# Small-Data Discovery of Complex Dynamics via Equivariant Neural Networks

Ruijie HE [a,b], Liu HONG [a], Wuyue YANG [c]

2025-10-29

a. Sun Yat-sen University, b. Great Bay University, c. BIMSA

# Outline

# Outline

# 1. Cellular Automata

Cellular automata is a discrete dynamical system that evolves according to a set of rules. All rules are local and deterministic and applied simultaneously to all cells in the grid.

Generally, let $X$ denote the states of all cells in the grid, each cell has a state $q$ in set $Q$ of all states, $f$ denote the rule function, which acts on the state of a cell $X_{t(\boldsymbol{v})}$ and its neighborhood $N_t(\boldsymbol{v})$ to produce the next state $X_{t+1}(\boldsymbol{v})$:

$$X_{t+1}(\boldsymbol{v}) = f(X_t(\boldsymbol{v}), N_t(\boldsymbol{v})).$$

For some specific cases, we can easily realize the transition on 1 or 2 dimensional cellular automata with the aid of convolution operation with carefully picked kernel $\varphi$ and post-processing function $g$:

$$X_{t+1} = g(X_t * \varphi).$$

# 1.1 1D-CA

If we consider 1-dimensional cellular automata, and let the neighborhood of a cell $\boldsymbol{v}$ be the set of cells that are adjacent to $\boldsymbol{v}$, we can uniquely determine the transition dynamics by exhausting the outcome of all possible combinations of the cell state $X_{t(\boldsymbol{v})}$ and that of its neighborhood $N_t(\boldsymbol{v})$:

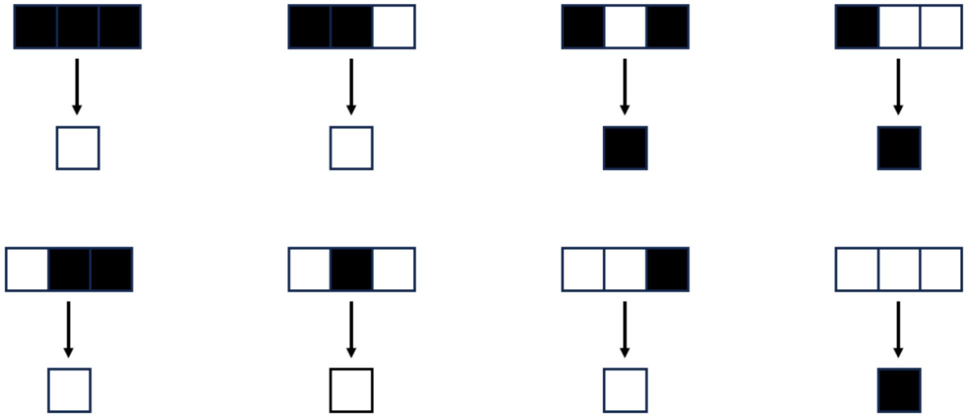


*Figure 1: 1D-Cellular Automata Transition Rules*

# 1.2 Classification of 1D-CA

Apparently, there are only 256 possible sets of transition rules for aforementioned 1D-Cellular Automata, we can number them from 1 to 256.Stephen Wolfram classified 1D-Cellular Automata into 4 classes based on their behavior:

1. **Stable**: System evolves quickly into a stable state.
2. **Oscillators**: System evolves between two or more stable states.
3. **Complex**: System non-periodic behavior, but not enough to be classified as chaos.
4. **Chaos**: System evolves into a highly sensitive, non-periodic behavior.

We can stack the evolution of system states in chronologically order to form a 2D image, to better investigate the behavior of the system.

# 1.2 Classification of 1D-CA

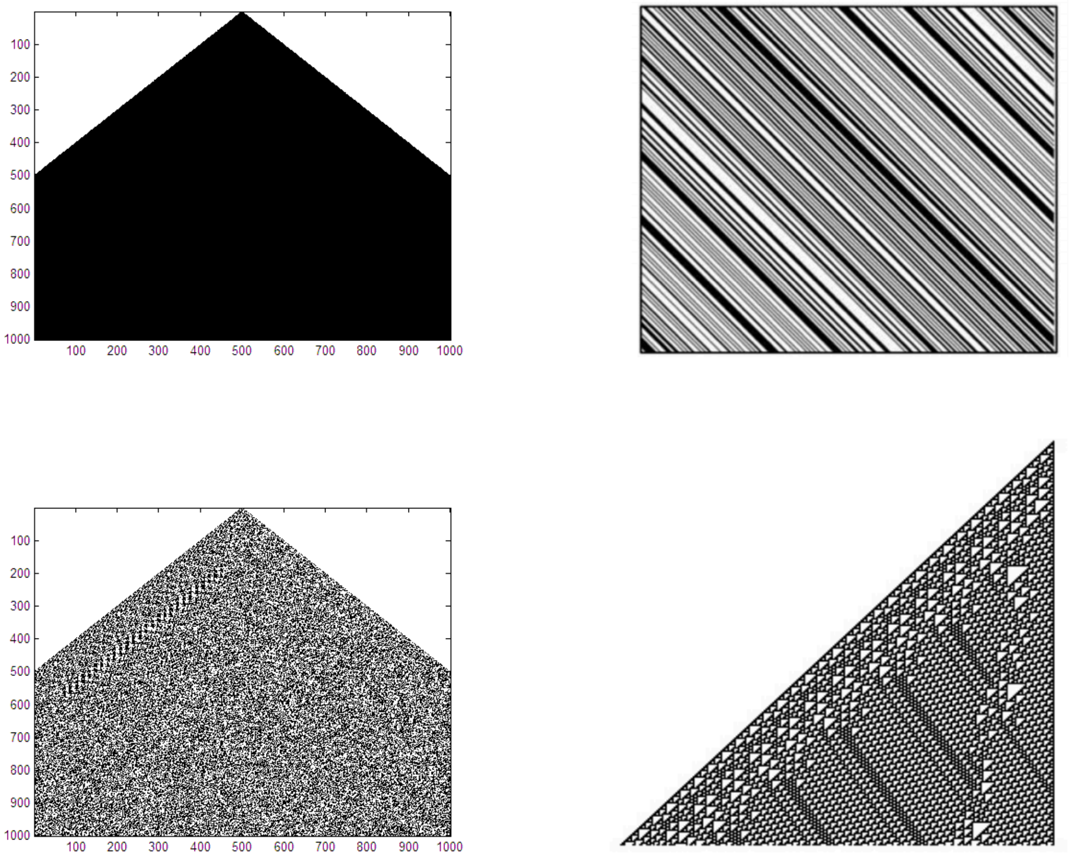



*Figure 2: Evolution trajectory of 1D-Cellular Automata with Different Types of Rules*

# 1.3 Conway's Game of Life

Conway's game of life is a typical 2D-CA with $3 \times 3$ sized neighborhood. The system evolves according to the following rules:

| | |
|---|---|
| **Underpopulation** | A cell dies if it has $< 2$ living neighbors. |
| **Overpopulation** | A cell dies if it has $> 3$ living neighbors. |
| **Survival** | A cell survives if it has 2 or 3 living neighbors. |
| **Birth** | A cell is born if it has exactly 3 living neighbors. |

# 1.3 Conway's Game of Life

## 1.3.1 Creatures

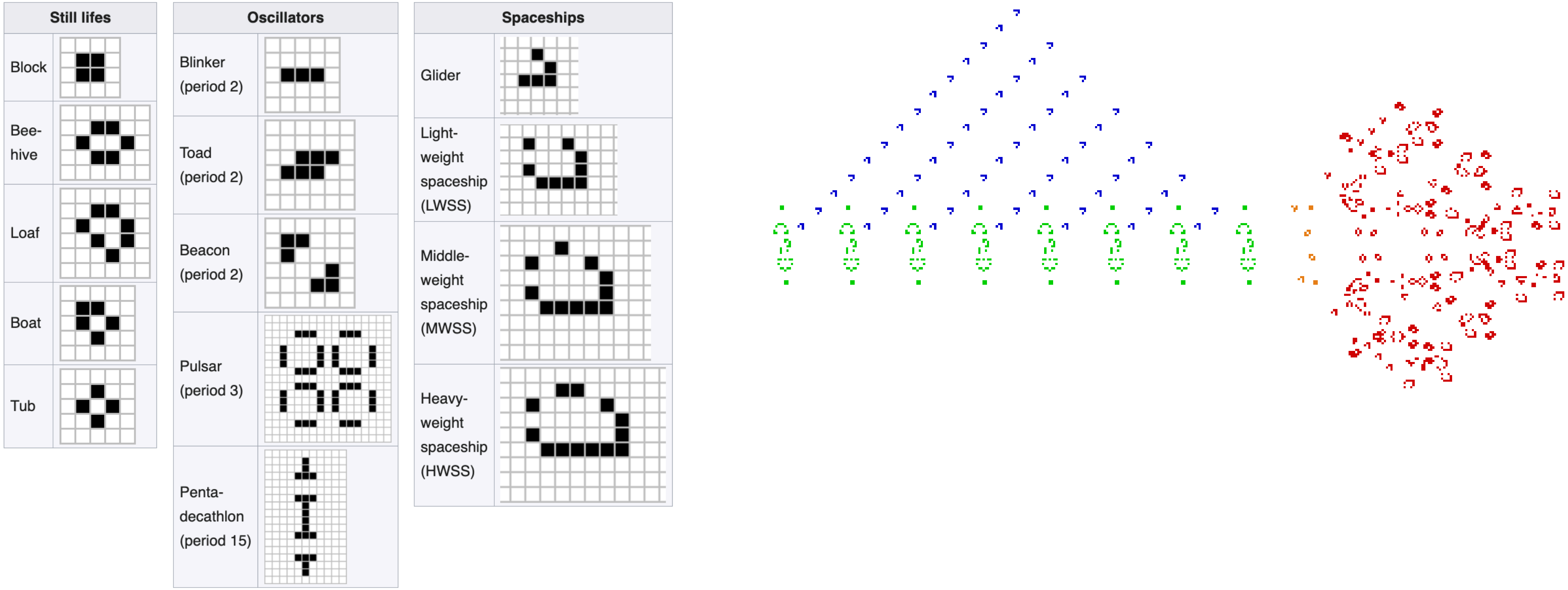



*Figure 4: Creatures in Conway's Game of Life.*

*Adopted from https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life#/media/File:Conways_game_of_life_breeder.png*

# 1.3 Conway’s Game of Life

## 1.3.2 Turing Completeness

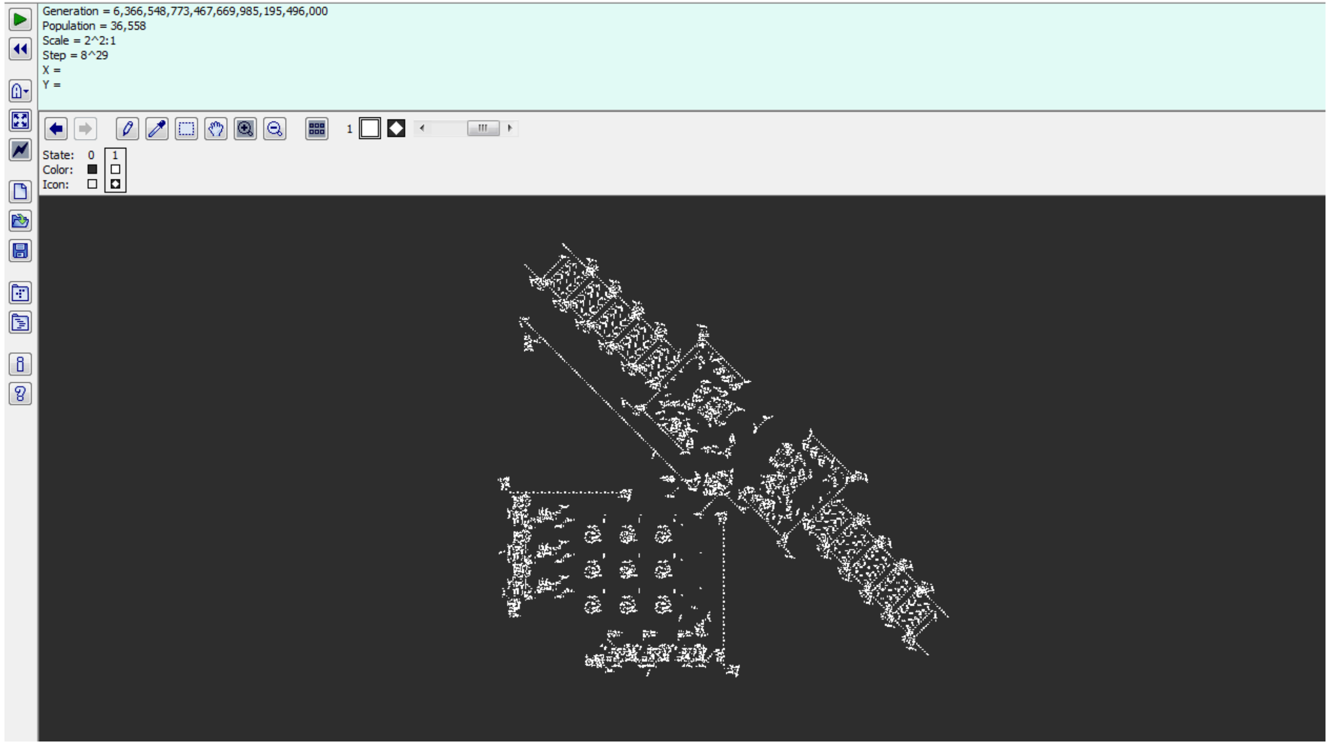



*Figure 5: Turing Completeness of Conway’s Game of Life Realized in Golly.*

*By Andrew Trevorrow and Tomas Rokicki - Screenshot of Golly program, GPL, https://commons.wikimedia.org/w/index.php?curid=18644263*

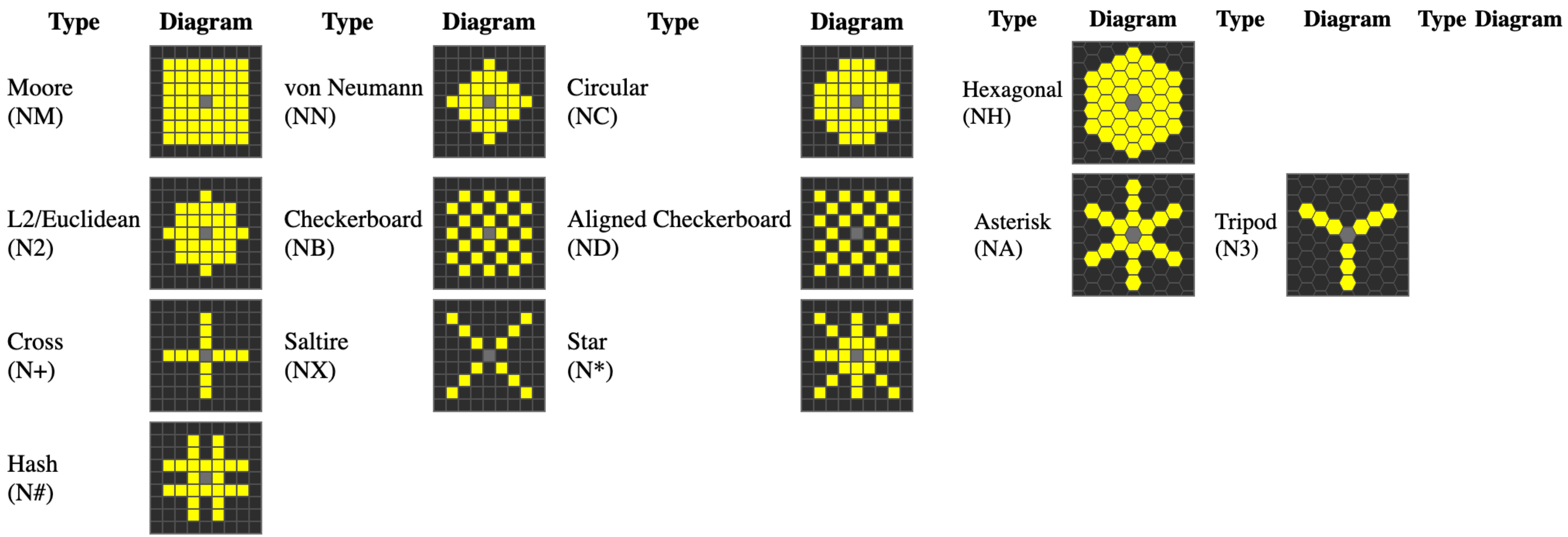
# 1.4 Variants of Life

Typical variants of Conway's Game of Life uses $3 \times 3$ neighborhood, which is called **Moore** neighborhood. We use `BXX/SXX` to denote a rule with `B` being the birth condition and `S` being the survival condition.

| Rule Symbol | Rule Name | Description |
|---|---|---|
| `B3/S23` | Life | John Conway's rule is by far the best known and most explored CA. |
| `B36/S23` | HighLife | Very similar to Conway's Life but with an interesting replicator. |
| `B3678/S34678` | Day & Night | Dead cells in a sea of live cells behave the same as live cells in a sea of dead cells. |
| `B35678/S5678` | Diamoeba | Creates diamond-shaped blobs with unpredictable behavior. |
| `B2/S` | Seeds | Every living cell dies every generation, but most patterns still explode. |
| `B234/S` | Serviettes or Persian Rug | A single 2x2 block turns into a set of Persian rugs. |
| `B345/S5` | LongLife | Oscillators with extremely long periods can occur quite naturally. |

# 1.4 Variants of Life

- **Von Neumann** neighborhood with 4 adjacent cells, has rule symbol like `BXX/SXXV`.
- Likewise, in **hexagonal grid**, we can use 6 adjacent cells to form a hexagonal neighborhood.
- When the neighbors don't contribute equally, we get **non-totalistic rules** or **MAP rules**.
- Rules can be generalized to bigger sizes of neighborhoods.



# Outline

# 2.1 Equivariance

Consider linear space $V$ with a transformation group $\mathfrak{G}$, which we call $\mathfrak{G}$-space, and a function $\Phi : V \to V$. We say $\Phi$ is **equivariant** if it satisfies

$$\Phi(T_{\mathfrak{g}}x) = T'_{\mathfrak{g}}\Phi(x)$$

In which $T_{\mathfrak{g}}$ is the transformation of $V$ corresponding to the group element $\mathfrak{g}$ in $\mathfrak{G}$. $T$ and $T'$ are not necessarily the same, but must be linear representations of elements in $\mathfrak{G}$, i.e., for any $\mathfrak{g}, \mathfrak{h} \in \mathfrak{G}$, we have $T(\mathfrak{g}\mathfrak{h}) = T(\mathfrak{g})T(\mathfrak{h})$.

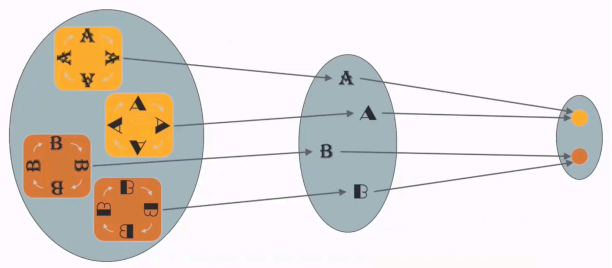


*Figure 7: Equivariance of the human eye*

# 2.1 Equivariance

Typical crystal group $p4$ contains all translation and rotation on $\mathbb{Z}^2$ with unit $\frac{\pi}{2}$. It has the following representation:

$$\mathfrak{g}(r,u,v)=\begin{bmatrix}\cos(\frac{r\pi}{2}) & -\sin(\frac{r\pi}{2}) & u\\ \sin(\frac{r\pi}{2}) & \cos(\frac{r\pi}{2}) & v\\ 0 & 0 & 1\end{bmatrix}$$

in which $r\in\{0,1,2,3\},\ \ (u,v)\in\mathbb{Z}^2$. The result of group element $\mathfrak{g}$ acting on vector $x$ can be written as

$$\mathfrak{g}x\simeq\begin{bmatrix}\cos(\frac{r\pi}{2}) & -\sin(\frac{r\pi}{2}) & u\\ \sin(\frac{r\pi}{2}) & \cos(\frac{r\pi}{2}) & v\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}u'\\ v'\\ 1\end{bmatrix}$$

# 2.1 Equivariance

We can rewrite feature map `F` with shape `[c, w, h]` as a function from $\mathbb{Z}^2$ to $\mathbb{R}^c$:

$$F : \mathbb{Z}^2 \to \mathbb{R}^c$$
$$(x, y) \mapsto \begin{cases} \mathtt{F}[:, \mathtt{x}, \mathtt{y}], & x \in [0, w-1], y \in [0, h-1] \\ \mathbf{0}, & \text{otherwise} \end{cases}$$

For a signal $F$, we define group element $\mathfrak{g}$ acting on $F$ as

$$L_{\mathfrak{g}} f(x) = f(\mathfrak{g}^{-1} x),$$

in which $L_{\mathfrak{g}}$ is a materialization of transform $T_{\mathfrak{g}}$ corresponding to group element $\mathfrak{g}$, such that $L_{\mathfrak{g}} L_{\mathfrak{h}} = L_{\mathfrak{g}\mathfrak{h}}$.

# 2.2 Equivariance of Convolution

Consider ordinary spatial convolution $*$ and correlation $\star$. Let input feature map to be $f : \mathbb{Z}^2 \to \mathbb{R}^{K^{(l)}}$ and kernel to be $\psi^{(i)} : \mathbb{Z}^2 \to \mathbb{R}^{K^{(l)}}$, we have

$$[f * \psi](\boldsymbol{x}) = \sum_{\boldsymbol{y} \in \mathbb{Z}^2} \sum_{k=1}^{K^{(l)}} f_k(\boldsymbol{y}) \psi_k^{(i)}(\boldsymbol{x} - \boldsymbol{y}) \qquad \text{convolution}$$

$$[f \star \psi](\boldsymbol{x}) = \sum_{\boldsymbol{y} \in \mathbb{Z}^2} \sum_{k=1}^{K^{(l)}} f_k(\boldsymbol{y}) \psi_k^{(i)}(\boldsymbol{y} - \boldsymbol{x}) \qquad \text{correlation}$$

It is not difficult to verify that they are equivariant w.r.t. translations but not equivariant to rotations.

# 2.2 Equivariance of Convolution

To get a convolution or correlation layer which is equivariant w.r.t. both translation and rotation, or more broadly, to any finite group, consider the following construction by Cohen et al with the first layers be

$$f^{(1)}(\mathfrak{g}) = \left[f \star \psi^{(i)}\right](\mathfrak{g}) = \sum_{\boldsymbol{y}\in\mathbb{Z}^2} \sum_{k=1}^{K^{(l)}} f_k^{(0)}(\boldsymbol{y})\psi_k^{(0,i)}(\mathfrak{g}^{-1}\boldsymbol{y}),$$

$$f^{(l+1)}(\mathfrak{g}) = \left[f \star \psi^{(i)}\right](\mathfrak{g}) \quad = \sum_{\mathfrak{h}\in\mathfrak{G}} \sum_{k=1}^{K^{(l)}} f_k^{(l)}(\mathfrak{h})\psi_k^{(l,i)}(\mathfrak{g}^{-1}\mathfrak{h}),$$

in which $l \geqslant 1$ and $\mathfrak{g}, \mathfrak{h} \in \mathfrak{G}$; $f$ and $\psi^{(0,i)}$ are maps from $\mathbb{Z}^2$ to $\mathbb{R}$. The group correlation operator is equivariant w.r.t. group elements in $\mathfrak{G}$:

$$[[L_{\mathfrak{u}} f] \star \psi](\mathfrak{g}) = [L_{\mathfrak{u}}[f \star \psi]](g).$$

# 2.3 Other Necessary Equivariant Blocks and Operations

First we notice that linear combination and element-wise non-linearities preserves equivariance:

$$\left[L_{\mathfrak{g}}[af+bg]\right](\mathfrak{u}) = [af+bg]\left(\mathfrak{g}^{-1}\mathfrak{u}\right) = \left[aL_{\mathfrak{g}}f+bL_{\mathfrak{g}}g\right](\mathfrak{u})$$

$$L_{\mathfrak{g}}[\sigma\circ f](\mathfrak{u}) = [\sigma\circ f]\left(\mathfrak{g}^{-1}\mathfrak{u}\right) = \sigma\left[f\left(\mathfrak{g}^{-1}\mathfrak{u}\right)\right] = \left[\sigma\circ L_{\mathfrak{g}}\right](\mathfrak{u}).$$

in which $\mathfrak{g}\in\mathfrak{G}$ and $\sigma:\mathbb{R}\longrightarrow\mathbb{R}$.

Moreover, we can construct group version of pooling with subset $\mathfrak{U}$ acting as "neighborhood" with the following:

$$Pf(\mathfrak{g}) = \max_{\mathfrak{k}\in\mathfrak{g}\mathfrak{U}} f(\mathfrak{k})$$

which is equivariant.

# Outline

# 3. Rules Learning

We may use equivariant neural networks to learn dynamics of non-linear systems like cellular automata or other more sophisticated ones with translation, rotation and permutation symmetries or equivariances.

For systems that is difficult to sample across the whole state space, we could use the extrapolation or generalization ability to learn on the evolution trajectories with limited state or motif diversity.

Moreover, we can let the well-trained network to be the system simulator, to predict future system states, which is difficult or impossible to model using explicit code.

Thanks for Listening :)