**KEY RESOURCES TABLE**

Code for training CRF models can be found at <https://github.com/kuitang/fwmatch-public>.

**CONTACT FOR REAGENT AND RESOURCE SHARING**

Further information and requests for resources and reagents should be directed to and will be fulfilled by the Lead Contact, Luis Carrillo-Reid ([lc2998@columbia.edu](mailto:lc2998@columbia.edu)).

**EXPERIMENTAL MODELS AND SUBJECT DETAILS**

**METHOD DETAILS**

**Spike detection**

Calcium traces were first processed by subtracting the baseline to get dF/F. Then, dF/F values of the top and bottom 30% were removed to obtain a baseline activity, and a threshold was defined by the mean of the baseline plus 5 times standard deviation of baseline values. Spikes were detected as time points where dF/F values are higher than the threshold.

**Conditional Random Fields**

We construct a conditional random field (CRF) as previously published ([Tang et al., 2016](#_ENREF_45)), using feature vector where , and the target binary population activity vector , where , for samples (time points). For each sample, the conditional probability can be expressed as:

where is a vector of sufficient statistics of the distribution, is a vector of parameters, and is the partition function:

The conditional probability can be factored over a graph structure , where is the collection of nodes representing observation variables and target variables, and is the collection of subsets of . The conditional dependencies can be then written as

This model is a generalized version of Ising models, which have been previously applied to model neuronal networks ([Yu et al., 2008](#_ENREF_50)). The log-likelihood of each observation can be then written as:

Given the inferred binary spikes from raw imaging data, we construct a CRF model by two steps: (1) structure learning, and (2) parameter learning. For structure learning, we learned a graph structure using -regularized neighborhood-based logistic regression ([Ravikumar et al., 2010](#_ENREF_35)):

,

where

.

Here is a regularization parameter that controls the density of constructed structure. Then, a graph structure is learnt by thresholding the edge potentials with a given density preference . Edges with potential values within top quantile were kept as the final structure. It is worth noting that although could bias the result, varying does not lead to density values that differ much. This is probably because of the sparse nature of the obtained Ising model.

Based on the learned structure, we use the Bethe approximation to approximate the partition function, and iterative Frank-Wolfe methods to perform parameter estimation by maximizing the log-likelihood of the observations with a quadratic regularizer ([Tang et al., 2016](#_ENREF_45)):

Here is a regularization that controls the learnt parameters. Cross-validation was done to find the best , and via model likelihood. We varied with 6 values between 0.002 and 0.5, d with 6 values between 0.25 and 0.3, and with 5 values between 10 and 10000, all sampled uniformly. To obtain the best model parameters, 90% data were used for training, while 10% data were withheld for cross-validation. The best model parameters were determined by calculating the likelihood of the withheld data and selecting the parameter set with a locally maximum likelihood in the parameter space.

**Identifying optimal cortical ensembles**

To find optimal cortical ensembles for each condition, we iterate through all the neurons and identify their contribution in predicting stimulus conditions with the population. To this end, for the neuron in population, we set its activity to be ‘1’ and ‘0’ in turn, in all M frames. With the two resulting population vectors in the frame among all samples, we calculate the likelihood of them coming from the trained CRF model:

Then, we computed the likelihood difference vector

and threshold it by baseline plus 3 times the standard deviation of noise. This gives a binary prediction vector for each neuron over the entire experiment. We then calculated the true positive rate (TPR) and false positive rate (FPR) of each neuron using each stimulus as following:

where represent the true positives, false positives, true negatives and false negatives of neuron predicting stimulus .

The optimal cortical ensembles for each visual stimulus were defined by varying the threshold in the TPR/FPR space and cross-validating the threshold using the accuracy of cosine similarity prediction for the corresponding stimuli as readout.

**Prediction with cosine similarity**

Identified optimal cortical ensembles were represented by a binary vector over all neurons, and the entries corresponding to the ensemble members were set to ‘1’, while the rest were set to ‘0’. Cosine similarities between ensemble vectors and frame activity vectors were calculated, and a threshold was determined by baseline plus 3 times the standard deviation of noise. The cosine similarity between two frame activity vectors depicts the angle between population vectors in a high-dimensional space.

**High-ranked cells**

We define the node strength as the sum of the ‘11’ term of edge potentials from all connecting edges:

Here denotes the number of connecting edges for node . The defined rank reflects the importance of a given cell in co-activating with other cells.

**Graph properties**

Given the adjacency matrix where if node is linked to node , we investigated the following graph properties: graph density, node degrees, local clustering coefficients, and eigenvector centrality. Graph density is calculated as the number of existing edges divided by the number of total possible edges:

where *NV* is the number of vertices in the graph. Node degree is defined for node as the number of edges connected to it:

.

Local clustering coefficient is defined for each node as the fraction edges connected to it over the total number of possible edges between the node's neighbors (nodes that have a direct connection with it). Eigenvector centrality is defined on the relative centrality score matrix , where

This can be written in the form of eigenvector equation:

Solving the above equation gives a set of eigenvalues and associated eigenvectors. The entry of the eigenvector associated with the largest gives the eigenvector centrality for the node .

**QUANTIFICATION AND STATISTICAL ANALYSIS**

CRF models were trained using the Columbia Yeti shared HPC cluster. MATLAB R2016a (MathWorks) were used for data analysis. Statistical details of each specific experiments, including the statistical methods, the meaning and value of n, and the significance level can be found in figure legends.

**DATA AND SOFTWARE AVAILABILITY**

The analysis of simultaneous two-photon imaging and two-photon optogenetic stimulation data was performed from experiments previously published ([Carrillo-Reid et al., 2016](#_ENREF_10)).

We also used a publicly available dataset from the Allen Brain Observatory (<http://observatory.brain-map.org/visualcoding>) along with the SDK for extracting related information (<http://alleninstitute.github.io/AllenSDK/>) by Allen Institute of Brain Science. The experiments IDs are: 511507650, 511509529, 511510650, 511510670, 511510718, and 511510855.

All data processing, analysis and plotting code in this paper can be found at <https://github.com/hanshuting/graph_ensemble>.