

The Spiking Tolman-Eichenbaum Machine: Emergent Geometry of Space through Spiking Network Dynamics

Daisuke Kawahara, Shigeyoshi Fujisawa

RIKEN Center for Brain Science, Saitama, Japan

Department of Complexity Science and Engineering, The University of Tokyo, Tokyo,
Japan

Table of Contents

1	Introduction	2
2	Results	3
2.1	Spatial navigation in a 2D environment	3
2.2	Spatial navigation in a linear track	3
2.3	Analysis of synaptic inputs to grid cells and place cells	9
3	Methods	9
3.1	Neuron model	9
3.2	Network architecture	12
3.3	Training procedure	15
3.4	Parameter	17
4	Discussion	18
5	Supplementary Material	20

Abstract

The hippocampal–entorhinal system supports spatial navigation and memory by orchestrating the interaction between grid cells and place cells. While various models have reproduced these patterns, many rely on predefined connectivity or fixed weights and lack mechanisms for learning or biologically realistic temporal dynamics. Here, we introduce the Spiking Tolman–Eichenbaum Machine (Spiking TEM)—a spiking neural network model that extends the original rate-based TEM to incorporate spike-based computation. Our model learns grid-like codes in the entorhinal module and context-specific place codes in the hippocampal module. These results demonstrate that structured spatial representations can emerge from biologically plausible spike-based learning and dynamics, offering a framework for understanding how the brain may encode space through time.

1 Introduction

The hippocampal–entorhinal circuit enables mammals to form structured internal representations of space, facilitating both navigation and memory. Within this circuit, grid cells in the medial entorhinal cortex (MEC) exhibit periodic spatial firing [1], while place cells in the hippocampus encode location-specific activity [2]. Understanding how these spatial codes emerge from neural dynamics remains a central challenge in neuroscience.

Several computational models have been proposed to explain the emergence of grid cells. These include oscillatory interference models [3], which rely on interference between theta-modulated inputs; continuous attractor models [4][5][6], which stabilize grid-like activity patterns via recurrent connectivity and update them based on self-motion cues such as velocity. While these frameworks have offered valuable insights, many rely on architectural assumptions—such as hand-crafted architectures or idealized input signals—that limit their generality.

The Tolman–Eichenbaum Machine (TEM) [7] introduced a new perspective by linking spatial coding in MEC with relational memory in the hippocampus. Unlike many prior models with hand-crafted weights, TEM explains how grid-like codes can emerge as part of a general-purpose structure learning system, unifying spatial and non-spatial memory under a common computational framework. However, as a rate-based model, TEM does not incorporate temporally precise neural activity or biologically realistic mechanisms such as spike timing, synaptic dynamics, and oscillatory modulation. These temporal features are prominent in the hippocampal–entorhinal system and play a crucial role in shaping spatial codes.

To address these limitations, we propose the Spiking Tolman–Eichenbaum Machine (Spiking TEM)—a biologically grounded extension of TEM that incorporates spiking neurons, spike-timing-dependent plasticity (STDP) [8][9], and theta-modulated input [10] into its architecture. By implementing the core computational principles of TEM in a spiking neural network (SNN), we investigate whether structured spatial codes can emerge within a biologically grounded framework.

Our results demonstrate that grid-like firing patterns naturally arise in the entorhinal module of Spiking TEM alongside context-dependent place representations in the hippocampal module. These spatial codes develop through learning and are not imposed by symmetry or handcrafted weights.

Furthermore, by analyzing the learned synaptic weights after training, we identify the specific structure of inputs that drive grid cell responses. This provides mechanistic insight into how spatially periodic patterns emerge through learning.

In sum, Spiking TEM bridges abstract relational modeling with spike-based neural computation, offering a unified and mechanistic account of how structured spatial codes—including both grid and place cells—can emerge through learning.

2 Results

2.1 Spatial navigation in a 2D environment

2.1.1 Emergence of grid cells and place cells

To evaluate whether the Spiking Tolman–Eichenbaum Machine (SpikingTEM) can replicate characteristic spatial firing patterns observed in the hippocampal–entorhinal circuit, we analyzed the neural activities in the entorhinal and hippocampal modules after unsupervised learning in a spatial navigation task.

We trained the model in an 8×8 square arena, where a virtual agent performed random exploration. At each timestep, the agent selected one of five discrete actions: up, down, left, right, or stay. The number of episodes is 10,000, corresponding to a sequence of movements across the environment. At each step, the agent received a one-hot observation corresponding to its position on a 8×8 square arena. No contextual inputs or reward signals were provided; learning was entirely unsupervised. Spiking activity was simulated for T time bins per step, and learning proceeded in an unsupervised manner via backpropagation and STDP. The model was trained according to Algorithm 2, with main hyperparameters listed in Table 1.

After training, we evaluated the spatial tuning properties of neurons. To this end, the agent performed an additional 2,000 steps of a random walk in the same 8×8 environment, without any further learning. The spiking activity of neurons during this evaluation phase was recorded and used to compute the spatial firing rate maps. The spatial firing fields in the entorhinal cortex module are shown in Figure 1. The autocorrelation maps of their spatial firing fields are presented in Figure 2. As reported in experimental studies [1], grid cells exhibit different grid scales, and this phenomenon also emerged in our model. For example, neuron #108 displayed larger grid scales, while neuron #200 exhibited a smaller-scale periodicity (see Figure 1).

The spatial firing fields in the hippocampal module after training are shown in Figure 3. Most neurons exhibit localized, place-specific firing patterns, consistent with the observation that the majority of CA1 neurons in the biological hippocampus function as place cells [11].

2.1.2 Remapping

Experimental studies have shown that neurons in the hippocampal–entorhinal system reorganize their spatial representations in response to contextual changes. Specifically, grid cells in the entorhinal cortex exhibited global realignment of their grid-like firing patterns across contexts. That is, the grid fields maintained their periodic structure, but the phase and orientation shifted between input conditions [12]—while hippocampal place cells undergo global remapping, where their place fields changed location, disappeared, or re-emerged in different contexts [13][14].

To test whether our model reproduces this context-dependent spatial coding, we altered the input observation vectors from one-hot to two-hot vectors after training, while keeping the environment’s geometry and model parameters fixed.

Under this condition, neurons in the entorhinal module preserved grid-like periodicity but exhibited realignment across contexts, with their spatial firing fields globally shifting in phase (see Figure 4). In contrast, neurons in the hippocampal module exhibited global remapping, where place fields relocated, disappeared, or re-emerged under the new input condition (see Figure 5).

2.2 Spatial navigation in a linear track

To evaluate whether the model forms spatially selective representations in a linear track setting, we tested it in a one-dimensional linear track environment. The agent was trained to move back and forth along a track consisting of 20 discrete positions, with each position divided into $T = 3$ temporal bins, as in the two-dimensional experiments. At each time step, the agent moved forward by one grid unit and reversed at each end. No explicit goals or external cues

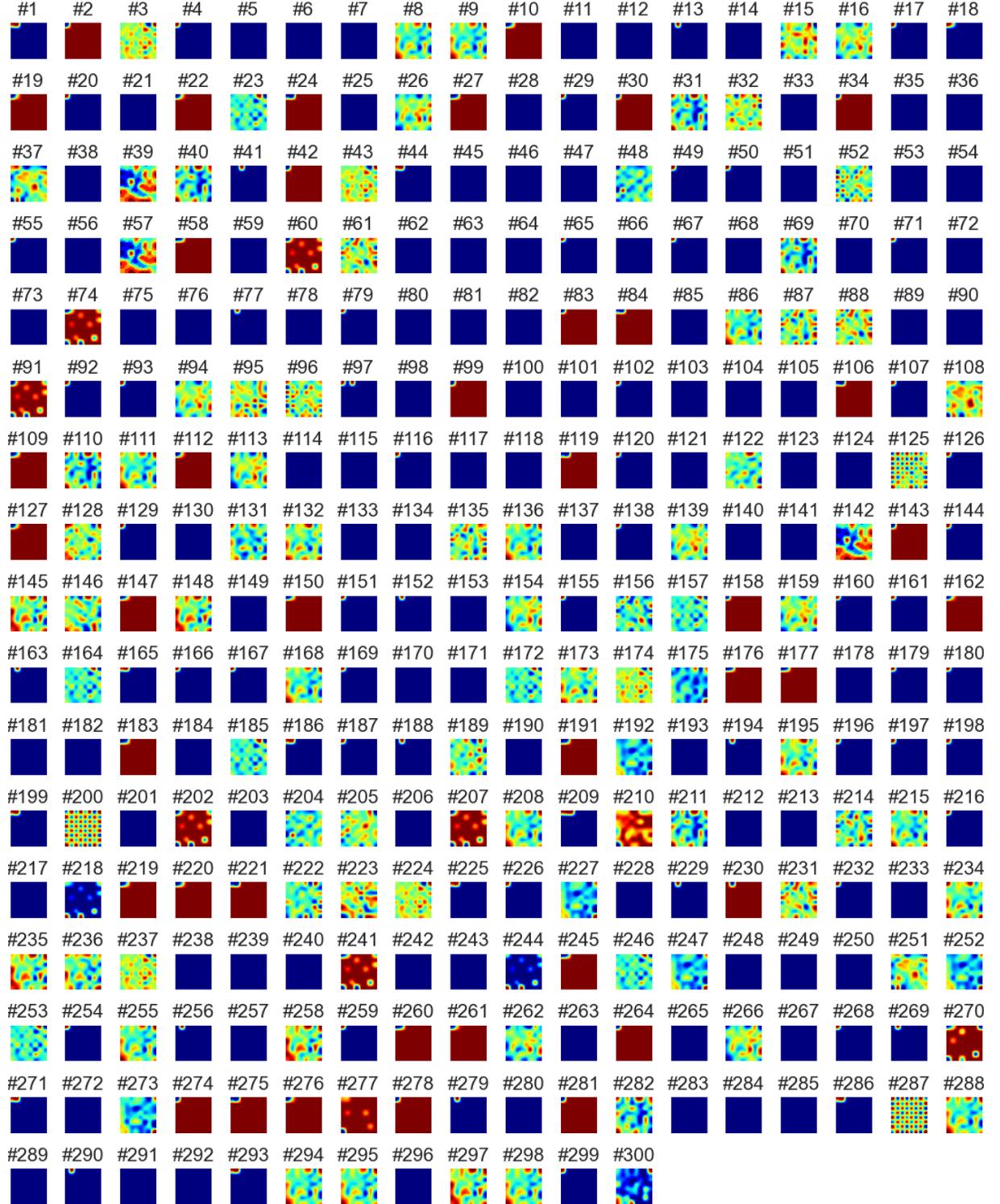


Figure 1 Spatial firing fields of neurons in the entorhinal cortex module. Each panel displays the spatial firing field of a neuron in the entorhinal cortex module after training. Some neurons exhibit periodic, grid-like activity (#139, #100), while others show irregular or localized patterns. Firing rates were computed by averaging spiking activity over 2,000 steps of random exploration.

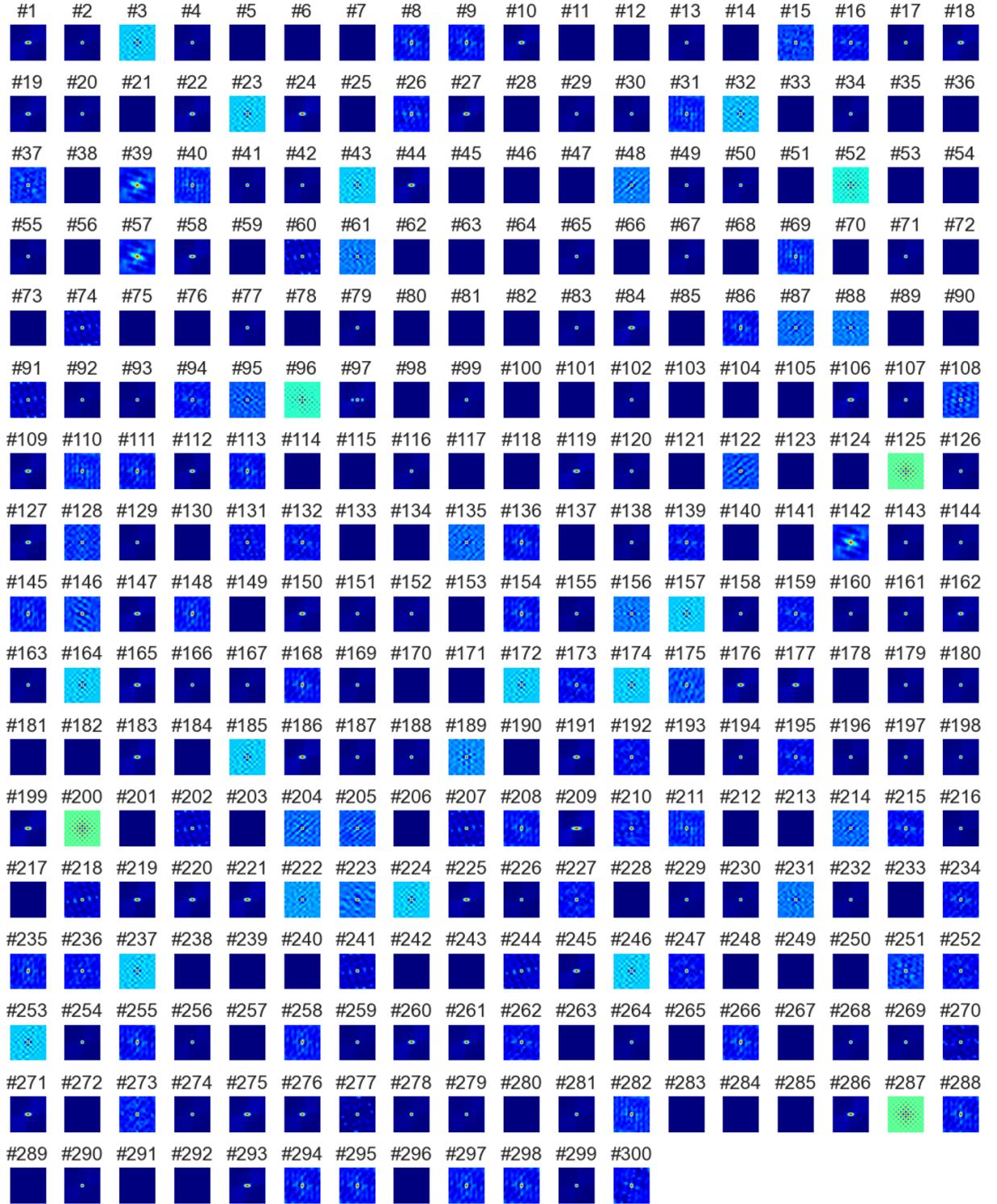


Figure 2 Autocorrelation maps of spatial firing fields in the entorhinal cortex module.
 Each panel shows the spatial autocorrelation of a neuron's firing field after training. Neurons with grid-like activity exhibit characteristic periodic symmetry in their autocorrelation maps (#125, #200, #287), whereas others display less regular or asymmetric patterns.

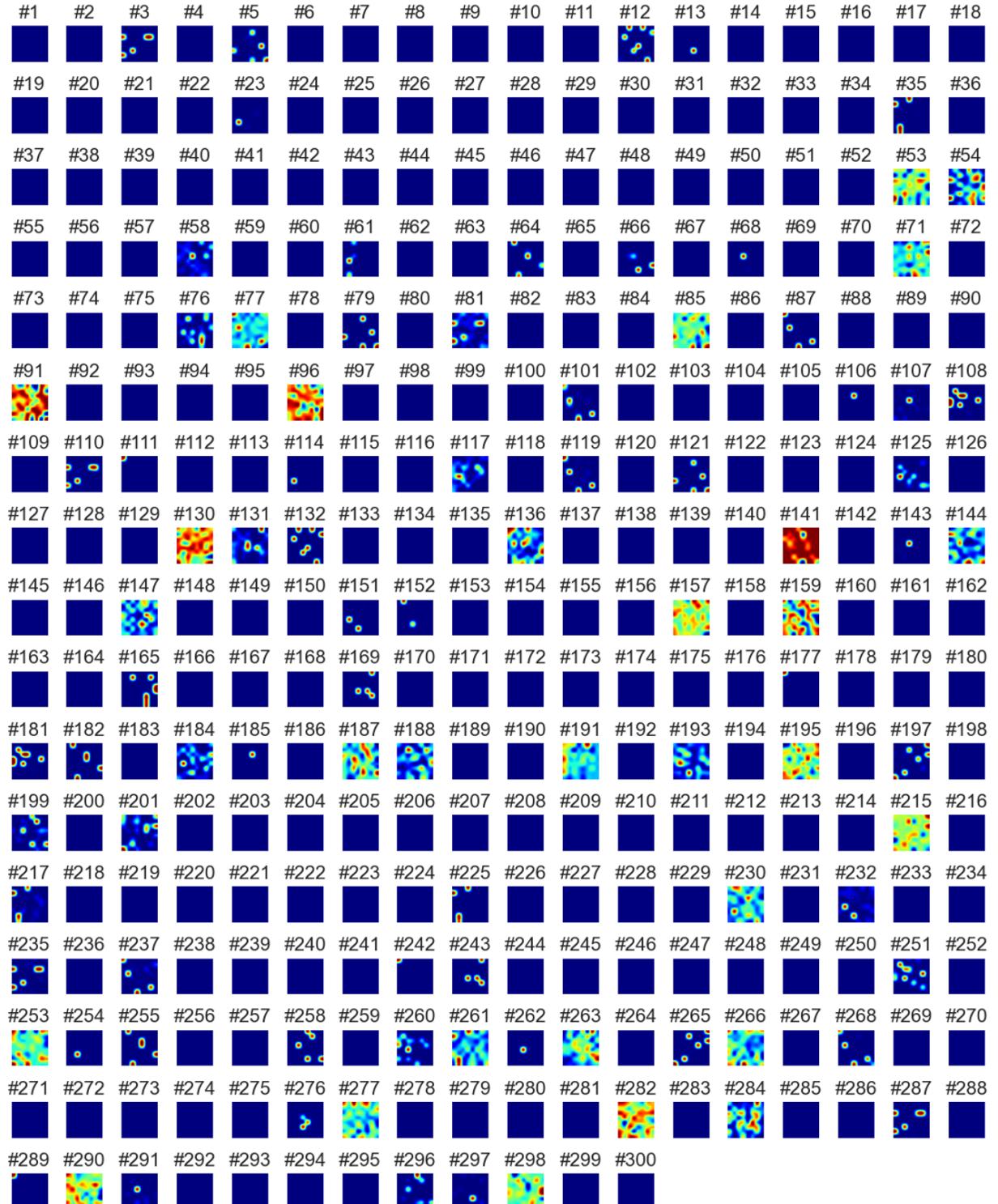


Figure 3 Spatial firing fields of all neurons in the hippocampus module. Each panel shows the spatial firing activity of a neuron in the hippocampus module after training. Firing rates were computed over 2,000 steps of random exploration in the environment.

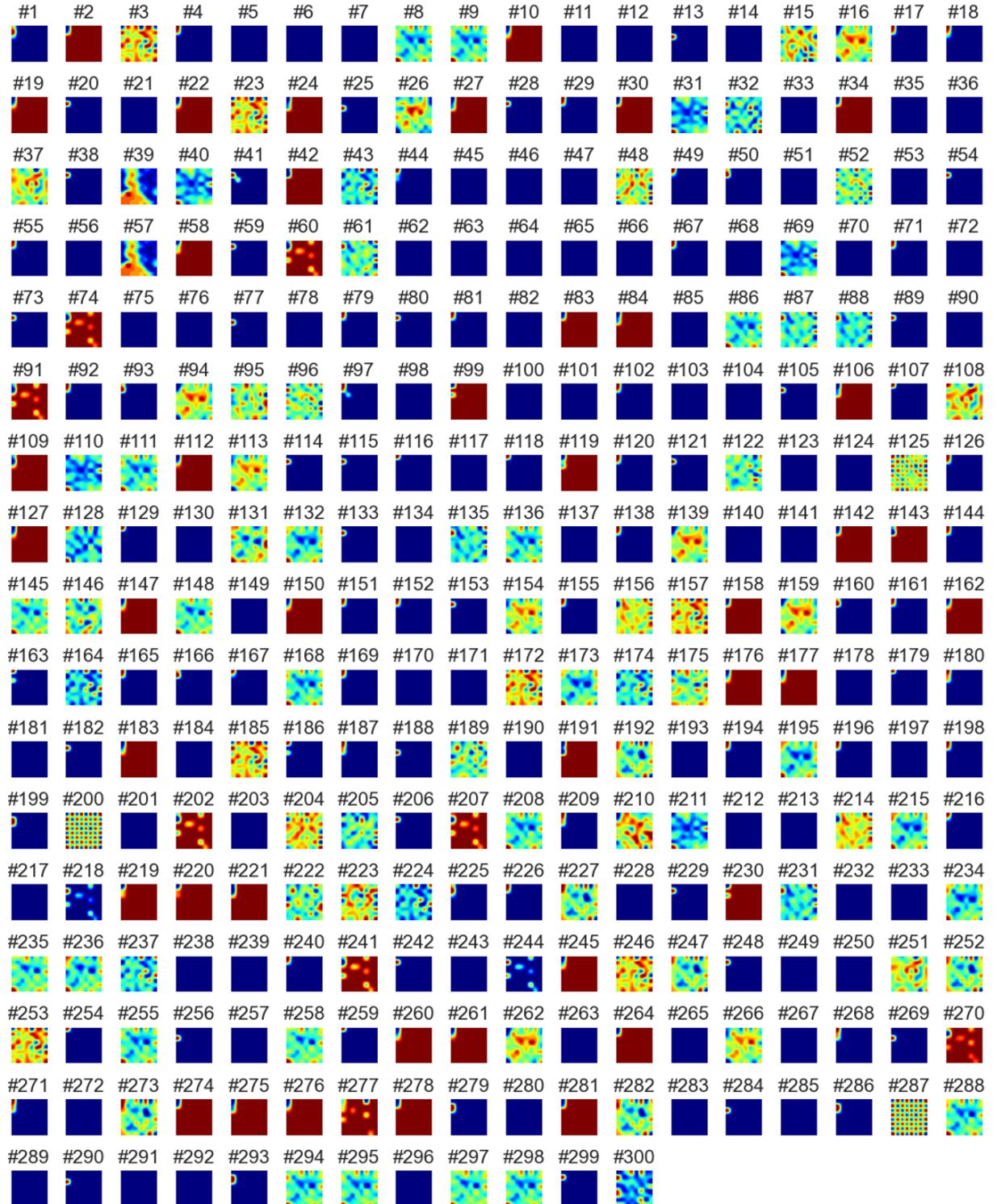


Figure 4 Realignment of grid cell spatial firing patterns in the entorhinal cortex module across different environments. In particular, neuron #287 exhibited a clear realignment of its grid pattern relative to the pattern in Figure 1.

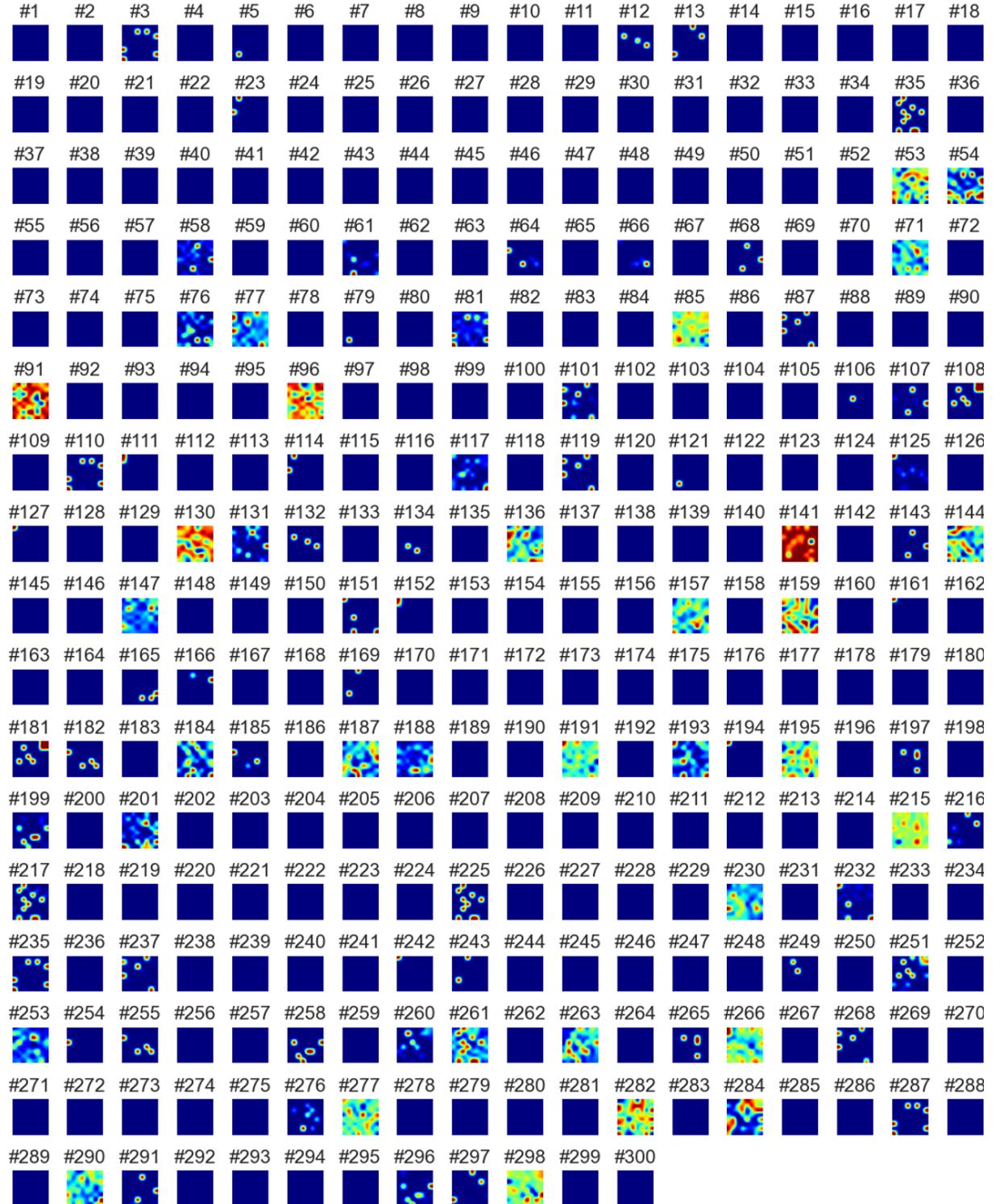


Figure 5 Remapping of spatial firing patterns in hippocampal neurons across different environments. A larger number of hippocampal neurons exhibited global remapping relative to the pattern in Figure 3.

were provided—learning relied solely on position-based inputs and synaptic plasticity.

After training, we evaluated the spatial firing patterns of neurons. Neurons in the entorhinal module developed grid-like firing with periodic activation along the track (Figure 6), while neurons in the hippocampal module formed sharply localized place fields (Figure 7).

2.3 Analysis of synaptic inputs to grid cells and place cells

To gain insights into the mechanisms underlying spatial selectivity, we analyzed the synaptic inputs received by grid cells in the entorhinal cortex and place cells in the hippocampus. Specifically, we examined how the spatial tuning of these cells emerged from their input patterns.

We focused on a subset of neurons that exhibited clear grid-like or place-selective firing patterns after training.

For grid cells, we visualized the weights from the preceding intermediate layer in the entorhinal cortex (Figure 8). For each input neuron, its place field is shown, and the value above each field indicates the corresponding synaptic weights.

For place cells, we plotted the weights from both the grid module (\mathbf{g}) and the sensory inputs ($x2p$) to a hippocampal neuron (Figure 9).

In all cases, the synaptic inputs were sorted in descending order of absolute weight magnitude, making it easy to identify the most influential inputs shaping the spatial responses of the target neuron.

3 Methods

The source code used in the experiment is available in <https://github.com/kdaisuke0203/spikingTEM>.

3.1 Neuron model

We model each neuron using the Leaky Integrate-and-Fire (LIF) [15]. The membrane potential $V(t)$ of a neuron evolves according to the following differential equation:

$$\tau_m \frac{dV(t)}{dt} = -(V(t) - V_{\text{rest}}) + I(t) \quad (1)$$

where:

- τ_m is the membrane time constant,
- V_{rest} is the resting membrane potential,
- $I(t)$ is the total synaptic input current.

When the membrane potential reaches a predefined threshold V_{th} , the neuron emits a spike and the membrane potential is reset:

$$\text{if } V(t) \geq V_{\text{th}}, \text{ then } V(t) \rightarrow V_{\text{reset}} \quad (2)$$

The spiking output of each neuron at discrete time t is represented by a binary variable $s(t)$:

$$s(t) = \begin{cases} 1, & \text{if } V[t] \geq V_{\text{th}} \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

The input current $I(t)$ to each neuron is composed of synaptic contributions from presynaptic neurons:

$$I(t) = \sum_j w_{ij} s_j(t) \quad (4)$$

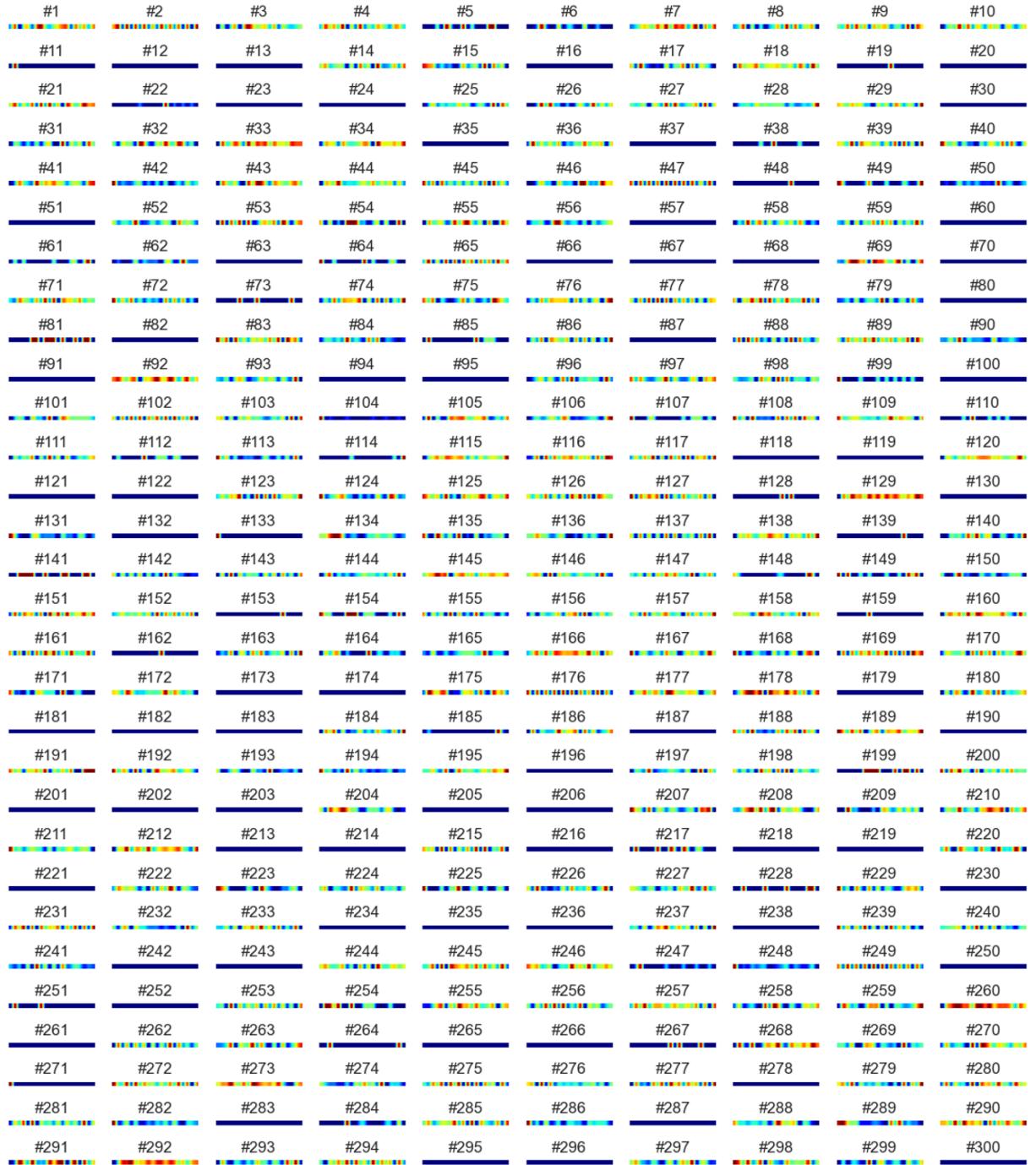


Figure 6 Spatial firing fields of entorhinal cortex neurons on a linear track. Each panel shows the firing rate map of a neuron in the entorhinal cortex module recorded during 20 laps of a linear track. Neurons exhibit diverse spatial tuning profiles, including periodic firing fields along the track.

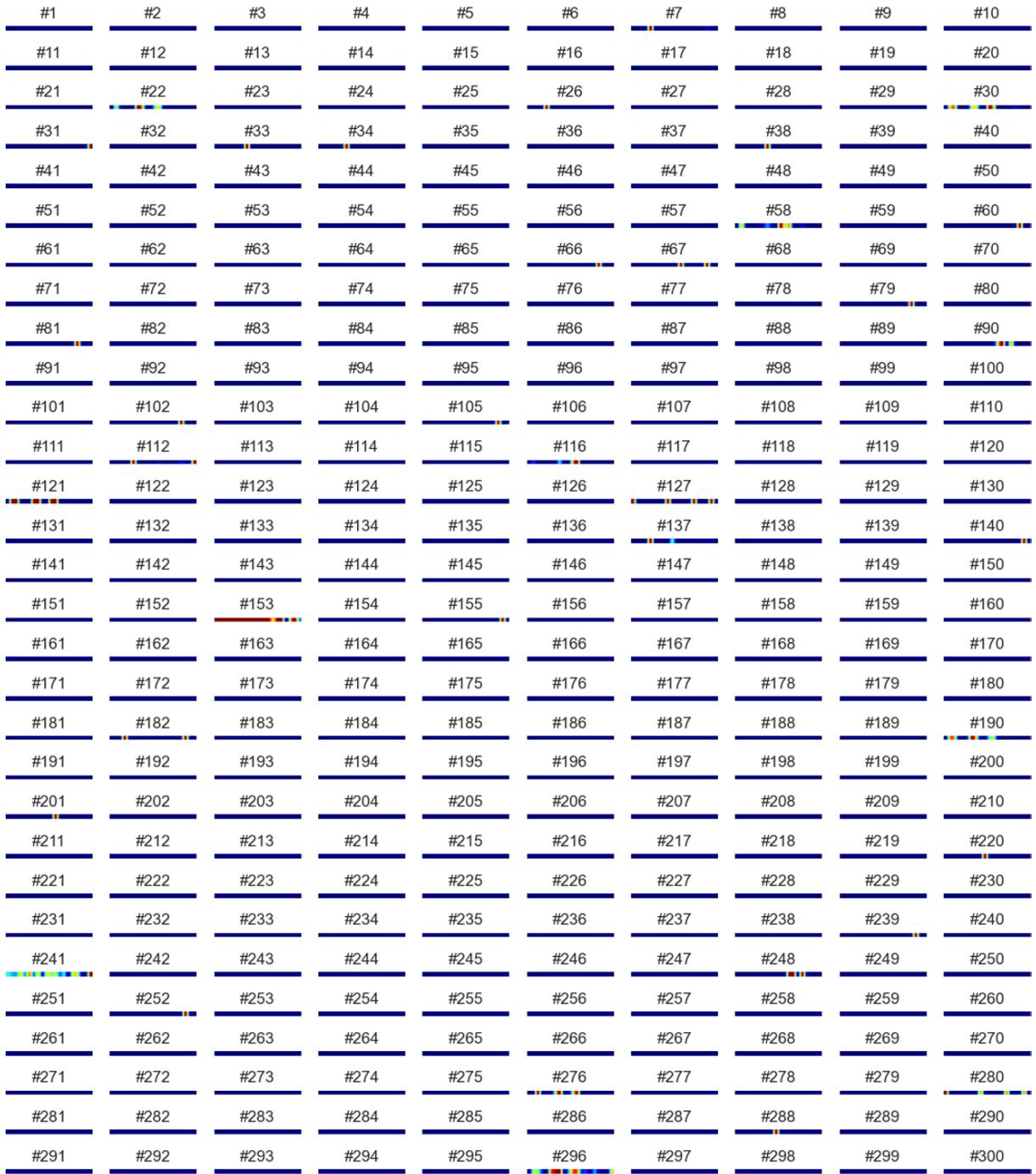


Figure 7 Spatial firing fields of hippocampal neurons on a linear track. Each panel shows the firing rate map of a neuron in the hippocampal module recorded during 20 laps of a linear track.

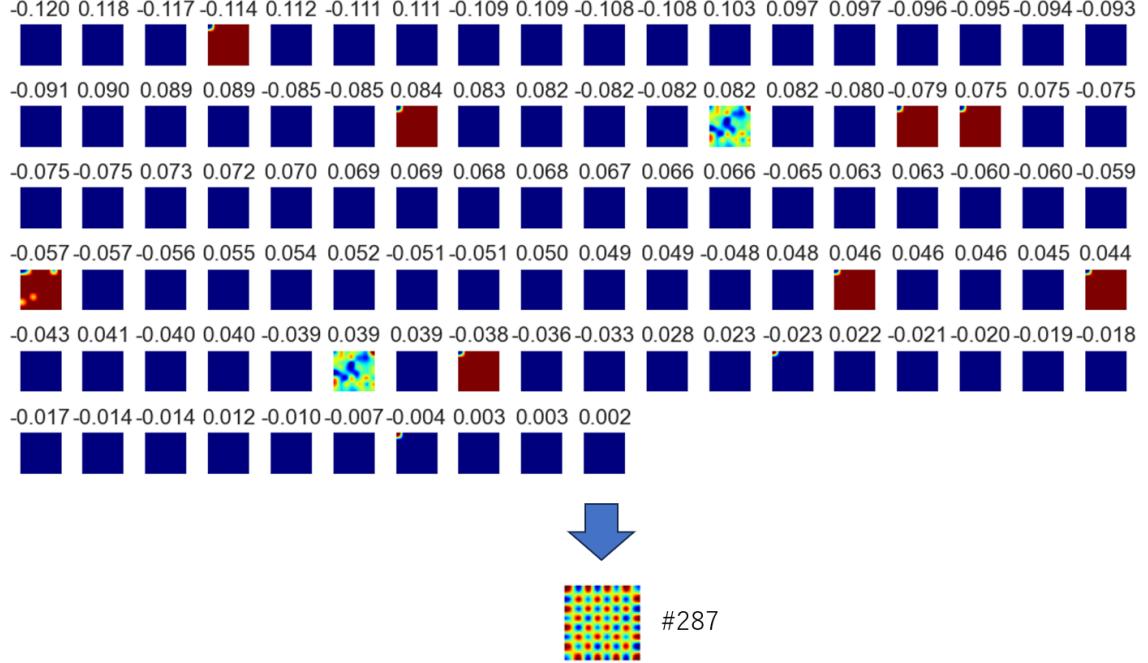


Figure 8 Synaptic inputs to a representative grid cell in the entorhinal cortex module.
Each panel displays the spatial firing fields of presynaptic neurons that provide the inputs to a selected grid cell. Inputs are sorted in descending order of synaptic weight magnitude, with the weight values shown above each map.

where:

- w_{ij} is the synaptic weight from neuron j to neuron i ,
- $s_j(t) \in 0, 1$ represents the spike at time t from neuron j .

Also, a theta oscillation was injected as a modulatory input to all neurons in both the entorhinal and hippocampal modules as follows:

$$I_\theta(t) = \alpha_\theta \cdot \sin 2\pi f_\theta t \quad (5)$$

where α_θ is the amplitude, $f_\theta = 8$ Hz is the theta frequency.

3.2 Network architecture

Our model comprises two components: a generative model that predicts future sensory observations from internal latent states and actions, and an inference model that infers internal states from current sensory observations (Figure 10), similar to the structure of the Tolman–Eichenbaum Machine (TEM)[7].

3.2.1 Generative model

In the generative model, the latent state \mathbf{g}_t , representing grid cell activity in the entorhinal cortex, is updated based on the previous state \mathbf{g}_{t-1} and the action \mathbf{a}_{t-1} . The updated latent state \mathbf{g}_t is used to generate the corresponding place cell activity \mathbf{p}_t , which is then decoded to reconstruct the sensory input \mathbf{x}_t . Formally, the generative model defines the following

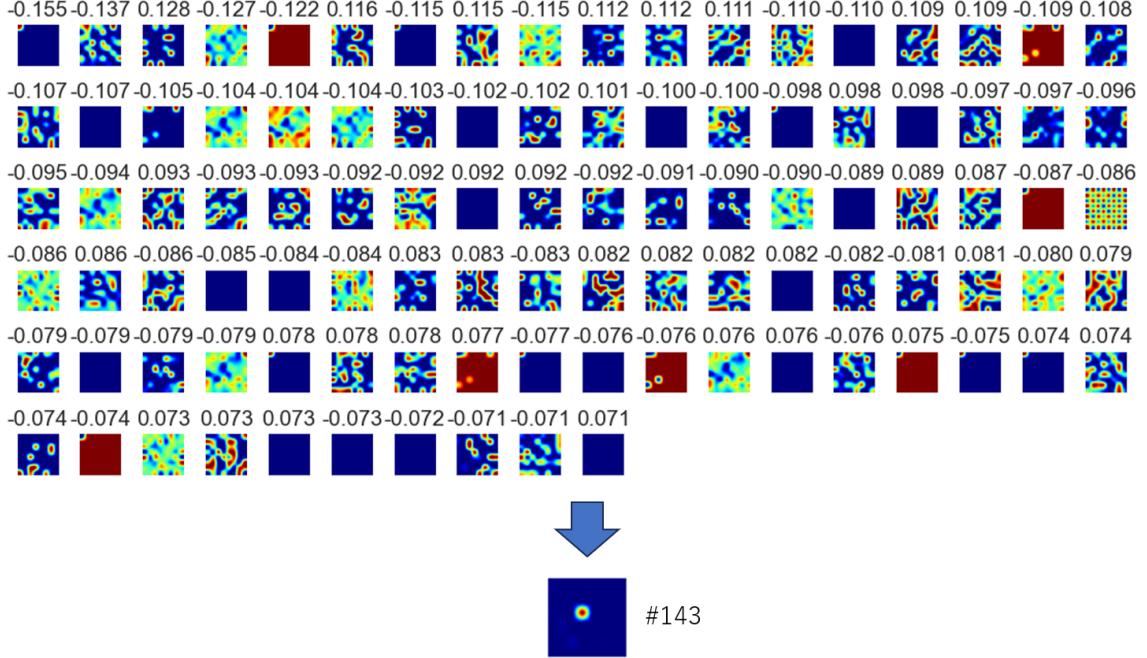


Figure 9 Synaptic inputs to a representative place cell in the hippocampus module. Each panel shows the spatial firing fields of presynaptic neurons that project to a selected place cell. These inputs originate from both the entorhinal grid module and a population of preprocessed sensory inputs, each comprising 300 neurons (600 neurons in total). For clarity, only the top 100 presynaptic neurons with the strongest synaptic weights are shown.

distributions:

$$\mathbf{g}_t \sim p(\mathbf{g}_t | \mathbf{g}_{t-1}, \mathbf{a}_{t-1}), \quad (6)$$

$$p_t \sim p(\mathbf{p}_t | \mathbf{g}_t, M_{t-1}), \quad (7)$$

$$x_t \sim p(\mathbf{x}_t | \mathbf{p}_t). \quad (8)$$

Each conditional distribution is implemented by a spiking neural network (SNN) with hidden layers. For architectural and training details, see Algorithm 1 and Parameters 1. The memory module M , which supports the transformation from grid to place representations, is implemented with synapses undergoing spike-timing-dependent plasticity (STDP)[9][16] as follows:

$$\Delta w = \begin{cases} A_+ \exp\left(-\frac{\Delta t}{\tau_+}\right), & \text{if } \Delta t > 0 \\ -A_- \exp\left(\frac{\Delta t}{\tau_-}\right), & \text{if } \Delta t < 0 \end{cases} \quad (9)$$

where A_+ , A_- , τ_+ , and τ_- are the amplitudes and time constants for potentiation and depression, respectively. The weight change Δw depends on the time difference $\Delta t = t_{\text{post}} - t_{\text{pre}}$ between presynaptic and postsynaptic spikes.

3.2.2 Inference model

The inference model performs the reverse direction of the generative process: it infers the latent spatial representation \mathbf{g}_t based on the current observation \mathbf{x}_t , the previous action \mathbf{a}_{t-1} , and the previous internal state \mathbf{g}_{t-1} . To infer the place cell representation \mathbf{p}_t , we follow the core principle of the TEM [7], in which place cells emerge as conjunctive representations of sensory

inputs \mathbf{x} and grid codes \mathbf{g} . In our model, this conjunctive mechanism is implemented using a spiking neural network composed of Leaky Integrate-and-Fire (LIF) neurons. Specifically, the sensory input \mathbf{x}_t is first processed through a fully connected spiking layer, denoted as the pre-processing entorhinal layer. This layer extracts relevant features from the input. This processed signal \mathbf{x}'_t is then combined with the grid representation \mathbf{g}_t , which is directly projected to the place cell module \mathbf{p}_t . Both signals jointly determine the input current to the hippocampal neurons as follows:

$$I_t = W_{g,x' \rightarrow p} \cdot \mathbf{g}_t \odot \mathbf{x}'_t \quad (10)$$

where $W_{g,x' \rightarrow p}$ is the synaptic weight matrix from the processed sensory input \mathbf{x}' and grid code input \mathbf{g}_t to hippocampal neurons \mathbf{p} .

The inference model defines the following distributions:

$$g_t \sim q(\mathbf{g}_t | \mathbf{g}_{t-1}, \mathbf{a}_{t-1}, \mathbf{x}_t), \quad (11)$$

$$p_t \sim q(\mathbf{p}_t | \mathbf{g}_t, \mathbf{x}_t). \quad (12)$$

$$(13)$$

Each conditional distribution Eq. (11), (12) is implemented by an SNN with hidden layers.

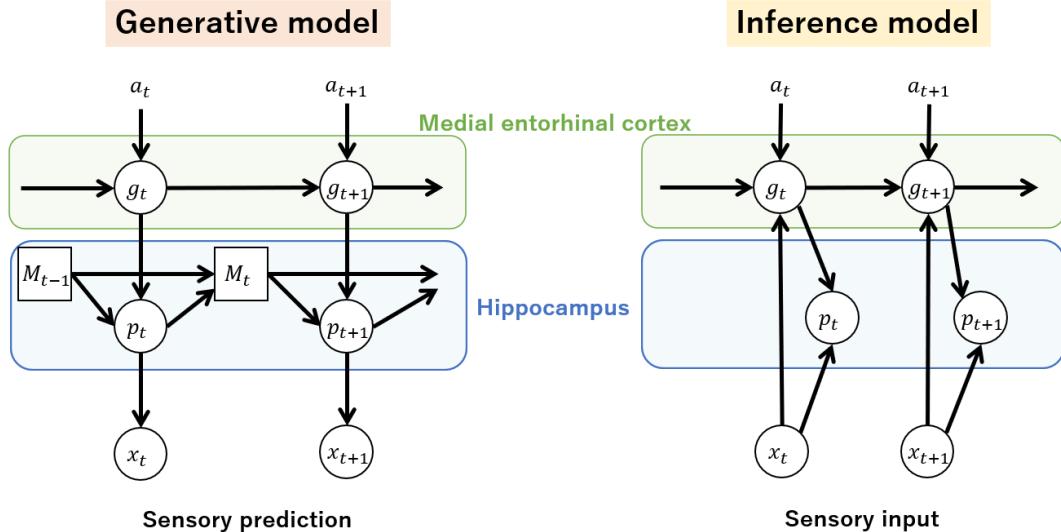


Figure 10 **Architecture of the Spiking Tolman–Eichenbaum Machine (Spiking TEM).**
The model comprises two main components: a generative model (left) and an inference model (right). All computations are implemented using spiking neural networks (SNNs).

3.2.3 Autoregressive Bernoulli Spike Sampling

In the original Tolman–Eichenbaum Machine (TEM), the grid representation \mathbf{g} and the place representation \mathbf{p} were treated as continuous latent variables sampled from Gaussian distributions, enabling differentiable inference in a variational framework. However, in our spiking implementation, both \mathbf{g} and \mathbf{p} are modeled as binary spike vectors. To generate these representations in a manner compatible with spiking networks, we adopted the autoregressive Bernoulli spike sampling method [17], originally developed for fully spiking variational autoencoders. This sampling framework enables the formulation of both the generative model and the inference model using discrete Bernoulli distributions over spike variables.

To construct the binary spike vectors, we first generate pre-activation values $\zeta_{q,t}$ and $\zeta_{p,t}$ in SNN. These pre-activations are overparameterized: the number of candidate neurons is expanded by a factor of k relative to the desired output dimension C , resulting in kC candidate

neurons. These are then divided into k non-overlapping groups, each containing C neurons. Sampling is performed by randomly selecting one element from each group:

$$\mathbf{z}_{q,t,c} = \text{random_select}(\zeta_{q,t}[k(c-1) : kc]) \quad (14)$$

$$\mathbf{z}_{p,t,c} = \text{random_select}(\zeta_{p,t}[k(c-1) : kc]) \quad (15)$$

By repeating this procedure for $1 \leq c \leq C$, we obtain binary spike vectors $\mathbf{z}_{q,t}, \mathbf{z}_{p,t} \in \{0, 1\}^C$.

This is equivalent to sampling from the following Bernoulli distributions, where the spike probability for each output neuron c is defined as the average of the k candidate values in its group:

$$\mathbf{z}_{q,t} \sim \text{Ber}(\boldsymbol{\pi}_{q,t}) \quad (16)$$

$$\mathbf{z}_{p,t} \sim \text{Ber}(\boldsymbol{\pi}_{p,t}) \quad (17)$$

$$\begin{cases} \boldsymbol{\pi}_{q,t,c} = \text{mean}(\zeta_{q,t}[k(c-1) : kc]) \\ \boldsymbol{\pi}_{p,t,c} = \text{mean}(\zeta_{p,t}[k(c-1) : kc]) \end{cases} \quad (18)$$

3.3 Training procedure

3.3.1 Algorithm

As in TEM, the algorithm is divided into two main phases: (1) Collect data phase, where episodes are executed to store sensory experiences and internal states into a buffer, and (2) Training phase, where the model parameters are updated using sampled batches from the collected episodes. The full procedure is described in Algorithm 1, 2.

Algorithm 1 Collect data phase in the SpikingTEM

```

1: Input: Initial latent state  $g_0$ , initial place cell activity  $p_0$ 
2: for each episode  $e = 1, \dots, E$  do
3:   for  $t = 1, 2, \dots, T$  do
4:     if  $t = T$  then
5:       Select action  $\mathbf{a}_{t-1} \in \{\text{stay, left, right, up, down}\}$ 
6:     else
7:        $\mathbf{a}_{t-1} \leftarrow \text{stay}$                                       $\triangleright$  No movement
8:     end if
9:      $\mathbf{g}_{gen,t}, V_{g2g,t} \leftarrow f_{g2g}(\mathbf{g}_{t-1}, \mathbf{a}_{t-1})$            $\triangleright$  State transition
10:     $\mathbf{x}'_t \leftarrow f_{x2g}(\mathbf{x}_t)$ 
11:     $\mathbf{g}_t, V_{g,t} \leftarrow f_{\text{infer\_g}}(\mathbf{x}'_t, \mathbf{g}_{gen,t})$            $\triangleright$  Inference of  $\mathbf{g}$ 
12:     $x2p_t \leftarrow f_{x2p}(\mathbf{x}_t)$ 
13:     $\mathbf{p}_t \leftarrow f_{\text{infer\_p}}(\mathbf{g}_t \cdot x2p_t)$                        $\triangleright$  Inference of  $\mathbf{p}$ 
14:     $\mathbf{p}_{gen,t} \leftarrow f_{\text{gen\_p}}(\mathbf{g}_{gen,t})$ 
15:     $M_t \leftarrow \text{update}(M_{t-1}, \mathbf{p}_{gen,t}, \mathbf{p}_t)$                    $\triangleright$  Associative memory by STDP
16:     $\mathbf{p}_{gen,t} \leftarrow M_t(\mathbf{p}_{gen,t})$                                  $\triangleright$  Retrieval by  $M$ 
17:     $V_{\bar{x},t} \leftarrow f_x(\mathbf{p}_t)$                                           $\triangleright$  Reconstructed membrane potential
18:     $V_{gen-x,t} \leftarrow f_x(\mathbf{p}_{gen,t})$ 
19:    Store:  $\{V_{x,t}, V_{\bar{x},t}, V_{gen-x,t}, V_{g2g,t}, V_{g,t}, \mathbf{g}_t, \mathbf{p}_t\}$  into buffer  $\mathcal{D}$ 
20:  end for
21: end for

```

Algorithm 2 Training phase in the SpikingTEM

```

1: for each batch  $b = 1, \dots, B$  do
2:   Sample batch  $\{V_{x,t}, V_{\bar{x},t}, V_{x',t}, V_{g2g,t}, V_{g,t}, \mathbf{g}_t, \mathbf{p}_t\}_{t=1}^T$  from buffer  $\mathcal{D}$ 
3:   Compute loss for batch  $b$ :

$$\mathcal{L}^{(b)} = \frac{1}{T} \sum_{t=1}^T (\lambda_{\text{loss1}} \|V_{x,t} - V_{\bar{x},t}\|^2 + \lambda_{\text{loss2}} \|V_{x,t} - V_{x',t}\|^2 + \|V_{g,t} - V_{g2g,t}\|^2 + \|\mathbf{g}_t\|^2 + \|\mathbf{p}_t\|^2)$$

4: end for
5: Compute total loss:  $\mathcal{L} = \frac{1}{B} \sum_{b=1}^B \mathcal{L}^{(b)}$ 
6: Update parameters:  $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$ 

```

The agent selects a new action (up, down, left, right, or stay) only every l steps and remains stationary during the intermediate steps. While it is possible to update the agent's position at every spiking time step, this would require a finer spatial discretization of the environment, resulting in significantly longer simulation times. Therefore, in this work, we simplify the setup by including stay periods between movements.

We used membrane potential prediction error instead of spike prediction error, as the latter failed to generate grid cells in our model. The use of membrane potential prediction error as a learning signal has also been explored in previous work [18], which demonstrated that synaptic

plasticity is driven by somatic membrane potential.

In the original TEM, the place cell representation \mathbf{p} was computed as a function of the element-wise product of \mathbf{g} and \mathbf{x} , i.e., $\mathbf{p}_t = \mathbf{f}(\mathbf{g}_t \odot \mathbf{x}_t)$. However, we found that simply providing \mathbf{g} and \mathbf{x} as separate inputs to the place cell module, such as $\mathbf{p}_t = \mathbf{f}(\mathbf{g}_t, \mathbf{x}_t)$, was also sufficient to produce grid cells and place cells (see Algorithm 1, line 13, Supplemental Figure 11, 12). This demonstrates that explicit conjunctive coding (e.g., $\mathbf{g}_t \odot \mathbf{x}_t$) is not necessary for the formation of place cell representations.

3.3.2 Surrogate gradient for backpropagation in SpikingTEM

Similar to the original Tolman–Eichenbaum Machine (TEM), we train the SpikingTEM using backpropagation to optimize model parameters. However, unlike the original TEM where all computations are differentiable, the use of spiking neurons in SpikingTEM introduces non-differentiable operations due to the discrete nature of spikes. To overcome this challenge, we adopt the surrogate gradient method [19], a standard approach in training spiking neural networks. Although the spiking function is non-differentiable, we approximate its gradient using a smooth surrogate during the backward pass, enabling gradient-based optimization. Let the spike generation be defined as: $s(v) = H(v - v_\theta)$ where $H(\cdot)$ is the Heaviside step function and v_θ is the firing threshold. Since $\frac{dH}{dv} = 0$ almost everywhere, direct backpropagation through $s(v)$ is not possible. Instead, during training, we use a surrogate derivative: $\frac{\partial s}{\partial v} \approx \tilde{\sigma}'(v)$ where $\tilde{\sigma}(v)$ is a smooth function:

$$\tilde{\sigma}(v) = \max \left(0, 1 - \left| \frac{v - v_\theta}{\gamma} \right| \right) \quad (19)$$

where γ is a scaling parameter controlling how broadly the surrogate gradient is spread around the threshold.

3.4 Parameter

We summarize the key parameters used in the results (Table 1).

Table 1 Parameters of the model

Parameter	Symbol	Value
Bernoulli sampling steps	k	2
Number of motor neuron	$\text{dim}(\mathbf{a})$	4
Number of hippocampal neurons	$\text{dim}(\mathbf{p})$	300
Number of entorhinal cortex neurons	$\text{dim}(\mathbf{g})$	300
Number of sensory neurons	$\text{dim}(\mathbf{x})$	30
Hidden units in $f_{\text{infer_g}}$	—	300
Hidden units in f_{p2x}	—	100
Hidden units in f_{g2g}	—	350
Hidden units in f_x	—	300
Training iterations	—	30,000
Batch size	B	30
Loss weight coefficient	λ_{loss1}	15.0
Loss weight coefficient	λ_{loss2}	15.0
Theta modulation input coefficient	α_θ	0.02
Threshold voltage	V_{thr}	0.2
Minimum membrane potential	V_{min}	-0.2
Maximum membrane potential	V_{max}	0.3
Membrane time constant	τ	1.0
Scaling parameter in surrogate gradient	γ	1.0
STDP potentiation	A_+	0.0005
STDP depression	A_-	-0.0005
STDP time constant	τ_{STDP}	0.5
STDP weight clipping range	W_{stdp}	(-2, 2)
Optimizer	—	Adam
Maximum learning rate	η_{max}	0.01
Minimum learning rate	η_{min}	0.0001

4 Discussion

Our spiking neural network implementation of the Tolman–Eichenbaum Machine (SpikingTEM) successfully reproduced key biological phenomena such as the emergence of grid cells and place cells.

We observed that a large proportion of neurons remained silent throughout the simulation. This is consistent with experimental findings showing that in a given environment, only about 40% of hippocampal pyramidal neurons are active, while the remaining 60% are silent cells [20]. Notably, in rate-based implementations of the TEM, silent cells rarely emerge, suggesting that the use of spiking dynamics contributes to the biological plausibility of the activity sparsity observed in our model.

Despite these promising results, several challenges remain for future work. First, to reduce computational cost, we simulated the network with an enlarged time resolution compared to the

actual timescale of biological spikes. Refining the simulation by using a finer temporal resolution would allow more accurate reproduction of temporal coding phenomena such as phase precession [21][22][23].

Second, the current model does not incorporate detailed anatomical circuitry—such as the layered organization of the entorhinal cortex (e.g., layers II, III, and V) or the differentiation of hippocampal subregions (e.g., CA1, CA3, and dentate gyrus)—which are likely critical for capturing structured temporal dynamics. For instance, grid cells in layer III (EC3) of the entorhinal cortex are known to exhibit phase locking [23], a phenomenon not yet reproduced in our model. Extending the architecture to reflect these biologically grounded pathways may be necessary to uncover the circuit mechanisms underlying such features.

Third, although grid-like firing fields emerged in our model, the diversity of grid cells was limited. In particular, the model failed to reproduce the range of grid scales observed in the entorhinal cortex [1], and did not exhibit variability in grid phases. This lack of diversity likely prevented the emergence of toroidal structure in the population activity of grid cells, which has been observed in biological data [24]. Addressing this limitation may require revisiting architectural components of the model, such as the network connectivity or learning rules, as well as tuning hyperparameters.

Fourth, the emergence of grid cells in our model appeared probabilistic rather than guaranteed. Clarifying the conditions that robustly lead to the formation of grid cells—such as initial weights, input statistics, and network architecture—will be necessary for stabilizing their emergence across training runs.

Finally, although we employed backpropagation for learning, it remains controversial whether such a mechanism is used in the brain [25]. More biologically plausible, local learning rules, such as predictive coding, may offer alternatives. While grid cells have been shown to emerge in rate-based predictive coding models [26], adapting such mechanisms to a spiking framework is an important step toward bridging theory and biology.

References

- [1] Torkel Hafting, Marianne Fyhn, Sturla Molden, May-Britt Moser, and Edvard I Moser. Microstructure of a spatial map in the entorhinal cortex. *Nature*, Vol. 436, No. 7052, pp. 801–806, 2005.
- [2] John O’keefe and Lynn Nadel. *The hippocampus as a cognitive map*. Oxford university press, 1978.
- [3] Neil Burgess, Caswell Barry, and John O’keefe. An oscillatory interference model of grid cell firing. *Hippocampus*, Vol. 17, No. 9, pp. 801–812, 2007.
- [4] Bruce L McNaughton, Francesco P Battaglia, Ole Jensen, Edvard I Moser, and May-Britt Moser. Path integration and the neural basis of the ‘cognitive map’. *Nature Reviews Neuroscience*, Vol. 7, No. 8, pp. 663–678, 2006.
- [5] Mark C Fuhs and David S Touretzky. A spin glass model of path integration in rat medial entorhinal cortex. *Journal of Neuroscience*, Vol. 26, No. 16, pp. 4266–4276, 2006.
- [6] Yoram Burak and Ila R Fiete. Accurate path integration in continuous attractor network models of grid cells. *PLoS computational biology*, Vol. 5, No. 2, p. e1000291, 2009.
- [7] James CR Whittington, Timothy H Muller, Shirley Mark, Guifen Chen, Caswell Barry, Neil Burgess, and Timothy EJ Behrens. The tolman-eichenbaum machine: unifying space and relational memory through generalization in the hippocampal formation. *Cell*, Vol. 183, No. 5, pp. 1249–1263, 2020.
- [8] Guo-qiang Bi and Mu-ming Poo. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of neuroscience*, Vol. 18, No. 24, pp. 10464–10472, 1998.
- [9] Wulfram Gerstner, Richard Kempter, J Leo Van Hemmen, and Hermann Wagner. A neuronal learning rule for sub-millisecond temporal coding. *Nature*, Vol. 383, No. 6595, pp. 76–78, 1996.
- [10] Mark P Brandon, Andrew R Bogaard, Christopher P Libby, Michael A Connerney, Kishan Gupta, and Michael E Hasselmo. Reduction of theta rhythm dissociates grid cell spatial

- periodicity from directional tuning. *Science*, Vol. 332, No. 6029, pp. 595–599, 2011.
- [11] Hannah S Wirtshafter and John F Disterhoft. In vivo multi-day calcium imaging of ca1 hippocampus in freely moving rats reveals a high preponderance of place cells with consistent place fields. *Journal of Neuroscience*, Vol. 42, No. 22, pp. 4538–4554, 2022.
 - [12] Marianne Fyhn, Torkel Hafting, Alessandro Treves, May-Britt Moser, and Edvard I Moser. Hippocampal remapping and grid realignment in entorhinal cortex. *Nature*, Vol. 446, No. 7132, pp. 190–194, 2007.
 - [13] Robert U Muller and John L Kubie. The effects of changes in the environment on the spatial firing of hippocampal complex-spike cells. *Journal of Neuroscience*, Vol. 7, No. 7, pp. 1951–1968, 1987.
 - [14] Stefan Leutgeb, Jill K Leutgeb, Carol A Barnes, Edvard I Moser, Bruce L McNaughton, and May-Britt Moser. Independent codes for spatial and episodic memory in hippocampal neuronal ensembles. *Science*, Vol. 309, No. 5734, pp. 619–623, 2005.
 - [15] Anthony N Burkitt. A review of the integrate-and-fire neuron model: I. homogeneous synaptic input. *Biological cybernetics*, Vol. 95, No. 1, pp. 1–19, 2006.
 - [16] Abigail Morrison, Markus Diesmann, and Wulfram Gerstner. Phenomenological models of synaptic plasticity based on spike timing. *Biological cybernetics*, Vol. 98, No. 6, pp. 459–478, 2008.
 - [17] Hiromichi Kamata, Yusuke Mukuta, and Tatsuya Harada. Fully spiking variational autoencoder. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 36, pp. 7059–7067, 2022.
 - [18] Robert Urbanczik and Walter Senn. Learning by the dendritic prediction of somatic spiking. *Neuron*, Vol. 81, No. 3, pp. 521–528, 2014.
 - [19] Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, Vol. 36, No. 6, pp. 51–63, 2019.
 - [20] Lucien Tres Thompson and PJ Best. Place cells and silent cells in the hippocampus of freely-behaving rats. *Journal of Neuroscience*, Vol. 9, No. 7, pp. 2382–2390, 1989.
 - [21] John O’Keefe and Michael L Recce. Phase relationship between hippocampal place units and the eeg theta rhythm. *Hippocampus*, Vol. 3, No. 3, pp. 317–330, 1993.
 - [22] John O’keefe and Neil Burgess. Dual phase and rate coding in hippocampal place cells: theoretical significance and relationship to entorhinal grid cells. *Hippocampus*, Vol. 15, No. 7, pp. 853–866, 2005.
 - [23] Torkel Hafting, Marianne Fyhn, Tora Bonnevie, May-Britt Moser, and Edward I Moser. Hippocampus-independent phase precession in entorhinal grid cells. *Nature*, Vol. 453, No. 7199, pp. 1248–1252, 2008.
 - [24] Richard J Gardner, Erik Hermansen, Marius Pachitariu, Yoram Burak, Nils A Baas, Benjamin A Dunn, May-Britt Moser, and Edvard I Moser. Toroidal topology of population activity in grid cells. *Nature*, Vol. 602, No. 7895, pp. 123–128, 2022.
 - [25] Timothy P Lillicrap, Adam Santoro, Luke Marris, Colin J Akerman, and Geoffrey Hinton. Backpropagation and the brain. *Nature Reviews Neuroscience*, Vol. 21, No. 6, pp. 335–346, 2020.
 - [26] Mufeng Tang, Helen Barron, and Rafal Bogacz. Learning grid cells by predictive coding. *arXiv preprint arXiv:2410.01022*, 2024.

5 Supplementary Material

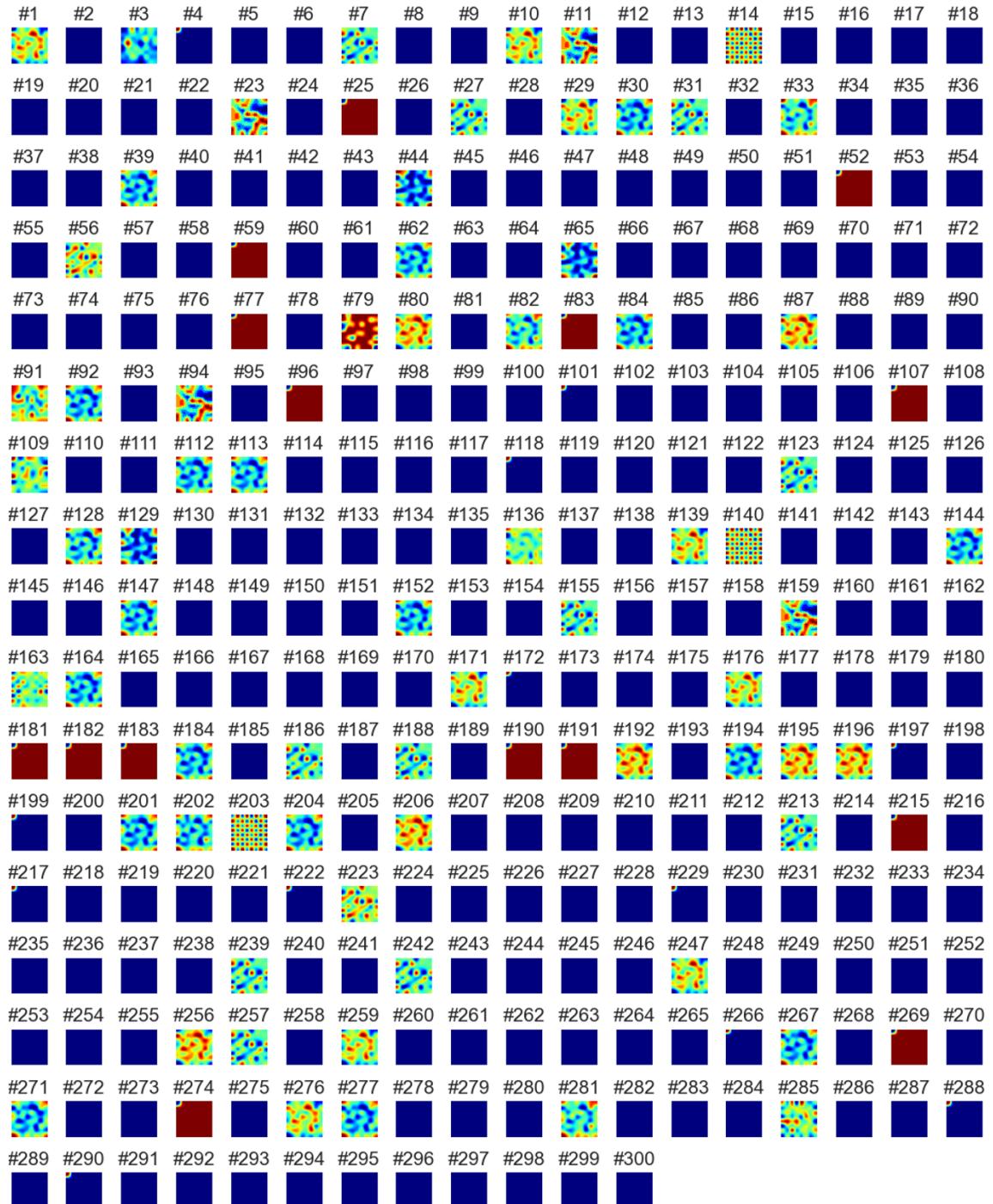


Figure 11 **Emergence of grid and place cells without conjunctive input.** Neurons 14, 140, and 203 show clear grid patterns.

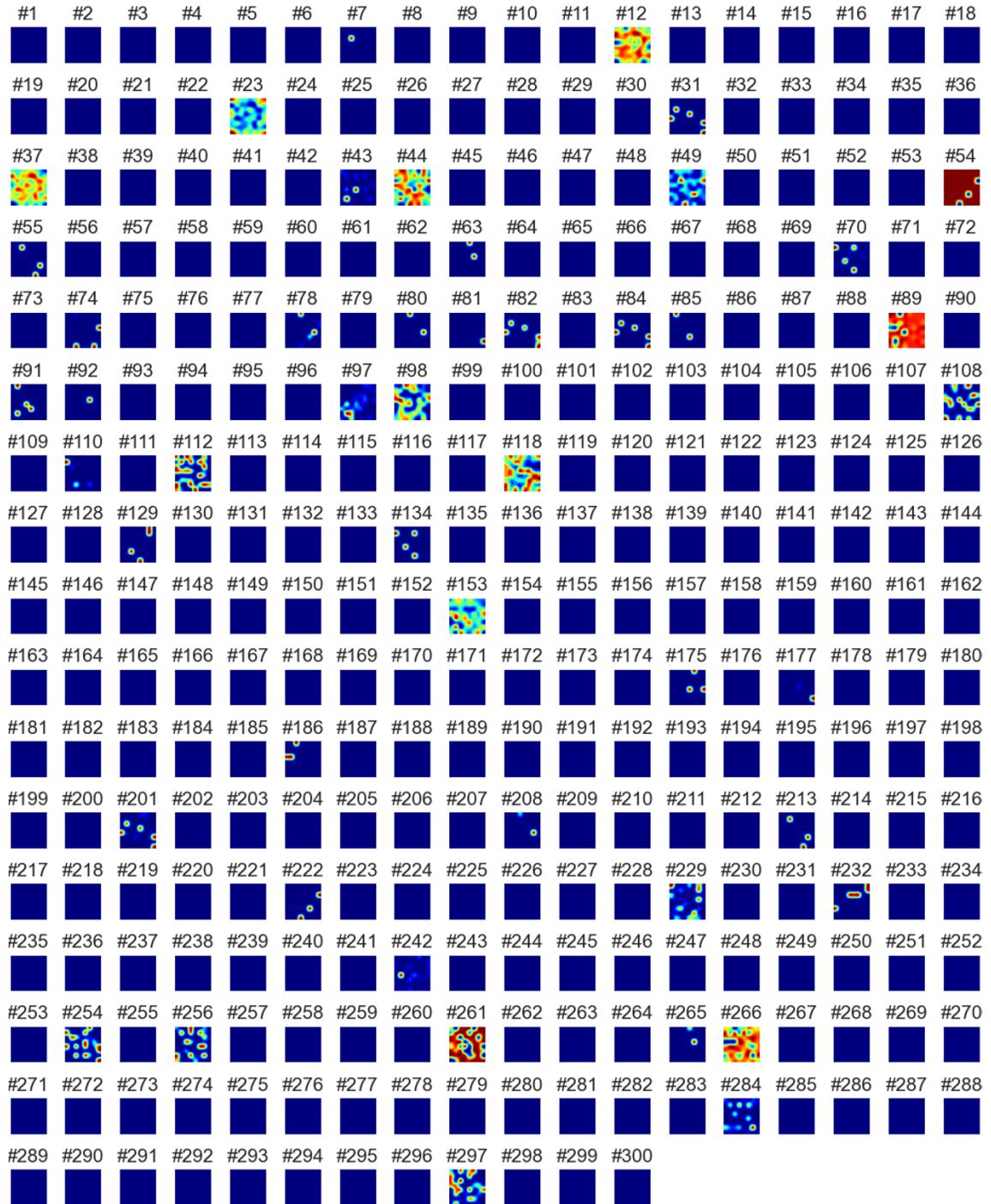


Figure 12 Place cell representations in the hippocampal module without conjunctive input. Localized place fields emerge in the hippocampal module when both grid cell activity g_t and preprocessed sensory inputs x'_t are provided as parallel inputs, without computing their element-wise product.