The structure of the population code in neural ensembles of V4 shapes pair-wise interactions on different time scales.

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

Highlights

- Responses of a population of neurons in V1 and V4 to complex naturalistic images have more information about correct choice than an average single neuron.
- Informative neurons are more strongly coupled, correlated and synchronized than uninformative neurons in V4, but not in V1.
- Neurons are more strongly coupled, correlated and synchronized within coding pools compared to across coding pools in V4, but not in V1.
- The structure of population responses contributes substantially to the performance of the decoder. In V1, the structure across all neurons matters, while in V4, only the structure within the coding pool is important.

Abstract

In visual areas of primates, neurons activate in parallel while the animal is engaged in a behavioral task. In this study, we examine the structure of the population code while the animal performs delayed match to sample task on complex natural images. The macaque monkeys visualized two consecutive stimuli that were either the same or different, while recorded with laminar arrays across the cortical depth in V1 and V4 cortical areas. We decoded correct choice behavior from the activity of single neurons as well as from neural populations of simultaneously recorded units. Comparing the predictive power of the activity of single neurons with the high-dimensional model of the population activity, we find that the high-dimensional read-out predicts correct choices better than an average single neuron from the same recording session. Utilizing decoding weights, we divide neurons in informative and uninformative, and show that informative neurons in V4, but not in V1, are more strongly synchronized and also have stronger correlations than uninformative neurons. As neurons are divided in two coding pools according to their coding preference for matching and non-matching stimuli, in V4, but not in V1, spiking synchrony and correlations within the coding pool are stronger than across coding pools. Finally, decorrelating neural activities within the coding pool increases the performance of the decoder, while decorrelating across pools does not affect the performance of the decoder. In summary, our analysis points out that cortical microcircuits are structured and that the structure of the population code importantly shapes pairwise correlations.

*Correspondence: koren@ni.tu-berlin.de [1]

[1]lead contact

Introduction

Traditionally, stimuli in choice probability paradigms targeted a specific feature of the stimulus that a given brain area is known to respond to [1, 2, 3, 4]. While cited studies made an important breakthrough in our understanding of the function of the cortical activity, a way to further improve our understanding of the neural function is to study responses of neural ensembles to complex natural stimuli. In real-world scenarios, animals use sensory information in a flexible way, showing the ability to manipulate the information with temporal delays and to classify stimuli with respect to abstract rules. The coding function of neural responses in the early visual cortex is often described as a set of filters [5], where each filter responds exclusively to a specific portion of the specific feature of the stimulus. We propose to study responses of neural ensembles to naturalistic stimuli, where the classification task does not rely on filtering properties of single neurons, but instead requires discrimination according to an abstract context. In particular, the animal visualizes two stimuli consecutively, interleaved with a delay, and has to judge whether they are same ("match") or different ("non-match").

The way neural populations encode task-relevant information could either be a by-product of the activity of single units or else have an intrinsic structure that cannot be explained by the response properties of single units [6]. The organism has to generate behavioral decisions in every trial while the trial-to-trial variability of cortical neurons makes the signal conveyed by single neurons unreliable. Summing activities across neurons can, in general, overcome the variability problem, but only in case where the variability is uncorrelated across neurons [7]. This clearly does not hold for local neural ensembles that share a big proportion of their inputs [8, 9]. On a conceptual level, if neurons are correlated, there should be a non-trivial relation between activities of single neurons and the population code. Recent theoretical and modeling work has shown that the variability of neural responses can be reconciled with deterministic population signals [10, 11]. This is possible if we assume that task variables (such as the stimulus and the behavioral choice) are represented by the network as a low-dimensional signal, and the coding of such a signal is distributed across a multitude of neurons with redundant coding function. While coding of low-dimensional variables with spiking neural networks has been laid down in theory, the relation between the activity of single neurons and the signal that is effectively transmitted downstream and contributes to the choice behavior remains elusive and unclear in a biological networks of behaving animal agents.

Over the last decade, two major insights have been made on the principles of neural coding in cortical networks. The first insight concerns the relation between the similarity of the tuning function of a pair of neurons (signal correlation) and their common variability across trials (noise correlation). If a pair of neurons has similar tuning, positive noise correlations will harm the information transfer. Vice-versa, if a pair of neurons has dissimilar tuning, positive noise correlation can enhance the quantity of the transferred information [12]. In order to maximize the information transfer, neurons with similar tuning therefore should have low correlations, while neurons with dissimilar tuning should have stronger correlations. However, in the brain, tens to hundreds of thousands of neurons activate during a visual task with complex stimuli. Neural signals are effective only if they are transmitted downstream, and the activity that maximizes the information transfer might result in a neural signal that cannot be reliably read-out by a downstream neuron. The second insight on neural coding and dynamics comes from the observation of neural activity with respect to animal's choice behavior. While many neurons activate during a visual task, in experiments, we only observe a tiny fraction of these. Surprisingly, this small subset of neurons appears to be informative about the choice behavior of animal agents. This can be

explained with the existence of coding pools, where a (large) number of neurons share a similar coding function. If within the coding pool neurons are more strongly correlated than across pools, the activity of a randomly chosen neuron from the pool has a high probability of being informative about the behavioral variable. In short, if correlations within the pool are stronger than across pools, such a correlation structure allows to infer task variables from the observation of only a small number of neurons [13, 14]. At the first glance, these two ideas imply divergent views on the role of correlations for coding of behavioral variables, since correlations that harm the information transfer are also those that should be the strongest.

In the present work, we study the coding function of the network and relate it to correlations. We assume that the computation that the neural network is performing is discrimination of binary stimulus classes. We compute the coding function of a single neuron relatively to the activity of the network that surrounds the neuron and also takes into account the inter-neuron interactions. We use decoding weights of an optimal decoder, since an optimal decoder gives the upper bound of the transferred information [15], and ensures that the relation between decoding weights and neural dynamics is actually relevant for the information processing and behavior [16]. As we relate the strength of the coding function to single neuron and pair-wise dynamics, we find a strong interplay between the two in neural networks of V4, but not in V1. We divide neurons in two mutually exclusive coding pools and show that in V4, but not in V1, correlations within pools are stronger than across pools. In general, the neural code in V1 and V4 shows many differences, which might reflect different stages of information processing from analysis of the visual scene (V1), towards convergent information about binary choice behavior (V4).

Results

Neurons in the brain read-out the activity of projecting units.

Two adult monkeys *Macaca mulatta* performed visual discrimination task on matching and non-matching stimuli. In every trial, the subject visualized two consecutive stimuli, with a delay period in between (fig. 1A). Stimuli were complex naturalistic images in black and white, lasted 300 ms each and were either identical (condition "match") or else the test stimulus was rotated with respect to the target stimulus (condition "non-match"). The angle of rotation changed randomly across trials and was adjusted on-line to have on average 70 % of correct responses with non-matching stimuli. The identity of the stimuli was changed from one trial to another. The task of the animal was to decide about the similarity of the two stimuli by holding a bar for the choice "different" and releasing the bar for the choice "same". Laminar arrays with 16 recording channels were inserted perpendicularly to the surface of the brain, recording the multi-unit signal across the cortical depth in V1 and V4 visual areas (see methods; fig. 1B). In both areas, neural populations respond with a transient, followed by tonic firing (fig. 1C).

The general understanding is that sensory neurons respond to (features of) sensory stimuli and transmit this information downstream, where other neurons read-out their activity and use it for generating the behavioral choice [4]. Neurons and neural ensembles perform their coding function with their dynamics (e.g., by firing more for stimulus 1 than for stimulus 2), and this process is conceptualized as a read-out or "decoding". Decoding of neural activity tells us how accurately one can predict the future behavioral choice of the animal by observing the neural dynamics that precedes it. Here, we investigate decoding models that are readily interpretable in

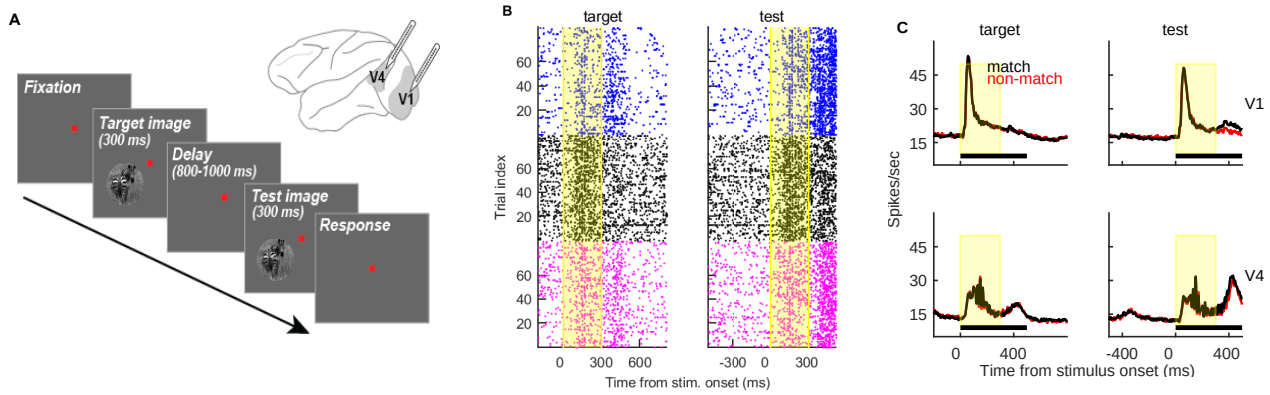

Figure 1. Experimental paradigm. A) Schema of the experimental paradigm and illustration of the macaque brain, with shaded areas indicating visual areas V1 and V4. **B)** Raster plots of three representative V4 neurons (blue, black, magenta) in condition "non-match", during the presentation of the target (left) and the test image (right). **C)** Population PSTH (see methods) in V1 (top) and in V4 (bottom), during target (left) and test stimuli (right).

the biological setting, i.e., models that could be implemented in the brain itself. We adopt the view of a read-out neuron that can estimate the average firing rate from the projecting unit and calculate the probability of discriminating between the two stimulus classes, given the firing rate of the projecting unit in the single trial. We limit our investigation to correct choices, i.e., trials where the visualization of one of the two possible stimulus types ("match" and "non-match") has led to the congruent behavioral choice. In correct trials, the representation of the stimulus is expected to be the primary source of information for determining the choice behavior. However, at least part of the information about the choice is likely to exist independently of the stimulus, possibly influencing sensory areas as the feedback activity from higher brain areas [17]. We therefore assume the existence of a mixed variable that contains the information about both the stimulus class as well as the behavioral choice (Table 1).

Stimulus	Choice	Performance	Info. content
"non-match"	"different"	correct	Stimulus + Choice
"match"	"same"	correct	

Table 1. Informational content of the task variable.

High-dimensional read-out of the activity of neural populations predicts correct choices better than an average single neuron.

We decode the choice behavior from the activity of neural populations, using an optimal linear classifier, linear Support Vector Machine (SVM) [15]. The quality of prediction of the SVM is assessed on hold out sets with balanced accuracy (BAC), which is the probability of correctly predicting the choice in single trials. In case all test samples were classified correctly, we would have the balanced accuracy of 1. The balanced accuracy of 0.5 indicates that the validation data has been classified equally often correctly as incorrectly, which is chance level performance. Significance of the prediction accuracy is evaluated with a permutation test, using models with randomly permuted class labels (see methods). Results show that the high-dimensional model performs

significantly better than chance during the test time window ($p < 0.001$ in both areas, permutation test, Bonferroni correction, fig. 2A). On average, the high-dimensional model predicts the choice behavior 8 % better than chance in V1 and 6 % in V4. During the target time window, the prediction is at chance ($BAC = 0.50$ in V1 and $BAC = 0.49$ in V4, non-significant). The use of a nonlinear decoder (SVM with Radial Basis Function kernel) did not improve the prediction accuracy with respect to the linear model (supplementary fig. S1 A). Since the SVM is the optimal classifier, the performance of the linear SVM coincides with the amount of information that spike counts of neural populations contain about the correct choice behavior. Seen the abstract nature of the discrimination task ("same" versus "different" with stimuli that change in each trial), that prevent neural populations to rely on filering properties of single neurons, better than chance prediction is an unexpected and nontrivial result.

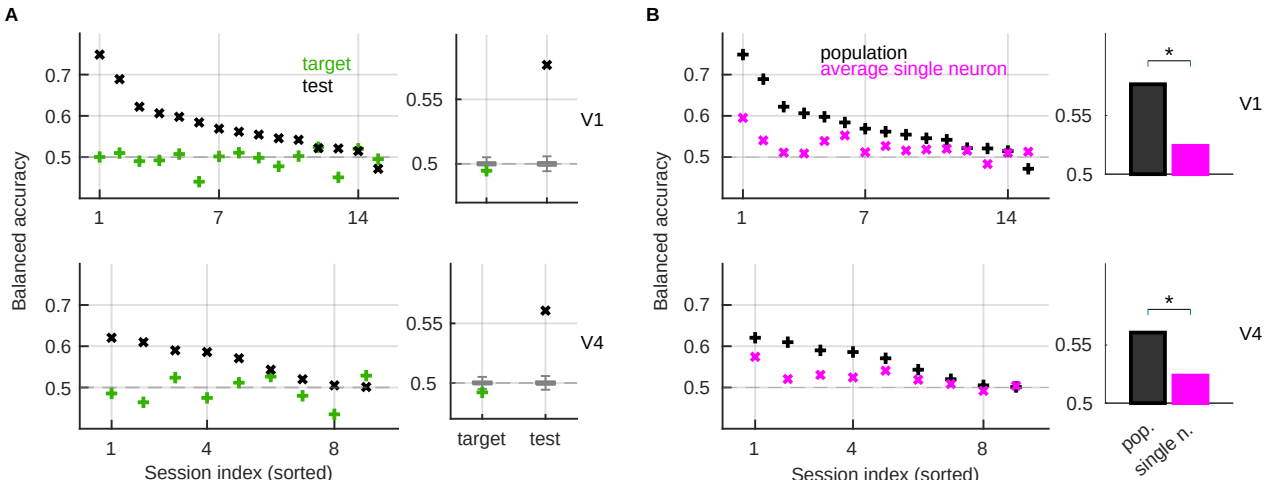

Figure 2. The high-dimensional model predicts the choice behavior better than an average single neuron. A) Left: Prediction accuracy of the population model during target (green) and test (black) in V1 (top) and in V4 (bottom). Results have been computed in each recording sessions and have been sorted, from best to worst performance, for BAC during test. Right: Session-averaged balanced accuracy and the distribution of results from models with permuted class labels (boxplots). **B)** Left: Prediction accuracy of the population model (black) and of the average single neuron (magenta) in V1 (top) and in V4 (bottom) during test. Results have been sorted with respect to the population model. Right: Average prediction accuracy across sessions for the population model (black) and for average single neuron model (magenta). The asterisk marks significance.

Next, we compare the prediction accuracy of the population model with the prediction accuracy of an average single neuron. For the latter, we apply a linear SVM to the activity of each single neuron and average results across neurons from the same recording session (see methods). Clearly, the prediction of the population model is more accurate than the prediction of an average single neuron ($p = 0.002$ in V1 and $p = 0.039$ in V4, Wilcoxon signed-rank test on results in sessions; fig 2B). The population model outperforms the average single neuron for 5.2 % in V1 and for 3.7 % in V4 ($p = 0.002$ in V1, $p = 0.008$ in V4, permutation test with 1000 permutations). During target, the difference between the population model and the single neurons model is not significant (supplementary fig. S1 B).

The population code has its own intrinsic structure.

The fact that the population model has higher prediction accuracy than an average single neuron suggests that the population model cannot be reduced to the sum of predictive functions of single neurons without loss of information [6]. While this is expected, the question remains, how does the population code relate to the discriminatory power of single neurons. The use of a linear SVM makes the comparison between the population code and the single neuron code straightforward, since it allows to compute weights of the population model (see methods). Population weights describe the role of each neuron for the classification task, relatively to the activity of other neurons from the population and by taking into account inter-neuron interactions (fig. 3A). An identical activity of a single neuron would results in a different decoding weight, if this neuron was surrounded by a different population. The classification model of the linear SVM is a high-dimensional plane in the space of inputs, fully defined by the vector of weights and the offset of the plane from the origin. If the activity of a particular neuron has no role in discrimination, its weight will be 0. The bigger the deviation of neuron's weight from 0, the stronger the importance of the neuron for discrimination. The sign of the weight is also informative, since weights of the opposite sign indicate opponent decoding functions. Computing weights of the population model for simultaneously recorded neurons and collecting results across recording sessions, we find that almost no neurons have significant weights (permutation test using weights from models trained on permuted class labels, see methods; fig. 3C, left). Pooling weights across neurons, we find that during test, positive and negative weights are balanced in V1, but unbalanced in V4, with an excess of positive weights ($p = 0.005$, permutation test; fig. 3C, right). During target, positive and negative weights are balanced in both areas (supplementary fig. S2 A).

Next, we estimate the discriminatory power of single neurons as the area under the Receiver-Operating characteristic curve ($AUROC$, see methods; fig. 3B). The $AUROC$ score measures the distance between distributions of spike counts in conditions "match" and "non-match" and is the probability of predicting the condition from the observation of spike counts. Identical distributions yield $AUROC = 0.5$. Neurons with $AUROC > 0.5$ are those that fire more in condition "match" while neurons with $AUROC < 0.5$ are those that fire more in condition "non-match". We measure the $AUROC$ score for each single neuron and collect results across recording sessions. We find that a small proportion of neurons have $AUROC$ significantly different than chance (13 % in V1 and 8 % in V4, 2-tailed permutation test with 1000 permutations, Bonferroni correction; fig 3D, left). As we pool $AUROC$ scores across neurons and across sessions, we find that during the test time window, there are more neurons that prefer matching than non-matching stimuli ($p < 0.001$ in V1 and in V4, permutation test on the sum of $AUROC$ scores; fig. 3D, right). The shift of the mean $AUROC$ from 0.5 is of 3% in V1 and of 4 % in V4. During the target, the distribution of AUROC scores is centered around 0.5 in both areas (supplementary fig. S2 B).

Finally, we compare weights of the population model and $AUROC$ scores of single neurons. The two measures are highly positively correlated during both target ($R = 0.75$ in V1, $R = 0.83$ in V4) and test ($R = 0.71$ in V1, $R = 0.80$ in V4, fig. 3D). These high positive correlations show that the weights of the population model are similar to the area under the ROC curve. The strength of the population weight therefore importantly relies on the difference of spike counts in conditions "match" and "non-match" of single neurons. However, the fact that the correlation is not perfect indicates that the two measures are not equivalent. Lower correlation in V1 compared to V4 indicates that the population code in V4 is more similar to the read-out of single neurons than

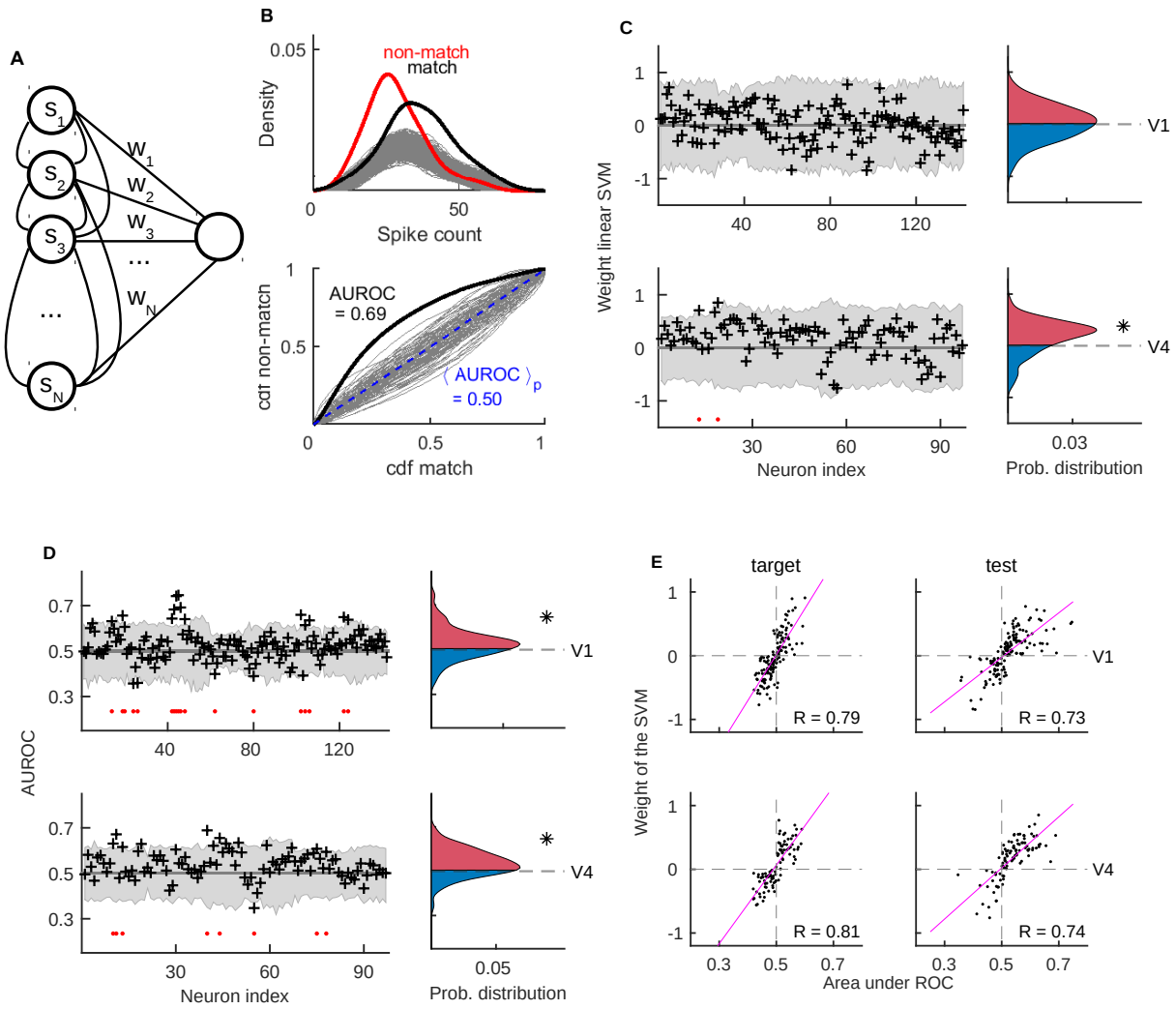

Figure 3. Decoding weights of single neurons are similar, but not identical to population decoding weights. **A)** Schema of the population model, depicting a population of projecting neurons and the read-out neuron. **B)** The ROC method utilizes probability distributions of spike counts in conditions "non-match" and "match" (top). Integration of the two distributions with a moving threshold and plotting the two integrals with respect to each other gives the ROC curve (bottom, black). Gray traces indicate probability distributions (top) and ROC curves (bottom) for the model with randomly permuted class labels for conditions "match" and "non-match". **C)** Left: Weights of the population model, collected across recording sessions (black) during test. The gray areas shows the distribution of weights of models with permuted class labels. Right: Population weights, pooled across neurons from all recording session. Asterisk marks significant deviation from 0 (permutation test, $nperm = 1000$). **D)** Left: AUROC scores (black crosses) in V1 (top) and V4 (bottom) during the test time window. The gray area indicates the distribution of results for models with permuted class labels ($nperm = 1000$). Significant neurons are marked with a red dot. Right: AUROC scores, pooled across neurons. The asterisk marks significant deviation of the distribution from 0.5 (permutation test, $nperm = 1000$). **E)** Scatter plot of population weights versus AUROC scores during target (left) and test (right). R marks the Pearson correlation coefficient and the magenta line is the least squares line.

in V1, where the population code has more of its own intrinsic structure.

In V4, but not in V1, correct assignment to the coding pool is sufficient to explain the classification performance of the population model.

We test the effect of the structure of the population code on the classification performance by removing the structure of responses in the population model and test whether such a perturbation has an impact on the classification performance. We perturb the data structure in two ways: 1.) the structure of responses is removed between all neurons from the same recording session, and 2.) the structure is removed only between neurons within the same pool (fig. 4A). The removal of the structure of responses is achieved by permuting neural indexes within the group of interest, without repetition and for each trial independently. This is equivalent to swapping the activity between neurons. The perturbation homogenizes the neural ensemble within the group of interest, in the sense that now all neurons within the group are equally informative for discrimination. We compute the classification performance of perturbed models and test it with respect to the performance of the regular model. Removing the structure across all neurons decreases the classification performance in both brain areas ($p < 0.001$, permutation test; fig. 4B). As the structure is removed only within pools, but the identity of pools is preserved, the performance of the perturbed model (type 2) in V1 is still lower than the performance of the regular model ($p < 0.001$), and the effect of the perturbation of the type 2 is similar to the effect of the perturbation of the type 1 (fig. 4B, top right). In V4, in contrast, the model with type 2 perturbation performs the same as the model with the intact structure (fig. 4B, bottom right), showing that the structure of responses within pools does not carry the information for classification. In V4, the knowledge about the identity of the pool is therefore sufficient to describe the population code, while in V1, the structure within coding pools also matters for classification.

In V4, but not in V1, informative neurons are more strongly coupled, synchronized and correlated than uninformative neurons.

In the following, we divide neurons in informative and uninformative. The criterion for division is the strength of the weight, $|\tilde{w}_n|$, the proxy of the importance of the activity of the neuron n for classification. First, we ask whether the activity of most informative neurons is sufficient to reproduce the classification performance with the entire population. We choose four neurons with the strongest weight and four neurons with the weakest weight in every recording session and run classification with linear SVM on each of these subnetworks, session by session. Not surprisingly, during test the classification performance of informative neurons is higher than the performance of uninformative neurons ($p < 0.001$ in both areas; fig. 5A). Interestingly, the subnetwork of informative neurons performs, on average, as good as the whole population ($BAC = 0.58$ in V1, $BAC = 0.58$ in V4). The number of neurons in the population therefore does not seem to be essential for the predictive power of the population model, and the latter rather seems to rely on the activity of a subset of the most informative neurons. The group of uninformative neurons performs at chance ($BAC = 0.50$ in V1, $BAC = 0.48$ in V4).

Next, we ask whether informative neurons differ from uninformative neurons in statistical properties of the single neurons as the absolute firing rate, the variance of spike counts and the strength of coupling to the population (see methods). During test in V4, informative neurons have higher firing rate ($p = 0.038$, permutation test with the permutation of neural indexes, 1000 permutations), stronger variance ($p = 0.007$) and stronger

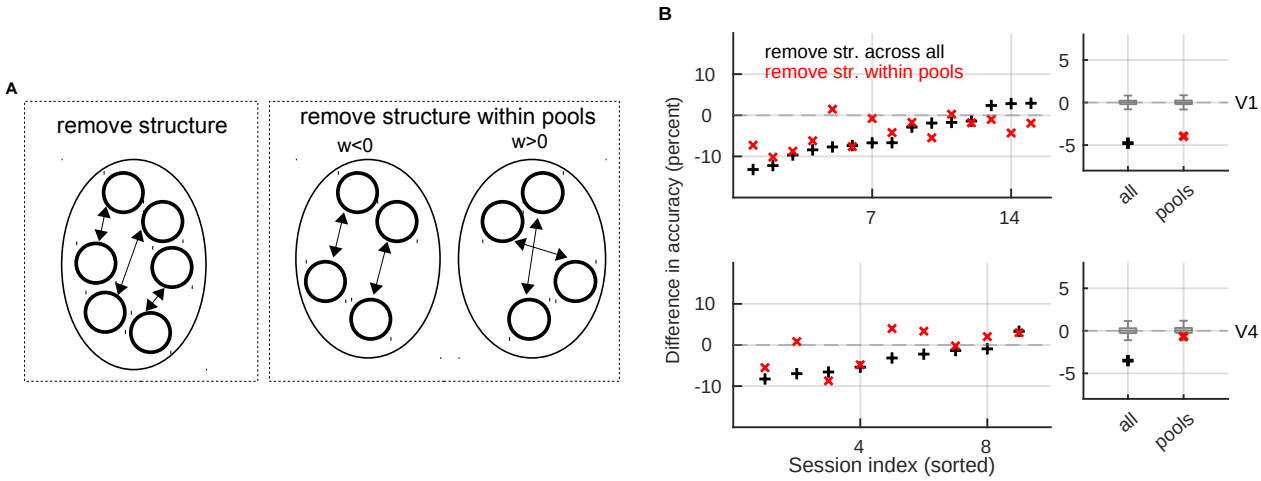

Figure 4. The effect of the removal of the structure of neural responses. A) Schema of the perturbation of the data structure of the type 1 (swap responses between all neurons, left) and of the type 2 (swap responses within the coding pool, right). **B)** Left: Classification results for perturbed models of the type one (black) and of the type two (red) during test in V1 (top) and in V4 (bottom). We plot the difference of the balanced accuracy between the regular model and the perturbed model. Right: Session-averaged results. Gray boxplots are for results of models with permuted class labels.

coupling to the population ($p = 0.035$, fig. 5B, bottom) than the uninformative neurons. During test in V1 (fig. 5B, top), as well as during target in both areas (supplementary fig. S3 A), there is no significant difference in single neuron statistics between the group of informative and uninformative neurons. As we compare the strength of coupling of informative neurons to their own subgroup (e.g., coupling of the informative neuron to other informative neurons), compared to coupling of uninformative neurons to their subgroup (fig. 5C), we find an important difference. During test in V4, but not in V1, informative neurons are more strongly coupled to their subgroup than uninformative neurons ($p = 0.023$, permutation test; fig. 5D). There is no difference in the strength of coupling across the two groups (e.g., informative to uninformative and vice-versa).

Finally, we ask whether informative and uninformative neurons differ in synchrony of spiking and correlations of spike counts. The synchrony of spiking is computed as the peak of the trial-specific (i.e., "noise") cross-correlation function ([18], see methods). Correlations of spike counts are computed as the correlations of the trial-to-trial variability and as correlations of binned spike trains in 50 ms bins (see methods). Interestingly, informative neurons are more strongly synchronized than uninformative neurons, but again only during test in V4 ($p = 0.0140$), while there is no effect in V1 (fig. 6A). Correlations of spike counts are also stronger between informative neurons in V4, but only in condition "match" ($p = 0.0058$ for correlated trial-to trial variability, $p = 0.0464$ for binned spike trains, $nperm = 1000$; fig. 6C) . In V1 (fig. 6D) and during the target time window (supplementary fig. S3 C-D), there is no difference in synchrony between groups of informative and uninformative neurons.

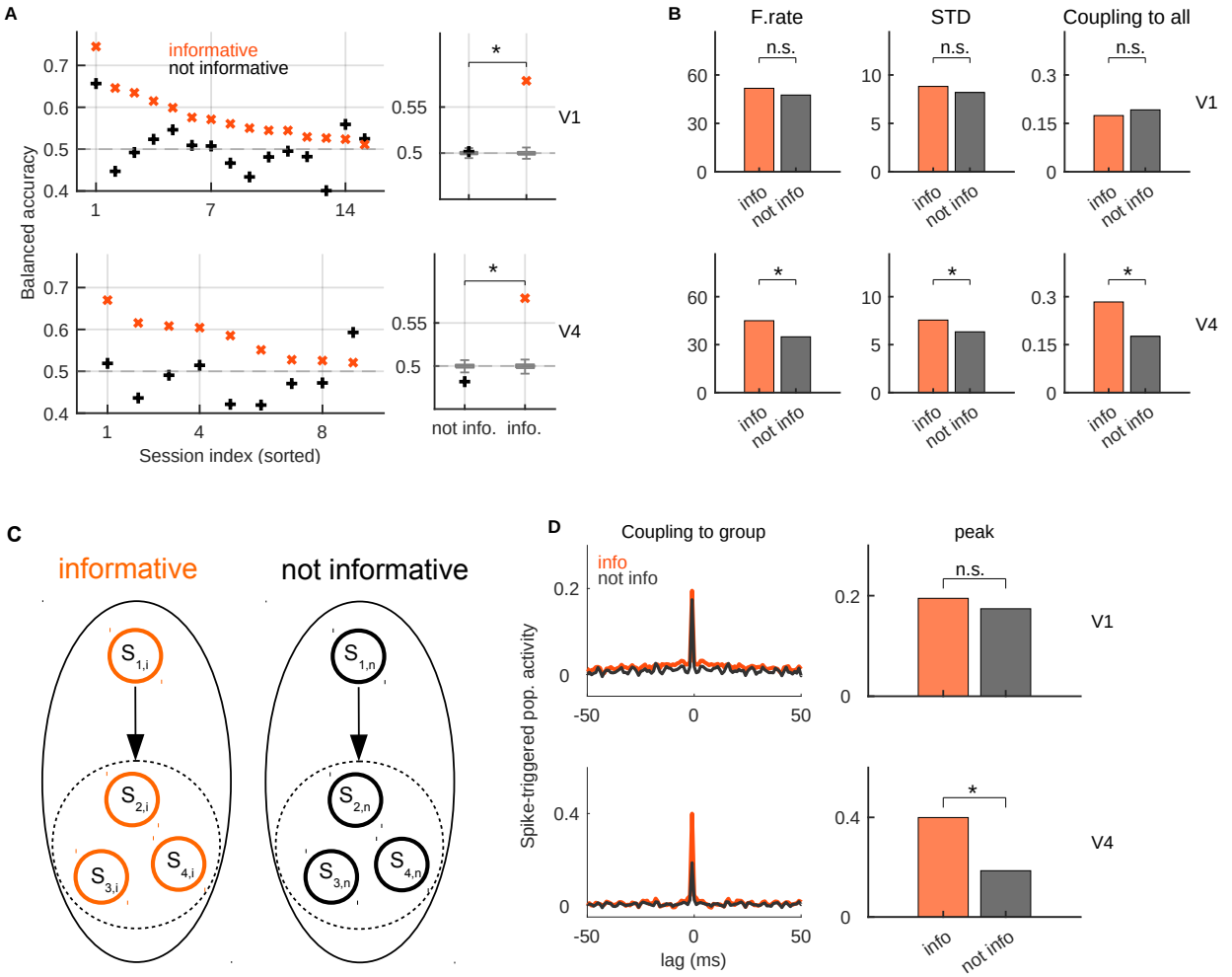

Figure 5. Informative neurons in V4, but not in V1, are more strongly coupled than uninformative neurons. A) Left: Balanced accuracy for the subnetwork of 4 informative neurons (orange) and uninformative neurons (black) during test in V1 (top) and in V4 (bottom). Results are sorted for informative neurons. Right: Session-averaged results and the distribution of results for models with permuted class labels (boxplots) **B)** The firing rate (left), STD of spike counts (middle) and coefficient of coupling to the population for informative (orange) and uninformative (dark gray) neurons. We show results in V1 (top) and in V4 (bottom) during test time window. **C)** Schema of coupling to own group for informative (orange) and uninformative (black) neurons. **D)** Left: Coupling function for coupling to own group between informative (orange) and uninformative (dark gray) neurons. The function is averaged across neurons. Right: Average peak of the coupling function.

In V4, but not in V1, correlations and spiking synchrony within the coding pool are stronger than across coding pools.

In the following, we distinguish neurons with negative weights, $\tilde{w}_n < 0$, from neurons with positive weights, $\tilde{w}_n > 0$. Notice that the sign of the weight corresponds to the preference of the neurons for one of the two stimulus classes and dividing neurons according to the sign of the weight, we obtain two mutually exclusive coding pools. As we decode conditions "match" and "non-match" with the group of neurons with positive weights (from now on, plus neurons) and with the neurons with negative weights (from now on, minus neurons),

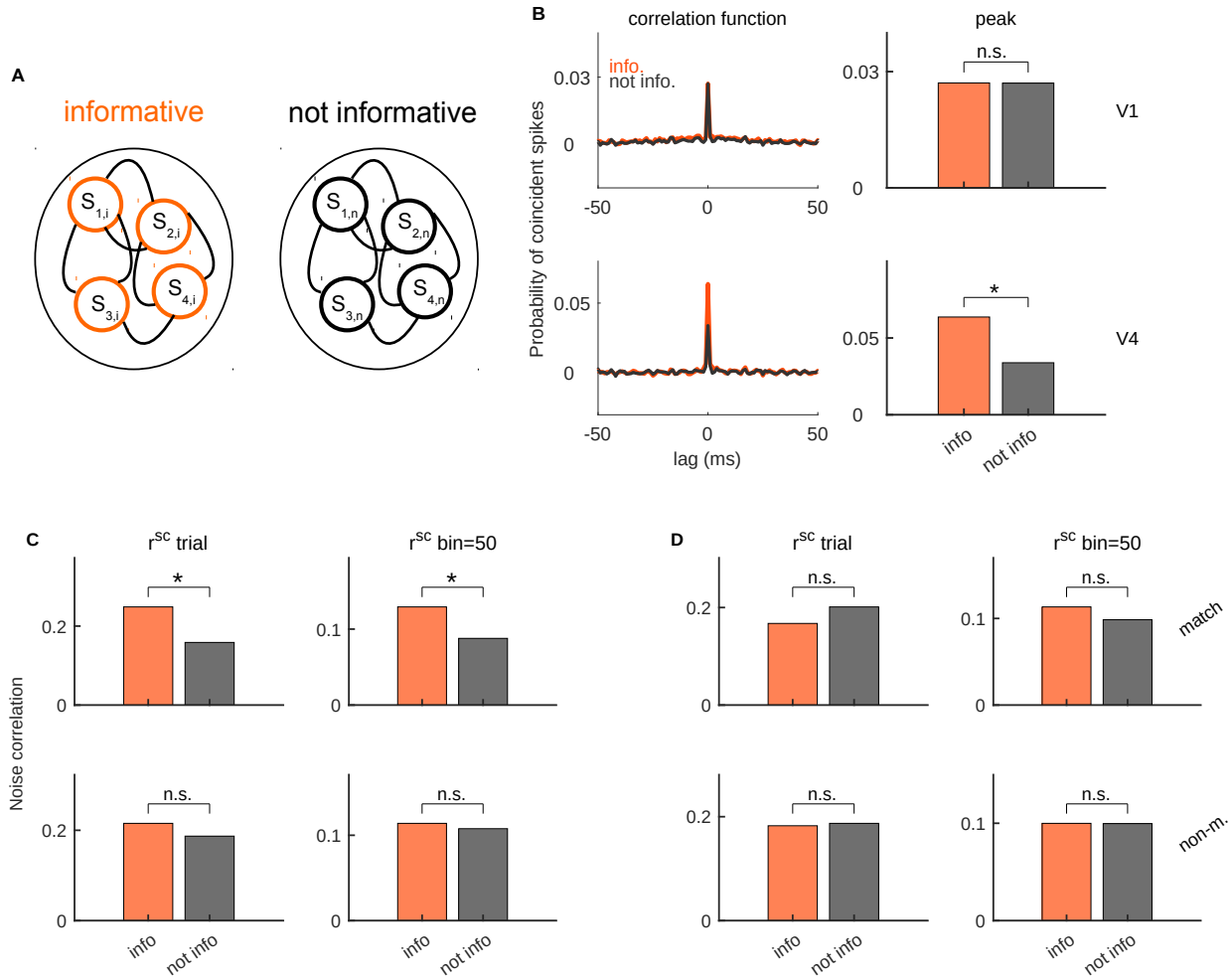

Figure 6. Informative neurons in V4, but not in V1, are more strongly synchronized and more strongly correlated than uninformative neurons. A) Schema of pair-wise interactions between informative (orange) and uninformative (black) neurons. **B)** Left: Average correlation function within groups of informative and uninformative neurons during test. Right: Peak of the correlation function. **C)** Noise correlations for informative and uninformative neurons in condition "match" (top) and "non-match" (bottom) during test in V4. **D)** Same as in **C)**, but for the test time window in V1.

we find that during test, plus neurons perform better than minus neurons in both brain areas ($p < 0.001$ in V1 and in V4, permutation test with randomly permuted class labels; fig. 7A). While in V1 both plus and minus neurons perform better than chance ($BAC = 0.53$ for minus neurons and $BAC = 0.57$ for plus neurons, $p < 0.001$ for both), in V4 only plus neurons perform better than chance ($BAC = 0.56$, $p < 0.001$), and minus neurons perform at chance ($BAC = 0.51$, non-significant). During target, both groups of neurons perform at chance (supplementary fig. S4 A).

We hypothesize that neurons within the same coding pool (plus or minus) might be more strongly coupled, synchronized and correlated (see [13] for a modeling study) than neurons from opposite coding pools. We measure the coupling strength within coding pools and across coding pools (fig. 7B) and compare the two. During the test time window in V4, the coupling strength is indeed significantly stronger within the pool compared to across pools ($p < 0.001$, permutation test with permutation of the neural index, $nperm = 1000$).

This is not the case in V1 (fig. 7C). Interestingly, during target, the coupling within pool is stronger than coupling across pools in both brain areas ($p < 0.001$ in V1, $p = 0.001$ in V4; fig. 7D).

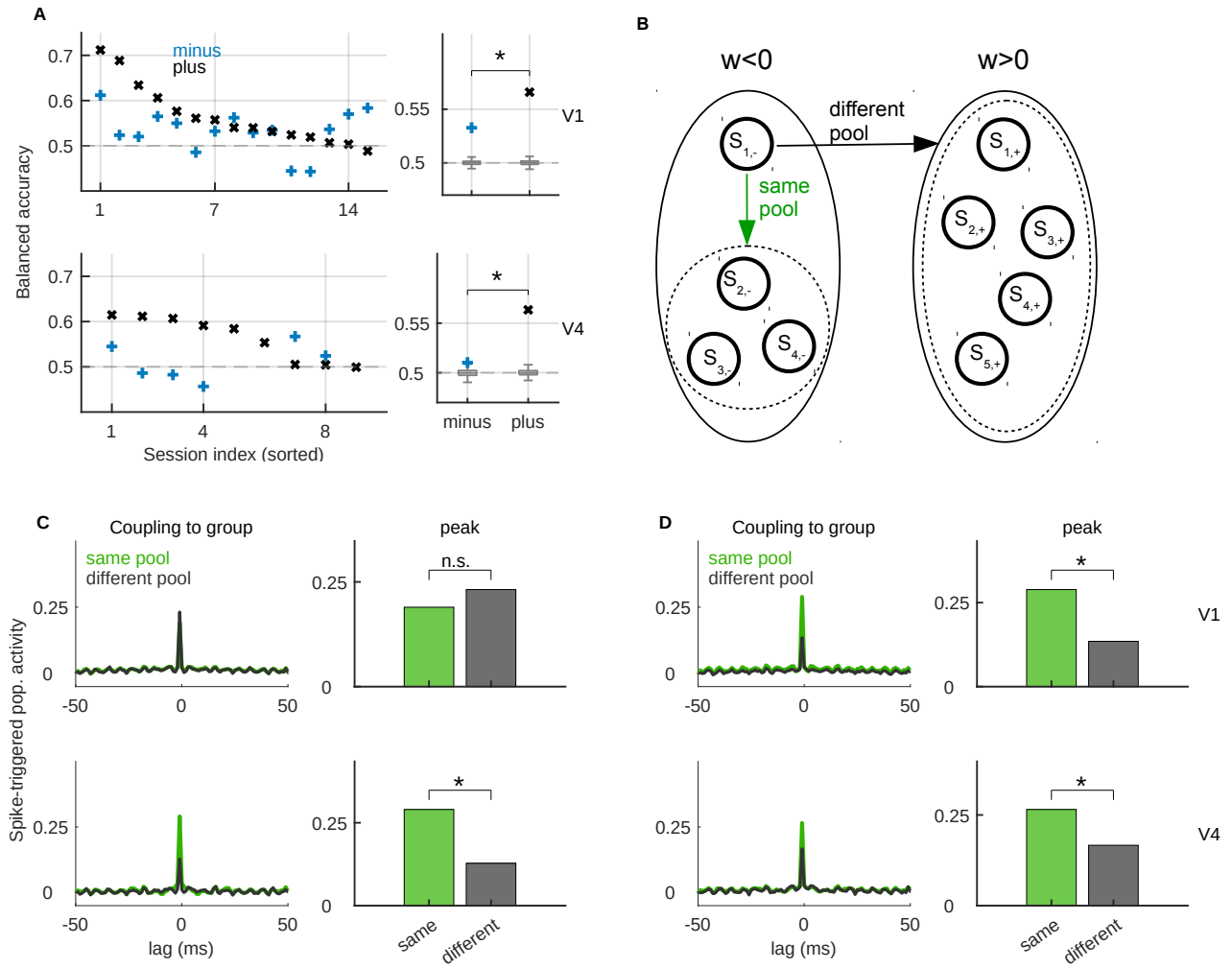

Figure 7. During target and during test in V4, neurons are more strongly coupled within pools compared to across pools. A) Left: Classification performance of plus (black) and minus (blue) subnetworks during test time window. Results are sorted for plus neurons. Right: Session-averaged results. **B)** Schema of coupling within pool (green arrow) and across pools (black arrow). The dashed line delimits the group of neurons that define the population activity (see methods). **C)** Left: Average coupling function within pool (green) and across pools (dark gray) during test. Right: Peak of the coupling function. **D)** Same as in **C)**, but for the target time window.

Next, we measure the pair-wise synchrony of spiking within and across coding pools (fig. 8A, green links are within coding pools and black links are across coding pools). During test, the synchrony of spiking within pool is stronger than across pools in V4 ($p = 0.001$, permutation test with randomized sign of the weight, see methods), but not in V1 (fig. 8A). Finally, we test the difference in noise correlations within and across groups. Noise correlations are measured as the correlation of the trial to trial variability of spike counts and as the correlation of binned spike counts in bins of 50 and 20 ms. Similarly to what we have reported for coupling and synchrony of spiking, during test, correlations are stronger within pool compared to across pools in V4

($p < 0.001$ for r_{trial} and $r_{bin,50}$, $p = 0.004$ for $r_{bin,20}$, permutation test with 1000 permutations; fig. 9A), but not in V1 (fig. 9B). During target, correlations within pool are stronger than across pools in both V4 ($p < 0.001$ for all three measures; fig. 9C) and in V1 ($p < 0.001$; fig. 9D). In summary, in target as well as during test in V4, pair-wise interactions on different time scales all reflect the functional separation into coding pools. Interactions are stronger within pools compared to across pools. This effect is absent during test in V1, where interactions within and across pools are of similar magnitude.

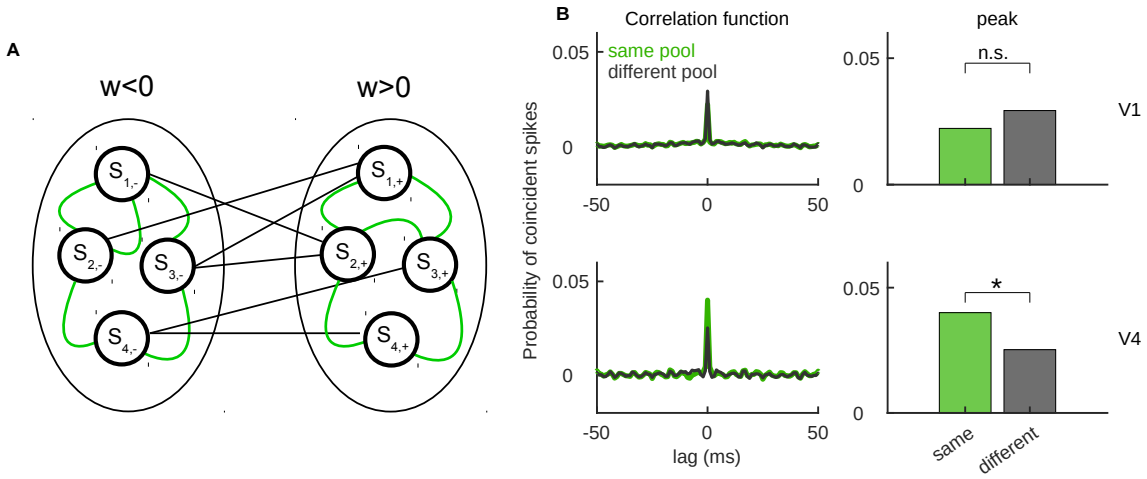

Figure 8. In V4, but not in V1, correlations within coding pools are stronger than across pools. A) Schema of two coding pools, one with negative (left) and the other with positive (right) weights. Green links are for pair-wise interactions within the pool and black are for interactions across pools. **B)** Left: Average correlation function during test for pairs of neurons within (green) and across pools (dark gray). We show results in V1 (top) and in V4 (bottom). Right: Peak of the correlation function, averaged across neurons.

In V4, correlations within pools are harmful for the decoder while the correlations across pools are not.

We test the effect of correlations on the classification performance by removing the correlation structure between (a specific group of) neurons from the same recording session and testing whether such a perturbation changes significantly the classification performance with respect to the regular model. The removal of the correlation structure is achieved by permuting the order of trials, independently for each neuron within the group of interest. We perturb the data in two ways: 1.) by removing all correlations and 2.) by removing correlations across the two coding pools but keeping correlations within the coding pools intact (fig. 10A). In V4, removal of correlations across all neurons increases the predictive power of the population model ($p < 0.001$, permutation test on the difference $BAC_{remove\ corr.} - BAC_{regular}$). As we keep correlations within coding pools and selectively remove correlations across the coding pools, we see no effect on the classification performance ($p = 0.527$). In V4, correlations within the same coding pool are therefore harmful for the decoder, while correlations across coding pools do not affect the coding performance (fig. 10B, bottom). In V1, both types of perturbation result in a small increase in performance ($p < 0.001$ for remove all, $p = 0.007$ for remove across pools; fig. 10B, top).

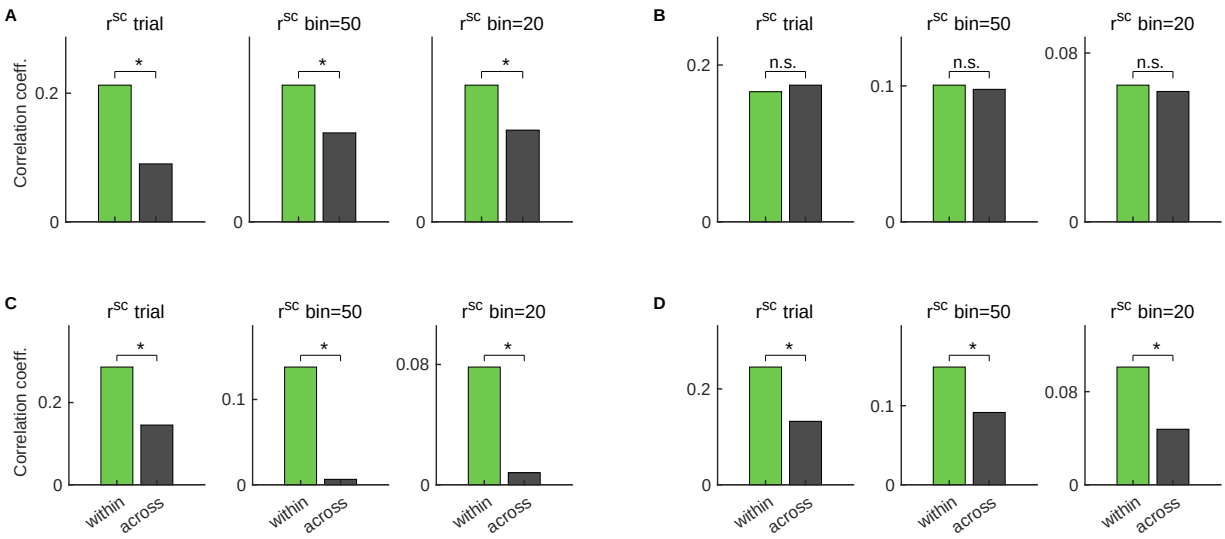

Figure 9. Correlations within the coding pool are stronger than across pools. A) Mean correlation coefficient within (green) and across coding pools (black) during test in V4. We show the correlation coefficient of the trial-to-trial variability (left), binned spike trains with 50 ms bin (middle), and 20 ms bin(right). **B)** Same as in **A)** but during test in V1. **C)** Same as in **A)** but during target in V4. **D)** Same as in **A)** but during target in V1.

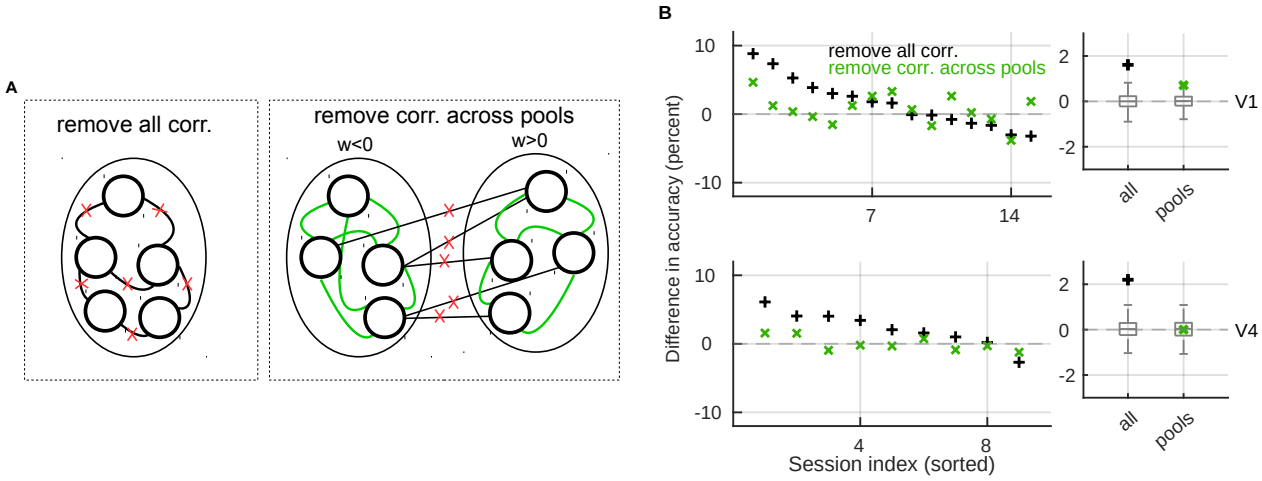

Figure 10. Removing correlations within but not across pools, increases the predictive power of the population model. A) Schema of removing correlations between all neurons (left) and of selectively removing correlations across coding pools, while correlations within pools are kept intact (right). **B)** Left: Difference in prediction accuracy between models with removed correlations and the regular model. We show the effect of removing all correlations (black) and of removing correlations across pools (green). Results are sorted for the perturbation of type 1. Right: Session-averaged results.

Discussion

We address the structure of the population code and the role of the inter-neuron correlation structure for discrimination of correct choices from the activity of neural ensembles in the early visual cortex. We show that

the population code is a better predictor of the correct choice behavior than an average single neuron. The population weights are strongly correlated with the area under the ROC curve, and therefore with discriminative capacity of single neurons. However, the population code cannot be fully accounted for by aggregating single neuron responses, since the removal of the structure of parallel spike counts decreases the predictive power of the decoder in both V1 and V4. In addition, we show that the discrimination capacity of the population code relies on a subset of most informative cells. In V4, but not in V1, the structure of the population code is strongly reflected in single neuron and pair-wise dynamics. Informative neurons are more strongly coupled, synchronized and correlated than uninformative neurons. As the population is split in two decoding pools according to the sign of the decoding weight, we show that in V4, but not in V1, pairs of neurons within the pool are more strongly coupled, synchronized and correlated than across pools. Finally, we show that correlations within pool are harmful for the decoder while correlation across pools are not.

We have shown that neural ensembles in V1 and V4 predict correct choices better than chance. While the choice probability has already been reported in V4 [19] as well as in V1 [20], previous studies have used artificial stimuli that specifically target filtering properties of observed neurons. We argue that our study adds an important piece of information by introducing naturalistic stimuli and a more complex behavioral task with classification into abstract classes. Present behavioral task requires to memorize the template image (the target stimulus) and compare it to a delayed sample (the test stimulus), and is expected to activate not only multiple brain areas through the visual ventral stream, but also the prefrontal cortex [21]. Moreover, stimuli are complex natural images that change from one trial to another, which prevents the discrimination to rely on the specific feature of the stimulus. In such a setting, it is even surprising that neural ensembles in V1 and V4 predict the correct choice behavior at all. Our suggestion is that the discriminative capacity of V1 and V4 ensembles comes from a top-down input, to reinforce the internal evidence for the choice.

It seems that neural circuits under investigation present a mixture of correlations, some of which are harmful for the decoder, and others that are not [22]. The reason for this might be to perform reasonably well in discrimination tasks, given the constraint of the noisy environment in which neurons operate [23]. We argue that in the present experimental task, the neural networks we study have two main functions, one is to collect the information about the stimuli and classify them in binary classes ("same" and "different"), and the other is to calculate a binary variable than can guide the choice behavior. While the code in an early sensory area such as V1 might be optimized for the information transfer, the quantity of transferred information might not need to be optimized in downstream areas, where the reliable transmission through correlated channels might be a priority. The repertoire of the choice behavior in the present setting is reduced to a binary variable (again, "same" and "different"), which only contains 1 bit of information. As correlations with the pool are stronger than across pools, they have as effect to amplify the signal within the pool and to help a reliable transmission of the signal. On a coarser spatial and temporal scale, it has been demonstrated that patterns of spontaneous activity reflect the functional architecture of the cortex [24, 25, 26], including the representation of abstract information, such as rules or goals [27]. Finding the same organization in V4 on a smaller spatial scale and during active classification of stimuli opens up the possibility that correlation structure also actively supports processing of stimuli in abstract categorization tasks.

Prediction accuracy of the population model is the best prediction that can be obtained from a linear classification model in this task, setting the higher bound on the information about the choice that can be extracted from spike counts of observed neural ensembles in V1 and in V4. We showed that the use of a

nonlinear SVM with RBF kernel does not improve the prediction accuracy. We also argue that the relatively low prediction accuracy of the population model might be importantly amplified as the coding function is exploited dynamically. If the coding function is applied on spike trains, e.g., with a leaky integrator, the difference between the read-out signals for conditions "match" and "non-match" can accumulate over time. In a related work, we constructed a dynamical model of the population read-out where a low-dimensional signal is computed from spike trains in single trials. We showed that low-dimensional signals, representing parallel spike trains in conditions "match" and "non-match" are significantly different and diverge over the time course of the trial (Koren et al., submitted).

It has to be pointed out that while our population model relies on a linear classifier, this linearity should not be understood as the linearity between responses of single neurons and the population. On the contrary, we show that single neurons cannot account for the structure and performance of the population code. In the biological setting, population weights have a straightforward interpretation as synaptic weights between a population of projecting neurons and a read-out neuron. The sign of the weight would correspond to strengthening (positive weight) or weakening (negative weight) of the synaptic weight between the projecting neuron and the read-out neuron with respect to the baseline. In biological networks, we have that the contributions of projecting neurons sum at the synapse and form the synaptic input. What the linearity of the population model does imply is that synaptic inputs sum linearly at the synapse as they create the synaptic current at the read-out neuron, a well-established experimental fact, and common use in modeling [28]. While the transfer function of the read-out neuron might not be linear, (but be better described by, e.g., a sigmoid) and might be subjected to non-linearity from the spike-triggered adaptation, voltage-gated currents, etc., this can be easily considered while modeling present results, but is beyond the scope of the present work. If population weights have been learned through a form of Hebbian synaptic plasticity [29], the formation of coding pools can be explained by the common top-down input. Nevertheless, in V1 this process is likely to be counteracted by another process, that aims at decorrelating neural activities in order to maximize the information transfer [12]. How do the two work together is at present unclear and could be the topic of future investigation.

Methods

Ethics statement. All experiments were performed in accordance with protocols approved by the Animal Welfare Committee (AWC) and the Institutional Animal Care and Use Committee (IACUC) for the University of Texas Health Science Center at Houston (UTHealth).

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 procedure

The trial started after 300 ms of successful fixation within the fixation area and consisted in visualizing the target and the test stimuli, with a delay period in between. Stimuli were shown for 300 ms each, while delay period had a random duration between 800 and 1000 ms, in order to prevent the expectation of the timing of the test stimulus. The target and the test stimuli, naturalistic images in black and white, were either identical (condition "match") or else the test stimulus was rotated with respect to the target stimulus (condition "non-match"). The task of the animal was to decide about the similarity/difference of the target and the test stimuli by holding a bar for "different" and releasing the bar for "same". The subject had to respond within 200 and 1200 ms from the offset of the test stimulus, otherwise the trial was discarded. The difficulty of the task was calibrated on-line, in such away as to have, on average, 70 percent of correct responses on non-matching stimuli. The difference in orientation of the test stimulus ranged between 3 and 10 degrees.

In every recording session, two laminar electrodes with 16 recording channels each were inserted in V1 and V4 areas, measuring the multi-unit signal with 16 recording channels (0.1 mm spacing between adjacent contacts). Electrodes were inserted perpendicularly to the cortical surface, and calibrated such that neurons from the two areas had overlapping receptive fields. Since recording sessions were performed on different days, it is extremely unlikely that the electrode captured same neurons in different recording sessions. We therefore treated spike-sorted units from different recording sessions as distinct cells. The analysis included all cells that responded to the stimulus with a 4-fold increase of the firing rate with respect to the baseline. Since we are interested in decoding population activities, we used recording sessions with at least 6 simultaneously recorded units. This amounts to 142 neurons in V1, collected in 15 recording sessions, and to 97 neurons in V4, collected in 9 recording sessions.

Classification methods and computation of decoding weights

The following analysis was done with Matlab, Mathworks, version R2017b. The spike train of a single neuron is a binary 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.

Area under the ROC curve for univariate analysis. The first decoding method consisted in evaluating the discriminability of conditions "match" and "non-match" from spike counts of single neurons, using the receiver operating characteristic (ROC) analysis [21, 3, 1, 4, 2]. The ROC is a non-parametric method that evaluates the distance between distributions of spike counts in conditions "match" and "non-match", collected in multiple trials. The spike count of the neuron n in trial j is defined as the sum of spikes in the desired time window: $s_{n,j} = \sum_{k=1}^{K} o_{n,j}(t_k)$.

Spike counts are collected across trials and define two probability distributions, corresponding to conditions "match", $P_M(s)$, and "non-match", $P_{NM}(s)$, for $s \in \{0, \Delta s, .., s_{max}\}$, where s_{max} is the maximal spike count across all recorded activities, $\Delta s = 2s_{max}/(nstep - 1)$, $nstep = 200$. Probability distributions are convolved with a normalized Gaussian kernel,

$$\begin{aligned} \tilde{P}_{NM}(s) &:= \sum_{x \in X} P_{NM}(s - x)u(x), \\ \tilde{P}_M(s) &:= \sum_{x \in X} P_M(s - x)u(x), \end{aligned} \tag{2}$$

with $u(x) = \left(\sum_{x \in X} \exp(-\frac{x^2}{2\sigma_u{}^2}\right)^{-1} \exp(-\frac{x^2}{2\sigma_u{}^2})$. The width of the kernel σ_u is defined as follows: first, an estimate for the optimal width of each distribution is computed using Silverman's rule, $\sigma^*_{M,NM} = \sigma_{M,NM}(\frac{4}{3J_{M,NM}})^{\frac{1}{5}}$, where σ_M (σ_{NM}) is the standard deviation of the distribution of spike counts in condition "match" ("non-match") and J_M (J_{NM}) is the number of trials in condition "match" ("non-match"). Then, we set $\sigma_u = \frac{\sigma^*_M + \sigma^*_{NM}}{2}$.

We define empirical cumulative distribution functions, $D_M(s)$ and $D_{NM}(s)$, as follows:

$$D_{M,NM}(s) := \sum_{s'=0}^{s} \tilde{P}_{M,NM}(s')\Delta s \tag{3}$$

Note that the cumulative distribution functions are bounded, $0 \leq D_{M,NM}(s) \geq 1$, and the area under the ROC curve is the following sum:

$$AUROC = \frac{1}{2} \sum_{s \in S} (D_{NM}(s+1) + D_{NM}(s))(D_M(s+1) - D_M(s)) \tag{4}$$

Two identical distributions give the $AUROC$ of 0.5, at chance prediction of conditions "match" and "non-match" from the observation of spike counts. In order to estimate if the prediction is significantly better than chance, we have to take into account the variance of the data. This is done with the permutation test. We pool the input statistics ($s_{n,j}$) from the two conditions and randomly assign labels "match" and "non-match" to every trial. The pooled distribution is then split into two subsets, with the number of samples in subsets corresponding to the number of original samples in conditions "match" and "non-match". Using these two random subsamples, the area under the curve is then computed by eq. 2 - 4. This procedure is repeated $nperm = 1000$ times, and results in a distribution of areas under the ROC curve, $AUROC_p,\ p = 1, ..., n_{perm}$. We compute the fraction of $AUROC_p$ that are larger (p_L) and smaller (p_S) than the true $AUROC$,

$$\begin{aligned} p_L &= \frac{1}{n_{perm}} \sum_{p=1}^{n_{perm}} \mathbb{1}(AUROC_p > AUROC) \\ p_S &= \frac{1}{n_{perm}} \sum_{p=1}^{n_{perm}} \mathbb{1}(AUROC_p < AUROC) \end{aligned} \tag{5}$$

where $\mathbb{1}(x > y)$ is the following logical function:

$$\mathbb{1}(x > y) = \begin{cases} 1 & \text{if } x > y \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

The $AUROC$ is considered to be significantly smaller (larger) than 0.5 if the following is true: $p_L \leq {}^{0.05}/_{N_{tot}}$ ($p_S \geq {}^{0.05}/_{N_{tot}}$), where N_{tot} is the total number of neurons in a given brain area.

Finally, we are interested in the distribution of $AUROC$ across neurons. We pool $AUROC$ across neurons from all recording sessions and from the two animal subjects and get the following distribution: $AUROC_n, n = 1, ..., N_{tot}$. We compute the mean of the distribution, $\langle AUROC \rangle_n$, and test its significance by comparing it with the distribution of averages from the permutation procedure, $\langle AUROC_p \rangle_n,\ p = 1, ..., n_{perm}$. The p-value is then computed by eq. 5-6, and corrected for multiple testing with Bonferroni correction.

Linear Support Vector Machine for multivariate analysis. The second read-out method consists in discriminating conditions "match" and "non-match" on high-dimensional activity profiles with linear Support vector machine (SVM). We use z-scored spike counts,

$$\tilde{s}_{n,j} = \frac{s_{n,j} - \langle s_{n,j} \rangle_j}{\sqrt{\mathrm{Var}_j\ (s_{n,j})}}. \tag{7}$$

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 activity vector of N simultaneously recorded neurons in trial j,

$$\tilde{\mathbf{s}}_j = [\tilde{s}_{1,j}, \tilde{s}_{2,j}, ..., \tilde{s}_{N,j}]. \tag{8}$$

The optimization problem of the linear SVM, in its primal form, is expressed with a Lagrangian [15],

$$L_p = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{j=1}^{J} \lambda_j [y_j(\mathbf{w}^T\tilde{\mathbf{s}}_j + b) - 1], \tag{9}$$

where $\mathbf{w}$ is $(N, 1)$-dimensional vector of weights and N is the number of simultaneously active neurons, b is the offset of the separating hyperplane from the origin, λ_j is the Lagrange multiplier ($\lambda_j \leq 0\ \forall j$) and y_j is the class label in trial j, with $y_j \in \{-1, 1\}$.

The model is trained on 80 % of trials, and tested on the remaining 20 %. We perform Monte-Carlo cross-validation for random splits in the training and test set with 100 iterations. The classification performance on the test set is computed with balanced accuracy,

$$BAC = \frac{TP}{2(TP + FN)} + \frac{TN}{2(TN + FP)}, \tag{10}$$

where TP, FN, TN and FP stand for the number of true positive, false negative, true negative and false positive classifications on the test set, respectively. We call a true positive the correct classification of condition "match", a true negative the correct classification of condition "non-match", etc. Reported balanced accuracy is averaged across cross-validations. The regularization parameter of the SVM is estimated with 10-fold cross-validation on the training data, and chosen from the following range: $C \in \{0.0012, 0.00135, 0.0015, 0.002, 0.005, 0.01, 0.05, 0.1, 0.5\}$.

If the classifier performs at chance, its balanced accuracy is 0.5. The significance of the deviation of balanced accuracy from 0.5 is determined with permutation test. We train models with randomly permuted class labels for "match" and "non-match". Iterating the permutation procedure gives a distribution of classification performances, $BAC_p,\ p = 1, ..., n_{perm}$, with $n_{perm} = 1000$. The p-value of the permutation test is the probability of models with permutation of having higher balanced accuracy than the true model.

$$p_{svm} = \frac{1}{n_{perm}} \sum_{p=1}^{n_{perm}} \mathbb{1}(BAC_p > BAC) \tag{11}$$

If $p_{svm} < \frac{0.05}{N_{test}}$, the balanced accuracy is considered to be significantly higher than chance, where $N_{test} = 4$ stands for the Bonferroni correction for multiple testing.

In order to compare the predictive power of the high-dimensional model with single neurons, we also compute the SVM on the single neuron data. The classification procedure is, in this case, done for each neuron independently. Balanced accuracy is then averaged across neurons from the same recording session.

Extraction of decoding weights. The minimization of the Lagrangian (eq. 9) allows to compute the vector of weights,

$$\mathbf{w} = \sum_{q=1}^{Q} \lambda_q y_q \mathbf{s}_q, \tag{12}$$

where Q is the number of support vectors [15]. The range of the weight vector depends on the regularization parameter of the SVM (C parameter) and is therefore different from one session to another. Since we combine results from different recording sessions, we normalize the weight vector with L^2 norm,

$$\tilde{\mathbf{w}} = \frac{\mathbf{w}}{||\mathbf{w}||} \tag{13}$$

with $||\mathbf{w}|| = \sqrt{w_1^2 + ... + w_N^2}$. The vector of decoding weights, $\tilde{\mathbf{w}} = [\tilde{w}_1, \tilde{w}_2, ..., \tilde{w}_N]$, associates the activity of each neuron with it's role for classification in the N-dimensional space of inputs.

The distance of weights is defined as the Euclidean distance of weights between pairs of neurons from the same recording session,

$$d_{n,m} = |\tilde{w}_n - \tilde{w}_m| \tag{14}$$

with $n, m = 1, ..., N$. Results are then pooled across recording sessions.

Coupling of single neurons with the population

We measure the strength of the coupling of single neurons with the population using a measure of spike-triggered population activity, similarly to the method in [10, 30]. We consider the spike train of a selected neuron n^* in trial j. Such a spike train can be expressed as a sum of Kronecker delta functions,

$$o_{n^*,j}(t_k) = \sum_{i=1}^{\#n*} \delta_{t_i}(t_k) \tag{15}$$

where t_i is the spike time of the $i-th$ spike and $\#n*$ is the number of spikes of neuron $n*$ in trial j. The population activity in trial j is the mean of spikes across neurons from the population, excluding the neuron

n^*:

$$\beta_j(t_k) = \frac{1}{N-1} \sum_{\substack{n=1 \\ n \neq n^*}}^{N} o_{n,j}(t_k). \tag{16}$$

For each spike of neuron n^*, we sum the population activity in a time window of 100 milliseconds before and after the spike:

$$\gamma_j(\tau) = \sum_{i=1}^{\#n*} \sum_{\tau=-100}^{100} \beta_j(t_k - \tau)\delta_{t_i}(t_k)\Delta t_k, \tag{17}$$

where Δt_k is the time step of 1 ms. The coupling function $\gamma_j(\tau)$ measures the population activity as a function of the time delay of the spike of neuron $n*$. Note that $\tau = 0$ corresponds to the coupling at the time of the spike. The coupling function $\gamma_j(\tau)$ depends on the number of spikes of the neuron $n*$, and we make it independent of the number of spikes by subtracting the trial-invariant coupling. The trial-invariant coupling function is computed by utilizing the spike train of the selected neuron and the population activity that are not from the same trial, e.g., we use $o_{n^*,j'}(t_k)$ and $\beta_j(t_k)$, where $j' \neq j$. The trial index for the former is determined with random permutation of the order of trials. The random permutation is repeated $nperm = 20$ times and the coupling function is averaged across repetitions, resulting in $\gamma_j^{perm}(\tau)$. Finally, we obtain the trial-specific (or "noise") coupling function by subtracting the trial-invariant from the original coupling function, and averaging across trials,

$$\gamma^{noise}(\tau) = \frac{1}{J} \sum_{j=1}^{J} (\gamma_j(\tau) - \gamma_j^{perm}(\tau)). \tag{18}$$

The coupling coefficient is defined as the coupling function at zero time lag, $\Gamma = \gamma^{noise}(0)$.

Measures of pair-wise interactions

Pair-wise measures are computed for pairs of neurons from the same recording session and are then pooled across sessions.

Correlation of spike count trial-to-trial variability. We use z-scored spike counts, $\tilde{s}_{n,j}$. For each neuron, we define a vector of activation across trials, $\tilde{\mathbf{s}}_n = [\tilde{s}_{n,1}, \tilde{s}_{n,2}, ..., \tilde{s}_{n,J}]$. The correlation coefficient between activation of a pair of neurons n and m is computed as the Pearson correlation coefficient,

$$\begin{aligned} r_{n,m}^{trial} &= \frac{\mathbb{E}[\tilde{\mathbf{s}}_n, \tilde{\mathbf{s}}_m] - \mathbb{E}[\tilde{\mathbf{s}}_n]\mathbb{E}[\tilde{\mathbf{s}}_m]}{\sqrt{\mathrm{Var}(\tilde{\mathbf{s}}_n)\mathrm{Var}(\tilde{\mathbf{s}}_m)}} \\ &= \mathbb{E}[\tilde{\mathbf{s}}_n, \tilde{\mathbf{s}}_m] \end{aligned} \tag{19}$$

where the second equality is due to z-scoring [18].

Correlation of spike counts with binned spike trains. We bin spike trains in bins of length $L = 20$ ms ($L = 50$ ms) and count spikes in bins. Spike counts are z-scored for each bin independently,

$$\tilde{sc}_{n,j}(b) = \frac{sc_{n,j}(b) - \langle sc_{n,j}(b) \rangle_j}{\sqrt{\mathrm{Var}_j\ (sc_{n,j}(b))}}, \tag{20}$$

where b is the index of the bin, $b = 1, 2, ..., B$, and the mean and variance are computed across trials. We define the vector of activation that concatenates binned spike trains across trials,

$\tilde{\mathbf{sc}}_n = [\tilde{sc}_{n,1}(b_1), \tilde{sc}_{n,1}(b_2), ..., \tilde{sc}_{n,1}(B), \tilde{sc}_{n,2}(b_1), \tilde{sc}_{n,2}(b_2)..., \tilde{sc}_{n,J}(B)]$. After this procedure is repeated for all neurons, we compute Pearson's correlation coefficient between activations for each pair of neurons n and m,

$$r_{n,m}^{binned} = \frac{\mathbb{E}[\tilde{\mathbf{sc}}_n, \tilde{\mathbf{sc}}_m] - \mathbb{E}[\tilde{\mathbf{sc}}_n]\mathbb{E}[\tilde{\mathbf{sc}}_m]}{\sqrt{Var(\tilde{\mathbf{sc}}_n)Var(\tilde{\mathbf{sc}}_m)}} \tag{21}$$

Synchrony Now, let $n, m \in \{1, ..., N\}$ be fixed, and we define the spike trains of neurons n and m in trial j by $f_j(t_k) := o_{n,j}(t_k)$ and $g_j(t_k) := o_{m,j}(t_k)$. The cross-correlation function measures the co-occurence of spikes,

$$ccg_j(\tau) = \frac{1}{2K} \sum_{\tau=-K}^{K} f_j(t_k) g_j(t_k + \tau) \Delta t_k. \tag{22}$$

We also measure autocorrelation functions for neurons n and m. The autocorrelation function for neuron n is the following:

$$acg_{1,j}(\tau) = \frac{1}{2K} \sum_{\tau=-K}^{K} f_j(t_k) f_j(t_k + \tau) \Delta t_k. \tag{23}$$

The cross-correlation function depends on the number of spikes of neurons n and m. To make it independent on the number of spikes, we normalize it with autocorrelations at zero time lag,

$$ccg_j^{norm}(\tau) = \frac{ccg_j(\tau)}{acg_{1,j}(0) acg_{2,j}(0)}. \tag{24}$$

It is expected that an important part of the correlation is generated by the common input to the two neurons [18]. The correlation due to the common input is captured by the trial-invariant correlation function. The latter is computed from spike trains that are not from the same trial,

$$ccg_{j,p}^{perm}(\tau) = \frac{1}{2K} \sum_{\tau=-K}^{K} f_{j'}(t_k) g_j(t_k + \tau) \Delta t_k, \ j' \neq j, \tag{25}$$

where p is the permutation index and j' is defined with random permutation of the order of trials for the neuron n, while the neuron m keeps the correct order of trials. The permutation procedure is iterated $nperm = 100$-times, and we average the resulting functions across permutations, to get $ccg_j^{perm}(\tau)$. We capture the trial-specific correlation (i.e., "noise" correlation) by subtracting the trial-invariant correlation from the overall correlation, and average across trials:

$$ccg^{noise}(\tau) = \frac{1}{J} \sum_{j=1}^{J} (ccg_j^{norm}(\tau) - ccg_j^{perm}(\tau)), \tag{26}$$

where J is the number of trials. The coefficient of synchrony is defined as the trial-specific cross-correlation function at zero time lag, $C = ccg^{noise}(0)$.

References

1. Newsome WT, Britten KH, Movshon JA: Neuronal correlates of a perceptual decision. Nature 1989, 341:52-54.

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

3. Britten KH, Newsome WT, Shadlen MN, Celebrini S, Movshon JA: A relationship between behavioral choice and the visual responses of neurons in macaque MT. Visual Neurosci. 1996, 13(1), 87-100.

4. Shadlen, M. N., & Newsome, W. T. (2001). Neural basis of a perceptual decision in the parietal cortex (area LIP) of the rhesus monkey. Journal of neurophysiology, 86(4), 1916-1936.

5. Rao, R. P., & Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. Nature neuroscience, 2(1), 79.

6. Elsayed, G. F., & Cunningham, J. P. (2017). Structure in neural population recordings: an expected byproduct of simpler phenomena?. Nature neuroscience, 20(9), 1310.

7. Britten KH, Shadlen MN, Newsome WT, Movshon JA, The analysis of visual motion: a comparison of neuronal and psychophysical performance, J. Neurosci 1992;12:4745.

8. Cohen, M. R., & Kohn, A. (2011). Measuring and interpreting neuronal correlations. Nature neuroscience, 14(7), 811.

9. Smith MA, Kohn A: Spatial and temporal scales of neuronal correlation in primary visual cortex. J Neurosci. 2008, 28(48), 12591-603.

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

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

12. Averbeck, B. B., Latham, P. E., & Pouget, A. (2006). Neural correlations, population coding and computation. Nature reviews neuroscience, 7(5), 358-366.

13. Shadlen MN: A computational analysis of the relation between neuronal and behavioral responses to visual motion, J Neurosci. 1996, 1486-1510

14. Nienborg HR, Cohen MR, Cumming BG: Decision-related activity in sensory neurons: Correlations among neurons and with behavior, Annu. Rev. Neurosci. 2012, 35, 46383.

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

16. Panzeri, S., Harvey, C. D., Piasini, E., Latham, P. E., & Fellin, T. (2017). Cracking the neural code for sensory perception by combining statistics, intervention, and behavior. Neuron, 93(3), 491-507.

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

18. Bair, W., Zohary, E., & Newsome, W. T. (2001). Correlated firing in macaque visual area MT: time scales and relationship to behavior. Journal of Neuroscience, 21(5), 1676-1697.

19. Shiozaki, H. M., Tanabe, S., Doi, T., Fujita, I. (2012). Neural activity in cortical area V4 underlies fine disparity discrimination. The Journal of Neuroscience, 32(11), 3830-3841.

20. Nienborg H, Cumming BG: Decision-related activity in sensory neurons may depend on the columnar architecture of cerebral cortex, J Neurosci. 2014 Mar 5;34(10):3579-85

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

22. Moreno-Bote, R., Beck, J., Kanitscheider, I., Pitkow, X., Latham, P., & Pouget, A. (2014). Information-limiting correlations. Nature neuroscience, 17(10), 1410.

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

24. Arieli, A., Sterkin, A., Grinvald, A., & Aertsen, A. D. (1996). Dynamics of ongoing activity: explanation of the large variability in evoked cortical responses. Science, 273(5283), 1868-1871.

25. Tsodyks, M., Kenet, T., Grinvald, A., & Arieli, A. (1999). Linking spontaneous activity of single cortical neurons and the underlying functional architecture. Science, 286(5446), 1943-1946.

26. Fox, M. D., & Raichle, M. E. (2007). Spontaneous fluctuations in brain activity observed with functional magnetic resonance imaging. Nature reviews neuroscience, 8(9), 700.

27. Waskom, M. L., & Wagner, A. D. (2017). Distributed representation of context by intrinsic subnetworks in prefrontal cortex. Proceedings of the National Academy of Sciences, 114(8), 2030-2035.

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

29. Abbott, L. F., & Nelson, S. B. (2000). Synaptic plasticity: taming the beast. Nature neuroscience, 3(11s), 1178.

30. Luczak, A., Bartho, P., & Harris, K. D. (2013). Gating of sensory input by spontaneous cortical activity. Journal of Neuroscience, 33(4), 1684-1695.

31. Hansen, B. J., Chelaru, M. I., & Dragoi, V. (2012). Correlated variability in laminar cortical circuits. Neuron, 76(3), 590-602.