# Self-supervised learning of scale-invariant neural representations of space and time

Abolfazl Alipour[1,2], Thomas W. James[1,2], Joshua W. Brown[1,2], and Zoran Tiganj*[1,2,3]

[1]Program in Neuroscience, Indiana University Bloomington
[2]Department of Psychological and Brain Sciences, Indiana University Bloomington
[3]Department of Computer Science, Indiana University Bloomington

**Abstract**

Hippocampal representations of space and time seem to share a common coding scheme characterized by neurons with bell-shaped tuning curves called place and time cells. The properties of the tuning curves are consistent with Weber's law, such that, in the absence of visual inputs, width scales with the peak time for time cells and with distance for place cells. Building on earlier computational work, we examined how neurons with such properties can emerge through self-supervised learning. We found that a network based on autoencoders can, given a particular inputs and connectivity constraints, produce scale-invariant time cells. When the animal's velocity modulates the decay rate of the leaky integrators, the same network gives rise to scale-invariant place cells. Importantly, this is not the case when velocity is fed as a direct input to the leaky integrators, implying that weight modulation by velocity might be critical for developing scale-invariant spatial receptive fields. Finally, we demonstrated that after training, scale-invariant place cells emerge in environments larger than those used during training. Taken together, these findings bring us closer to understanding the emergence of neurons with bell-shaped tuning curves in the hippocampus and highlight the critical role of velocity modulation in the formation of scale-invariant place cells.

***Keywords*** Place cells, Time cells, Scale invariance, Hippocampus, Self-supervised learning

# 1 Introduction

Hippocampal activity is characterized by neurons tuned to the magnitude of spatial, temporal, or other task-relevant variables. For instance, place cells increase their firing rate when an animal is at a specific location in the environment [O'Keefe, 1976]. Analogously, time cells increase their firing rate at a specific temporal distance from a salient input [Pastalkova et al., 2008, MacDonald et al., 2011, Umbach et al., 2020]. A similar coding scheme has been observed for sound frequency [Aronov et al., 2017] and amount of accumulated evidence [Nieh et al., 2021, Morcos and Harvey, 2016] when those were relevant for

*Corresponding author. Email: ztiganj@iu.edu.
This manuscript has been accepted for publication in the Journal of Computational Neuroscience.

animals to obtain a reward. Together, these neural representations form mental or cognitive maps [O'Keefe and Nadal, 1978, Behrens et al., 2018].

The similarity of representations for different variables has led researchers to propose that they rely on a common computational mechanism [Buzsáki and Tingley, 2018, Eichenbaum, 2013, 2017a, Howard and Hasselmo, 2020]. This view aligns with empirical evidence showing substantial overlap between place and time cells, with many hippocampal neurons modulated by both spatial and temporal information [Kraus et al., 2013]. Furthermore, Weber's law [Portugal and Svaiter, 2011] provides a common denominator for representations of non-circular variables [Gibbon, 1977, Wilkes, 2015]. For instance, the variance in distance estimation scales with the mean for both spatial [Lappe et al., 2007, Yoshioka, 1929, Harris and Wolbers, 2012] and temporal distances [Buhusi and Meck, 2005, Gibbon, 1977, Buhusi et al., 2009, Balcı and Freestone, 2020]. This behavior can be explained if the magnitude of the variable is represented internally on a logarithmic axis [Dehaene, 2003], which confers scale invariance to the neural representation.

Scale invariance is a property where the structure or behavior of a system remains unchanged under scaling transformations. Mathematically, a function $f(x)$ exhibits scale invariance if scaling the input $x$ by a constant factor $k$ results in a proportional scaling of the output: $f(ax) = a^{\gamma} f(x)$, where $\gamma$ is a scaling exponent that characterizes the system's response to scaling. This property implies that the functional form of $f(x)$ remains consistent across different scales of $x$, allowing the system to operate uniformly regardless of the magnitude of the input. In the context of neural coding, representing variables on a logarithmic scale ensures that multiplicative changes in input result in additive shifts in the internal representation. Specifically, if a variable $x$ is encoded as $y = \log(x)$, then scaling the input by a factor $a$ transforms the representation as follows: $y' = \log(ax) = \log(a) + \log(x) = y + \log(a)$. This expression demonstrates that a multiplicative change in the input $x$ by a factor $k$ leads to an additive shift of $\log(k)$ in the neural representation $y$. As a result, the relative differences between inputs are preserved across scales, because the distance between any two points on the logarithmic scale corresponds to the logarithm of their ratio: $\Delta y = \log(x_2) - \log(x_1) = \log\left(\frac{x_2}{x_1}\right)$. By preserving the ratios of inputs rather than their absolute differences, the logarithmic representation aligns with Weber's law, where perceptual sensitivity is proportional to the relative change in stimulus rather than the absolute change.

To construct a logarithmic representation, time and place cells should exhibit two specific properties. First, the width of the time (or place) fields, defined as the standard deviation of their Gaussian activation profiles, should increase proportionally with the peak time latency (or distance), so they appear equally wide when plotted against the logarithm of latency (or distance). Second, the distribution of peak latencies (or distances) should follow a power-law distribution, making the peak positions equidistant on a logarithmic axis. Neural representations with these properties will exhibit a constant coefficient of variation—the ratio of the standard deviation to the mean remains constant—reflecting Weber's law. This implies that the relative uncertainty in the representation scales with the magnitude of the variable, embodying a scale-invariant coding scheme that provides a unified framework for representing different variables such as time and space [Howard et al., 2014, Tiganj et al., 2021, 2022, Howard et al., 2015a].

Neural activity in the hippocampus is consistent with the two properties listed above: The width of temporal fields increases with the peak latency, and the distribution of peak latencies is sublinear [MacDonald et al., 2011, Cruzado et al., 2020, Tiganj et al., 2017, 2018, Cao et al., 2022]; Similarly, in spatial navigation, when animals are believed to use path integration rather than visual inputs, the width of the place fields increases with the distance from boundaries, and the distribution of peak distances is sublinear [Sheehan et al., 2021, Kraus et al., 2013].

Several models have been developed to explain the activity of temporally and spatially modulated cells in the hippocampus. Time cells that give rise to an internal logarithmic representation of time emerge from recurrently connected populations of neurons with specific connectivity profiles [Tank and

Hopfield, 1987, Grossberg and Schmajuk, 1989, Shankar and Howard, 2012, Voelker and Eliasmith, 2018]. Place cells are thought to emerge through the process of path integration, where an individual's velocity and head direction are used to continually update their current position relative to a starting point. Path integration is widely believed to be a fundamental component of spatial navigation across multiple animal species, including humans [Etienne, 1992, Mittelstaedt and Mittelstaedt, 1982]. For example, when an animal navigates in the dark (in the absence of external cues), place cells typically preserve the spatial tuning, suggesting they may be using path integration to keep track of the animal's position [Quirk et al., 1990, Zhang et al., 2014, Mcnaughton et al., 1989, Save et al., 1998]. While place cell firing is also thought to be impacted by other sensory inputs that provide an estimate of position relative to familiar landmarks [Jayakumar et al., 2019, Etienne et al., 1996, Chen et al., 2013], these estimates are proposed to serve as a recalibration signal to compensate for errors in path integration [Jayakumar et al., 2019, McNaughton et al., 1996]. Moreover, place cells can be observed as early as the time of first exploration in rats (post-natal day 16) [Muessig et al., 2016], preceding the formation of other specially tuned cells, such as grid cells [Bjerknes et al., 2018].

To describe path integration in place cells, computational models assume that animals have access to a neural signal corresponding to speed and movement direction. Neurons with activity correlated to the running speed were found in several areas, primarily in the medial entorhinal cortex [Sargolini et al., 2006, Kropff et al., 2015, Hinman et al., 2016]. Neurons tuned to head direction were also reported in several areas, including dorsal presubiculum [Taube et al., 1990] and the medial entorhinal cortex [Sargolini et al., 2006] (note though that head direction does not necessarily correspond to the movement direction, which was represented less strongly by the neurons in those regions [Raudies et al., 2015, Alexander et al., 2020]).

Well-established path integration models are based on continuous attractor neural networks (CAN) and describe the network activity as a 'bump' that moves continuously over a topological map of neurons. In these models, each neuron is associated with a particular location, and it has excitatory connections with nearby neurons (representing nearby locations) and inhibitory connections with distant neurons. This configuration ensures that if a neuron fires, it stimulates nearby neurons to fire while suppressing distant neurons, creating a localized 'bump' of activity that represents the current position of the animal. As the animal moves, this bump of activity moves correspondingly across the network [Samsonovich and McNaughton, 1997, Conklin and Eliasmith, 2005, Redish, 1999]. While successful in generating place cells, this family of models does not give rise to place fields that obey Weber's law, where the width of the fields would increase linearly as a function of the distance from the spatial landmark.

Howard et al. [2014] proposed another approach in a spirit similar to the attractor models. That model provides a spatial extension of an earlier model for hippocampal time cells [Shankar and Howard, 2012] that builds on top of a temporal context model [Howard and Kahana, 2002, Sederberg et al., 2008]. The key idea behind this approach was that the velocity signal modulates the rate of sequential activation of time cells, therefore converting time cells into place cells while preserving the properties necessary for Weber's law. To implement this in the form of a neural network, [Howard et al., 2014] uses a two-layer architecture, where the first layer is composed of a set of neurons that perform leaky integration (characterized by exponentially decaying impulse responses), each with a different time constant. The population of such neurons constitutes an approximation of a real-domain Laplace transform facilitating the implementation of a variety of cognitive computations [Howard et al., 2015b]. One such computation includes modulation of the decay rate of the leaky integrators by velocity, which turns exponential functions of time into exponential functions of distance. Inverting the Laplace transform, which can be approximated with matrix multiplication, gives rise to time and place cells that are consistent with Weber's law. While analytically tractable, this approach requires the setting of specific weights to implement the inverse Laplace transforms. These weights need to implement high-order spatial derivatives, which resemble lateral inhibition. While spatial derivatives have been observed in the brain, for instance, in the retina [Kandel et al., 2000], it is not clear how such an operation could

emerge in the hippocampus. Existing work does not provide a learning mechanism that gives rise to time and place cells that obey Weber's law.

Here, we sought to provide an unsupervised learning algorithm for the formation of time and place cells consistent with Weber's law by building on Howard et al. [2014] and previous deep-learning models. Specifically, we trained autoencoder models with inputs that decayed exponentially with a spectrum of time constants. We found that scale-invariant place and time cells emerge especially under the conditions of local and shared connectivity. This finding was quite robust to hyperparameter settings. Moreover, sharing connectivity weights allowed the model to generalize to differently-sized environments. We first trained and evaluated the model on a linear track and then showed that it can be generalized to 2 dimensions, giving rise to log-polar place cells tuned to point landmarks.

# 2 Methods

## 2.1 Training data: linear track

To train the network, we generated a set of random trajectories along a 180 cm long track (following the track length in Sheehan et al. [2021] and inspired by several other studies that examined neural representations in a linear track [Gothard et al., 2001, Bjerknes et al., 2018]). We used 1mm spatial resolution and sampled with replacement 1800 spatial locations along the track and computed velocity between each two sampled locations. Subsequently, the obtained velocities were fed into the input layer of the neural network, as described below. Note that since the network was trained to reconstruct the *current* input, the temporal order of the samples did *not* impact the training. To construct the training and testing data, we used 6000 trials, each comprising 1800 velocity samples.

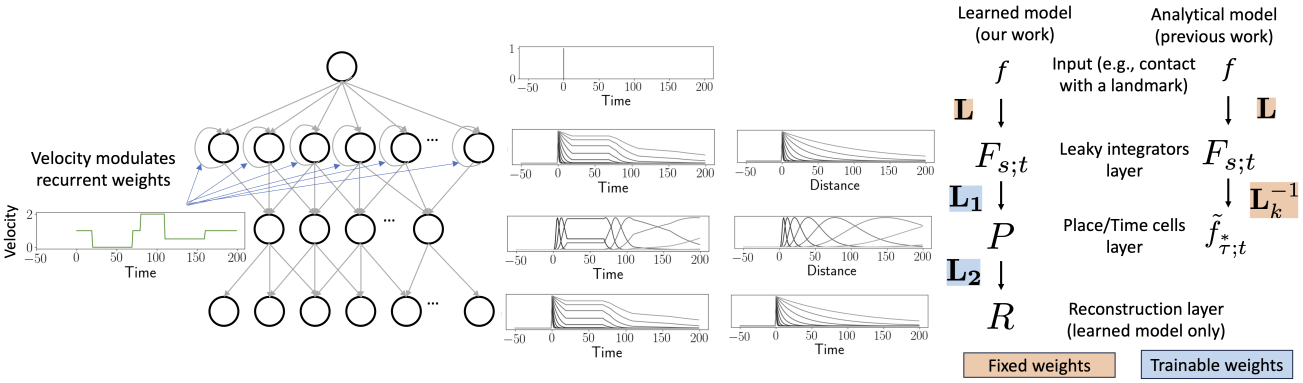


Figure 1: Sparse autoencoder model proposed here builds upon a previous computational model from Howard et al. [2014] and provides a learning mechanism for the formation of scale-invariant time and place cells. The network on the left illustrates the three-layer autoencoder architecture, where the last layer is trained to reconstruct the activity of the leaky first layer. Activity traces for each layer are shown in the middle as a function of time and distance, showing exponential decay and scale-invariant sequential activation. The activity traces were constructed using the computational model with analytically derived weights from Howard et al. [2014]. We investigate under what conditions a model trained using a reconstruction objective gives rise to a scale-invariant sequential activation.

## 2.2 Network architecture

Following the model described in Howard et al. [2014], we assume that hippocampal cells receive input that decays exponentially as a function of traveled distance or elapsed time. We hypothesize that spatial and temporal tuning of hippocampal cells emerges in an unsupervised manner through the reconstruction of the activity of exponentially decaying cells. To implement these operations, the proposed network consists of three layers: leaky integrators, place cells, and reconstruction layer (Figure 1). Below, we describe each of the layers in detail.

### 2.2.1 Layer 1: Leaky integrators

The first layer of the network is composed of a set of leaky integrators $F(t)$ with a spectrum of time constants $1/s$, each modulated by velocity $v$:

$$\frac{dF(t)}{dt} = v(t)\left(-sF(t) + f(t)\right). \tag{1}$$

For encoding time, we set $v(t) = 1$ and write the integral form of the above equation:

$$F(s;t) = \int_0^t e^{-s(t-t')} f(t')dt'. \tag{2}$$

As noted in Shankar and Howard [2012], $F(s;t)$ is a real-value approximation of the Laplace transform of $f(t)$. It is a real-value transform since $s$ is real, not complex. It is only an approximation since $s$ is discrete, not continuous. If $s$ was continuous (which would require infinitely many neurons since each value of $s$ corresponds to one leaky integrator), at time $t'$, $F(s;t=t')$ would encode a perfect memory of $f(0 < t < t')$. With discrete $s$, $F(s;t=t')$ encodes only an approximation of $f(0 < t < t')$.

To encode elapsed time since some salient transient stimulus and subsequently give rise to time cells, we consider a simple case where $f(t)$ is the stimulus onset. Without loss of generality, we assume that stimulus onset is at time $t = 0$: $f(t) = \delta(t = 0)$. Therefore, following the stimulus onset, units in $F(t)$ will decay exponentially from 1 towards 0 as a function of time with a spectrum of decay rates $s$ (Figure 1):

$$F(t) = e^{-st}. \tag{3}$$

When encoding distance, velocity $v$ can take any positive or negative value. For the moment, we focus on representing distance in 1-D space (linear track) before extending it to 2-D. Modulating the decay rate by velocity turns exponentially decaying functions of time into exponentially decaying functions of distance. To show that analytically, we write velocity as a time derivative of distance $v = \frac{dx}{dt}$:

$$\frac{dF(t)}{dt} = \frac{dx}{dt}\left(-sF(t) + f(t)\right). \tag{4}$$

Multiplying the above equation by $dt/dx$ gives:

$$\frac{dF(x(t))}{dx} = -sF(x(t)) + f(x(t)). \tag{5}$$

Now the units in $F$ decay exponentially as a function of distance $x$ from the start point such that $f(x(t)) = \delta(x = 0)$:

$$F(x) = e^{-sx}. \tag{6}$$

Here $\delta(x = 0)$ represents contact with a landmark such as a wall or an object (e.g., the beginning of the linear track). The leaky integrators decay towards zero when the movement direction is away

from the origin and increase to their maximum value (i.e., 1) when the agent moves toward the origin (Figure 1).

Note that the above formalism can be generalized beyond time and distance. If instead of velocity, the decay is modulated by some variable $\alpha = \frac{dz}{dt}$, then $F$ decays exponentially as a function of $z$. Thus $F$ can be turned into an exponentially decaying function of any variable whose derivative is observable from the inputs [Mochizuki-Freeman et al., 2023, Maini et al., 2023]. For instance, this can provide an explanation for neurons tuned to numerosity (if $\alpha$ is a change in the count).

In our experiments we used 100 neurons in the $F$ layer with values of $s$ from 0.001 to 0.1 spaced either logarithmically or linearly as specified in the Results section.

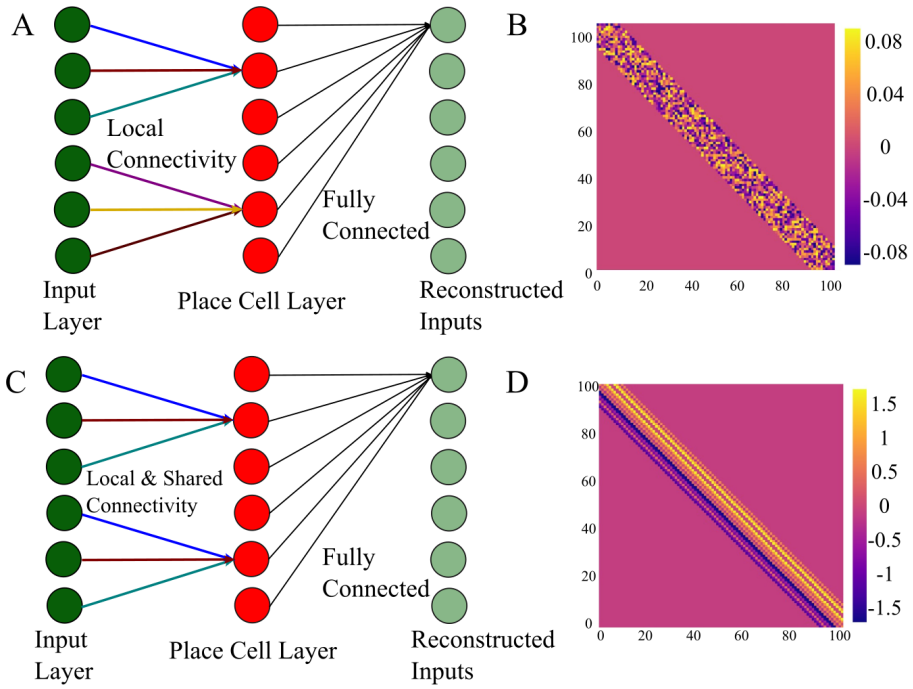


Figure 2: To promote the emergence of scale-invariant place and time cells, following the theoretical results, we experimented with two different connectivity constraints: Local connectivity and local connectivity with shared weights. A) Local connectivity only allows connection between neurons across two layers if they have the same index or their indices are less than $\pm$ 9 apart. B) example of the resulting connectivity matrix. C) Local connectivity with shared weights: in addition to being locally connected, all neurons in the $P$ layer share the same set of 19 weights. Therefore, for all neurons, the values of weights for i-th neighbor change simultaneously. D) example of the resulting connectivity matrix.

### 2.2.2 Layer 2: Time/place cells (P) layer

Cells in the leaky integrator layer project into a set of cells we expect to become time and place cells after the self-supervised training. Previous work [Shankar and Howard, 2012, Howard et al., 2014] has shown that there is a linear mapping from a set of leaky integrators to a set of scale-invariant time and place cells (depending on whether the exponential decay in $F$ was a function of time or distance).

Specifically, that linear mapping approximates a $k$-th order numerical derivative with respect to $s$:

$$\tilde{f}(\overset{*}{\tau};t) = \mathbf{L}_k^{-1} F(s;t) = \frac{(-1)^k}{k!} s^{k+1} \frac{d^k}{ds^k} F(s;t), \tag{7}$$

where $\overset{*}{\tau} := k/s$ and $\mathbf{L}_k^{-1}$ represents the inverse Laplace transform operator approximated by a finite difference of order $k$. When $k \to \infty$, the above equation inverts the Laplace transform [Post, 1930], bringing functions of $s$ from the Laplace domain into functions of $\overset{*}{\tau}$ which represents an internal temporal axis.

Importantly, functions of time in the inverse are encoded through instantaneous activity in $\tilde{f}(\overset{*}{\tau};t)$ meaning that $\tilde{f}(\overset{*}{\tau};t=t')$ encodes approximation of $f(0<t<t')$ such that value of $\tilde{f}(\overset{*}{\tau}_i;t=t')$ approximates the value of $f(t=\overset{*}{\tau}_i)$, where $\overset{*}{\tau}_i$ is $i$-th value of $\overset{*}{\tau}$. When the input function is a delta pulse, like in our case, then units in $\tilde{f}$ activate sequentially as time cells with peak times equal to $\overset{*}{\tau}$: $\partial F/\partial t = 0 \Rightarrow t = \overset{*}{\tau}$. Furthermore, the coefficient of variation (CV) computed as the ratio of the standard deviation and the mean of $\tilde{f}$ across $\overset{*}{\tau}$ is not a function of $t$: $CV = (k-2)^{-1/2}$ implying that the width of the temporal fields scales linearly with the peak time, consistently with Weber's law.

The same equation holds for exponentially decaying functions of distance such that $\tilde{f}(\overset{*}{x};x) = \mathbf{L}_k^{-1} F(s;x)$. In this case, $\tilde{f}(\overset{*}{x};x)$ is a set of place cells that activate sequentially as a function of distance from the origin such that the width of the place fields scales linearly with the peak distance (Figure 1).

Our objective here is to evaluate whether and under what conditions the inverse transform giving rise to time and place cells can emerge through self-supervised learning. Therefore, rather than setting the weights between the leaky integrator and $P$ layer to $\mathbf{L}_k^{-1}$, we initialized those weights using a random Gaussian distribution with a mean of zero and standard deviation of $1/l$, where $l$ is the number of neurons in the leaky integrator layer. This initialization method follows commonly used Kaiming's method [He et al., 2015]. Subsequently, we trained the network in a self-supervised way: $P = ReLU(\mathbf{L_1}F)$. For 100 leaky integrators and 100 cells in P. $\mathbf{L_1}$ is a 100 by 100 matrix of trainable weights, ReLU is the activation function that ensures that activities of neurons in the $P$ layer are positive.

### 2.2.3 Layer 3: Reconstruction layer

To implement self-supervised learning, we use autoencoder architecture such that the $P$ layer projects onto the reconstruction layer $R$ via a 100 by 100 matrix of trainable weights $\mathbf{L_2}$: $R = \mathbf{L_2}P$

The network was trained using the mean square error (MSE) loss between $R$ and $F$:

$$l_{MSE} = \sum_{n=1}^{N} (R_n - F_n)^2, \tag{8}$$

where N=1800 is the number of samples in each trial.

## 2.3 Regularization

For time and place cells to emerge, it was necessary to incorporate regularization. We investigated the following regularization regimes: dropout, $L2$ norm, activity penalty, sparsity, shared weights, and local weights.

To motivate different regularization approaches, we note that the inverse transform matrix $\mathbf{L}_k^{-1}$ approximates a $k$-th-order numerical derivative with respect to $s$. In practice, this means that each row of $\mathbf{L}_k^{-1}$ is a shifted version of the previous row and has only $2k+1$ non-zero values. The fact that $\mathbf{L}_k^{-1}$ weights repeat across rows means that the connectivity pattern between a set of leaky integrators and

each cell in the $P$ layer is the same. This implies that learning the connectivity pattern for one cell can be sufficient if that connectivity pattern can be replicated across other cells.

The properties of $\mathbf{L}_k^{-1}$ inspired our regularization methods. The local weight regularization restricts the connectivity of each neuron to local neighbors in the previous layer based on the following criteria:

$$\mathbf{L_1}_{i,j} = \begin{cases} \mathcal{N}(\mu,\, \sigma^2), & \text{if } |i-j| \leq 9 \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

where $\mathcal{N}(\mu,\, \sigma^2)$ is a normal distribution with a zero mean and unit variance. This criterion restricted the connectivity of each neuron to only 19 other neurons in the previous layer (mimicking value of $k = 9$, which is similar to values used in previous studies [Howard et al., 2014]), i.e., neurons in the previous layer with the same index $\pm 9$, (Figure 2A, B). A network composed of 100 neurons in the leaky integrator layer and 100 neurons in the $P$ layer has 1900 trainable parameters in $\mathbf{L_1}$.

Furthermore, we used shared weights regularization inspired by the fact that rows of $\mathbf{L}_k^{-1}$ are translated versions of each other. The shared weights regularization implies that each cell in the $P$ layer is connected with a subset of leaky integrators using the same weights (Figure 2C, D). This further restricts the number of trainable parameters in $\mathbf{L_1}$ such that for a network composed of 100 neurons in the leaky integrator layer and 100 neurons in the $P$ layer, matrix $\mathbf{L_1}$ would have only 19 trainable parameters.

Another regularization that we investigated to promote the formation of time and place cells was the activity penalty ($l_A$). Specifically, we defined an activity penalty as the squared root of the sum of squares of activities in the $P$ layer:

$$l_A = \sqrt{\sum_{i=1}^{M} p_i^2} \tag{10}$$

where $M = 100$ is the number of neurons in the $P$ layer and $p_i$ is the firing rate of the i-th neuron in the $P$ layer. To control the magnitude of loss generated by activity penalty, we multiplied it by a coefficient $\lambda_2$ with values between $10^{-7}$ and $10^{-9}$.

Finally, as another constraint of the activity in the $P$ layer, we implemented sparsity regularization loss that promotes sparse activations across the neuron population. Specifically, for each training example $j$, we calculated the average normalized firing rate of the neurons in the $P$ layer using the sigmoid activation function:

$$\hat{\rho}_j = \frac{1}{M} \sum_{i=1}^{M} \sigma(p_i^{(j)}) \tag{11}$$

where $\sigma(p_i^{(j)})$ is the activation of neuron $i$ for training example $j$ after applying the sigmoid function, $\hat{\rho}_j$ represents the average activation across all neurons in the place cell layer for the $j$-th training example. We used the sigmoid function to ensure that the firing rates are within the range $[0, 1]$, making them suitable for interpretation as probabilities.

We then computed the Kullback–Leibler (KL) divergence between the desired sparsity level $\rho$ and the average firing rate $\hat{\rho}_j$ for each training example:

$$\text{KL Loss} = \sum_j \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho) \log \frac{1-\rho}{1-\hat{\rho}_j}, \tag{12}$$

where $j$ is the training step. The parameter $\rho$ is the sparsity parameter that represents the desired mean normalized firing rate across the neuron population for each input.

By minimizing this KL divergence, we encourage the average activation across neurons for each input to match $\rho$, effectively promoting sparsity at the population level per sample. This arrangement

forces the representations in the $P$ layer to have an average firing probability of $\rho$ across neurons for each input. Unlike simple activity penalties that might push activations toward zero, this KL divergence penalizes deviations from the desired sparsity level $\rho$. The loss is minimized when $\hat{\rho}_j = \rho$ for each training example, encouraging the network to maintain the target average activation level rather than suppressing activations entirely. The KL loss was scaled by a coefficient $\lambda_1$ with a value of $10^{-4}$.

Accordingly, the total loss ($l$) was the sum of MSE, KL loss, activity penalty and $L2$ regularization loss:

$$l = l_{MSE} + \lambda_1 l_{KL} + \lambda_2 l_A + \lambda_3 L2 \tag{13}$$

## 2.4 Hyper-parameter sweeps

To investigate the conditions in which place and time cells emerge, we performed a systematic sweep on the following hyper-parameters: $P$ cell layer size: 50, 100, dropout value (between the leaky integrators and the $P$ layer): 0.1, 0.2, 0.3, activity penalty coefficient ($\lambda_2$): $10^{-9}$, $10^{-8}$, $10^{-7}$, $L2$ (weight decay) regularization coefficient $\lambda_3$: 0.0001, 0.001, 0.01. In all experiments, we used the Adam optimizer and evaluated learning rates of 0.001, 0.01, and 0.1. We found the best results, in terms of stability of convergence and the number of place cells, with a learning rate of 0.1 and all the results presented here are obtained with that learning rate. All of the experiments were run with three different seeds and presented results include standard deviations and confidence intervals computed from the three sets of results. All simulations were run using PyTorch version 2.5.1 and Pytorch Lightning 2.3.0 [Paszke et al., 2019, Falcon, 2019].

## 2.5 Velocity as an input rather than a modulator

In the model described in Section 2.2.1, we used velocity as a modulator of the decay rate of leaky integrators. To examine whether scale-invariant place cells can emerge from integrating velocity, we performed a control experiment and used the velocity as an input to the model using the following equation instead of Equation 1:

$$\frac{dF(t)}{dt} = -sF(t) + v(t). \tag{14}$$

The rest of the model was kept the same to isolate the effect of velocity integration.

## 2.6 Evaluation on linear track

We focused our evaluation on place cells since, in our model, time cells are a special case of place cells with a velocity fixed to 1. We quantified the number and properties of place cells emerging in the $P$ layer. To count a cell as a place cell, we required it to fire above 50% of its maximum firing rate only in a small area of the track (less than 20% of locations) and to have low activity (below 25% of the maximum) in the rest of the locations on the track. Additionally, we excluded cells that had a peak within the five bins from the beginning or end of the track since those cells did not have a typical bell-shaped firing characteristic for place cells but a decaying or growing profile. For the evaluation of scale invariance, we examined whether the coefficient of variation remained constant across the place cells.

## 2.7 Evaluation in 2-D environment

While we used a linear track to train the network, we also evaluated the spatial tuning in a 2-D environment. Specifically, we tested the network within a circular area with a radius of 80 cm (Figure 3A). We

designed 50 trajectories, each comprising 8000 time steps, to emulate the random exploration patterns of rodents. This was achieved by aligning the rodent's facing direction with its heading direction [Cueva and Wei, 2018]. Each trajectory point was computed by selecting a velocity from a Rayleigh distribution (scale parameter = 1.3) and a rotation angle from a Gaussian distribution (mean = 0, standard deviation = 165 degrees). To ensure the trajectories remained within the boundaries, if a point strayed beyond 80 cm from the center, the heading angle was adjusted to direct back towards the center.

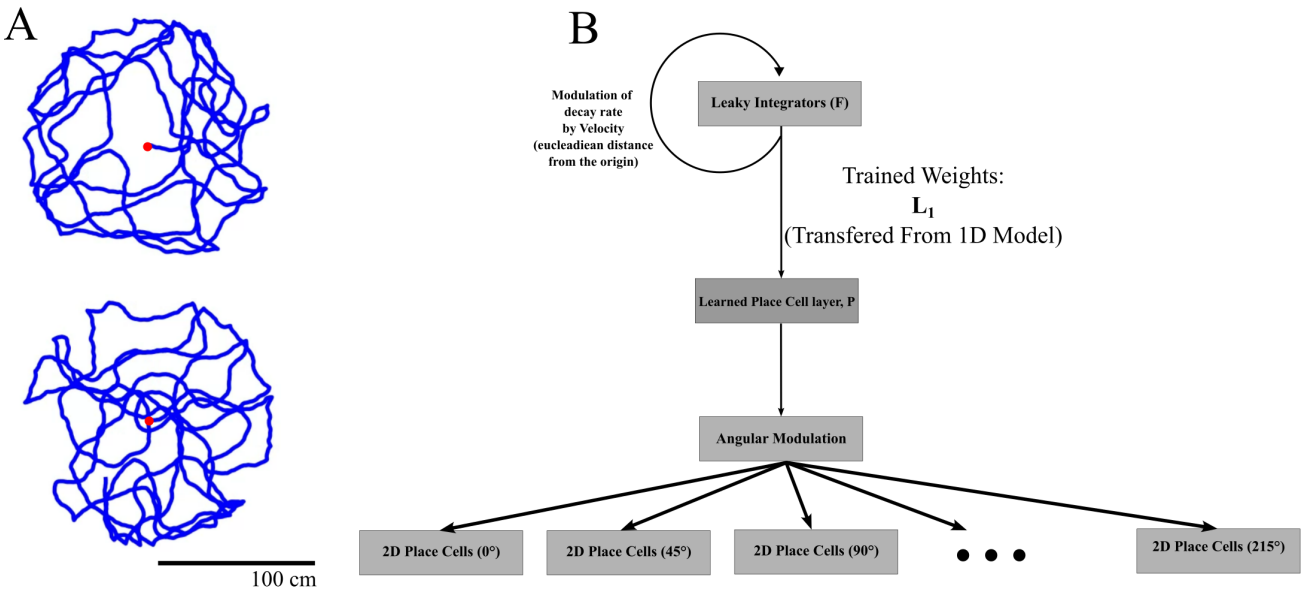


Figure 3: The 2-D model uses the connectivity matrix pretrained on the 1-D model. A) Two examples of 2-D trajectories. The origin (starting point) is marked with a red dot. B) The 2-D model uses velocity inputs and the pretrained matrix from the 1-D model to generate a set of place cells. Place cell layer output was copied into 8 copies, and each copy was modulated by a von Mises distribution with a different center, giving rise to the angular specificity of cells.

The 2-D model was built by expanding the linear track model: we used the weights obtained by training on the linear track. The velocity here refers to the rate of change in Euclidean distance from the central origin of the circle, influencing the leaky integrators similar to the linear model. To incorporate angular specificity, the output from the linear track's place cell layer—now functioning as the input for the 2-D place cells—was replicated eight times. Each replication was then modulated by a distinct von Mises function, defined as $f(\theta) = \frac{e^{\kappa \cos(\theta-\mu)}}{2\pi I_0(\kappa)}$, where $\theta$ represents the angle in polar coordinates from the central origin, $\mu$ is the mean direction, $\kappa$ is the concentration parameter, and $I_0$ is the modified Bessel function of the first kind at order zero. This modulation ensured that each set of 2-D place cells was activated within a specific angular range. This approach models the angular input similarly to the angular input in grid cell models [Banino et al., 2018], emphasizing that the analysis of place cell outputs in this 2-D environment depends critically on the original place cell output from the linear track model.

# 3 Results

We first show the results of training the network on data from a linear track and investigate the importance of shared weights and sparsity constraints for the emergence of scale-invariant place cells. Then, we illustrate that connectivity patterns learned in 1-D can give rise to 2-D place cells. We also

show that the connectivity pattern learned on a short track can give rise to scale-invariant place cells at arbitrarily long tracks.

The leaky integrator layer of the network consisted of 100 neurons with either logarithmically or linearly spaced values of $s$ from 0.001 to 0.1. These values were chosen so that $\overset{*}{\tau} = k/s$ spans the length of the track. Given the spatial resolution of 1mm and with $k = 9$, $\overset{*}{\tau}$ values ranged from 80mm to 8000mm. The default choice for spacing was set to logarithmic spacing unless it is specified otherwise. The reconstruction layer had the same number of neurons as the leaky integrator layer. The number of neurons in the place cell layer varied from 10 to 300 as described in Section 2.4.

## 3.1 Autoencoder with no connectivity constraints

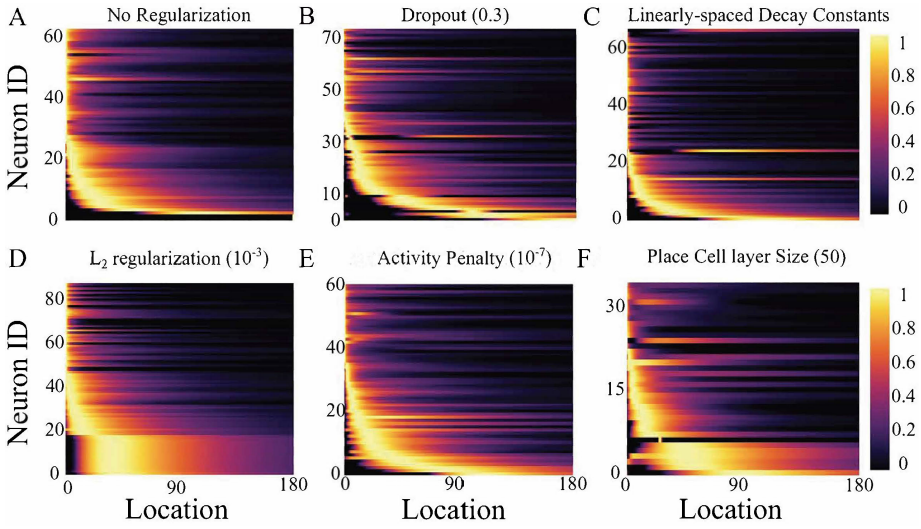


Figure 4: Activity of neurons in the $P$ layer for an autoencoder without sparsity or connectivity constraints. Only neurons with non-zero activity are shown and the activity of each neuron is normalized to be between 0 and 1. Each panel except $E$ has 100 neurons in the $P$ layer. Each panel except $C$ has geometrically spaced time constants. Panel $A$ shows the activity with no regularization, while panels $B$, $D$, and $E$ have different types of regularization with the value of regularization parameters written in the parentheses (see supplemental figures for results with different values of the regularization parameters).

We first trained the network described in Section 2.2 without any regularization such that the loss function consisted only of the MSE loss. Therefore, for a network composed of 100 neurons in the $P$ layer, the number of trainable parameters was 10000 in matrix $\mathbf{L_1}$ and 10000 in matrix $\mathbf{L_2}$. The activity of neurons in the $P$ layer as a function of position along the track is shown in Figure 4A for 100 neurons in the $P$ layer. Some of the neurons remained silent, some had gradually decreasing activity (similar to their input, leaky integrators), and some developed unimodal activity profiles, like biological place cells. It is worth noting that neurons with unimodal activity profiles mostly reached a peak firing rate in a narrow spatial range. This is in contrast to biological place cells, which typically span the environment that the animal is exploring. Adding dropout (Figure 4B and Figure A3), $L2$ regularization (Figure 4D and Figure A2), activity penalty (Figure 4E and Figure A4) and different number of units in the $P$ layer (Figure 4F and Figure A1) had an impact on the neural activity as detailed in Table 1 and in Figure 5. Some configurations resulted in a large number of place cells (e.g., activity penalty of $10^{-8}$ led to 50 place cells). To examine whether the width of the place fields scaled linearly with the peak distance, as expected if the activity is scale-invariant, we computed the correlation coefficient between the width

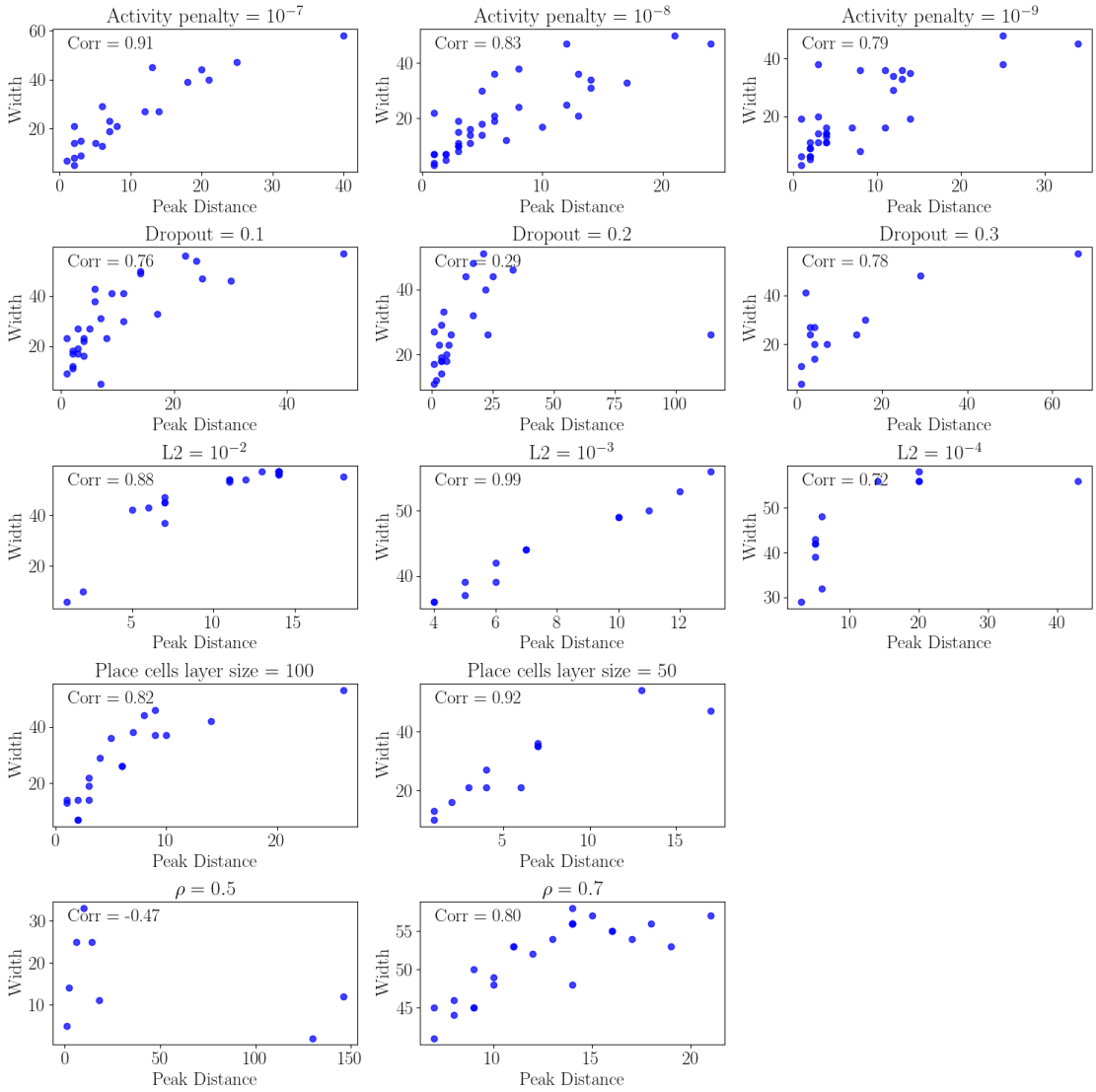


Figure 5: Relationship between place field widths and peak distances for different values of the hyperparameters for autoencoder with no connectivity constraints. Each row corresponds to a sweep over a different parameter. Note that some hyperparameters values resulted in no place cells and those are not shown here (see Section 2.4 for all the values of the hyperparameters). The results shown here are correspond to one of the three seeds. Correlation coefficient is indicated in the upper left corner. For average number of place cells and average correlation coefficient see Table 1.

and the peak. For a scale-invariant activity, the CV would be constant resulting in perfect correlation. While we observed large variability with hyperparameter setting, the largest correlation coefficient in Table 1 of 0.98 indicates that some hyperparameter combinations can result in scale-invariant learning in the absence of connectivity constraints. Replacing the logarithmic spacing of time constants in the leaky integrator layer with linear spacing did not change these properties (Figure 4C).

Table 1
Number of place cells and correlation coefficient between width and peak distance across hyperparameter choices for autoencoder without connectivity constraints. The reported values are the average across three runs with standard deviation for the number of place cells and confidence intervals for the correlation coefficient.

| **Type** | **Value** | **Corr. Coeff. (CI)** | **# Place Cells (±SEM)** |
|---|---|---|---|
| Activity Penalty | $10^{-7}$ | 0.86 (0.79, 0.91) | 26.0 ± 3.06 |
| | $10^{-8}$ | 0.83 (0.75, 0.89) | 30.0 ± 4.16 |
| | $10^{-9}$ | 0.80 (0.71, 0.86) | 32.67 ± 2.33 |
| Dropout | 0.1 | 0.72 (0.60, 0.81) | 29.0 ± 1.15 |
| | 0.2 | 0.48 (0.25, 0.65) | 22.0 ± 1.15 |
| | 0.3 | 0.47 (0.20, 0.67) | 16.67 ± 4.70 |
| L2 | $10^{-2}$ | 0.60 (0.40, 0.75) | 20.0 ± 2.65 |
| | $10^{-3}$ | 0.98 (0.96, 0.99) | 14.33 ± 1.33 |
| | $10^{-4}$ | 0.68 (0.41, 0.84) | 11.0 ± 1.53 |
| $\rho$ | 0.1 | $< 5$ Place Cells | 0 ± 0 |
| | 0.2 | $< 5$ Place Cells | 0 ± 0 |
| | 0.3 | $< 5$ Place Cells | 0 ± 0 |
| | 0.5 | -0.64 (-0.90, -0.06) | 5.67 ± 1.45 |
| | 0.7 | 0.84 (0.76, 0.89) | 33.0 ± 7.64 |
| Size | 50 | 0.82 (0.64, 0.91) | 11.33 ± 0.67 |
| | 100 | 0.81 (0.67, 0.89) | 16.67 ± 2.03 |

## 3.2 Sparse autoencoder with no connectivity constraints

Next, we added a sparsity constraint to the network as described in Section 2.3. We used sparsity coefficient of 0.5 and repeated all the parameter sweeps as in the previous section. Results without any additional regularization are shown in Figure 6A. Other panels in Figure 6 show the effects of applying $L2$ regularization, dropout, activity penalty, using linear instead of log-spacing of time constants and using a different number of neurons in the $P$ layer (see also Figure A7, Figure A6, Figure A8 and Figure A5 for more examples). Summary of the results is provided in Table 2. Overall, the resulting neural activity shows the emergence of relatively broad place fields in some neurons and a much wider spatial range of the peak firing rate than without the sparsity term.

## 3.3 Sparse autoencoder with local and shared weights

Subsequently, we investigated the impact of localizing the connectivity between the input layer and place cell layer (see Section 2.3 for details). This often resulted in place cells with narrow place fields that do not resemble biological data (Figure 7, Figure 8, Table 3, see also Figure A9, Figure A10, Figure A11 and Figure A12 for more examples). Adding activity penalty and dropout made some of the place fields wider

Lastly, we constrained the connectivity matrix between the input layer and the place cell layer to have both local and shared weights (see Section 2.3 for details). This gave rise to scale-invariant place cells that match the theoretical model described in Section 2.2 and resemble the experimental data

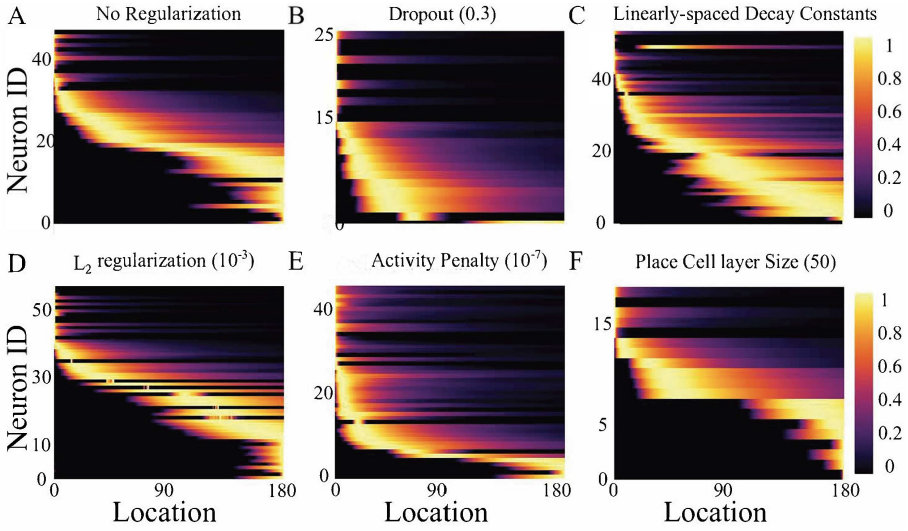


Figure 6: Activity of neurons in the $P$ layer for an autoencoder with sparsity constraints but no connectivity constraint. Panels follow the same conventions as in Figure 4.

Table 2
Number of place cells and correlation coefficient between width and peak distance across hyperparameter choices for autoencoder without connectivity constraints, but with sparsity coefficient $\rho = 0.5$. The reported values are the average across three runs with standard deviation for the number of place cells and confidence intervals for the correlation coefficient.

| **Type** | **Value** | **Corr. Coeff. (CI)** | **# Place Cells (±SEM)** |
|---|---|---|---|
| Activity Penalty | $10^{-7}$ | 0.64 (0.45, 0.77) | 20.33 ± 1.20 |
| | $10^{-8}$ | 0.29 (-0.08, 0.60) | 11.67 ± 0.33 |
| | $10^{-9}$ | 0.69 (-0.50, 0.98) | 3.67 ± 0.88 |
| Dropout | 0.1 | $< 5$ Place Cells | 3.33 ± 0.33 |
| | 0.2 | $< 5$ Place Cells | 3.33 ± 0.67 |
| | 0.3 | $< 5$ Place Cells | 3.33 ± 0.33 |
| L2 | $10^{-2}$ | 0.72 (0.40, 0.89) | 8.33 ± 0.67 |
| | $10^{-3}$ | -0.31 (-0.69, 0.20) | 7.67 ± 1.33 |
| | $10^{-4}$ | -0.69 (-0.95, 0.13) | 4.33 ± 0.67 |
| Size | 50 | $< 5$ Place Cells | 1.0 ± 0.58 |
| | 100 | 0.88 (0.24, 0.99) | 4.0 ± 1.15 |

[Sheehan et al., 2021] (Figure 9, Table 4 and Figure 11, see also Figure A13, Figure A14, Figure A15 and Figure A16 for more examples). The scale invariance is apparent when the spatial distance from the origin is shown on a log scale (second column in Figure 11). The peaks of the place fields were spaced logarithmically (following the theoretical model, this was a consequence of logarithmically spaced time constants), and the width of the place fields increased linearly with the peak location (Figure 10), consistent with Weber's law. The weights learned by the network (fourth column in Figure 11) resembled a spatial derivative, which provides an approximation of the inverse transform (see Section 2.2). Overall, this constraint led to linear relationship between width and the peak distance for a wide range of

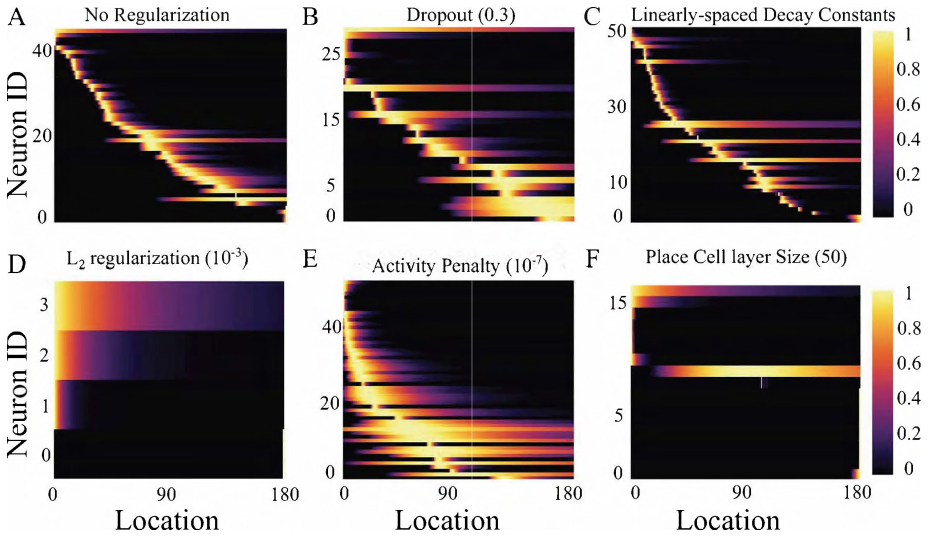


Figure 7: Activity of neurons in the $P$ layer for an autoencoder. Panels follow the same conventions as in Figure 4. Note the sharpening of the place fields compared to the plots without connectivity constraints.

Table 3
Number of place cells and correlation coefficient between width and peak distance across hyperparameter choices for autoencoder with local connectivity. The reported values are the average across three runs with standard deviation for the number of place cells and confidence intervals for the correlation coefficient. Sparsity coefficient $\rho = 0.5$ unless specified otherwise.

| **Type** | **Value** | **Corr. Coeff. (CI)** | **# Place Cells (±SEM)** |
|---|---|---|---|
| Activity Penalty | $10^{-7}$ | 0.31 (0.08, 0.50) | 26.0 ± 0.58 |
| | $10^{-8}$ | 0.49 (0.28, 0.65) | 24.67 ± 2.33 |
| | $10^{-9}$ | 0.36 (0.14, 0.55) | 25.33 ± 0.88 |
| Dropout | 0.1 | 0.50 (0.26, 0.68) | 19.0 ± 2.52 |
| | 0.2 | 0.37 (0.10, 0.60) | 17.67 ± 1.76 |
| | 0.3 | 0.31 (-0.04, 0.60) | 12.67 ± 1.20 |
| L2 | $10^{-2}$ | $< 5$ Place Cells | 1.67 ± 0.67 |
| | $10^{-3}$ | $< 5$ Place Cells | 0 ± 0 |
| | $10^{-4}$ | 0.44 (-0.07, 0.77) | 7.33 ± 0.67 |
| $\rho$ | 0.1 | 0.89 (0.82, 0.94) | 19.0 ± 1.15 |
| | 0.2 | 0.87 (0.78, 0.92) | 19.33 ± 4.18 |
| | 0.3 | 0.75 (0.59, 0.85) | 18.0 ± 3.79 |
| | 0.5 | 0.61 (0.45, 0.73) | 28.67 ± 1.86 |
| | 0.7 | -0.09 (-0.31, 0.14) | 27.0 ± 0.58 |
| Size | 50 | $< 5$ Place Cells | 2.33 ± 0.33 |
| | 100 | 0.76 (0.64, 0.84) | 25.67 ± 2.73 |
| No Sparsity | — | 0.41 (-0.22, 0.80) | 6.33 ± 1.45 |

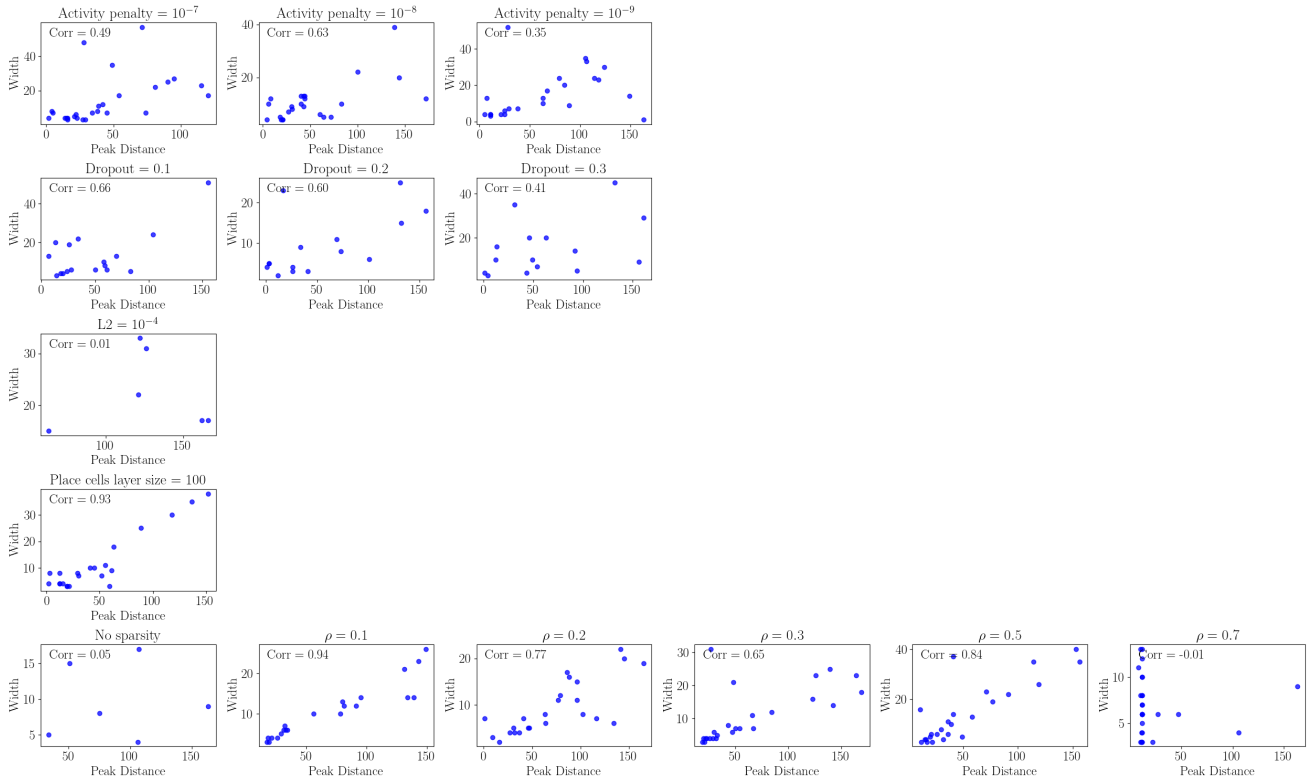


Figure 8: Relationship between place field widths and peak distances for different values of the hyperparameters for autoencoder with local connections. Each row corresponds to a sweep over a different parameter. Note that some hyperparameters values resulted in no place cells and those are not shown here (see Section 2.4 for all the values of the hyperparameters). The results shown here are correspond to one of the three seeds. Correlation coefficient is indicated in the upper left corner. For average number of place cells and average correlation coefficient see Table 3.

hyperparameter choices (Table 4).

We also noted that most of the training time was spent on minimizing the KL loss (coming from the sparsity regularization) rather than the MSE loss (Figure 12). Inserts in panels *A* (local, not shared weights) and *B* (local shared weights) in Figure 10 show decrease of KL (left) and MSE (right) loss as a function of training steps. For both shared and non shared weights, the loss MSE loss decreases rapidly towards a convergence value, while KL loss decreases much more slowly. The difference is especially pronounced for model architecture with local, not shared weights (Figure 12A).

## 3.4 Using velocity as an input to the network does not lead to the emergence of place cells

We investigated if scale-invariant place cells can also emerge when simply using the velocity as an input rather than a modulator, as described in Equation 14. To this end, we trained the model with velocity as input and found that it does not produce place cells (Figure 13). We found no emerging place cells. This finding was not affected by adding sparsity, local weights, and weight sharing. Also, we observed no place cells when we used different value of dropout, $L_2$ loss, activity penalty and sparsity loss (we conducted the same hyperparameter search that was used for other network configurations, as specified in Section2.4).

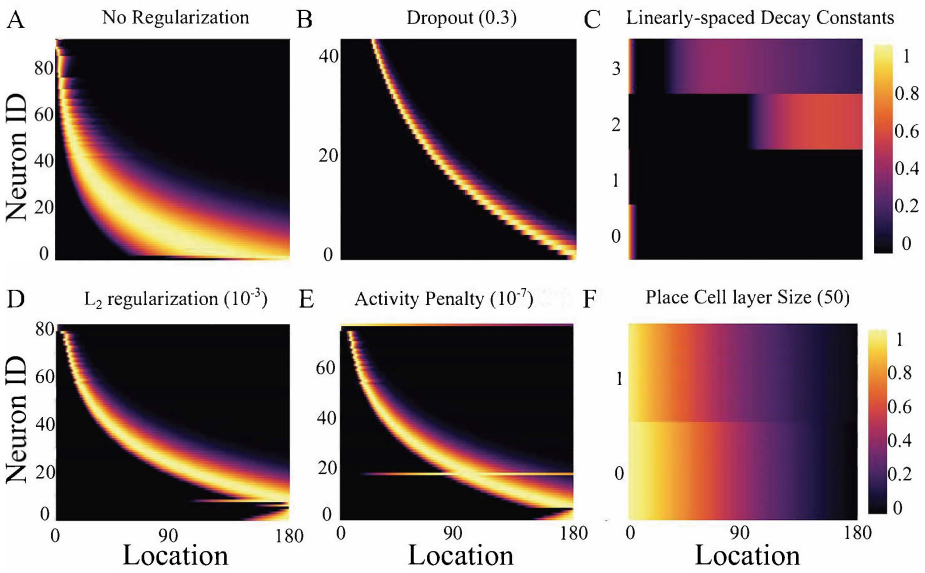


Figure 9: Activity of neurons in the $P$ layer for an autoencoder with sparsity constraint and with local connectivity with shared weights. Panels follow the same conventions as in Figure 4. Neural activity is now consistent with Weber's law (see main text and Figure 11 for details about the evaluation).

## 3.5 Emergence of log-polar place cells in 2-D environment

Adopting the connectivity matrix learned in the 1-D environment and using it in the 2-D environments as described in Section 2.7 gives rise to log-polar spatially modulated fields that resemble findings of [Høydal et al., 2019] and more generally show similar tuning as see in the retina (see Section 4 for discussion). Since the 2-D representation was constructed by replicating multiple 1-D representations (without additional training), it inherits the scale-invariant property demonstrated in the previous section. Therefore the width of the place fields scales linearly as a function of distance from the center, regardless of the angular position.

## 3.6 Emergence of place cells in large environments

The mechanism of shared weights enables the emergence of time and place cells at any temporal or spatial scale, including those not available during training. Because the weights are shared, each time or place cell receives the same projection but from a different set of neurons in the leaky integrator layer (Figure 2C). This means if the network has learned the projection weights on short tracks, it will still have place cells on long tracks. Furthermore, as shown in Section 2.2, those cells will be scale-invariant. The range of scales is determined by the choice of the minimum and maximum value of $\overset{*}{\tau}$ as it determines the shortest and longest peak time or distance.

Figure 15 shows that shared weights learned with the proposed self-supervised setup indeed give rise to place cells at distances larger than the training distances. In this case, the training data was the same as the training data used to generate 1-D place cells in a 180 cm long track, but the testing track had a length of 300 cm. Therefore the same neurons continue having peak activities in the first 180 cm, without any change. The neurons added after training will have peaks distances between 180 cm and 300 cm. The only requirement for this is that there are neurons in the leaky integrator layer with sufficiently large time constants to cover the range of 300 cm.

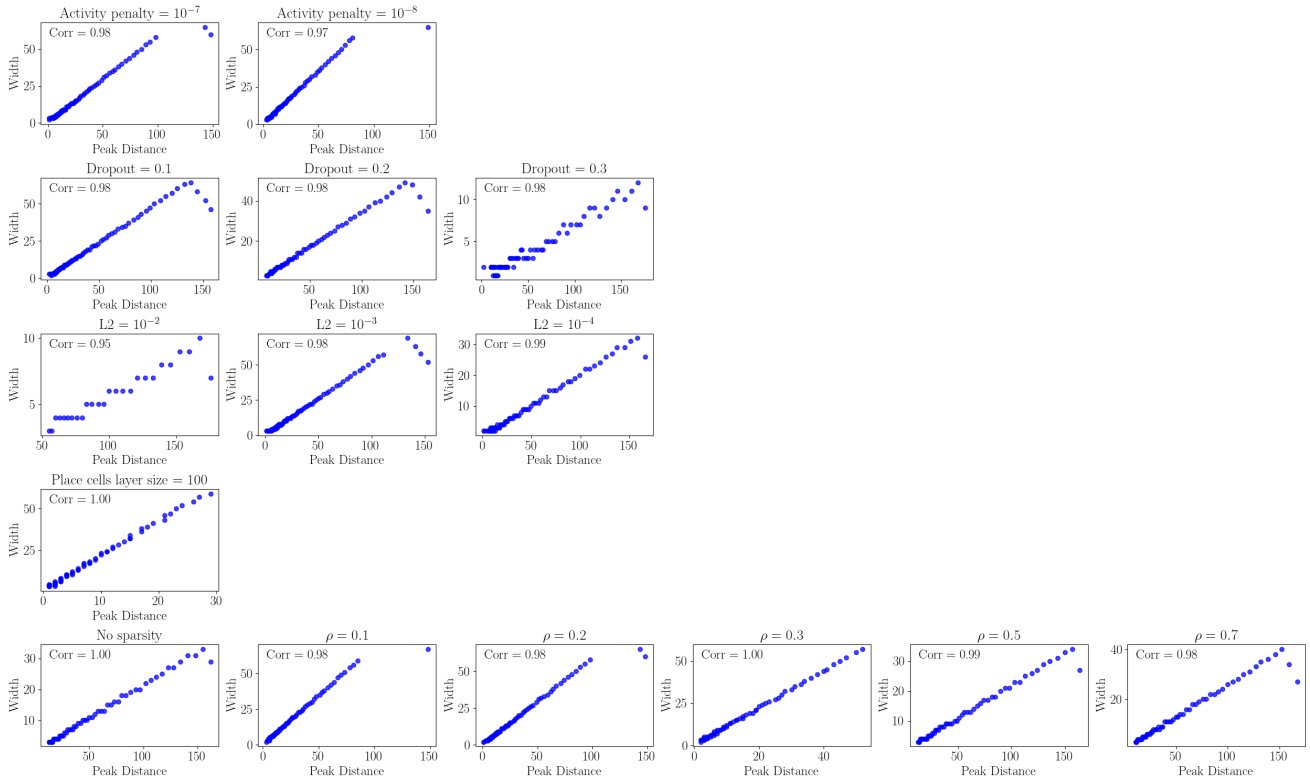


Figure 10: Relationship between place field widths and peak distances for different values of the hyperparameters for autoencoder with local and shared connections. Each row corresponds to a sweep over a different parameter. Note that some hyperparameters values resulted in no place cells and those are not shown here (see Section 2.4 for all the values of the hyperparameters). The results shown here are correspond to one of the three seeds. Correlation coefficient is indicated in the upper left corner. The Strong correlation between width and peak distances is broken only at the edges, where the shared connectivity pattern is incomplete. For average number of place cells and average correlation coefficient see Table 4.

# 4 Discussion

Extensive efforts have been put into better understanding the computational mechanisms behind hippocampal neurons. Our work builds on similarity observed in neural representations of time and space [Eichenbaum, 2017b, Howard and Eichenbaum, 2015, Ekstrom and Ranganath, 2018, Banquet et al., 2021, Alexander et al., 2020]. Specifically, we investigated whether and under what connectivity contracts self-supervised learning can result in weights analytically derived in a computational model that unifies hippocampal neural representations of space, time and concepts [Howard et al., 2014].

The model from Howard et al. [2014] converts representations of time into representations of other variables for which a temporal derivative is available. Thus, representations of time can be converted into representations of distance using a velocity signal. The key property of the model is that it results in scale-invariant representations, consistent with Weber's law. To construct the representation of time, the model requires a spatial derivative that resembles lateral inhibition. We found that self-supervised learning can indeed give rise to such a connectivity profile (Figure 11).

The connectivity profile did not emerge until we restricted weights to be local, and imposed sharing of the weights across the neurons. Specifically, in an autoencoder without weight constraints, with all to all connections (all neurons in the leaky integrator layer connecting to all neurons in the $P$ layer),

Table 4
Number of place cells and correlation coefficient between width and peak distance across hyperparameter choices for autoencoder with local connectivity and shared weights. Sparsity coefficient $\rho = 0.3$ unless specified otherwise.

| **Type** | **Value** | **Corr. Coeff. (CI)** | **# Place Cells (±SEM)** |
|---|---|---|---|
| Activity Penalty | $10^{-7}$ | 0.97 (0.96, 0.98) | 46.0 ± 23.07 |
| | $10^{-8}$ | 0.96 (0.93, 0.97) | 24.67 ± 20.80 |
| | $10^{-9}$ | $< 5$ Place Cells | 0 ± 0 |
| Dropout | 0.1 | 0.98 (0.97, 0.99) | 37.33 ± 21.95 |
| | 0.2 | 0.99 (0.98, 0.99) | 65.33 ± 3.53 |
| | 0.3 | 0.98 (0.96, 0.99) | 20.67 ± 20.67 |
| L2 | $10^{-2}$ | 0.95 (0.89, 0.98) | 9.0 ± 8.50 |
| | $10^{-3}$ | 0.98 (0.97, 0.99) | 65.33 ± 3.84 |
| | $10^{-4}$ | 0.99 (0.99, 1.00) | 23.67 ± 23.67 |
| $\rho$ | 0.1 | 0.99 (0.98, 0.99) | 42.67 ± 21.96 |
| | 0.2 | 0.97 (0.96, 0.98) | 45.0 ± 22.50 |
| | 0.3 | 0.97 (0.96, 0.98) | 46.33 ± 23.21 |
| | 0.5 | 1.00 (1.00, 1.00) | 24.0 ± 21.55 |
| | 0.7 | 1.00 (0.99, 1.00) | 23.4 ± 14.37 |
| Size | 50 | $< 5$ Place Cells | 0 ± 0 |
| | 100 | 0.98 (0.98, 0.99) | 64.0 ± 2.52 |
| No Sparsity | — | 1.00 (1.00, 1.00) | 43.33 ± 21.86 |

the reconstruction loss was reduced, but the neural activity in the $P$ layer in general did not resemble scale-invariant place cells. This did not change after making the architecture overcomplete (having more neurons in the $P$ layer than in the leaky integrator layer) or undercomplete (having more neurons in the leaky integrator layer than in the $P$ layer), as shown in Figure A5.

Enforcing local connectivity between the leaky integrator layer and the $P$ layer (Figure 2A) gave rise to place cells, but with unrealistically narrow tuning curves (Figure 7). However, when local connectivity used shared weights, such that each neuron in the leaky integrator layer projected with the same but translated connectivity pattern to the $P$ layer (Figure 2C) emerging place cells resembled those from neural recordings (e.g., [Sheehan et al., 2021] in linear track and [O'Keefe and Burgess, 1996] in 2-D environments).

Shared weights enable the network to do path integration on tracks longer than those used during training (Figure 15). We also showed that in the 2D environment, using the weights resulting from training on a linear track and modulating the outputs with the animal's angle results in spatially tuned cells that resemble 2D place cells and object vector cells, similar to ones reported in Høydal et al. [2019], Deshmukh and Knierim [2013]. In this context, the animal's angle is defined relative to the log-polar space with the center of the space representing the origin and value of the angle corresponding to difference of the animal's angle relative to north.

The current model relies on the presence of cells that decay exponentially as a function of elapsed time. Recent studies found such neurons in the Lateral Entorhinal Cortex (LEC) of rodents and primates and named them temporal context cells [Bright et al., 2020, Tsao et al., 2018]. Therefore, if

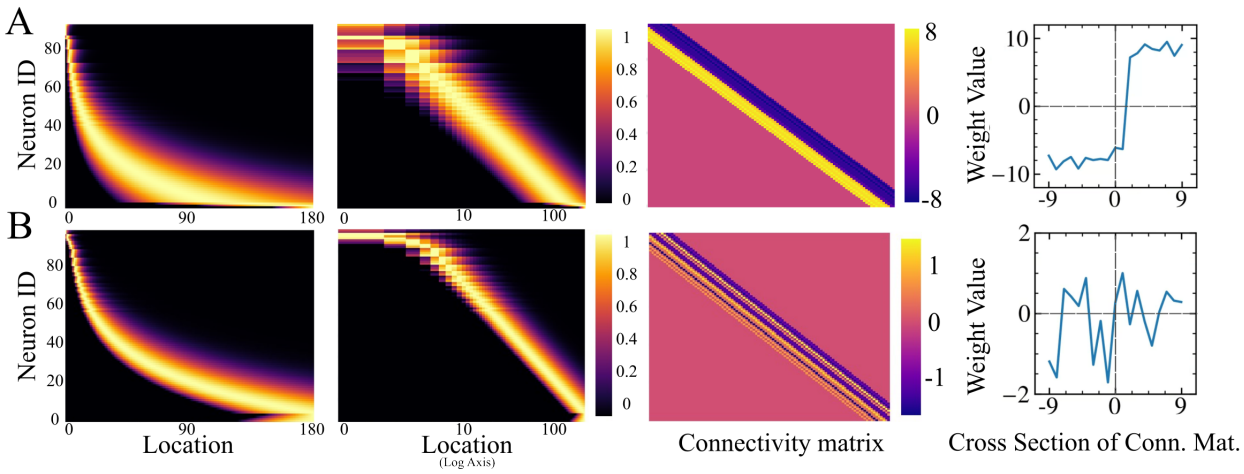


Figure 11: Place cells show scale-invariant properties when weights are local and shared, and time constants are geometrically spaced. The first two columns show the activity of neurons in the $P$ layer with linear (first column) and logarithmically (second column) spaced x-axis. Equal with and uniform spacing of units in the second column indicates the scale invariance. The third column shows the learned connectivity matrix between the leaky integrators and cells in the $P$ layer – note that the weights are local and shared. The fourth column shows the cross-section of the connectivity matrix displaying the weights between a single neuron in the leaky integrator layer and the neurons it is connected to within the $P$ layer. Because the weights are shared, this connectivity pattern is the same for all neurons in the leaky integrator layer. Note that the weights switch signs consistent with the idea that they are implementing an approximation of a spatial derivative. A) Network trained using a sparsity parameter of 0.7. B) Network trained using a sparsity parameter of 0.1. Other regularization terms (dropout, $L_2$ and activity penalty) were set to 0, same as in Figure 9A).

LEC projections to the hippocampus go through the connection matrix $\mathbf{L_1}$, they would give rise to time cells. If the LEC signal were modulated by velocity, then the hippocampal activity would include place cells.

For plausibility of our framework, it is essential that neural firing can change with a wide range of time constants, and that those time constants can be modulated by external signals, in particular the speed of the animal. Previous work has addressed computational mechanisms that could gives rise to neurons with such properties [Tiganj et al., 2015, Liu et al., 2019]. That work focused on calcium-controlled cation (CAN) current and demonstrated that when the amount of calcium leaving the cell during an interspike interval is larger than the calcium influx during a spike, the overall decay in calcium concentration can be exponential. This is shown to result in exponentially decaying firing rate and with realistic biophysical values, the time constants can range up to several minutes. Tiganj et al. [2015] have also demonstrated that the time constants can be modulated with biologically plausible mechanisms including external signal (e.g., neuromodulators) that can impact the time constant of calcium clearance, the maximum conductivity of the CAN current channels, the amount of calcium influx during spikes or the charge needed to cause each spike. Specifically, based on a large body of previous work on modulation of CAN current by the acetylcholine level and changes in persistent firing [Yoshida and Hasselmo, 2009, Yoshida et al., 2012, Fransen et al., 2002, Fransén et al., 2006, Egorov et al., 2002], the authors proposed that cholinergic modulation can influence the amount of charge necessary to generate an action potential, therefore providing means for modulating the time constants. The time course of cholinergic modulation depends on how fast the concentration of acetylcholine can change and on the

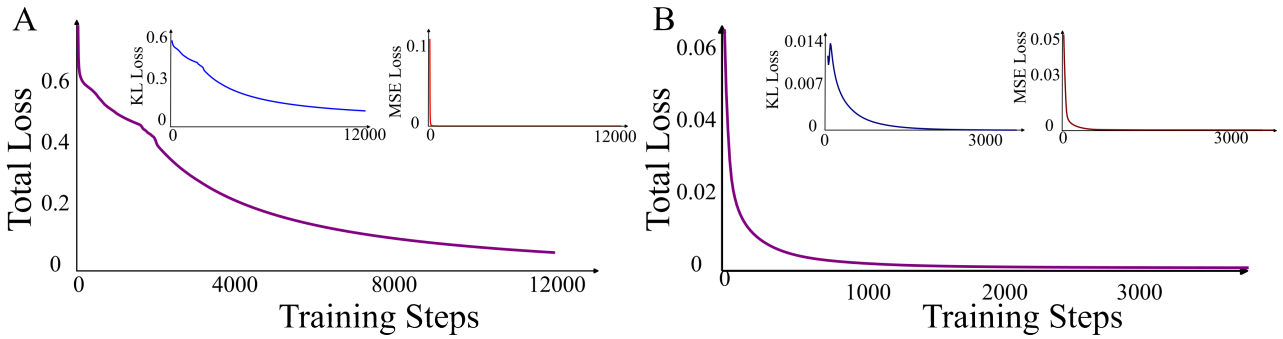


Figure 12: Total loss values during training with local (A) and shared weights (B). Most of the optimization time is spent on lowering the KL loss, as indicated by insets showing KL and MSE loss separately.

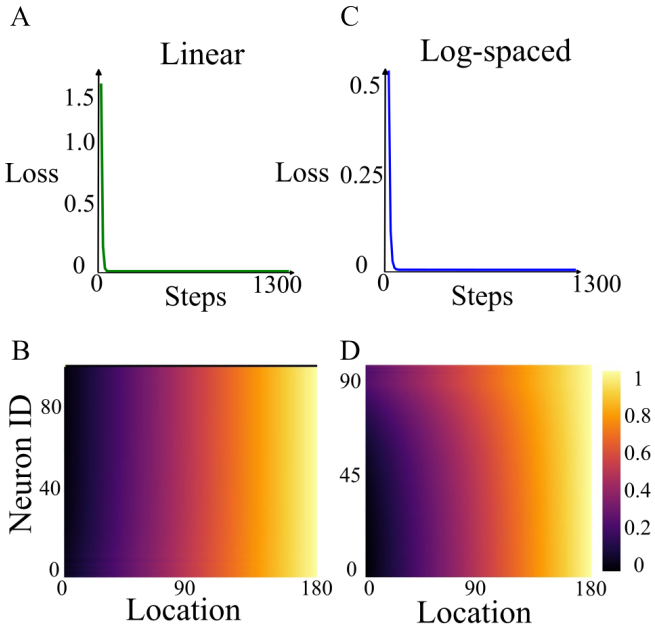


Figure 13: Using velocity as an input to the network rather than a modulator of recurrent weights does not give rise to place cells. A-B) The network successfully minimizes the loss and reconstructs the inputs, but this does not give rise to place cells (B). C-D) logarithmic spacing of the time constants $1/s$ from Equation (14) does not change the results: inputs are reconstructed with high sparsity, but place cells do not emerge (D). All parameters were the same as in Figure 11B, specifically we did not use dropout, and $L_2$ and activity penalty were set to 0, while sparsity loss was 0.7.

time course of muscarinic receptor activation. If the acetylcholine level can be affected by the speed of the animal, then such mechanism would directly translate into modulation of the firing rate (see discussion in Tiganj et al. [2015] for more examples of possible modulatory mechanisms).

Sparsity and local connectivity with lateral inhibition are common in computational models of neural circuits. In particular, sparsity has been a common assumption underlying neural coding across the brain [Olshausen and Field, 1996, Chen et al., 2018, Foldiak, 2003, Lundqvist et al., 2018]. Similarly, the existence of local connectivity with lateral inhibition has also been widely assumed and often associated with sparsity [Yu et al., 2014, Munoz and Istvan, 1998, Xing and Gerstein, 1996, Burke et al., 2017]. The

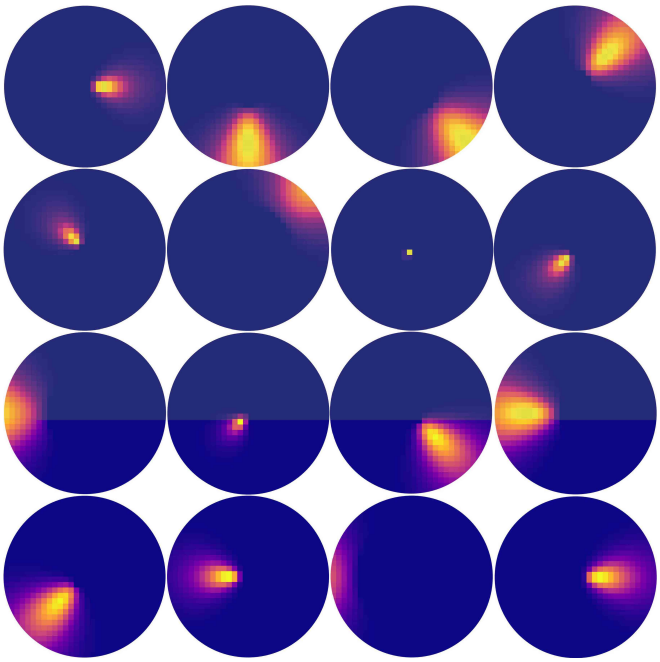

Figure 14: Scale invariance in the radial direction combined with angular tuning gives rise to log-polar tuning curves. Each plot shows an example of a place cell formed in the 2-D model. Color represents the average activity within a 4 by 4 square in the environment (scale is the same as in Figure 3) normalized between 0 (dark blue) and 1 (bright yellow).

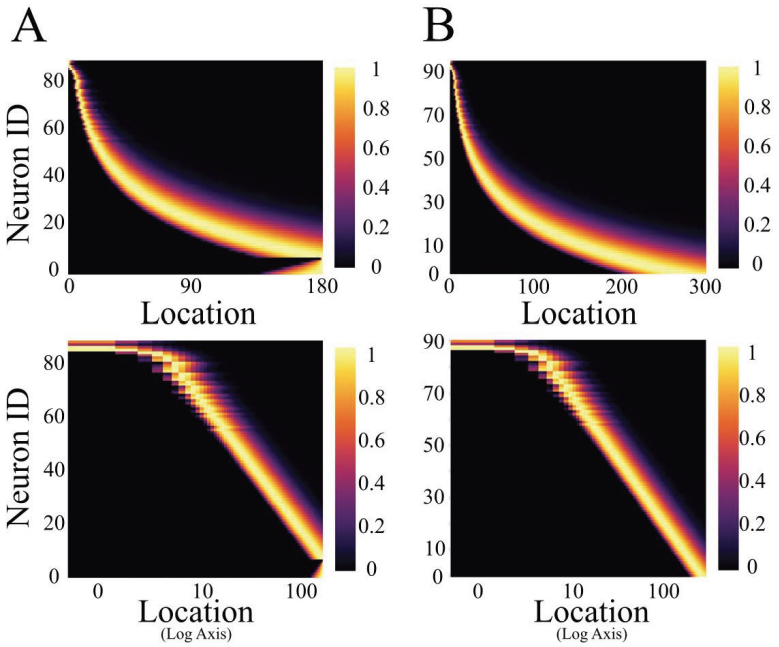


Figure 15: Evaluating place cells on a track longer than the training track. Due to shared weights, once trained, place cells maintain scale invariance across unseen track lengths. A) Place cells after training on a 180 cm long track. B) Place cells after evaluation on a 300 cm long track. Top row: Linear scale. Bottom row: Logarithmic scale.

fact that these constraints were necessary for the emergence of scale invariance suggests a potentially important role of evolution in shaping the connectivity patterns into local and repetitive across neurons (shared weights).

In recent years, several other models of spatially tuned cells have emerged. Cueva and Wei [2018] have used recurrent networks fed with velocity inputs to generate grid and border cells. Vijayabaskaran and Cheng [2022] showed that place cells and grid cells emerge in a model trained on goal-directed navigation. Benna and Fusi [2021] showed that by giving the angle and distance from the wall to the network and imposing sparsity on the inputs, place cells can form, and their shape changes depending on the previous movement history of the animal. Aziz et al. [2022] used a combination of path integration and vision inputs to train an autoencoder, resulting in spatially modulated cells, including place cells and grid cells. While these models were remarkably successful in giving rise to spatially modulated cells, their focus was not on constructing *scale-invariant* time and place cells.

In complex real-world environments, factors beyond speed are likely to affect the neural correlates of spatial position. In particular, visual inputs may be used to correct the spatial representation, for instance through graph relaxation [Erdem et al., 2015]. This could disrupt scale invariance since the variance in the estimate of the traveled distance would become dependent on visual inputs and might no longer scale with the mean. This would have a similar effect on scale invariance as counting or having access to a watch in interval reproduction tasks [Fetterman and Killeen, 1990]—counting improves performance and breaks scale invariance. Nevertheless, understanding the neural mechanisms of path integration remains critical for comprehending how spatial representations are formed and maintained, especially since path integration provides a foundational process upon which additional sensory cues can be utilized. Path integration also has valuable clinical relevance since deficits in path integration have been linked to neural degeneration related to Alzheimer's disease [Segen et al., 2022, Castegnaro et al., 2023].

Our experiments focused on the spatial domain, however the equivalent results can be generated in the temporal domain. Setting velocity input to 1 converts the place cells into time cells – the spatial domain can be seen as a generalization of the model, while the temporal domain is a special case with a velocity equal to 1. Therefore, learning the weights could happen in either temporal or spatial domain. In the temporal domain, the onset of a salient stimulus triggers a set of neurons with exponentially decaying activity with a spectrum of time constants. The activity of these neurons is reconstructed through the autoencoder circuit. With temporal input, the latent space of the autoencoder would give rise to scale-invariant time cells. The same weights can then be used for spatial representations, and they will give rise to scale-invariant place cells as long as the velocity is used to modulate the decay rate of the exponentially decaying cells.

Since in our framework the only difference between time and place cells is in modulation by velocity, it implies two possible relationships between time and place cells. First, it could be that these are the same neurons, whose firing can be tuned to time or position depending on the gating of the velocity inputs, consistent with the observations reported in Kraus et al. [2013]. It is also possible that velocity modulation of some neurons is permanent and not gated, implying that downstream neurons would always be tuned to the spatial location. Analogously, it is possible that some of the leaky integrators do not receive any velocity modulation, resulting in exponential decay as a function of time and downstream time cells. It is worth noting that these possibilities are not mutually exclusive and that some neurons might by switching between representations of space and time while others might be specialized and consistently tuned to either space or time.

Our work illustrates that self-supervised learning can, under specific constraints, lead to powerful neural representations of time and space, shedding light on the mechanisms of memory formation in the entorhinal-hippocampal circuit. Future studies could attempt to integrate grid cells in the same framework with the scale-invariant time and place cells.

# Acknowledgement

We gratefully acknowledge support from the National Institutes of Health's National Institute on Aging, grant 5R01AG076198-02.

# References

Andrew S Alexander, Jennifer C Robinson, Holger Dannenberg, Nathaniel R Kinsky, Samuel J Levy, William Mau, G William Chapman, David W Sullivan, and Michael E Hasselmo. Neurophysiological coding of space and time in the hippocampus, entorhinal cortex, and retrosplenial cortex. *Brain and Neuroscience Advances*, 4:2398212820972871, 2020.

Dmitriy Aronov, Rhino Nevers, and David W Tank. Mapping of a non-spatial dimension by the hippocampal–entorhinal circuit. *Nature*, 543(7647):719–722, 2017.

Azra Aziz, Peesapati SS Sreeharsha, Rohan Natesh, and Vaddadhi S Chakravarthy. An integrated deep learning-based model of spatial cells that combines self-motion with sensory information. *Hippocampus*, 32(10):716–730, 2022.

Fuat Balcı and David Freestone. The peak interval procedure in rodents: a tool for studying the neurobiological basis of interval timing and its alterations in models of human disease. *Bio-protocol*, 10(17):e3735–e3735, 2020.

Andrea Banino, Caswell Barry, Benigno Uria, Charles Blundell, Timothy Lillicrap, Piotr Mirowski, Alexander Pritzel, Martin J Chadwick, Thomas Degris, Joseph Modayil, et al. Vector-based navigation using grid-like representations in artificial agents. *Nature*, 557(7705):429–433, 2018.

Jean-Paul Banquet, Philippe Gaussier, Nicolas Cuperlier, Vincent Hok, Etienne Save, Bruno Poucet, Mathias Quoy, and Sidney I Wiener. Time as the fourth dimension in the hippocampus. *Progress in Neurobiology*, 199:101920, 2021.

Timothy EJ Behrens, Timothy H Muller, James CR Whittington, Shirley Mark, Alon B Baram, Kimberly L Stachenfeld, and Zeb Kurth-Nelson. What is a cognitive map? organizing knowledge for flexible behavior. *Neuron*, 100(2):490–509, 2018.

Marcus K Benna and Stefano Fusi. Place cells may simply be memory cells: Memory compression leads to spatial tuning and history dependence. *Proceedings of the National Academy of Sciences*, 118(51): e2018422118, 2021.

Tale L Bjerknes, Nenitha C Dagslott, Edvard I Moser, and May-Britt Moser. Path integration in place cells of developing rats. *Proceedings of the National Academy of Sciences*, 115(7):E1637–E1646, 2018.

Ian M Bright, Miriam LR Meister, Nathanael A Cruzado, Zoran Tiganj, Elizabeth A Buffalo, and Marc W Howard. A temporal record of the past with a spectrum of time constants in the monkey entorhinal cortex. *Proceedings of the National Academy of Sciences*, 117(33):20274–20283, 2020.

Catalin V Buhusi and Warren H Meck. What makes us tick? functional and neural mechanisms of interval timing. *Nature reviews neuroscience*, 6(10):755–765, 2005.

Catalin V Buhusi, Dyana Aziz, David Winslow, Rickey E Carter, Joshua E Swearingen, and Mona C Buhusi. Interval timing accuracy and scalar timing in c57bl/6 mice. *Behavioral neuroscience*, 123(5): 1102, 2009.

Dennis A Burke, Horacio G Rotstein, and Veronica A Alvarez. Striatal local circuitry: a new framework for lateral inhibition. *Neuron*, 96(2):267–284, 2017.

György Buzsáki and David Tingley. Space and time: the hippocampus as a sequence generator. *Trends in cognitive sciences*, 22(10):853–869, 2018.

Rui Cao, John H Bladon, Stephen J Charczynski, Michael E Hasselmo, and Marc W Howard. Internally generated time in the rodent hippocampus is logarithmically compressed. *Elife*, 11:e75353, 2022.

Andrea Castegnaro, Zilong Ji, Katarzyna Rudzka, Dennis Chan, and Neil Burgess. Overestimation in angular path integration precedes alzheimer's dementia. *Current Biology*, 33(21):4650–4661, 2023.

Guifen Chen, John A King, Neil Burgess, and John O'Keefe. How vision and movement combine in the hippocampal place code. *Proceedings of the National Academy of Sciences*, 110(1):378–383, 2013.

Yubei Chen, Dylan Paiton, and Bruno Olshausen. The sparse manifold transform. *Advances in neural information processing systems*, 31, 2018.

John Conklin and Chris Eliasmith. A controlled attractor network model of path integration in the rat. *Journal of computational neuroscience*, 18:183–203, 2005.

Nathanael A Cruzado, Zoran Tiganj, Scott L Brincat, Earl K Miller, and Marc W Howard. Conjunctive representation of what and when in monkey hippocampus and lateral prefrontal cortex during an associative memory task. *Hippocampus*, 30(12):1332–1346, 2020.

Christopher J Cueva and Xue-Xin Wei. Emergence of grid-like representations by training recurrent neural networks to perform spatial localization. In *International Conference on Learning Representations*, 2018.

Stanislas Dehaene. The neural basis of the weber–fechner law: a logarithmic mental number line. *Trends in cognitive sciences*, 7(4):145–147, 2003.

Sachin S Deshmukh and James J Knierim. Influence of local objects on hippocampal representations: Landmark vectors and memory. *Hippocampus*, 23(4):253–267, 2013.

Alexei V Egorov, Bassam N Hamam, Erik Fransén, Michael E Hasselmo, and Angel A Alonso. Graded persistent activity in entorhinal cortex neurons. *Nature*, 420(6912):173–178, 2002.

Howard Eichenbaum. Memory on time. *Trends in cognitive sciences*, 17(2):81–88, 2013.

Howard Eichenbaum. On the integration of space, time, and memory. *Neuron*, 95(5):1007–1018, 2017a.

Howard Eichenbaum. Time (and space) in the hippocampus. *Current opinion in behavioral sciences*, 17:65–70, 2017b.

Arne D Ekstrom and Charan Ranganath. Space, time, and episodic memory: The hippocampus is all over the cognitive map. *Hippocampus*, 28(9):680–687, 2018.

Uğur M Erdem, Michael J Milford, and Michael E Hasselmo. A hierarchical model of goal directed navigation selects trajectories in a visual environment. *Neurobiology of learning and memory*, 117: 109–121, 2015.

Ariane S Etienne. Navigation of a small mammal by dead reckoning and local cues. *Current Directions in Psychological Science*, 1(2):48–52, 1992.

Ariane S Etienne, Roland Maurer, and Valérie Séguinot. Path integration in mammals and its interaction with visual landmarks. *The Journal of experimental biology*, 199(1):201–209, 1996.

William A Falcon. Pytorch lightning. *GitHub*, 3, 2019.

J Gregor Fetterman and Peter R Killeen. A componential analysis of pacemaker-counter timing systems. *Journal of Experimental Psychology: Human Perception and Performance*, 16(4):766, 1990.

Peter Foldiak. Sparse coding in the primate cortex. *The handbook of brain theory and neural networks*, 2003.

Erik Fransen, Angel A Alonso, and Michael E Hasselmo. Simulations of the role of the muscarinic-activated calcium-sensitive nonspecific cation currentincm in entorhinal neuronal activity during delayed matching tasks. *Journal of neuroscience*, 22(3):1081–1097, 2002.

Erik Fransén, Babak Tahvildari, Alexei V Egorov, Michael E Hasselmo, and Angel A Alonso. Mechanism of graded persistent cellular activity of entorhinal cortex layer v neurons. *Neuron*, 49(5):735–746, 2006.

John Gibbon. Scalar expectancy theory and weber's law in animal timing. *Psychological review*, 84(3): 279, 1977.

Katalin M Gothard, Kari L Hoffman, Francesco P Battaglia, and Bruce L McNaughton. Dentate gyrus and ca1 ensemble activity during spatial reference frame shifts in the presence and absence of visual input. *Journal of Neuroscience*, 21(18):7284–7292, 2001.

Stephen Grossberg and Nestor A Schmajuk. Neural dynamics of adaptive timing and temporal discrimination during associative learning. *Neural networks*, 2(2):79–102, 1989.

Mathew A Harris and Thomas Wolbers. Ageing effects on path integration and landmark navigation. *Hippocampus*, 22(8):1770–1780, 2012.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

James R Hinman, Mark P Brandon, Jason R Climer, G William Chapman, and Michael E Hasselmo. Multiple running speed signals in medial entorhinal cortex. *Neuron*, 91(3):666–679, 2016.

Marc W Howard and Howard Eichenbaum. Time and space in the hippocampus. *Brain research*, 1621: 345–354, 2015.

Marc W Howard and Michael E Hasselmo. Cognitive computation using neural representations of time and space in the laplace domain. *arXiv preprint arXiv:2003.11668*, 2020.

Marc W Howard and Michael J Kahana. A distributed representation of temporal context. *Journal of mathematical psychology*, 46(3):269–299, 2002.

Marc W Howard, Christopher J MacDonald, Zoran Tiganj, Karthik H Shankar, Qian Du, Michael E Hasselmo, and Howard Eichenbaum. A unified mathematical framework for coding time, space, and sequences in the hippocampal region. *Journal of Neuroscience*, 34(13):4692–4707, 2014.

Marc W Howard, Karthik H Shankar, William R Aue, and Amy H Criss. A distributed representation of internal time. *Psychological review*, 122(1):24, 2015a.

Marc W Howard, Karthik H Shankar, and Zoran Tiganj. Efficient neural computation in the laplace domain. In *CoCo@ NIPS*, 2015b.

Øyvind Arne Høydal, Emilie Ranheim Skytøen, Sebastian Ola Andersson, May-Britt Moser, and Edvard I Moser. Object-vector coding in the medial entorhinal cortex. *Nature*, 568(7752):400–404, 2019.

Ravikrishnan P Jayakumar, Manu S Madhav, Francesco Savelli, Hugh T Blair, Noah J Cowan, and James J Knierim. Recalibration of path integration in hippocampal place cells. *Nature*, 566(7745): 533–537, 2019.

Eric R Kandel, James H Schwartz, Thomas M Jessell, Steven Siegelbaum, A James Hudspeth, Sarah Mack, et al. *Principles of neural science*, volume 4. McGraw-hill New York, 2000.

Benjamin J Kraus, Robert J Robinson, John A White, Howard Eichenbaum, and Michael E Hasselmo. Hippocampal "time cells": time versus path integration. *Neuron*, 78(6):1090–1101, 2013.

Emilio Kropff, James E Carmichael, May-Britt Moser, and Edvard I Moser. Speed cells in the medial entorhinal cortex. *Nature*, 523(7561):419–424, 2015.

Markus Lappe, Michael Jenkin, and Laurence R Harris. Travel distance estimation from visual motion by leaky path integration. *Experimental brain research*, 180:35–48, 2007.

Yue Liu, Zoran Tiganj, Michael E Hasselmo, and Marc W Howard. A neural microcircuit model for a scalable scale-invariant representation of time. *Hippocampus*, 29(3):260–274, 2019.

Mikael Lundqvist, Pawel Herman, and Earl K Miller. Working memory: delay activity, yes! persistent activity? maybe not. *Journal of neuroscience*, 38(32):7013–7019, 2018.

Christopher J MacDonald, Kyle Q Lepage, Uri T Eden, and Howard Eichenbaum. Hippocampal "time cells" bridge the gap in memory for discontiguous events. *Neuron*, 71(4):737–749, 2011.

Sahaj Singh Maini, James Mochizuki-Freeman, Chirag Shankar Indi, Brandon G Jacques, Per B Sederberg, Marc W Howard, and Zoran Tiganj. Representing latent dimensions using compressed number lines. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10. IEEE, 2023.

Bruce L Mcnaughton, B Leonard, and L Chen. Cortical-hippocampal interactions and cognitive mapping: A hypothesis based on reintegration of the parietal and inferotemporal pathways for visual processing. *Psychobiology*, 17(3):230–235, 1989.

Bruce L McNaughton, Carol A Barnes, Jason L Gerrard, Katalin Gothard, Min W Jung, James J Knierim, H Kudrimoti, Y Qin, WE Skaggs, M Suster, et al. Deciphering the hippocampal polyglot: the hippocampus as a path integration system. *Journal of Experimental Biology*, 199(1):173–185, 1996.

H Mittelstaedt and ML Mittelstaedt. Homing by path integration. *Avian navigation*, pages 290–297, 1982.

James Mochizuki-Freeman, Sahaj Singh Maini, and Zoran Tiganj. Characterizing neural activity in cognitively inspired rl agents during an evidence accumulation task. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 01–09. IEEE, 2023.

Ari S Morcos and Christopher D Harvey. History-dependent variability in population dynamics during evidence accumulation in cortex. *Nature neuroscience*, 19(12):1672–1681, 2016.

L Muessig, J Hauser, TJ Wills, and F Cacucci. Place cell networks in pre-weanling rats show associative memory properties from the onset of exploratory behavior. *Cerebral Cortex*, 26(8):3627–3636, 2016.

Douglas P Munoz and Peter J Istvan. Lateral inhibitory interactions in the intermediate layers of the monkey superior colliculus. *Journal of Neurophysiology*, 79(3):1193–1209, 1998.

Edward H Nieh, Manuel Schottdorf, Nicolas W Freeman, Ryan J Low, Sam Lewallen, Sue Ann Koay, Lucas Pinto, Jeffrey L Gauthier, Carlos D Brody, and David W Tank. Geometry of abstract learned knowledge in the hippocampus. *Nature*, 595(7865):80–84, 2021.

John O'Keefe. Place units in the hippocampus of the freely moving rat. *Experimental neurology*, 51(1): 78–109, 1976.

John O'Keefe and Neil Burgess. Geometric determinants of the place fields of hippocampal neurons. *Nature*, 381(6581):425–428, 1996.

Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

J O'Keefe and L Nadal. The hippocampus as a cognitive map: Oxford university. 1978.

Eva Pastalkova, Vladimir Itskov, Asohan Amarasingham, and Gyorgy Buzsaki. Internally generated cell assembly sequences in the rat hippocampus. *Science*, 321(5894):1322–1327, 2008.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

RD Portugal and Benar Fux Svaiter. Weber-Fechner law and the optimality of the logarithmic scale. *Minds and Machines*, 21(1):73–81, 2011.

Emil L Post. Generalized differentiation. *Transactions of the American Mathematical Society*, 32(4): 723–781, 1930.

Gregory J Quirk, Robert U Muller, and John L Kubie. The firing of hippocampal place cells in the dark depends on the rat's recent experience. *Journal of Neuroscience*, 10(6):2008–2017, 1990.

Florian Raudies, Mark P Brandon, G William Chapman, and Michael E Hasselmo. Head direction is coded more strongly than movement direction in a population of entorhinal neurons. *Brain research*, 1621:355–367, 2015.

A David Redish. Beyond the cognitive map. *The MIT Press, Cambridge*, 1999.

Alexei Samsonovich and Bruce L McNaughton. Path integration and cognitive mapping in a continuous attractor neural network model. *Journal of Neuroscience*, 17(15):5900–5920, 1997.

Francesca Sargolini, Marianne Fyhn, Torkel Hafting, Bruce L McNaughton, Menno P Witter, May-Britt Moser, and Edvard I Moser. Conjunctive representation of position, direction, and velocity in entorhinal cortex. *Science*, 312(5774):758–762, 2006.

Etienne Save, Arnaud Cressant, Catherine Thinus-Blanc, and Bruno Poucet. Spatial firing of hippocampal place cells in blind rats. *Journal of Neuroscience*, 18(5):1818–1826, 1998.

Per B Sederberg, Marc W Howard, and Michael J Kahana. A context-based theory of recency and contiguity in free recall. *Psychological review*, 115(4):893, 2008.

Vladislava Segen, Johnson Ying, Erik Morgan, Mark Brandon, and Thomas Wolbers. Path integration in normal aging and alzheimer's disease. *Trends in cognitive sciences*, 26(2):142–158, 2022.

Karthik H Shankar and Marc W Howard. A scale-invariant internal representation of time. *Neural Computation*, 24(1):134–193, 2012.

Daniel J Sheehan, Stephen Charczynski, Blake A Fordyce, Michael E Hasselmo, and Marc W Howard. A compressed representation of spatial distance in the rodent hippocampus'. *bioRxiv*, pages 2021–02, 2021.

David W Tank and JJ Hopfield. Neural computation by concentrating information in time. *Proceedings of the National Academy of Sciences*, 84(7):1896–1900, 1987.

Jeffrey S Taube, Robert U Muller, and James B Ranck. Head-direction cells recorded from the postsubiculum in freely moving rats. i. description and quantitative analysis. *Journal of Neuroscience*, 10 (2):420–435, 1990.

Zoran Tiganj, Michael E Hasselmo, and Marc W Howard. A simple biophysically plausible model for long time constants in single neurons. *Hippocampus*, 25(1):27–37, 2015.

Zoran Tiganj, Min Whan Jung, Jieun Kim, and Marc W Howard. Sequential firing codes for time in rodent medial prefrontal cortex. *Cerebral Cortex*, 27(12):5663–5671, 2017.

Zoran Tiganj, Jason A Cromer, Jefferson E Roy, Earl K Miller, and Marc W Howard. Compressed timeline of recent experience in monkey lateral prefrontal cortex. *Journal of cognitive neuroscience*, 30(7):935–950, 2018.

Zoran Tiganj, Wei Tang, and Marc Howard. A computational model for simulating the future using a memory timeline. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 43, 2021.

Zoran Tiganj, Inder Singh, Zahra G Esfahani, and Marc W Howard. Scanning a compressed ordered representation of the future. *Journal of Experimental Psychology: General*, 2022.

Albert Tsao, Jørgen Sugar, Li Lu, Cheng Wang, James J Knierim, May-Britt Moser, and Edvard I Moser. Integrating time from experience in the lateral entorhinal cortex. *Nature*, 561(7721):57–62, 2018.

Gray Umbach, Pranish Kantak, Joshua Jacobs, Michael Kahana, Brad E Pfeiffer, Michael Sperling, and Bradley Lega. Time cells in the human hippocampus and entorhinal cortex support episodic memory. *Proceedings of the National Academy of Sciences*, 117(45):28463–28474, 2020.

Sandhiya Vijayabaskaran and Sen Cheng. Navigation task and action space drive the emergence of egocentric and allocentric spatial representations. *PLOS Computational Biology*, 18(10):e1010320, 2022.

Aaron R Voelker and Chris Eliasmith. Improving spiking dynamical networks: Accurate delays, higher-order synapses, and time cells. *Neural computation*, 30(3):569–609, 2018.

Jason T Wilkes. *Reverse first principles: Weber's law and optimality in different senses*. PhD thesis, UNIVERSITY OF CALIFORNIA, SANTA BARBARA, 2015.

Jing Xing and George L Gerstein. Networks with lateral connectivity. i. dynamic properties mediated by the balance of intrinsic excitation and inhibition. *Journal of neurophysiology*, 75(1):184–199, 1996.

Motoharu Yoshida and Michael E Hasselmo. Persistent firing supported by an intrinsic cellular mechanism in a component of the head direction system. *Journal of Neuroscience*, 29(15):4945–4952, 2009.

Motoharu Yoshida, Beate Knauer, and Arthur Jochems. Cholinergic modulation of the can current may adjust neural dynamics for active memory maintenance, spatial navigation and time-compressed replay. *Frontiers in neural circuits*, 6:10, 2012.

Joseph Geno Yoshioka. Weber's law in the discrimination of maze distance by the white rat. *University of California Publications in Psychology*, 1929.

Yuguo Yu, Michele Migliore, Michael L Hines, and Gordon M Shepherd. Sparse coding and lateral inhibition arising from balanced and unbalanced dendrodendritic excitation and inhibition. *Journal of Neuroscience*, 34(41):13701–13713, 2014.

Sijie Zhang, Fabian Schönfeld, Laurenz Wiskott, and Denise Manahan-Vaughan. Spatial representations of place cells in darkness are supported by path integration and border information. *Frontiers in behavioral neuroscience*, 8:222, 2014.

# Appendix

## A.1 Simple autoencoder with no constraints

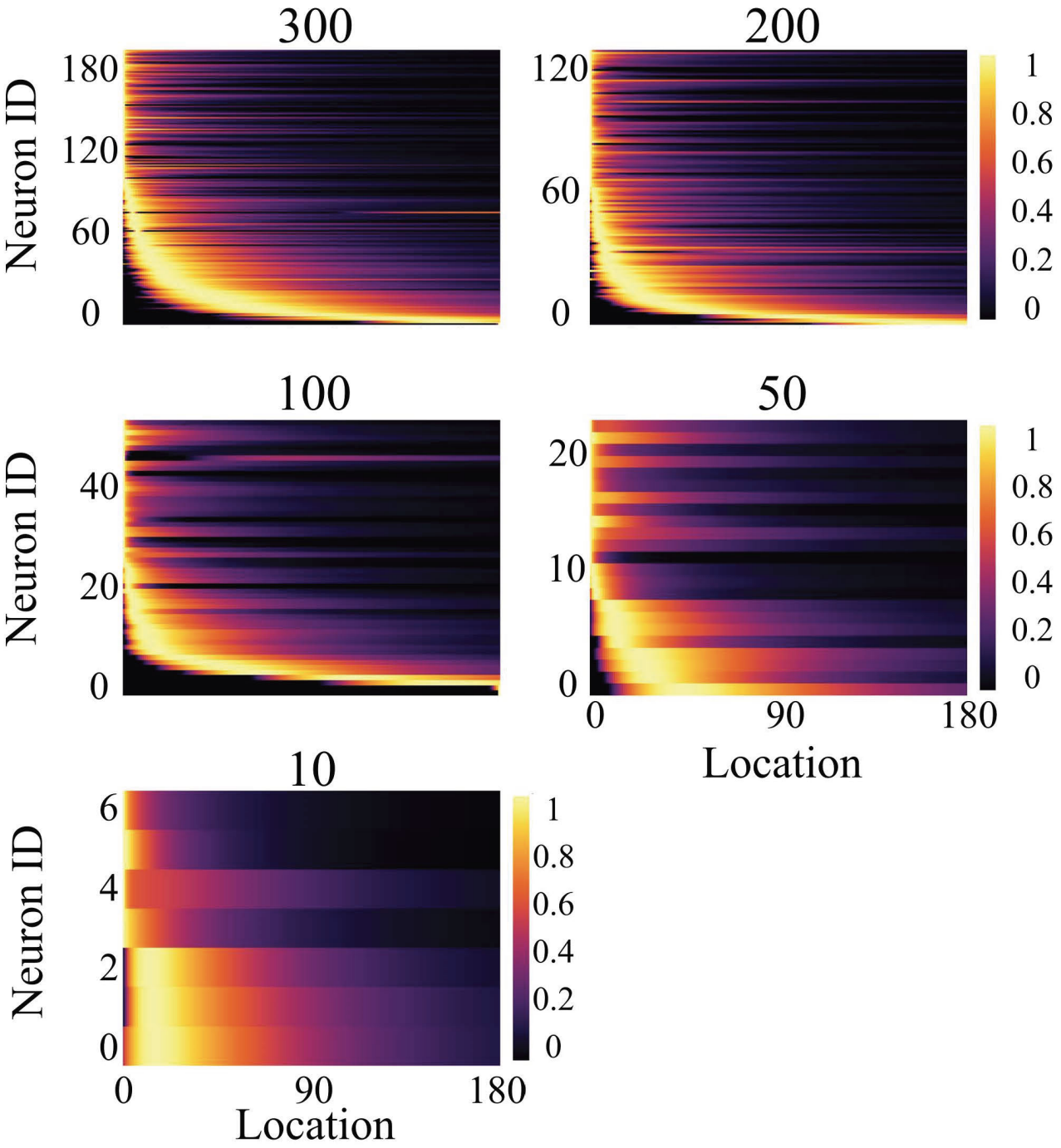


Figure A1: Activity of neurons in the $P$ layer for an autoencoder without sparsity or connectivity constraints. Each panel corresponds to a different number of neurons in the $P$ layer, which is indicated on the top of each panel.

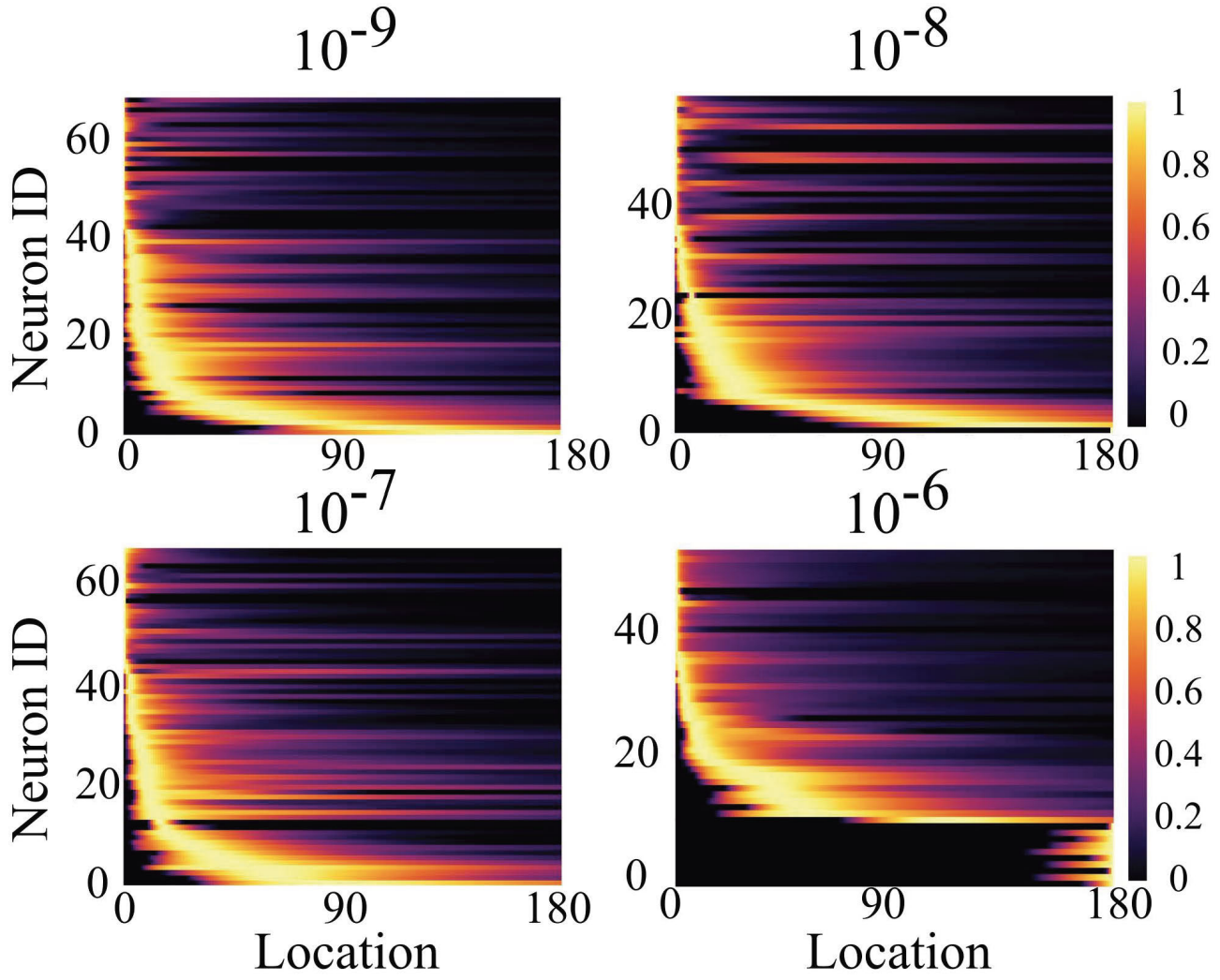


Figure A2: Activity of neurons in the $P$ layer for an autoencoder without sparsity or connectivity constraints. Each panel corresponds to a different activity penalty. Note that small activity penalties were selected since larger activity penalties wipe out the activity in the entire place cell layer.

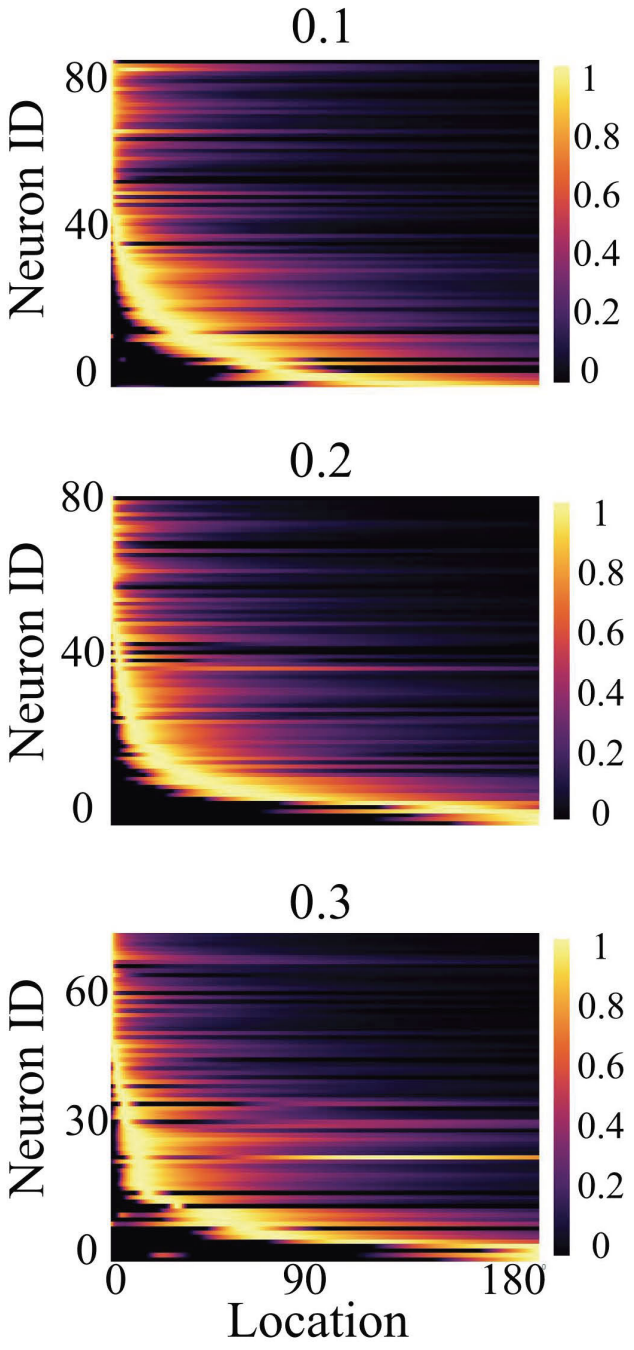


Figure A3: Activity of neurons in the $P$ layer for an autoencoder without sparsity or connectivity constraints. Each panel corresponds to a different value of dropout.

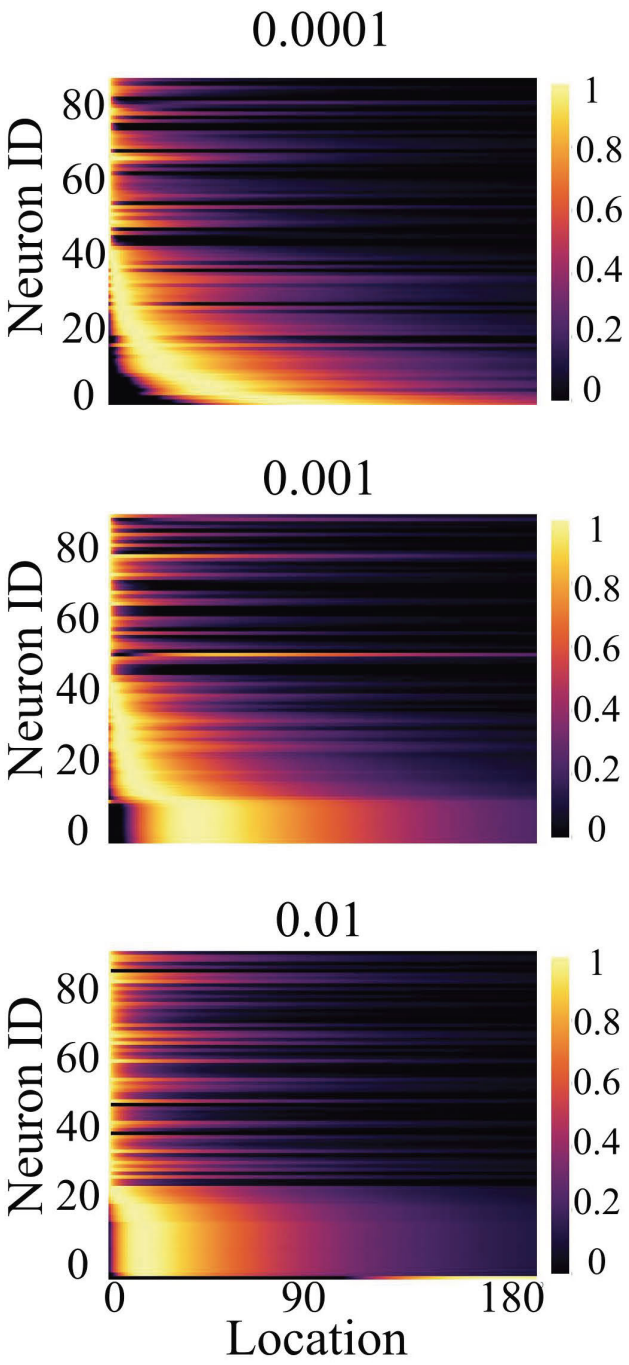


Figure A4: Activity of neurons in the $P$ layer for an autoencoder without sparsity or connectivity constraints. Each panel corresponds to a different value of $L2$ regularization.

## A.2 Sparse autoencoder with no constraints

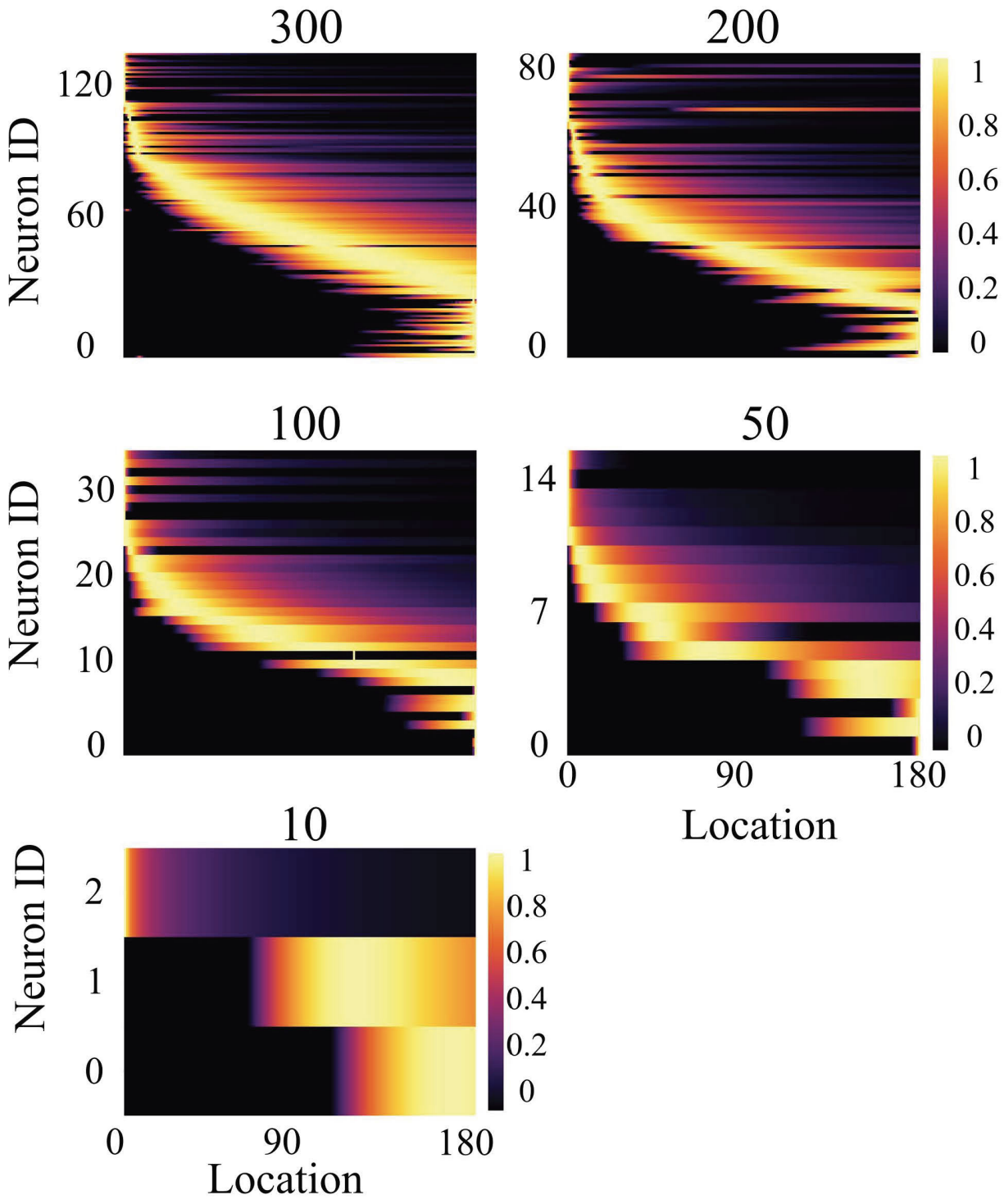


Figure A5: Activity of neurons in the $P$ layer for an autoencoder with sparsity constraint and no connectivity constraints. Each panel corresponds to a different number of neurons in the $P$ layer.

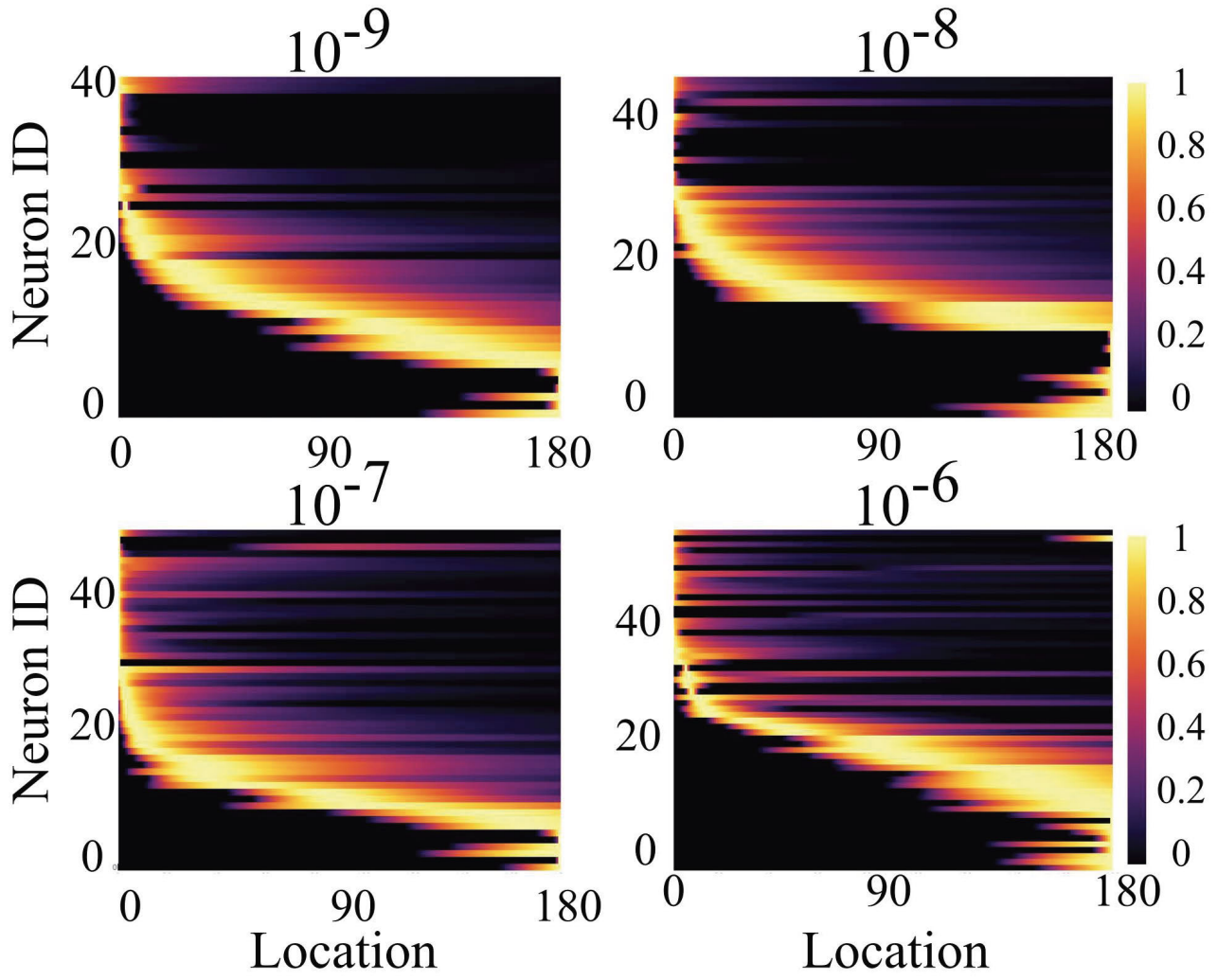


Figure A6: Activity of neurons in the $P$ layer for an autoencoder with sparsity constraint and no connectivity constraints. Each panel corresponds to a different activity penalty.

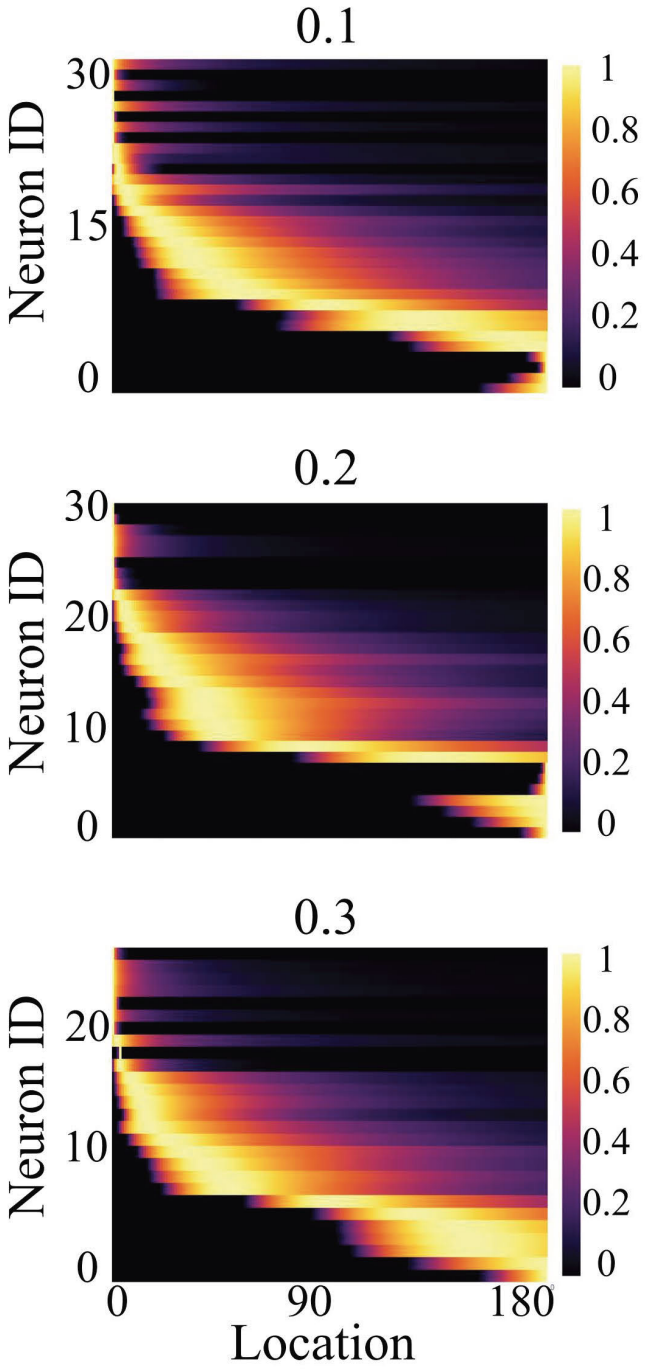


Figure A7: Activity of neurons in the $P$ layer for an autoencoder with sparsity constraint and no connectivity constraints. Each panel corresponds to a different dropout value.

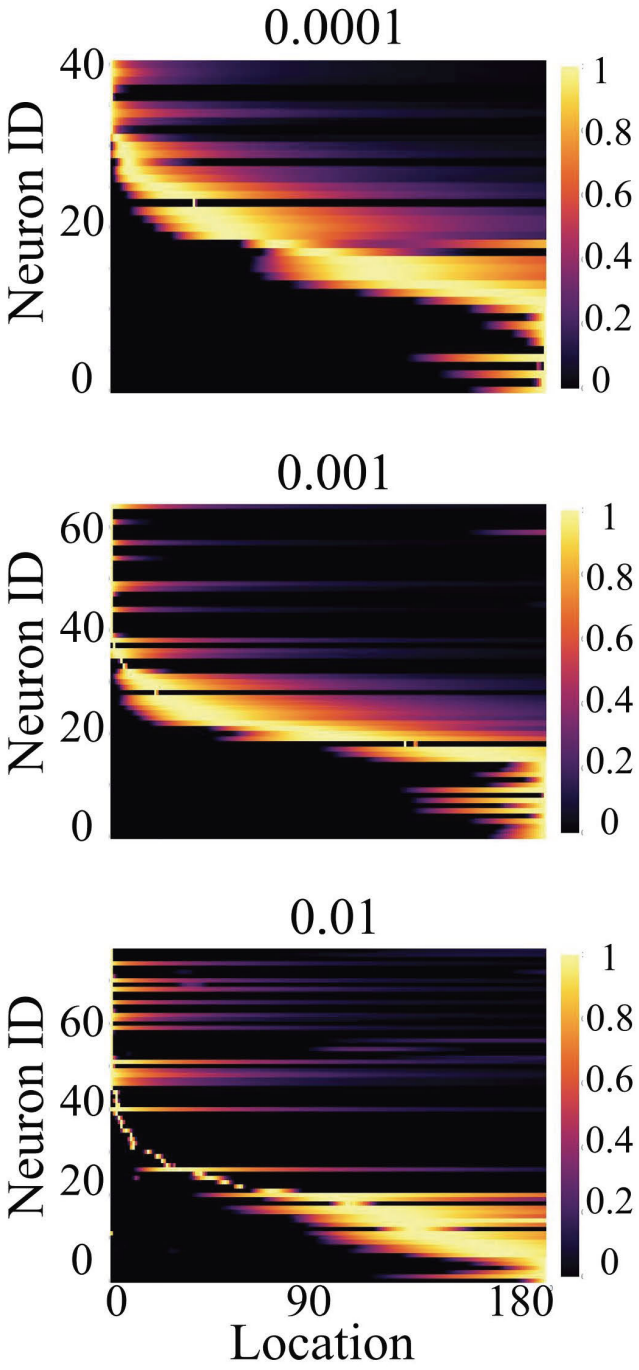


Figure A8: Activity of neurons in the $P$ layer for an autoencoder with sparsity constraint and no connectivity constraints. Each panel corresponds to a different $L2$ regularization value.

### A.3 Sparse autoencoder with local weights

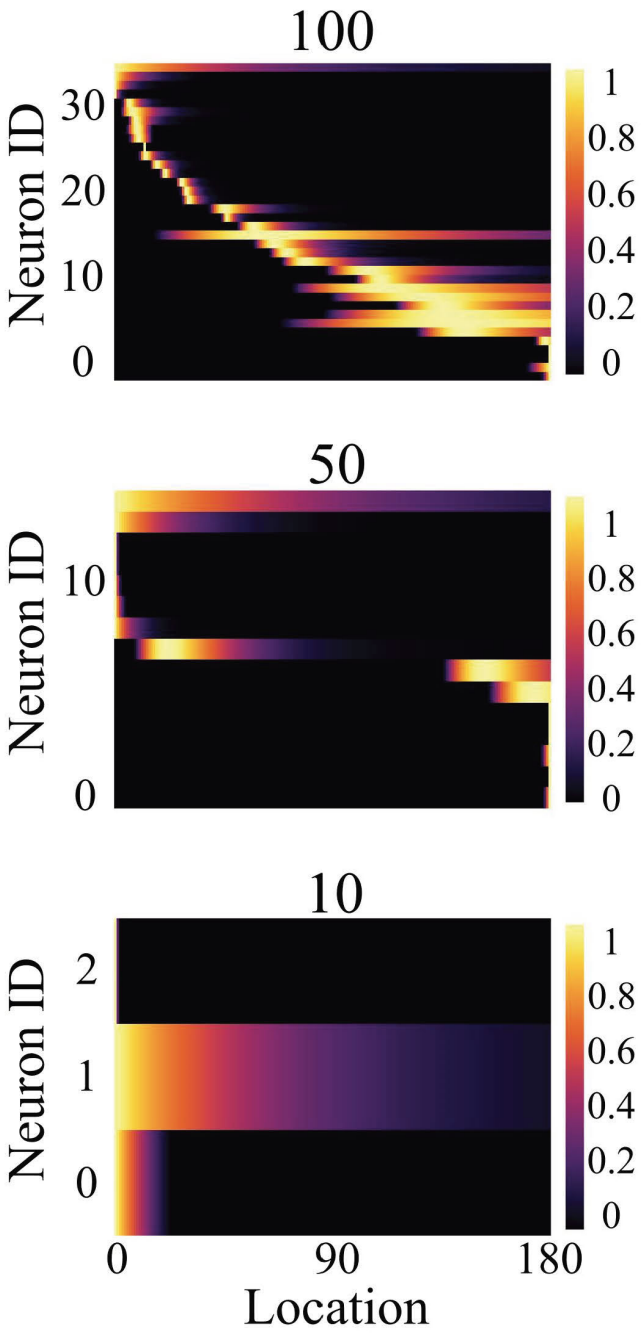


Figure A9: Activity of neurons in the $P$ layer for an autoencoder with sparsity constraint and local weights. Each panel corresponds to a different number of neurons in the $P$ layer.

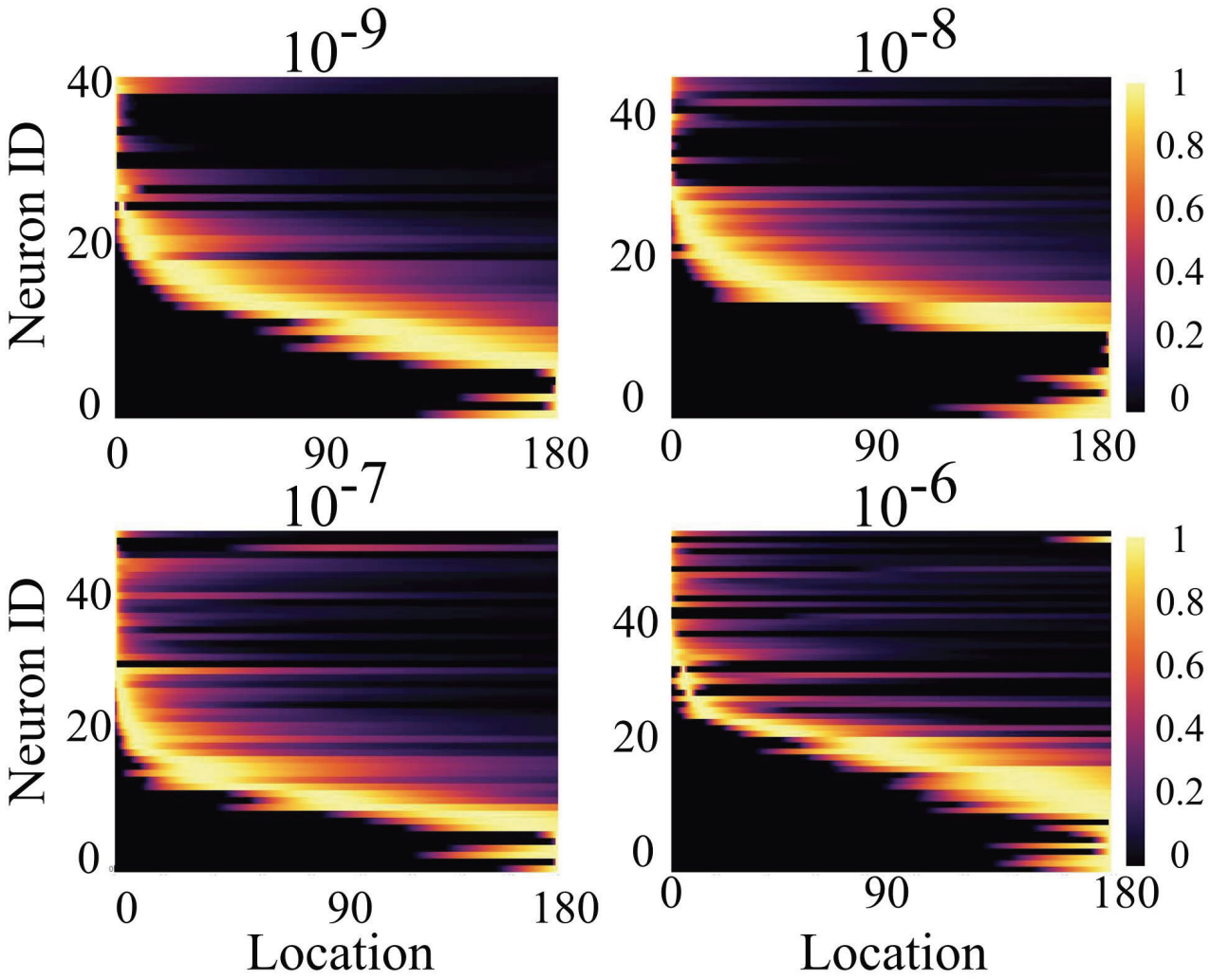


Figure A10: Activity of neurons in the $P$ layer for an autoencoder with sparsity constraint and local weights. Each panel corresponds to a different activity penalty.

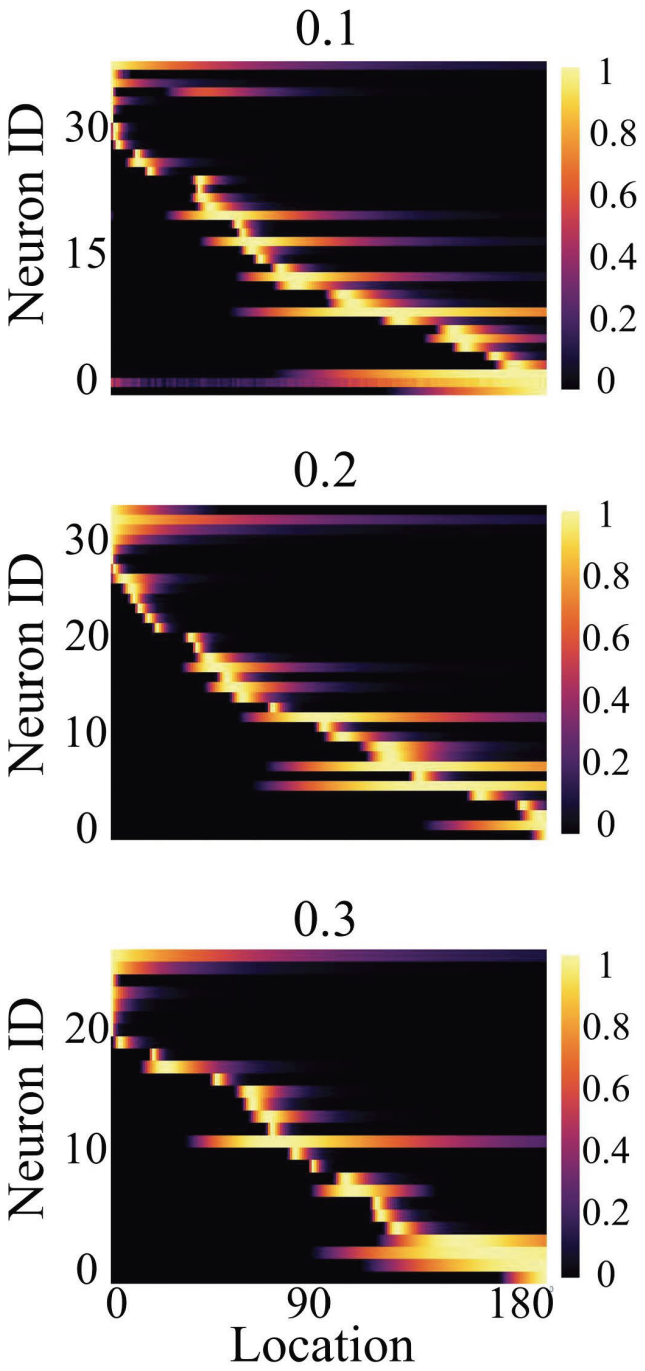


Figure A11: Activity of neurons in the $P$ layer for an autoencoder with sparsity constraint and local weights. Each panel corresponds to a different dropout value.

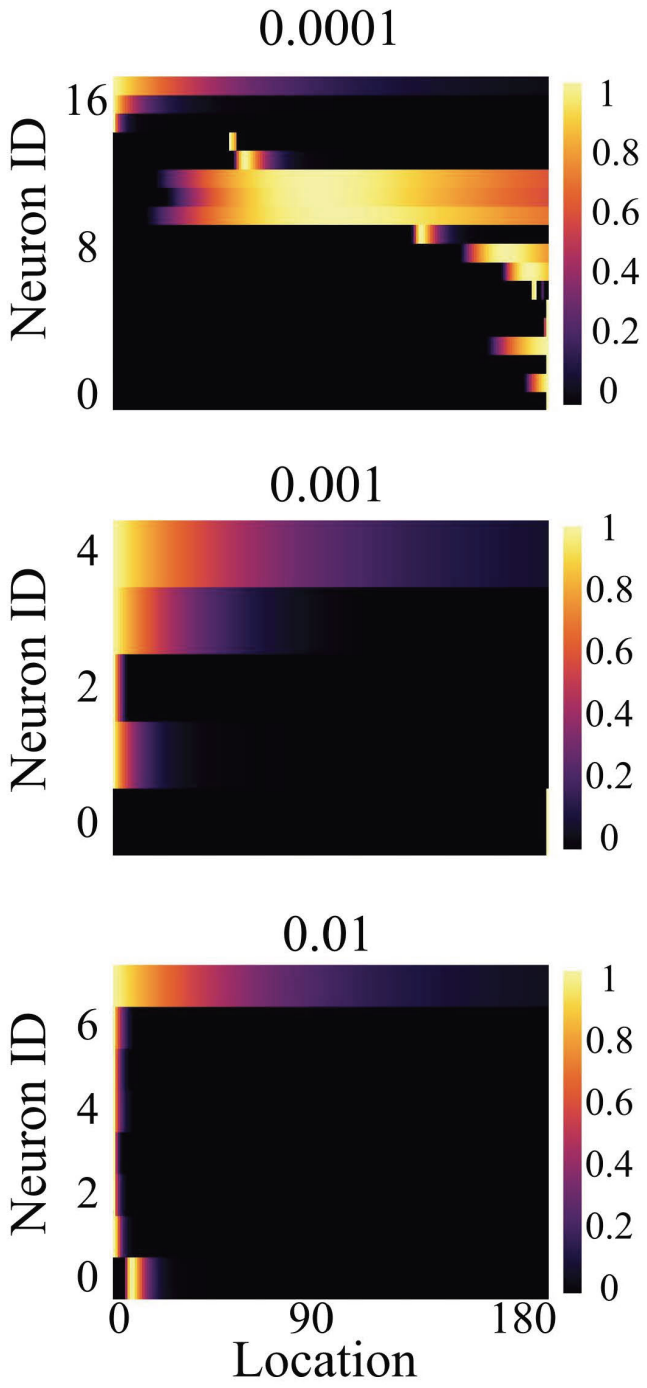


Figure A12: Activity of neurons in the $P$ layer for an autoencoder with sparsity constraint and local weights. Each panel corresponds to a different $L2$ regularization.

## A.4 Sparse autoencoder with local and shared weights

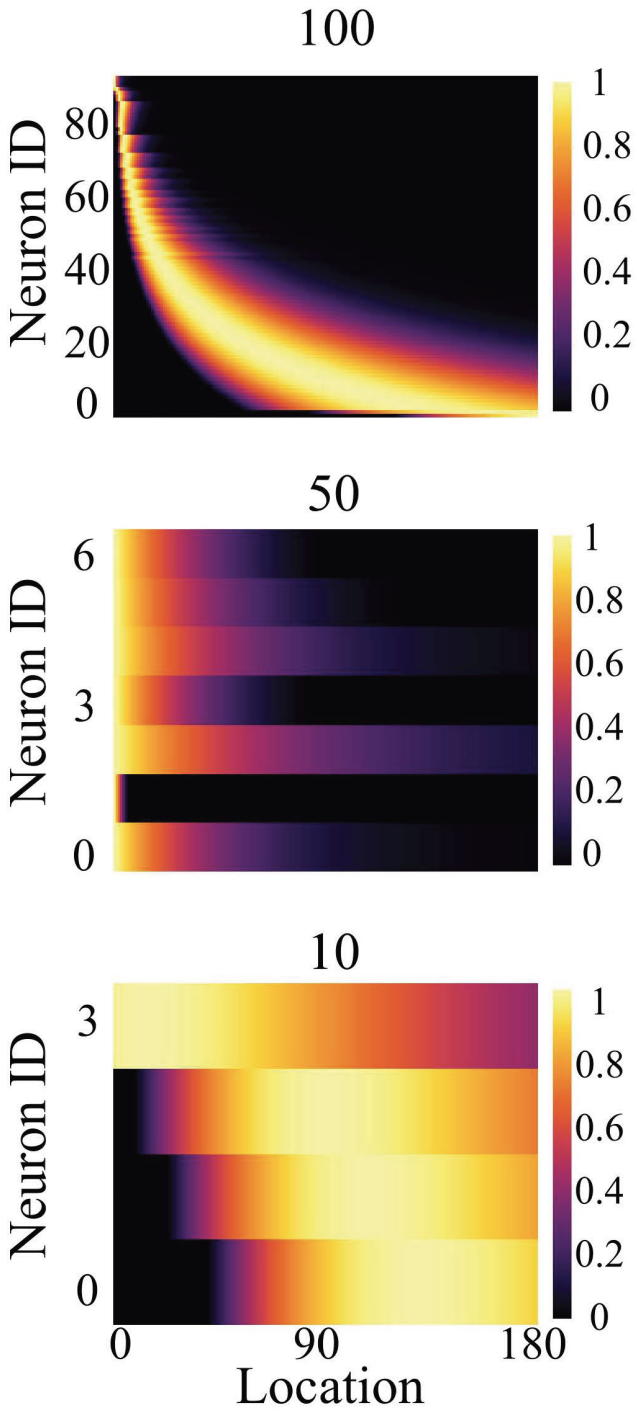


Figure A13: Activity of neurons in the $P$ layer for an autoencoder with sparsity constraint and local and shared weights. Each panel corresponds to a different number of neurons in the $P$ layer.

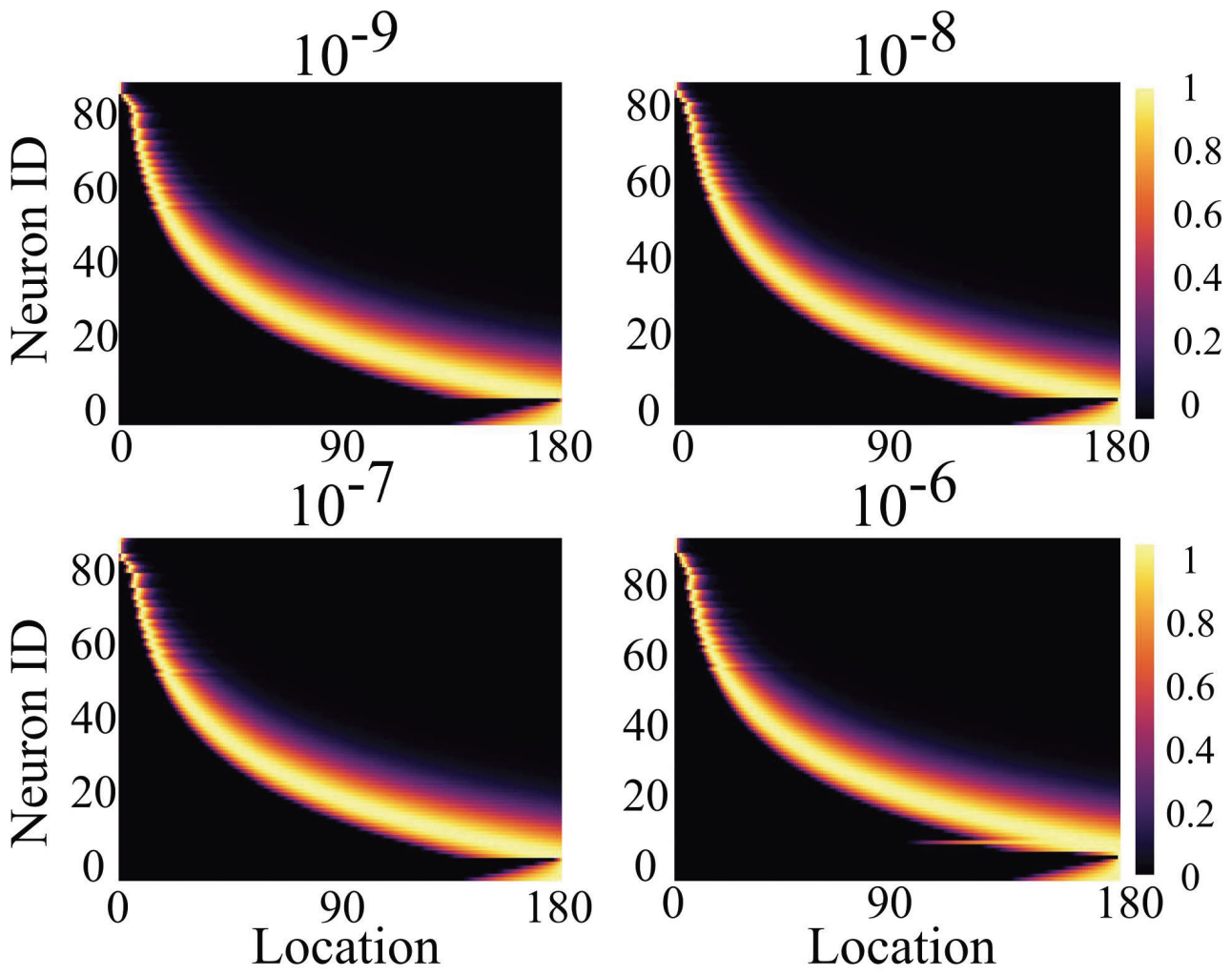


Figure A14: Activity of neurons in the $P$ layer for an autoencoder with sparsity constraint and local and shared weights. Each panel corresponds to a different activity penalty.

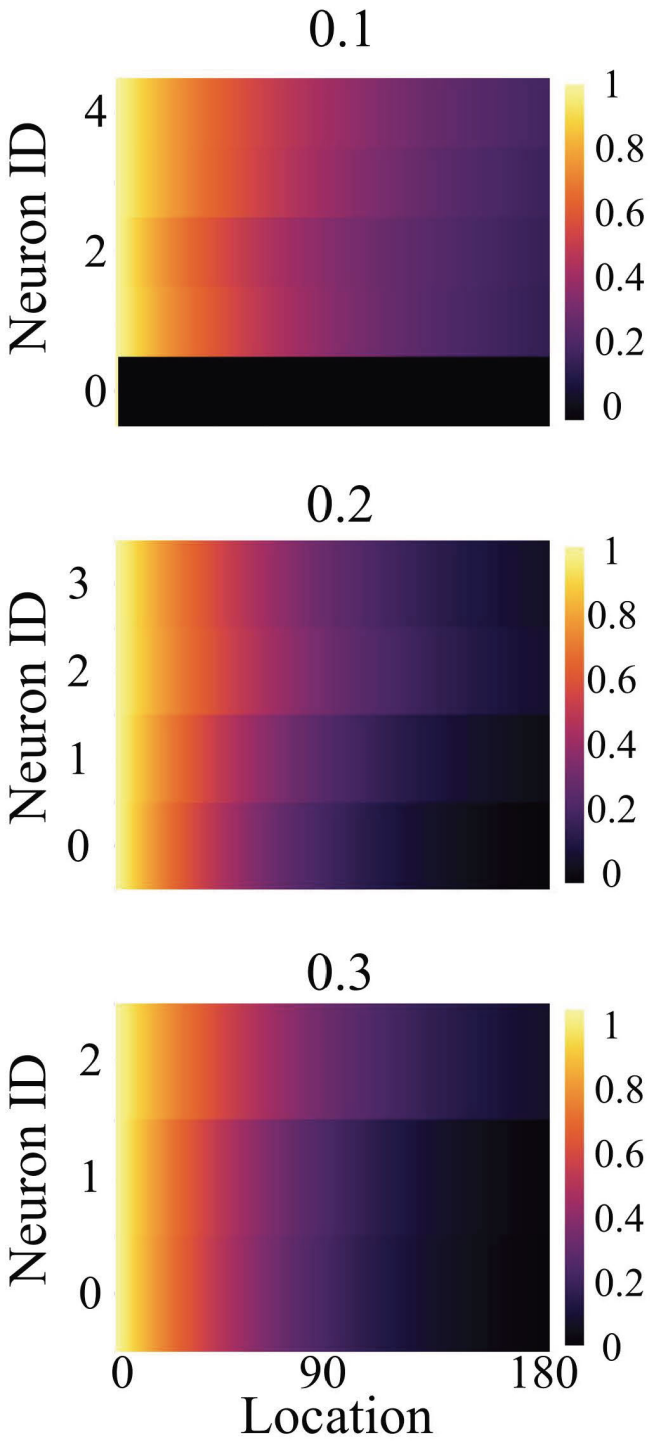


Figure A15: Activity of neurons in the $P$ layer for an autoencoder with sparsity constraint and local and shared weights. Each panel corresponds to a different dropout value.

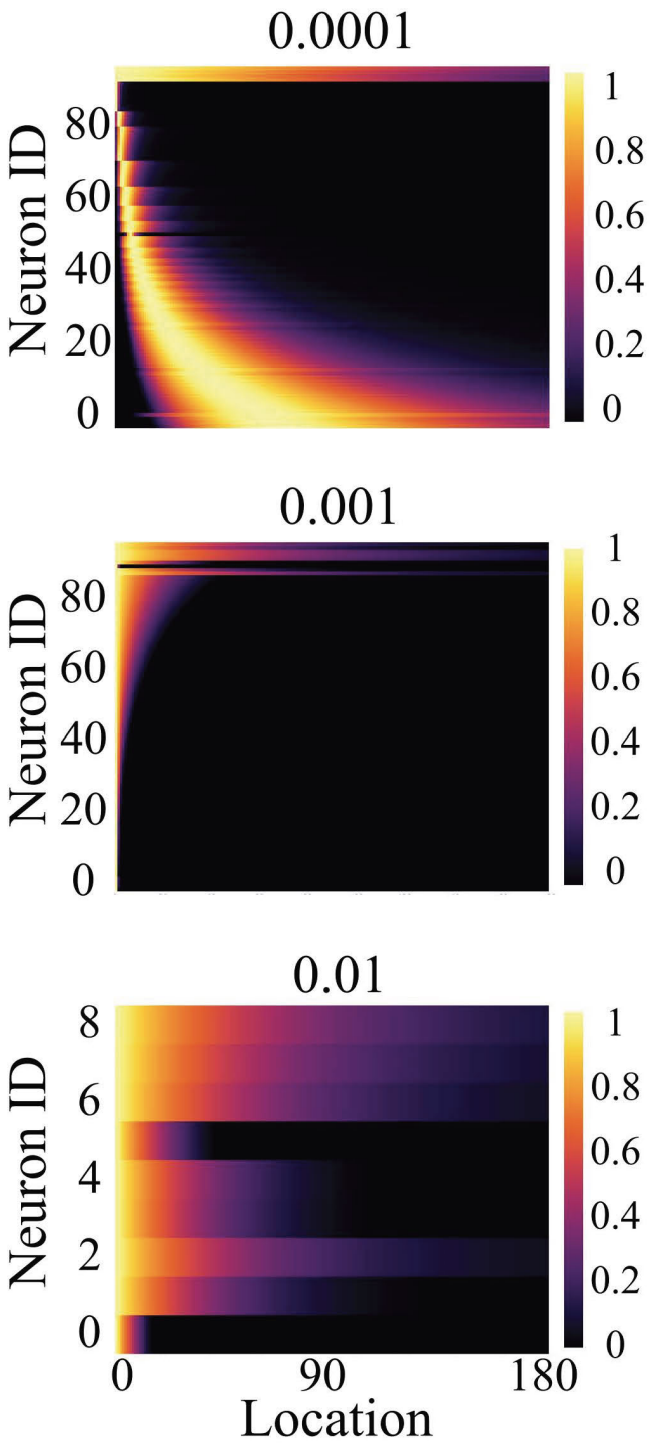


Figure A16: Activity of neurons in the $P$ layer for an autoencoder with sparsity constraint and local and shared weights. Each panel corresponds to a different $L2$ regularization.

## A.5 Linear spacing of time constants also gives rise to place cells, but not with scale-invarint properties

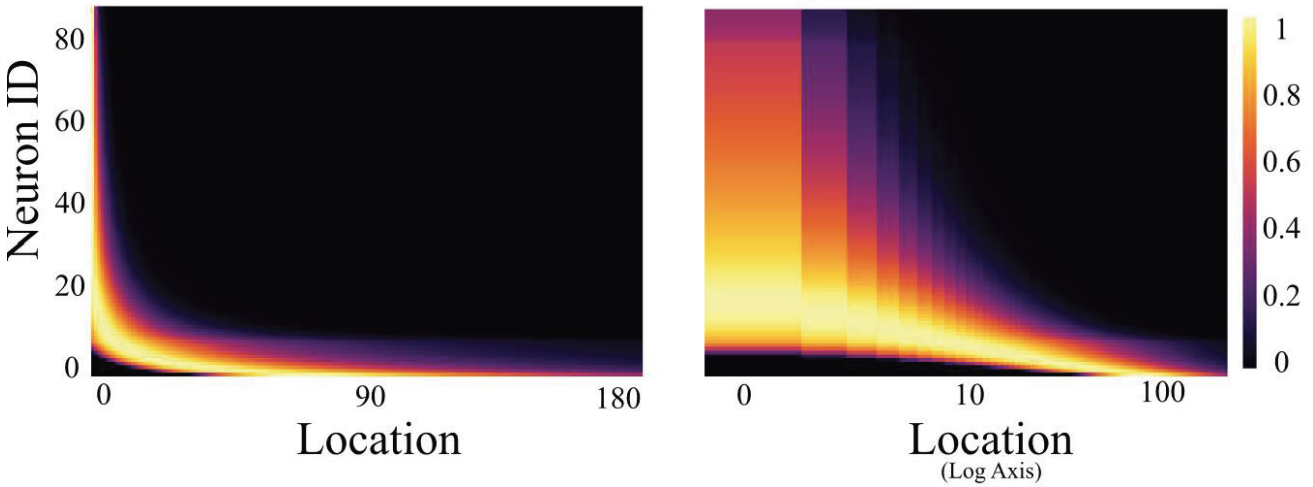


Figure A17: Linear spacing of time constants also results in place cells. However, such place cells are not scale-invariant. The left panel shows the activity profile of place cells in a model that used the linear spacing of time constants (values of $\mathbf{s}$ were linearly sampled from 0.001 to 0.1 as described in Section 2.2) with a place cell layer size of 100, and a sparsity parameter ($\rho$) of 0.8, without dropout and with $L_2$ and activity penalty set to 0. The right panel shows the same graph but with a logarithmic axis. Note that activations are not equally wide and equidistant as with geometrically spaced time constants.