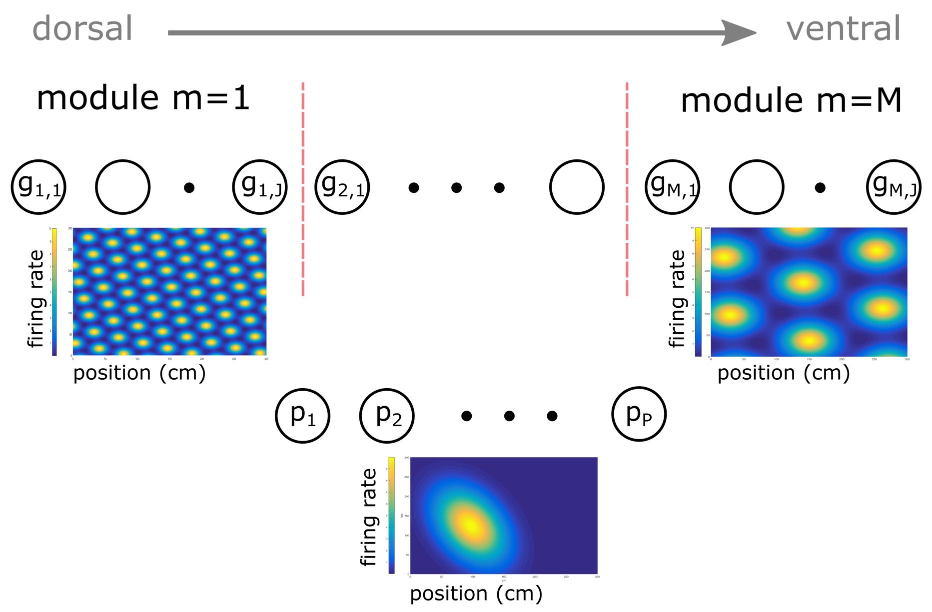
A hybrid code from grid and place cells

David M. Schwartz and O. Ozan Koyluoglu
Laboratory for Information Processing Systems
University of Arizona

Abstract

- The joint activity of grid and place cell populations forms a neural code for space.
- We measure the performance of a network of these populations, as well as interneurons, which implement biologically realizable de-noising algorithms.
- Simulations demonstrate that these de-noising mechanisms can significantly reduce mean squared error (MSE) of location decoding.
- The modular organization of grid cells can improve MSE.

The hybrid code



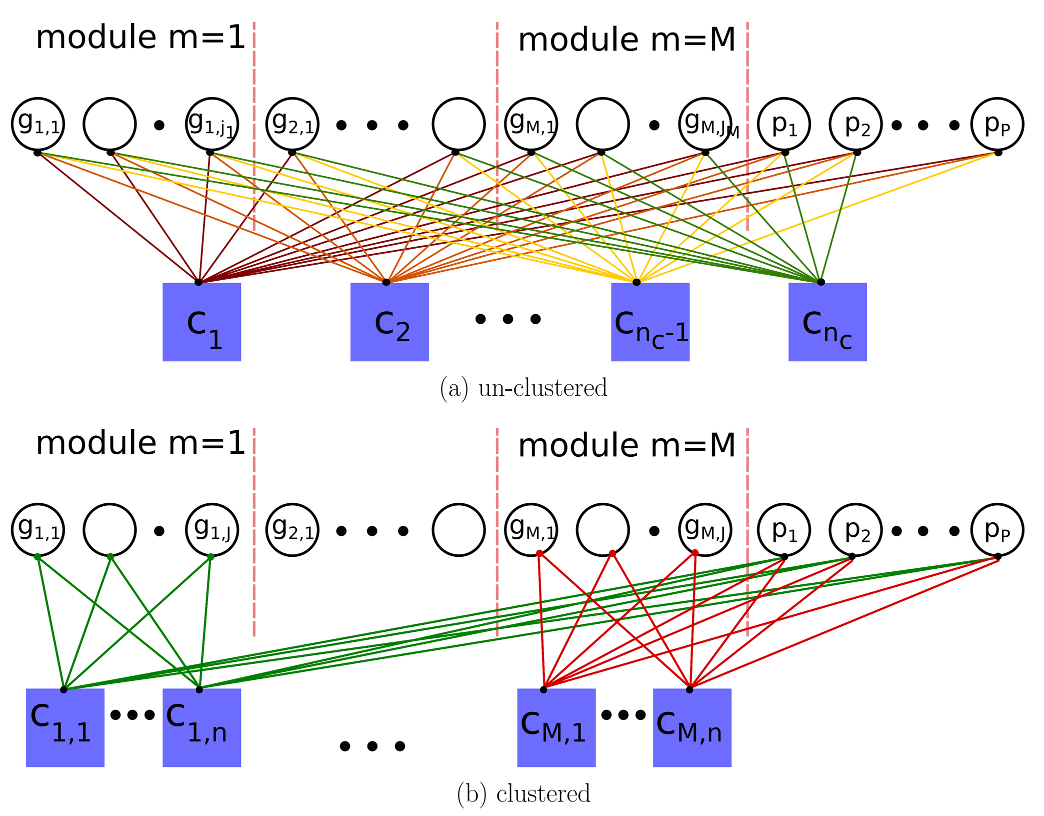
- **Components:** N neurons, M grid modules (m), with J neurons.
- **Grid cell tuning curves**

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

- $\mathbf{u}(\theta_k-\theta_{m,j})$ 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
- $\mathbf{c}_{m,j}$, $\theta_{m,j}$, and λ_m are spatial phase offset, orientation offset, and scaling ratio
- Orientations, $\theta_{m,j}\in\{-60°, 0°, 60°\}$
- Z is a normalizing constant (≈ 2.857399)
- $f_{\max}$ is the grid cell's maximum firing rate

- **Place cells** have bivariate Gaussian tuning curves with mean $\boldsymbol{\xi}\in[0,L]\times[0,L]$, $\rho\in[-\frac{1}{2},\frac{1}{2}]$, and covariance $\begin{pmatrix}\sigma_1^2 & \rho\sigma_1\sigma_2\\ \rho\sigma_1\sigma_2 & \sigma_2^2\end{pmatrix}$
- **Codewords** are formed by concatenating actities of these cells

De-noising network



(a) un-clustered

(b) clustered

- **This network** is a bipartite graph consisting of N pattern neurons and n_i interneurons
- **The un-clustered design:** Interneurons are connected to a random set of grid and place cells
- **The clustered design:**
  - Interneurons are split into M distinct clusters of n interneurons per cluster with each cluster connected to a distinct grid module.
  - Interneurons are connected randomly to pattern neurons chosen from a set consisting of every grid cell in the corresponding module, and every place cell.

Subspace learning

- **Before denoising** is possible, this network must learn (i.e. adapt its weights for) the hybrid code.
- **Code subspace learning** is complete when the interneurons may be read to determine if the states of the pattern neurons map to a valid codeword, i.e. when the network has developed a connectivity matrix, W, whose rows are approximately perpendicular to the code space.
- **(anti)Hebbian learning update rule:**

$$\mathbf{w}\leftarrow\mathbf{w}-\alpha_t\left(y\left(\mathbf{x}-\frac{y\mathbf{w}}{\|\mathbf{w}\|^2}\right)+\eta\Gamma(\mathbf{w},\theta)\right),$$

- α_t is the learning rate at iteration t
- $y=\mathbf{x}'\mathbf{w}$ is the scalar projection of $\mathbf{x}$ onto $\mathbf{w}$
- θ is a sparsity threshold
- η is a penalty coefficient
- Γ is a sparsity enforcing function, approximating the gradient of a penalty function, $g(\mathbf{w})=\sum_{k=1}^{m}\tanh(\sigma\mathbf{w}_k{}^2)$, which, for appropriate choices of σ, penalizes non-sparse solutions early in the learning procedure.

Algorithm 1 Neural Learning

```
Require: set of C patterns, 𝒞, stopping point, ϵ
Ensure: learned weights matrix, W
for rows, w, of W do
    for t ∈ {1, ..., T_max} do
        α_t ← max{ 50·α_0 / (50+log_10(t)), 0.005 }
        θ_t ← θ_0 / t
        for c ∈ 𝒞 do
            y ← c · w
            if ‖c‖ > ϵ then
                α_t ← α_0 / ‖c‖²
            end if
            w ← Dale(update(c, w, α_t, θ_t, η))
        end for
        if ‖C̲w′‖ < ϵ then
            break
        end if
        t ← t + 1
    end for
    for components, w_i of w do
        if |w_i| ≤ ϵ then
            w_i ← 0
        end if
    end for
end for
```

De-noising algorithms

- **Goal:** Recover the correct pattern of activity, $\mathbf{x}$ from the noisy state, $\mathbf{x}_n=\mathbf{x}+\mathbf{n}$, where $\mathbf{n}$ is this noise pattern.
- **$\mathbf{x}_nW'$ reveals inconsistencies in $\mathbf{x}_n$** that the de-noising algorithm seeks to correct in the feedback stage. To see this, consider that $\mathbf{x}_nW'=(\mathbf{x}+\mathbf{n})W'=\mathbf{x}W'+\mathbf{n}W'\approx 0+\mathbf{n}W'$.
- **Clustered de-noising** begins with Algorithm 2. Algorithm 3 is invoked if errors are detected.
- **Un-clustered de-noising** utilizes Algorithm 3, treating the entire network as a single cluster.

Algorithm 2 Sequential de-noising

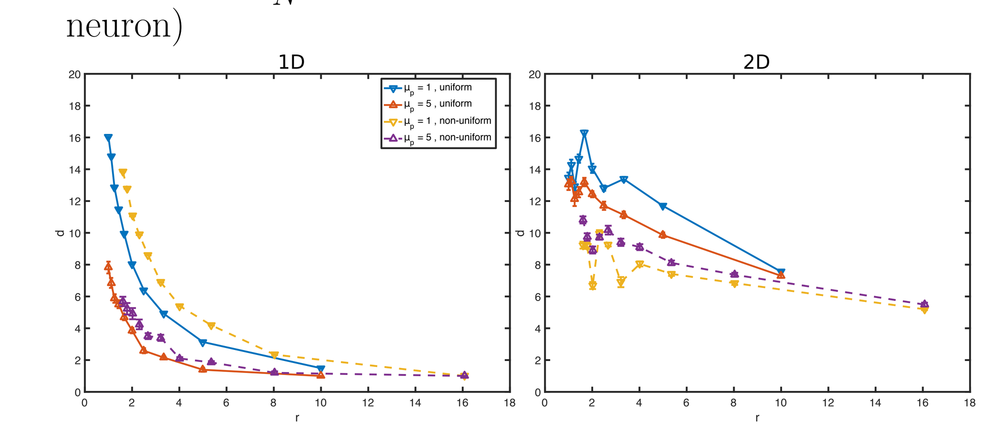
```
Require: local weights, W_i, for each cluster, i ∈ {1, ..., M}, noisy pattern, x_n, stopping threshold, ϵ
Ensure: denoised pattern, x_d
x_d ← x_n
while t < T_max or a cluster has an unsatisfied constraint do
    for each cluster, i ∈ {1, ..., M} do
        x ← subpattern corresponding to cluster i
        d ← Modular_Recall(x, W_i)
        if |dW_i| ≤ ϵ then
            x_d(cluster i's subpattern indices) ← d
        end if
    end for
    t ← t + 1
end while
```

Algorithm 3 Modular Recall

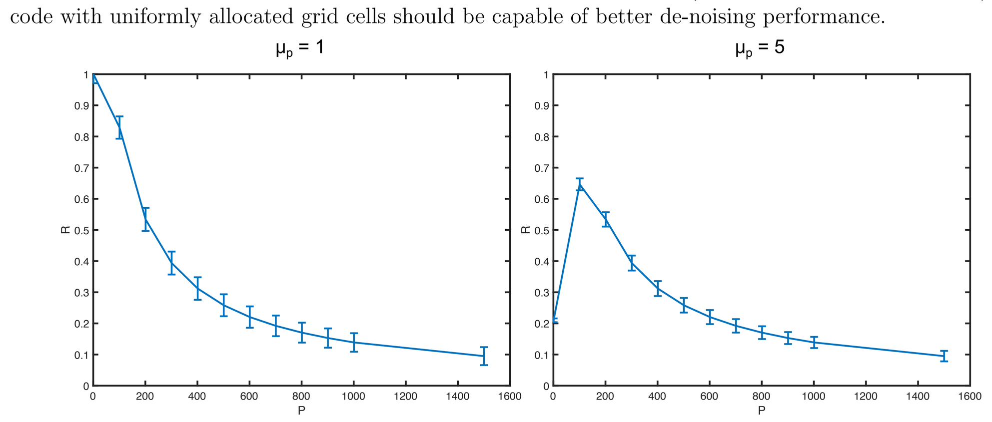
```
Require: local weights for this cluster, W, maximum number of iterations, T_max, noisy subpattern, x, feedback threshold, ϕ
Ensure: denoised subpattern, d
d ← p
while t < T_max do
    y ← xW′
    r ← y′W
    if ‖y‖ < ϵ then
        break;
    end if
    f ← (|y′|·|W|) / (Σ_{i=1}^{m} |W|)
    for each pattern neuron, j do
        if f_j ≥ ϕ then f_j = sign(x_j)
        else f_j = 0
        end if
    end for
    d ← d + f
end while
```

Coding theoretic results

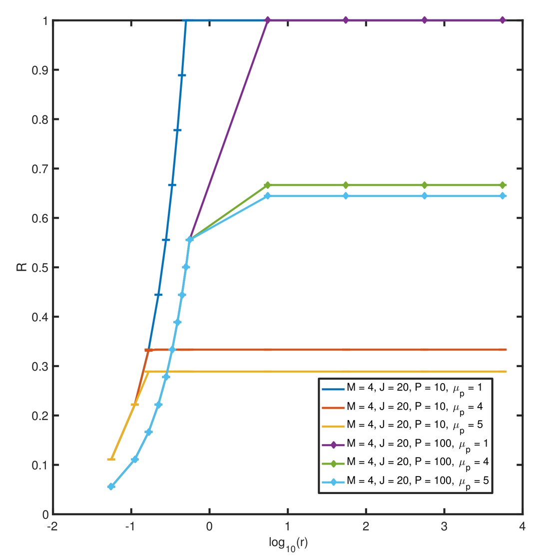
- **Define** μ_p, a hybrid code configuration's spatial phase multiplicity (i.e. maximum number of grid cells with the same phase in the same module)
- **Define** $\mu_o=\frac{J}{3}$, the code's orientation multiplicity
- **Define** d, minimum pairwise distance between codewords
- **Define** $R=\frac{\text{rank}(\underline{\mathbb{C}})}{N}$, normalized rank of the code
- **Define** $r=\frac{C}{N}$, code rate (number of locations represented per neuron)



A steep tradeoff between d and r is shown in 1 and 2 dimensions. In 2D, the hybrid code generates a better d for large code r in all configurations. Further, in 2D, the code with non-uniformly allocated grid cells has significantly smaller d for a fixed r. Thus, in 2D, for a fixed r (i.e. for codes of the same rate), the code with uniformly allocated grid cells should be capable of better de-noising performance.



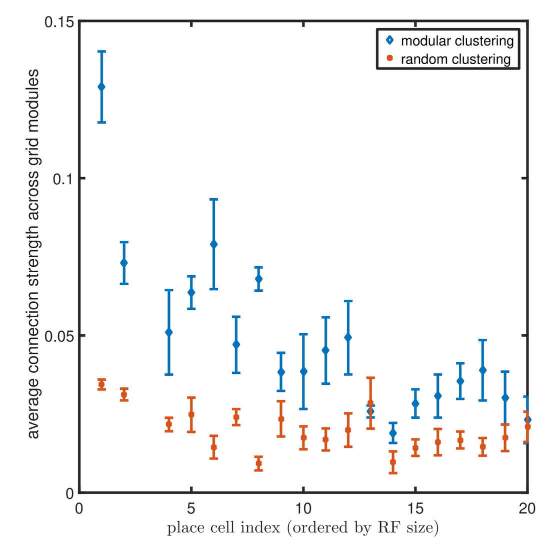
Code rank (R) vs. number of place cells (P) for the hybrid code in 2D, with a uniform allocation of grid cells; here (and in any other plot containing them) error bars indicate standard error of the mean



- **Random phases** often produces a code with $R=1$, independent of how grid cells are distributed to modules.
- **Choosing** $\mu_p>1$ enables the code to achieve low rank at high rate (important for de-noising a code with a large number of locations)

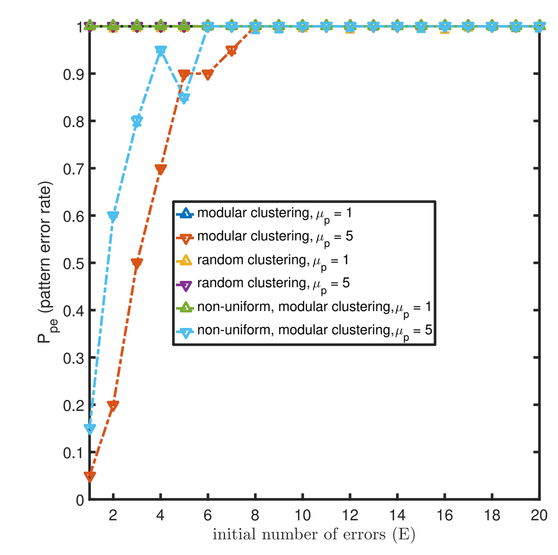
Subspace learning results

- **Define** connection strength from place cells to grid modules by $\frac{1}{n_i}(\sum_{(i,j)}|w_{i,j}w_{i,p}|)$ (for interneurons, i, grid cells, j, in module m).

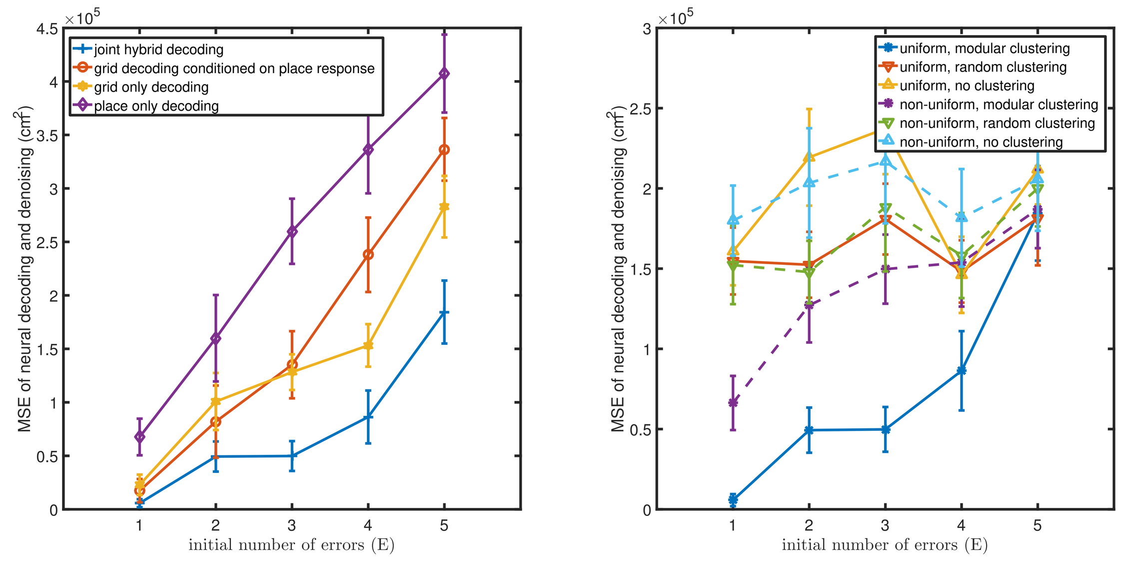


- **Average connectivity** appears to decrease with increasing place cell size, for the modularly clustered network, as compared to a random clustering which produces nearly the same connectivity for each place cell. This trend appears for any $\mu_p>1$.

De-noising results



- **Pattern error rate** (rate of occurrence of incorrect codewords after de-noising); only clustered configurations with $\mu_p>1$ perform well here



MSE (in cm^2) of maximum likelihood position estimation from de-noised codewords; (left) comparison of decoding algorithms incorporating different cellular information; (right) comparison of MSE of joint hybrid decoding after de-noising for different de-noising network configurations; (both) for a hybrid code with $M=4$, $J=20$, $P=10$, and $\mu_p=5$, and deliberately chosen grid cell parameters

Discussion

- **The grid code** is dense.
- **Inclusion of place cells** and in the future, other cell types (e.g. head direction cells, border cells, time cells) - this code could be made sparser.
- **Codes** with any desired rank can be constructed by proper choice of population parameters.
- **Random choices** of these parameters render the code too dense for effective de-noising.
- **Biological choices** of orientation and phase produce readily de-noisable codes for position.

References

[1] Stensola, Hanne, Tor Stensola, Trygve Solstad, Kristian Frland, May-Britt Moser, and Edvard I. Moser. "The Entorhinal Grid Map Is Discretized." Nature 492, no. 7427 (2012): 72-78. doi:10.1038/nature11649.
[2] Salavati, Amir Hesam, K. Raj Kumar, and Amin Shokrollahi. "Nonbinary Associative Memory With Exponential Pattern Retrieval Capacity and Iterative Learning." IEEE Transactions on Neural Networks and Learning Systems 25, no. 3 (2014): 557-70. doi:10.1109/tnnls.2013.2277608.
[3] Oja, Erkki, and Juha Karhunen. "On Stochastic Approximation of the Eigenvectors and Eigenvalues of the Expectation of a Random Matrix." Journal of Mathematical Analysis and Applications 106, no. 1 (1985): 69-84.
[4] Karbasi, Amin, Amir Hesam Salavati, and Amin Shokrollahi. "Iterative Learning and Denoising in Convolutional Neural Associative Memories." In ICML (1), pp. 445-453. 2013.
[5] Fiete, Ila R., Yoram Burak, and Ted Brookings. "What grid cells convey about rat location." The Journal of Neuroscience 28, no. 27 (2008): 6858-6871.
[6] Wenberg, Daniel, and Yasser Roudi. The Distribution of Spatial Phases of Grid Cells. Master's thesis, NTNU, 2015. 1-89. Accessed November 7, 2016. http://hdl.handle.net/11250/2359809.