Appendix A: Emission Feature Specifications

For: *Memory Without Storage, Learning Without a Learner* (Neale, 2026)

A.1 General Principles

Emission features are bulk observables computed from the system state at each timestep. They are chosen to be quantities that any instrument pointed at the system would naturally measure — densities, entropies, gradients, activity levels. They are not designed to detect M, L, or D; these measures are derived from the observer's predictive model applied to whatever emission features are available.

All emission features are normalised to comparable scales (typically [0, 1] or small positive reals) to prevent any single feature from dominating the prediction error computation.

The observer receives only the emission vector $E(t) \in \mathbb{R}^d$ at each timestep. It does not have access to the raw system state (the CA grid, the GoL grid, the concentration fields, or the array contents). The emission computation is performed once; the observer's predictive model operates entirely on the emission stream.

A.2 Elementary Cellular Automata

Simulation Parameters

Parameter	Value	Notes
Grid width	101 cells	Odd width avoids symmetric artifacts
Boundary conditions	Periodic (wraparound)	
Initial condition	Random binary, seed = 42	Reproducible via NumPy RandomState
Timesteps	300	
Rules simulated	All 256 (rules 0–255)	
Update rule	Standard Wolfram elementary CA	3-cell neighbourhood, 8 possible configurations

Emission Features (d = 4)

E_1: Density

$$d(t) = \frac{1}{W} \sum_{i=1}^{W} s_i(t)$$

Fraction of live (state = 1) cells. Range [0, 1].

E_2: Spatial entropy Shannon entropy of the distribution of 3-cell blocks. For each position i, compute the 3-bit neighbourhood pattern b_i = (s_{i}, s_{i+1}, s_{i+2}). Count the frequency of each of the 8 possible patterns. Compute:

$$H_s(t) = -\sum_{k=0}^{7} p_k \log_2 p_k$$

where p_k is the fraction of positions showing pattern k. Range [0, 3] bits. Captures spatial structure: low entropy = uniform or simple pattern; high entropy = complex or random.

E_3: Boundary density Normalised count of state transitions (0→1 or 1→0) along the row:

$$b(t) = \frac{1}{W-1} \sum_{i=1}^{W-1} |s_i(t) - s_{i+1}(t)|$$

Range [0, 1]. Captures spatial fragmentation: 0 = uniform; 1 = maximally alternating.

E_4: Activity Fraction of cells that changed between timestep t−1 and t:

$$a(t) = \frac{1}{W} \sum_{i=1}^{W} 1[s_i(t) \neq s_i(t-1)]$$

Range [0, 1]. Zero at t = 0. Captures temporal dynamics: 0 = static; 1 = every cell changed.

A.3 Game of Life

Simulation Parameters

Parameter	Value	Notes
Grid size	50×50 (60×60 for Acorn)	Periodic boundaries
Pattern placement	Centre of grid	
Timesteps	300 (still lifes, oscillators, spaceships)	
	400 (methuselahs)	Extended for slow transients
Patterns simulated	18 total	See Table A.1

Table A.1: Game of Life patterns

Pattern	Type	Size	Period/Behaviour
Block	Still life	2×2	Static
Beehive	Still life	4×3	Static
Loaf	Still life	4×4	Static
Blinker	Oscillator	3×1	Period 2
Toad	Oscillator	4×2	Period 2
Beacon	Oscillator	4×4	Period 2
Pulsar	Oscillator	13×13	Period 3
Pentadecathlon	Oscillator	10×3	Period 15
Glider	Spaceship	3×3	Period 4, translates
LWSS	Spaceship	5×4	Period 4, translates
MWSS	Spaceship	6×5	Period 4, translates
HWSS	Spaceship	7×5	Period 4, translates
R-pentomino	Methuselah	3×3	Stabilises ~1100 steps
Acorn	Methuselah	7×3	Stabilises ~5200 steps
Diehard	Methuselah	8×3	Dies at step 130

Emission Features (d = 4)

E_1: Density

$$d(t) = \frac{1}{R \times C} \sum_{r,c} g_{r,c}(t)$$

Fraction of live cells. Range [0, 1].

E_2: Spatial entropy Shannon entropy of the distribution of 2×2 blocks. For each position (r, c), compute the 4-bit block pattern from the 2×2 neighbourhood. Count frequencies of all 16 possible patterns. Compute Shannon entropy. Range [0, 4] bits.

Note: 2×2 blocks are used rather than 3×3 for computational efficiency on the 2D grid. This provides 16 possible patterns versus 512 for 3×3, reducing noise in the entropy estimate for small active regions.

E_3: Boundary density Fraction of all grid cells that are live and have at least one dead neighbour (8-connectivity):

$$b(t) = \frac{1}{R \times C} \sum_{r,c} g_{r,c}(t) \cdot 1[\exists (r', c') \in N_8(r, c) : g_{r',c'}(t) = 0]$$

Range $[0, 1]$. Captures the perimeter-to-area ratio of the live cell population.

E_4: Activity Fraction of cells that changed between timesteps:

$$a(t) = \frac{1}{R \times C} \sum_{r,c} 1[g_{r,c}(t) \neq g_{r,c}(t-1)]$$

Range $[0, 1]$.

A.4 Gray-Scott Reaction-Diffusion

Simulation Parameters

Parameter	Value	Notes
Grid size	64×64	Periodic boundaries
Diffusion coefficients	D_u = 0.16, D_v = 0.08	Standard Gray-Scott values
Time step	dt = 1.0	
Laplacian	5-point stencil	$\nabla^2 f \approx f(i\pm1,j) + f(i,j\pm1) - 4f(i,j)$
Total simulation steps	5000	
Sampling interval	Every 10 steps	Yields 500 emission samples
Initial condition	U = 1, V = 0 everywhere, with central perturbation	

Parameter	Value	Notes
Central perturbation	Square of side gs/5 centred on grid	U = 0.5 + noise, V = 0.25 + noise
Noise	Gaussian, σ = 0.02, seed = 42	

Table A.2: Gray-Scott parameter regimes

Regime name	F	k	Qualitative behaviour
Uniform death	0.078	0.061	V decays to zero, U → 1
Stable spots	0.035	0.065	Turing instability → static spots
Labyrinthine stripes	0.040	0.065	Stripe pattern formation
Moving spots	0.014	0.054	Spots drift and collide
Mitosis (dividing spots)	0.028	0.062	Spots form and split
Pulsing	0.025	0.060	Oscillating pattern
Coral growth	0.062	0.061	Branching growth front
Spatiotemporal chaos	0.026	0.051	No stable pattern
Worm-like meandering	0.054	0.063	Elongated drifting structures
Sparse spots	0.030	0.062	Widely spaced stable spots

Emission Features (d = 6)

E_1: Mean U concentration

$$\bar{U}(t) = \frac{1}{N^2} \sum_{r,c} U_{r,c}(t)$$

Range $[0, 1]$. Tracks bulk substrate level.

E_2: Mean V concentration

$$\bar{V}(t) = \frac{1}{N^2} \sum_{r,c} V_{r,c}(t)$$

Range $[0, 1]$. Tracks bulk activator level.

E_3: Spatial variance of V

$$\sigma_V^2(t) = \frac{1}{N^2} \sum_{r,c} (V_{r,c}(t) - \bar{V}(t))^2$$

Captures how patterned the V field is. Zero = uniform; large = structured pattern.

E_4: Gradient energy Mean squared gradient of the V field:

$$G(t) = \frac{1}{N^2} \left(\sum_{r,c} (\Delta_x V)^2 + \sum_{r,c} (\Delta_y V)^2 \right)$$

where Δ_x and Δ_y are first-difference operators along each axis. Captures sharpness of pattern features.

E_5: Activity Mean absolute change in V between sampled timesteps:

$$a(t) = \frac{1}{N^2} \sum_{r,c} |V_{r,c}(t) - V_{r,c}(t-1)|$$

Captures rate of change. Note: this measures change between *sampled* frames (every 10 simulation steps), not between consecutive simulation steps.

E_6: Pattern entropyV field discretised into 8 bins ([0, 0.125), [0.125, 0.25), ..., [0.875, 1.0]). Shannon entropy of the bin distribution:

$$H_p(t) = -\sum_{k=0}^{7} p_k \log_2 p_k$$

Range $[0, 3]$ bits. Low = V concentrated in few bins (uniform or bimodal); high = spread across bins.

A.5 Sorting Algorithms

Simulation Parameters

Parameter	Value	Notes
Array size	30 elements	Values 0–29, shuffled
Initial permutation	NumPy RandomState seed = 123	Reproducible
Variants	6	See Table A.3

Table A.3: Sorting algorithm variants

Variant	Description	Deterministic?	Steps
Already sorted	Control — array [0,1,...,29] repeated 20 times	Yes	20
Bubble sort	Standard top-down bubble sort	Yes	28
Self-sorting bubble	Elements visited in random order each pass; each decides locally whether to swap	No (random visit order, seed = 42)	20
Self-sorting with defects	As above, but elements at positions 7, 15, 22 are frozen	No	6
Selection sort	Standard selection sort	Yes	31
Reverse bubble	Bubble sort starting from [29, 28, ..., 0] (worst case)	Yes	31

Emission Features (d = 5)

E_1: Sortedness Fraction of adjacent pairs in non-decreasing order:

$$\text{sort}(t) = \frac{1}{n-1}\sum_{i=1}^{n-1} 1[a_i(t) \leq a_{i+1}(t)]$$

Range $[0, 1]$. Zero = perfectly reverse-sorted; 1 = perfectly sorted.

E_2: Mean displacement Average absolute distance of each element from its sorted position, normalised by array size:

$$\mathrm{disp}(t) = \frac{1}{n^2} \sum_{i=1}^{n} |i - \mathrm{sorted\_position}(a_i(t))|$$

Range [0, ~0.5]. Zero = every element in its correct position.

E_3: Inversion fraction Fraction of all element pairs (i, j) with i < j where a_i > a_j:

$$\mathrm{inv}(t) = \frac{2}{n(n-1)} \sum_{i<j} 1[a_i(t) > a_j(t)]$$

Range [0, 1]. Zero = sorted; 1 = reverse-sorted; ~0.5 = random permutation.

E_4: Activity Fraction of elements that occupy a different position than in the previous step:

$$a(t) = \frac{1}{n} \sum_{i=1}^{n} 1[a_i(t) \neq a_i(t-1)]$$

Range [0, 1].

E_5: Block entropy Array divided into blocks of size n/10 (= 3 elements). Mean value of each block computed. These means normalised to a probability distribution. Shannon entropy of that distribution. Captures large-scale ordering structure. Range depends on number of blocks.

A.6 Habituation Protocol Parameters

Parameter	Value	Notes
Base system	Gray-Scott, F=0.035, k=0.065	Stable spots regime
Growth period	3000 steps before first tap	Pattern fully formed
Tap location	Centre of highest-V cluster	Determined automatically
Tap strength	ΔV = +0.15, ΔU = −0.15	Applied within tap radius
Tap radius	3 cells	Circular region
Relaxation period	50 steps after each tap	Before measuring response
Response measure	RMSE of emission change	pre-tap vs post-relaxation

Parameter	Value	Notes
Taps per condition	12	
Tap intervals tested	30, 50, 80, 120, 200, 400, 800 steps	Frequency sweep

A.7 Multi-Observer Protocol Parameters

Parameter	Value
Systems tested	CA Rule 54, CA Rule 30, GoL R-pentomino, GS Mitosis, Bubble Sort
Full access	All d emission features
Pair access	All (d choose 2) pairs, results averaged
Single access	Each feature alone, results averaged
Feature decomposition	L computed for each individual feature separately

A.8 Software and Reproducibility

All simulations implemented in Python 3.12 using NumPy (no other dependencies required for core analysis). SciPy used only for Spearman correlation in the dual-memory comparison. Random seeds specified for all stochastic components. Complete source code available at goleudy.ai.

No pre-existing datasets were used. All emission streams were generated from first principles in scripts written for this paper.