Reading-out task variables as a low-dimensional reconstruction of neural spike trains in single trials.

Veronika Koren[1,2], Ariana R. Andrei[3], Ming Hu[4], Valentin Dragoi[3], Klaus Obermayer[1,2]

1 Neural Information Processing Group, Institute of Software Engineering and Theoretical Computer Science, Technische Universität Berlin, Berlin, 10587, Germany **2** Bernstein Center for Computational Neuroscience Berlin, Germany **3** Department of Neurobiology and Anatomy, University of Texas Medical School, Houston, Texas, 77030, US **4** Picower Institute for Learning and Memory, Massachusetts Institute of Technology, Cambridge, Massachusetts, 02139, US

Correspondence should be addressed to: koren@ni.tu-berlin.de

Abstract

We propose a novel method of the read-out of spike trains that exploits the structure of responses of neural ensembles. Assuming the point of view of a read-out neuron that receives synaptic inputs from a population of projecting neurons, synaptic inputs are weighted with a heterogeneous set of weights. We propose that synaptic weights reflect the role of each neuron within the population for the computational task that the network has to solve. In our case, the computational task is discrimination of binary classes of stimuli, and weights are such as to maximize the discrimination capacity of the network. We compute synaptic weights as the feature weights of an optimal linear classifier. Once weights have been learned, they weight spike trains in real time and allow to compute the post-synaptic current that modulates the spiking probability of the read-out unit. We apply the method on parallel spike trains from V1 and V4 areas in the behaving monkey *macaca mulatta*, while the animal is engaged in a visual discrimination task with binary classes of stimuli. The read-out of spike trains with our method allows to discriminate the two classes of stimuli, while simpler methods as the population PSTH entirely fail to do so. Splitting neurons in two subpopulations according to the sign of the weight, we show that population signals of the two functional subnetworks are negatively correlated. Disentangling the superficial, the middle and the deep layer of the cortex, we show that in both V1 and V4, superficial layers are the best in discriminating binary classes of stimuli.

Keywords: Visual cortex, spike trains, coding, stimulus, choice, dimensionality reduction, computation, neural network

Introduction

A half century ago, pioneers of neuroscience have stated the following: “At present we have no direct evidence on how the cortex transforms the incoming visual information. Ideally, one should determine the properties of a cortical cell, and then examine one by one the receptive fields of all the afferents projecting upon that cell. ” (Hubel and Wiesel, 1962, Journal of Physiology, [1]). While lots of insights in the computations in cortical circuits have been made in the meantime, the question, posed by Hubel and Wiesel, has not yet found a clear answer [2]. Addressing this question requires observing the activity of many neurons simultaneously and has demanded an important progress in recording techniques. Besides the advances on the experimental side, major challenges consist also in interpreting rich datasets and understanding the underlying principles of cortical computation [3]. One of the biggest conceptual gaps to bridge is between sensory processing and animal’s behavior, addressed by decision-making studies [4, 5, 6, 7]. Linking behavioral choices with the neural activity in the sensory areas requires the understanding of the transformation between sensory and decision-related signals. On the one hand, spiking patterns of neural populations in sensory areas are highly variable across trials [8], and can be described as a probabilistic process. The choice behavior of animal agents, on the other hand, appears to be highly precise and coherent with respect to the incoming (natural) stimuli. Even though the behavior is noisy [9], and prone to errors due to wrong internal representation [10], this might be so because it can adapt to perturbations and the uncertainty in the environment [11]. Our main question here is how does the brain transform a high-dimensional probabilistic signal, enacted by spike trains of cortical populations, into a reliable signal, that presumably underlies coherent animal behavior.

Recent theoretical and modeling work has shown that it is possible to read-out a deterministic population signal from variable spike trains of a spiking neural network [12]. In [12], it is assumed that the population signal is encoded at the network level and cannot be accounted for by the observation of single neurons. The population signal is distributed among single neurons in a non-linear fashion, giving rise to a distributed code. The distributed code maps from the high-dimensional space of spike trains of many neurons to the low-dimensional space of the population signal. If the coding function of individual neurons is redundant within the network, a single population signal can give rise to many different spiking patters. Vice-versa, variable spiking patterns can be read-out as a deterministic population signal. This coding scheme therefore allows to reconcile variable spike trains with a deterministic signal, that might underlie animal’s behavior. While efforts have been made to design an efficient network that is biologically plausible [14], no convincing evidence for such a computation in biological ensembles has been presented so far.

In the present study, we apply theoretical propositions of the model with the distributed code [12, 15] to experimental data. Our goal is to connect the theory on representation of an abstract and arbitrary population signal to behaviorally relevant variables in the biological brain. While the idea of weighting spikes has been put forward by the aforementioned studies [12, 13, 15], these studies utilize random weighs. Here, we propose that weights are not random but depend on the computational task at hand, in our case, discrimination of binary stimulus classes. Decoding here consists in transforming spike trains of simultaneously recorded neurons, a high-dimensional variable, into a low-dimensional population signal. The core of the transformation is to weight the spike trains of individual neurons by their coding function and then sum across neurons, giving a low-dimensional representation of network's spiking activity. In the biological setting, such a low-dimensional signal corresponds to the synaptic current, received by a read-out neuron, and modulating its probability of spiking. In short, this study attempts to bridge the gap between abstract models of computation in neural networks and activity of neural populations, recorded *in vivo* in the visual cortex. It addresses the sensory features in visual areas V1 and V4 that pertain to correct choice behavior and tries to bring insights about "...how do the connectivity and dynamics of distributed neural circuits give rise to specific behaviors and computations" (Gao & Ganguli, 2015, *Curr. Op. in Neurobiology*, [3]).

Materials and Methods

Ethics statement All experiments were conducted in accordance with protocols approved by The Animal Welfare Committee (AWC) and the Institutional Animal Care and Use Committee (IACUC) for McGovern Medical School at The University of Texas Health Science Center at Houston (UTHealth), and met or exceeded the standards proposed by the National Institutes of Healths Guide for the Care and Use of Laboratory Animals.

Animal subjects Two male rhesus macaques (Macaca mulatta; M1, 7 years old, 15kg; M2, 11 years old, 13kg) were previously trained to perform visual discrimination task, and each implanted with a titanium head post device and two 19mm recording chambers (Crist Instruments) over V1 and V4. All surgeries were performed aseptically, under general anesthesia maintained and monitored by the veterinary staff from the Center for Laboratory Animal Medicine and Care (CLAMC), with appropriate analgesics as directed by the specialized non-human primate veterinarian at CLAMC. During the study the animals had unrestricted access to fluid, except on days when behavioral tasks were performed. These days, animals had unlimited access to fluid during the behavioral task, receiving fluid for each correctly completed trial. Following the behavioral task, animals were returned to their home cage and were given additional access to fluid. During the study, the animals health and welfare was monitored daily by the veterinarians and the animal facility staff at CLAMC and the labs scientists, all specialized with working with non-human primates.

Experimental setup

Animals performed a visual, delayed-match-to-sample task. The trial started after 300 ms of successful fixation within the fixation area consisted in displaying the target and the test stimuli, naturalistic images in black and white, with a delay period in between. The target and the test stimuli were either identical (condition "match") or else the test stimulus was rotated with respect to the target stimulus (condition "non-match"). The target and the test stimuli were shown for 300 ms each while the delay period had a random duration between 800 and 1000 ms. The task of the animal was to decide about the similarity of the target and the test stimuli by holding a bar for "different" and releasing the bar for "same". The subject was required to respond within

200 and 1200 ms from the time the test stimulus was off, otherwise the trial was discarded. The difference in orientation of the test stimulus ranged between 3 and 10 degrees and was calibrated on-line in order to have on average 70 percent correct responses on non-matching stimuli.

In every recording session, laminar electrode with 16 recording channels each were inserted in V1 and V4 areas. In part of sessions, recordings were made in V1 and V4 simultaneously, with one laminar electrode in each area, while in other sessions, only V1 has been recorded. The position of the electrode was calibrated in such a way that neurons from the two areas had overlapping receptive fields. The multi-unit signal and the local field potential were recorded in 20 recording sessions in V1 and in 10 recording sessions in V4. We analyzed the activity of cells that responded to the stimulus by increasing their firing rate at least four-fold with respect to their baseline. We used the activity of all neurons that obeyed that criterion, which gave 160 neurons in V1 and 102 neurons in V4. The number of trials was roughly balanced across conditions. In V1, 107 trials per recording session were collected, on average, in condition “non-match”, and 118 trials in condition “match”. In V4, there were, on average, 99 trials per session in condition “non-match” and 76 in condition “match”.

The spike train and the population PSTH

The following analyses were done with Matlab, Mathworks, version R2017b.

The spike train of a single neuron n in trial j is a vector of zeros and ones,

$$o_{n,j}(t_k) = \begin{cases} 1, & \text{if neuron } n \text{ in trial } j \text{ spikes during the } k\text{-th millisecond} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where $n = 1, ..., N$ is the neural index, $j = 1, .., J$ is the trial index and $k = 1, ..., K$ is the time index with step of 1 millisecond. The population PSTH is computed by averaging spike trains across neurons and across trials,

$$PSTH(t_k) = \frac{1}{N}\frac{1}{J}\sum_{n=1}^{N}\sum_{j=1}^{J} o_{n,j}(t_k). \tag{2}$$

The PSTH is convolved with a Gaussian kernel, $w(\tau) = \left(\sum_{\tau \in T} \exp(-\frac{\tau^2}{2\sigma_w^2})\right)^{-1} \exp(-\frac{\tau^2}{2\sigma_w^2})$, with variance $\sigma_w^2 = 10\ ms$ and support $\mathrm{T} = \{-10, ..., 10\}$.

$$PSTH^{conv}(t) = \sum_{\tau \in T} PSTH(t-\tau)w(\tau) \tag{3}$$

Estimation of decoding weights.

Trials were split into training and validation set. We use the training set to compute decoding weights and the validations set to apply weights to spike trains and compute the population signal. The training and the validation set are non-overlapping and utilize half of the available trials each. The split into training and validation set is cross-validated with the Monte Carlo method. In every cross-validation run, the data is randomly split into training and validation set. Throughout the paper, we used 100 cross-validations and the reported results are averages across cross-validations.

Constructing features. Decoding weights were computed as the feature weights of the linear Support Vector Machine (SVM, [16]). The SVM utilizes spike count statistics. The spike count of the neuron n in trial j, $s_{n,j} = \sum_{k=1}^{K} o_{n,j}(t_k)$, is computed in target and test time windows, corresponding to the interval of $[0, 400]$ milliseconds with respect to the onset of the target and the test stimuli. In order to correctly associate the change in the firing rate of a neuron with the sign of its decoding weight, spike counts are $z-$scored, for each neuron independently,

$$\tilde{s}_{n,j} = \frac{s_{n,j} - \langle s_{n,j} \rangle_j}{\sqrt{\mathrm{Var}_j \ (s_{n,j})}}, \tag{4}$$

where $\langle s_{n,j} \rangle_j$ is the empirical mean and $\mathrm{Var}_j \ (s_{n,j})$ is the empirical variance across trials. One sample for the classifier is an N-dimensional vector of activities, $\tilde{\mathbf{s}}_j = [\tilde{s}_{1,j}, \tilde{s}_{2,j}, ..., \tilde{s}_{N,j}]$.

Model fitting and extraction of weights. The optimization problem that the linear SVM solves is, in its primal form, expressed with a Lagrangian,

$$L_p = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{j=1}^{J} \lambda_j [y_j(\mathbf{w}^T\tilde{\mathbf{s}}_j + b) - 1], \tag{5}$$

with $\mathbf{w}$ the $(N, 1)$-dimensional vector of weights, b the offset of the separating hyperplane from the origin, λ_j the Lagrange multiplier, with $\lambda_j \geq 0 \ \forall j$, $\tilde{\mathbf{s}}_j$ the $(N, 1)$-dimensional input, and y_j the class label in trial j, with $y_j \in \{-1, 1\}$. The minimization of the Lagrangian will result in Lagrange multipliers equal to zero for as many samples as possible, and only a subset of samples, those that lie on the margin, will be used to determine the separation boundary. Those samples are called the support vectors: $\mathbf{v}_q = [v_{1,q}, v_{2,q}, ..., v_{N,q}]$, $q = 1, ..., Q$, $v_{n,q} \in \mathbb{R}$. The separating boundary is the linear combination of the support vectors, $H_0 : \mathbf{w}^T\mathbf{v}_q + b = 0$. Minimizing the Lagrangian, i.e., differentiating the Lagrangian with respect to the weight vector $\mathbf{w}$ and setting the derivative to zero, one obtains the expression for the weight vector,

$$\mathbf{w} = \sum_{q=1}^{Q} \lambda_q y_q \mathbf{v}_q. \tag{6}$$

The separation boundary is fully described by the weight vector, that determines its direction (i.e., the weight vector is perpendicular to the separation boundary) and the offset from the origin b. We normalize the weight vector with the L^2 norm,

$$\tilde{\mathbf{w}} = \frac{\mathbf{w}}{||\mathbf{w}||}, \tag{7}$$

with $||\mathbf{w}|| = \sqrt{w_1^2 + ... + w_N^2}$. The normalized weight vector, $\tilde{\mathbf{w}} = [\tilde{w}_1, ..., , \tilde{w}_N]$, is computed for every recording session (i.e., for simultaneously recorded neurons), and associates the activity of each neuron within the population with neuron's role for classification. Note that weights are computed from the firing rate statistics, but are then applied to spike trains.

In general, it is not necessarily possible to linearly separate all the input samples, not even in the training data. The classification model of the SVM optimizes the separating hyperplane also with respect to the data points that lie on the wrong side of the margin (slack points). The regularization parameter determines how much do slack points contribute to the error that the model is minimizing. If slack points contribute strongly to the error, the model converges to a narrow margin, which might result in bad generalization of the hyperplane to the test data (overfitting). Here, we chose the regularization parameter with 5-fold cross-validation on the training set. The training set was split into 5 folds, the classifier was trained on 4 folds and validated on the remaining fold with balanced accuracy:

$$BAC = \frac{1}{2}\frac{TP}{TP+FN} + \frac{1}{2}\frac{TN}{TN+FP}, \tag{8}$$

where TP, TN, FP and FN stand for the number of test samples that have been classified as the true positive, the true negative, the false positive and the false negative, respectively. We call the true positive a correct classification of condition “match”, associated with the label $y_j = 1$, and a true negative a correct classification of condition “non-match”, associated with the label $y_j = -1$. The performance measure of the balanced accuracy is used since it accounts for imbalanced classes, that is, different number of trials in conditions “match” and “non-match”. When the 5 combinations of training/validation folds are exhausted, we compute the average balanced accuracy across folds. Iterating this procedure for a range of regularization parameters,
$C \in \{0.0012, 0.0015, 0.002, 0.005, 0.01, 0.05, 0.1, 0.5\}$, we chose the regularization parameter that maximized the balanced accuracy. The regularization parameter is fitted to every recording session independently. The range of the weight vector (6) depends on the regularization parameter C, and is therefore different across recording sessions. Since we combine results from many recording sessions, normalization in (7) is necessary to keep decoding weights in the same range across recording sessions.

Low-dimensional population signal

The population signal as the weighted sum of spikes of simultaneously recorded neurons. Imagine a population of N neurons that project to a read-out neuron. Every spike of a projecting neuron creates a small jump in the membrane potential of the read-out neuron, followed by a decay towards the baseline [17]. Moreover, spikes of all projecting neurons are summed up in the membrane potential of the read-out neuron [17]. Consider the spike train of N simultaneously recorded neurons in trial j and in time step t_k,

$$\mathbf{o}_j(t_k) = o_{1,j}(t_k), o_{2,j}(t_k), ..., o_{N,j}(t_k), \tag{9}$$

where $j = 1, 2, ..., J', J'+1, ..., J$ are trials. The trials in condition “non-match”, $j = 1, 2, ..., J'$, are followed by trials in condition “match”, $j = J'+1, J'+2, ..., J$. The transformation of spike trains into a low-dimensional population signal consists in multiplying the spike train of each neuron with the corresponding weight, summing across neurons and convolving with an exponential kernel. Equivalently, this can be written as the projection of spike trains on the vector of decoding weights,

$$x_j(t_k) = F(\tilde{\mathbf{w}}^T \mathbf{o}_j(t_k)), \tag{10}$$

where $F(y)$ is the transfer function, consisting of the convolution with an exponential kernel, $u(t) = exp(-\lambda t)$,

$$F(y) = \sum_{t \in T} F(y-t)u(t). \tag{11}$$

Note that the transformation applies to a specific trial and maintains the temporal dimension of the spike train. The only manipulation is to reduce the dimensionality from N (dimensionality of the spike train) to 1 (dimensionality of the population signal). We compute the deviation of the resulting signal from the mean,

$$\tilde{x}_j(t_k) = x_j(t_k) - z(t_k), \tag{12}$$

where $z(t_k)$ is the average population signal across trials from both conditions,

$$z(t_k) = \frac{1}{J} \sum_{j=1}^{J} x_j(t_k). \tag{13}$$

The low-dimensional signal is then averaged across trials, distinguishing trials from condition “match” and “non-match”.

$$\begin{aligned} \tilde{x}^{nm}(t_k) &= \frac{1}{J'} \sum_{j=1}^{J'} \tilde{x}_j(t_k) \\ \tilde{x}^{m}(t_k) &= \frac{1}{J-J'} \sum_{j=J'+1}^{J} \tilde{x}_j(t_k) \end{aligned} \tag{14}$$

The significance of the discrimination between the population signal in “match” and “non-match” is evaluated with he permutation test. The test statistics is the difference of population signals, $x^{diff}(t_k) = \tilde{x}^m(t_k) - \tilde{x}^{nm}(t_k)$. We compare $x^{diff}(t_k)$ with $x_p^{diff}(t_k)$, where the latter has been computed with random permutation of class labels “match” and “non-match. First, decoding weights are computed with SVMs that are trained on randomly permuted class labels. Second, those weights are applied to spike trains where class labels for “match” and “non-match” have been randomly permuted. The whole procedure is repeated $nperm = 1000$ times and gives a distribution of results for each time step . When the result of the true model appears outside of the distribution of results for the null model, the difference of signals in conditions “match” and “non-match is considered to be significant.

The population signal of neurons with positive and negative weights. We separate the population of simultaneously recorded neurons with respect to the sign of the decoding weight. In each recording session, we split neurons into the subpopulation of neurons with positive weight and negative weight (in the following, plus and minus neurons). We define the weight vector of plus neurons, $\tilde{\mathbf{w}}^+$, and minus neurons, $\tilde{\mathbf{w}}^-$, by replacing weights of the opposite sign with zero. Note that spikes of neurons with the opposite sign are weighted by a zero weight and therefore do not contribute to the projection. The population signal is computed with each of the two weight vectors.

$$\begin{aligned} x_j^+(t_k) &= F((\tilde{\mathbf{w}}^+)^T \mathbf{o}_j(t_k)) \\ x_j^-(t_k) &= F((\tilde{\mathbf{w}}^-)^T \mathbf{o}_j(t_k)) \end{aligned} \tag{15}$$

The number of plus and minus neurons is not balanced. During the target time window, the proportion of plus neurons is, on average, 0.58 in V1 and 0.54 in V4. During the test time window, this proportion is 0.54 in V1 and 0.62 in V4. To be able to compare results for the two neuronal types, we scale the weight vector with the correction factor, $\tilde{\mathbf{w}}^+ c^+$, $(\tilde{\mathbf{w}}^- c^-)$. The correction factor c is computed as as the ratio of $1/2$ with the proportion of neurons with positive (negative) weight,

$$\begin{aligned} c^+ &= {}^{N}\!/_{2N^+} \\ c^- &= {}^{N}\!/_{2N^-} \end{aligned} \tag{16}$$

where N^+ (N^-) is the number of plus (minus) neurons. For example, if there are more plus than minus neurons in a session, weights of plus neurons are multiplied with correction factor that is smaller than one and weights of minus neurons with correction factor that is bigger than one.

Same as before, we compute the deviation of the population signal from the mean,

$$\begin{aligned} \tilde{x}_j^+(t_k) &= x_j^+(t_k) - z^+(t_k) \\ \tilde{x}_j^-(t_k) &= x_j^-(t_k) - z^-(t_k) \end{aligned} \tag{17}$$

with $z^+(t_k)$ and $z^-(t_k)$ the sign-specific average of population signals.

$$\begin{aligned} z^+(t_k) &= \frac{1}{J}\sum_{j=1}^{J} x_j^+(t_k) \\ z^-(t_k) &= \frac{1}{J}\sum_{j=1}^{J} x_j^-(t_k) \end{aligned} \tag{18}$$

Finally, we average each of the signals across trials, distinguishing conditions “match” and “non-match”. For plus neurons this gives the following:

$$\begin{aligned} \tilde{x}^{+,nm}(t_k) &= \frac{1}{J'}\sum_{j=1}^{J'} \tilde{x}_j^+(t_k) \\ \tilde{x}^{+,m}(t_k) &= \frac{1}{J-J'}\sum_{j=J'+1}^{J} \tilde{x}_j^+(t_k) \end{aligned} \tag{19}$$

As before, we evaluate the significance of results with the permutation test. The test statistic is the sign-specific difference of signals in conditions “match” and “non-match”, e.g., for plus neurons: $x^{diff,+}(t_k) = \tilde{x}^{+,m}(t_k) - \tilde{x}^{+,nm}(t_k)$. As before, the null model utilizes random permutation of class labels when computing the SVMs as well as in the reconstruction step. In addition, we use a random assignment to the class of plus and minus neurons. In the regular model, the sign-specific weight vector is defined by splitting neurons in two groups according to the sign of the weight vector. In the null model, we remove this information by randomly permuting neural indexes before splitting neurons in two groups.

The population signal in cortical layers. Similarly to the split of the neural population according to the sign of decoding weight, we compute the population signal in three cortical layers. The method for determining cortical layers is described in the last section of methods, “Determining cortical layers from the current source density”. From the vector $\tilde{\mathbf{w}}$, we define three layer-specific weight vectors, $\tilde{\mathbf{w}}^r$, $r \in \{“SG”, “G”, “IG”\}$. The layer-specific weight vectors take into account weights of neurons from the specific layer, while weights of other neurons are replaced with zero. The reconstruction in layer r is defined as follows:

$$x_j^r(t_k) = F((\tilde{\mathbf{w}}^r)^T \mathbf{o}_j(t_k)) \tag{20}$$

Similarly to the reconstruction of spike trains for plus and minus neurons, the number of neurons across layers is not balanced. In order to compare the population signals across layers, we scale the weights with the number of neurons in layers, $\tilde{\mathbf{w}}^r c^r$, with the correction factor,

$$c^r = {}^{N}\!/_{3N^r}, \tag{21}$$

where N^r is the number of neurons in layer r. For the rest, the procedure is identical as for the plus and minus neurons, substituting sign-specific with layer-specific signals.

We compute the deviation of the population signal from the mean,

$$\tilde{x}_j^r(t_k) = x_j^r(t_k) - z^r(t_k) \tag{22}$$

with $z^r(t_k)$ the layer-specific average population signal.

$$z^r(t_k) = \frac{1}{J}\sum_{j=1}^{J} x_j^r(t_k) \tag{23}$$

Finally, we average across trials, distinguishing conditions “match” and “non-match”.

$$\begin{aligned} \tilde{x}^{r,nm}(t_k) &= \frac{1}{J'}\sum_{j=1}^{J'} \tilde{x}_j^r(t_k) \\ \tilde{x}^{r,m}(t_k) &= \frac{1}{J-J'}\sum_{j=J'+1}^{J} \tilde{x}_j^r(t_k) \end{aligned} \tag{24}$$

Correlation analysis

Correlation function of the population signals of plus and minus neurons. We compute the correlation function between population signals of plus and minus neurons in trial j,

$$R_j^{+-}(\tau) = \frac{1}{2K}\int_{-K}^{K} \tilde{x}_j^{+}(t_k+\tau)\tilde{x}_j^{-}(t_k)\; dt, \tag{25}$$

with time lag $\tau = -K+1, ..., 0, ..., K-1$. The correlation function is normalized with autocorrelation functions at zero time lag,

$$\tilde{R}_j^{+-}(\tau) = \frac{R_j^{+-}(\tau)}{R_j^{--}(0)R_j^{++}(0)}, \tag{26}$$

where R^{++} (R^{--}) is the autocorrelation functions for plus (minus) neurons,

$$R_j^{++}(\tau) = \frac{1}{2K}\int_{-K}^{K} \tilde{x}_j^{+}(t_k+\tau)\tilde{x}_j^{+}(t_k)\; dt. \tag{27}$$

The correlation function is computed in all trials and then averaged across trials, distinguishing conditions "match" and "non-match",

$$\begin{aligned} \tilde{R}_{nm}^{+-}(\tau) &= \frac{1}{J'}\sum_{j=1}^{J'} \tilde{R}_j^{+-}(\tau) \\ \tilde{R}_{m}^{+-}(\tau) &= \frac{1}{J-J'}\sum_{j=J'+1}^{J} \tilde{R}_j^{+-}(\tau). \end{aligned} \tag{28}$$

We estimate the significance of the correlation function with the permutation test. We compute the correlation function using trials from both conditions,

$$\tilde{R}^{+-}(\tau) = \frac{1}{J}\sum_{j=1}^{J} \tilde{R}_j^{+-}(\tau) \tag{29}$$

and compare it with the null model. The null model is computed with random assignment to the group of "plus" and "minus" neurons.

Correlation function of the population signals in cortical layers. Similarly to plus and minus neurons, we compute the correlation function between the population signals for each pair of cortical layers (SG and G, SG and IG, G and IG layer). As before, the correlation function is computed for the two population signals in the same trial,

$$R_j^{c_1,c_2}(\tau) = \frac{1}{2K}\int_{-K}^{K} \tilde{x}_j^{c_1}(t_k+\tau)\tilde{x}_j^{c_2}(t_k)\; dt, \tag{30}$$

with $(c_1, c_2) \in \{$"(SG, G)", "(SG, IG)", "(G, IG)"$\}$. The rest of the procedure is the same as for plus and minus neurons. The significance of results is evaluated with the permutation test. The null model is computed with random assignment to one of the three cortical layers.

Determining cortical layers from the current source density

Within the cortical depth, we distinguish three cortical layers, the middle or the granular layer, the superficial or the supragranular layer and the deep or the infragranular layer. Besides the multiunit signal, we also recorded the local field potential. We compute the current source density (CSD) as the second spatial derivative of the trial-averaged local field potential [18]. The CSD is a 3-dimensional tensor, that associates the direction of the current flow to every point in space and time. The normalized CSD is defined as follows: A_{ijk}, $i = 1, ..., N_{space}$, $j = 1, ..., N_{time}$, $k \in [-1, 1]$, where i extends in the spatial dimension, j in the temporal dimension and k is the direction of the current flow. Upon the presentation of a visual stimulus, the CSD shows a characteristic pattern of sinks and sources, with a current sink in the granular (input) layer and simultaneous current sources in the supra- and infragranular layers [19]. We utilize the pattern of sink and sources to determine the borders of the granular layer. First, we search for the strongest current sink after the onset of the test stimulus, which is the point in space and time with the maximal value of the current flow , $A_{max}(i', j', k_{max})$. The pattern of sink and sources is primarily a spatial feature and we capture it with the spatial covariance of the current source density,

$$\mathbf{C} = \frac{1}{N_{time}} \mathbf{A}^T \mathbf{A}, \tag{31}$$

where T denotes the transpose. We search for the vector of covariance, $\mathbf{c}_{max}$, that passes through the point A_{max},

$$\mathbf{c}_{max} = c_{1,max}, ..., c_{j,max}, ..., c_{N_{space},max}. \tag{32}$$

Along this particular vector of covariance, current sinks correspond to peaks and current sources correspond to troughs. We capture the peak that corresponds to the strongest current sink, $c_{j',max}$. On each side of this peak, the two troughs correspond to the two current sources of interest. We determine the upper and the lower border of the granular layer as points where the vector of covariance crosses the zero between the aforementioned peak and the neighboring troughs, since zero crossing corresponds to current inversion. Every recorded multi-unit was associated with the closest recording channel. After determining the borders of the granular layer, units above the upper border of the granular layer were assigned to the the supragranular layer and units recorded below the lower border to the infragranular layer.

Results

The population PSTH does not allow to discriminate correct choices on binary stimulus classes “match” and “non-match”.

Two adult monkeys *macaca mulatta* have been tested on the visual discrimination task with complex naturalistic images. In each trial, animal subjects visualized two images, the target and the test, with a delay period in between (fig. 1A). Each of the two stimuli was presented for 300 ms and the delay period lasted between 800 to 1000 ms. The length of the delay period has been randomized, to prevent that the subject anticipates the time of arrival of the second stimulus. Visual stimuli were images in black and white and represented an outdoor scene. The identity of the stimulus changed on every trial, but always fell into one of the two categories, “match” and “non-match”, where “match” indicates an identical pair of target and test stimuli, and “non-match” indicates the rotation of the test stimulus with respect to the target. After the visualization of the two stimuli, the subject was required to communicate its choice (“same” or “different”) by holding of releasing a bar and was rewarded for a correct response with fruit juice. Laminar arrays were inserted into the visual areas V1 and V4, perpendicularly to the cortical surface, spanning the cortical depth (fig. 1B). We recorded the multi-unit activity and, after the spike sorting, reliably identified between 3 and 17 neurons per recording session (on average, 8 neurons in V1 and 10.2 neurons in V4). Across the recording sessions, we collected the activity of 160 neurons in V1 and 102 neurons in V4.

We limited the analysis to correct trials and therefore distinguish two conditions, “match”' (correct choice on matching stimuli) and “non-match” (correct choice on non-matching stimuli). Classes “match” and “non-match” are conditioned on a mixed variable that potentially contains the information about both the stimulus and the choice and discriminating the two classes relies on either of these two sources of information or, more likely, a combination of both. Throughout the study, we analyze neural responses in two time windows, corresponding to the interval [0,400] ms with respect to the onset of the target and the test stimuli. The information, necessary for discriminating matching from non-matching stimuli, is only available during the test time window and we use the target time window to verify the coherence of the methods we use. The basic statistical methods do not allow to discriminate conditions “match” and “non-match” (fig 1D-E). The average firing rate of single neurons, for example, is highly variable across neurons but very similar across conditions (fig 1D). Similarly, population peri-stimulus time histograms (PSTHs) in conditions “match” and “non-match” are highly overlapping and do not allow to discriminate between the two conditions (fig 1E). With the population PSTH, the spiking activity is summed across neurons, with each neuron contributing equally to the sum (see methods, eq. 2). In the following, we will assume that the contribution of neurons within the population is not equal but is weighted according to neuron’s decoding weight. This reflects the fundamental idea that neural networks are not homogeneous ensembles but instead have a structure that allows them to perform relatively complex computations in a straightforward manner.

Weighting spike trains with population decoding weights allows to discriminate conditions “match” and “non-match”.

A single cortical neuron typically receives synaptic inputs from many projecting cells [8]. Through learning, cortical neurons adjust the strength of their synapses and here we assume that the preference of the projecting neuron for one of the two stimuli can determine whether its synapse with the read-out unit strengthens of weakens with respect to the baseline. If the input neuron on average fires more in condition “match” with respect to “non-match”, its synapse with the read-out neuron strengthens with respect to the baseline. Such a neuron is assigned a positive decoding weight. Inversely, if the input neuron fires more strongly in condition

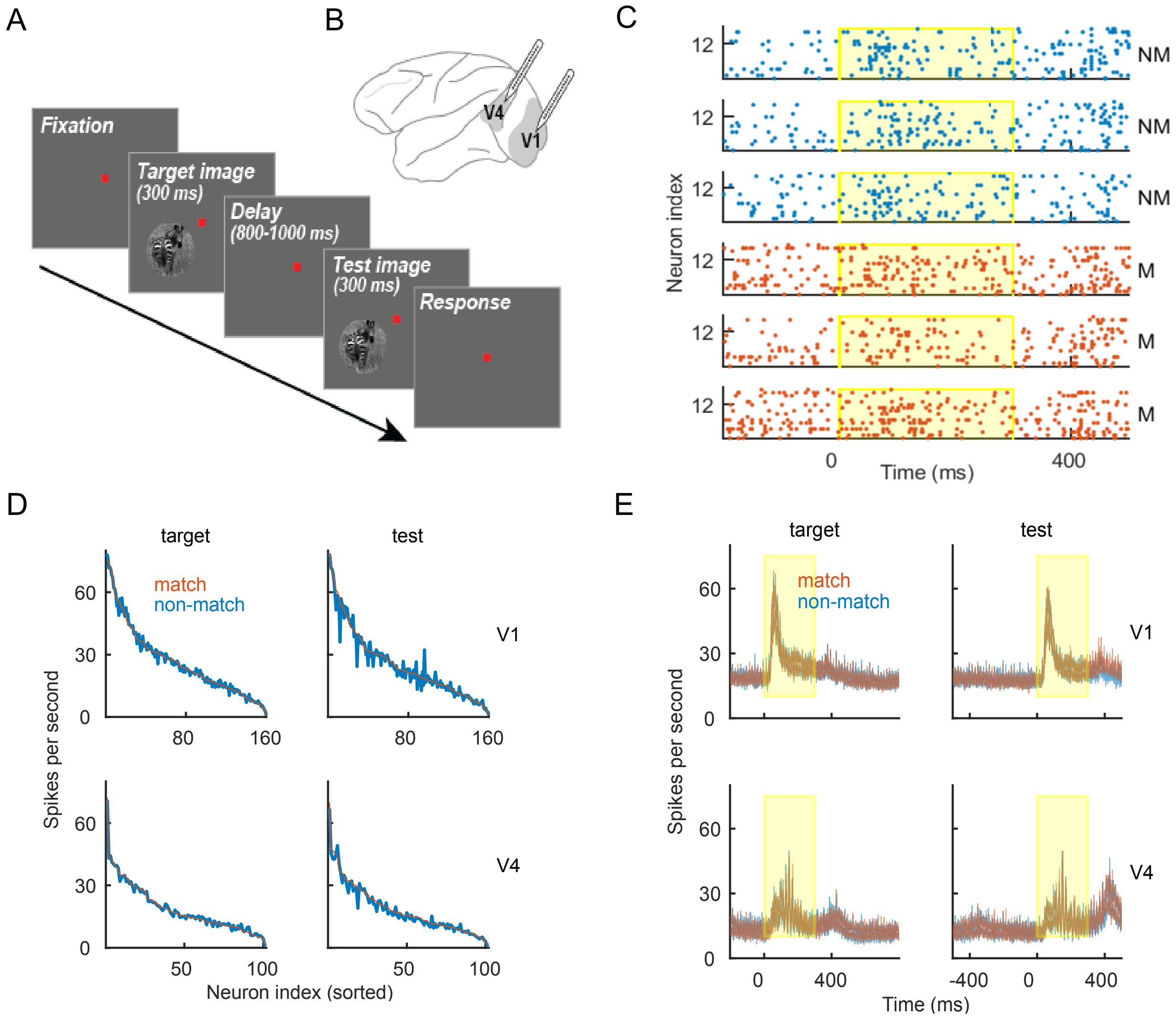

Figure 1. Experimental paradigm and spiking data. (A) Experimental paradigm. One trial consisted of the visualization of the target and the test stimuli, interleaved with the delay period. **(B)** Schema of the macaque brain with the location of recording sites. **(C)** Spike trains of an example recording session in V4 during the visualization of the test stimulus. We show 3 randomly selected trials in condition “non-match” (blue) and “match” (red). The yellow region marks the presence of the stimulus. **(D)** Mean firing rate of single neurons from all recording sessions in V1 (top) and in V4 (bottom) during the target time window (left) and the test time window (right). We show the firing rate in condition “match” (red) and “non-match” (blue). Neurons are sorted for the firing rate in condition “match”. **(E)** Population PSTH with the mean ± SEM across recording sessions in conditions “match” (red) and “non-match” (blue). The presence of the stimulus is marked as the yellow region.

"non-match" with respect to "match", its synapse with the read-out neuron weakens with respect to the baseline. This second neuron is assigned a negative weight. Notice that the positive and the negative sign of weights are only relative to each other and can be flipped, what matters is that signs of weights of the two neurons are opposite. Moreover, we suppose that weights are not learned for each single neuron independently, but are learned jointly for the entire population of projecting neurons. Decoding weights therefore describe the role of each neuron for the classification task, by also taking into account interactions between neurons.

Such decoding weights can be computed as the feature weights of an optimal linear classifier, the Support Vector Machine [16]. The latter has been chosen for its optimality and for the interpretability of results in the biological setting. We assume that learning of synaptic weights relies on firing rates and utilize spike counts of simultaneously recorded neurons as input to the classifier (see methods). Learning of weights would presumably take place while the animal is in the training phase of the experiment and is repeatedly exposed to the task. The animal is rewarded for correct behavior, Once the weights are learned, they are fixed and can be used in single trials and in real time to compute the population signal. The population signal is computed by weighting the spike trains, convolving with an exponential filter and summing across neurons. In mathematical terms, the population signal in trial j is given by the following expression: $x_j(t_k) = F(\tilde{\mathbf{w}}^T \mathbf{o}_j(t_k))$, where $\tilde{\mathbf{w}} = \tilde{w}_1, \tilde{w}_2, ..., \tilde{w}_N$ is the normalized vector of decoding weights of N simultaneously recorded neurons, $\mathbf{o}_j(t_k)$ is the N-dimensional vector of spike trains at the k-th time step and $F(y)$ is the transfer function, implementing the convolution with the exponential filter, $F(y) = \sum_{t \in T} F(y-t)u(t)$, with filter $u(t) = exp(-\lambda t)$. The convolution with an exponential filter models the causal effect of the presynaptic spike on the neural membrane of the read-out neuron. The population signal is computed in single trials and in real time, using decoding weights of neurons from the same recording session. We collect and average the population signal across trials, distinguishing conditions "match" and "non-match". For the step-by-step procedure and formal definitions, see methods (eq. 9-14).

We compute decoding weights and population signal independently in target and test time windows and report results in recording sessions as well as averaged across recording sessions. As expected, the population signal is highly overlapping in conditions "match" and "non-match" during the target time window (fig. 2A, left plots), while it diverges during the test time window (fig. 2A, right plots). This is true in both V1 and V4, however, temporal profiles are quite different between the two brain areas. In V1, the population signal in conditions "match" and "non-match" diverges early in the trial and stays approximately constant throughout the trial. In V4, the difference between the signals in the two conditions builds up in time and is the biggest towards the end of the trial. Interestingly, the difference in signals for "match" and "non-match" in V4 slowly oscillates and at the same time increases in every cycle. Considering the population signal as the input current to a read-out neuron, such dynamics would give windows of high and low probability for spiking in the read-out neuron. In order to estimate the significance of the difference between population signals in conditions "match" and "non-match", we compare the difference of signals from the true model, $x^{diff}(t_k) = \tilde{x}^m(t_k) - \tilde{x}^{nm}(t_k)$, with the same result from the null model. The null model is computed with random permutation of class labels (see methods). It can be appreciated that the difference of signals is not significant during the target time window,

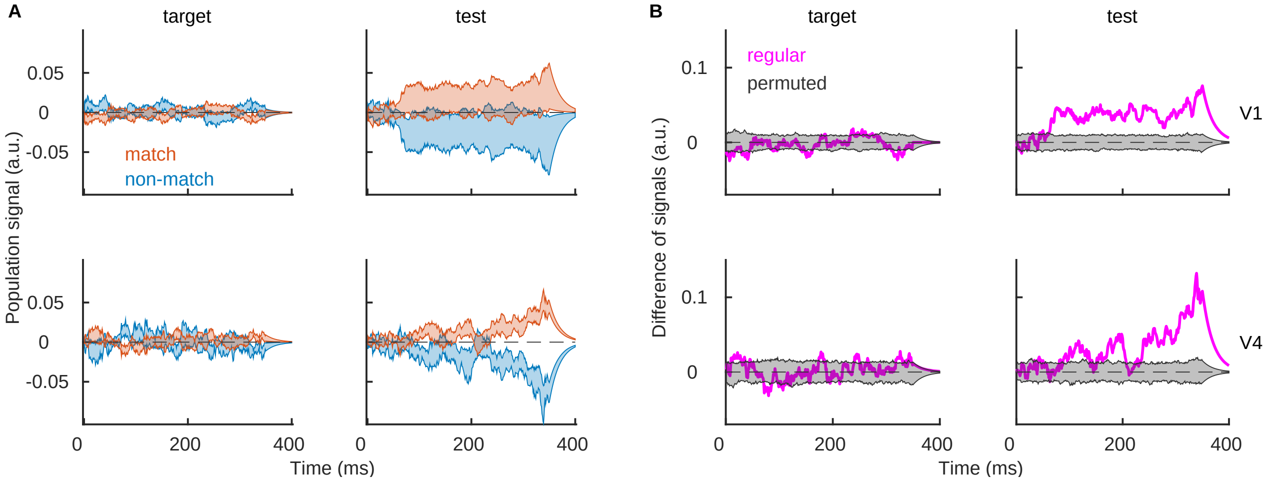

Figure 2. Population signal. (A) Population signal in V1 (top) and V4 (bottom) during the target (left) and the test time window (right). Shaded areas indicate the mean ± SEM for the variability across sessions in conditions "match" (red) and "non-match" (blue). **(B)** Difference between session-averaged population signals, showing the regular model (magenta trace) and the distribution of results of the null model (gray area). Parameters: $\lambda = 20^{-1}\ ms$, $nperm = 1000$.

(fig. 2B, left plots) while it is significant during the test time window (fig. 2B, right plots).

The exact time course of the population signal depends on the time constant of the convolution λ. With longer time constant, the effect of a spike lasts longer, giving rise to a signal that integrates more over time (fig. 3, left plots). The use of a linear filter, however, essentially only scales the amplitude of the signal, and this effect can be reversed by rescaling the signal with the area under the temporal filter, $\sum_{t\in T} u(t)$ (fig. 3, right plots). As signals are rescaled, the only difference between signals that use different time constants is smoothness, since longer time constants give smoother signal (compare the blue trace for the long time scale and the magenta trace for the short time scale). It has to be emphasized that the dynamics of the oscillating population signal in V4 therefore cannot be due to the convolution with a particular filter, since the same oscillatory dynamics is present for different time constants (fig. 3, bottom right) and since it is present for time constants that are shorter than the oscillation cycle. In the rest of the paper, we use the time constant $\lambda = 20^{-1}$. Moreover, consider that the stimulus is a static image and, as such, does not require integration over time. The observed temporal profile of the population signal therefore likely reflects the internal dynamics of the population.

Correct sign of weights is necessary and sufficient for discrimination.

There are two major components of the model that might inform successful discrimination, the weights and the spike timing (see methods, eq. 10). Note that weights have been computed utilizing spike counts and therefore do not contain information about spike timing. The information in decoding weights can be further split into the information contained in the sign (positive or negative) and in the modulus (the absolute value) of the entries of the weight vector. Here, we are interested in disentangling the contribution of different sources

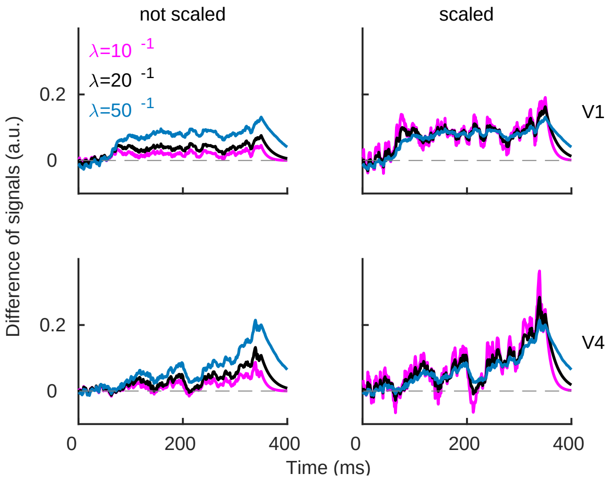

Figure 3. Dependency of the population signal on the time scale of the convolution and on the modulus of weights. Difference of population signals during the test time window for different values of parameter λ. We use time constants $\lambda = [10^{-1},\ 20^{-1}\ , 50^{-1}]\ ms$ and plot results without normalization (left) and with normalization (right).

of information for discrimination. We do this by randomizing one source and keeping others intact, and test whether the discrimination of conditions "match" and "non-match" is still possible. If a particular source of information is critical, randomizing it will compromise the discrimination.

Utilizing regular spike trains and random decoding weights, the signals in conditions "match" and "non-match" are highly overlapping (p=0.6046 in V1, p=0.8499 in V4, t-test on time-averaged signals with 1000 permutations), indicating that weights are critical for discrimination (fig. 4A, left plots). The random weight vector is given by drawing N random samples (where N is the number of neurons in the recording session) from the uniform distribution with the same range as the range of the regular weight vector.

Next, we disentangle the importance of the sign and of the modulus of weights for discrimination. The sign of weights is randomized by drawing N random samples from the uniform distribution, collecting their signs and applying them to the regular weights. Removing the information on the sign of weights gives highly overlapping population signals in conditions "match" and "non-match" (p=0.7354 in V1, p=0.4774 in V4, fig. 4, middle left). If, instead, we keep the correct sign and use random modulus, the population signal in conditions "match" and"non-match" are now diverging and can clearly be discriminated ($p < 10^{-8}$ in both areas, fig. 4A, middle right). We conclude that the information, contained in the sign of weights, is critical for discrimination, while the modulus is not.

Finally, we use regular weights but randomize the spike timing in the reconstruction step. This is done by randomly permuting the order of time steps of the spike train. The order of time steps is the same for all neurons within the population. Interestingly, with random spike timing, discrimination remains possible ($p < 10^{-8}$ in both areas, fig. 4A, right plots), showing that the spike timing is not critical for discrimination. However, randomizing the spike timing obviously affects the reconstructed signals, since the population signals,

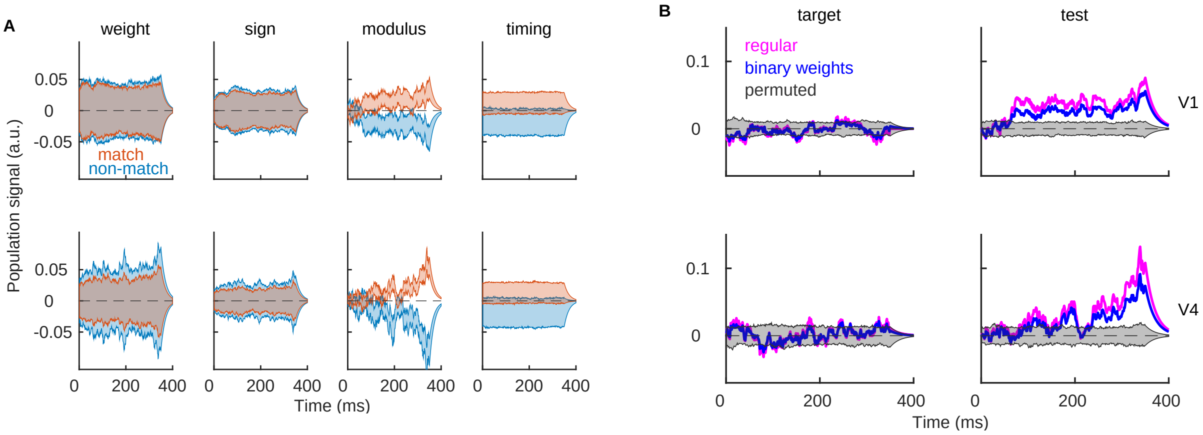

Figure 4. Distribution of population signals for models with specific permutation. (A) Population signal during the test time window, computed with random weights (left), random sign of weights (middle left), random modulus of weights (middle right) and randomly permuted spike timing (right). We show results in V1 (top) and in V4 (bottom) in conditions “non-match” (blue) and “match” (red). For all procedures, we used 1000 random permutations and we plot the entire distribution. **(B)** Difference of population signals for the regular model (magenta) and for the model with binary weights (blue). The gray area marks the distribution of results using models with permuted class labels. Parameters: $\lambda = 20^{-1}$ ms, $nperm = 1000$.

computed with random spike timing, have lost their temporal profiles. In particular, in might be that the oscillation, observed in the population signal of V4 neurons, is important for the transmission of the signal. If that is so, random spike timing would still compromise the read-out of the population signal.

Since results have shown that the modulus of weights is not critical for discrimination, we compute the population signal using binary weights. All positive weights are set to the same value, a, and all negative weights are set to its inverse, $-a$. We estimate the scalar a in such a way that the range of the weight vector is the same as with original set of weights, i.e., $a = N^{-1} \sum_1^N |w_n|$. Results show that the population signal with binary weights is similar to the population signal with regular weights (fig. 4B), and allows discrimination. Optimal weights still allow slightly greater discriminatory power than binary weights (e.g., the difference of signals in “match” and “non-match” is greater with optimal weights). Nevertheless, since the loss of discriminatory power is relatively small, we can relax the hypothesis of optimality, questionable in the biological network, to the correct assignment of the sign of weights.

Neurons with positive and negative weights respond with anti-symmetry.

Since the sign of weights is the crucial component of the model, we split neurons in two subpopulations according to the sign of weights and compute the population signal with each of the two subpopulations (see methods, eq. 15). During the test time window, we would expect neurons with positive and negative weights (in the following, plus and minus neurons) to respond differently. By design, minus neurons are those that prefer condition “non-match” and plus neurons are those that prefer condition “match”. Indeed, neurons with positive and negative

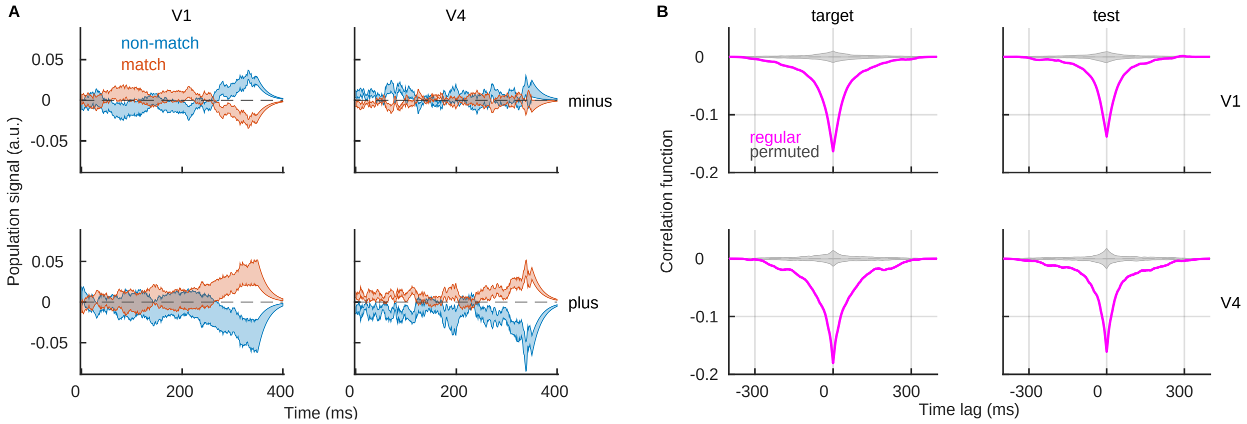

Figure 5. Plus and minus neurons respond with anti-symmetry. (A) Population signal of minus neurons (top) and plus neurons (bottom) during test time window. We show the mean ± SEM for the variability across recording sessions in conditions "match" (red) and "non-match" (blue). **(B)** Cross-correlation function between population signals of plus and minus neurons. We use trials from both conditions. Results are shown for the true model (magenta) and for models with random assignment to the plus and minus subpopulation. Parameters: $\lambda = 20^{-1}\ ms$, $nperm = 1000$.

weights respond with anti-symmetry, in particular in V1 (fig. 5A). Minus neurons increase the activity above the baseline in condition "non-match" and decrease below the baseline in condition "match" (fig. 5A, top left). Plus neurons, on the contrary, increase the activity above the baseline in condition "match" while they decrease the activity in condition "non-match" (fig. 5A, bottom). The divergence of the signals in "match" versus "non-match" appears most prominently at the end of the trial, which could be due to a neuron-type specific feedback from higher brain areas [23]. In V4, the anti-symmetric response is observed only with plus neurons (fig. 5A, bottom), whereas minus neurons do not seem to have any significant discriminatory power (fig. 5A, top right). As before, we test the significance of the difference between the population signals in conditions "match" and "non-match" with the permutation test (see methods). The signal in conditions "match" and "non-match" are significantly different in all cases but for minus neurons in V4 (supplementary fig. S1 C).

During the target time window, the population is expected to represent a "zero" signal [15]. This is indeed the case, the signal fluctuates around zero (supplementary fig. S1 A). There is no significant difference between the population signals in V1, while in V4, the difference of the signals shows a slow oscillation that crosses the significance boundary towards the middle of the trial and is in anti-phase for plus and minus neurons (supplementary fig. S1 B). Interestingly, such oscillation only appears in sign-specific subnetworks and was not present when the population signal was computed with all neurons (fig. 2B). This raises the possibility that in V4, neurons of both signs are required to maintain the correct representation of the "zero" signal.

The low-dimensional signal for plus and minus neurons has been computed by splitting the population of simultaneously recorded neurons into two subpopulations. The signals of plus and minus neurons therefore evolve simultaneously within the same trial. We compute the interaction between them with the cross-correlation function (see methods, eq. 25-28). Interestingly, there is a negative correlation between the two signals, consistently across the two brain areas and in both target and test time window (fig. 5B). There is no significant difference between correlation functions in conditions “match” and “non-match” (supplementary fig. S1 D). In general, the negative interaction between subnetworks of neurons with positive and negative weights appears as a strong and robust effect in both brain areas.

The superficial layer of the cortex discriminates best conditions “match” and “non-match”.

In the previous sections, we have split neural populations according to sign of their decoding function. In the last part, we split the neural population in three cortical layers, according to the spatial location of neurons across the cortical depth. We distinguish the superficial or the supragranular layer (SG), the middle or the granular layer (G), and the deep or the infragranular (IG) layer [20]. To split neurons in layers, we designed a simple method that utilizes patterns of activation of the current source density [18]. In general, the granular layer of the cortex is the input layer for sensory stimuli and is characterized by a current sink upon the presentation of a salient visual stimulus, while, simultaneosly, the supra- and infragranular layers present a current source [19] (fig. 6, left). This characteristic pattern of current sink and sources appears even more clearly in the spatial covariance of the current source density (see methods, eq. 31; fig. 6, top right). We determine the strongest current sink in the time window [20,100] ms after the onset of the test stimulus. We find the vector of covariance that passes through this particular point in space and time (eq. 32, vector $\mathbf{c_{max}}$ on fig. 6, top right). The strongest current sink is one of the peaks of this particular vector of covariance while neighboring troughs correspond to neighboring current sources (fig. 6, bottom right, red point for the sink and blue points for the sources). Between the peak and each of the troughs, the vector of covariance crosses the zero line. The upper and the lower boundary of the G layer are determined as the zero crossing, since these points correspond to the current inversion. All units above the G layer are assigned to the SG layer and all units below to the IG layer. For more information, see methods. In V1, there were 48, 51 and 61 neurons in the SG, G and IG layer, respectively. In V4, we identified 18 (SG), 42 (G) and 42 (IG) neurons in the respective layers.

As the population is split into three layers, we compute the population signal in each of the layers (see methods, eq. 20). Layer-specific population signals reveal that the superficial layers have the strongest discriminatory capacity of conditions “match” and “non-match” in both V1 and V4 (fig. 7). In V1, the middle layer shows a small effect during the first half of the trial (fig.7B, middle left) and in V4, middle and deep layers show an effect towards the end of the trial (fig. 7B, middle and bottom right). This is true for the test time window, while during the target time window, the population signal in all layers stays close to zero (supplementary figure **??**). The deep layer of V1 shows almost no discriminatory capacity in either target or test (fig. 7A, B, bottom left, supplementary figure **??**A,B, bottom left).

With layer-specific reconstruction of spike trains, we obtain three simultaneous population signals, one in each layer. We measure the linear interaction between population signals with the cross-correlation function (see methods, eq. 30). The correlation function is computed for every pair of layers (SG & G, SG & IG, and G

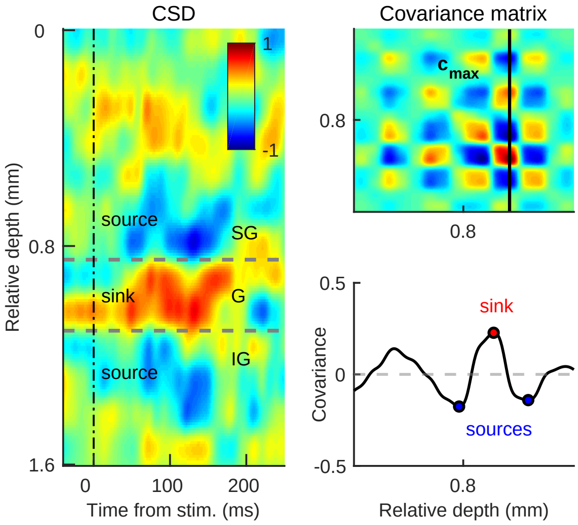

Figure 6. Method for assignment of cortical layers from the current source density. Left: Current source density during one recording session in V4. The CSD is the map of current sinks (red) and sources (blue). The x-axis shows time, relative to the onset of the test stimulus, and the y-axis shows the cortical depth, relative to the position of the upper channel on the laminar probe. After the onset of the stimulus, we observe a characteristic pattern of current sink and sources. The sink is a hallmark of the granular (G) layer, while sources characterize the supragranular (SG) and the infragranular (IG) layers. Top right: The pattern of sinks and sources is captured by the spatial covariance of the current source density. We select the vector of covariance that passes through the strongest sink, $\mathbf{c}_{max}$. Bottom right: Plotting the vector of covariance as a function of the cortical depth, one of the peaks corresponds to the strongest current sink (red point) and neighboring troughs correspond to current sources (blue).

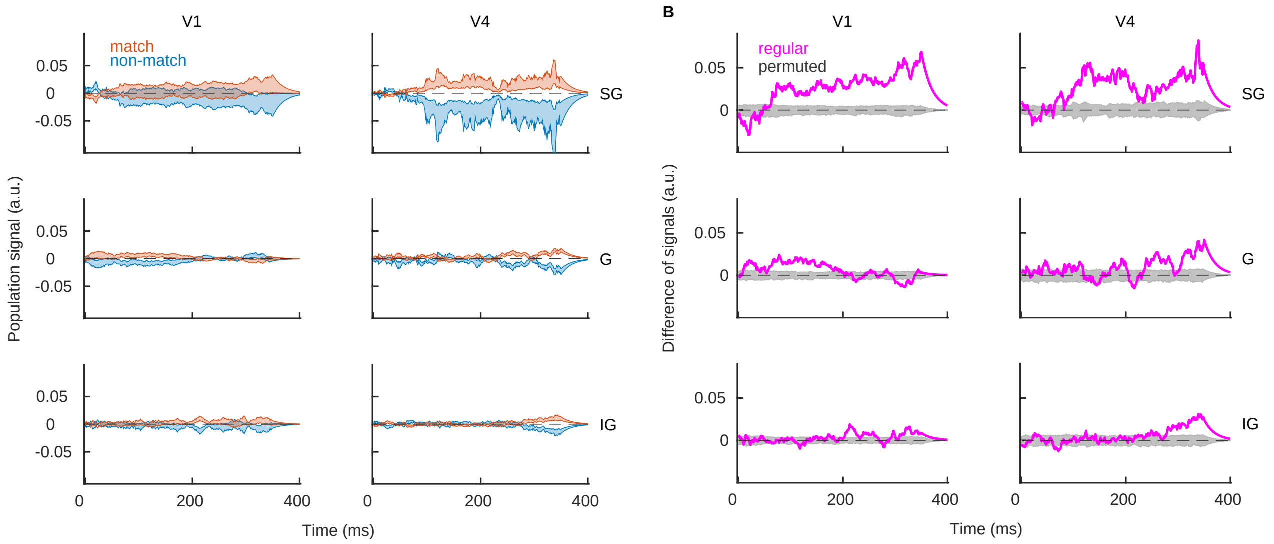

Figure 7. The superficial layer of the cortex is the best in discriminating conditions "match" from "non-match". (A) Population signal in recording sessions in the superficial (top), middle (middle) and deep cortical layers (bottom) in V1 (left) and in V4 (right). We show the mean ± SEM across recording sessions in conditions "match" (red) and "non-match" (blue). **(B)** Same as in **A**, but for the session averaged difference of signals. We show results of the regular model (magenta) and the distribution of results for the model with permutation (gray). Parameters: $\lambda = 20^{-1}\ ms$, $nperm = 1000$.

& IG layers). Results show positive correlation across all pairs of layers during both test (fig. 8) and target time window (supplementary fig. **??**A). There is no significant difference across conditions "match" and "non-match" fig. **??**B,C). We conclude that positive correlation across layers is a robust and generic property of the cortex, similarly to the negative correlation between subnetworks of plus and minus neurons.

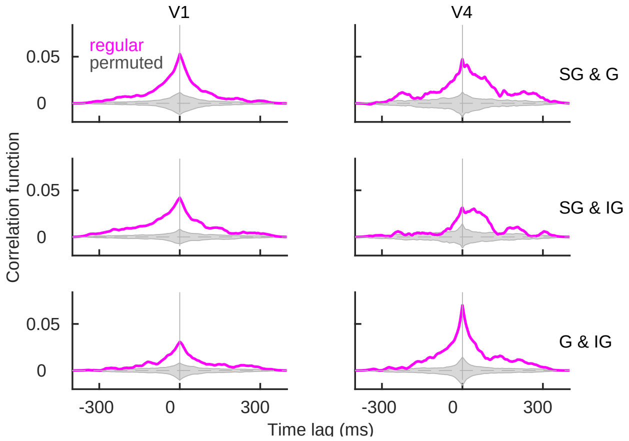

Figure 8. Population signals are positively correlated across the cortical layers. The cross-correlation function for pairs of cortical layers, SG & G (top), SG & IG (middle) and G & IG (bottom) during the test time window. The cross-correlation function uses trials from both conditions. We plot results of the regular model (magenta) and distribution of results for models with permutation (gray). Parameters: $\lambda = 20^{-1}\ ms$, $nperm = 1000$.

Discussion

We presented a novel method of the read-out of parallel spike trains that exploits the structure of the population code. We assumed the point of view of a read-out neuron, receiving synaptic inputs from a population of projecting neurons. The main feature of the present read-out method is to determine the synaptic weight between the projecting neuron and the read-out unit according to the functional role of the projecting neuron for the classification of the stimuli. The weight determines how does the spike of a particular neuron affect the population signal. After weighting spikes and summing across neurons, we get a 1-dimensional signal, evolving in real time, that can be thought of as the sum of post-synaptic potentials at the read-out unit, modulating the spiking probability of the latter. We have shown that the population signal clearly allows to discriminate conditions "match" and "non-match" in V1 and V4 during the test time window, when we would expect them to do so. Importantly, the population signal does not allow the discrimination during the target time window, when we would not expect it to do so. We demonstrated that discrimination critically depends on the correct assignment of the sign of the weight, corresponding to the correct assignment of neuron's preference for matching versus non-matching stimuli. Results show that neurons with the opposite sign of weight respond in anti-symmetric fashion to the mutually exclusive stimulus classes "match" and "non-match" and that the population signals of neurons from the opposing coding pools are negatively correlated (for a related analysis in the retina, see [21]). Distinguishing superficial, middle and deep layers of the cortex, we show that the superficial layer is the most performant in discriminating the two behavioral conditions in both brain areas.

The present method gives insights in the structure of the population code and into how this structure allows computation at the level of neural networks. The method can be easily adapted to any data set with parallel spike trains where it is possible to assume the nature of the computation that underlies the neural activity. While the concept of weighted spike trains is generally applicable, the way the population decoding weights are computed has to be adapted to the specific case at hand. Moreover, decoding weights do not necessarily need to be computed through a supervised training method, as we have done here, but an unsupervised method could be used instead. Our choice of the linear SVM is justified with SVM's optimality for binary classification tasks and we would expect that the use of another classifier would results in a less performant decoding model. We argue that, in the present context, supervised learning with the help of a teaching signal is a plausible assumption. The animal is rewarded for the correct behavior and the reward signal could enact the teaching signal.

In the present experimental setting, the behavioral task consists in matching delayed samples. The decision of the animal ("same" or "different") is based on the comparison of the test stimulus with the stimulus from the past (the target stimulus). This is a relatively complex cognitive task that presumably requires the activation of the working memory [22]. The visual areas, in particular their superficial layers, receive top-down projections [23], while, at the same time, they are also driven by the bottom-up inputs. It has been shown theoretically that in the presence of the common top-down input, population-wise coding weights can be learned with local synaptic plasticity rules [24]. We speculate that after learning, the top-down input to V1 and V4 could selectively target plus or minus neurons, depending on the condition. Such a context-dependent top-down signal could be computed in the prefrontal cortex, as described in [25]. Neurons with positive weight would be preferentially targeted in condition "match" and neurons with negative weight in condition "non-match". Such a top-down signal could explain the anti-symmetric activation of plus and minus neurons at the end of the trial. Among the three cortical layers that have been tested, the superficial layers have shown the best discrimination capacity. The strength of the effect in the superficial layers of V1 and V4 with respect to other layers corroborates our idea of the top-down influence on representation, since the superficial layers receive top-down inputs most abundantly and are as such best suited to perform the required computation.

From the biophysical point of view, it has been understood that variable spiking of single neurons can be mechanistically accounted for by the high-conductance regime [26], where the stream of excitatory and inhibitory synaptic inputs largely cancel each other out [27]. While the mean excitatory and inhibitory currents cancel each other out, remaining fluctuation of the membrane potential gives rise to variable spiking. It has been proposed that similar mechanism operates at the level of the population signal, where neurons with opposite coding function cancel out each other's effect[15], and that such canceling is critical for the correct representation of the population signal. In this respect, the method presented here is fundamentally different from methods that do not assume a coding function and where the dimensionality reduction consists in collapsing the dimensionality of homogeneous neural ensembles (with a population PSTH, as done here, or using a more advanced method [28, 29]). According to our results, it is possible to simplify the read-out of cortical spike trains by assuming binary read-out weights (e.g., for the purpose of the analytical treatment). However, any further simplification would wipe away the ability of the neural ensemble to perform computation. This is obvious if we consider that the population PSTH during test time window does not allow to discriminate conditions "match" and "non-match", while the population signal clearly does. It is also interesting to consider the striking difference between the population PSTH and the population signal during target time window. While we observe a strong peak of activity after the stimulus onset with the former, the latter stays around zero. Presumably, this is so because neurons with opposite coding function cancel out each other's effect and maintain the representation of the "zero" signal [15].

In the present work, we have assumed, for simplicity, that all neurons within the population project to the same read-out unit. We argue that in our case, this assumption is reasonable, since neural populations have been recorded across the cortical depth. Because of the retinotopic organization of the cortex [30, 31], neurons that span the cortical depth perpendicularly to the surface share a large proportion of their inputs, and project, at least partially, to the same read-out units. Moreover, it has to be emphasized that removing some of the neurons from the population does not change much the population signal, as long as the sign of weights of remaining neurons is correctly assigned. Nevertheless, it would be interesting to directly verify the validity of present results with an experimental assay in the behaving animal, where the activity of the read-out unit and the projecting neurons is monitored simultaneously, the experiment envisioned by Hubel and Wiesel [1]. On the modeling side, an interesting way of extending present method would consist in simulating a realistic model of the cortical column with coding functionality. Having a realistic number of neurons and biologically plausible connectivity structure would allow to estimate how does the discrimination capacity of the network scale with the number of neurons. Such a model could also help to gain further insights on the representation of task variables in real time. Such questions are still difficult to address in data sets, that give only a small subsample of all the units that are active in parallel in the biological networks.

Acknowledgements

This work was supported by Deutsche Forschungsgemeinschaft (GRK 1589/2).

References

1. Hubel, D. H., & Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. The Journal of physiology, 160(1), 106-154.

2. Adesnik, H., & Naka, A. (2018). Cracking the Function of Layers in the Sensory Cortex. Neuron, 100(5), 1028-1043.

3. Gao, P., & Ganguli, S. (2015). On simplicity and complexity in the brave new world of large-scale neuroscience. Current opinion in neurobiology, 32, 148-155.

4. Shadlen, M. N., & Newsome, W. T. (1994). Noise, neural codes and cortical organization. Current opinion in neurobiology, 4(4), 569-579.

5. Zohary E, Shadlen MN, Newsome WT, Correlated neuronal discharge rate and its implications for psychophysical performance, Nature 1994;370:140.

6. Britten, K. H., Newsome, W. T., Shadlen, M. N., Celebrini, S., & Movshon, J. A. (1996). A relationship between behavioral choice and the visual responses of neurons in macaque MT. Visual neuroscience, 13(1), 87-100.

7. Shadlen, M. N., Newsome, W. T. (1998). The variable discharge of cortical neurons: implications for connectivity, computation, and information coding. The Journal of neuroscience, 18(10), 3870-3896.

8. Destexhe, A., Rudolph, M., & Paré, D. (2003). The high-conductance state of neocortical neurons in vivo. Nature reviews neuroscience, 4(9), 739.

9. Faisal, A. A., Selen, L. P., & Wolpert, D. M. (2008). Noise in the nervous system. Nature reviews neuroscience, 9(4), 292.

10. Beck, J.M., Ma, W.J., Pitkow, X., Latham, P.E., Pouget, A.: Not noisy, just wrong: the role of suboptimal inference in behavioral variability. Neuron 2012, 74:30-39.

11. Körding, K. P., & Wolpert, D. M. (2004). Bayesian integration in sensorimotor learning. Nature, 427(6971), 244.

12. Boerlin, M., Machens, C. K., & Denève, S. (2013). Predictive coding of dynamical variables in balanced spiking networks. PLoS computational biology, 9(11), e1003258.

13. Denève, S., & Machens, C. K. (2016). Efficient codes and balanced networks. Nature neuroscience, 19(3), 375.

14. Chalk, M., Gutkin, B., & Deneve, S. (2016). Neural oscillations as a signature of efficient coding in the presence of synaptic delays. Elife, 5, e13824.

15. Koren, V., & Denève, S. (2017). Computational Account of Spontaneous Activity as a Signature of Predictive Coding. PLoS computational biology, 13(1), e1005355.

16. Vapnik, V. N., & Vapnik, V. (1998). Statistical learning theory (Vol. 1). New York: Wiley.

17. Dayan, P., & Abbott, L. F. (2001). Theoretical neuroscience (Vol. 806). Cambridge, MA: MIT Press.

18. Pettersen, K. H., Devor, A., Ulbert, I., Dale, A. M., & Einevoll, G. T. (2006). Current-source density estimation based on inversion of electrostatic forward solution: effects of finite extent of neuronal activity and conductivity discontinuities. Journal of neuroscience methods, 154(1), 116-133.

19. Maier, A., Aura, C. J., & Leopold, D. A. (2011). Infragranular sources of sustained local field potential responses in macaque primary visual cortex. Journal of Neuroscience, 31(6), 1971-1980.

20. Hansen BJ, Chelaru MI, Dragoi V, Correlated variability in laminar cortical circuits, Neuron. 2012 Nov 8;76(3):590-602

21. Kühn, N. K., & Gollisch, T. (2019). Activity Correlations between Direction-Selective Retinal Ganglion Cells Synergistically Enhance Motion Decoding from Complex Visual Scenes. Neuron.

22. Gold, J. I., & Shadlen, M. N. (2007). The neural basis of decision making. Annual review of neuroscience, 30.

23. Gilbert, C. D., & Sigman, M. (2007). Brain states: top-down influences in sensory processing. Neuron, 54(5), 677-696.

24. Denve, S., Alemi, A., & Bourdoukan, R. (2017). The brain as an efficient and robust adaptive learner. Neuron, 94(5), 969-977.

25. Mante, V., Sussillo, D., Shenoy, K. V., & Newsome, W. T. (2013). Context-dependent computation by recurrent dynamics in prefrontal cortex. nature, 503(7474), 78.

26. Fellous, J. M., Rudolph, M., Destexhe, A., & Sejnowski, T. J. (2003). Synaptic background noise controls the input/output characteristics of single cells in an in vitro model of in vivo activity. Neuroscience, 122(3), 811-829.

27. Vreeswijk, C. V., & Sompolinsky, H. (1998). Chaotic balanced state in a model of cortical circuits. Neural computation, 10(6), 1321-1371.

28. Augustin, M., Ladenbauer, J., Baumann, F., & Obermayer, K. (2017). Low-dimensional spike rate models derived from networks of adaptive integrate-and-fire neurons: comparison and implementation. PLoS computational biology, 13(6), e1005545.

29. Schwalger, T., Deger, M., & Gerstner, W. (2017). Towards a theory of cortical columns: From spiking neurons to interacting neural populations of finite size. PLoS computational biology, 13(4), e1005507.

30. Blasdel, G., & Campbell, D. (2001). Functional retinotopy of monkey visual cortex. Journal of Neuroscience, 21(20), 8286-8301.

31. Blasdel, G. G., & Fitzpatrick, D. (1984). Physiological organization of layer 4 in macaque striate cortex. Journal of Neuroscience, 4(3), 880-895.