

On the Organization of Grid and Place Cells: Neural Denoising via Subspace Learning

David M. Schwartz

dmschwar@email.arizona.edu

*Department of Electrical and Computer Engineering, University of Arizona,
Tucson, AZ 85719, U.S.A.*

O. Ozan Koyluoglu

ozan.koyluoglu@berkeley.edu

*Department of Electrical Engineering and Computer Science, University
of California, Berkeley, Berkeley, CA 94720, U.S.A.*

Place cells in the hippocampus (HC) are active when an animal visits a certain location (referred to as a place field) within an environment. Grid cells in the medial entorhinal cortex (MEC) respond at multiple locations, with firing fields that form a periodic and hexagonal tiling of the environment. The joint activity of grid and place cell populations, as a function of location, forms a neural code for space. In this article, we develop an understanding of the relationships between coding theoretically relevant properties of the combined activity of these populations and how these properties limit the robustness of this representation to noise-induced interference. These relationships are revisited by measuring the performances of biologically realizable algorithms implemented by networks of place and grid cell populations, as well as constraint neurons, which perform denoising operations. Contributions of this work include the investigation of coding theoretic limitations of the mammalian neural code for location and how communication between grid and place cell networks may improve the accuracy of each population's representation. Simulations demonstrate that denoising mechanisms analyzed here can significantly improve the fidelity of this neural representation of space. Furthermore, patterns observed in connectivity of each population of simulated cells predict that anti-Hebbian learning drives decreases in inter-HC-MEC connectivity along the dorsoventral axis.

1 Introduction ---

Place cells are a class of spatially modulated neurons with an approximately bivariate gaussian tuning curve centered on a particular location in the environment and have been identified in the hippocampus (O'Keefe

& Dostrovsky, 1971; O'Keefe, 1976; Ekstrom et al., 2003). Grid cells are spatially modulated neurons with firing fields that form a periodic and hexagonal tiling of the environment and are found in the entorhinal cortex (EC) of rats, mice, bats, and humans (Hafting, Fyhn, Molden, Moser, & Moser, 2005; Fyhn, Hafting, Witter, Moser, & Moser, 2008; Yartsev, Witter, & Ulanovsky, 2011; Doeller, Barry, & Burgess, 2010; Jacobs et al., 2013). Grid cells are clustered in discrete modules wherein cells share grid scale (Stensola et al., 2012). Anatomically, both cell types share a dorsoventral organization, with cells possessing wider receptive fields distributed toward the ventral end (Strange, Witter, Lein, & Moser, 2014; Stensola et al., 2012). It is known that the rat grid cell network requires communication from the hippocampus to maintain grid-like activity (Bonnievie et al., 2013) and that a significant improvement in accuracy of the rodent place cell representation is tightly correlated with the emergence of the grid cell network (Muessig, Hauser, Wills, & Cacucci, 2015). However, the mechanisms by which these networks communicate and how each may bolster the other's accuracy are unknown. Objectives of this work include the investigation of coding theoretic limitations of the mammalian neural code for location and how communication between grid and place cell networks may improve the accuracy of each population's representation.

Associative memories are a class of biologically implementable content addressable memory consisting of networks of neurons, a learning rule, and, in some instances, a separate recall process (Hopfield, 1982; Amit & Treves, 1989). This means that they can be exploited to stabilize the states of their constituent neurons to match a previously memorized network state if enough of the network already lies in this state. The information capacity of the simplest of these constructions is quite limited: $\frac{n}{2 \log n}$ bits for a network of n binary neurons (McEliece, Posner, Rodemich, & Venkatesh, 1987). However, recent advances by Salavati et al. take advantage of sparse neural coding and nonbinary neurons to design an associative memory with information storage capacity exponential in the number of neurons (Salavati, Kumar, & Shokrollahi, 2014). Sparse connectivity confers the memory network with other performance improvements: infrequent spiking implies reduced energy costs and faster convergence to a stable state.

In communications, this principle is leveraged by low-density parity check codes (LDPC), a class of linear block code whose power (in coding and decoding complexity) depends on the sparsity of the code's parity check matrix. Commonly, denoising a LDPC code involves iteratively passing messages along edges of a bipartite graph consisting of a collection of nodes that stores and updates an estimate of the originally transmitted word connected to a collection of nodes that computes the code's parity check equations (Chen & Fossorier, 2002; Declercq & Fossorier, 2007). Recent developments in the intersection of coding theory and machine learning demonstrate that neural networks can learn an approximation of

an LDPC code's parity structure, and by executing belief propagation algorithms, they can recover memorized patterns in the presence of noise (Salavati et al., 2014).

Nature provides myriad circumstances in which many neural computations (e.g., object recognition, acoustic source localization, and self-localization) must be executed robustly in the presence of neural noise if the organism is to survive. We propose a denoising mechanism for populations of grid and place cells in the form of the associative memories described in Salavati et al. (2014), Karbasi, Salavati, Shokrollahi, & Varshney (2014), and Karbasi, Salavati, & Shokrollahi (2013), which takes advantage of coding theoretic properties of these populations to ameliorate the negative impacts of noise. We observe that after learning, average connectivity between place cells and grid modules decreases with increasing place field size for each module. We demonstrate that the effectiveness of the proposed denoising algorithm relies on the biological organization of grid cells into discrete modules. Additional contributions of this work include the coding model and denoising systems themselves as a framework in which to characterize limits on the fidelity of cooperating neural codes subject to noise, for physical position or other variables such as the auditory code studied in Aronov, Nevers, and Tank (2017), and improved clarity about how parameterization of grid and place cell populations affects these fundamental information and coding theoretic limits.

Redundancy in receptive field (RF) population codes is known to confer improvements in decoding accuracy when a small tolerance to error is introduced (expressed in this case in the stimulus space to which we decode; Curto, Itskov, Morrison, Roth, & Walker, 2013). To our knowledge, we are the first to investigate coding theoretic impacts of redundancy in grid cell populations. We study the impact of this redundancy on decoding accuracy by comparing denoising and decoding performance across codes of varying redundancies. We demonstrate that after denoising, a maximum likelihood (ML) estimator reliably decodes position from population activity with small position estimation error in the presence of bounded noise. Overall, our work shows that the biological organization of grid cells into modules may be necessary for optimal self-localization.

This article is organized as follows. In section 1, we introduce a few key concepts and present the main results. Section 2 introduces the theoretical framework on which our model is built, describing code construction, denoising network, learning algorithms, and denoising algorithms in sections 2.1, 2.2, 2.3, and 2.4, respectively. Section 3.1 presents results of all coding theoretic analysis and experimentation. Section 3.2 annotates results of the learning algorithms. Section 3.3 describes outcomes of performance tests of the denoising algorithms. Section 4 consists of a discussion of these results, their implications and limitations, and a physiologically testable hypothesis they inform.

2 Theoretical Framework

2.1 A Hybrid Code. We consider a population of place and grid cells, a total of N neurons. There are M grid cell modules, each module, m , containing J_m neurons, and P place cells. Throughout this article, we use J to refer to the number of grid cells in module 1, which—if grid cells are allocated to modules nonuniformly—is not equal to each other module's J_i . The firing rate of each grid cell is denoted as $g_{m,j}$, for $m \in \{1, \dots, M\}$ and $j \in \{1, \dots, J_m\}$. Place cells' firing rates are denoted as p_i , for $i \in \{1, \dots, P\}$. The activity of this population, as a function of location ℓ , is represented by

$$\mathbf{x}_i(\ell) = \begin{cases} g_{m,j}(\ell), & i = \sum_{k=1}^{m-1} J_k + j, i \leq \sum_{m=1}^M J_m \\ p_{i-MJ}(\ell), & i > \sum_{m=1}^M J_m \end{cases},$$

where the location-dependent mean firing rates of the grid cells, $g_{m,j}(\ell)$, are given by the following two-dimensional distributions resembling von Mises density functions,

$$g_{m,j}(\mathbf{s}) = \frac{f_{\max}}{Z} \exp \left[\sum_{k=1}^3 \cos \left(\frac{4}{\lambda_m \sqrt{3}} \mathbf{u}(\theta_k - \theta_{m,j}) \right) \cdot \left(\mathbf{s} - \mathbf{c}_{m,j} \right) + \frac{3}{2} \right] - 1, \quad (2.1)$$

where $\mathbf{u}(\theta_k - \theta_{m,j})$ is a unit vector in the direction of $\theta_k - \theta_{m,j}$; $\mathbf{s} \in [0, L] \times [0, L]$ is the position stimulus; and $\mathbf{c}_{m,j}$, $\theta_{m,j}$, and λ_m are the grid cell's spatial phase offset, orientation offset, and scaling ratio. The angles of the sinusoids composing the grid pattern (i.e., θ_k) were taken to be ideal values about which the measurements presented in Stensola et al. (2012) appear to fluctuate. More precisely, we choose $\theta_k \in \{-60^\circ, 0^\circ, 60^\circ\}$. A scaling ratio of λ defines the scale of module m as $\lambda_m = \lambda_1(\lambda)^{m-1}$. Z is a normalizing constant (≈ 2.857399), and f_{\max} is the grid cell's maximum firing rate. Unless otherwise stated, $\theta_{m,j}$ is chosen so as to mirror the observations in Stensola et al. (2012): $\theta_{m,j}$ is identical across grid cells in the same module (i.e., for indices m, i, j , $\theta_{m,i} = \theta_{m,j}$), and these orientation offsets are selected randomly. In two dimensions, place cells have bivariate gaussian tuning curves, with mean $\xi \in [0, L] \times [0, L]$, correlation, $\rho \in [-\frac{1}{2}, \frac{1}{2}]$ (chosen uniformly randomly), and covariance $\begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$, where σ_1 and σ_2 are chosen independently and uniformly randomly from $[0.9\lambda_1, 1.1\lambda_M]$. We require that σ_1 and σ_2 depend on λ_1 so that both grid and place cell receptive fields lie in similar spatial scales.

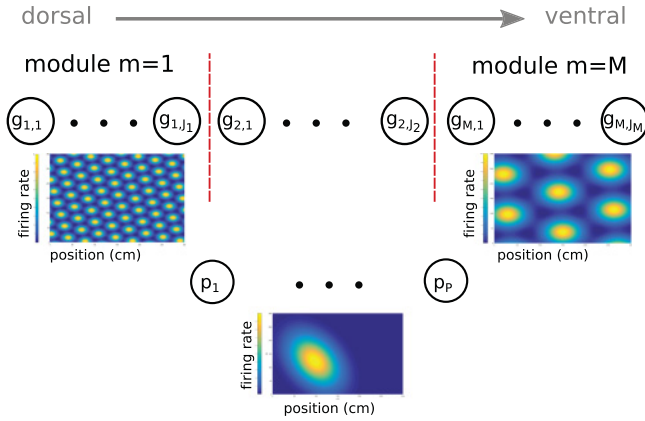


Figure 1: Concatenation of activities of grid and place cells (shown with typical idealized model receptive fields) to form the hybrid code.

2.1.1 *A Hybrid Codebook.* C code words, of length $N = P + \sum_{m=1}^M J_m$, are generated by choosing locations from the vertices of a square lattice imposed on the plane, with unit area equal to $(\Delta L)^2$ and total area equal to L^2 . C is assembled by placing these code words in its rows and represents the states of the grid and place cells when stimulated with these positions. The mapping that forms this code is illustrated in Figure 1.

2.2 Denoising Network. Two high-capacity associative memory designs are considered to test the hybrid code’s resilience to noise. In each case, the memory network is a bipartite graph consisting of N pattern neurons (i.e., grid and place cells) and n_c constraint neurons. In the unclustered design, all constraint neurons are connected to a random set of pattern neurons. In the clustered configuration, the constraint neurons were split into M distinct clusters of n constraint neurons per cluster, with each cluster connected to a distinct grid module. Each cluster’s constraint neurons were connected randomly to pattern neurons, chosen from a set consisting of every grid cell in the corresponding module and every place cell.

We also consider a foil to this systematic clustering architecture organized by grid modules: grid and place cells are randomly assigned to clusters. Figures 2a and 2b depict the general connectivity structure of the unclustered and clustered designs, respectively. In both the clustered and unclustered configurations, a neurally plausible modified version of Oja’s subspace learning rule was applied to learn the code, that is, a sparse connectivity matrix is found such that the weights of connections from constraint neurons to pattern neurons lie orthogonal to the code space (the space spanned by C ; Oja & Kohonen, 1988). This way, constraint neuron

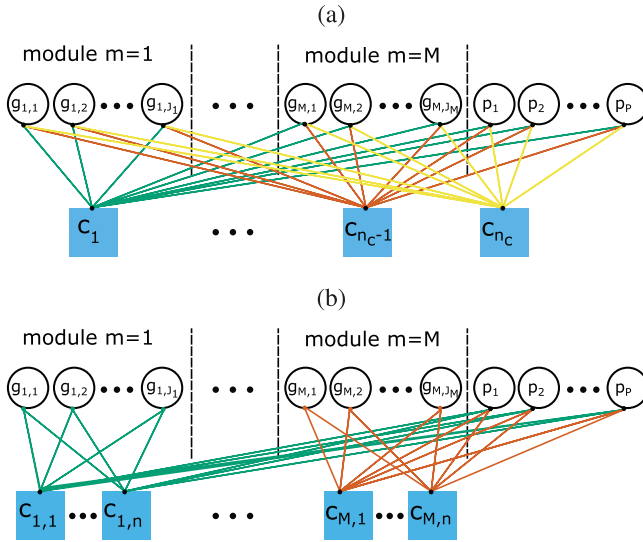


Figure 2: (a) Structure of an unclustered denoising network, considered as a baseline for comparison to the neurophysiologically inspired systematic clustering scheme. (b) Structure of a systematically clustered denoising network in which clusters of constraint neurons connect to all place cells but only to the corresponding module of grid cells.

connectivity converges to the parity structure of the code and may be used in denoising operations.

2.3 Code Construction via Subspace Learning. Before we can use the denoising system to correct corrupted code word, it must learn (i.e., adapt its weights for) the hybrid code. This process is complete when the constraint neurons may be read to determine if the states of the pattern neurons map to a valid code word. Formally, this amounts to finding a connectivity matrix, W ($W_{i,j}$ is the synaptic weight between constraint neuron i and pattern neuron j), whose rows are approximately perpendicular to the code space. A procedure to procure such a matrix is outlined in Oja and Kohonen (1988) and improved in Salavati et al. (2014). Note here that this learning process is not a model for the development of either grid or place cells' apparent receptive fields nor their remapping, as in Monaco & Abbott (2011). These algorithms begin with a random set of vectors, and for each, they seek a nearby vector orthogonal to \mathcal{C} (i.e., a vector onto which each element of \mathcal{C} has minimal projection). We implement this in algorithm 1 (a derivation of this algorithm is in appendix B. In the clustered design, algorithm 1 is applied to each cluster's local connectivity matrix. Note that

Algorithm 1: Neural Learning.**Input:** set of C patterns, \mathcal{C} , stopping point, ϵ **Output:** learned weights matrix, W

```

1: for rows,  $\mathbf{w}$ , of  $W$  do
2:   for  $t \in \{1, \dots, T_{\max}\}$  do
3:      $\alpha_t \leftarrow \max\left\{\frac{50 \cdot \alpha_0}{50 + \log_{10}(t)}, 0.005\right\}$ 
4:      $\theta_t \leftarrow \frac{\theta_0}{t}$ 
5:     for  $\mathbf{c} \in \mathcal{C}$  do
6:       if  $\|\mathbf{c}\| > \epsilon$  then
7:          $\alpha_t \leftarrow \frac{\alpha_0}{\|\mathbf{c}\|^2}$ 
8:       end if
9:        $\mathbf{w} \leftarrow \text{Dale}(\text{update}(\mathbf{c}, \mathbf{w}, \alpha_t, \theta_t, \eta))$ 
10:    end for
11:    if  $\|\underline{\mathbf{C}}\mathbf{w}'\| < \epsilon$  then
12:      break
13:    end if
14:     $t \leftarrow t + 1$ 
15:  end for
16:  for components,  $w_i$  of  $\mathbf{w}$  do
17:    if  $|w_i| \leq \epsilon$  then
18:       $w_i \leftarrow 0$ 
19:    end if
20:  end for
21: end for

```

here, all arithmetic on the synaptic weights, $W_{i,j}$, is performed in \mathbb{R} , while arithmetic on states of neurons (i.e., their firing rates) is quantized to the nearest integer in $[0, Q - 1]$. The maximum firing rate, $f_{\max} = Q - 1$, is identical for all neurons. With each update, $\mathbf{w} \leftarrow \mathbf{w} - \alpha_t (y(\mathbf{x} - \frac{y\mathbf{w}}{\|\mathbf{w}\|^2}) + \eta\Gamma(\mathbf{w}, \theta))$, where θ is a sparsity threshold, η is a penalty coefficient, $y = \mathbf{x}^T \mathbf{w}$ is the scalar projection of \mathbf{x} onto \mathbf{w} , and α_t is the learning rate at iteration t . Γ is a sparsity-enforcing function, approximating the gradient of a penalty function, $g(\mathbf{w}) = \sum_{k=1}^n \tanh(\sigma \mathbf{w}_k^2)$, which, for appropriate choices of σ , penalizes nonsparse solutions early in the learning procedure (Salavati et al., 2014).

As in Salavati et al. (2014), to speed up learning, we approximate $\Gamma = \nabla g$ with

$$\Gamma(w_t, \theta_t) = \begin{cases} w_t & : |w_t| \leq \theta_t \\ 0 & : \text{otherwise} \end{cases}.$$

This update rule is an improved approximation to Oja's Hebbian learning algorithm (Oja & Kohonen, 1988), with advantages in both biological plausibility and computational complexity. For connections of fixed type (i.e., inhibitory versus excitatory), Oja's rule alone is biologically dubious without the inclusion of many constraint neurons to manage this change in type. Dale's principle states that real synaptic connections change type rarely, if ever (Eccles, 1976). In accordance with this principle, our update rule does not allow weights to change sign. This is accomplished after the updated weights are determined. If the sign has changed after applying the update, set the new weight to a value just above (resp. just below) zero if the previous weight was positive (resp. negative). Thus, when learning is complete, these weights will be small in magnitude and are thresholded to zero.

In algorithm 1, line 12 terminates learning of the current weight, \mathbf{w} , if the sum of the projections of \mathbf{w} on each pattern is no more than ϵ away from zero, that is, if the current weight vector is approximately orthogonal to the code space. Lines 17 to 19 perform a thresholding operation that maps to zero any weight sufficiently small in magnitude. This is primarily to suppress numerical errors and promote consistency, as in line 11, we use ϵ as a small, positive constant. Note that since the weights processed on each iteration are independent of those in other iterations, this algorithm can be readily parallelized so that each constraint neuron learns its weights simultaneously.

2.4 Denoising and Decoding. We implemented a bit flipping style neural denoising process, which we applied to both the clustered and unclustered denoising networks. For all configurations (clustered and unclustered and for a fixed maximum number of denoising iterations, the bit flipping algorithm performs no worse than winner-take-all. Moreover, since it requires only the additional implementation of parallel thresholding operations for each pattern neuron, a biological realization of their inclusions is no less plausible. The goal of this algorithm is to recover the correct activity pattern, \mathbf{x} , which has been corrupted by noise and, as such, is currently (and errantly) represented by a noisy version, $\mathbf{x}_n = \mathbf{x} + \mathbf{n}$, where \mathbf{n} is this noise pattern. Since each weight vector is nearly perpendicular to every pattern, for a matrix of weights, W , $\mathbf{x}_n W'$ reveals inconsistencies in \mathbf{x}_n , which the denoising algorithm seeks to correct in the feedback stage.¹ In denoising, feedback weights from constraint neurons to pattern neurons are taken to

¹To see this, consider that $\mathbf{x}_n W' = (\mathbf{x} + \mathbf{n})W' = \mathbf{x}W' + \mathbf{n}W' \approx 0 + \mathbf{n}W'$.

Algorithm 2: Modular Recall.

Input: local weights for this cluster, W , maximum number of iterations, T_{\max} , noisy subpattern, \mathbf{x} , feedback threshold, ϕ

Output: denoised subpattern, \mathbf{d}

```

1:  $\mathbf{d} \leftarrow \mathbf{x}$ 
2: while  $t < T_{\max}$  do
3:    $\mathbf{y} \leftarrow \mathbf{x}W'$ 
4:   if  $\|\mathbf{y}\| < \epsilon$  then
5:     break;
6:   end if
7:    $\mathbf{f} \leftarrow \frac{|\mathbf{y}'| \cdot |W|}{\sum_{i=1}^m |W|}$ 
8:   for each pattern neuron,  $j$  do
9:     if  $\mathbf{f}_j \geq \phi$  then  $\mathbf{f}_j = \text{sign}(\mathbf{x}_j)$ 
10:    else  $\mathbf{f}_j = 0$ 
11:    end if
12:  end for
13:   $\mathbf{d} \leftarrow \mathbf{d} - \mathbf{f}$ 
13: end while

```

be equal to the corresponding feedforward weight (i.e., synaptic connectivity is symmetric). The clustered denoising process begins with algorithm 3, in which each cluster attempts to detect errant pattern neurons. If no errors are detected, the process is complete. Otherwise, algorithm 2 is invoked for each cluster that detected errant neurons. This and other denoising processes are discussed in greater detail in Karbasi et al. (2013) and Salavati et al. (2014). Note that this denoising mechanism differs from error correction methods presented in Fiete, Burak, and Brookings (2008) and Stemmler, Mathis, and Herz (2015) in that information contributed by place cells reaches grid cells only through constraint neurons, and place information contributed by grid cells at module i reaches other modules only through constraint neurons if connectivity allows.

In order to quantify the information content of the population, we estimated the location encoded by the population using a maximum likelihood decoder in four different schemes. Joint hybrid decoding utilizes information from all cells. Grid- (resp. place-) only decoding utilizes information from only grid (resp. place) cells. Grid decoding conditioned on place response performs decoding using only information provided by the grid cells; however, the only candidate locations considered for the estimate are those that are not impossible given the place cell activity.

Algorithm 3: Sequential Denoising.

Input: local weights, W_i , for each cluster, $i \in \{1, \dots, M\}$, noisy pattern, \mathbf{x}_n , stopping threshold, ϵ **Output:** denoised pattern, \mathbf{x}_d

```

1:  $\mathbf{x}_d \leftarrow \mathbf{x}_n$ 
2: while  $t < T_{\max}$  and a cluster has an unsatisfied constraint do
3:   for each cluster,  $i \in \{1, \dots, M\}$  do
4:      $\mathbf{x} \leftarrow$  subpattern corresponding to cluster  $i$ 
5:      $\mathbf{d} \leftarrow$  Modular_Recall( $\mathbf{x}, W_i$ )
6:     if  $|\mathbf{d}W_i| \leq \epsilon$  then
7:        $\mathbf{x}_d(\text{cluster } i\text{'s subpattern indices}) \leftarrow \mathbf{d}$ 
8:     end if
9:   end for
10:   $t \leftarrow t + 1$ 
11: end while

```

3 Results

3.1 Coding Theoretic Results. We now endeavor to disentangle the connections between grid and place cell parameter choices and coding theoretically relevant dependent variables and understand these links. The results presented here motivate the questions answered in section 3.3, in which we investigate how the coding parameters studied here limit fidelity and the error correction capability of the corresponding representation of space. We begin our investigation of coding theoretic properties of the hybrid code by defining a measure of redundancy of grid cell population response: μ_p . More precisely, we define μ_p , a hybrid code's spatial phase multiplicity, as the number of grid cells with the same phase in the same module (e.g., if $\mu_p = 5$, in a module with 20 grid cells, there must be four unique spatial phases). This replication of grid cell phases can be considered as a repetition code in the activity of the grid cell population. Wennberg (2015) revealed that there may be a highly nonuniform distribution of phases among grid cells. Considering replication of grid cells (i.e., modules consisting of multiple grid cells of the same phase) allows us to investigate coding theoretic repercussions of this phenomenon. Inspired by Mosheiff, Agmon, Moriel, and Burak (2017) for each of these regimes, we consider two distributions of grid cells to modules: uniform and nonuniform. Mosheiff et al. (2017) find that choosing $J_m \propto \frac{1}{\lambda^{m-1}}$ produces a more efficient representation of space. When modeling the nonuniform allocation of grid cells to modules, we chose $J_m = \lfloor \frac{J}{\lambda^{m-1}} \rfloor$, since the scale of module m is defined as

$\lambda_m = \lambda_1(\lambda)^{m-1}$. Neural recordings show that the smallest scale is $\lambda_1 \approx 40$ cm (the value used here; Stensola et al., 2012).

We construct a codebook matrix, \underline{C} , by placing elements of \mathcal{C} in its rows. We computed normalized rank of the code, $R = \frac{\text{rank}(\underline{C})}{N} \in [0, 1]$ as a function of the grid scaling ratio. Normalized rank is an indicator of a code's density, expressed as the fraction of possible dimensions of the code space occupied by a particular code. R is an important feature to consider since a code's dimensionality determines the dimensionality of its null space, the object that is learned by the denoising network. As discussed in Salavati et al. (2014), if we suppose that $\mathcal{C} \subset \mathbb{R}^n$ and $\dim(\mathcal{C}) = k < n$, then there are $n - k$ mutually orthogonal vectors that are also orthogonal to our code space (e.g., any basis for the null space of the code), each representing one valid constraint equation. Thus, rank provides a fundamental limit on the number of unique effective constraint nodes the denoising network may learn.

The grid cell code is known to be dense (Fiete et al., 2008). This is especially pronounced when all orientations and phases are chosen randomly (uniformly from $[0, 2\pi]$ and $[0, L] \times [0, L]$, respectively), where for all choices of other parameters, the hybrid code achieves full rank at a low rate. That is, the experimentally observed properties of the grid cell code described in Stensola et al. (2012) produce a measurable decrease in rank compared to typical ranks observed when all orientations and phases are chosen randomly.

Place cell activity forms a relatively sparse code (for enough cells and a sufficiently large environment); thus, combining populations of grid and place cells realizes codes that are sparser than the grid cell component of the code. When $\mu_p = 1$, a hybrid code with no place cells achieves the largest normalized rank. Since place cells communicate redundant information, their inclusion also reduces rank, which is precisely the trend observed in Figure 3. However, this appears to reverse when $\mu_p > 1$ for a sufficiently small number of participating place cells. This occurs because rendering grid cells redundant by increasing phase multiplicity lowers the rank of the grid-only component of the code. Consequently, including place cells increases rank, until the information contributed by the place cells reaches its maximum, at which point the inclusion of additional place cells only lowers rank. Error bars (measuring SEM) are included due to the stochastic nature of instantiating certain parameters (e.g., ξ , which is always chosen uniformly randomly from the set of quantized locations).

We also computed rank, R , as a function of code rate, $r = \frac{C}{N}$ (number of locations represented per neuron), a measure of spatial resolution and efficiency of the encoding (i.e., for a fixed L , a higher code rate, r , is obtained by lowering ΔL or by decreasing N). It is their common denominator (N) that links the dependence on population size of both rank and rate. When phases are chosen randomly, low rank is difficult to obtain at all, but the smallest of code rates tested ($r \in [0, 1]$ and $\mu_p > 1$ may result in low ranks if enough place cells are included). In contrast, Figure 4 shows that codes

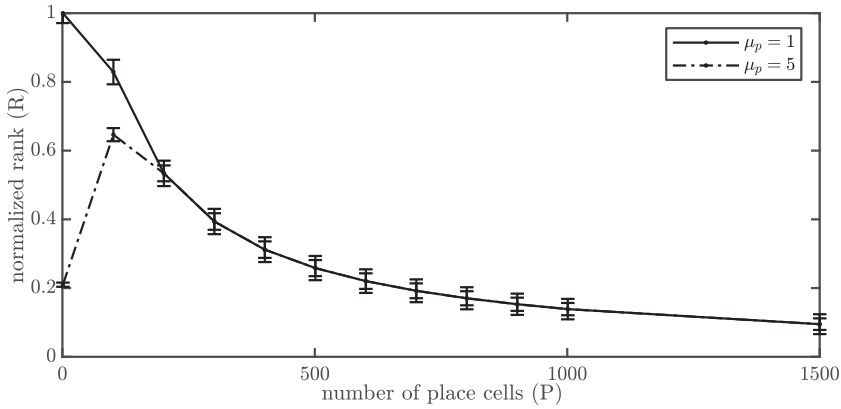


Figure 3: Code rank (R) versus number of place cells (P) for a uniform allocation of grid cells. Increasing phase multiplicity (μ_p) produces a code with low rank until sufficiently many place cells are included in the code that additional place cells contribute only redundant location information. Here (and in any other plot containing them) error bars show standard error of the mean (SEM). The codes with $\mu_p = 1$ exhibit slightly greater variability in R (i.e., larger SEM) than those with $\mu_p = 5$. $N = 80 + P$.

spanning the spectrum of normalized ranks may be instantiated over a wide range of rates with appropriate choice of parameters. Further, this indicates that redundancy reduces dimensionality so low ranks are achievable even at rates much greater than biologically relevant. Later, we show that this low dimensionality is important in constructing sparse and readily denoisable representations of space. Figure 4 demonstrates that without the redundancy introduced by increasing $\mu_p > 1$, a hybrid code that encodes in 90 neurons more than 90 locations in a 9 m^2 environment has full rank. However, when $\mu_p > 1$, there is a stark drop in the maximum rank achieved. As shown, when $\mu_p > 1$, one may encode orders-of-magnitude more locations while maintaining low dimensionality. This trend is observed in each configuration shown and when grid cells are allocated to modules nonuniformly. Thus, both dense and sparse hybrid codes may be developed with proper choices of redundancy parameters.

A code's resilience to neural noise can be assessed by the minimum pairwise (Euclidean) distance between code words, (d). Traditionally, Hamming distance is used as the operative metric for characterizing minimum distance of a code. However, in cases when soft information is used by the decoder, Euclidean distance can prove to be more useful. Higher d (i.e., larger distances between code words) corresponds to a more noise-tolerant neural representation of space (Lin & Costello, 1983). In fact, ideally all errors induced by noise with amplitude less than $\lfloor \frac{d-1}{2} \rfloor$ are correctable (Lin

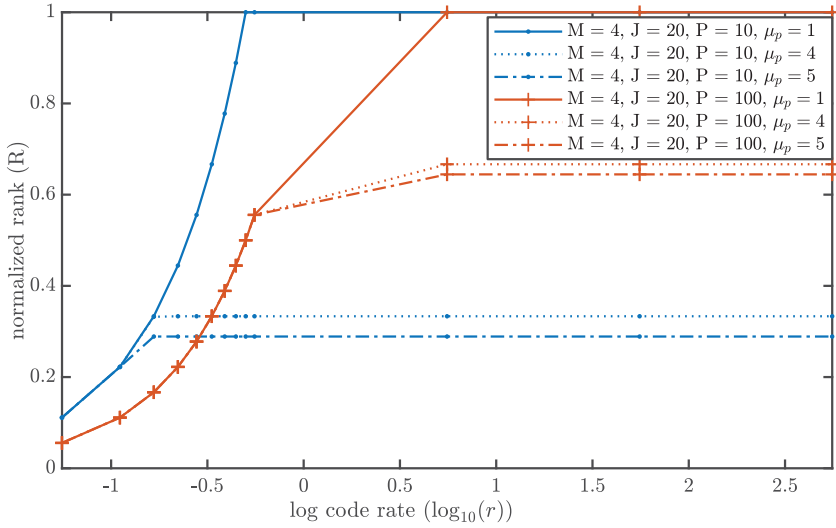


Figure 4: Code rank (R) versus logarithm of code rate ($\log_{10}(r)$) for grid cell organization choices consistent with observations in Stensola et al. (2012) with $M = 4$, $J = 20$, $P \in \{10, 100\}$, grid cells allocated to modules uniformly, and $\mu_p \in \{1, 4, 5\}$. Without grid cell phase redundancy, rank saturates for relatively small rates. In contrast, when phase redundancy is imposed on the grid cell population, low ranks are achievable at a wide range of rates. In each case considered here, $N = 80 + P$.

& Costello, 1983; Sreenivasan & Fiete, 2011). (For an intuitive illustration of this, see appendix D). We computed d as a function of rate, r , for different phase multiplicities, μ_p , (see Figure 5). For each configuration there is a trade-off between d and r . Since rank tends to increase and saturate with rate, this is also a trade-off between d and rank. When the rate is low, a low resolution of location is targeted: d is larger, so more erroneous neurons may be corrected. Note that for a fixed value of r , the codes with $\mu_p = 5$ have slightly smaller d , and this difference grows to saturation as r increases. Interestingly, at high rates, the decrease in d produced by increasing μ_p is much smaller for the population with grid cells distributed to modules nonuniformly. This observation applies for the highest rates for which computation of d is tractable with modern high-performance computers: $r < 10^6$. Thus, for a fixed r and large enough μ_p , the code with grid cells allocated to modules nonuniformly should exhibit measurably better denoising performance. We test this prediction by simulating the denoising process and collecting statistics presented in Figures 10 through 14. Surprisingly, for small r , with a uniform allocation of grid cells to modules, increases in μ_p appear to effect small decreases in d , while when grid cells

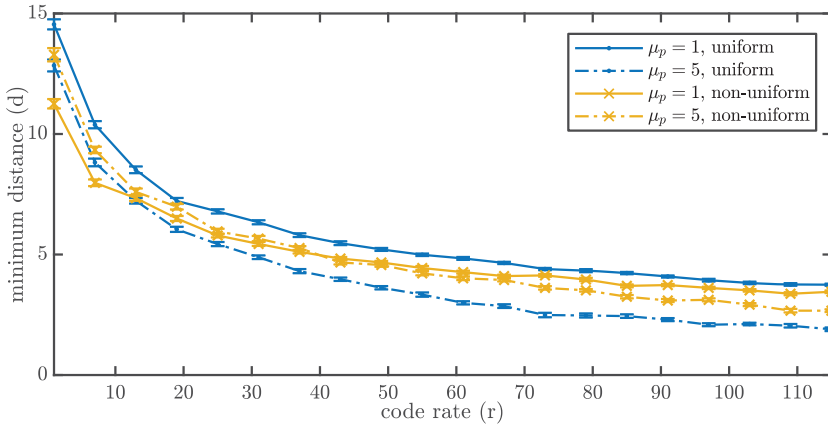


Figure 5: Minimum distance (d) versus code rate (r), for grid cell organization consistent with observations in Stensola et al. (2012), with $M = 4$, and $P = 10$. Data corresponding to $\mu_p = 1$ are marked with dashed curves and $\mu_p = 5$ by dot-dashed curves. In order to ensure $N = 90$, we choose $J = 20$ in the uniform case and $J = 32$ in the nonuniform case. For $r > 20$, d decreases with increases in r and μ_p . However, for small r and grid cells allocated to modules nonuniformly, increasing μ_p evokes an increase in d . The apparent ordering of configurations considered extends to much larger r than shown here (specifically, we probed $r \leq 10^6$).

are allocated to modules nonuniformly, increases in μ_p produce small but discernable increases in d .

For environments of a fixed size, x_{\max}^2 cm², and a hybrid code with N neurons, varying code rates implies quantizations of space with varying unit width ($\Delta L = \frac{x_{\max}}{\sqrt{C}}$). Since rate, $r = \frac{C}{N}$, $\Delta L = \frac{x_{\max}}{\sqrt{Nr}}$. Thus, the spatial sampling period, ΔL , is inversely proportional to \sqrt{r} . In order to ensure we probed reasonable code rates, we estimate the typical perceivable spatial period of a rat (through its place cells) by considering its running speed (ranging from .1 to 100 $\frac{\text{cm}}{\text{s}}$), and average ISI of 150 ms (Gupta, Van Der Meer, Touretzky, & Redish, 2012), which bounds neural sampling periods for space, implying that ΔL should lie somewhere in [0.15, 15] cm. Code rates considered in this work assume $\Delta L < 15$ cm. To satisfy curiosity and probe rate-dependent phenomena at even greater rates, the smallest ΔL considered is 0.0022 cm.

In order to investigate how the fundamental limits on denoisability of the code scale with the number of pattern neurons (i.e., grid and place cells), we compute d as a function of N , independently varying P , M , $\{J_i\}_{i \in \{1, \dots, M\}}$, fixing other parameters. As illustrated in Figure 6, minimum distance increases exponentially with increases in N due to increases in the number of place cells, P , and number of grid cells per module, J_i . In contrast, increases of M

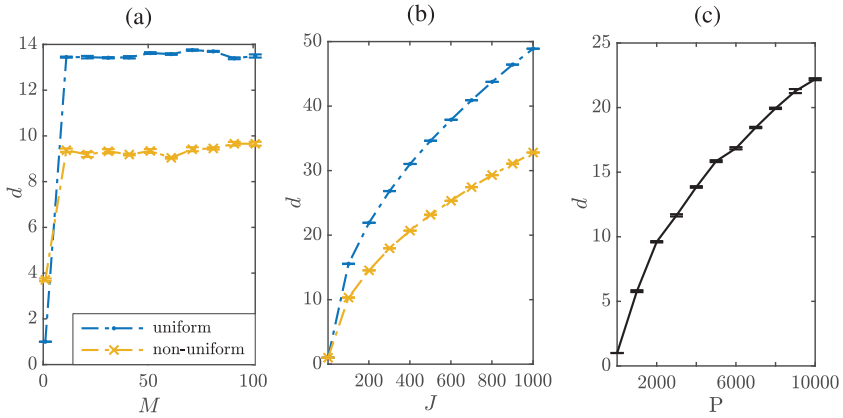


Figure 6: (a) Minimum distance, d , versus number of grid modules, M , with $J = 20$ grid cells in the first module, five-fold replication of grid cell phases (i.e., $\mu_p = 5$), uniform and nonuniform allocations of grid cells to modules, and no place cells. In the uniform (resp. nonuniform) case, $N = M \cdot 20$ (resp. $N = \sum_{m=1}^M \lfloor \frac{20}{\lambda^{m-1}} \rfloor$). (b) Minimum distance, d , versus number of neurons in the first module, J , for uniform and nonuniform allocations of grid cells to modules, and no place cells. N is calculated as in panel a. (c) Minimum distance, d , versus number of place cells, P , for a population of place cells resembling those observed in experiment (Nadel & O’Keefe, 1978; Muessig et al., 2015; Aronov et al., 2017) with no grid cells (i.e., $N = P$). A complete list of parameters is in appendix F.

past a critical value cease to improve minimum distance because the spatial scale at which higher-order modules represent position fails to capture relevant differences in location encoded. Notably, when all other parameters are fixed, nonuniform allocations of grid cells to modules provide a code with inferior minimum distance. This is a consequence of the greater number of pattern neurons in the uniform case and can be considered the loss incurred in exchange for an increase in coding efficiency, (measured by number of neurons used to encode position), as discussed in Mosheiff et al. (2017).

3.2 Code Learning Results. In order to study how algorithm 1, neural learning, affects the denoising network, we assess the changes in connectivity that it produces. Typical learned connectivity matrices and their associated normalized degree distributions (empirical distributions of the number of connections emanating from pattern neurons, normalized to the total number of pattern neurons, N) are found in Figures 7 and 8. These demonstrate that for a typical hybrid code, the clustered network has a sparser connectivity, with less variability in its sparsity compared to the

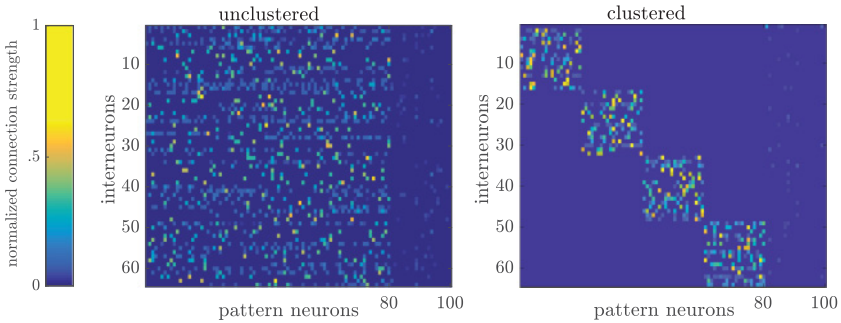


Figure 7: Image of typical synaptic weight matrices learned by two denoising networks for a hybrid code with $M = 4$, $J = 20$, and $P = 20$. In this example, $N = 100$ and $n_c = 64$.

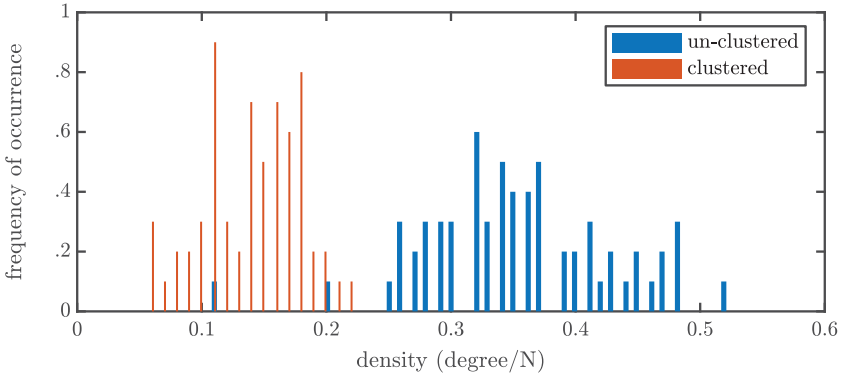


Figure 8: Degree distributions of the connectivity matrices shown in Figure 7.

unclustered network. This is because clustering enforces a tighter limit on the number of pattern neurons to which a constraint neuron may connect. We simulated an ensemble of 4 modules of 20 grid cells each, together with 20 place cells, which produced the following connectivity matrices and associated degree distributions. Interestingly, in both cases, there are place cells (i.e., pattern neurons with index exceeding 80) that are left unconnected to grid modules via constraint neurons. An illustration of the learned weights matrix corresponding to a randomly clustered denoising network was omitted, as it is sparser, but otherwise very similar to that of the unclustered weights image.

Figure 9 depicts the average connection strength between place cells and grid modules, where the connection strength between place cell p and grid module m is defined as $\frac{1}{n_c} (\sum_{(i,j)} |w_{i,j} w_{i,p}|)$, where i indexes constraint

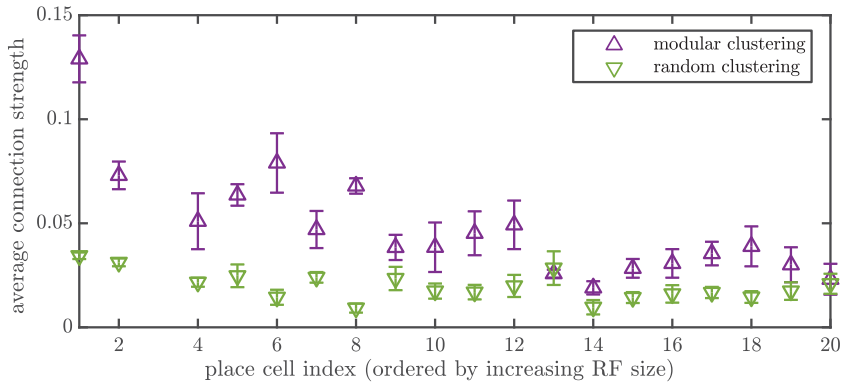


Figure 9: Average connectivities between place cells (index along the x -axis) and grid modules for configurations with $M = 4$, $J = 20$, $P = 20$, $N = 100$, and $n_c = 64$. Place cell indices are ordered from smallest to largest receptive field size; grid cell phases were uniformly distributed on the environment. Grid cells were uniformly distributed to modules. Connection strengths depicted are averaged over 50 networks.

neurons, and j indexes grid cells in module m . Note here that connectivity implies not direct synaptic connection but effective connectivity through constraint neurons. Results were obtained from configurations with $M = 4$, $J = 20$, and $P = 20$; connectivities depicted are averaged over 50 networks. Place cells are ordered by increasing the size of the receptive field. This trend appears for any $\mu_p > 1$ (i.e., whenever the responses of at least some grid cells are replicated by instantiating multiple grid cells with the same phase in the same module). In the modularly clustered case, average connectivity (between place cells and all grid modules) appears to decrease with increasing place cell size, as compared to a random clustering that produces nearly the same connectivity for each place cell. This phenomenon was not observed when grid cell phases and orientation offsets were chosen randomly and does not appear in the unclustered configuration.

3.3 Denoising and Decoding Results. In order to study the relationship between coding theoretically relevant variables, population parameters, denoising network configuration, and fidelity of the hybrid code's representation of space, we empirically evaluate the denoising network's performance. To measure the effectiveness of the denoising network, we first perturb the states (i.e., firing rates) of the grid and place cells by incrementing or decrementing randomly and clipping to the boundaries of $[0, Q - 1]$. A pattern error occurs if, after denoising, any entry of the denoised pattern differs from the corresponding component of the original pattern. A symbol error occurs each time any symbol of the denoised

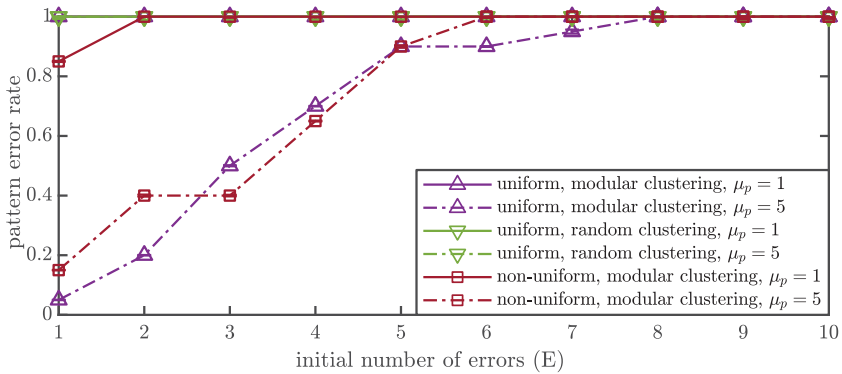


Figure 10: Pattern error rate, P_{pe} , versus initial number of errors, E , for a clustered hybrid code with $M = 4$, $J = 20$ (in the networks with grid cells allocated to modules nonuniformly, $J = 32$ to ensure $N = 90$ pattern neurons) and $P = 10$. Other configurations assessed (specifically, those with random redundancy parameters and those with an unclustered denoising network) have $P_{pe} = 1$ for any initial number of errors. In each case considered, $n_c = R \cdot N$. This choice is discussed in further detail in appendix A.

pattern differs from the corresponding symbol of the correct pattern. For identical populations of grid and place cells ($M = 4$, $J = 20$, and $P = 10$) in pattern error rate, the clustered network dramatically outperforms the unclustered (when the grid cells have sufficient redundancy), and the modular clustering scheme always outperforms the random clustering scheme. By fixing the size of the populations we compare, we ensure no improvement in d results from a larger N . Figure 10 depicts pattern error rate (P_{pe}) for a clustered hybrid code, with varying phase multiplicity. The missing configuration (consisting of a randomly clustered network with a code with a nonuniform allocation of grid cells to modules) had a 100% pattern error rate for every nonzero number of initial errors. This shows that for a small number of initial errors, the full pattern of population activity corresponding to the correct location may be recovered, but in general, this is rarely possible. That only the modularly clustered denoising networks are able to achieve low P_{pe} shows that the biological organization of grid cells into discrete modules is important for high-quality self-localization in the presence of noise. Further, clustering is the only way to achieve such a small P_{pe} , since no unclustered denoising network consistently reduced P_{pe} below 0.99. It is surprising that the modularly clustered denoising mechanism achieves a better P_{pe} when denoising hybrid codes with uniform allocations of grid cells to modules (as compared to nonuniform allocations of grid cells to modules), as Figure 5 demonstrates that such codes tend to have a larger minimum distance at any rate probed. This result also demonstrates that

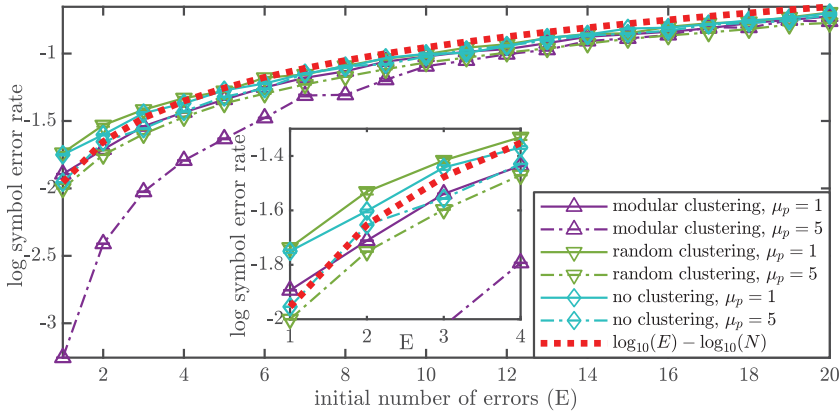


Figure 11: Symbol error rate, P_{se} , versus initial number of errors, E , for clustered and unclustered hybrid codes. Here, each code utilizes a uniform distribution of grid cells to modules and deliberately chosen spatial phases and orientations (i.e., so as to mirror those observed in Stensola et al., 2012). $N = 90$. The dotted red curve (with no markers), $\log_{10}(\frac{\text{initial number of errors}}{N})$, is a threshold between regions of desirable and unacceptable P_{se} (i.e., $\log_{10}(P_{se})$ for a network that performs no denoising). For $E < 10$, configurations are ordered by increasing P_{se} as modular clustering, $\mu_p = 5$; random clustering, $\mu_p = 5$; modular clustering, $\mu_p = 1$; no clustering, $\mu_p = 5$; no clustering, $\mu_p = 1$; and random clustering, $\mu_p = 1$. As shown in the inset, for $E < 5$, configurations with $\mu_p = 1$ produce symbol error rates above $\log_{10}(\frac{\text{initial number of errors}}{N})$, that is, they increase the number of symbol errors.

whether grid cells are distributed uniformly to modules has a smaller impact on P_{pe} than μ_p . That the codes with larger μ_p tend to outperform those with $\mu_p = 1$ is also surprising, since at high rates (in Figure 10, $r \approx 10^3$), codes with larger μ_p are restricted to smaller d .

Figure 11 shows symbol error rates of hybrid codes for several configurations with deliberately chosen grid cell phases and orientations (i.e., so as to mirror those observed in Stensola et al. (2012)). This demonstrates that generally, clustered denoising networks do not offer improved symbol error rate, P_{se} , compared to their unclustered counterparts. However, for a small initial number of errors, when the grid cells exhibit sufficient redundancy in their phases, a randomly clustered denoising network is outperformed only by a modularly clustered network. Figure 12 shows P_{se} for a hybrid code with deliberately chosen phases and orientations, denoised by a modularly clustered network. Consistent with observations on pattern error rate, hybrid codes with grid cells uniformly allocated to modules achieve better P_{se} . This may result from the fact that d is larger for such codes when μ_p is small. However, this explanation is incomplete as when

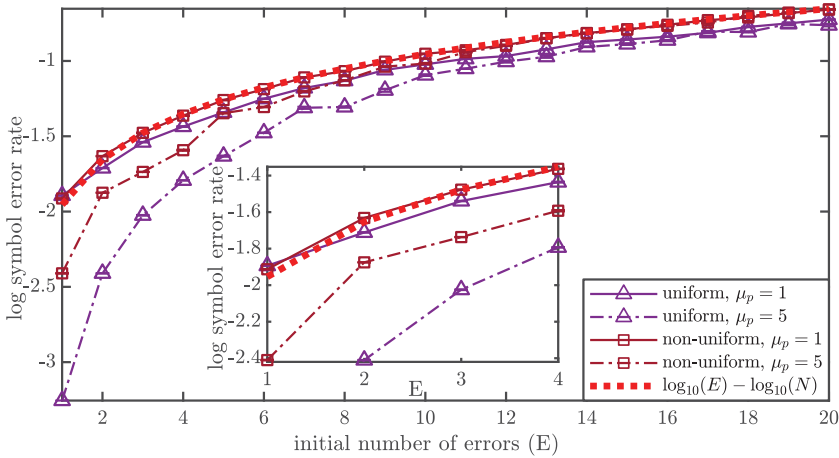


Figure 12: Symbol error rate, P_{se} , versus initial number of errors, E , for uniform and nonuniform clustered hybrid codes. Here, each denoising network employs the modular clustering scheme. In each case, $N = 90$, $M = 4$, $P = 10$. For grid cells allocated to modules uniformly (resp. nonuniformly), $J = 20$ (resp. $J = 32$). The dotted red curve, $\log_{10}(\frac{\text{initial number of errors}}{N})$, is a threshold between regions of desirable and unacceptable P_{se} (i.e., $\log_{10}(P_{se})$ for a network that performs no denoising).

$\mu_p = 5$, a code generated by a nonuniform allocation of grid cells to modules, tends to achieve a larger minimum distance than those generated by uniform allocation of grid cells to modules. Plotted in both Figures 11 and 12 is a dotted red curve, $\log_{10}(\frac{\text{initial number of errors}}{N})$. This curve is a threshold between regions of desirable and unacceptable P_{se} (i.e., $\log_{10}(P_{se})$ for a network that performs no denoising). To see this, consider a denoising network that does not change the initial number of errors, E . For this network, $P_{se} = \frac{E}{N}$, so $\log_{10}(P_{se}) = \log_{10}(E) - \log_{10}(N)$. Surprisingly, Figure 11 shows that for a small initial number of errors, configurations with $\mu_p = 1$ have $\log_{10}(P_{se})$ above this threshold, that is, they increase the number of symbol errors! Figure 12 quantifies the loss incurred by the nonuniform allocation of grid cells to modules (i.e., $J_m \propto \frac{1}{\lambda^{m-1}}$) for a modularly clustered denoising network. Note that both grid cell allocation schemes produce networks that introduce additional errors during denoising when $\mu_p = 1$ and $E = 1$, as these conditions result in $P_{se} > \frac{E}{N}$. Note that for $E > 1$, no network introduces extraneous errors by denoising. Additionally, networks with $\mu_p = 5$ dramatically outperform those with $\mu_p = 1$, when E is small.

Figure 13 shows MSE of different decoding processes after denoising for a hybrid code with $M = 4$, $J = 20$, $P = 10$, and $\mu_p = 5$, for deliberately chosen grid cell parameters (i.e., so as to mirror those observed in Stensola et al., 2012). This plot demonstrates that an ideal observer decoder that considers

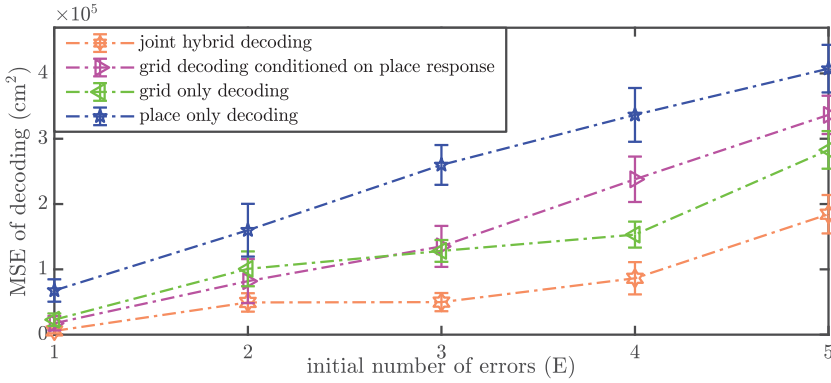


Figure 13: MSE of decoding after denoising for a hybrid code with $M = 4$, $J = 20$, $P = 10$, and $\mu_p = 5$, and deliberately chosen grid cell parameters (i.e., so as to mirror those observed in Stensola et al., 2012). Grid cells are allocated to modules uniformly, so $N = 90$. Here, $n_c = 64$.

information from all cells outperforms all others for any initial number of errors. This disparity may in part be accounted for by the difference between the number of grid cells and the number of place cells. Figure 14 shows MSE of joint hybrid decoding after denoising for a hybrid code with $\mu_p = 5$, for the configurations that achieved the best error correction performance in both P_{pe} and P_{se} . This plot demonstrates that the code with grid cells distributed to modules uniformly with a modularly clustered denoising network achieves the best decoding performance, outperforming its nonuniformly arranged analogue. Since the code with a nonuniform allocation of grid cells to modules had a larger minimum distance (compared to the same code with a uniform allocation of grid cells to modules), this result challenges our earlier hypothesis that codes with nonuniform allocations of grid cells across modules may be denoised more effectively. This is especially remarkable since in section 3.1, we demonstrated that these codes achieve larger minimum distance for identical N at large r , such as the rate considered in Figure 14. Further, this demonstrates (in a natural metric of the stimulus space) that in the most redundant hybrid code considered, a modularly clustered denoising network is far superior to a randomly clustered or unclustered one. Interestingly, for a small number of initially erroneous pattern neurons, the loss (in MSE) due to a lack of modular clustering is much greater than the loss due to nonuniformity.

4 Discussion

We demonstrated that both dense and sparse hybrid codes may be constructed by proper choice of grid and place cell parameters. We also showed

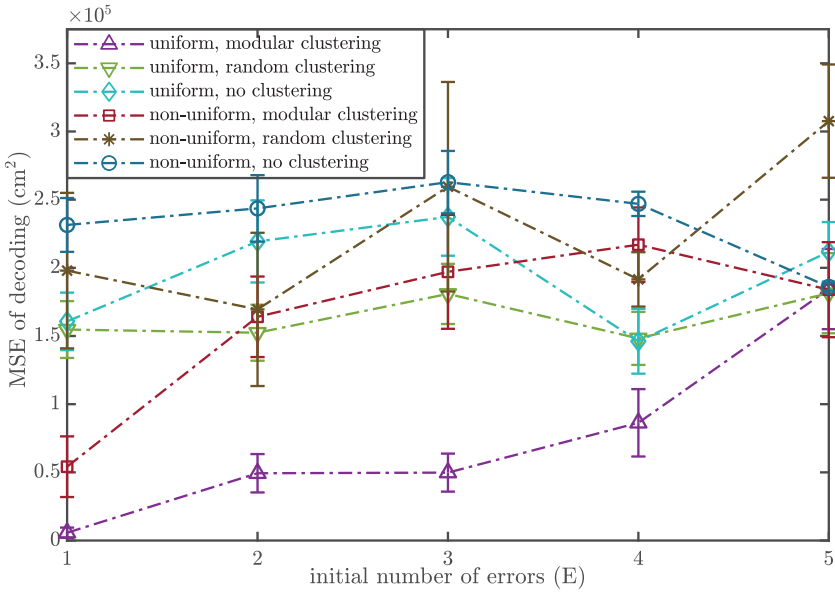


Figure 14: MSE of decoding after neural denoising (cm^2) versus initial number of errors for a hybrid code with $\mu_p = 5$, $N = 90$.

that in the presence of neural noise, the activity of only configurations with sufficient redundancy in the grid cell component of the code may be consistently denoised. It is somewhat counterintuitive that populations with replicated grid cell responses (i.e., $\mu_p > 1$) produce a more noise-resilient code (as shown in the denoising performance results). This is surprising because the populations with uniformly allocated grid cells and largest d are those with unique spatial phases (i.e., $\mu_p = 1$; see Figure 5). This result is counterintuitive (in the biological sense) as in Hafting et al. (2005), it is noted that the distribution of grid cell phases observed in experiment did not deviate significantly from uniformity. Wennberg (2015) revealed that the distribution of spatial phase offsets of grid cells may be significantly nonuniform. The data set from which this conclusion is drawn was obtained from rat 14147 in Stensola et al. (2012). Our results imply that this observed nonuniformity in distribution of grid cell phases provides value in denoisability and accuracy of decoding.

Our results reveal another surprise in Figure 5, in which, for $\mu_p > 1$, codes with nonuniform allocations of grid cells to modules achieve demonstrably larger d . However, in Figure 12, the networks with $\mu_p = 5$ and grid cells allocated to modules uniformly achieve the smallest P_{se} . Further, in Figures 10 and 14, for a small number of initially erroneously signaling

neurons (E), these networks outperform those with grid cells allocated to modules nonuniformly. These observations demonstrate that the hybrid code for space may trade off improvements in denoising performance (in d) for efficiency of encoding (r) by distributing grid cells to modules nonuniformly, as suggested in Mosheiff et al. (2017).

Hybrid codes of widely varying rank, minimum distance, and code rate (R , d , and r , respectively) may be instantiated by choosing appropriate parameters for the populations of grid and place cells, a fact that showcases the code's adaptability. This means that grid and place cells may participate in neural computations that rely on assumptions other than those presented here, which insist on a low-dimensional code space and a sparse connectivity matrix. It is particularly difficult to characterize the trade-off between code rate and d , presented in Figure 5, as it indicates that for biologically reasonable values of r , increases in μ_p should reduce a code's minimum distance, d (a fundamental limit of the code's denoisability). Surprisingly, the configuration with uniformly allocated grid cells and $\mu_p = 5$ tends to outperform the others in P_{pe} , P_{se} , and MSE. It is possible that the denoising networks presented here are incapable of achieving the codes' error correction capacities in the cases considered. This would allow for characteristics endowed by a larger μ_p to effect the stark differences observed in denoising efficacy and decoding accuracy. Furthermore, this explanation seems likely, as coding theory suggests that the maximum number of correctable errors in a linear block code (as a function of d) can be computed as $t = \lfloor \frac{d-1}{2} \rfloor$ (Lin & Costello, 1983). For example, the strongest code (as measured by largest value of d in Figure 5) achieves $d \approx 5$ for intermediate r , so $t \approx 2$. Figure 10 corroborates this in demonstrating that pattern error rate exceeds 0.5 (and quickly saturates at 1) for more than two errant pattern neurons.

We demonstrate that the chosen denoising network architecture performs satisfactorily for hybrid codes that fit its requirements regarding rank and poorly for those that do not. Additionally, we assessed average connectivity between place cells of varying receptive field sizes and modules of grid cells by analyzing the learned connectivity matrix. This analysis demonstrates that our model place cells of smaller receptive field size are more strongly connected to grid modules and that they are most strongly connected to grid modules of the smallest scale. Moreover, this result presents a physiologically testable hypothesis. While difficult, two-photon microscopy has been successfully employed to accurately image the microscopic structure of nervous tissue (Svoboda & Yasuda, 2006). One way to estimate connection strength between real neurons is to count the number of boutons expressed on the presynaptic neurons, assuming that weight should be proportional to this number, though there may be simpler ways to estimate connection strength (Bi & Poo, 1998). Thus, if groups of place cells connected via constraint neurons to several distinct grid modules may be identified, this theoretical prediction—that connectivity between the hippocampus and MEC will decrease along the dorsoventral axis—can be

confirmed or refuted. Another interesting experiment is made possible by advances in optogenetics, which enable single cell resolution of network activity for a population of inoculated cells (e.g., a collection of grid cells, as in Sun et al., 2015). While technically challenging due to the physical separation of each population in the brain, it should be possible to image simultaneous activity of grid and place cells at high temporal precision (Grewe, Langer, Kasper, Kampa, & Helmchen, 2010). From these measurements, for a set of quantized locations, simultaneous firing rates may be estimated (Theis et al., 2016). Then the rank, rate, and minimum distance of this empirical codebook may be computed to offer insight into limits of noise tolerance of real spatial navigation circuitry. Of particular interest is discovering the extent to which neural noise transiently varies such attributes for grid and place cells in real brains and how these coding theoretic properties adapt (if at all) to changes in speed, context, and other variables.

In Figures 10, 12, 13, and 14, we demonstrate the differences in performance of each network structure and of the various decoding algorithms. The universal improvements from place-only decoding to joint-hybrid decoding show that highly accurate position estimation can be significantly more difficult without both populations of cells. The discrepancy between grid-only decoding and grid decoding conditioned on place response shows that even utilizing place cell information indirectly (by eliminating candidate locations deemed impossible given the state of the place cell population) yields a sizable improvement in decoding accuracy when there are many place cells or when place cells are less noisy than grid cells. That the modularly clustered networks tend to best the corresponding randomly clustered networks implies that the physiological organization of grid cells by their spatial scale may provide a computational advantage in denoising and decoding. This notion is further supported by the observation that a randomly clustered network sometimes introduces additional errors by attempting to denoise, as shown in Figure 11. This may be because the unclustered network is essentially a randomly clustered network that does not take advantage of synergistic cluster computing. In any cluster, both grid cells and place cells are able to correct each other's errant activity. However, under modular clustering, in order for a grid cell in module i to correct the activity of a grid cell in a different module j , the activity of each neuron in module i must be correct so that the activity of place cells (connected to both modules i and j) will contradict and correct the erroneous activity.

It should be noted that the denoising constraint neurons are a hypothetical construct and need not reside in the hippocampus or MEC in order to execute the previously described computations. Our conception of these constraint nodes is as single units. However, these may represent larger networks of neurons performing identical computations. Furthermore, this work is not intended to convince readers of the necessity or existence of these cells, only to demonstrate tangible coding theoretic advantages

conferred by constraint neuron moderated communication between grid and place cells. Additionally, some models of development of the grid and place cell networks demonstrate dependence between properties of each population's apparent receptive fields that our model is unable to capture (Monaco & Abbott, 2011). Thus, coding theoretic results presented here are confined to consideration of a more static code than what is often observed in recordings of real neuronal populations. While our model is limited in the sense that neurons are defined functionally (in contrast with biophysical models, where behavior emerges from the time evolution of the model's physics), the learning algorithms considered are analogous to a Hebbian plasticity, and operations required for denoising can be feasibly implemented by networks of real neurons (if not by single units). Hence, the results discussed here have potential implications about neural codes for other continuously valued stimuli (e.g., pitch of an auditory signal, another variable encoded in the mammalian hippocampus; Aronov et al., 2017).

Contributions of this work include the coding model and denoising systems themselves, as a framework in which to characterize limits on fidelity of cooperating neural codes subject to noise (for physical position or other variables such as the auditory pitch code studied in Aronov et al., 2017), and improved clarity about how parameterization of grid and place cell populations affects these fundamental information and coding theoretic limits. Further development along these threads of investigation of neural codes for space includes studying coding theoretic properties of more complete navigational codes, including head direction cells, boundary vector cells, and time cells (Lever, Burton, Jeewajee, O'Keefe, & Burgess, 2009; Salz et al., 2016; Taube, Muller, & Ranck, 1990). It would be most interesting to probe coding and information-theoretic properties of place cells that encode 3D space as demonstrated to reside in the bat hippocampus (Yartsev & Ulanovsky, 2013). Even with these classes of neuron, the hybrid code might be unable to encode and denoise path information without supplementary structure to process its sequentiality. One strong candidate solution for this is to include so-called hippocampal time cells. Just as place cells code for distinct locations on paths through space, time cells encode ordered moments in a temporally ordered sequence of events, precisely the information, which, when coupled with location, should allow for the encoding of paths (MacDonald, Lepage, Eden, & Eichenbaum, 2011).

Appendix A: Network Size

N , the number of pattern neurons in a network, is the sum of the sizes of the constituent grid and place cell populations. When grid cells are allocated to modules uniformly, $N = P + M \cdot J$. Otherwise, $N = P + \sum_{m=1}^M \frac{J}{\lambda^{m-1}}$.

Since a code of normalized rank R admits at most $N(1 - R)$ unique constraint equations (i.e., linear combinations of pattern neuron activities that

evaluate to zero only when this activity forms a code word and the functions computed by constraint neurons), we use $n_c = N(1 - R)$.

Appendix B: Subspace Learning

Oja and Karhunen (1985) propose an algorithm that is capable of computing a basis for the null space of a random matrix, A , which is assumed to be the expected value of sample matrices, A_t . The update rule for the matrix whose columns are the resulting basis vectors is

$$\tilde{W}_t = W_{t-1} + A_{t-1}W_{t-1}\alpha_{t-1}, \quad (\text{B.1})$$

$$W_t = \tilde{W}_t R_t^{-1}, \quad (\text{B.2})$$

where α_t is a diagonal (and compatible) matrix of gain factors. As in Oja and Karhunen (1985), equations A.1 and A.2 may be rewritten as operations on column vectors, \mathbf{w}_t ,

$$\tilde{\mathbf{w}}_t = \mathbf{w}_{t-1} + \alpha_{t-1}A_{t-1}\mathbf{w}_{t-1}, \quad (\text{B.3})$$

$$\mathbf{w}_t = \frac{\tilde{\mathbf{w}}_t}{\|\tilde{\mathbf{w}}_t\|}, \quad (\text{B.4})$$

in which α_t is the gain factor corresponding to the current column. This number may be equivalently understood as a learning rate. Indeed Xu, Krzyzak, and Oja (1991), show that for appropriate choices of A_t , the update rule is a form of anti-Hebbian learning. Oja and Karhunen (1985) prove convergence of this algorithm to the eigenvectors of A corresponding to the largest eigenvalues. Further, when A_t is replaced by $-A_t$, \mathbf{w}_t converges to the eigenvectors of A corresponding to the smallest eigenvalues. Oja and Karhunen (1985) demonstrate that by combining equations A.3 and A.4, expanding as a power series in α_t , and ignoring second (and higher) order terms, we arrive at

$$\mathbf{w}_t = \mathbf{w}_{t-1} + \alpha_{t-1} \left(A_{t-1}\mathbf{w}_{t-1} - \frac{\mathbf{w}_{t-1}^T A_{t-1} \mathbf{w}_{t-1}}{\mathbf{w}_{t-1}^T \mathbf{w}_{t-1}} \mathbf{w}_{t-1} \right). \quad (\text{B.5})$$

(Salavati et al., 2014) choose $A_t = (\mathbf{x}_t^T \mathbf{x}_t)P_{\mathbf{x}_t} = \mathbf{x}_t \mathbf{x}_t^T$, the product of projections onto the space spanned by \mathbf{x}_t , and define $y_t = \mathbf{x}_t^T \mathbf{w}_t = \mathbf{w}_t^T \mathbf{x}_t$. Oja and Karhunen (1985) mentioned that this update rule finds eigenvectors corresponding to the largest eigenvalue of A_t or those corresponding to the smallest eigenvalues of $-A_t$ when this matrix is used instead. Since A_t is a projection matrix, it has rank 1. Thus, it has one eigenvector with nonzero eigenvalue, \mathbf{x}_t , and $\dim(\mathbf{x}) - 1$ eigenvectors with eigenvalue 0. Each of these eigenvectors, \mathbf{v} , is guaranteed to be perpendicular to \mathbf{x} because $A_t \mathbf{v} = 0\mathbf{v} = \mathbf{0}$, that is, the \mathbf{v} 's projection onto \mathbf{x} has magnitude 0. By choosing $\mathbf{x}_t \in \mathcal{C}$, with

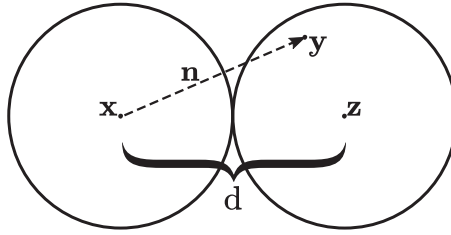


Figure 15: An illustration of the relationship between minimum distance of a code and its resilience to noise.

the aforementioned choice for A_t , this algorithm should compute vectors approximately perpendicular to the code space.

Now, we may rewrite equation A.5 as

$$\begin{aligned}
 \mathbf{w}_t &= \mathbf{w}_{t-1} - \alpha_{t-1} \mathbf{x}_{t-1} \mathbf{x}_{t-1}^T \mathbf{w}_{t-1} + \alpha_{t-1} \frac{\mathbf{w}_{t-1}^T \mathbf{x}_{t-1} \mathbf{x}_{t-1}^T \mathbf{w}_{t-1}}{\|\mathbf{w}_{t-1}\|^2} \mathbf{w}_{t-1} \\
 &= \mathbf{w}_{t-1} - \alpha_{t-1} y_{t-1} \mathbf{x}_{t-1} + \alpha_{t-1} \frac{y_{t-1}^2}{\|\mathbf{w}_{t-1}\|^2} \mathbf{w}_{t-1}.
 \end{aligned}
 \tag{B.6}$$

To obtain a sparse basis for $\text{null}(\mathcal{C})$, one may add to equation A.6 a regularizing term that penalizes nonsparse solutions. In particular, using $\eta \Gamma(\mathbf{w}_{t-1}, \theta_{t-1})$, as considered in Salavati et al. (2014), to arrive at

$$\mathbf{w}_t = \mathbf{w}_{t-1} - \alpha_{t-1} \left(y_{t-1} \left(\mathbf{x}_{t-1} - \frac{y_{t-1} \mathbf{w}_{t-1}}{\|\mathbf{w}_{t-1}\|^2} \right) \right) - \alpha_{t-1} \eta \Gamma(\mathbf{w}_{t-1}, \theta_{t-1}). \tag{B.7}$$

Appendix C: Structure of the Performance Testing Simulations _____

In order to evaluate the performance of the denoising mechanisms proposed here, we first generate codes from the parameters considered in appendix E. Then algorithm 1 is applied to the chosen denoising network. After learning is complete, in sequence, C randomly chosen code words are corrupted and presented to the network to denoise using algorithms 2 and 3. After the denoising process is complete, the denoised pattern is assessed and performance is computed incrementally.

Appendix D: How Minimum Distance Limits Ideal Decoding _____

Suppose \mathbf{x} and \mathbf{z} are two code words separated by their code’s minimum distance, d , as shown in Figure 15, and that during transmission of \mathbf{x} , our channel adds noise, \mathbf{n} . If the magnitude of this noise ($\|\mathbf{n}\|$) exceeds $\frac{d}{2}$, the received word (\mathbf{y}), may lie a distance $t < \frac{d}{2}$ away from \mathbf{z} . As a result, a

minimum distance decoder (which outputs the code word nearest to the received word) incorrectly declares that \mathbf{z} was transmitted. If error events at the symbols of code words are independent and the probability of error does not depend on the position of the symbol in question, as long as this probability does not exceed $\frac{1}{2}$, minimum distance decoding is maximum likelihood decoding.

Appendix E: Parameter and Variable Definitions

We present a table of definitions considered in this article.

Parameter	Definition
L	Length of simulated square arena
C	Number of locations (code words) that comprise the code in question
M	Number of modules of grid cells
J	Number of neurons in the first module of grid cells
P	Number of place cells
μ_p	Number of grid cells with the same phase in the same module
λ	Scaling ratio between grid modules
λ_i	Scale of the i th grid module
$\theta_{m,j}$	Orientation offset of the j th grid cell of module m
α_t	Learning rate at iteration t
ϵ	Learning completion threshold
η	Sparsity penalty coefficient
\mathcal{C}	Codebook: A collection of code words formed by the simultaneous activity of pattern neurons
$\bar{\mathcal{C}}$	Codebook matrix constructed by placing elements of \mathcal{C} in rows
R	Normalized rank of the code, $\frac{\text{rank}(\bar{\mathcal{C}})}{N}$
r	Normalized code rate (i.e., number of locations represented per neuron): $\frac{C}{N}$
d	Minimum distance of a code (minimum among all distances between code words)

Appendix F: Choices of Parameters

In learning, normalized weights are initialized randomly with degree $\lceil 4 \log_e(n) \rceil$, where n is the length of the weight vector. We used, $\theta_0 = 0.031$, $\eta = 0.075$, and $\alpha_0 = 0.95$. In denoising, we set $\phi = 0.95$. Unless otherwise noted, dependent variables measured and computed are mean values averaged over 100 networks. Error bars represent standard error of the mean.

Here we present a table of parameters indexed by figure in this article. N/A means either that this parameter was varied or was not used.

Table 1: Parameters Indexed by Figure.

Figure	L (cm)	C	M	J	P	λ	λ_1 (cm)	μ_p	ϵ
3	300	1000	4	20	NA	$\sqrt{2}$	40	NA	NA
4	300	1000	NA	NA	NA	NA	40	NA	NA
5	300	NA	NA	NA	NA	$\sqrt{2}$	40	NA	NA
6	300	NA	4	20	10	$\sqrt{2}$	40	NA	NA
7	300	NA	4	20	NA	$\sqrt{2}$	40	5	NA
8	300	10^5	4	20	20	$\sqrt{2}$	40	5	$C10^{-3}$
9	300	10^5	4	20	20	$\sqrt{2}$	40	5	$C10^{-3}$
10	300	10^5	4	20	20	$\sqrt{2}$	40	5	$C10^{-3}$
11	300	10^5	4	20	10	$\sqrt{2}$	40	NA	$C10^{-3}$
12	300	10^5	4	20	10	$\sqrt{2}$	40	NA	$C10^{-3}$
13	300	10^5	4	20	10	$\sqrt{2}$	40	NA	$C10^{-3}$
14	300	10^5	4	20	10	$\sqrt{2}$	40	5	$C10^{-3}$
15	300	10^5	4	20	10	$\sqrt{2}$	40	5	$C10^{-3}$

Acknowledgments

This work is supported in part by National Science Foundation grants IIS-1464349 and CCF-1748585.

References

- Amit, D. J., & Treves, A. (1989). Associative memory neural network with low temporal spiking rates. *Proceedings of the National Academy of Sciences*, 86(20), 7871–7875.
- Aronov, D., Nevers, R., & Tank, D. W. (2017). Mapping of a non-spatial dimension by the hippocampal–entorhinal circuit. *Nature*, 543(7647), 719.
- Bi, G.-q., & Poo, M.-m. (1998). Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of Neuroscience*, 18(24), 10464–10472.
- Bonnevie, T., Dunn, B., Fyhn, M., Hafting, T., Derdikman, D., Kubie, J. L., . . . Moser, M.-B. (2013). Grid cells require excitatory drive from the hippocampus. *Nature Neuroscience*, 16(3), 309–317.
- Chen, J., & Fossorier, M. P. (2002). Near optimum universal belief propagation based decoding of low-density parity check codes. *IEEE Transactions on Communications*, 50(3), 406–414.
- Curto, C., Itskov, V., Morrison, K., Roth, Z., & Walker, J. L. (2013). Combinatorial neural codes from a mathematical coding theory perspective. *Neural Computation*, 25, 1891–1925.
- Declercq, D., & Fossorier, M. (2007). Decoding algorithms for nonbinary LDPC codes over GF(q). *IEEE Transactions on Communications*, 55(4), 633–643.

- Doeller, C. F., Barry, C., & Burgess, N. (2010). Evidence for grid cells in a human memory network. *Nature*, *463*(7281), 657.
- Eccles, J. (1976). From electrical to chemical transmission in the central nervous system. *Notes and Records of the Royal Society of London*, *30*(2), 219–230.
- Ekstrom, A. D., Kahana, M. J., Caplan, J. B., Fields, T. A., Isham, E. A., Newman, E. L., & Fried, I. (2003). Cellular networks underlying human spatial navigation. *Nature*, *425*(6954), 184–188.
- Fiete, I. R., Burak, Y., & Brookings, T. (2008). What grid cells convey about rat location. *Journal of Neuroscience*, *28*(27), 6858–6871.
- Fyhn, M., Hafting, T., Witter, M. P., Moser, E. I., & Moser, M.-B. (2008). Grid cells in mice. *Hippocampus*, *18*(12), 1230–1238.
- Grewe, B. F., Langer, D., Kasper, H., Kampa, B. M., & Helmchen, F. (2010). High-speed in vivo calcium imaging reveals neuronal network activity with near-millisecond precision. *Nature Methods*, *7*(5), 399–405.
- Gupta, A. S., Van Der Meer, M. A., Touretzky, D. S., & Redish, A. D. (2012). Segmentation of spatial experience by hippocampal theta sequences. *Nature Neuroscience*, *15*(7), 1032.
- Hafting, T., Fyhn, M., Molden, S., Moser, M.-B., & Moser, E. I. (2005). Microstructure of a spatial map in the entorhinal cortex. *Nature*, *436*(7052), 801.
- Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, *79*(8), 2554–2558.
- Jacobs, J., Weidemann, C. T., Miller, J. F., Solway, A., Burke, J. F., Wei, X.-X., . . . Kahana, M. J. (2013). Direct recordings of grid-like neuronal activity in human spatial navigation. *Nature Neuroscience*, *16*(9), 1188–1190.
- Karbasi, A., Salavati, A. H., & Shokrollahi, A. (2013). Iterative learning and denoising in convolutional neural associative memories. In *Proceedings of the 30th International Conference on Machine Learning* (pp. 445–453). New York: ACM.
- Karbasi, A., Salavati, A. H., Shokrollahi, A., & Varshney, L. R. (2014). Noise facilitation in associative memories of exponential capacity. *Neural Computation*, *26*, 2493–2526.
- Lever, C., Burton, S., Jeewajee, A., O’Keefe, J., & Burgess, N. (2009). Boundary vector cells in the subiculum of the hippocampal formation. *Journal of Neuroscience*, *29*(31), 9771–9777.
- Lin, S., & Costello, D. (1983). *Error-correcting codes*. Upper Saddle River, NJ: Prentice Hall.
- MacDonald, C. J., Lepage, K. Q., Eden, U. T., & Eichenbaum, H. (2011). Hippocampal “time cells” bridge the gap in memory for discontinuous events. *Neuron*, *71*(4), 737–749.
- McEliece, R. J., Posner, E. C., Rodemich, E. R., & Venkatesh, S. S. (1987). The capacity of the Hopfield associative memory. *IEEE Transactions on Information Theory*, *33*(4), 461–482.
- Monaco, J. D., & Abbott, L. F. (2011). Modular realignment of entorhinal grid cell activity as a basis for hippocampal remapping. *Journal of Neuroscience*, *31*(25), 9414–9425.

- Mosheiff, N., Agmon, H., Moriel, A., & Burak, Y. (2017). An efficient coding theory for a dynamic trajectory predicts non-uniform allocation of entorhinal grid cells to modules. *PLoS Computational Biology*, *13*(6), e1005597.
- Muessig, L., Hauser, J., Wills, T. J., & Cacucci, F. (2015). A developmental switch in place cell accuracy coincides with grid cell maturation. *Neuron*, *86*(5), 1167–1173.
- Nadel, L., & O'Keefe, J. (1978). *The Hippocampus as a Cognitive Map*. Oxford: Clarendon.
- Oja, E., & Karhunen, J. (1985). On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix. *Journal of Mathematical Analysis and Applications*, *106*(1), 69–84.
- Oja, E., & Kohonen, T. (1988). The subspace learning algorithm as a formalism for pattern recognition and neural networks. In *IEEE International Conference on Neural Networks, 1988* (pp. 277–284). Piscataway, NJ: IEEE.
- O'Keefe, J. (1976). Place units in the hippocampus of the freely moving rat. *Experimental Neurology*, *51*(1), 78–109.
- O'Keefe, J., & Dostrovsky, J. (1971). The hippocampus as a spatial map: Preliminary evidence from unit activity in the freely-moving rat. *Brain Research*, *34*, 171–175.
- Salavati, A. H., Kumar, K. R., & Shokrollahi, A. (2014). Nonbinary associative memory with exponential pattern retrieval capacity and iterative learning. *IEEE Transactions on Neural Networks and Learning Systems*, *25*(3), 557–570.
- Salz, D. M., Tiganj, Z., Khasnabish, S., Kohley, A., Sheehan, D., Howard, M. W., & Eichenbaum, H. (2016). Time cells in hippocampal area CA3. *Journal of Neuroscience*, *36*(28), 7476–7484.
- Sreenivasan, S., & Fiete, I. (2011). Grid cells generate an analog error-correcting code for singularly precise neural computation. *Nature Neuroscience*, *14*(10), 1330.
- Stemmler, M., Mathis, A., & Herz, A. V. (2015). Connecting multiple spatial scales to decode the population activity of grid cells. *Science Advances*, *1*(11), e1500816.
- Stensola, H., Stensola, T., Solstad, T., Frøland, K., Moser, M.-B., & Moser, E. I. (2012). The entorhinal grid map is discretized. *Nature*, *492*(7427), 72–78.
- Strange, B. A., Witter, M. P., Lein, E. S., & Moser, E. I. (2014). Functional organization of the hippocampal longitudinal axis. *Nat. Rev. Neurosci.*, *15*(10), 655–669.
- Sun, C., Kitamura, T., Yamamoto, J., Martin, J., Pignatelli, M., Kitch, L. J., . . . Tonegawa, S. (2015). Distinct speed dependence of entorhinal island and ocean cells, including respective grid cells. *Proceedings of the National Academy of Sciences*, *112*(30), 9466–9471.
- Svoboda, K., & Yasuda, R. (2006). Principles of two-photon excitation microscopy and its applications to neuroscience. *Neuron*, *50*(6), 823–839.
- Taube, J. S., Muller, R. U., & Ranck, J. B. (1990). Head-direction cells recorded from the postsubiculum in freely moving rats. I. Description and quantitative analysis. *Journal of Neuroscience*, *10*(2), 420–435.
- Theis, L., Berens, P., Froudarakis, E., Reimer, J., Rosón, M. R., Baden, T., Euler, T., . . . Bethge, M. (2016). Benchmarking spike rate inference in population calcium imaging. *Neuron*, *90*(3), 471–482.
- Wennberg, D. (2015). *The distribution of spatial phases of grid cells*. Master's thesis, Norwegian University of Science and Technology.

- Xu, L., Krzyzak, A., & Oja, E. (1991). Neural nets for dual subspace pattern recognition method. *International Journal of Neural Systems*, 2(3), 169–184.
- Yartsev, M., & Ulanovsky, N. (2013). Representation of three-dimensional space in the hippocampus of flying bats. *Science*, 340, 367–372.
- Yartsev, M. M., Witter, M. P., & Ulanovsky, N. (2011). Grid cells without theta oscillations in the entorhinal cortex of bats. *Nature*, 479(7371), 103–107.

Received December 14, 2017; accepted April 2, 2019.