Supporting Information

Jin et al. 10.1073/pnas.0909881106

SI Text

Materials and Methods. ***Behavior and electrophysiology.*** Each of two monkeys was fitted with a head post, a recording chamber whose position was determined after MRI analysis, an eye coil and, intermittently, digastric electromyographic (EMG) electrodes that had been surgically implanted according to National Institutes of Health and Massachusetts Institute of Technology guidelines for animal experimentation (1). Eye position was recorded with conventional search coil techniques (2), and eye and licking movements were sampled at 1 kHz. The monkeys were trained to make saccades in response to visual targets presented sequentially on a computer screen in front of them. The screen displayed a 9×9 grid of gray potential visual targets spaced at 5° of visual angle. In each trial of the standard task (the RSQ4 task), the monkey had to fixate for 1 sec a central red fixation spot and then was required to make saccades to a series of 0.8° red target spots that were illuminated at 400-msec intervals in up, down, right, or left directions at adjacent points on the grid. Four targets were chosen in pairs (for example, right followed by up or down, up followed by left or right) so that targets never appeared at the edges of the 9×9 target display. Each target remained illuminated for 400 msec and then was extinguished when the next target turned red. The monkey was required to acquire the targets with an eye position accuracy of 2°. If the monkey performed correctly, then, after a randomly varied 400- to 800-msec delay, it received a drop of water or juice delivered from a spout in front of the monkey's mouth. If the monkey broke fixation or did not respond to a target accurately within 400 ms, the trial was terminated and was treated as an error trial. The intertrial interval (ITI) was 1.5 sec for both correct and error trials. Trials of a given condition were presented in blocks of 30–40 trials. In the standard RSQ4 task, different sequences were presented in pseudorandom order. These blocks of trials predominated in the ≈800 trials per day recording sessions. Other sequences with fixed sequences of target presentation, other target presentation intervals (600- or 800-msec intervals) or randomly varying target intervals (400, 600, or 800 msec), or randomly varying target intervals and randomly varying directions, were given, along with some blocks in which the amount or timing of reward varied (3–4). These are referred to as non-RSQ4 tasks.

Multiple (up to 24) tungsten electrodes (1–2 MOhm) were implanted in the prefrontal cortex and caudate nucleus bilaterally. Electrodes were implanted chronically and used for up to 1–3 months. Recordings were made on average for five sessions per week. Neural activity was acquired at a sampling rate of 32 kHz. Spike activity was sorted into clusters by Autocut under manual control and was analyzed with custom software (3–4).

Peri-event time histograms. The spikes of each single unit accepted for initial analysis were analyzed for each recording session. Spikes were aligned with fixation onset. The time span of −100 msec to 5,000 msec relative to fixation onset was divided into 10-msec bins, and spikes in each bin were summed across the multiple trials of RSQ4 blocks. The firing rate at a bin was obtained by dividing the number of spikes in the bin by 10 ms, which produced the peri-event time histograms (PETH) of the single unit aligned at the fixation onset. We also aligned spikes with other events, including task start, the onset of each target (Go signal), saccade onsets, offset of the last saccade target, and reward delivery. Corresponding PETHs were similarly constructed.

Criteria for selecting single units. For some single units, the maximum firing rate in the PETH of the RSQ4 block of trials was <2 Hz. We excluded these units from the analysis. Some single units had gradual drifts of trial-by-trial firing rates during the recording sessions. These drifts could have resulted from unstable recording, or from the effects of changes in attention, adaptation or learning. We excluded single units with large variations in trial-by-trial spike rates, based on the van der Waerden normal-scores test for k independent samples (5). This statistical procedure tests whether samples are drawn from the same underlying distribution. For each single unit, there were four to five blocks of the standard RSQ4 task presented at different times during the session. For all of the RSQ4 trials in a session, we calculated the average spike rates during the fixation period. We then tested the hypothesis that the firing rates in the different RSQ4 blocks of the session were drawn from the same distribution. If the hypothesis was rejected with $P = 10^{-10}$, it was highly unlikely that the firing rates obeyed the same distribution in the different blocks. The recordings were then assumed to be unstable, and the single unit was excluded from further analysis. Some single units had noisy firing patterns, and their PETHs did not have task-related structures that were significantly beyond the noisy fluctuations. We decided that the activity of a single unit would be designated as noise if the maximum of its corresponding smoothed PETH was <1 Hz or one standard deviation of the noisy fluctuations obtained by subtracting the smoothed PETH from the PETH. These noisy single units were excluded from further analysis.

In our recordings, the electrodes were either not moved daily or were advanced as little as possible. This raised the possibility that single units recorded from the same electrode on consecutive days might be the same neuron. It is not yet possible to resolve this issue unambiguously, but we used the following approximate method to eliminate putative repeat neurons. We collected all single units recorded from one electrode within a period during which the electrode was not moved by >42 μm. If the period exceeded 2 weeks, it was broken into consecutive 2-week spans. For each single unit, we calculated the smoothed PETH. We then computed the Pearson's product-moment correlation coefficient between the smoothed PETHs of the collected single units. The single units with correlations that exceeded 0.8 were considered as putative repeat units—possibly the same neuron. Among them, the one with the largest firing-rate range in the smoothed PETH was selected to represent the cell, and all others were excluded from analysis. Table S1 shows the results of these analyses.

Smoothing. PETHs are smoothed to improve the estimate of the underlying firing rate changes (6). The smoothing is done by using the least-square fitting to the PETH with B-form cubic splines. It was important to correlate the density of the knots for the splines to the shape of the PETH. More knots were placed at the peaks than at the slowly varying parts of the PETH. To accomplish this, we first smoothed the PETH with a Savitzky–Golay FIR filter of second order (7) using window size 15. The resulting curve was further smoothed with the robust fitting method LOWESS (8) with window size 60. The second smoothing finds the slowly varying part of the PETH by weighting less those points with large deviations from the mean. We used the difference between these two smoothed curves to place more knots at peaks of the PETH relative to the slowly varying parts. To counter the possibility that the features in the smoothed curve from the B-form cubic splines were due to fluctuations in

the spike trains, we created 50 bootstrap samples of the spike trains and averaged the smoothed curves. Fig. S1 shows an example of the smoothed curve.

Clustering. To determine the similarity of the smoothed PETHs (sPETHs) of the single units, we used a clustering algorithm modified from the shared nearest-neighbor clustering method (9). Each sPETH is considered as a point in the N-dimensional space, where N is the number of time bins in the sPETH. The similarity of two sPETHs is quantified by the distance, defined as one minus the Pearson's product-moment correlation coefficient between the sPETHs with the means subtracted. The clustering algorithm, which we call the "core point clustering algorithm," consists of the following steps. (*i*) Construct the neighborhood structure of each point by listing the nearby points with a distance of <0.15. (*ii*) Among the points that are not yet part of a cluster, select the one with the maximum number of neighbors as the core point of a new cluster. (*iii*) All points that are the neighbors of the core point and are not yet clustered are assigned to the new cluster, and the center of all points in the new cluster is calculated by taking the mean of the positions of the points. (*iv*) Go through all points that are not clustered, in the descending order of the number of neighbors. For each point and its neighboring points, compute the center. If the distance from the center to that of the new cluster is <0.15, the point and all its neighbors not yet clustered are included in the new cluster. The center of the new cluster is updated. This is done until no more points can be merged into the new cluster. (*v*) Repeat from step *ii* onward until no new cluster can be created. (*vi*) Merge clusters whose centers are within a distance of 0.2. (*vii*) Go through all points with no neighbors and assign them to the cluster with the nearest distance to the center if the distance is <0.4. The number of clusters depends on the parameters in the algorithm. The number is selected such that, according to subjective judgments, points with similar profiles are not split into different clusters, and those with different profiles are not joined into the same clusters. The exact number is not important. There are points that are not assigned to any cluster, because they are sufficiently far from any other points according to the distance measure criterion. Most of these points have noisy firing patterns.

Peak detection. To assess the phasic responses of the single units, we used an in-house peak detection algorithm on the smoothed PETHs aligned with various events. The peak detection procedure for a curve $y(x)$ was as follows. We fitted the curve to an asymmetric Gaussian function with a linear base by using the least square method. The formula of the fit function was $b_1\exp(-b_2(x - b_4)^2/(1 + b_3(x - b_4))^2) + b_5, + b_6x$, where b_i, $i = 1,\ldots, 6$ are the fitting parameters. The peak position (or the latency) is taken as the position of the maximum of the fitted curve. The peak height was defined as the maximum value of the fitted curve. The relative height of the peak is the difference between the peak height and the height of the inflection point, which is defined as the larger of the minima of the two halves of the fitted curve divided at the peak position. The width is the interval of the fitted curve above the inflection point. The half-width is the interval of the fitted curve above the value obtained by the height minus half of the relative height.

To avoid fitting fictitious peaks to noisy fluctuations, we rejected the peak if (*i*) the product of the width and the relative height was less than the product of the bin size (10 msec) and the noise threshold, set as twice the standard deviation of the difference between sPETH and PETH; or (*ii*) the relative height was less than the noise threshold; or (*iii*) the difference between the height and the value of the second highest point in the smoothed y outside of the peak was smaller than the noise threshold; or (*iv*) the peak position was less than five times the bin size to either boundary of the span of the curve.

Robustness of peak. A peak in a PETH could arise because of incidental alignments of random spikes in a subset of trials. Such a random peak is fragile and tends to disappear if different combinations of trials are used for constructing the PETH. Moreover, when a random peak appears in some of the combinations, its position should vary randomly. We therefore assessed the robustness of a peak by creating 100 sets of trials by using the bootstrap re-sampling technique. A bootstrap sample of a set of n trials is created by randomly selecting n times one of the trials in the set. We constructed PETH for each set of trials, and detected peaks. The peak was considered robust if the peak was accepted in $>50\%$ of the samples, and the standard deviation of the peak positions was <50 msec. We have tested, using random spike trains, that these two criteria reject all random peaks.

Distinguishing sensory and motor responses. A peak in the firing rate after a visual signal could be related to the visual signal itself (sensory) or the associated saccade (motor). To distinguish between these possibilities, we took advantage of the fact that the saccade timing relative to the visual signal is dispersed from trial to trial (Fig. 1*B*). If the peak were related to the saccade, its height should be smaller than the original value if the spikes are aligned with the saccade timings shuffled among the trials. For each set of shuffled saccade timings, we constructed the PETH and computed the peak height. The distribution of these peak heights, denoted as D_S, was constructed by using 200 sets of shuffled timings. We tested (z test, one-tailed) the null hypothesis that the peak height aligned with the original saccade timings, $r_{\text{peak,saccade}}$, is drawn from the distribution D_S. If the null hypothesis is rejected ($P = 0.05$) and $r_{\text{peak,saccade}}$ is larger than the mean of D_S, the peak is judged as tightly locked to the saccade, because shuffling saccade timings significantly reduces the peak height. On the other hand, if the null hypothesis is not rejected, the peak is not tightly locked to the saccade; it is highly likely that the peak is due to the visual signal. We assigned the peak as sensory-related in this case. Thus, although we could not distinguish intermediate states between the initial sensory response and the ultimate motor response, we were able to use locking to saccade onset to identify "motor"-related units, and for simplicity we termed the remaining units as "sensory." An example of a single unit responding to the first saccade rather than to the first Go signal is shown in Fig. S3. Fig. S4 illustrates an example of a neuron that responded to the first Go better than to the first saccade.

Population coding of time. The diversity of the response profiles that we found made it an attractive possibility that the population activity might be distinctive at each time point during the task. A decoding neuron driven by the population of neurons might be able to detect such distinctions, and thus would be able to tell the time by firing only at one time point. A simple model of the decoding neuron is the perceptron (10–11). This simple and widely used model for the decoding neuron mimics the neural integration and firing processes. We used this model to test our hypothesis of time encoding by populations of DLPFC and CN neurons.

In the perceptron, the firing rates of input neurons are weighted and summed, and the perceptron fires if the sum exceeds a threshold. Mathematically, a perceptron corresponds to a hyperplane in a M-dimensional space, in which each dimension represents the firing rate of one of the M input neurons. The firing rates of the input neurons at one time are represented as a point in the space. The hyperplane separates the points at different times into two groups, one for the population profiles that make the perceptron fire, the other for those that do not. The weights and the threshold, respectively, determine the orientation and the position of the hyperplane. If suitable weights and a threshold can be found such that the hyperplane separates one point from all others, the perceptron fires only at

one time-point during the task. If this can be done for all points, perceptrons can be constructed to distinguish all times. This result can be achieved in general if the number of input single-units N is large enough compared to the total number of time points, and the firing profiles of the single units are sufficiently distinctive (12).

The weights and thresholds can vary within some range without altering the response property of the perceptron to a given group of points. A useful concept for distinguishing these functionally equivalent weights and thresholds is the separation margin, which is the sum of the smallest distances to the hyperplane from the two separated groups of points. The separation margin is a measure of the robustness of the perceptron against the noisy fluctuations of the input firing rates. From trial to trial, the firing rates of inputs fluctuate, and hence the positions of the points jitter. A small margin makes the perceptron prone to inconsistent responses because of such fluctuations. The set of weights and thresholds that maximizes this margin gives the most robust decoder. Such a perceptron is also called a support vector machine (13). The maximum margin that can be achieved for a given set of points also indicates the degree of separation between the two groups that the perceptron distinguishes. To minimize the impact of noise fluctuations of the firing rates on the maximum margin, the firing rates of each input neuron is scaled with the noise level of the neuron before the weights and thresholds are computed. We describe below the mathematical process of calculating the maximum margin.

Let $R_i = (r_{i1},r_{i2},\ldots r_{iN})$ be the input profile of the ith neuron obtained by aligning spikes with the fixation, smoothing the PETH (sPETH) and further scaled with the noise level defined as the standard deviation of the difference between the PETH and the sPETH (if the standard deviation is <1 Hz, the noise level is set to 1 Hz). Here r_{ij} is the scaled firing rate at the jth time bin, and $n = 510$ is the total number of time bins (the duration of profile is from -100 msec to 5,000 msec relative to the fixation onset, and the time bin is 10 msec). Let M be the number of the neurons. At the jth time bin, the population profile of the neural responses is given by the firing rates of the M neurons at that time point: $P_j = (r_{1j},r_{2j},\ldots r_{Mj})$. Consider N perceptrons, each getting inputs from the M single units. The response D_{kj} of the kth decoder at the jth time bin is determined by the population profile P_j of the M input neurons at that time:

$$D_{kj} = H(W_k \cdot P_j - b_k),$$

where $W_k = (w_{k1},w_{k2},\ldots w_{kM})$ is the weigh vector, with w_{ki} being the synaptic weight of neuron i on the decoder k, and b_k is the firing threshold. $H(\cdot)$ is a step function whose value is 1 if the argument is positive and is 0 otherwise. The decoder fires ($D_{kj} = 1$) if the sum of the weighted inputs is larger than the threshold, and is silent ($D_{kj} = 0$) otherwise.

If the population profile at the kth time bin is sufficiently distinctive from the population profiles at all other time bins, it is possible to find the weights such that $D_{kj} = 0$ for $j \neq k$, and $D_{kk} = 1$. In other words, the kth decoding neuron fires only at $t = k10 - 95$ msec after the fixation onset. We searched for the maximum margin perceptron that decodes the kth time bin as follows. We first translated the coordinates of the space such that the point P_k, which corresponds to the population response profile at kth time bin, is at the origin. This shifts all points according to $P_j \rightarrow P_j = P_k$, and changes the perceptron criterion to $W_k \cdot P_j - b_k < 0$ if $j \neq k$, and $-b_k > 0$. Changing b_k shifts the separating hyperplane without altering its orientation. We set b_k such that the hyperplane is the farthest from the origin and still separates the origin from all other population points. We then scaled the weights by $(-b_k)$: $W_k \rightarrow W_k/(-b_k)$, which does not change the location of the hyperplane. The perceptron criterion now reads $W_k \cdot P_j + 1 < 0$ if $j \neq k$. The margin, which is the distance from the origin to the hyperplane, is equal to $1/|W_k|$, where $|W_k|$ is the length of the weight vector. Finding the maximum margin is thus equivalent to minimizing $|W_k|$ with the constraints given by the perceptron criterion. We used the multiplicative margin maximization algorithm proposed by Sha et al. (14) to solve this constrained maximization problem.

The maximum margin is positively correlated to the distinctiveness of the population profiles at different time points. In general, the distinctiveness increases with the number of the input neurons and the sharpness of the temporal variations of their profiles. High firing rates improve the maximum margin, because the points representing the population profiles in the M-dimensional space are further apart with increased firing rates. The maximum margin also improves if all time points within a window $t \pm \Delta t$ are excluded. This is because the firing rates of the neurons do not change much in a short time period that is comparable to the neuron's membrane time constant, and the population profiles of nearby time points are similar to each other. The half-width of the exclusion window, Δt, is referred to as the resolution of the perceptron.

To illustrate the properties of the maximum margin, we studied a simple hypothetical case in which the response profiles of the input neurons are Gaussians. The peak position t_k of the kth profile is set to $t_k = 5{,}100k/M - 95$ msec. This makes the centers of the profiles evenly distributed from -100 msec to 5,000 msec. In Fig. S8*A*, the maximum margins of the decoders decoding a time at the 10-msec time bins from -100 to 5,000 msec are plotted. The number of input neurons is 300; the maximum and the half-width of the Gaussians are 10 Hz and 50 msec, respectively. Because the maximum firing rates of each input neuron are the same, the maximum margin is the same for all decoders except those near the either ends because of the boundary effects. Fig. S8*B* shows that the maximum margin deceases as the half-width increases, or equivalently, as the sharpness of the response profiles decreases. Fig. S8*C* shows that the maximum margin increases with the number of the input neurons. The rate of increase is larger if the response profiles are sharper. Coarsening the resolution also increases the maximum margin, as shown in Fig. S8*D*. The effect saturates if the resolution is coarsened beyond the width of the response profiles.

Uneven distributions of the maximum rates or the half widths of the Gaussian response profiles lead to variations in the maximum margins of the decoders. A peak in the maximum margin distribution can result either due to a peak in the maximum rates of the input profiles alone, or due to a peak in the sharpness of the input profiles. The two effects can be combined to produce peaks and troughs in the maximum margin distribution across the decoders. Changing the distribution of the centers of the response profiles also leads to the variations in the maximum margin distribution.

We studied the maximum margins of the perceptron decoders by using the smoothed PETHs of DLPFC neurons as input neural profiles. We selected profiles with a maximum rate >5 Hz and the range (maximum $-$ minimum) greater than twice that of the noise level. The number of neurons selected was 506. In Fig. S6*A*, we plot the maximum margins of the 510 decoders decoding times from -100 msec to 5,000 msec with a 10-msec interval. The results for four resolutions are plotted. The maximum margins are prominent during the Go period, especially during the first Go, during the extra-peak period, and during the first 500 msec of the fixation period. To compare these results to a "null" case, we created random input neurons by shuffling the spikes in each trial of the input neurons while preserving the statistics of the interspike intervals, smoothing the resulting PETHs, and scaling the smoothed rates with the noise levels. The maximum margins of the 510 decoders with the random input neurons are also plotted in Fig. S6. The maximum margins are

significantly larger during the above-mentioned periods compared to those using the random input neurons, even at 10-msec resolution. Similar results are obtained by using 492 CN neurons (maximum rate >3 Hz, range greater than twice that of the noise levels), as shown in Fig. S6*B*. The maximum margins based on DLPFC neurons are in general larger than those based on CN neurons. Coarsening the resolutions increases the maxima of the maximum margins in these periods, as shown in Fig. S6. The increase during the first Go period is most dramatic, and saturates at the resolution at ≈60 msec in DLPFC and 40 msec in CN. The increase during the extra-peak period is also quite dramatic in DLPFC. In both areas, the maximum margins during the third and fourth Go periods are much smaller than during the first Go, also than during the extra-peak period and the fixation period. The increase of the margins with random inputs is modest, as expected. The maximum margins during the reward period are not significantly different from those with random inputs. Changing the number of input neurons by varying the maximum rate threshold of the selection criterion does not change the results much, as shown in Fig. S6 for 50-msec resolution. These results demonstrate that time encoding is robust during the first Go period, the extra-peak period, and the first 500 msec of the fixation period, which are the boundaries of the action sequences.

The weak dependence of the maximum margins on the number of input neurons indicates that many input profiles are not critical for the decoders. The importance of an input profile for shaping the response properties of the decoders can be accessed by the absolute values of the weights assigned to the profile. A large weight indicates that the profile contributes significantly to the firing pattern of a decoder. To identify the profiles that were most responsible for the large maximum margins observed during the first Go period (1,150–1,550 msec after the fixation), we ranked the absolute values of the weights to each decoder during the period, and selected the input profiles if their absolute weights are among the top five of the rank. Some input profiles are selected multiple times across the decoders. The profiles have sharp peaks during the first Go period. Many are the time-stamp units with dominant peak responses during the first Go period. The sharpness of the peaks of these profiles explains the saturation effects of the maximum margins during the first Go period as the resolution changed (Fig. S6). These results show that neurons with sharp changes in their response profiles have dominant contributions to the robustness of the time decoding based on the population activity. The dispersion of the timings of these sharp changes, such as the timings of the peaks of the neurons responding mainly to the first GO signal, is crucial for the decoding work in a wide range of times.

Online decoding of time. The response profile of a neuron, which is a smoothed PETH, is based on spikes over multiple trials. The success of time decoding using these profiles at inputs, therefore, does not necessarily imply that it is possible to construct an "online decoder" that receives the raw spikes of the neurons and indicates time at every trial by spiking at a particular time. The online decoding is possible if each neuron belongs to a group of neurons that have the same or similar response profiles. The key is that the spikes of the group at one trial are equivalent to the spikes of one neuron over multiple trials.

To construct the decoder, we created $N_a = 500$ artificial neurons from each of the M response profiles used in the "offline" decoders discussed in the previous section. As an example, consider the creation of the spikes of an artificial neuron from the response profile $R_i = (r_{i1},r_{i2},\ldots r_{iN})$ of neuron i. At each trial, the artificial neuron spikes $n_i = T\Sigma_{j=1}^{N} r_{ij}$ times, where $T = 5{,}100$ msec is the trial duration, and n_i is the mean number of spikes emitted by neuron i at each trial. The timing of each spike is determined in two steps. First, the spike is assigned to one of the N time bins, with the probability of assigning to the jth bin being $p_{ij} = r_{ij}/\Sigma_{j=1}^{N} r_{ij}$. Second, a random time within the bin is assigned to the spike. Thus, if the kth bin is selected, the spike time is $(k - 1 + rand)\Delta t$, where *rand* is a random number between 0 and 1. This process guarantees that the PETH of the artificial neuron is close to R_i.

We denote the spike rates averaged over the N_a artificial neurons at one trial ("group spike rates") as $\bar{R}_i = (\bar{r}_{i1},\bar{r}_{i2},\ldots\bar{r}_{iN})$. At the jth time bin, the group spike rate is given by $\bar{r}_{ij} = m_{ij}/N_a\Delta t$, where m_{ij} is the number of spikes emitted by the group in the bin. Mathematically, the spikes of N_a neurons at one trial are equivalent to the spikes of one neuron over N_a trials. Therefore, $\bar{R}_i$ should approximate R_i well, and the difference should decrease as N_a increases.

Because a response profile can be approximated with the group spike rates at one trial, online decoding is possible. An offline decoder can be converted into an online decoder by replacing the response profiles with the corresponding group spike rates at each trial, and scaling the weights with the noise level (note that in computing the decoders the response profiles were scaled with the noise level). As an example of this procedure, consider the kth online decoder, which should spike during each trial only near time $(k - 1)\Delta t - 95$ msec. At the jth time bin, the input to the decoder is the weighed sum of the group spike rates of all groups at the time point:

$$\bar{I}_{kj} = \bar{W}_k \cdot \bar{P}_j$$

where $\bar{W}_k$ is the scaled weight vector of the kth offline decoder computed in the previous section, and $\bar{P}_j = (\bar{r}_{1j},\bar{r}_{2j},\ldots\bar{r}_{Mj})$ is the group spike rates of M groups of the artificial neurons. The online decoder spikes once in the time bin if the input exceeds a threshold θ_k. The exact value of the threshold is not critical, as long as it allows the decoder to spike if the input is near the maximum of the weighted sums. If the threshold is set high, the spike timing of the decoder is precise, but the decoder might not fire in some trials because of fluctuations of the group spike rates. If the threshold is set low, the decoder spikes reliably, but the timing tends to be imprecise. We found that a good value for θ_k is 0.8 point from the minimum to the maximum of the weighted sum at all times to the kth offline decoder.

It was also possible to construct online decoders with the raw spikes of the recorded neurons as inputs, instead of the activity of the artificial neurons, by using the clustering results shown in Fig. 2 and Fig. S2. The profiles of neurons in the same cluster are similar; therefore, the population spikes of these neurons at each trial should approximate the averaged profile of the cluster. The approximation should be good if the number of neurons in the cluster is large. Thus offline decoders constructed with the averaged profiles of the clusters can be converted into online decoders with the recorded neurons in the same way as described above for the artificial neurons (in this case the firing rates and the weights were not scaled with the noise levels).

To show this, we selected all clusters with at least 10 members, and we used their averaged cluster profiles to construct offline decoders. This led to 35 clusters in DLPFC (Fig. S2*A*), and 27 clusters in CN (Fig. S2*B*). The separation margins of these offline decoders are shown in Fig. S7*A* for DLPFC and Fig. S7*D* for CN. If we included more cluster profiles, the margins improved, but the approximations of the averaged cluster profiles with the population spikes deteriorated. The decoders based on CN neurons have worse margins than those based on DLPFC neurons, as CN neurons with sharp peaks were excluded because they were in clusters with few units. We selected three decoders with the largest maximum margins during the fixation period, the first Go period, and the extra-peak period. The weighted sums of the inputs to the selected decoders in DLPFC are shown in Fig. S7*B* and those to four decoders in CN are shown in Fig. S7*E*. The offline decoders were converted to online decoders by

replacing the averaged cluster profiles with the averaged spike rates of neurons in the clusters in each trial, smoothed by using a Savitzky–Golay FIR filter of second order and window size 15. The thresholds were set to the 0.9 points from the minimum to the maximum of the inputs to the corresponding offline decoders. The trial-by-trial spikes of the online decoders corresponding to those shown in Fig. S7 *B* and *E* are shown in Fig. S7 *C* and *F*. The DLPFC online decoder during the first Go (D2 in Fig. S7*C*) does spike roughly at the decoding time, although there are some noisy fluctuations. This is true to some extent for the DLPFC online decoder (D3 in Fig. S7*C*) during the extra-peak period. Other decoders are swamped by noise, reflecting the fact that the margins of the corresponding offline decoders are relatively small.

Discussion. Several technical points need to be considered in relation to our findings. First, we found that the response profiles of neural population in the prefrontal cortex and striatum can be used to decode times in the task. This is not the same as decoding time trial by trial. A profile is computed by using spikes over multiple trials. Moreover, the saccade sequences were different for each trial. A direction-selective neuron would spike more in trials that contain its preferred directions. However, online decoding would be possible if there were groups of neurons sharing the same profile, and if in each group, it were equally probable to find a neuron preferring any of the four directions. In this case, the group-averaged spike rates at each trial would approximate well the profile, so that even if different subsets of neurons were active trial-to-trial, the group-averaged spikes would remain stable because the spikes of the subsets are equivalent. This stability enables a decoder to tell time in each trial. The existence of such groups is suggested by our clustering results and is a reasonable assumption, but it remains to be proven unequivocally.

When we limited our analysis to the recorded neurons only, without assuming such groups, it was still possible to decode time online during the first Go period, but not during other periods (Fig. S7). The number of neurons recorded may not have been large enough for adequate analysis by the perceptron model. There are methods that are mathematically more powerful than the perceptron for decoding population activity. For example, Zhang et al. (15) proposed a Bayesian probabilistic method for decoding the locations of rats by using the spikes of a population of place cells recorded in the hippocampus, and this method could be adapted to decode time in our case. We chose the perceptron as the decoder because it mimics the synaptic integration process of a real neuron, and it was adequate for our goal of demonstrating the fine-scaled time encoding in the population activity.

Second, our data were recorded over a period of 3 years. It is possible that the neural profiles are not representative of those in a single day, as some of the profiles could exist only transiently. We judge this to be unlikely, as the recordings were made after the monkeys had been overtrained, but we cannot exclude this possibility.

Third, the percentage of neurons in our sample that exhibited the time-stamp property was small, yet distributed time coverage was evident. Because of the sampling problems inherent in the recording methods, we cannot judge the true frequency of such neurons. We suspect that we were able to find them because we sampled the activity of thousands of neurons.

Finally, the maximum peak latency for the time-stamp responses during the first Go period was 435 msec. This was close to the upper limit possible for analysis of first Go responses in our experiments, because the second Go signal in the standard RSQ4 task came at 400 msec. We consider that the response at a 435-msec latency was unlikely to be a reaction to the second Go signal, given that this would require a response latency of 35 msec, which is shorter than the shortest latency we observed for the first Go signal (145 msec).

Our experiments differ from those in previous electrophysiological studies searching for time-related activity in behaving animals, as we did not require explicit timing of our experimental animals. Thus, time information was not imprinted in the neurons as a result of reward-based learning for coding particular time intervals. Our monkeys did have extensive exposure to the 400-msec RSQ4 time intervals, but for long periods they also were given exposure to other intervals (600 msec, 800 msec) in dedicated trial blocks as well as in blocks in which these different intervals were randomly mixed. At all of these intervals, the saccadic latencies of the monkeys grouped around the mean of *ca.* 280 msec, so that the saccade latencies did not suggest interval timing on the part of the monkeys.

1. Blazquez P, Fujii N, Kojima J, Graybiel AM (2002) A network representation of response probability in the striatum. *Neuron* 33:973–982.
2. Fuchs AF, Robinson DA (1966) A method for measuring horizontal and vertical eye movement chronically in the monkey. *J Appl Physiol* 21:1068–1070.
3. Fujii N, Graybiel A (2003) Representation of action sequence boundaries by macaque prefrontal cortical neurons. *Science* 301:1246–1249.
4. Fujii N, Graybiel A (2005) Time-varying covariance of neural activities recorded in striatum and frontal cortex as monkeys perform sequential-saccade tasks. *Proc Natl Acad Sci USA* 102:9032–9037.
5. Sheskin DJ (2000) *Handbook of Parametric and Nonparametric Statistical Procedures* (CRC, Boca Raton, FL).
6. Kass RE, Ventura V, Brown EN (2005) Statistical issues in the analysis of neuronal data. *J Neurophysiol* 94:8–25.
7. Orfanidis SJ (1996) *Introduction to Signal Processing* (Prentice–Hall, Englewood Cliffs, NJ).
8. Cleveland WS (1979) Robust locally weighted regression and smoothing scatterplots. *J Am Stat Assoc* 74:829–836.
9. Ertoz L, Steinbach M, Kumar V (2003) Finding clusters of different sizes, shapes and densities in noisy high dimensional data. *Proceedings of the 2003 SIAM International Conference on Data Mining* (SIAM, New York).
10. Rosenblatt F (1958) The perceptron: A probabilistic model for information storage and organization in the brain. *Psychol Rev* 65:386–408.
11. Matell MS, Meck WH (2004) Cortico-striatal circuits and interval timing: coincidence detection of oscillatory processes. *Brain Res Cogn Brain Res* 21:139–170.
12. Cover TM (1965) *IEEE Trans Electron Comput* EC-14:326-.
13. Vapnik VN (1982) *Estimation of Dependences Based on Empirical Data* (Springer, New York).
14. Sha F, Saul LK, Lee DD (2002) *Advances in Neural Information Processing Systems* (MIT Press, Cambridge, MA), Vol 15, pp 1065–1072.
15. Zhang K, Ginzburg I, McNaughton BL, Sejnowski TJ (1998) Interpreting neuronal population activity by reconstruction: Unified framework with application to hippocampal place cells. *J Neurophysiol* 79:1017–1044.

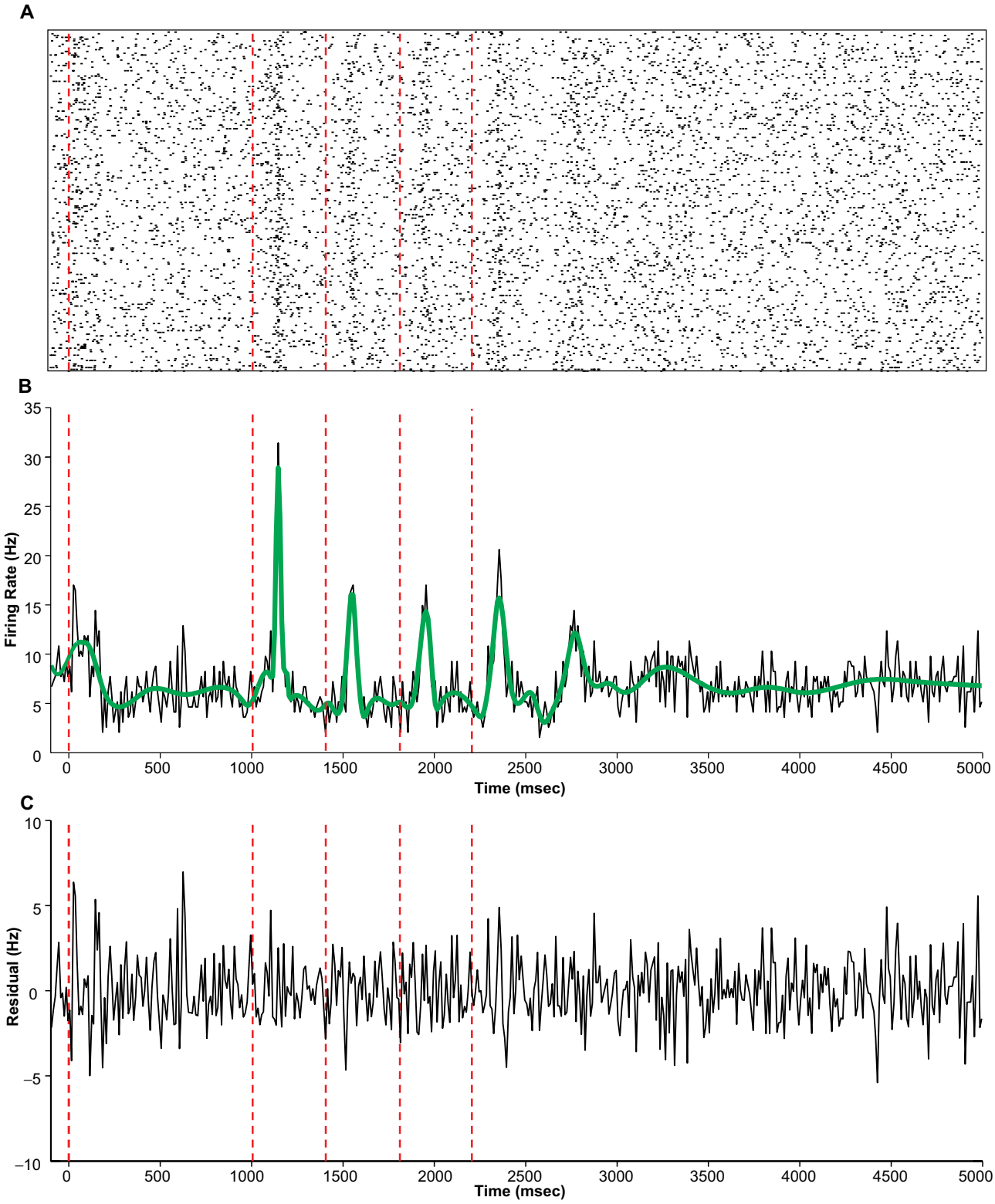

Fig. S1. An example of the method used to obtain the peri-event time histogram (PETH) and the smoothed PETH (sPETH). (*A*) Spike raster of a single unit. Spikes from multiple trials are aligned at the fixation onset, indicated by the first red vertical line (the other vertical lines indicate the Go times). (*B*) The constructed PETH (black curve) and its sPETH version (green curve). (*C*) Noisy fluctuations of the PETH around the corresponding sPETH.

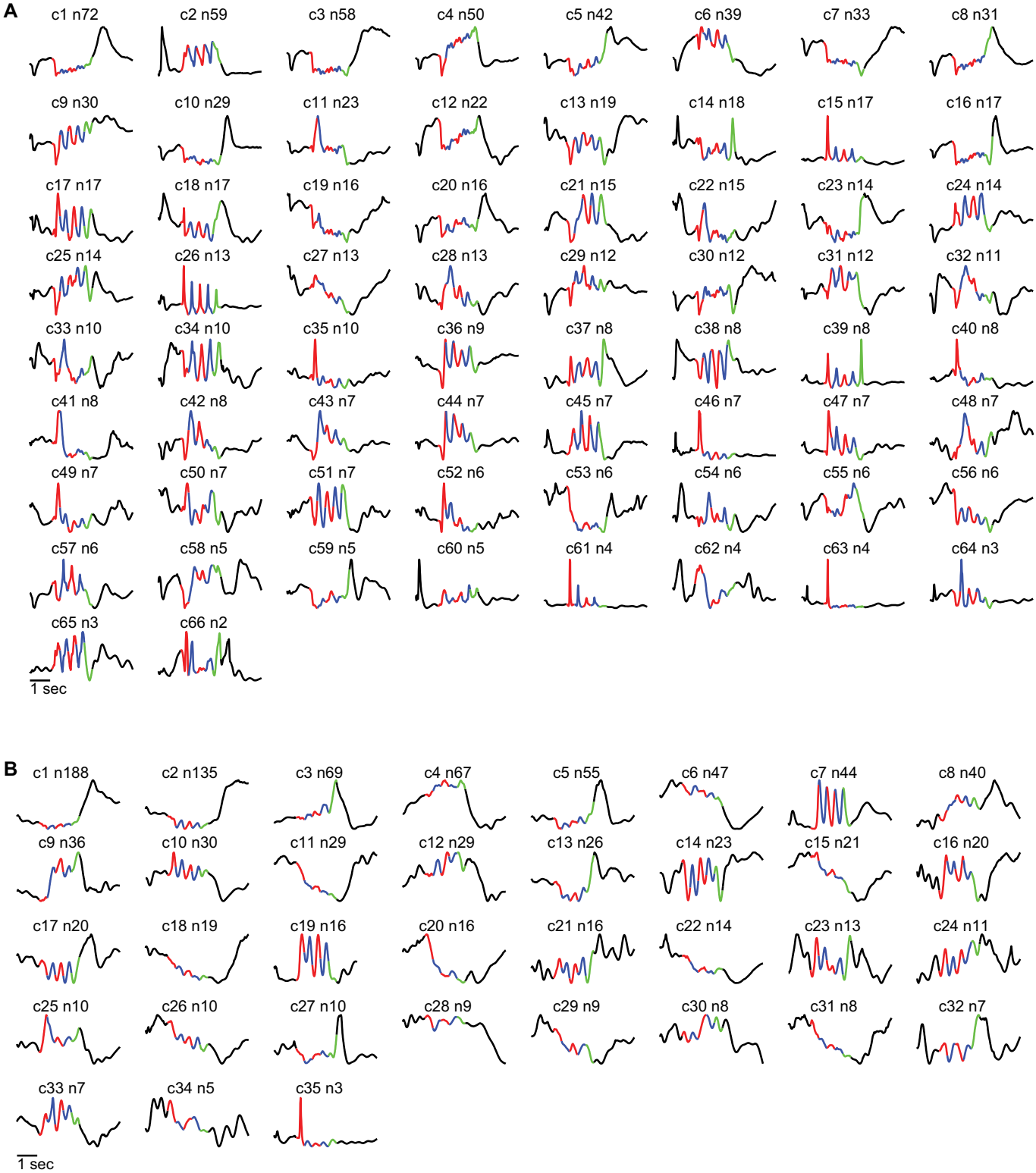

Fig. S2. The averaged response profiles of the clusters of DLPFC (*A*) and CN (*B*) units identified by the clustering algorithm. The cluster IDs and the number of single units in the cluster are shown above each profile. The colors indicate successive periods of the task. The first black trace indicates the fixation period; alternating red and blue indicate the consecutive 400-msec periods of the four Go signals; green indicates the extra-peak period, which is the 400-msec period after the last target-off; and the last black trace indicates the reward period and intertask-interval.

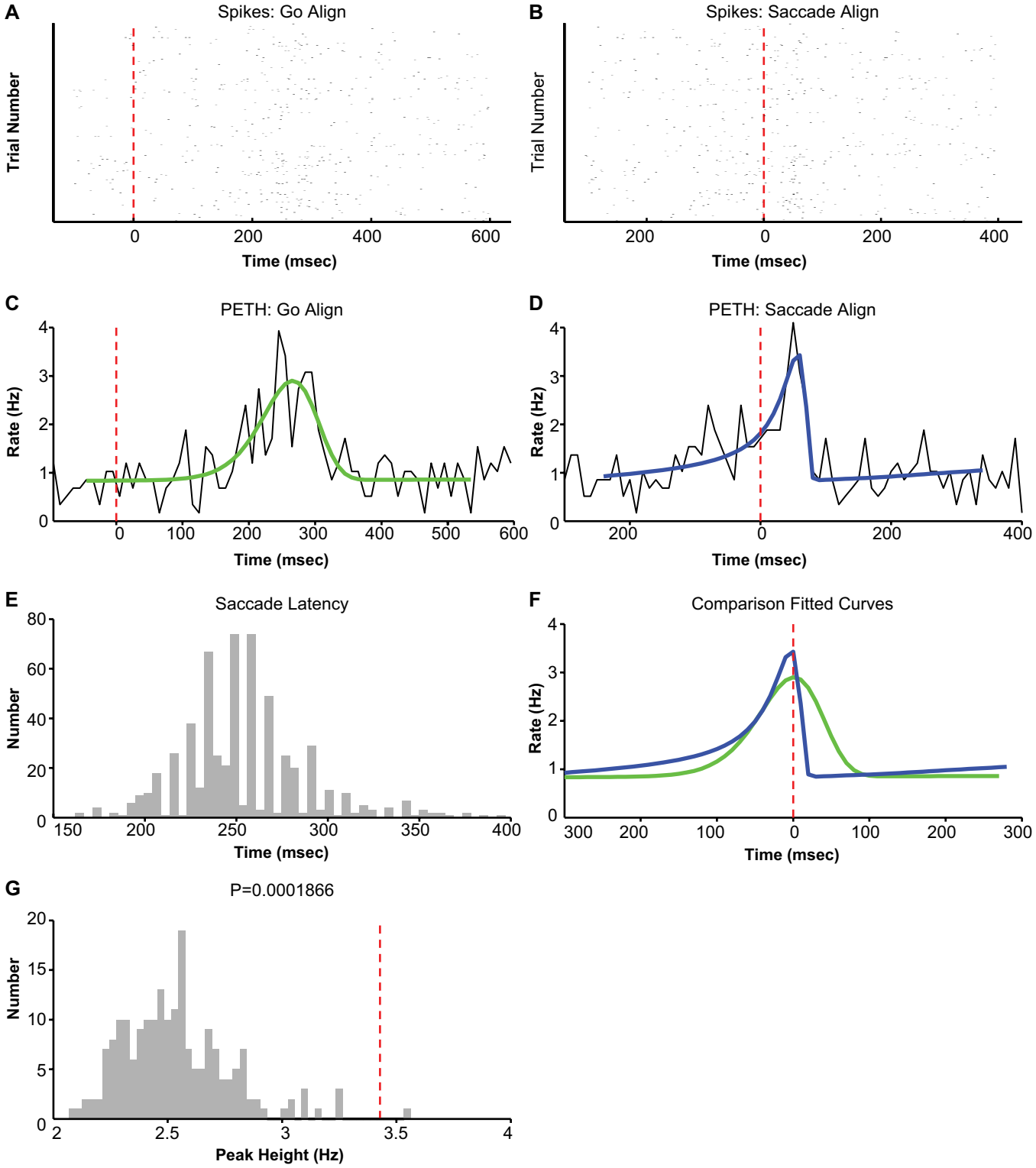

Fig. S3. An example of a single unit identified as a first saccade-related unit. (*A*) Spike raster aligned on the first Go. (*B*) Spike raster aligned on the first saccade onset. (*C*) The PETH of the Go-aligned spikes (thin black line), along with the fitted peak profile (thick green line). (*D*) The same as in *C* but the fitted peak profile with saccade onset alignment (thick blue line). (*E*) The distribution of the timings of the first saccades relative to the first Go signal. (*F*) A comparison between the fitted peak profiles (green, Go alignment; blue, saccade alignment). (*G*) Distribution of the peak heights of PETH constructed with shuffled saccade timings (200 samples). The red line indicates the peak height of the PETH constructed with the original saccade timings, as shown in *D*. The *P* value of the *z* test is shown.

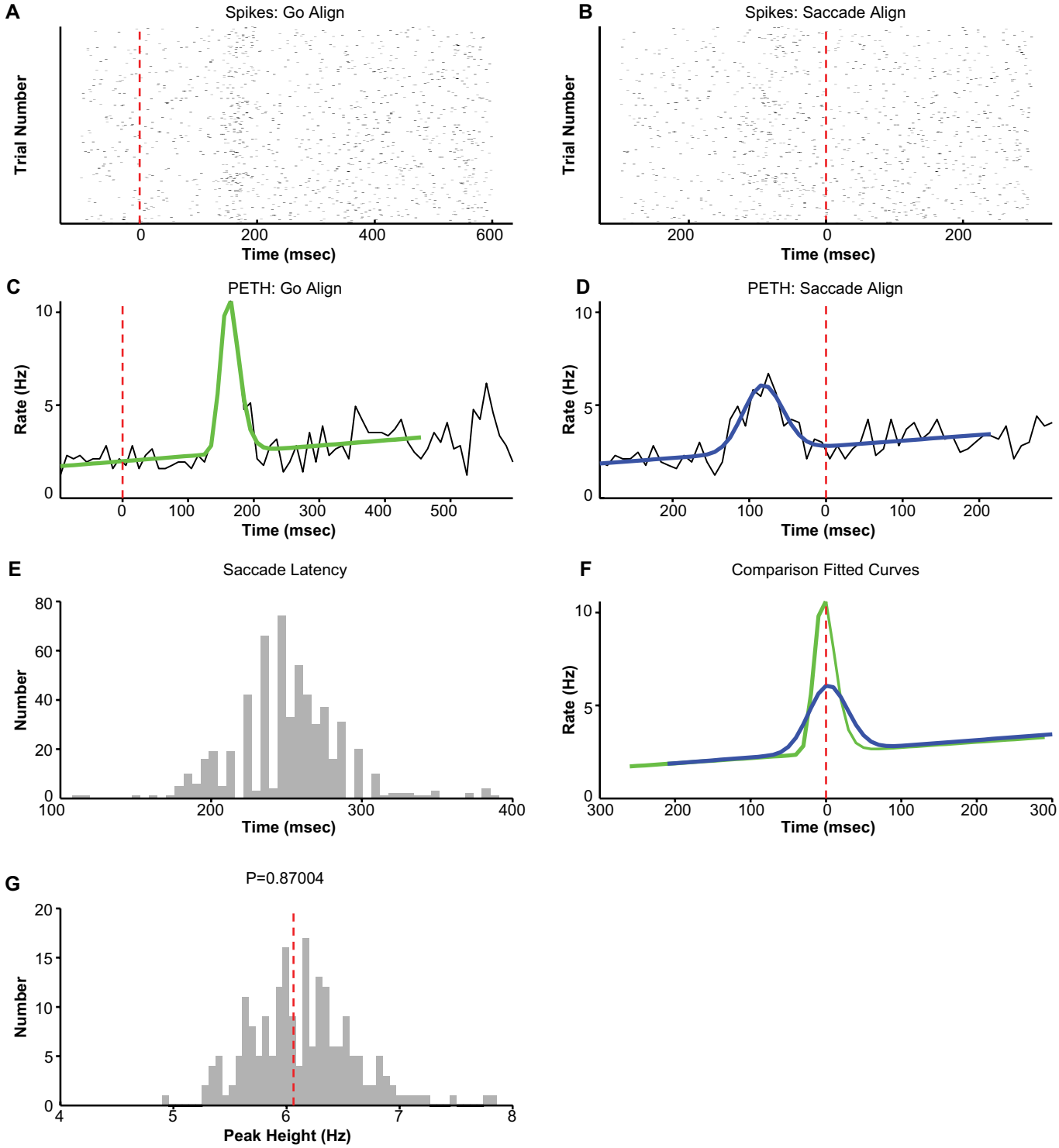

Fig. S4. An example of a ''visual'' (Go-responding) single unit. The plots in *A–G* correspond to those in Fig. S3.

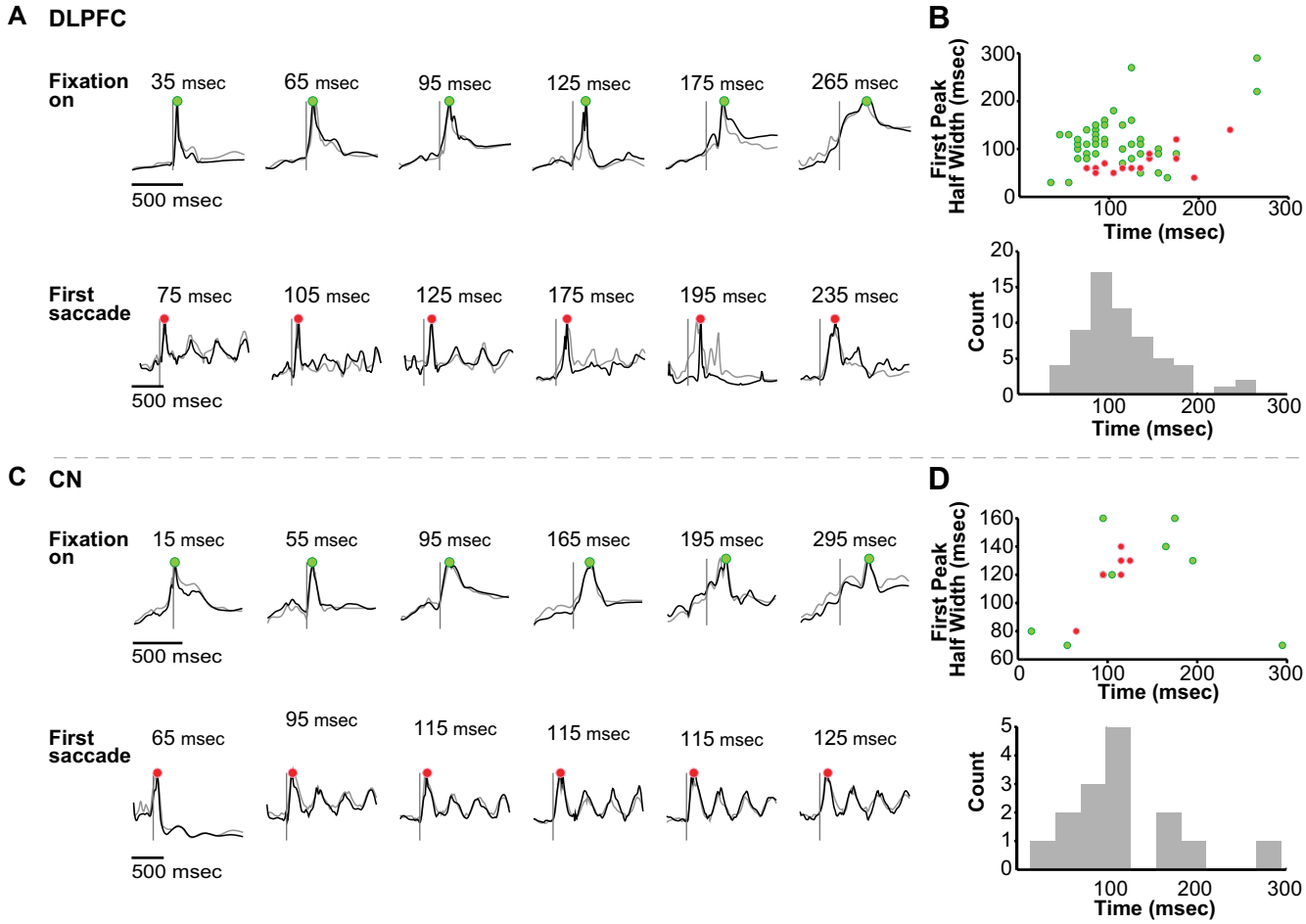

Fig. S5. (*A*) The smoothed profiles of single units in the DLPFC with dominant fixation on responses during the fixation period or dominant first saccade responses during the first Go period. The gray vertical lines indicate the spike alignment timings in each plot. Other conventions are the same as in Fig. 3. (*B*) Peak half-widths and distribution of peak timings for the first saccade (red) and fixation on (green) responsive neurons in the DLPFC. (*C* and *D*) Similar plots for CN units.

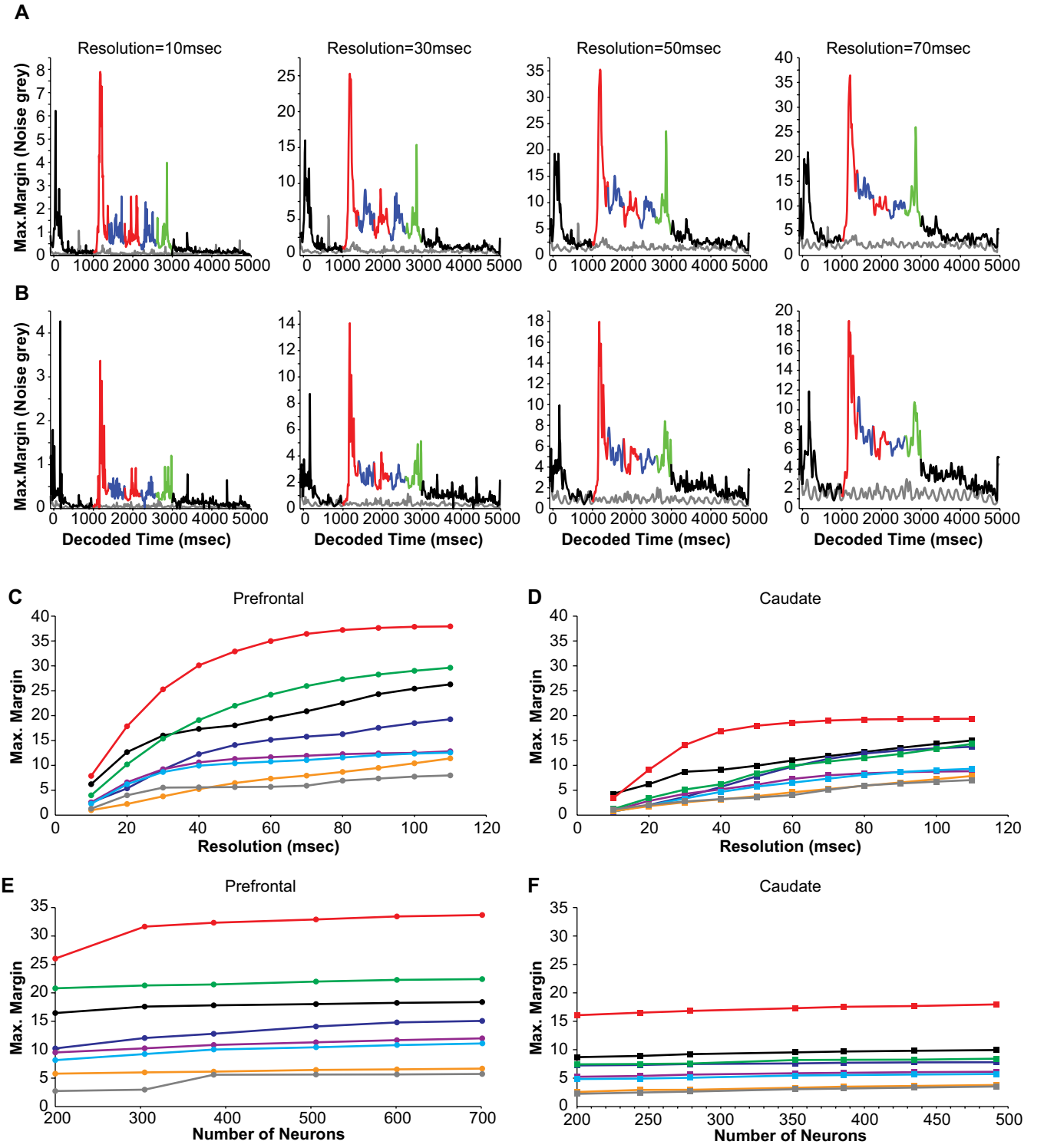

Fig. S6. (*A* and *B*) Maximum margins of decoders with 10-msec, 30-msec, 50-msec, and 70-msec resolutions for DLPFC (*A*) and CN (*B*) units. The color code is the same as in Fig. 4*A* of the main text. The noise level is shown in gray. (*C* and *D*) The maxima of the margins as a function of the resolution for the DLPFC (*C*) and CN (*D*) during the fixation period (black); the first, the second, the third, and the fourth Go periods (red, blue, magenta, and cyan, respectively); the extra-peak period (green); and the reward period (orange). The noise level is shown in gray. (*E* and *F*) The maxima of the margins as a function of the number of neurons during various periods in the DLPFC (*E*) and CN (*F*). The color scheme is the same as in *C* and *D*.

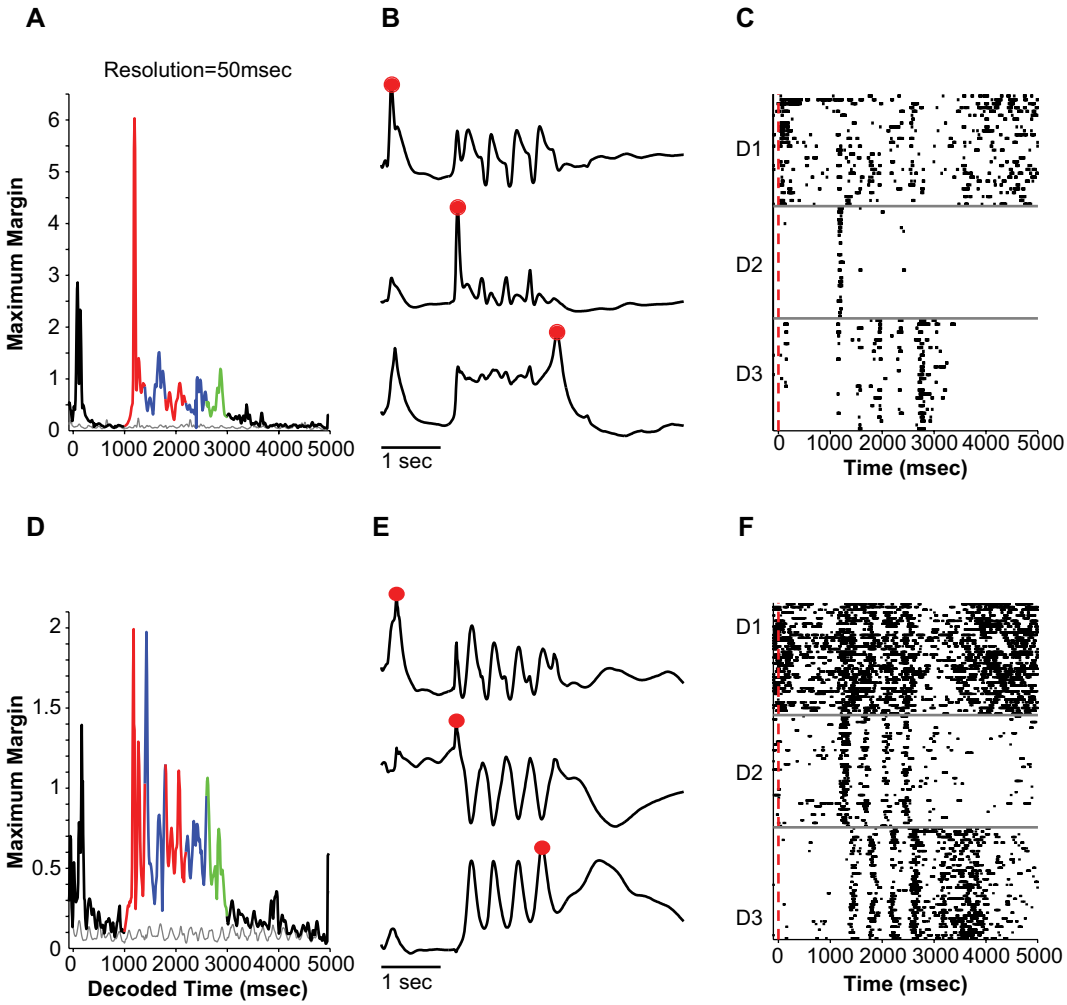

Fig. S7. Perceptrons driven by DLPFC neurons in clusters c1 to c35 (*A*–*C*) or by CN neurons in clusters c1 to c27 (*D*–*F*) in Fig. S2. The plotting convention is the same as in Fig. 4. (*A* and *D*) Maximum margin for perceptrons using the averaged profiles of the clusters as inputs. (*B* and *E*) Examples of the weighted sums of the inputs for three selected perceptrons during the task. (*C* and *F*) Trial-by-trial spike responses of three online decoders receiving raw spikes of neurons in the clusters as inputs. The number of trials in each decoder is 29 for the prefrontal inputs (*C*) and 40 for the striatal inputs (*F*).

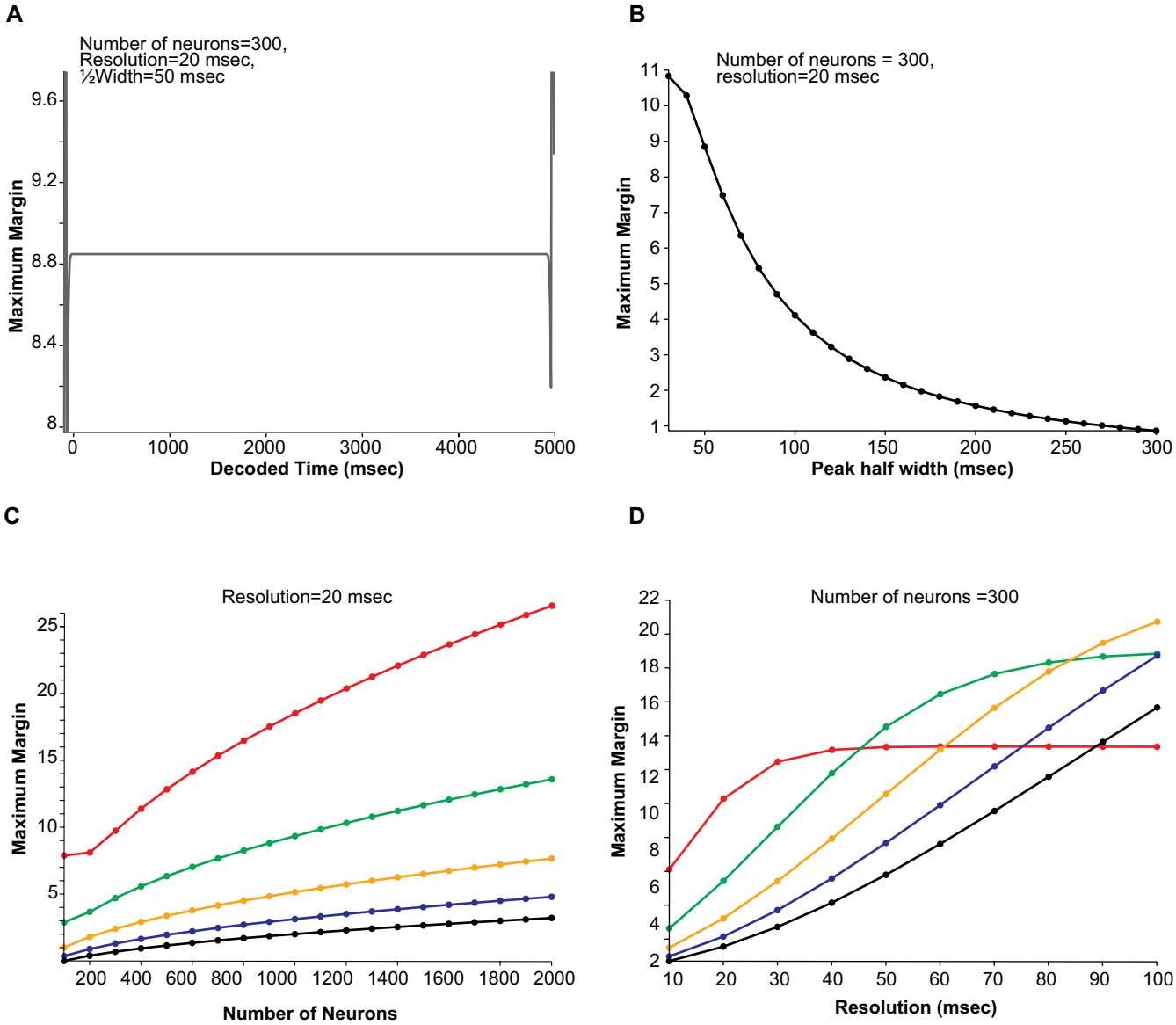

Fig. S8. Maximum margins in a model in which the input neural profiles are Gaussians. (*A*) Maximum margins of 510 decoders, each decoding a time point from −100 msec to 5,000 msec with an interval of 10 msec. The number of input neurons is 300. The resolution of the decoders is 20 msec. The half-width of the response profiles is 50 msec. (*B*) Maximum margin as a function of the half-width of the response profiles. Other parameters are the same as in *A*. (*C*) Maximum margin as a function of the number of input neurons. The perceptron resolution is 20 msec. The red, green, yellow, blue, and black curves are for the case of the half-width being 40, 80, 120, 160, and 200 msec, respectively. (*D*) Maximum margin as a function of the resolution of the perceptron. The number of input neurons is 300. The colors of the curves represent the same half-widths as in *C*.

Table S1. Classification of recorded units

Parameters	DLPFC neurons	CN neurons
All recorded units	2,483	3,203
Max rate $<$ 2 Hz (rejected)	293	358
Unstable rate (rejected)	173	341
Total accepted	2,017	2,504
Noisy (excluded)	275	399
Total excluding noisy units	1,742	2,105
Total nonrepeat units	1,613	2,035
Total used in population time coding analysis	506	492
Total units assigned to clusters	1,004 (66clusters)	1,070 (35clusters)
Total unclustered units	609	965