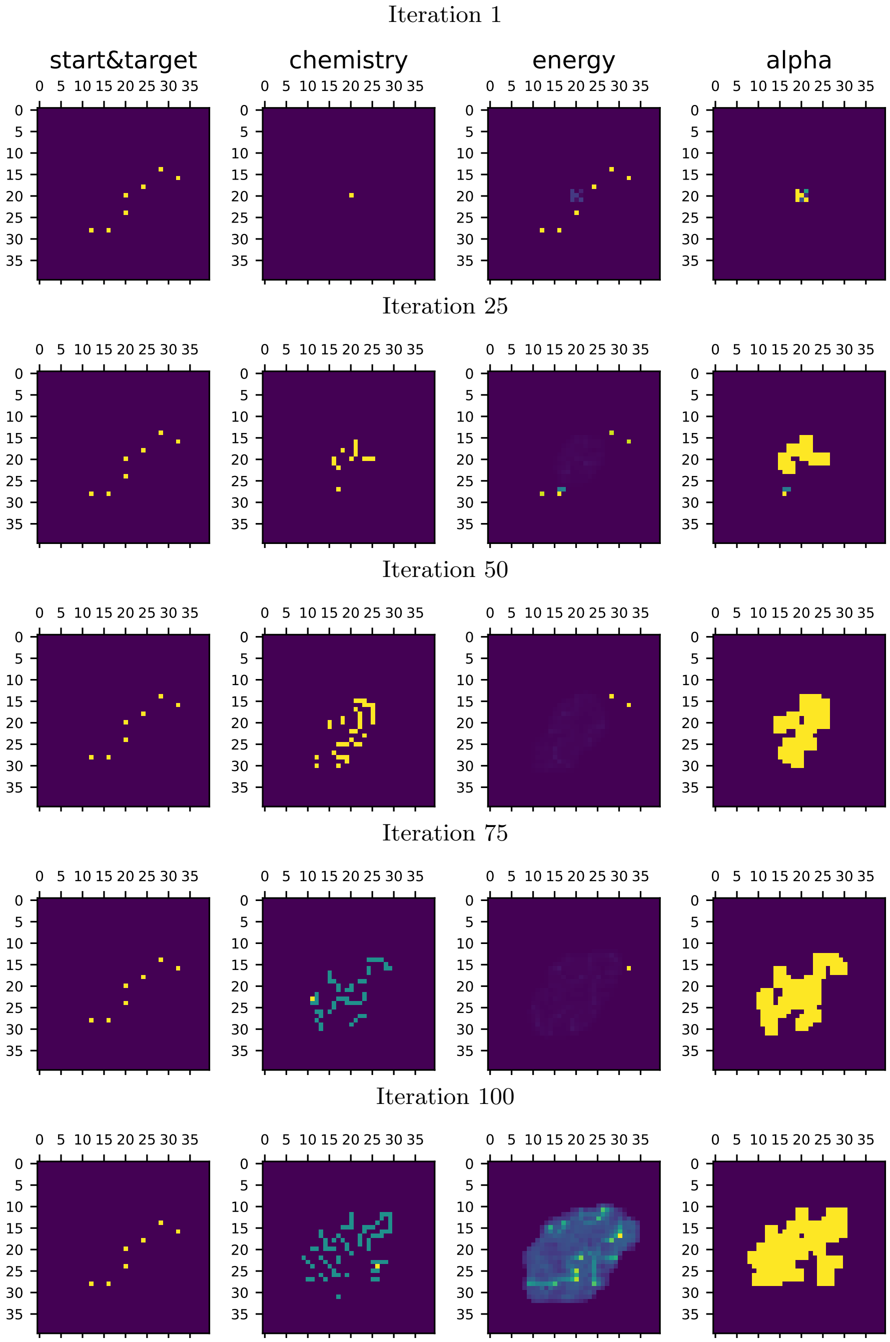

Fig. 4: *Split.* From left to right: seed of the grid with the energy depot at the starting point and additional locations, chemistry channel, energy channel, alpha channel. The energy is located in two paths from the center. All energy depots are initialized with a value of 5.

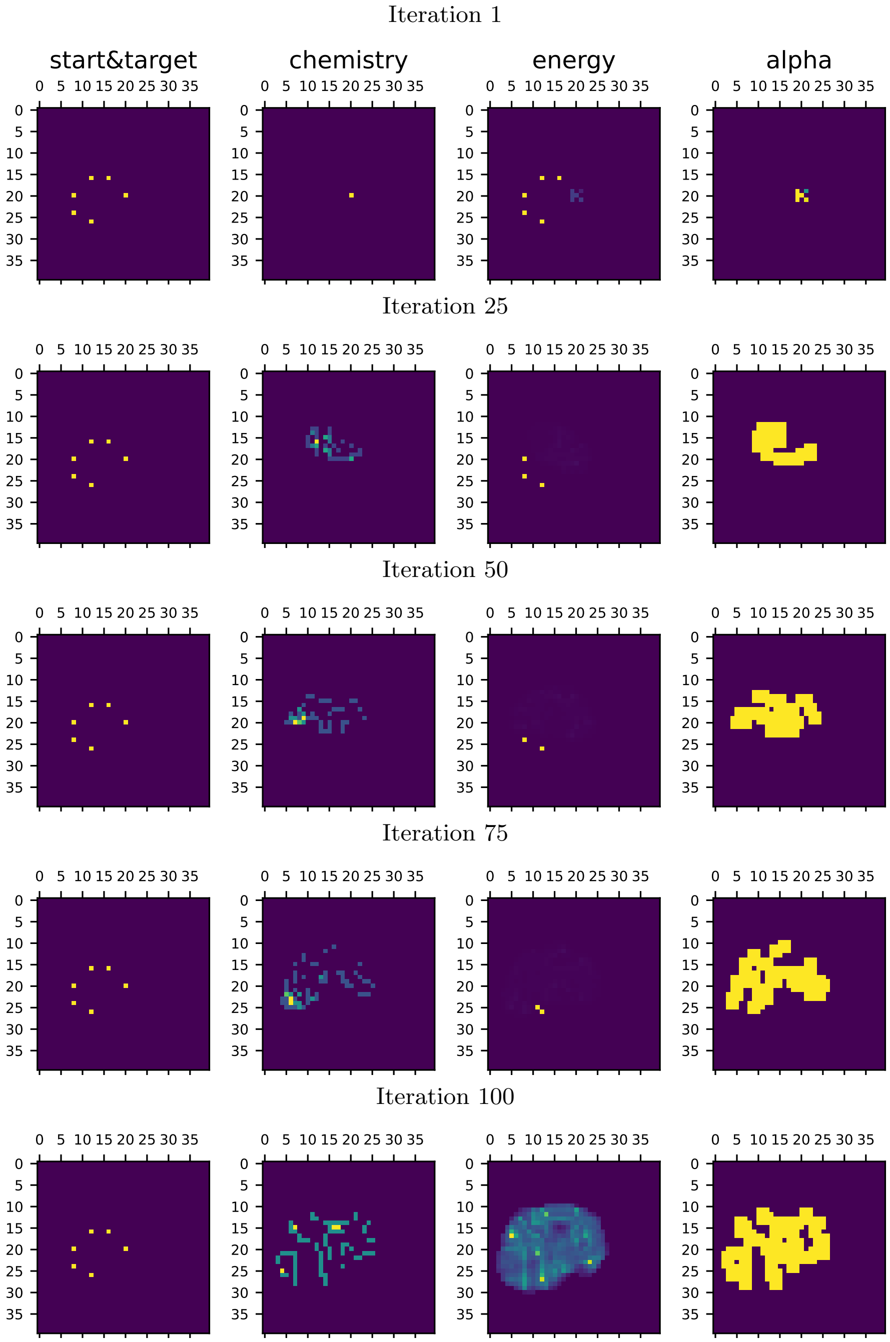

Fig. 5: *Spiral.* From left to right: seed of the grid with the energy depot at the starting point and additional locations, chemistry channel, energy channel, alpha channel. The energy is located in spiral starting from the center. All energy depots are initialized with a value of 8.