A Extended dataset descriptions

A.1 Cellular Automata

As a first interesting step towards learning algorithms, we focus on cellular automata probelms. We consider a variety of different settings such as 1D and 2D automata, these include more well known instances such as Wireworld or Game of Life, which despite its simple behaviour is known to be Turing commplete.

1D-Cellular Automata The 1D-Cellular Automata dataset consists of one-dimensional cellular automata systems, each defined by one of the possible 256 rules. Each rule corresponds to a distinct mapping of a cell's state and its two neighbors to a new state. Figure 7 provides a graphical illustration of one such automata. Note, that we need todistinguish between the left and right neighbors in order to capture all rules in GraphFSA.

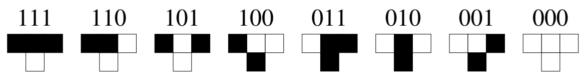

Figure 7. Visualization of 1D CA rule 30 - top row shows different combinations (left neighbor state, current cell state, right neighbor state), bottom row center shows new state value for the cell

Game Of Life The GameOfLife dataset captures the essence of Conway's Game of Life that progresses based on its initial state and a set of simple rules.

The progression of Conway's Game of Life is dictated by a set of simple rules applied to each cell in the grid, considering its neighbors. Using the Moore neighborhood, which includes all eight surrounding cells, these rules are as follows:

1. Birth: A cell that is dead will become alive if it has exactly three living neighbors.
2. Survival: A cell that is alive will remain alive if it has two or three living neighbors.
3. Death:
   (a) *Loneliness*: A cell that is alive and has fewer than two living neighbors will die, mimicking underpopulation.
   (b) *Overpopulation*: A cell that is alive and has more than three living neighbors will die, representing overpopulation.

Toroidal vs. non-toroidal For this dataset, we consider both toroidal and non-toroidal variations:

1. *Toroidal*: In the toroidal variation, the board's edges wrap around, creating a continuous, closed surface. This means cells on the edge have neighbors on the opposite edge.
2. *Non-Toroidal*: In the standard, non-toroidal variation, cells on the board's edge only consider neighbors within the boundary.

Our dataset consists of input/output pairs where we randomly initialize the grid and then apply the Game of Life rules for a fixed number of steps. We represent this dataset through grid graphs and use the Moore neighborhood.

Hexagonal Game Of Life The Hexagonal Game Of Life introduces a variation where cells are hexagonal as opposed to the traditional square grid. This change in cell structure offers a fresh set of neighbour cells, which can lead to distinct patterns and evolutions. A visual representation can be found in Figure 4.

WireWorld The WireWorld dataset revolves around the cellular automaton 'WireWorld' where cells can take one of four states: empty, electron head, electron tail, and conductor. It's especially renowned for its capability to implement digital circuitry. In the dataset, we observe the evolution of a given cellular configuration over specified iterations.

A.2 Graph algorithms

Taking inspiration from the datasets utilized by Grötschla et al. [6], we create datasets for various classical graph problems. To ensure versatility and scalability, we generate new graphs for each problem instance and compute the corresponding ground truth during dataset creation. This approach enables us to construct datasets that not only encompass graphs of specific sizes but also facilitate evaluation on larger extrapolation datasets. We explore the following graph problems that helps us to explore different capabilities of our model.

A.2.1 Distance

The distance problem involves determining whether each node in the graph has an even or odd distance from the root node. To formulate this problem, we define input values for each problem instance, representing each node's state in the graph. Among these nodes, one is designated as the root, while the others are marked as non-root inputs. The output is assigned a binary value (0 or 1) for each node based on distance $\mod 2$ from the root, where "distance" represents the length of the shortest path between the root and a node.

A.2.2 RootValue

In the root value problem, we want to propagate a value from the root throughout a path graph. One node in the graph is assigned a root label and a binary value (0,1). The objective is to propagate this binary value from the root across the entire graph.

A.2.3 PathFinding

The PathFinding problem determines whether a given node lies on the path between two marked nodes within a tree. The dataset comprises different trees, and two nodes are explicitly marked as relevant nodes within each tree. The objective is to predict whether a specific node in the tree, which is not one of the labeled nodes, lies on the path connecting these two marked nodes.

A.2.4 PrefixSum

The PrefixSum Dataset involves paths represented as sequences of nodes where each has an initial binary feature (either 1 or 0). The task is to predict the sum of all initial features to the right of each node in the path, considering modulo two arithmetic.

B Additional model evaluation

B.1 1D cellular automata

We evaluate the models on 1D Cellular Automata evaluation for the different baselines and consider a larger graph during and multiple timesteps t for our evaluation. We report the results in Table 3 for models that were trained on $t = 1$ and in Table 4 for models trained on $t = 2$.

Table 3. 1D Cellular Automata evaluation for the different baselines, we consider a path of size 10 for the extrapolation data, and t indicates the number of times the CA rule has been applied. We report accuracy and all models were trained for $t = 1$.

Model	t = 1	t = 2	t = 5	t = 10	t = 20	t = 50	t = 100
GNCA	0.75 ± 0.00	0.66 ± 0.05	0.71 ± 0.08	0.79 ± 0.17	0.78 ± 0.17	0.77 ± 0.18	0.77 ± 0.18
Recurrent GNN	0.75 ± 0.00	0.50 ± 0.08	0.47 ± 0.17	0.47 ± 0.29	0.47 ± 0.30	0.47 ± 0.30	0.47 ± 0.30
Diff-FSA	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00

Table 4. 1D Cellular Automata evaluation for the different baselines, we consider a path of size 10 for the extrapolation data, and t indicates the number of times the CA rule has been applied. We report accuracy and all models were trained for $t = 2$.

Model	t = 1	t = 2	t = 5	t = 10	t = 20	t = 50	t = 100
GNCA	0.49 ± 0.11	0.73 ± 0.03	0.70 ± 0.13	0.86 ± 0.16	0.86 ± 0.16	0.86 ± 0.16	0.86 ± 0.16
Recurrent GNN	0.64 ± 0.15	0.79 ± 0.00	0.58 ± 0.13	0.66 ± 0.26	0.67 ± 0.29	0.67 ± 0.30	0.68 ± 0.30
Diff-FSA	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00

B.2 2D cellular automata

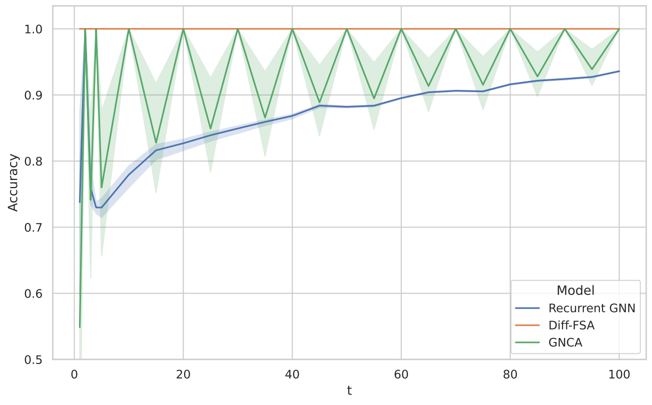

Figure 8. Mean accuracies for Game of Life on a regular grid when learned with 2 steps and applied for t steps during inference (for 10 seeds each). All models report high accuracy for the training setup. However, performance deteriorates during inference when more steps are executed except for Diff-FSA.

Game of Life results. Specifically, we use a state space of two for Game of Life and a bounding parameter of five. We let all models execute 2 steps in order to learn the appropriate transition rules after two iterations for Game of Life. Further, we investigate the generalization capabilities and especially the iteration stability of the learned models. For this, we consider all models which achieve perfect training accuracy and let them run on larger 10 node paths for more timesteps t than during training. The results are depicted in Figure 8 for the regular Game of Life and in Figure 9 for the hexagonal variant. Note, that the recurrent GNN deteriorates outside the training distribution. Similarly, the GNCA baseline struggles to generalize as it has not learned the correct intermediate transitions for odd number of rules. The Diff-FSA on the other hand exhibits good iteration stability across the whole range of iterations.

B.3 Dataset generator

We perform further evaluations on more datasets generated by our dataset generator GRAB. The groudturth automaton consists of 4 states and we test multiple Diff-FSAwith 4, 5, and 6 states as well as the baselines over 10 runs. We report the achieved accuracies for the different experiments in Table 5.

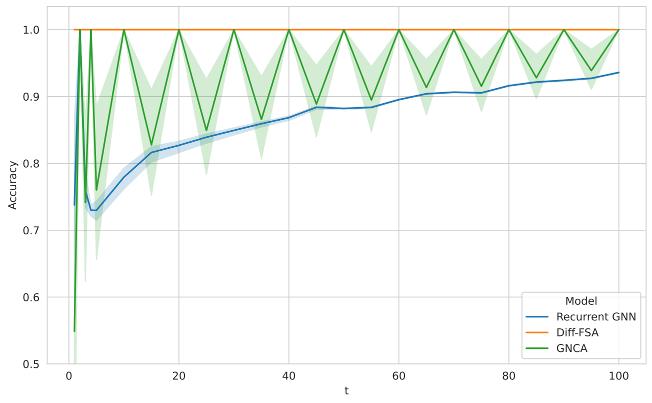

Figure 9. Mean accuracies for Game of Life on a hexagonal grid when learned with 2 steps and applied for t steps during inference (for 10 seeds each). All models report high accuracy for the training setup. However, performance deteriorates during inference when more steps are executed except for Diff-FSA.

B.4 Algorithms

We evaluate the GraphFSA framework and in particular our Diff-FSAand the baselines on more challenging algorithmic datasets. We evaluate on the four datasets Distance, PrefixSum, PathFinding and RootValue. The report the results in the tables: 6, 8 and 7.

Table 6. Evaluation of learning graph algorithms on the RootValue dataset. We report the accuracy and standard deviation over 10 runs. All models are trained on graphs of size at most 10 and then tested for extrapolation on larger graph sizes n.

Model	n = 10	n = 20	n = 50	n = 100
GNCA	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.00
Recurrent GNN	1.00 ± 0.00	1.00 ± 0.00	0.92 ± 0.16	0.91 ± 0.16
Diff-FSA	1.00 ± 0.02	0.99 ± 0.02	0.99 ± 0.03	0.99 ± 0.04

Table 7. Evaluation of learning graph algorithms on the PathFinding dataset. We report the accuracy and standard deviation over 10 runs. All models are trained on graphs of size at most 10 and then tested for extrapolation on larger graph sizes n.

Model	n = 10	n = 20	n = 50	n = 100
GNCA	0.58 ± 0.00	0.73 ± 0.00	0.83 ± 0.00	0.89 ± 0.00
Recurrent GNN	0.97 ± 0.02	0.90 ± 0.03	0.88 ± 0.04	0.89 ± 0.09
Diff-FSA	0.85 ± 0.00	0.85 ± 0.00	0.87 ± 0.00	0.91 ± 0.00

Table 8. Evaluation of learning graph algorithms on the PrefixSum dataset. We report the accuracy and standard deviation over 10 runs. All models are trained on graphs of size at most 10 and then tested for extrapolation on larger graph sizes n.

Model	n = 10	n = 20	n = 50	n = 100
GNCA	0.49 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.00
Recurrent GNN	0.97 ± 0.04	0.95 ± 0.06	0.90 ± 0.13	0.84 ± 0.14
Diff-FSA	0.74 ± 0.14	0.67 ± 0.17	0.63 ± 0.20	0.61 ± 0.20

Table 5. Evaluation of GraphFSA on synthetic data provided by GRAB. We report the accuracy and standard deviation over 10 runs. The underlying ground truth consists of an FSA using 4 states. We can test the in-distribution validation accuracy to see how well a model can fit the data. Moreover, we test extrapolation to larger graphs to verify that the rules for underlying automata were successfully learned. Our Diff-FSA models generally perform well across all scenarios. Note that the recurrent GNN performs well but lacks the interpretation and visualization of the learned mechanics as discrete state automata.

Model	Val Acc	n=10	n=20	n=50	n=100
			Experiment 1		
GNCA	0.38 ± 0.00	0.39 ± 0.00	0.40 ± 0.00	0.40 ± 0.00	0.39 ± 0.00
Recurrent GNN	1.00 ± 0.00	0.91 ± 0.10	0.85 ± 0.13	0.82 ± 0.16	0.81 ± 0.16
Diff-FSA (4 states)	0.99 ± 0.02	0.97 ± 0.02	0.96 ± 0.02	0.95 ± 0.03	0.94 ± 0.03
Diff-FSA (5 states)	1.00 ± 0.01	0.98 ± 0.01	0.96 ± 0.02	0.95 ± 0.02	0.95 ± 0.02
Diff-FSA (6 states)	0.99 ± 0.01	0.98 ± 0.01	0.95 ± 0.02	0.94 ± 0.02	0.94 ± 0.02
			Experiment 2		
GNCA	0.66 ± 0.00	0.67 ± 0.00	0.66 ± 0.00	0.65 ± 0.00	0.65 ± 0.00
Recurrent GNN	1.00 ± 0.00	0.84 ± 0.02	0.82 ± 0.02	0.81 ± 0.02	0.80 ± 0.02
Diff-FSA (4 states)	1.00 ± 0.00	0.94 ± 0.01	0.93 ± 0.02	0.92 ± 0.02	0.91 ± 0.02
Diff-FSA (5 states)	1.00 ± 0.00	0.93 ± 0.02	0.92 ± 0.02	0.91 ± 0.02	0.90 ± 0.02
Diff-FSA (6 states)	1.00 ± 0.00	0.94 ± 0.02	0.93 ± 0.02	0.92 ± 0.02	0.91 ± 0.02
			Experiment 3		
GNCA	0.92 ± 0.00	0.95 ± 0.00	0.93 ± 0.00	0.93 ± 0.00	0.93 ± 0.00
Recurrent GNN	0.98 ± 0.03	0.98 ± 0.02	0.96 ± 0.03	0.95 ± 0.03	0.95 ± 0.03
Diff-FSA (4 states)	0.92 ± 0.00	0.95 ± 0.00	0.93 ± 0.00	0.93 ± 0.00	0.93 ± 0.00
Diff-FSA (5 states)	0.91 ± 0.02	0.94 ± 0.03	0.92 ± 0.01	0.93 ± 0.01	0.93 ± 0.01
Diff-FSA (6 states)	0.91 ± 0.02	0.94 ± 0.03	0.92 ± 0.01	0.93 ± 0.01	0.93 ± 0.01
			Experiment 4		
GNCA	0.56 ± 0.00	0.64 ± 0.00	0.63 ± 0.00	0.64 ± 0.00	0.63 ± 0.00
Recurrent GNN	1.00 ± 0.00	0.98 ± 0.05	0.97 ± 0.05	0.97 ± 0.05	0.97 ± 0.05
Diff-FSA (4 states)	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Diff-FSA (5 states)	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Diff-FSA (6 states)	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
			Experiment 5		
GNCA	0.54 ± 0.00	0.54 ± 0.00	0.55 ± 0.00	0.54 ± 0.00	0.55 ± 0.00
Recurrent GNN	1.00 ± 0.00	0.97 ± 0.03	0.94 ± 0.03	0.93 ± 0.03	0.93 ± 0.04
Diff-FSA (4 states)	0.98 ± 0.00	0.96 ± 0.00	0.95 ± 0.00	0.93 ± 0.00	0.93 ± 0.00
Diff-FSA (5 states)	0.98 ± 0.00	0.97 ± 0.00	0.95 ± 0.00	0.94 ± 0.00	0.94 ± 0.00
Diff-FSA (6 states)	0.98 ± 0.00	0.96 ± 0.01	0.95 ± 0.01	0.94 ± 0.01	0.94 ± 0.00