Emergent interactions within widely distributed neural populations

Ziv M. Williams[1,*], Robert Haslinger[2,3†] & Rollin C. Hu[1†]

[1]Department of Neurosurgery, MGH-HMS Center for Nervous Systems Repair, Harvard Medical School, Boston, Massachusetts 02114
[2]Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139
[3]Martinos Center for Biomedical Imaging, Massachusetts General Hospital, Charlestown, Massachusetts 02129
[†]These authors contributed equally to the work

[*]Correspondence: zwilliams@partners.org

SUMMARY

Coordinating the activity of widely distributed neural populations is fundamental to many of their functions. Since most cells possess only limited direct contact with each other or their external environment, however, it remains unclear by what interaction may neurons within these vast networks profitably coordinate their individual activities. We investigated this question within intact frontal cortical populations in primates by devising a combined bottom-up control and population-wide modeling approach. We find that when natural variations in the activity of arbitrarily selected neurons are allowed to influence the system's profit, other cells distributed widely across the population develop new, selective interactions with them. Counter to predictions based on local synaptic connectivity, most targeted interactions develop without direct synaptic contact and without a predicated structure by which to predict outcome. These findings reveal a novel innate property of indirect neural interaction, and demonstrate a neurobiological adaptation to a basic distributed systems problem.

INTRODUCTION

A principal function of neuronal populations is to enhance profit, maximizing the organism's receipt of reward over time or their chance of survival under changing and unpredictable conditions (Castellucci et al., 1970; Fu et al., 2012; Komiyama et al., 2010; Murray et al., 2000; Schultz, 1998). Such function is achieved by coordinating the activities of neurons which are often widely distributed, phenotypically similar and vast in number (Dan and Poo, 2006; Fujisawa et al., 2008; Roelfsema et al., 1997; Trappenberg, 2006). Yet, individually, most neurons contact only an exceedingly small fraction of the population and, aside from certain circuits such as the cortico-stiral loop, have no immediate access to information about changes in their external environment or a predicated connectivity structure by which to influence the systems receipt of reward (Abbott and Regehr, 2004; Beaulieu and Colonnier, 1989; Maass et al., 2002; Moore et al., 1970; Moritz and Fetz, 2011; Pearl, 2009; Prinz et al., 2004; Schneidman et al., 2006; Stevenson et al., 2009; Strogatz, 2001; Trappenberg, 2006; Uhlhaas et al., 2009; Uva and de Curtis, 2005; Williams and Eskandar, 2006). What has remained fundamentally unclear, therefore, is by what interactions do such widely separated neurons within these populations coordinate their individual activities to achieve profit? In other words, if

most cells within a network have no direct contact with each other or immediate access to information about how their shared activities affect or are affected by their external environment, in what way do they profitably interact?

We examined this question by devising a combined bottom-up control and population-wide modeling approach that allowed changes in the recorded activity and connectivity structure of widespread cortical populations in primates to be simultaneously tested based on the arbitrary influence of each neuron on receipt of reward. First, in order to identify which spatiotemporal interactions across the network are *specific* to the influence cells may have on profit, we used a closed-loop design in which natural variations in the activity of randomly selected single neurons were allowed to control a defined profitable behavior or function. Here, receipt of reward was importantly based on time-varying and unpredictable changes in the sensory stimuli presented to the animals, and the correspondences between each neuron's activity, presented stimulus and reward were randomly altered every session. Second, to quantify changes in the recorded population's activity collectively, we performed multiple-unit recordings of neurons across widespread cortical populations, and modeled each population's spiking activities using a statistically rigorous methodology that *simultaneously* took into account each neuron's firing rate, the influence of their prior spiking histories and the cross-correlated interactions between cells. While many studies have traditionally considered changes in activity of cells or cell-pairs individually, the activities of these cells are not independent (i.e. they interact). This approach, therefore, importantly allowed us to capture how the concurrent firing activity and interaction structure of the recorded network as a function of each individual cell's influence on the system's profit.

We make three basic observations. First, when a neuron empirically influences the system's receipt of reward under a time-varying environment, other cells distributed widely across the population (both locally and between distant areas) will develop new, selective interactions with them. Second, even though the ability of neurons to enhance profit is associated with a targeted increase in their interaction with the remainder of the population, this change is not accompanied by firing rate co-variations with other 'non-

profitable' cells or a net increase in the functional connectivity within the population. Third, and perhaps least intuitively, while new targeted interactions are both spatially and temporally selective, most empirically formed without direct synaptic contact and without a predicated network structure by which to predict reward outcome.

RESULTS

Bottom-up control of receipt of reward

A total of 517 neurons were recorded from three frontal cortical areas including the dorsolateral pre-frontal cortex (DLPFC), dorsal premotor cortex (PMd), and supplemental motor area (SMA) of two rhesus primates (*macaca mulatta*; **Supplemental Figure 1A**) (Nicolelis, 2008). During this time, and over different sessions, the animals performed a task involving successive trials in which a visual target was variably displayed in one of two random locations on a computer screen (top vs. bottom). The animals were rewarded based on natural variations in the firing rate of a single neuron which was arbitrarily chosen from the recorded population on each session, and which was termed the effector neuron (**Figure 1A**).

Prior to displaying the targets, the natural firing rate distribution of the selected neuron was determined by recording its spiking activities within successive 1000 ms windows during which the monkey remained at rest for five minutes. From this, the upper 90% and lower 10% quantiles of the cell's firing rate distribution were calculated to determine the upper and lower firing rate thresholds (**Figure 1A** and **Supplemental Figure 2**). Following, on each trial, a target would be randomly displayed on either the top or bottom of the screen. During this same time, the firing rate of the effector neuron would be calculated in 1000 ms windows advanced in 100 ms increments. If, at any point between 1500 ms to 3000 ms from the time of target presentation, the firing rate of the effector neuron reached either the upper 90% or the lower 10% threshold, the animal would be rewarded based on two randomly assigned conditions/mappings between the target location and firing rate threshold. On a given trial, for example, the animal would

be rewarded if either (1) the target was positioned on the top + the firing rate of the neuron went below the lower 10% threshold *or* (2) if the target was positioned on the bottom + the firing rate of the neuron went above the upper 90% threshold (**Figure 1B** and **Supplemental Figure 1B**). On other sessions, this relationship might be *reversed*. The monkey would not receive reward if the incorrect threshold or neither threshold were reached.

In order to limit the primate's ability to develop a single stereotypic behavior in which only one threshold was consistently reached, primates were required to repeat incorrect trials (i.e. the same target location would be presented if a non-rewarded threshold was reached). Moreover, the mapping between target location, threshold and reward was chosen randomly at the onset of each session (**Figure 1C**). In practice, the monkeys reached either threshold for more than 99% of the trials (i.e. it took 2000 ± 30 ms for the effector cells to reach either threshold). Therefore, importantly, effector cell performance was measured as the ability of individual effector cells to reach the *appropriate* threshold during the course of each different trial, rather than reaching any threshold.

Enhancement of effector neuron performance

Of the 44 effector cells recorded from, most developed the ability to reach the target-appropriate threshold over time. For 32 (73%) of the selected effector neurons, the proportion of trials in which the primates received reward significantly increased over the course of the session (first 10 trials versus the second half of the session; binomial test, $p < 0.05$). The average increase in performance was 26±3% (± s.e.m.; t-test, $p < 1e^{-10}$; **Figure 2A** and **Supplemental Figure 3**), and the peak increase in performance was 35±3% (**Figure 2B**). Taking the *null* hypothesis that all selected cells, as a group, would be equally likely to demonstrate an increase versus a decrease in performance, this improvement in performance was highly unlikely to have occur by chance (one-way t-test, $p = 2.9 \times 10^{-6}$; H_o=0% difference). For most cells, performance improved relatively rapidly, reaching learning criterion after a mean of 43±9 trials or approximately 4 minutes. Performance then slowly plateaued over the remainder of the session. Overall, we find no significant difference between cells recorded from the prefrontal (DLPFC)

versus premotor (PMd or SMA) cortical areas in the number of effector cells demonstrating improved performance (χ^2 test, $p = 0.34$) or their net improvement performance over the course of the session (t-test, $p = 0.49$). These observations, together, suggest that when arbitrary cells within a network influence receipt of reward under a varying environment, most empirically enhance the system's profit over time.

Network modeling

Neural activity can be accurately described by three principal metrics; the neuron's firing rate or the number of spikes that occur over a given time interval, its self-correlated activity in which the current spiking of a given neuron influences its own subsequent spiking and the interaction between neurons whereby the current spiking of a given neuron influences the subsequent spiking activity of other cells in the population (i.e. cross-correlated activity). Whereas the firing rate captures the relatively slowly varying or co-varying spiking of neurons over tens to hundreds of milliseconds, the cross-correlated activity describes the rapid millisecond-level time sequence in which individual spikes co-occur, and largely represents time dependency or interactivity between separate neurons (Abbott and Regehr, 2004; Schneidman et al., 2006; Trappenberg, 2006).

Identifying such spike patterns across large distributed networks and determining how they may concomitantly change during a behavioral task requires a computational framework that can capture them simultaneously (Okatan et al., 2005; Pillow et al., 2008; Truccolo et al., 2010). Here, we used logistic regression type Generalized Linear Models (GLM) based on the discretized spiking of all neurons in the recorded population to model each neuron's probability of spiking as a function of its own time varying firing rate, self-correlated activity and cross-correlated activity. This latter function has the form of fitted filters convolved with the other cells' spike trains, and reflects the time lagged cross-correlated interactions between cells. As detailed in the Experimental Procedures and in the supplement, a rigorous nested model and cross-validation approach was used to quantify these concomitant activities (**Figure 3A**).

We find that, consistent with prior studies (Asaad et al., 1998; Fujisawa et al., 2008; Fuster, 2008), mean firing rate of neurons at baseline was 3.9±0.3 spikes per second, and the number of functionally interacting neuron pairs (those with significant cross-correlated spiking) was 1.3±0.3% of all possible recorded network connections ($p < 0.01$ with Bonferroni correction). The overall ratio of excitatory versus inhibitory interactions was 3.5±0.8 (**Figure 3B**).

Population firing activities

One of the principal reasons that we recorded from multiple neurons across the population and modeled their spiking simultaneously, is that it allowed us to account for potential common influences on their firing rates and the cross-correlated interactions across multiple cells (**Figure 4A**). Network activity was compared between trials recorded at the beginning of each session, defined here as the baseline period, and those recorded at the end of the session, or asymptote period.

We find that enhancement of reward receipt of was not associated with a net increase in the firing rates of cells between baseline and asymptote. Specifically, there was little change in the trial-averaged firing rate of effector cells (0.07 spike/second difference; t-test, $p = 0.93$) or the population (0.09 spike/second difference; t-test, $p = 0.83$). On a single-cell basis, only 6% of effector neurons and 3% of neurons within the population demonstrated an increase in their trial-averaged firing rates (t-test, $p < 0.05$). Consistently, the modeled time-varying activity remained similar between periods (ANOVA, $p = 0.43$).

When further considering variations in the firing activity of effector cells, improved task performance was not associated with an increase in firing rate fluctuations (i.e. the tendency of the firing rate of a cell to range between high and low within a trial). Both the variance (+0.04±1.5; t-test, $p = 0.99$) and entropy (-0.02±0.08; t-test, $p = 0.72$; **Supplemental Figure 4**) of effector firing rates remained largely the same between baseline and asymptote. Consistent with this, it took effector cells 2100±40 ms to reach threshold at baseline versus 1990±30 ms at asymptote (i.e. with a minimum start time

1500 ms from target onset; t-test, $p = 0.12$; **Figure 4C**). There was also no significant difference in the time it took to reach the bottom or top thresholds (ANOVA, $p = 0.32$), and no significant difference in the time it took the individual 16 low-firing versus 16 high-firing effector cells to reach their threshold at 2130 ± 20 ms and 2010 ± 50 ms, respectively (t-test, $p > 0.05$).

Next, we examined whether the changes in the firing rates of the effector cells co-varied with that of other cells in the population. In other words, did the firing rate of effector cells simply 'follow' the activity of other cells within the network such that they increased/decreased their firing rates concurrently? An important distinction between cross-correlated interactions and co-variations in the firing rates of neurons is that the former occurs at the temporal scale of milliseconds and reflects the time-dependency of spiking in one neuron on another (as described above). To further illustrate this distinction, we display the mean firing rates of two simultaneously recorded cells over two separate trials **Figure 4D**. When further examined across all trials, the firing rates of the two cells do not co-vary (Pearson's correlation; $r = 0.017$, $p = 0.83$; **Figure 4E**), but their exact spike timings are significantly correlated as demonstrated by their cross-correlogram (**Figure 4F**, *top*). When randomly shuffling the spike timings within each 1000 ms window, the correlation coefficient for the firing rates remains unchanged (since the firing rates within each window remains the same), but their precise cross-correlated interactions are abolished (**Figure 4F**, *bottom*). Similarly, we find essentially no net co-variation in firing rates between effector cells and other cells within the population when compared across all tested sessions and, more importantly, no differences between baseline and asymptote (Pearson's correlation, $r = 0.0098 \pm 0.007$ versus 0.0102 ± 0.007; t-test, $p = 0.96$).

Finally, we also examined the individual firing rates of cells across the population in order to estimate how they, themselves, would have performed if target selection was controlled based on their own firing rate thresholds. To do this, we retrospectively examined changes in the firing rate of each population cell and then used the same threshold-target-reward criteria to determine if they would have similarly lead to an

enhancement in receipt in reward over time. We find that these cells did not demonstrate an independent increase in performance over time (1.8% change; t-test, $p > 0.05$; see also behavioral controls, below).

These findings, together, indicated that neurons on which receipt of reward was not directly contingent did not modulate their firing activities with the rewarded target and that effector cells did not simply co-vary their firing rates with that of the population. In other words, changes in the firing modulation of neurons based on profit were selective to the effector cells and were not associated with similar changes by other neurons in the population.

Targeted increase in network interactions

Following, we examined how the functional connectivity structure of the network may have changed based on the influence individually selected neurons had on receipt of reward. We find that once a neuron empirically influenced profit, other cells within the population developed new, selective interactions with it.

For the 32 sessions in which there was an improvement in task performance, cross-correlated interactions with effector cells accounted for 1.1±0.4% of all possible effector cell connections at baseline. However, the percentage of interactions during asymptote was 2.3±0.5%. This constituted a significant 2.1±0.2 – fold increase in the number of interactions between effector cells and the population (t-test, $p = 0.012$; **Figure 5A**). Consistent with this, most effector cells (63%) demonstrated an increase in the number of interactions with only a few cells (9%) demonstrating no change.

Increase in the number of cross-correlated interactions was only observed for effector neurons, and was not associated with a general increase in interactions between other neurons in network. In contrast to the two-fold increase in interactions observed for effector cells, other cells within the population demonstrated only a 1.2±0.2 – fold change (t-test, $p = 0.18$; **Figure 5A**). On a session-by-session basis, this constituted a significant

difference in the distribution of connectivity changes for effector neurons in comparison to all other recorded neurons within the population (KS test, $p = 0.007$; **Figure 5B**).

We next importantly asked how likely is it that such a change in connectivity would have been observed if, instead of the effector cell, the connectivity change of another randomly chosen cell had been considered. In other words, how likely is it that we randomly pick a single cell from the primate's brain and observe the same effect? To answer this, we performed a pseudo-effector GLM resampling analysis in which connectivity changes of cells in the populations, as determined by GLM model fitting and validation, were randomly sampled and substituted for the true effector cells (see Experimental Procedures). From this non-parametric distribution, we observe that the probability that a randomly chosen cell in the population would demonstrate the same or greater increase in connectivity as the effector cells was only 0.019 (**Figure 5C**). In other words, on a cell-by-cell basis, the increase in connectivity was spatially *selective* to within 1.9% of the recorded population. This was statistically significant at $p < 0.05$.

Finally, increase in the number of effector cell interactions was not the result of added cross-correlated interactions being statistically significant but weak. Interaction strength can be quantified from the fitted parameters that define each interaction filter in the GLM, and can also be tracked as a function of interaction latency (i.e. cross-correlated spiking at specific time lags). We find that the mean absolute value of the significant interactions was largely similar between baseline and asymptote ($\beta_{q,m}$ 0.32±0.04 versus 0.37±0.06, respectively; t-test, $p = 0.52$; **Supplemental Figure 5A**). In other words, new effector cell interactions observed during asymptote were as strong, if not slightly stronger, than those observed during baseline. Moreover, when examining all (significant and non-significant) effector cell parameters describing the interactions across all time lags, there was an increase in their mean value during asymptote ($\beta_{q,m}$ 0.11±0.02 versus 0.17±0.03; t-test, $p = 0.04$; **Supplemental Figure 5B**). Therefore, effector cells not only demonstrated an increase in the number of statistically significant interactions, they also demonstrated a broader increase in the strength of their cross-correlations even when not strictly considered significant.

Therefore, once a neuron empirically influences receipt of reward under a time-varying environment, other cells within the population develop new, selective interactions with it. This increase in functional connectivity, however, is not associated with concomitant changes in the firing rate modulation of other 'non-profitable' cells in the population (as noted above) or a corresponding net increase in connectivity between them.

Changes in second-order network structure

Although other recorded cells within the population did not demonstrate a net increase in their interactions, we investigated whether their second-order structure may have changed based on formal graph theoretic measures such as hub and clustering coefficients. Hub coefficients describes how many connections *go through* a particular node and clustering coefficients describes how *other* nodes, connected to the node in question, are themselves interconnected (**Figure 5D**, *bottom*). Importantly, a cell may demonstrate a change in connectivity but not necessarily become a network hub or demonstrate a change its clustering coefficient (Newman, 2010).

We observe that effector cells were more likely to become network hubs between baseline and asymptote (mean hub coefficient 0.051±0.020 to 0.089±0.023; t-test, $p = 0.15$), whereas cells in the population were not (0.062±0.010 to 0.068±0.007; t-test, $p = 0.43$). The difference in the distributions of hub coefficients for effector cells compared to other cells in the population also trended towards significance when compared between baseline and asymptote (KS test, $p = 0.07$). In contrast, effector cell clustering coefficients remained essentially unchanged between baseline and asymptote (0.052±0.023 to 0.053±0.022; t-test, $p = 0.98$), whereas clustering for the population significantly increased (0.046±0.013 to 0.081±0.017; t-test, $p = 0.02$; **Figure 5D**, *top*). Similarly, the difference in the distributions of clustering coefficients for effector cells in comparison to the population significantly diverged (KS test, $p = 0.04$). First-order measures such as the proportion of excitatory versus inhibitory connections and incoming versus outgoing connections remained essentially unchanged for both effector cells and the population (data not shown).

It, therefore, appeared that, when individual cells affect profit, other cells within the population develop new targeted interactions with them. Consistently, but not necessarily predicted by these first-order changes, cells that influence profit are more likely to become centralized hubs. In comparison, other 'non-profitable' cells are more likely to become clustered even though net connectivity between them remains largely unchanged.

Relation between enhanced interactions and profitable behavior

Although we observed that it was unlikely that the spatially selective formation of effector interactions occurred by chance, we also investigated whether the population's ability to enhance profit was temporally dependent on the formation of new interactions with profitable effector cells. Here, the primates performed nine dual recording sessions in which an effector neuron was selected in one session, as before, but was then followed by a second session in which another effector neuron was randomly picked. Similar to prior observations, behavioral performance significantly improved over the course of the first session (t-test, $p < 0.001$; **Figure 6A**). Once a new effector neuron was selected, however, behavioral performance immediately dropped back to a level similar to its initial baseline, and then again gradually improved over the remaining session (t-test, $p = 0.013$).

As before, we similarly find a 1.9±0.5 - fold increase in the number of interactions between baseline and asymptote for effector cells during the first session (t-test, $p = 0.024$; **Figure 6B**). When a new effector cell was selected, however, the number of interactions with the original effector neuron dropped back to a level slightly higher than but not significantly different from the number of interactions found in the original baseline (t-test, $p = 0.41$). More importantly, interactions for that effector neuron then stayed largely unchanged over the remainder of the second session even though performance improved (t-test between second baseline and asymptote, $p = 0.88$). At the same time, the number of interactions for the second actively selected effector neuron increased 1.7±0.6 - fold, from 1.2 to 2.0% (t-test, $p = 0.25$). This suggested that the

ability of the network, in aggregate, to enhance profit over time depended on their ability to form and re-form targeted interactions with neurons whose activities empirically influenced reward under a time-varying environment.

To further examine whether the increase in effector cell interactions reflected their immediate role in modulating receipt of reward, we also looked at temporal differences in effector cell interactions for the same population of cells but during the inter-trial-interval at which time no sensory cues were displayed (and no reward was possible). We find that even though these trial windows were interleaved by as little as 3 seconds and the proceeding sensory stimuli were identically distributed throughout the session, there was virtually no difference in the number of interactions between baseline and asymptote during this inter-trial-interval. In fact, there was a slight 0.9±0.2 fold drop in the number of interactions for effector cells between periods (t-test, $p = 0.41$) and essentially no difference in the number of interactions for the population (t-test, $p = 0.27$; **Figure 6C**).

Additional analyses included in the Supplement show that effector cells which possess a lower overall number of interactions at baseline were slightly less likely to demonstrate an increase in performance over the session. Effector cells demonstrating a lower starting performance were also slightly more likely to have a lower number of baseline interactions, higher firing rate and higher trial variance. We, otherwise, find no significant differences in effector neuron connectivity based on their individual properties compared to the population or the qualities of the GLM fits used to describe them (see supplemental results).

Spatiotemporal properties of new targeted interactions

Many cells within the population formed spatiotemporally targeted interactions with effector neurons across relatively long distances (e.g. DLPFC-PMd cell pairs) and in the absence of direct synaptic contact. At baseline, effector neurons displayed a higher number of cross-correlated interactions with cells located within the same area of the brain (1.5±0.6%) compared to cells located distantly (0.8±0.4%). While the number of interactions at asymptote was far higher for local cell pairs (3.7±0.8%) compared to those

that were distant (2.4±0.5%), distant cell pairs, in fact, demonstrated a slightly larger 3.0±0.2 - fold increase in interactions compared to the 2.6±0.2 - fold increase for those recorded locally (ANOVA, $p = 0.02$; see Experimental Procedures for how local versus distant percentages were calculated). Consistent with this, the distribution of connectivity between local versus distant effector cell interactions also significantly changed between baseline and asymptote (KS test, $p = 0.02$ and $p = 0.01$, respectively). Using pseudo-effector GLM resampling analysis, these increases were highly spatially selective for effector cells, both for local as well as distant interactions ($p = 0.02$ and $p = 0.02$, respectively; **Figure 7A**).

Second, although the development of new interactions was spatiotemporally selective to effector cells, most interactions were not relayed through direct putative synaptic contact. The development new interactions, or the statistical likelihood that spiking in one neuron is followed at a consistent time-lag by another, implies a functional connectivity between them and can be used to infer the whether their spiking is monosynaptically coupled (Berry and Pentreath, 1976; Moore et al., 1970; Uva and de Curtis, 2005). We find that 59% of effector cell interactions were new in comparison to baseline. Of these, 73% displayed significant cross-correlated interactions at time-lags longer than 10 ms, suggesting that most new targeted interactions with the effector cells were not monosynaptically relayed. Time-lags for local effector cell cross-correlations were 67±6 ms at baseline and 66±7 ms at asymptote. In comparison, time-lags for distant cell cross-correlations were 73±6 ms at baseline and 89±6 ms at asymptote, and constituted a predominant redistribution towards longer time-lagged interactions (**Figure 7B,C**).

The formation of new interactions with effector neurons was not associated with a common synchronous drive at varied time-lags (i.e. an extrinsic source that simultaneously influenced spiking of cell pairs at temporally specific delays). Rather, there was a significant shift in the dispersion of individual effector cell time-lags between baseline and asymptote (Ansari-Bradley test; $p = 0.029$). This shift was especially pronounced when examining time-lags for distant cell pairs compared to local cell pairs (Ansari-Bradley test; $p = 0.003$; **Figure 7B**). Consistent with this, the self-correlated

activity of neurons, which may be influenced by such drive, remained individually the same across time lags (ANOVA, $p = 0.45$).

Behavioral controls

Finally we examined whether changes in effector cell performance or their interactions could have been influenced by more simple or non-specific differences in sensory input, reward frequency or unconstrained physical behavior by the animals over the course of the sessions.

First, we find that neurons did not 'learn' to enhance receipt of reward when reward was not directly contingent on variations in their individual firing activities. Animals performed a control task in which they were given the same random display of targets on a computer screen and their receipt of reward was made to increase from 25% to 75% over the course of each session (i.e. they experienced the same/similar sensory input and increase in reward as during the standard sessions). In this task, however, increase in reward was not contingent on whether each neuron's firing rates appropriately went above/below their thresholds (i.e. reward increased over time irrespective of the neurons activity). We find that, of the 13 cells tested using this control, none demonstrated a significant difference in their firing rate when comparing the first and second half of the session, either individually (t-test, $p > 0.05$) or as a population (5.2±0.7 *versus* 5.6±0.8 spikes/sec; t-test, $p = 0.73$; **Supplemental Figure 6A**). More importantly, when using each neuron's own upper/lower firing rate thresholds, none of the cells demonstrated an increase in 'performance' based on variations in their own individual firing activities (binomial test, $p > 0.05$). In other words, unlike effector cells observed during the standard sessions, these cells did not develop the ability appropriately modulate their firing activities over the course of the task. There was also essentially no difference in the cross-correlated interactions of these cells when comparing the first versus second half of the sessions (1.0±0.4 - fold change; t-test, $p = 0.92$). These observations, together, indicated that the firing activity of cells did not simply passively respond to changes in the sensory stimuli presented during the task or increase their connectivity when receipt of reward was not directly contingent on changes in their individual activities.

In second control, we examined whether reward expectancy, in itself, could have affected the firing rates of the recorded neurons (Schultz, 1998). Here, reward was given at fixed frequencies of 25%, 50% or 75% over separate sets of 50 trials, each, during which targets were randomly presented on the top/bottom of the screen. As above, receipt of reward in this control was not dependent on fluctuations in the neurons rates. When comparing trials between these control sets, we find that none of the 14 recorded neurons demonstrated a significant difference in their firing rates either individually (t-test, $p > 0.05$) or as a population (ANOVA, $p = 0.85$; **Supplemental Figure 6B**). In other words, variations in neuronal firing rates did not appear to be associated with differences in reward expectancy or frequency *per se*.

Third, even though the animals remained visibly motionless, we examined for potential differences in unconstrained movements made by the primates based on simultaneously recorded electromyographic (EMG) activity during performance of the main task. We found no change in EMG activity magnitude during selection of top versus bottom displayed targets (t-test, $p = 0.43$), or between pre- versus post-target presentation time periods (windows = 3000 ms; t-test, $p = 0.19$). There was also no correlation between firing rates across 8 tested neurons and changes in the magnitude of EMG response during the same recordings (Pearson's correlation, $p > 0.05$). Most importantly, there was no difference in normalized EMG activity during target presentation periods between baseline and asymptote (from 0.91 to 1.04; t-test, $p = 0.16$; **Supplemental Figure 6C**).

Finally, we examined the locations and saccade magnitudes of eye movement for the primates over the course of the session (ISCAN Inc., MA). For most of the trial duration, the primates viewed the displayed target. Between baseline and asymptote, there was no significant difference in either horizontal or vertical axis eye position (ANOVA; $p = 0.94$) or magnitude of eye movement (t-test, $p = 0.54$); keep in mind that essentially the exact same distribution of top/bottom targets was displayed between baseline and asymptote. In a separate control recording, we also had the primates perform free eye movements with the screen turned off and simultaneously recorded the activity of 10

neurons. Only one neuron demonstrated a differential response in its mean firing rate when freely viewing the top half versus bottom half of these fields (t-test, $p > 0.05$).

While the above controls strongly suggest that potential concomitant factors such as general reward expectancy, unconstrained motor responses or eye movements were unlikely to have contributed to effector neuron activity or changes in cross-correlated interactions, perhaps the strongest indicator of this is that the neuronal modulation and increase in the number of interactions was highly selective to effector cells and was not observed for other cells in the general population. Reward related responses, for example, are known to be subject to widely divergent input from lower mesencephalic areas that will often simultaneously influence the activity and interaction pattern of whole cortical areas (Schultz, 1998). Altering the functional connectivity of a single arbitrarily selected effector cell at variable time-delays out of the population would not be generally possible under such known mechanisms. Similarly, changes in limb and eye movement (even when subtle) are commonly associated with robust population-wide changes in firing activities over far broader spatial scales than that of a single neuron (Watanabe et al., 2009). This makes it physiologically unlikely that even gross motor differences in the primate would account for such cell-selective changes in connectivity.

DISCUSSION

The formation of consistent correlations in the time-delayed spiking between neurons indicates the presence of a dependency or functional interaction between them. Such interactions have been broadly demonstrated in experimental settings under canonical paradigms of synaptic modification, including Hebbian learning and spike timing dependent plasticity (Abbott and Regehr, 2004; Ahissar et al., 1992; Bao et al., 1997; Pinsker et al., 1970; Trappenberg, 2006), and have shown that conditioned responses can be serially trained by enhancing the interaction between synaptically coupled sensory-motor neurons or nodes. At the population level, learning models such as Hopfield nets and heteroasociative networks have also shown that multiple interconnected nodes can be

trained to increase the profitable output of a network based on variations in their connectivity weights (Chakrabarti and Basu, 2008; Gurney, 1999; Trappenberg, 2006). Other examples include bi-layered feed forward neural networks and reinforcement algorithms. In essentially all of these cases, however, learning is based on nodes that are fully interconnected and have access to all other nodes in the system (i.e. they are spatially interconnected but their trained weights may be sparsely enhanced). Perhaps more importantly, the input-output structure all nodes in the system are explicitly known or defined.

Therefore, while these observed mechanisms and models have provided invaluable insight into synaptic function and adaptive behavior, the spatiotemporal pattern and means by which most neurons profitably interact within intact cortical networks in animals has remained less well understood. In particular, most neurons within the brain are physically sparsely connected, with each individual cell contacting only an exceedingly small fraction of the network (Abbott and Regehr, 2004; Beaulieu and Colonnier, 1989; Stevenson et al., 2009). While adult neurons may have the capacity to flexibly modulate existing synaptic connections and develop new dendritic boutons (Glanzman et al., 1990; Kauer et al., 1988; Zito and Svoboda, 2002), the capacity to form new lines of communication between distant neurons is also extremely limited (Darian-Smith et al., 1990). More importantly, with the exception of certain cortical-subcortical loops, most neurons also have no predicated output structure in which to affect receipt of reward, and no constructive way of predicting that variations in the activity of a particular cell in the population may contribute to profitable behavior under a changing environment (Fuster, 2008; Trappenberg, 2006). What has remained fundamentally unclear, therefore, is by what interactions do such widely separated neurons, as individual cells within the population, coordinate their activities to achieve profit?

We find here that when natural variations in the activity of arbitrary single neurons within intact cortical networks in animals influence profit, other cells distributed widely across the population developed new, selective interactions with them. These interactions form empirically, without a direct means by which to predict which neuron, among many, may

influence receipt of reward (i.e. cell selection and trial-by-trial target locations as well as the mappings between firing activity/target location/reward across sessions were all varied randomly). Perhaps more importantly, even though these interactions were spatiotemporally targeted to effector cells, they were also able to form across long distances and in the absence of direct putative monosynaptic contact. Yet, their development was not associated with a net increase in connectivity between other surrounding cells in the network and was not due to potential changes in common extrinsic drive. When cells no longer influenced receipt of reward, however, enhanced connectivity was abolished but could newly shift to other cells that empirically influenced receipt of reward. These observations suggest that, while interactions between cells may develop through known synaptic-based rules (Abbott and Regehr, 2004; Maass et al., 2002; Trappenberg, 2006), they can also form between any cells in the network that profitably influence the population's outcome. In other words, interactions (as determined by synchronization between neurons' precise spike timings) can develop, in principal, between any nondescript neurons in the network but do so in a memory-less fashion that depends on the same trained sensory input being present and the cell's immediate influence on receipt of reward (Maass et al., 2002).

This study reveals a novel innate property of neural interaction at the collective network level that allows cells across the population to empirically 'seek out' and develop targeted interactions with neurons that influence the system's profit. By allowing wide-ranging cells to selectively synchronize their spiking activities with that of other individual neurons within the network, such a property may provide large populations the ability to profitably coordinate such individual activities under rapidly changing and unpredictable environments. These findings provide added insight into adaptive neuronal function and demonstrate a neurobiological adaptation to a fundamental distributed systems problem faced by multi-cellular populations – the need to profitably coordinate the activities of individual cells within networks that are both widely distributed and sparsely connected/contacting (Banchereau and Steinman, 1998; Beaulieu and Colonnier, 1989; Brock et al., 2011; Chin et al., 2002; Diggle et al., 2007; Holland, 1998; Kollmann

et al., 2005; Komiyama et al., 2010; Laubach et al., 2000; Pearl, 2009; Trappenberg, 2006).

EXPERIMENTAL PROCEDURES

Task structure

Two adult rhesus monkeys (*macaca mulatta*), each weighing 6 and 7 Kg, viewed a monitor that displayed targets in one of two possible varying locations. During each session, a single neuron from within the recorded population was randomly designated to be the effector neuron. The firing rate of the effector neuron was continuously monitored over the course of each trial. If at any point, after a brief delay, the firing rate of the neuron reached the 10% or 90% quantile (lower or upper threshold) of its firing rate distribution, and that threshold appropriately corresponded to the displayed target location, the primate would receive a drop of juice reward. For example, the upper 90% threshold + top target position or lower 10% threshold + bottom target position would be associated with receipt of reward, or *vice versa*. Only one target would be displayed on a given trial, and its location would be distributed pseudo-randomly across the two possible locations. The primate would only get rewarded on each trial only if the correct threshold was reached between 1500-3000 ms from the time the target was first displayed, and would perform many such trial repetitions over the course of the session. The mapping between target selection and firing activity was chosen randomly at the onset of each session (e.g. high firing + bottom target = reward). No reward would be given if the incorrect threshold was reached or if neither threshold was reached.

Neural population recordings

Silicone multi-electrode recording arrays (NeuroNexus Technologies Inc., MI) were surgically implanted in each monkey. A craniotomy was performed over the frontal lobe under standard stereotactic guidance (David Kopf Instruments, CA) (Nicolelis, 2008; Paxinos, 2000). After directly visualizing the cortical gyral pattern underlying the craniotomy, 1-3 multi-electrode arrays each spaced up to 10 millimeters apart arrays were implanted into the DLPC, SMA and PMd (**Supplemental Figure 1**). Each array

contained 32 electrode contacts in a 4x8 configuration horizontally spaced by 400 μm and vertically spaced by 200 μm. A Plexon multichannel acquisition processor was used to amplify and band-pass filter the neuronal signals (150 Hz – 8 kHz; 1 pole low-cut and 3 pole high-cut with 1000x gain; Plexon Inc., TX). Signals were then digitized at 40 kHz and processed to extract action potentials in real-time by the Plexon workstation. Putative neurons were required to separate clearly from any channel noise, to demonstrate waveform morphology consistent with that of a cortical neuron, and to have at least 99% of spikes separated by a minimum refractory inter-spike interval of 1 ms. No multiunit activity was used.

Anatomically, *local* cell pairs were defined as all pairs recorded from any electrode contact located within the same anatomically defined area (i.e. DLPC). *Distant* cell pairs were defined as all pairs recorded from electrode contracts located within any two distinct areas (i.e. DLPC-PMd, DLPC-SMA or PMd-SMA).

Modeling network activity

For a population of N neurons, the spiking activity of each neuron $n \in 1 \ldots N$ was binned at a millisecond time-scale with 1 indicating the occurrence of a spike and 0 indicating no spike. Each neuron n's discrete time spike probability $\lambda_n(t)$ was described by fitting a GLM of the logistic regression type;

$$\log\left[\frac{\lambda_n(t)}{1-\lambda_n(t)}\right] = \mu_n + f_{tvar,n} + g_{hist,n} + h_{pop,n}$$

The logistic form of the left hand side keeps $\lambda_n(t)$ constrained between 0 and 1. μ_n is a fitted parameter quantifying the trial averaged mean *firing rate* (over entire 3000 ms trial) and $f_{tvar,n}(t)$ quantifies the *time-varying firing rate* at a 500 ms resolution. $g_{hist,n}(t)$ quantifies the neuron's *self-correlated* spiking (i.e. the influence of its own spiking history on its spike probability at time t). This term describes, for example, the neuron's refractory period and/or subsequent rebound/bursting behavior, and has the form of a

fitted filter convolved with the neuron's own past spike train. Exact functional forms for all these terms are given in the Supplement. Finally, $h_{pop,n}(t)$ quantifies the *cross-correlated* interactions of all other neurons $m \neq n$ in the recorded population onto neuron n. This is written as a sum over each neuron m's independently acting interactions whereby;

$$h_{pop,n}(t) = \sum_{m \neq n} h_{pop,n,m}(t) = \sum_{m \neq n} \left[\sum_{t'=1}^{T_{hist}} \theta_m(t - t') H_{n,m}(t'; \{\beta_{m,n}\}) \right]$$

Similarly to $g_{hist,n}(t)$, each $h_{pop,n,m}(t)$ (bracketed term) takes the functional form of a fitted filter $H_{n,m}(t'; \{\beta_{n,m}\})$ but convolved with neuron m's spike train ($\theta_m(t) = 1$ for neuron m spiking in bin t and 0 otherwise). The set of parameters $\{\beta_{n,m}\}$ describing the filter are fit via the logistic regression model and exact equations are given in the supplement. The fitted filters describe neuron m's time lagged interaction onto neuron n's probability of spiking. Together, the set of all N logistic regression models defines a neuronal population model describing the second order network interaction structure or connectivity matrix (Okatan et al., 2005; Pillow et al., 2008; Truccolo et al., 2010).

As detailed in the supplement, all logistic regression models were fit using a rigorous nested model approach. Each term of the model (μ, f_{tvar}, g_{hist}, h_{pop}) was sequentially fit to training data and only kept if its inclusion improved the fit (log likelihood) of an independent test data set. In addition, cross-correlated interactions were only kept if the fitted parameters $\{\beta_{n,m}\}$ describing them had Bonferroni corrected, significant p-values. This combined model selection approach ensured that the identified cross-correlated interactions were justified by the data and not a result of over-fitting.

Statistical testing

Recording sessions were divided into equal halves based on the number of trials performed. The first half was defined as the baseline period and the second half was defined as the asymptote period. This was done in order to maintain the same number of trials, relative spike count and statistical power between periods, and also to limit any

assumptions made on when a given effector cell had first 'learned' an association or reached learning criterion (see also additional neurophysiologic controls, above).

Two cells were defined as having an interaction if at least one of the fitted parameters $\beta_{q,m,n}\,(q \in 1\ldots5)$ defining that cross-correlated interaction term $h_{pop,m,n}(t)$ had a statistically significant Bonferroni corrected value (p = 0.01). The percentage of interactions per cell was defined as the number of significant cross-correlations with the cell divided by all possible connections between cells across the recorded population. A two-tailed student's t-test ($p < 0.05$) was used determine whether there was a significant difference in the percentage of cross-correlated interactions between the baseline and asymptote periods across sessions (keep in mind that these comparisons were based on the validated results obtained from the GLMs). A Kolmogorov-Smirnov test (KS-test; $p < 0.05$) was used to determine whether the distribution of differences in the percentage of cross-correlated interactions between baseline and asymptote per session were significantly different between effector cells and the population. All values were given with their standard-error.

Pseudo-effector GLM resampling analysis was performed to determine the probability that a randomly chosen cell within the recorded population would demonstrate the same or greater increase in connectivity as the effector cells. For each session, we recorded from N neurons (1 effector neuron and N-1 neurons from the population). Each cells' connectivity change between baseline and asymptote, in turn, was determined based on GLM fits that were previously trained/validated from each cell's spikes. However, here, the connectivity change of the assigned effector neuron was randomly substituted 1000 times with the modeled connectivity change of a cell picked from the population. From this non-parametric distribution, we calculated the proportion of substituted connectivity changes that met/surpassed those of the actual effector neurons using the original GLM fits.

Finally, graph theoretic measures were calculated based on standard forms described previously (Newman, 2010). The hub coefficients were calculated as the proportion of

nodes in the network passing through the node in question but which were not connected to each other (i.e. for B, if A-B-C but not A-C). The clustering coefficients were calculated as the proportion of nodes connected to each other but not connected to the node in question (i.e. for B, if B-C-D but not B-D). Connectivity graphs were displayed as closest neighbor circular graphs using Cytoscape (Yeung et al., 2008).

SUPPLEMENTAL INFORMATION

Supplemental Information includes additional Supplemental data and figures, Supplemental Figures (1-7) and references, and can be found with this article online at doi:

ACKNOWLEDGEMENTS

We thank E. Brown and E. Bizzi for their initial input into the project; K. Harous, M. Campos and K. Spiliopoulos for their insightful discussion and review of the manuscript. Z.M.W. designed the study, performed neural recordings, analyzed the data and wrote the paper, R.H. developed the generalized linear models and edited the manuscript, and R.C.U performed neural recordings. Z.M.W is funded by NIH 5R01-HD059852, PECASE and the Whitehall Foundation, R.C.U is funded by the NREF and R.H is funded by NIH K25-NS052422.

REFERENCES

Abbott, L.F., and Regehr, W.G. (2004). Synaptic computation. Nature *431*, 796-803.

Ahissar, E., Vaadia, E., Ahissar, M., Bergman, H., Arieli, A., and Abeles, M. (1992). Dependence of cortical plasticity on correlated activity of single neurons and on behavioral context. Science *257*, 1412-1415.

Asaad, W.F., Rainer, G., and Miller, E.K. (1998). Neural activity in the primate prefrontal cortex during associative learning. Neuron *21*, 1399-1407.

Banchereau, J., and Steinman, R.M. (1998). Dendritic cells and the control of immunity. Nature *392*, 245-252.

Bao, J.X., Kandel, E.R., and Hawkins, R.D. (1997). Involvement of pre- and postsynaptic mechanisms in posttetanic potentiation at Aplysia synapses. Science *275*, 969-973.

Beaulieu, C., and Colonnier, M. (1989). Number and size of neurons and synapses in the motor cortex of cats raised in different environmental complexities. J Comp Neurol *289*, 178-181.

Berry, M.S., and Pentreath, V.W. (1976). Criteria for distinguishing between monosynaptic and polysynaptic transmission. Brain Res *105*, 1-20.

Brock, D.A., Douglas, T.E., Queller, D.C., and Strassmann, J.E. (2011). Primitive agriculture in a social amoeba. Nature *469*, 393-396.

Castellucci, V., Pinsker, H., Kupfermann, I., and Kandel, E.R. (1970). Neuronal mechanisms of habituation and dishabituation of the gill-withdrawal reflex in Aplysia. Science *167*, 1745-1748.

Chakrabarti, B.K., and Basu, A. (2008). Neural network modeling. Prog Brain Res *168*, 155-168.

Chin, A.I., Dempsey, P.W., Bruhn, K., Miller, J.F., Xu, Y., and Cheng, G. (2002). Involvement of receptor-interacting protein 2 in innate and adaptive immune responses. Nature *416*, 190-194.

Dan, Y., and Poo, M.M. (2006). Spike timing-dependent plasticity: from synapse to perception. Physiol Rev *86*, 1033-1048.

Darian-Smith, C., Darian-Smith, I., and Cheema, S.S. (1990). Thalamic projections to sensorimotor cortex in the newborn macaque. J Comp Neurol *299*, 47-63.
Diggle, S.P., Griffin, A.S., Campbell, G.S., and West, S.A. (2007). Cooperation and conflict in quorum-sensing bacterial populations. Nature *450*, 411-414.
Fu, M., Yu, X., Lu, J., and Zuo, Y. (2012). Repetitive motor learning induces coordinated formation of clustered dendritic spines in vivo. Nature *483*, 92-95.
Fujisawa, S., Amarasingham, A., Harrison, M.T., and Buzsaki, G. (2008). Behavior-dependent short-term assembly dynamics in the medial prefrontal cortex. Nat Neurosci *11*, 823-833.
Fuster, M.J. (2008). The prefrontal cortex, 4 edn (London, UK, Elsevier).
Glanzman, D.L., Kandel, E.R., and Schacher, S. (1990). Target-dependent structural changes accompanying long-term synaptic facilitation in Aplysia neurons. Science *249*, 799-802.
Gurney, K. (1999). An Introduction to Neural Networks (London, UCL Press Limited).
Holland, J.H. (1998). Emergence: from chaos to order (TN, USA, Perseus Books).
Kauer, J.A., Malenka, R.C., and Nicoll, R.A. (1988). NMDA application potentiates synaptic transmission in the hippocampus. Nature *334*, 250-252.
Kollmann, M., Lovdok, L., Bartholome, K., Timmer, J., and Sourjik, V. (2005). Design principles of a bacterial signalling network. Nature *438*, 504-507.
Komiyama, T., Sato, T.R., O'Connor, D.H., Zhang, Y.X., Huber, D., Hooks, B.M., Gabitto, M., and Svoboda, K. (2010). Learning-related fine-scale specificity imaged in motor cortex circuits of behaving mice. Nature *464*, 1182-1186.
Laubach, M., Wessberg, J., and Nicolelis, M.A. (2000). Cortical ensemble activity increasingly predicts behaviour outcomes during learning of a motor task. Nature *405*, 567-571.
Maass, W., Natschlager, T., and Markram, H. (2002). Real-time computing without stable states: a new framework for neural computation based on perturbations. Neural Comput *14*, 2531-2560.
Moore, G.P., Segundo, J.P., Perkel, D.H., and Levitan, H. (1970). Statistical signs of synaptic interaction in neurons. Biophys J *10*, 876-900.

Moritz, C.T., and Fetz, E.E. (2011). Volitional control of single cortical neurons in a brain-machine interface. J Neural Eng *8*, 025017.
Murray, E.A., Bussey, T.J., and Wise, S.P. (2000). Role of prefrontal cortex in a network for arbitrary visuomotor mapping. Exp Brain Res *133*, 114-129.
Newman, M. (2010). Networks, An Introduction (Oxford, Oxford University Press).
Nicolelis, M.A.L. (2008). Methods for neural ensemble recordings, 2 edn (Boca Raton, FL, Frontiers in Neuroscience).
Okatan, M., Wilson, M.A., and Brown, E.N. (2005). Analyzing functional connectivity using a network likelihood model of ensemble neural spiking activity. Neural Comput *17*, 1927-1961.
Paxinos, G.H., X. F.; Toga, A. W. (2000). The rhesus monkey brain: in stereotaxic coordinates (San Diego, CA, Academic Press).
Pearl, J. (2009). Causality: Models, Reasoning and Inference, 2 edn (New York, NY, Cambridge University Press).
Pillow, J.W., Shlens, J., Paninski, L., Sher, A., Litke, A.M., Chichilnisky, E.J., and Simoncelli, E.P. (2008). Spatio-temporal correlations and visual signalling in a complete neuronal population. Nature *454*, 995-999.
Pinsker, H., Kupfermann, I., Castellucci, V., and Kandel, E. (1970). Habituation and dishabituation of the gill-withdrawal reflex in Aplysia. Science *167*, 1740-1742.
Prinz, A.A., Bucher, D., and Marder, E. (2004). Similar network activity from disparate circuit parameters. Nat Neurosci *7*, 1345-1352.
Roelfsema, P.R., Engel, A.K., Konig, P., and Singer, W. (1997). Visuomotor integration is associated with zero time-lag synchronization among cortical areas. Nature *385*, 157-161.
Schneidman, E., Berry, M.J., 2nd, Segev, R., and Bialek, W. (2006). Weak pairwise correlations imply strongly correlated network states in a neural population. Nature *440*, 1007-1012.
Schultz, W. (1998). Predictive reward signal of dopamine neurons. J Neurophysiol *80*, 1-27.

Stevenson, I.H., Rebesco, J.M., Hatsopoulos, N.G., Haga, Z., Miller, L.E., and Kording, K.P. (2009). Bayesian inference of functional connectivity and network structure from spikes. IEEE Trans Neural Syst Rehabil Eng *17*, 203-213.

Strogatz, S.H. (2001). Exploring complex networks. Nature *410*, 268-276.

Trappenberg, T. (2006). Fundamentals of computational neuroscience (Oxford, UK, Oxford University Press).

Truccolo, W., Hochberg, L.R., and Donoghue, J.P. (2010). Collective dynamics in human and monkey sensorimotor cortex: predicting single neuron spikes. Nat Neurosci *13*, 105-111.

Uhlhaas, P.J., Pipa, G., Lima, B., Melloni, L., Neuenschwander, S., Nikolic, D., and Singer, W. (2009). Neural synchrony in cortical networks: history, concept and current status. Front Integr Neurosci *3*, 17.

Uva, L., and de Curtis, M. (2005). Polysynaptic olfactory pathway to the ipsi- and contralateral entorhinal cortex mediated via the hippocampus. Neuroscience *130*, 249-258.

Watanabe, Y., Kajiwara, R., and Takashima, I. (2009). Optical imaging of rat prefrontal neuronal activity evoked by stimulation of the ventral tegmental area. Neuroreport *20*, 875-880.

Williams, Z.M., and Eskandar, E.N. (2006). Selective enhancement of associative learning by microstimulation of the anterior caudate. Nat Neurosci *9*, 562-568.

Yeung, N., Cline, M.S., Kuchinsky, A., Smoot, M.E., and Bader, G.D. (2008). Exploring biological networks with Cytoscape software. Curr Protoc Bioinformatics *Chapter 8*, Unit 8 13.

Zito, K., and Svoboda, K. (2002). Activity-dependent synaptogenesis in the adult Mammalian cortex. Neuron *35*, 1015-1017.

FIGURE LEGENDS

Figure 1. Neural population recordings and bottom-up control design. (A) Schematic illustration of the experimental set-up. The primates viewed a monitor which displayed a target randomly positioned in one of two possible locations over multiple trials. At the same time, neural recordings were performed from multiple neurons within the frontal cortex one of which was randomly selected to control the animal's receipt of reward. On a trial-by-trial basis, receipt of reward was dependent on whether the firing rate threshold reached by the neuron correctly corresponded to the location of the displayed. The mapping between firing rate threshold, target location and reward was chosen randomly at the beginning of each session. (B) Example of the firing rate and individual spikes (below) of the selected effector neuron (cell #2 in this case) over the course of two trials. In this session a bottom displayed target and upper firing rate threshold corresponded to receipt of reward. The colored arrows indicate the time when the neuron's firing rate reached its respective upper versus lower thresholds. (C) Permutation of possible firing rate thresholds, target locations and reward mappings for this session (i.e. 4 possibilities *per* session or 8 possibilities *across* sessions for which correspondences may be reversed). Top versus bottom targets were displayed randomly.

Figure 2. Enhancement in receipt of reward. (A) Mean performance (percent of trials in which reward was given; solid line) and standard error (dashed line) for all sessions in which learning was present. The *inset* exhibits performance and associated confidence bounds for a single effector neuron over one session. The arrow indicates the trial in which learning criterion was first reached. (B) Distribution of change in performance comparing beginning to peak performance during asymptote for the 32 sessions in which learning was present.

Figure 3. Modeling neural network activity. (A) Schematic illustration of the population-wide modeling. (B) An example of the interpolated time varying firing rate,

fitted self-correlated spike filter and one cross-correlated interaction filter onto a single cell recorded at baseline. Statistically significant time lagged interactions are shaded in gray. For self-correlations, the shaded time lag at 0-10 ms corresponds to the cell's early negative refractory period whereas, for cross-correlations, the shaded time lag at 10-50 ms corresponds to an excitatory time-delayed interaction with another cell in the population.

Figure 4. Examining concomitant changes in population activity. (A) Two scenarios in which enhanced correlated activity may be influenced by a common extrinsic source (e.g. from cells in the network other than those recorded). The *left* schematic illustrates a global increase in population activity and the *right* illustrates co-variations in firing rates produced by a common source. (B) Mean difference between the time taken to reach threshold during baseline and asymptote for all effector cells. (C) Example of the firing activity of an effector cell (black) during two trials; the *top* one in which the upper threshold is reached and the *bottom* one in which the lower threshold is reached. The firing activity of a simultaneously recorded cell in the population is shown in gray. The colored arrows indicate the time when the effector neuron's firing rate reached its respective upper versus lower thresholds. (D) Pearson's correlation coefficients (r) between the firing rate of the effector neuron (black) and all other cells in the population (gray) over time during a representative session. (E) Cross-correlogram of spiking between the two neurons before (*above*) and after (*below*) shuffling their spikes within the same successive windows used to calculate their firing rates. The dashed horizontal line indicates the 3 standard deviation threshold for significance.

Figure 5. Development of targeted interactions within the population. (A) Proportional change in firing rates, self-correlations and cross-correlations between baseline and asymptote. White bars indicate proportional changes for the population and gray bars indicate proportional changes for the effector neurons. The asterix indicates significance (t-test; $p < 0.05$). (B) Differences in the percentages of interactions (cross-correlations) between the baseline and asymptote periods across all sessions. Gray bars indicate difference in the percentage of interactions for effector cells per session, and

white bars indicate difference in the percentage of interactions averaged per session across the population. The arrows indicate the median for each distribution. (C) Cumulative distribution of connectivity changes calculated using pseudo-effector GLM resamplings. The black shaded area indicates the proportional probability that a randomly selected cell within the population displays the same or greater increase in connectivity as the true effector cell. (D) Changes in the network structure based on graph-theoretic measures. The *left* graph demonstrates the relative change in hub coefficients for effector cells (gray) and the population (white), whereas the *right* graph demonstrates the relative change in clustering coefficients for effector cells (gray) and the population (white). Below, are schematic illustrations depicting the general difference between a network hub (*left*) and a network cluster (*right*).

Figure 6. Temporal dependence of effector cell interactions on profit. (A) Mean performance and standard error of effector cells recorded during two sessions, but with a different effector (not shown) chosen for the second session. The performances shown for both sessions are that of the first effector neurons. (B) Percentage of cross-correlated interactions during the baseline and asymptote periods for effector neurons that previously controlled target selection only during the first session, measured across all four periods. The asterix indicates statistical significance between asymptote and baseline (t-test; $p < 0.05$). (C) Differences in the percentages of cross-correlated interactions between the baseline and asymptote periods during the inter-trial-interval when no sensory input was present. Gray bars correspond to effector cells and white bars correspond to the population. The arrows indicate the median of each distribution.

Figure 7. Spatiotemporal properties of effector cell interactions. (A) Cumulative distribution of connectivity changes calculated using pseudo-effector GLM resamplings for local (within area) and distant (between area; i.e. DLPFC-PMd) interactions. The black shaded area indicates the proportional probability that a randomly selected cell within the population displays the same or greater increase in connectivity as the effector cell per interaction type. (B) Distribution of time-lags for all local (solid lines) and distant (dashed lines) interactions, as a percentage of all significant interactions, during

baseline (*left*) and asymptote (*right*). The blue area indicates the time lags corresponding to putative monosynaptic couplings based on spike interval delays (see Text). (C) Connectivity of a sample population recorded during baseline (*left*) and asymptote (*right*) periods. The effector cell is positioned in the center and shaded gray to aid with visualization. Green circles indicate cells recorded in the DLPFC and white and gray circle(s) indicate cells recorded in the PMd. Lines indicate presence of significant interactions between cells, with arrows indicating the time-delayed direction of driving. Solid lines indicate local interactions within the same area whereas dashed lines indicate distant interactions between areas (cells displaying no connectivity are not depicted).

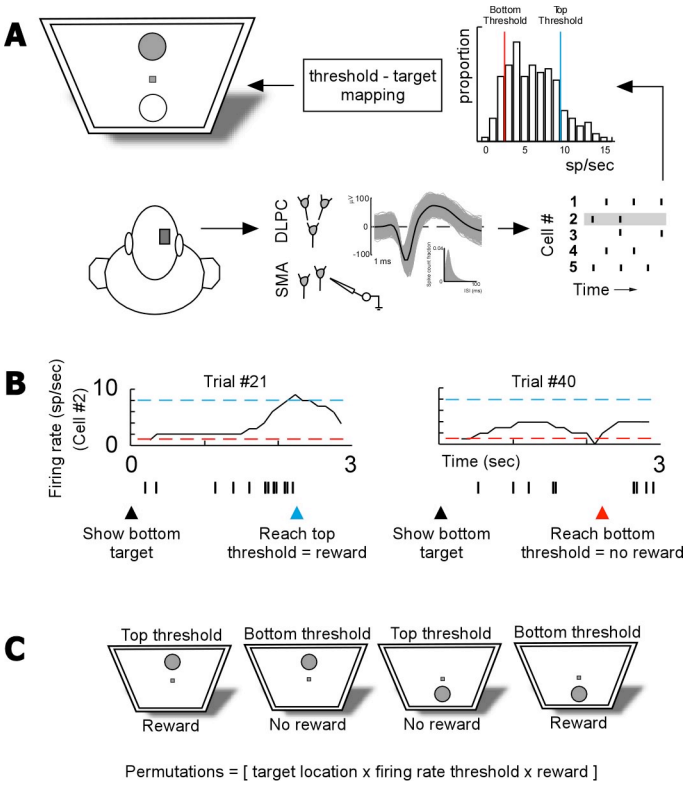
A
threshold - target mapping
Bottom Threshold
Top Threshold
proportion
sp/sec
DLPC
SMA
Cell #
Time
B
Firing rate (sp/sec) (Cell #2)
Trial #21
Trial #40
Time (sec)
Show bottom target
Reach top threshold = reward
Show bottom target
Reach bottom threshold = no reward
C
Top threshold
Bottom threshold
Top threshold
Bottom threshold
Reward
No reward
No reward
Reward
Permutations = [target location x firing rate threshold x reward]

Figure 1.

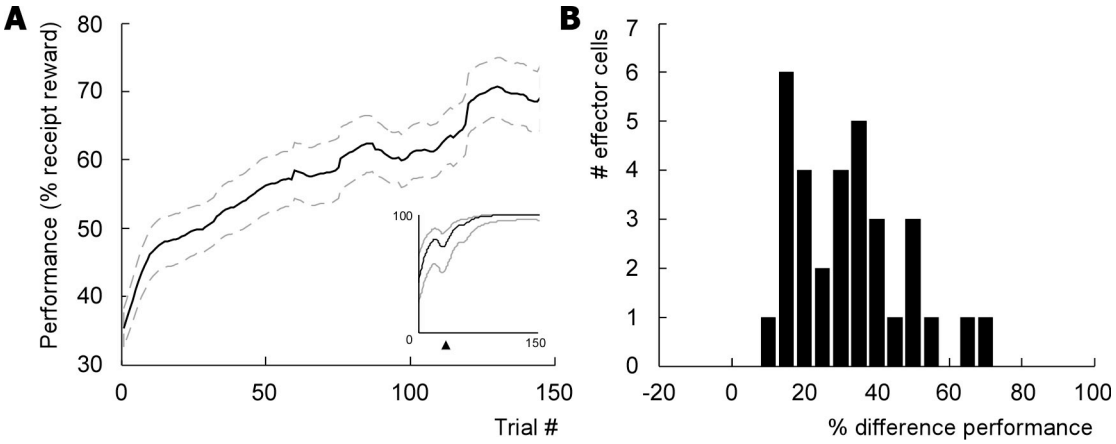
A
Performance (% receipt reward)
80
70
60
50
40
30
0
50
100
150
Trial #
100
0
150
B
effector cells
7
6
5
4
3
2
1
0
-20
0
20
40
60
80
100
% difference performance

Figure 2.

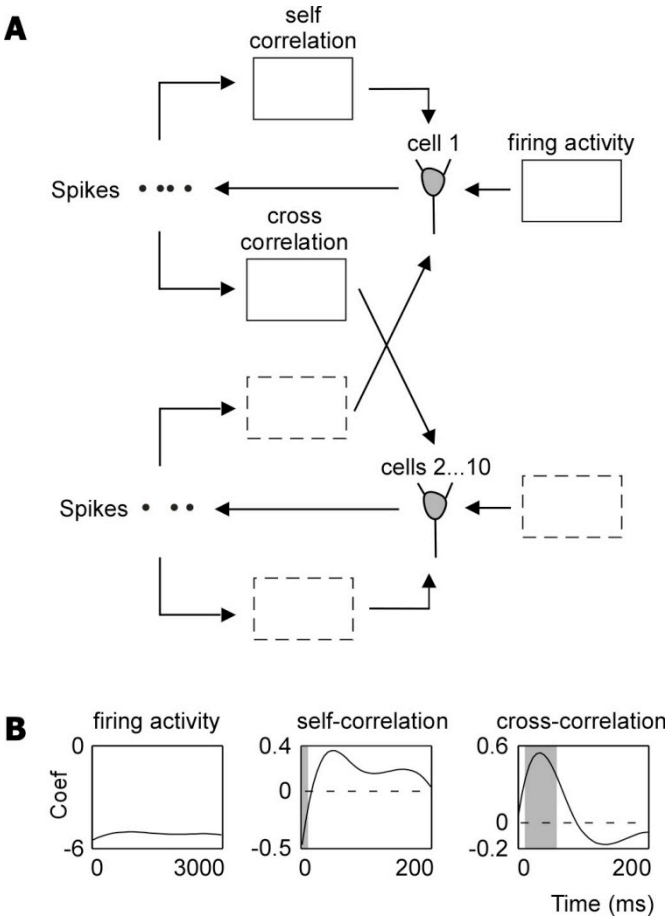
A
self
correlation
cell 1
firing activity
Spikes
cross
correlation
cells 2...10
Spikes
B
firing activity
self-correlation
cross-correlation
Coef
0
-6
0
3000
0.4
0
-0.5
0
200
0.6
0
-0.2
0
200
Time (ms)

Figure 3.

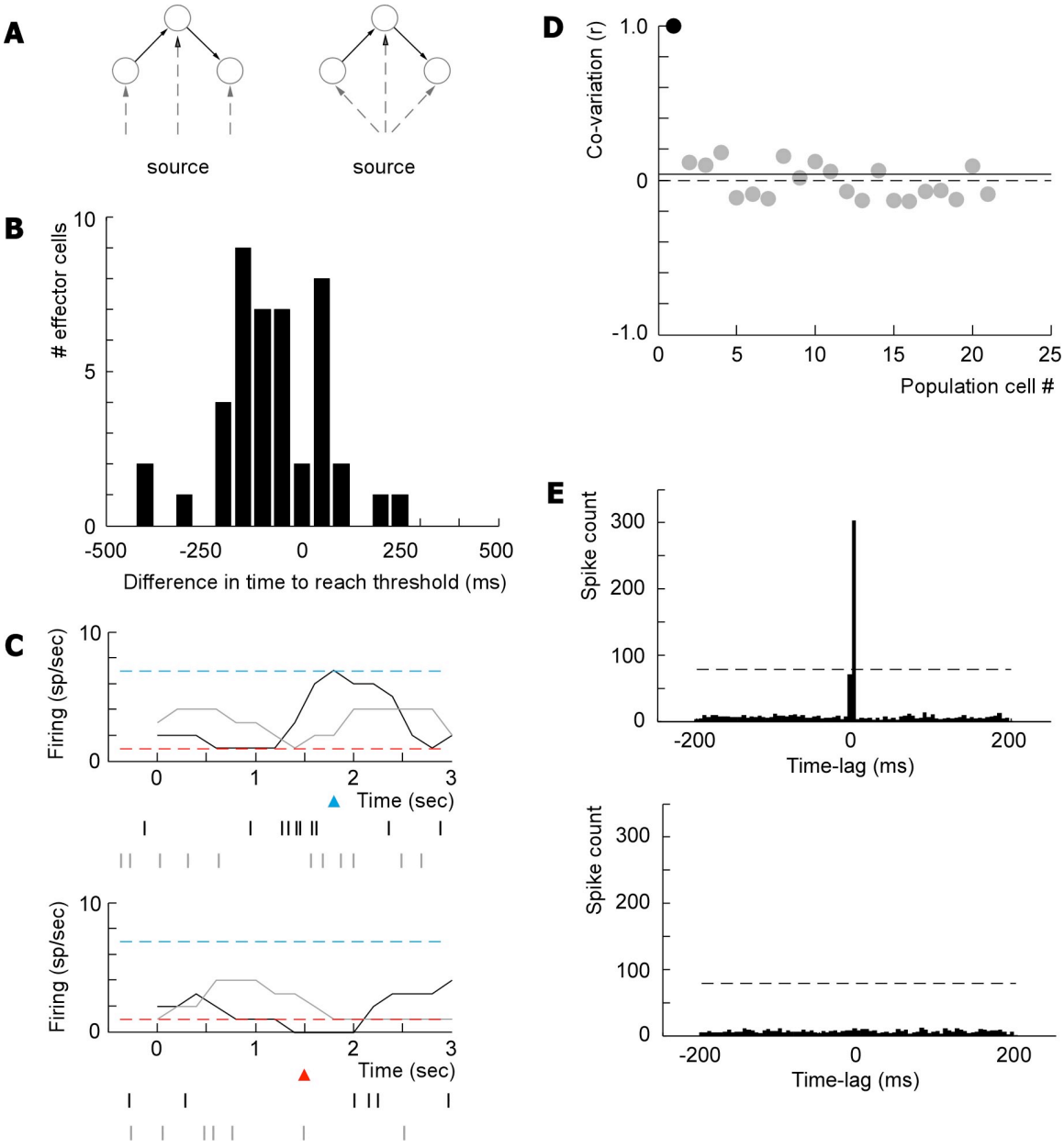
A
source
source
B
effector cells
10
5
0
-500
-250
0
250
500
Difference in time to reach threshold (ms)
C
Firing (sp/sec)
10
0
0
1
2
3
Time (sec)
Firing (sp/sec)
10
0
0
1
2
3
Time (sec)
D
Co-variation (r)
1.0
0
-1.0
0
5
10
15
20
25
Population cell #
E
Spike count
300
200
100
0
-200
0
200
Time-lag (ms)
Spike count
300
200
100
0
-200
0
200
Time-lag (ms)

Figure 4.

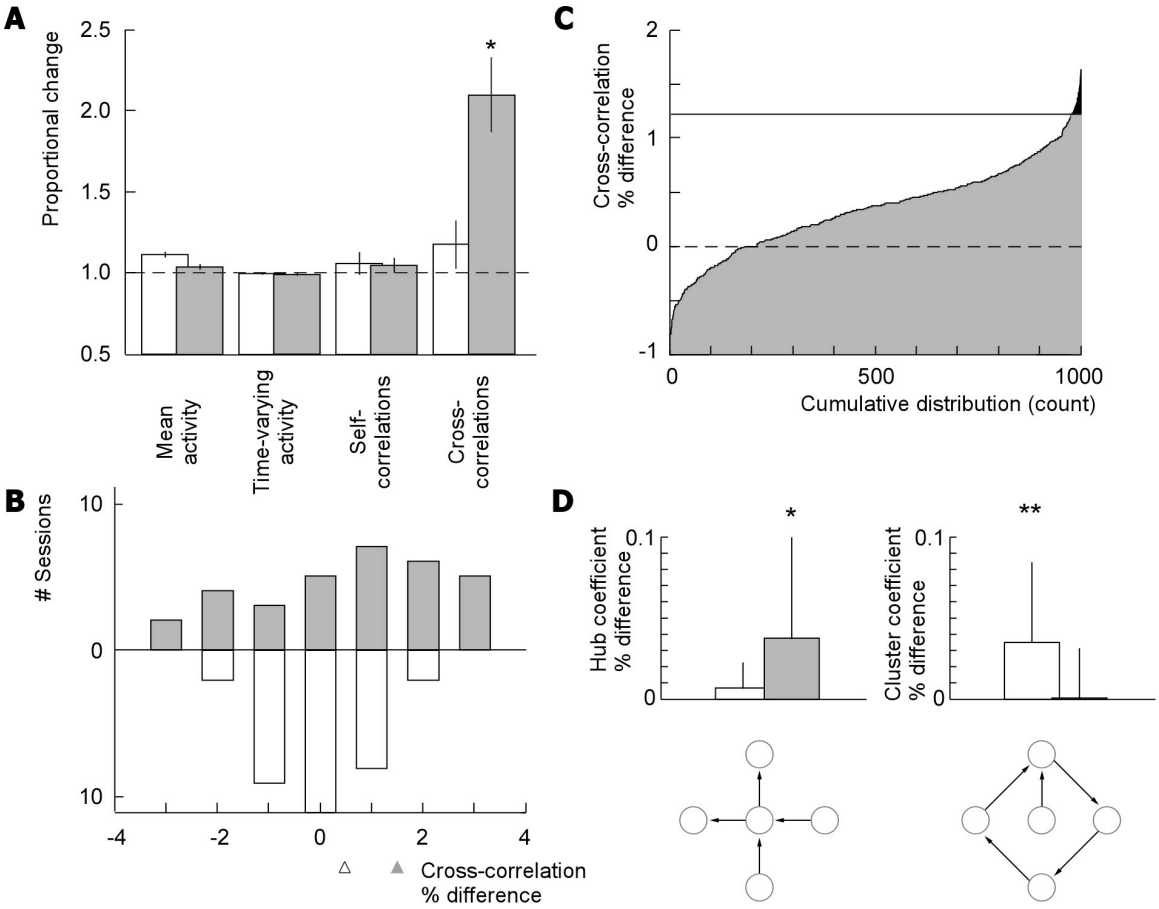
A
Proportional change
2.5
2.0
1.5
1.0
0.5
*
Mean activity
Time-varying activity
Self-correlations
Cross-correlations
C
Cross-correlation % difference
2
1
0
-1
0
500
1000
Cumulative distribution (count)
B
Sessions
10
0
10
-4
-2
0
2
4
Cross-correlation % difference
D
Hub coefficient % difference
0.1
0
*
Cluster coefficient % difference
0.1
0
**

Figure 5.

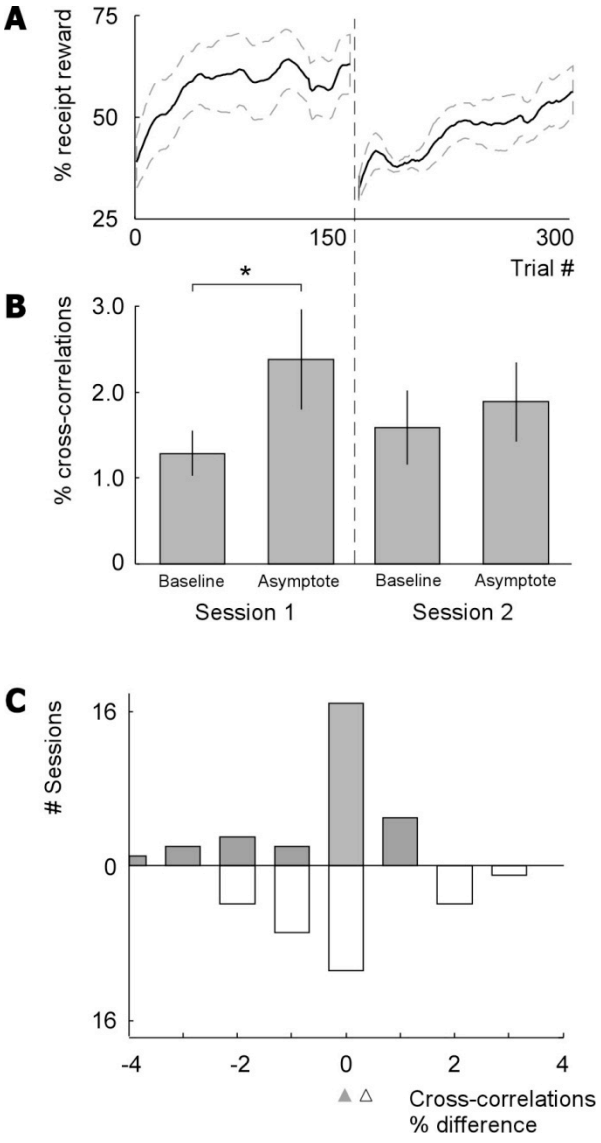
A
% receipt reward
75
50
25
0
150
300
Trial #
B
% cross-correlations
3.0
2.0
1.0
0
*
Baseline
Asymptote
Baseline
Asymptote
Session 1
Session 2
C
Sessions
16
0
16
-4
-2
0
2
4
Cross-correlations
% difference

Figure 6.

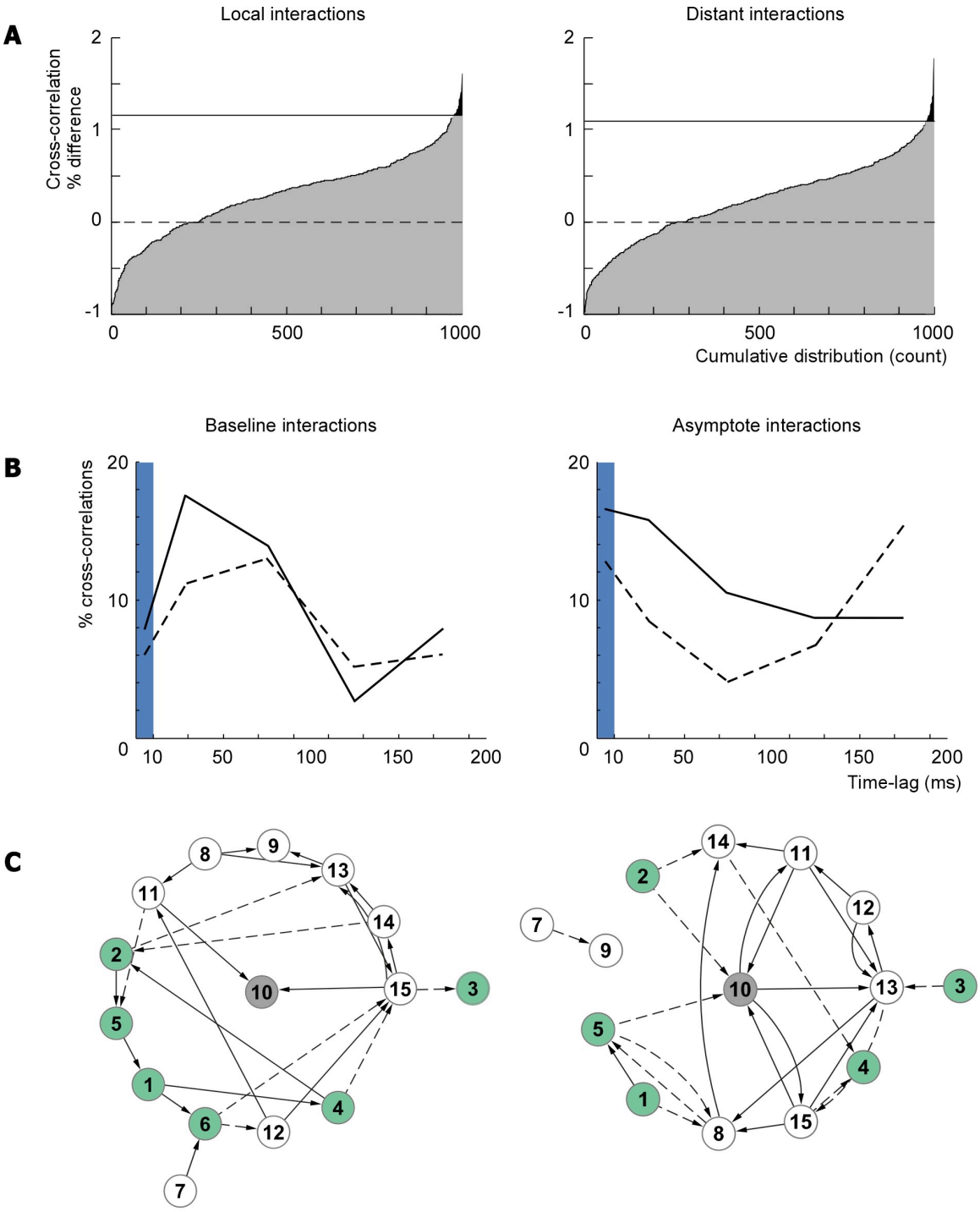
A
Local interactions
Distant interactions
Cross-correlation
% difference
2
1
0
-1
0
500
1000
Cumulative distribution (count)
B
Baseline interactions
Asymptote interactions
% cross-correlations
20
10
0
10
50
100
150
200
Time-lag (ms)
C
8
9
13
11
14
2
10
15
3
5
1
4
6
12
7
14
11
2
12
7
9
10
13
3
5
4
1
8
15

Figure 7.

Encoding through patterns: regression tree based neuronal population models

Robert Haslinger[1,2], Gordon Pipa[3], Laura Lewis[2], Danko Nikolić[4,5,6], Ziv Williams[7] and Emery Brown[2,8]

1 Martinos Center for Biomedical Imaging, Massachusetts General Hospital, 149 13th Street, Suite 2301, Charlestown, MA 02129, USA

2 Massachusetts Institute of Technology (MIT), Department of Brain and Cognitive Sciences, 77 Massachusetts Ave., Cambridge, MA 02139, USA

3 Department of Neuroinformatics, University of Osnabrück, Albrechtstr. 28, 49069 Osnabrück, Germany

4 Department of Neurophysiology, Max Planck Institute for Brain Research, 60528 Frankfurt am Main, Germany

5 Frankfurt Institute for Advanced Studies, Johann Wolfgang Goethe University, 60325 Frankfurt am Main, Germany

6 Ernst Strüngman Institute (ESI) for Neuroscience in Cooperation with Max Planck Society. Frankfurt am Main, Germany

7 Massachusetts General Hospital, Department of Neurosurgery, 55 Fruit Street, Boston, MA 02114, USA

8 Massachusetts General Hospital, Department of Anesthesia and Critical Care, 55 Fruit Street, Boston, MA 02114, USA

April 17, 2012

Abstract

Although the existence of correlated spiking between neurons in a population is well known, the role such correlations play in encoding stimuli is not. That is in addition to correlations being present (noise correlations) are they also stimulus modulated so that precise patterns of spikes, synchronized across neurons, encode beyond what would be expected under an independent neuron assumption? We address this question by constructing pattern based encoding models. The challenge is that large populations may express an astronomical number of unique patterns and determining each individual patterns encoding properties is not feasible. We avoid this combinatorial problem using a dimensionality reduction approach based upon regression trees. Using the insight that some patterns may, from the perspective of encoding, be statistically indistinguishable, the tree divisively clusters the observed patterns into groups whose member patterns possess similar encoding properties. These groups, corresponding to the leaves of the tree, are much smaller in number than the original patterns, and the tree itself constitutes a tractable encoding model for each pattern. Our formalism can detect extremely weak stimulus driven pattern structure and is based upon maximizing the data likelihood, not making a priori assumptions as to how patterns should be grouped. Most importantly, by comparing pattern encodings with independent neuron encodings, one can determine if neurons in the population are driven independently or collectively. We demonstrate this method using multiple unit recordings from area 17 of anaesthetized cat in response to a sinusoidal grating and show that pattern based encodings are superior to that of independent neuron models. The agnostic nature of our clustering approach allows us to investigate encoding by the collective statistics that are actually present, rather than those (such as pairwise) that might be presumed.

Key words and phrases:

Neuronal Population Models, Patterns, Encoding, Clustering, Regression Tree,

Introduction

A central question of neuroscience is whether neurons within a population encode stimuli independently, or collectively, as complex spatio-temporal patterns of action potential activity. One way to address this is to compare encoding models based upon independent spike trains, with encoding models based upon patterns of spiking across the entire population. Stimulus encoding by individual neurons is routinely quantified by fitting probability models to their spike trains Brown et al. (2001); Truccolo et al. (2005). In contrast, quantifying how a neuronal population uses *patterns* of spikes to encode has been less feasible, if only because a population of N neurons may express any one of 2^N patterns at any given time. Although it is likely that the number of patterns M that are actually observed is far smaller ($M << 2^N$), an approach that directly models encoding by each individual pattern would still have far too many parameters to accurately fit.

Thus most techniques for modeling population spiking have either considered neurons as independent, or have considered neurons as coupled (through second or higher order connections), but with those couplings independent of the stimulus. Both Ising models Martignon et al. (2000); Schneidman et al. (2006); Tang et al. (2008), and GLM based conditional intensity function models with cross history terms Okatan et al. (2005); Pillow et al. (2008); Truccolo et al. (2010) are examples of the latter approach. These methods only address whether collective patterns are present (noise correlations) not whether they actually useful for encoding. An approach that can *directly* quantify the relation(s) between pattern and stimulus would be highly desirable. Further, recent studies have indicated that second order models are likely to be insufficient for capturing the pattern statistics of neural populations, particularly as the populations being recorded from grow in size Roudi et al. (2009); Ganmor et al. (2011). The goal of this paper is to define a pattern probability model and fitting methodology which captures the relation between stimulus and pattern and does so for pattern statistics of arbitrary order and neural populations of arbitrary size.

To construct a pattern encoding model we proceed via the insight that *some patterns may convey statistically identical information* about a stimulus. The brain is a highly redundant system which processes information in a manner robust to noise. Shadlen and

Newsome (1994); Jacobs et al. (2009); Chelaru and Dragoi (2008), Thus if one spike is dropped from a $N >> 1$ neuron pattern, it could matter for encoding, but very likely it does not and it is even more likely that one does not possess sufficient experimental data to make such a distinction anyway. Guided by this intuition we propose that patterns be grouped into a set of $C << M$ clusters. For all patterns within a given cluster, the effect of any stimulus upon the pattern probability is modeled identically. That is, instead of modeling each pattern parametrically, we make parametric models of the clusters.

Since it is possible that patterns which appear very different may encode similar information, we do not presume that spiking patterns which appear similar must be clustered together. Instead we use a regression tree to group patterns so that the data likelihood is maximized Breiman et al. (1984); Quinlan (1993). The regression tree recursively splits the observed patterns into subgroups such that the expression probability of each pattern within a subgroup covaries with the stimulus in an identical manner. Thus the grouping does not depend upon the patterns themselves being similar, although this can be reintroduced by adding a "regularization" term dependent upon the Hamming distance. By allowing the clusters to be defined by the data, rather than by one's preconceptions, pattern encoding models can be generated irrespective of the order of correlative (between neurons) statistics present in the data.

Regression tree generated pattern clusters constitute a true population encoding model. Using the data likelihood as a measure of encoding (goodness of fit) they can be used to determine if stimuli are encoded by independent neurons or by collectively driven patterns. We demonstrate this using both simulated populations and real multiple unit data from area 17 of an anesthetized cat. In the case of the simulated data, pattern probabilities are accurately recovered for large (60 neuron) populations expressing many (thousands) of unique patterns, even when the collective structure is weak compared to neurons' independent stimulus drive. In the case of the cat data, we find that a grating stimulus is better encoded by collectively driven patterns, than by independent neuron encodings.

1 Methods

In this paper we consider discrete time, "spatial" patterns of spikes across a neuronal population. That is, each pattern is an N element vector of ones and zeros corresponding to neurons which are, or are not, firing within a particular time bin (Figure 1A). For N neurons, 2^N patterns are possible, however in general only a subset $M << 2^N$ will be observed. Each observed pattern is assigned an ordinal numeral $m \in 1 \ldots M$ and this designation is used to define a multinomial process that describes the temporal evolution of the population spiking.

We wish to construct an encoding model of how each pattern's expression probability depends upon an arbitrary stimulus $\mathbf{s}(t)$. We use the form

$$P_m(\mathbf{s}(t)) = P_m^{null} P_m^{stim}(\mathbf{s}(t)) \tag{1}$$

P_m^{null} is the mean expression probability of pattern m and is independent of the stimulus. This is what a second order coupled model, such as an Ising model, would estimate and much effort has gone into accurately estimating such null models. Our goal is to determine how the pattern probabilities given by the null model are *modulated* due to stimulus drive. Towards this end, we will initially evaluate P_m^{null} by simply counting the number of times each pattern occurs (N_m), in a manner analogous to estimating a neuron's mean firing rate by counting spikes.

$$P_m^{null} = \frac{N_m}{\sum_m N_m} \tag{2}$$

Later we will generalize the null distribution so that all patterns are allowed some non-zero probability.

$P_m^{stim}(\mathbf{s}(t))$ modulates the expression probability of pattern m as a function of the stimulus. In general there will be too many patterns to accurately model this for each individual pattern m. Instead, we partition the patterns into a set of $C << M$ clusters whose member patterns have probabilities that covary identically with the stimulus (Figure 1 B). That is, for each pattern m in a cluster c, $P_m^{stim}(\mathbf{s}(t)) = P_c^{stim}(\mathbf{s}(t))$, Since it is not *a priori* clear how many clusters should be used or from which patterns they should be constructed, we use a logistic regression tree to perform a recursive binary partitioning of the patterns.

An example of such a tree is shown in Figure 1 C. The root node comprises all the patterns. The patterns are then split into two groups defining child nodes. Patterns within each of the child nodes have the same stimulus dependent probability term $P_{child}(\mathbf{s}(t))$. This is estimated using a logistic regression model which depends on the stimulus, and also the partitioning of patterns between child nodes. Optimum partitioning is achieved using an expectation maximization type algorithm (described below) which maximizes the data likelihood. Crucially, this algorithm does not depend upon the patterns in each child node "looking similar", although this can reintroduced by adding a prior (see below). The child nodes are then themselves split into additional child nodes and the process repeated until further splitting is no longer justified by the data. In this paper, the Bayesian Information Criterion (BIC) is required to be further minimized by each branching, although other criteria could be envisioned, see Hastie et al. (2009).

The "leaves" of the tree, i.e, the nodes at the edges, are the clusters $c \in 1 \dots C$ and the tree itself constitutes a stimulus dependent probability model for each leaf. The logistic regression fitted probabilities of the child nodes of each branching are conditioned on the probability of their parent node. Thus the probability $P_c(\mathbf{s}(t))$ that a member pattern from leaf c is observed is the product of the conditional probabilities of each node along the path connecting leaf c to the root of the tree.

$$P_c(\mathbf{s}(t)) = \prod_{i \in \ path} P_i(\mathbf{s}(t)|\text{parent nodes}) \tag{3}$$

where the product is over the nodes i along the path. The modulatory term $P_c^{stim}(\mathbf{s}(t))$ is then simply obtained by dividing by the mean leaf probability.

$$P_c^{stim}(\mathbf{s}(t)) = P_c(\mathbf{s}(t)) \frac{T}{\sum_{m \in c} N_m} \tag{4}$$

The regression tree can be thought of as a type of latent class model for which both the correct number of classes, and the partitioning of patterns between classes (clusters) must be determined.

1.1 Splitting Algorithm

At each branching we partition the patterns into child nodes using an expectation maximization type procedure that increases the log likelihood of the observed patterns. The

algorithm is shown schematically in Figure 2. First the patterns within a node are assigned randomly to two child nodes designated here as "+" and "-". Second, we fit a logistic regression model to the clustered patterns

$$P_{\pm}(t) \;=\; \frac{e^{\pm[\beta_0+\mathbf{X(t)}\beta]}}{1+e^{\pm[\beta_0+\mathbf{X(t)}\beta]}} \tag{5}$$

where P_+ and P_- are the node probabilities, $X(t)$ is a covariate matrix which parameterizes the stimulus and β_0 and β are the parameters.

Third, we fix the parameters of the logistic regression model and reassign the patterns between the child nodes so as to maximize the likelihood. It is shown in Appendix A, that the total log likelihood is maximized when each pattern m is assigned to the cluster which individually maximizes

$$\log \mathcal{L}_m = \sum_{t\in T_m} \log P_m(t) \tag{6}$$

where the sum is over the time bins T_m in which pattern m appears. (The full log likelihood is the sum of equation 6 over all patterns.) This sum can be calculated explicitly, although we also show in Appendix A that if the covariate matrix $X(t)$ is standardized so that each column has zero mean, then the log likelihood is equivalently maximized when each pattern m is assigned to child node $+$ if

$$\sum_{t\in T_m} X(t)\beta > 0 \tag{7}$$

and to child node $-$ if the sum is less than zero. [1] Thus the assignment of each pattern depends not upon its mean probability of expression (bias) but upon how that probability varies with changing stimuli.

The logistic regression model is then refit, the patterns reassigned, and this process is then iterated until patterns no longer shift between nodes, or an upper iteration bound is reached. If the final result minimizes the BIC then the split into child nodes is retained and they become new leafs on the tree. Otherwise the child nodes are removed from the tree. The whole procedure is then repeated at all leafs on the tree until the BIC is no longer minimized and the splitting terminates. In essence, the tree recursively partitions the stimulus space into smaller subspaces which are predictive of the patterns

[1] The pattern is assigned randomly if the sum equals zero exactly.

(see Figure 1D). See Appendix B for algorithmic benchmarking using various simulated data sets.

Generating Logistic Regression Trees

1. Assign each unique observed pattern to an ordinal numeral $1 \ldots M$ and generate a multinomial time series. Designate the group of all patterns as the root node of the tree.

2. Using the occupation probability of each pattern calculate the log likelihood (see equation 17) of the multinomial time series and the BIC.

3. Split the set of patterns $1 \ldots M$ into two groups (child nodes) by random assignment. Generate a binomial time series where 0 corresponds to the first group and 1 to the second.

4. Fit a logistic regression model to this binomial time series using the stimulus covariate matrix X.

5. Using equation 6, reassign each pattern to the group (node) which maximizes the log likelihood.

6. Iterate steps 4-5 until either no patterns are reassigned or an upper iteration bound is reached.

7. Recalculate the log likelihood and BIC according to equations 17 and 3. If the BIC post splitting is less than that pre splitting, the child nodes become new leafs on the tree. Otherwise, remove them.

8. Iterate steps 2-7 at all leaves until no additional leaves are generated.

1.2 Hamming "Regularization"

Due to insufficient observations, patterns which occur infrequently may be difficult to assign to a cluster based solely upon the data likelihood. To take advantage of any prior information one might have about how patterns should be grouped a "regularization" term can be added to the log likelihood. That is, instead of maximizing the log

likelihood (equation 6), it can be required that

$$\log \mathcal{L}_m + \eta\mathcal{R} = \sum_{t \in T_m} \log P_m(t) + \eta\mathcal{R} \tag{8}$$

be maximized for each pattern m within step 5 of the expectation maximization algorithm. η is a tunable regularization parameter, and $\mathcal{R}$ depends upon the prior chosen.

In this paper we use a regularization which depends upon the Hamming distance (binarized L1 norm) between patterns. That is, patterns which "look similar" tend to be grouped together.

$$\mathcal{R} = -\frac{T}{NM_c} \sum_{n \in c} D_H(n, m) \tag{9}$$

where $D_H(n, m)$ denotes the Hamming distance between patterns n and m, and the sum is over patterns currently assigned to the child node in question. The denominator NM_c is for convenience and normalizes out the experiment specific factors of total neuron number N and number of patterns M_c in each child node. Use of T in the numerator puts the regularization on the same scale as the log likelihood $\mathcal{L}_m$ and weighs the regularization more heavily (compared to $\mathcal{L}_m$ which scales as T_m) for patterns which occur infrequently. Thus patterns which occur infrequently tend to be grouped with others that are similar, while frequently occurring patterns are assigned based on likelihood maximization.

1.3 Generalization

As discussed above, the data set used to fit the regression tree most likely contains only $M << 2^N$ unique patterns. For fitting purposes, the probability of the other $2^N - M$ patterns is set to zero. However most likely these patterns do not have zero probability and it was simply due to chance (and finite data lengths) that they were not observed. Recall that the tree describes the training data pattern probabilities using a multiplicative form

$$P_m(\mathbf{s}(t)) = P_m^{null} P_m^{stim}(\mathbf{s}(t)) \tag{10}$$

Generalization the tree so that it describes the stimulus driven probabilities of all 2^N patterns requires three steps. 1) Assignment of a null probability to every possible

pattern. 2) Assignment of each unique pattern to a leaf on the tree (so that its covariation with the stimulus is determined. 3) Normalization of the full probability distribution so that $\sum_m P_m(s) = 1$ for all stimuli $\mathbf{s}$.

1) To generalize the null probabilities we replace the null distribution used for fitting the tree ($P_m = N_m/T$) with one that allows all the patterns non-zero probability, but is hopefully not too different from that used for fitting. One possibility is to simply use an Ising model (or other parametric model) for the generalized null distribution. A second is to use the Good-Turing estimator of the missing probability mass Orlitsky et al. (2003). This is estimated solely from the training data as the fraction of observations which only occur once.

$$\alpha_{GT} = \text{\# patterns occurring once}/T \tag{11}$$

The null probability of patterns observed in the training data is then updated as

$$P_m^{null} \rightarrow P_{m,GT}^{null} = (1 - \alpha_{GT}) P_m^{null} \tag{12}$$

Novel patterns are assigned a null probability that is a fraction of the missing mass, with that fraction is based upon an Ising (or other parametric) model assumption.

$$P_{m,GT}^{null} = \alpha_{GT} \frac{P_m^{ising}}{\sum_{m \notin train} P_m^{ising}} \tag{13}$$

This null probability distribution is normalized over all possible patterns, i.e. $\sum_m P_{m,GT}^{null} = 1$.

2) To generalize $P_m^{stim}(\mathbf{s}(t))$ to all patterns, each novel pattern is assigned to a leaf on the tree. Specifically, motivated by the Hamming regularization used above, it is assigned to the leaf composed of patterns with which which it is "closest" via minimum mean Hamming distance.

$$c = \text{argmin}_c \frac{1}{M_c} \sum_{m' \in c} D_H(m, m') \tag{14}$$

and we set $P_m^{stim}(\mathbf{s}(t)) = P_c^{stim}(\mathbf{s}(t))$ as above. Intuitively, in the absence of any information about how the novel pattern covaries with the stimulus we revert to our "prior" that patterns that look similar are grouped together.

3) Finally, the full stimulus dependent probabilities are normalized by simple division.

$$P_m(\mathbf{s}(t)) \rightarrow P_m^{norm}(\mathbf{s}(t)) = \frac{P_m(\mathbf{s}(t))}{\sum_m P_m(\mathbf{s}(t))} \tag{15}$$

This last step ensures normalization for all possible stimuli $\mathbf{s}$, however it is important to note that since $P_m^{stim}\mathbf{s}(t)$ is a modulatory term, $\int P_m(\mathbf{s}(t))\rho(s)ds = 1$ even prior to this step no matter which leaf each pattern is assigned to. $\rho(s)$ is the distribution of stimuli in the training set. If the missing mass is low, the normalization of step 3 will be minimal. In this case, one can skip not only this final step, but also the majority of step 2, only assigning to leafs novel patterns which are actually observed in the test set, and greatly diminishing computation time. For example the V1 cat data presented in the Results section below has a missing mass of 0.006, and the final normalization step does not change the pattern probabilities or test data log likelihood in any appreciable fashion.

1.4 Bagging Trees

The regression tree algorithm uses random initial assignments of patterns to sub nodes. Thus each implementation will generate a different tree. This variability is an advantage because the stimulus dependent probabilities P_m^{stim} can be averaged across trees, or *bagged* Breiman (1996). Specifically, if N different trees are generated from the data, and the stimulus dependent probability for pattern m provided by tree n is $P_{m,n}^{stim}$, then the bagged estimate is

$$P_{m,bag}^{stim} = \frac{1}{N}\sum_{n=1}^{N} P_{m,n}^{stim} \tag{16}$$

The full pattern probability is $P_{m,bag}(s(t)) = P_m^{null} P_{m,bag}^{stim}(s(t))$. Bagging improves generalizability to a validation data set by emphasizing structure conserved throughout the data set and reducing the risk of overfitting.

1.5 Goodness of fit: patterns versus independent neurons

To determine if stimuli are encoded collectively or by independent neurons we compare the goodness of fit of the pattern model with that of independent neuron models (one

for each neuron) regressed upon the same stimulus covariate matrix $X(t)$. We use the log likelihood (deviance) of an independent test data set to quantify goodness of fit . An increase in log likelihood is analogous to a reduction in mean squared error (for Gaussian variables it is identical). It is important to distinguish between patterns being present in the data, and patterns actually covarying with the stimulus. Our formalism uses a multiplicative form for each pattern probability $P_m(\mathbf{s}(t)) = P_m^{null} P_m^{stim}(\mathbf{s}(t))$ (equation 1). This allows the data log likelihood to be split into two terms with these different physical interpretations.

Denoting m_t as the pattern m that is observed at time t, the log likelihood may be written as:

$$\begin{aligned} \log \mathcal{L} &= \sum_{t=1}^{T} \log \left[P_{m_t}(\mathbf{s}(t)) \right] = \sum_{t=1}^{T} \log \left[P_{m_t}^{null} P_{m_t}^{stim}(\mathbf{s}(t)) \right] \\ &= \sum_{t=1}^{T} \log \left[P_{m_t}^{null} \right] + \sum_{t=1}^{T} \log \left[P_{m_t}^{stim}(\mathbf{s}(t)) \right] \\ &= \log \mathcal{L}^{null} + \log \mathcal{L}^{stim} \end{aligned} \tag{17}$$

The first term only depends upon the mean expression probability of the pattern. If it is larger than what would be expected under an independent neuron assumption, then structured patterns are present in the data. The second term depends upon how the stimulus modulates each pattern's expression probability. If it is larger than what would be expected under an independent neuron assumption, then in addition to being patterns being present, their probabilities are modulated by the stimulus in a way that can not be described by independent neurons.

Analogous null, and stimulus dependent pattern probabilities can be calculated for a set of N independent neuron models. If each neuron is modeled with its own discrete time, stimulus dependent, probability of spiking (conditional intensity function) $\lambda_n(\mathbf{s}(t))$, the probability of pattern m is given by a product.

$$P_{m,indep}(s(t)) = \prod_{n=1}^{N} \lambda_n(s(t))^{\theta_{m,n}(t)} (1 - \lambda_n(s(t)))^{1-\theta_{m,n}(t)} \tag{18}$$

where $\theta_{m,n}(t) = 1$ if pattern m_t has a spike for neuron n and is 0 otherwise. In analogy with equation 1, $P_{m,indep}^{stim}(s(t)) = P_{m,indep}(s(t))/P_{m,null}$ where

$$P_{m,indep}^{null} = \frac{1}{T} \sum_{t=1}^{T} P_{m,indep}(s(t)) \tag{19}$$

is the mean probability of the pattern (similar to the mean firing rate of a neuron). These probabilities can be used to calculate $\log \mathcal{L}_{indep}^{null}$ and $\log \mathcal{L}_{indep}^{stim}$ for comparison with the regression tree pattern model.

It should be noted that $P_{m,indep}^{null}$ is not the same as the pattern probability that would be calculated for a mean firing rate model. This is given by

$$P_{m,frate} = \prod_{n=1}^{N} \bar{\lambda}_n^{\theta_{m,n}(t)} (1 - \bar{\lambda}_n))^{1-\theta_{m,n}(t)} \tag{20}$$

where $\bar{\lambda} = (1/T) \sum_t \lambda(s(t))$. Essentially the difference is that the product of a sum is not the same as the sum of a product.

2 Results

We first present results from simulated data to illustrate four points. First, that the algorithm can recover the correct pattern groupings. Second, that bagging can substantially improve goodness of fit. Third, that Hamming regularization can improve generalizability to an independent test set. Fourth, that our method can distinguish between neurons driven independently by stimuli, and neurons driven collectively. We then apply the regression tree algorithm to population activity collected in V1 of an anesthetized cat stimulated by a moving grating and show that the grating stimulus is encoded collectively as patterns.

2.1 Algorithm Recovers Correct Pattern Groupings

To demonstrate that the algorithm accurately groups patterns according to their encoding properties, we generated "ordinal patterns" from a logistic regression tree (shown in Figure 3 A). In the absence of Hamming regularization nothing in the formalism or regression tree algorithm requires that the patterns be generated by spiking neurons. Instead they can simply be designated by an ordinal list $m \in 1 \ldots M$ of distinct observation types, which is what we do in this subsection.

The tree used to generate the data has 20 leaves, each containing 20 patterns such that the total number was $M = 400$. Given a particular leaf, each pattern within it was equally probable. The leaf probabilities were generated using a covariate matrix X with

1 constant bias column, and 24 time varying columns. The time varying columns were chosen to mimic processes occurring at different time scales and had the form.

$$x_t = cos(2\pi ft + \phi_0) \tag{21}$$

where f was chosen randomly between 0.1 and 10 Hz, and ϕ_0 was chosen randomly between 0 and 2π. The parameters β defining the logistic regression model of each branching were chosen randomly within $[-0.5, 0.5]$. We simulated 200 seconds of data at 1 msec resolution from this model.

Since the initial assignment of patterns at each branching is random, the regression tree algorithm generates a different tree each time it is applied. We show one tree fit to the data in Figure 3 B. Although the fitted tree looks different from the model, it has 20 leaves, and each leaf in the fitted tree is isotropic (is comprised of the same patterns) to a leaf on the model tree. It is well known that different regression trees can have similar or even identical input output mappings Ripley (1996). For example if there are three true leaves, splitting the root node $\{$A,B,C$\}$ into children $\{$A,B$\}$ and $\{$C$\}$ and then $\{$A,B$\}$ into $\{$A$\}$ and $\{$B$\}$ produces the same partition as splitting $\{$A,B,C$\}$ first into $\{$A$\}$ and $\{$B,C$\}$ and then $\{$B,C$\}$ into $\{$B$\}$ and $\{$C$\}$. What matters is not the exact structure of the tree, but how well it fits the data. Measured from the baseline null model the fitted tree accounts for 96% of the log likelihood that the tree used to simulate the data does [2]. Moreover the mean correlation coefficient between the pattern probabilities as defined by the model and as deduced by the regression tree is $r = 0.94$ ($p < 0.001$). Finally in Figure 3C we show the probability of four of the leaves (over a 2 second example epoch) both as defined by the model tree, and as predicted by the fitted tree and demonstrate extremely accurate agreement.

2.2 Bagging Trees

The fact that the algorithm produces different trees for different implementations, is an advantage because the pattern probabilities predicted by different trees can be averaged or *bagged* to improve goodness of fit. In this example we used the same (as above)

[2]We calculate the log likelihood fraction as the ratio of the stimulus driven log likelihoods (eq 17) of the fitted regression tree and the model. That is $\Delta\mathcal{LL} = (\mathcal{LL}_{tree} - \mathcal{LL}_{null})/(\mathcal{LL}_{model} - \mathcal{LL}_{null})$.

grouping of 400 ordinal patterns into 20 groups, but used a multinomial logit model to generate the group probabilities (Figure 4A)

$$P_c(X_t) = \frac{e^{X_t \beta_c}}{\sum_{c'=1}^{C} e^{X_t \beta_{c'}}}; \tag{22}$$

The covariate matrix X was identical to that of the above section, and the parameters β were again chosen randomly within $[-0.5, 0.5]$.

Figure 4 B shows two different twenty leaf regression trees, generated by different applications of the algorithm. Each has an isotropic mapping of leaves to multinomial logit groups. Both trees account for 83% of the log likelihood of the original the multivariate logit model, and the mean correlation coefficients (over all 400 patterns) between the fitted and model probabilities are $r = 0.81$ and $r = 0.82$ ($p < 0.001$ for both) respectively. The time varying probabilities (over a representative 1 second epoch) for six patterns (each from a different group) as defined by the model and the first fitted tree, are shown in Figure 4C. This level of goodness of fit is repeatable as shown by the log likelihoods and correlation coefficients of 50 different trees in Figure 4D.

The individual trees have poorer fit than in the previous example, because they were not generated from a model with tree like structure. However the fit can be substantially improved by bagging (averaging pattern probabilities) across all 50 trees (black lines in 4C and vertical dashed lines in 4D). The bagged trees account for 90% of the log likelihood, and the mean correlation coefficient between the bagged and model pattern probabilities increases to 0.93. Figure 4E shows how these goodness of fit measures increase with the number of trees used for bagging. Thus even when data is not generated by a tree, multiple bagged trees can still provide a good probability model.

2.3 Simulated Cell Assemblies: Hamming Regularization

For large neuronal populations many of the patterns which are observed may occur only rarely, making it difficult to accurately group them. Here, regularization can be used to stabilize the regression trees and improve generalizability to an independent test data set. We simulated neural spiking patterns from 60 cells which normally fired independently at 40 Hz. However, for certain values of a simulated input stimulus groups of 6 cells had a high probability of firing together, expressing 4 or more spikes

(out of 6). Patterns of 3 spikes were not allowed. Given a group or "cell assembly" each of its member patterns (22 total per group) were equally probable. See Figure 5A for a raster plot in which patterns corresponding to "cell assemblies" are in grey.

Formally, the data was generated using a multinomial logit model (see equation 22) with 11 groups. 10 of these were "cell assembly" groups, the 11th was the "independent neuron" group. The same 25 column stimulus covariate matrix X as above was used. The parameters of the multinomial logit were chosen so that independent neuron group was most probable (independent firing 92% of the time), and the cell assembly group were expressed more rarely (each roughly 0.8% of the time). Specifically β_c was chosen randomly within $[-1, 1]$ except for the bias parameter which was chosen to be 4 for the independent neuron group and -4 for the other 10 groups. This model has 2051 unique patterns, and from it we simulated 100 seconds (at 1 msec resolution) of training data, and 50 seconds of test data. The total number of patterns expressed by the training data, and the percentages of time for which they were expressed are both shown in Figure 5 B as a function of the number of spikes in the pattern.

In Figure 5C we show an 11 leaf tree which recovers the correct pattern groupings, and in Figure 5D we show compare 4 pattern probabilities from the model with those deduced by bagging over 50 trees. The tree and probabilities shown were generated by *regularizing* the node splitting algorithm. As discussed in the Methods, a penalization was introduced so that patterns which "look similar" via Hamming distance tended to be grouped together. The strength of this regularization was controlled using a regularization parameter η. For small values of η the algorithm produced trees with too many leaves and did not group the patterns properly (Figure 5E *upper*). This is likely due to both large number of patterns and the fact that many of them were observed infrequently. However for a regularization of approximately $\eta = 10^{-2}$, the the correct number of leaves was recovered and the patterns were grouped correctly. This was the case across the majority of fitted trees as evidenced by the mean (within group) Hamming distance between patterns approaching that the original model for all fitted trees (Figure 5E *lower*).

The optimal value for η is that which maximizes the test data log likelihood (Figure 5F *lower*). Specifically, we used 50 logarithmically spaced values of $\eta \in [10^{-4}, 10^{-1}]$. Increased regularization degrades the log likelihood of the training data, however this

is due to over fitting when no regularization is used (Figure 5F *upper*). In contrast, the log likelihood of the test data is maximized near $\eta = 10^{-2}$, at the same regularization that recovers the correct pattern groupings. In addition to showing the mean log likelihoods across 50 individual trees (black) we also show the log likelihoods when trees are bagged (grey). Validation data maxima are denoted by grey and black dots. In summary, regularization of the regression tree algorithm, motivated by the idea that patterns that "look similar" should be clustered together, produces a more parsimonious, and in this case at least, more accurate representation of the data.

2.4 Distinguishing Patterns from Independent Neurons

Our algorithm can distinguish between neurons being driven independently, and neurons being driven collectively. To demonstrate this we used the same 60 neuron cell assembly model as above, but varied the percentage of time during which cell assemblies were observed. Specifically, we varied the multinomial logit model bias parameters, but kept the form of the multinomial logit model, the patterns forming cell assemblies, the covariate matrix $X(s(t))$ and the parameters corresponding to stimulus modulation (non-bias parameters) fixed.

What was crucially different was that when neurons fired independently they did so with rates that were stimulus modulated (the previous section used a mean rate). To simulate independent firing for each neuron n we modeled its independent stimulus driven spike probability (per time bin) with a logistic regression model.

$$\lambda_n(t) = e^{X\alpha_n}/(1 + e^{X\alpha_n}) \tag{23}$$

and randomly drew spikes according to these stimulus dependent probabilities. The parameter vectors α_n were chosen so that the 60 neurons had mean firing rates ranging between 3 and 10 Hz (population mean 5.5 Hz). These rates were strongly modulated by the stimulus, with standard deviations ranging between 10 and 30 Hz (population mean standard deviation was 17.7 Hz). In summary, for each time t we used the multinomial logit model to determine if the population was firing collectively (in which case we drew a pattern as in the previous section) or if it was firing independently (in which case we drew independent spikes according to equation 23. We simulated 100 s of training data

and 50 s of test data at 1 ms resolution. Both bagged (un-regularized) regression trees, and independent neuron models (eq 23) were fit to the simulated population spikes.

We then compared (in Figure 6) both the full test data log likelihoods, and the stimulus driven log likelihoods of the tree based, and independent neuron based models (see equation eq:loglikelihood). The full log likelihoods were higher for the trees (the superior goodness of fit indicating that the algorithm could detect the *existence* of collective patterns) when the percentage of time during which cell assemblies appeared was greater than 0.7%. It could detect that these patterns were *driven* by the stimulus when the cell assemblies appeared more than 2% of the time. That is, the stimulus driven log likelihood was greater for the trees than the independent neuron models. To ensure that this was not a function of the exact manner in which the patterns were assigned to cell assemblies, we randomized the patterns with 4 or more spikes across cell assemblies (while retaining 10 assemblies total and 22 patterns per assembly) and repeated the calculations. The percentages for which pattern existence and pattern drive could be detected changed only marginally (to 0.8% and 3.5% respectively. These results demonstrate that even when the collective patterns being driven by stimuli appear only rarely (and the independent drive to neurons is strong) the collective structure can still be detected.

3 V1 Cat Population Data

To demonstrate the algorithm on real neurobiological data, we used 20 neurons recorded in area 17 of an anesthetized cat in response to a drifting sinusoidal grating. The neurons had firing rates which ranged between 2 and 21 Hz, with median rate 7.4 Hz. A raster plot is shown in Figure 7A. The grating stimulus had a temporal frequency of 0.83 Hz and were presented in repeated trials of 3.5 seconds each. 20 trials for each of 12 different directions (30 degree spacing) were recorded. We partitioned these into 15 training data trials and 5 test data trials for each direction. We concatenated the last 3 seconds of each trial (to eliminate transients from the stimulus onset) and discretized the spikes using 3 ms bins. This resulted in 2600 unique patterns in the training data and 1213 patterns in the test data, 860 of which were also in the training data (Figure 7 B). Further experimental details are in Appendix C and also see Yu et al. (2008).

We first modeled the response of each neuron independently using a standard logistic regression model.

$$\lambda_n(\phi, t) = \frac{e^{X(\phi,t)\alpha_n}}{1 + e^{X(\phi,t)\alpha_n}} \tag{24}$$

$\lambda(\phi, t)$ is the discrete time probability that the modeled neuron has a spike in time bin t. $X(\phi, t)$ parameterized the grating stimulus as a function of both grating direction ϕ and the time since stimulus onset t. α_n is a parameter vector fit individually to each neuron's spikes.

In defining $X(\phi, t)$ we noted that most neurons exhibited periodic responses (reflecting the sinusoidal grating stimulus) but with different phase shifts (presumably reflecting their different retnotopies). Further, the magnitude of these responses varied considerably with grating direction. (See Figure 7A for an example). In order to account for all these effects we modeled the influence of the stimulus on each neuron's spikes as

$$X(\phi, t)\beta = \mu + A\cos(\phi - \phi_1) + B\cos(2\phi - 2\phi_2) + C(\phi)\cos(\omega t - \Psi(\phi)) \tag{25}$$

We show in Appendix D of Supplementary Text S1 that the right hand side of the above equation can be rewritten as a linearly weighted sum where X is a 13 column covariate matrix, and β is a 13 element parameter vector. Briefly, μ is a fitted parameter that quantifies the mean firing rate. The second and third terms model how this rate is modulated by grating direction ϕ. A, B, ϕ_1 and ϕ_2 are parameters to be fit. The fourth term modulates the response as a sinusoidal function of time since stimulus onset. The phase shift of this last sinusoid depends upon the grating direction, as does the magnitude of the sinusoidal response. This model provided good fits to each neuron's direction dependent PSTHs (see Figure 7C for an example. The mean correlation coefficient between the fitted probabilities $\lambda(t)$ and the neurons' PSTHs was 0.72.

We next used exactly the same covariate matrix $X(\phi, t)$ to model how the *patterns* of spikes across the population depended upon the grating stimulus. We fit 50 different trees at each of 50 different logarithmically spaced values of the Hamming regularization parameter ($\eta \in [10^{-4}, 10^{1}]$). Figure 7D shows an optimally regularized tree, i.e. fit using the regularization corresponding to the largest test data bagged stimulus driven log likelihood. In Figure 7E *left* we show these bagged stimulus driven log likelihoods

(in blue) of both the training and test data sets, normalized to a scale where the corresponding log likelihoods of the above independent neuron model is 1. The stimulus dependent part of the test data log likelihood (that modeled by the regression trees) has a normalized maximum of 1.04, indicating that at least some of the patterns covary with the grating stimulus in a manner which can not be fully explained by an independent neuron model. This difference is highly statistically significant at p=1.4 $\times 10^{-4}$ by t-test on the set (over all time bins) of differences between the stimulus dependent log likelihoods [3]. To ensure that this improvement in fit was robust, we split the test data into two halves and recalculated their stimulus driven log likelihoods (inset). We found that regardless of which half was used to chose the optimum regularization parameter (validation data) the other (test data) demonstrated significant ($p < 0.05$; t-test) improvement in fit over the independent neuron model.

Importantly, the stimulus driven portion of the log likelihood is invariant to exactly which generalization of the null distribution is used. Neither the Good Turing generalization, nor replacing the null distribution with an Ising model alters the stimulus driven log likelihoods (curves of Figure 7E *left*) are visually indistinguishable, not shown). This is an advantage of the pattern model's multiplicative nature, i.e. $P_m(s) = P_m^{null} P_m^{stim}(s)$. Generalizing the null distribution only influences P_m^{stim} through a common stimulus dependent normalization factor (step 3 in section 1.3, equation 15). This was minimal for the cat data, a consequence of the training data's missing probability mass being very small, 0.006 by the Good Turing estimate, and 0.0075 and 0.0026 when estimated by Ising, and independent neuron models respectively. For all stimulus values, the renormalization of equation 15 was never more than a factor of 1 ± 0.01 and averaged 1 ± 0.0015 over the set of all stimuli. It should also be noted that as long as P_m^{null} is itself normalized, the renormalization of equation 15 cancels out when averaged over all stimuli (see the discussion of section 1.3). The crucial point is that either the Good Turing, the Ising, or even another null distribution (perhaps with higher order correlations) can be used for generalization. Our approach is designed to determine how null pattern probabilities are *modulated* by stimuli, not necessarily de-

[3]The t-test is appropriate because the distribution of likelihood differences is asymptotically normal (p=0.001; Jarque-Bera). See Appendix C of Clauset, Shalizi and Newman (2009) and also Vuong (1989) for more extensive discussions.

termine those null probabilities with precision. For this particular data set, the Ising model is a sightly better generalization of the null distribution than the Good Turing estimate (Figure 7E *right*) but the difference is not statistically significant (p=0.82; t-test). However, both null distribution generalizations are significant improvements ($p = 0.03$ Good-Turing and $p = 0.02$ Ising; t-test) over the independent neuron model.

To ensure that we were actually modeling how the pattern structure depended upon the grating stimulus, we randomly shuffled the individual test data trials and also temporally jittered the test data spikes (by ± 15 ms, e.g. ± 5 time bins). This destroyed the coordinated (across neurons) spike timing (pattern) structure, but retained the relation between each individual neuron's spikes with the stimulus. This was confirmed by fitting independent neuron models to each neuron's original and shuffled spikes, visually comparing the firing probabilities fitted to the original and shuffled data, calculating the correlation coefficients between these probabilities (correlation coefficient $r = 1$; $p < 0.001$ for all neurons) and calculating the log likelihoods, which were unchanged for all neurons. However, when considering the bagged trees the log likelihoods of the shuffled spikes (Figure 7 E, red) did change. The stimulus driven portions of the normalized log likelihood dropped below 1. That is, once the pattern structure is destroyed, the fit of the bagged regression trees dropped to that of the independent neuron model. The null log likelihood dropped as well, but remained greater than 1 since the zero pattern (no spikes in any neuron) is not destroyed by shuffling and jittering, and was better described by the pattern model.

Regularization was crucial to getting fits superior to that of independent neuron models, In Figure 8 A we show both unregularized and optimally regularized trees. Figures 8 B and C the patterns of the twelve most probable leaves belonging to each of these trees. The effect of the regularization is to bias the algorithm so that patterns that "look similar" via Hamming distance are grouped together. Common singlets, doublets and triplets are clearly visible in many of the leaves. Such structure is less apparent in the un-regularized tree. Next, we explored which patterns fits were being improved by regularization. After breaking down the stimulus log likelihood as a function of the number of spikes in each pattern (Figure 8 D *left*) we found that the increased fit came mostly from spikes with 3 or more patterns. Proportionally, the log likelihood more than doubled for patterns with 3 spikes (Figure 8D *right*). Thus the effect of regularization

was to stabilize the fits of patterns that occurred rarely (had high numbers of spikes).

The bagged regression trees constitute a probability model for each pattern which depends upon both the grating direction and time since stimulus onset. We calculated tuning curves for all of the observed patterns, and also how their probabilities varied with time since onset (temporal profile). Most tuning curves and temporal profiles changed little when calculated from the regression trees versus the independent neuron models, but a minority did. We show three examples in Figure 9. We investigated whether these changes could be used for improved decoding of grating direction (over independent neuron models) using a maximum *a posteriori* decoding scheme. We found slight improvement (41/60 versus 39/60 test data trials decoded the grating direction accurately), but this was not statistically significant (p=0.34; bootstrap test of proportions). This lack of significance is not surprising given that the improvement in fit (stimulus log likelihood) was only 4% for this data set. It is possible that if the population size was larger, the decoding could have been improved further.

4 Discussion

With the advent of multi-electrode arrays Nicolelis (2008) it has become possible to simultaneously record the action potentials of hundreds, and potentially thousands, of neurons. Such experiments provide an unprecedented opportunity to study how computations are expressed as spatio-temporal patterns of neural activity. Unfortunately, statistical methodologies for analyzing neural population data have not kept pace with the experiments Brown et al. (2004); Macke et al. (2011). One reason is combinatorial, the sheer number of patterns a large neural population can express precludes any sort of direct modeling approach. Another reason is that neuronal populations, or at least the small subsets of the network that can be recorded from, are highly stochastic Schiller et al. (1976); Tolhurst et al. (1983); Vogels et al. (1989); Mainen and Sejnowski (1995); Chelaru and Dragoi (2008); London et al. (2010). In this paper we presented a probabilistic model of encoding by population activity, and an algorithm for fitting it, which circumvents these two difficulties.

To circumvent the combinatorial problem we used a logistic regression tree to group patterns into clusters, the members of which convey equivalent information about the

stimulus being presented. The regression tree was generated via an expectation maximization type, divisive clustering algorithm which iteratively split the expressed patterns into subgroups with similar encoding properties. At each split, a logistic regression model was fit (E step) and the patterns were then reassigned between groups so as to maximize the data likelihood (M step). To circumvent the stochasticity problem, we regularized the regression tree to bias the clustering towards grouping patterns that were "similar" by Hamming distance. This requires fitting trees for multiple values of the regularization parameter, and choosing the optimum value by cross validation. While regularization is computationally intensive, it can substantially improve generalizability to independent test data. Regularization tends to improve model fit for patterns with several (two or more) spikes. These patterns occur rarely, and are harder to assign by likelihood maximization alone.

Regression trees constitute a true pattern encoding model that identifies how the probability of each pattern covaries with an arbitrary stimulus, instead of merely identifying that patterns are present beyond chance. We demonstrated our algorithm on both simulated data and also population data recorded in area 17 of anesthetized cat driven by a grating type stimulus. The simulation studies showed that our algorithm can 1) accurately deduce pattern probabilities even when the data contains thousands of unique patterns and 2) deduce when neurons are being driven collectively, versus individually, even if the collective drive is relatively weak. The analysis of the cat data showed that, certain visual stimuli can be encoded collectively as patterns, at least to some degree. Regarding this result we note that the population we used was small (20 neurons) and it could well be that the influence of patterns would grow if a larger population was used.

Population encoding is most commonly studied under an independent neuron assumption. Nirenberg et al. (2001); Petersen et al. (2001). The population vector model is perhaps one of the simplest examples, merely averaging firing rates Georgopoulos et al. (1986), but more sophisticated approaches are also commonly used Truccolo et al. (2005). Most methods that move beyond independence have attempted to determine the existence of correlations between neurons, the joint peristimulus time histogram being one of the earlier efforts Gerstein and Perkel (1969). More recently, researchers have used second order models to calculate the probability of all patterns under a pairwise coupling assumption. The Ising model couples neurons bi-directionally, so that each

neuron in a pair influences each other in the same manner. It has been used in demonstrate the existence of second order correlations between neurons Schneidman et al. (2006); Tang et al. (2008). Although Ising models do include a constant bias (mean firing rate) term for each neuron, they do not allow for a time varying, perhaps stimulus driven, firing rate. In contrast, cross coupled Generalized Linear Models, in which each neuron's spike probability is modeled as a function of the previous spiking history of all neurons in the population also allow for independent stimulus drive to each neuron Okatan et al. (2005); Pillow et al. (2008); Truccolo et al. (2010); Gerhard et al. (2011). This is also a second order model, but has the advantage that the effects of stimuli upon the spike probability can be included. Further, the couplings are directional, which is perhaps more biologically realistic, than the bi-directional couplings of the Ising model, as there is no reason that neurons in a pair should influence each other in an identical fashion. Recent studies have suggested that as the size of a recorded neuronal population grows, higher (than second) order statistics become critical Roudi et al. (2009); Ganmor et al. (2011), and certain researchers have attempted to capture these higher order effects via various approaches Martignon et al. (2000); Pipa et al. (2008); Ganmor et al. (2011).

The above approaches describe the existence of correlations between neurons. They do not directly address whether these correlations convey any information about stimuli, or other covariates of interest. That is, these approaches can quantify the presence of patterns, but they do not say whether those patterns encode. Towards this point it has been shown by several groups that discerning whether correlations between neurons are driven by each other or by a common input can be problematic Kulkarni and Paninski (2007); Lawhern et al. (2010); Macke et al. (2011). Certain authors have attempted to address whether patterns are important by looking at the information between patterns and stimuli Oram et al. (2001); Nirenberg et al. (2001), others by comparing the performance of decoding models when correlations are, and are not, included Barbieri et al. (2004); Pillow et al. (2011). We took a different approach, directly generating an encoding model of the pattern probability. One of the authors (E.B.) previously addressed encoding by spike patterns of small populations of thalamic barreloid neurons by directly modeling each patterns' probability of expression Ba et al.. However this brute force approach becomes intractable as the population size grows.

It is an experimental reality that one simply does not, and likely never will, have enough data to model each pattern independently. Instead, guided by the data, we clustered the patterns so that only the pattern statistics present in the data were modeled. This clustering does not depend on patterns "looking the same" although we found that reintroducing this prior helped in practice. Instead patterns are clustered together if they are collectively more probable in some subset of the stimulus space. Our divisive clustering approach is similar in spirit to the Causal State Splitting Reconstruction (CSSR) algorithm that generates hidden Markov models of time series by divisively clustering past histories into equivalence classes which are maximally predictive of the future evolution of the time series Shalizi and Klinkner (2004). In this case the patterns are temporal, and the clustering is based entirely upon the internal dynamics of the time series. CSSR performs the clustering completely non-parametrically and hence requires large amounts of data. Still one the authors (R.H.) adapted and applied this technique for calculating the computational complexity of spike trains Haslinger et al. (2010). Our regression tree approach reduces the amount of data required by employing parametric logistic regression models, and also allows for the effects of external stimuli to be included. Another algorithm with similarities to our own is the k-means algorithm for clustering Euclidean data. In k-means the data points, defined in some Euclidean space, are randomly assigned to k-clusters, the cluster's centroids are calculated, and the data points reassigned to the cluster with the closest centroid. Calculating the centroids is analogous to our fitting a logistic regression model at each tree branching while reassigning the data points between clusters is similar to our reassigning patterns between child nodes. In both k-means and our algorithm, the missing data of the EM procedure is the knowledge of which cluster (leaf) a data point (pattern) is a member of.

Regression trees have long been used to model complex functions since their introduction by Brieman and colleagues in 1984 with the CART (classification and regression tree) algorithm Breiman et al. (1984), and the later C4.5 algorithm of Quinlan Quinlan (1993). Trees are unbiased estimators but notoriously variable due to the recursive partitioning, if two classes of observations are separated by a split near the root of the tree, they can never be remerged Ripley (1996). A number of strategies have been developed to not only handle, but also take advantage of this, most notably bagging. Bagging averages the response (here the pattern probability) from many trees to

achieve a more reliable estimator Breiman (1996). We used this technique to improve significantly goodness of fit and generalizability. In contrast, boosting combines many weak classifiers to obtain an optimal one, and can be applied at each split in the tree Schapire (1990); Freund (1995) . Specific algorithms such as ADABOOST Freund and Schapire (1997) and notably the LOGITBOOST algorithm of Friedman Hastie and Tibshiani Friedman et al. (2000) that combines many weak logistic regression models at each split, would likely prove very applicable although more work is required to determine the best approach for modeling neural patterns.

Other algorithmic improvements suggest themselves. First, we currently fit logistic regression models at each branching using *all* of the stimulus covariates. At times more optimal models can be achieved by fitting only a subset of the covariates at each branching Hastie et al. (2009). Second, we calculated the BIC at each branching and only kept the branching if it minimized the BIC. However an initial branching that leads to a poor improvement in fit may lead to a subsequent branching with substantial fit improvement. For this reason, many studies, including the original work of Breiman Breiman et al. (1984) grow their trees "large", continuing the splitting until each pattern is in its own leaf. They then "prune" the branches of the tree, from bottom to top, until an optimally sized tree is found. We have yet to fully explore either of these strategies in the context of neuronal patterns, although we note that both will likely increase computation time. For both the simulated and cat data, a tree could be fit using a standard work station in only a few minutes.

Perhaps the most significant drawback of the current formalism is that the pattern model is not *nested* McCullagh (1989). That is, it would be most ideal to use the independent neuron case as the null model, and build the regression tree based pattern model from that starting point. This would have two advantages, first it would facilitate the comparison of pattern based encodings with independent neuron based encodings. Second, since correlations between neurons tend to be weak and sparse, the encoding properties of most observed patterns are likely best described using an independent neuron assumption. A nested model would allow one to identify exactly which patterns are, in contrast, best described collectively. We are currently investigating how a nested model could be constructed, but such a model is beyond the scope of the present work.

It is commonly held that neuronal networks store and process information in spatio-

temporal patterns of spiking activity but exactly how they do this is hotly debated Averbeck et al. (2006); Jacobs et al. (2009); Abbott and Dayan (1999). If the functional role of neuronal correlations and patterns of population spiking activity are to be fully understood, it is crucial to develop methodologies capable of characterizing how the collective spiking statistics of neuronal populations depend upon stimuli and other variables of interest. Our approach has the advantage of being extremely general, allowing any variable to be included in the covariate matrix. For example, although we restricted ourselves in this paper to analyzing how patterns depend upon external stimuli, information about the internal dynamics of the system, such as spike history dependence, or macroscopic state, such as local field potentials could also be included. Regarding this generality, it is interesting to note that recent theoretical work on reservoir computing models has shown that recurrent networks can store vast amounts of information in overlapping patterns of activity, and that different linear readouts can reliably access different encodings Buonomano and Maass (2009). In principle, regression trees could allow one to experimentally investigate if different encodings are simultaneously present in a true population code of arbitrary correlative order.

The authors would like to thank Demba Ba, Zhe Chen and Cosma Shalizi for helpful conversations regarding the research presented in this paper. This work was supported by NIH grants K25 NS052422-02 (R.H.), DP1 OD003646-0, MH59733-07 (E.N.) and the Hertie Foundation (G.P.). Experiments with cat recordings (D.N.) were supported by a Deutsche Forschungsgemeinschaft Grant NI 708/2-1, Hertie Stiftung, Max-Planck Society, and the Frankfurt Institute for Advanced Studies.

5 Appendix A: Pattern Reassignment

Assume the patterns have each been initially assigned to one of two child nodes, denoted here by $-$ and $+$ and a logistic regression model has been fit such that the probability of each node is given by

$$P_{\pm}(t) = \frac{e^{\pm[\beta_0 + X(t)\beta]}}{1 + e^{\pm[\beta_0 + X(t)\beta]}} \tag{26}$$

The log likelihood of the observed patterns is then

$$\begin{aligned} \log \mathcal{L} &= \sum_{t=1}^{T} \log P_{m_t}(t) \\ &= \sum_{m=1}^{M} \sum_{t \in T_m} \log P_m(t) \end{aligned} \tag{27}$$

T is the total number of observations and T_m is the subset of observations for which pattern m is observed. $P_m(t)$ is the probability of pattern m at time t and $P_{m_t}(t)$ is the probability at time t of the pattern m which was actually observed at time t. The goal is to maximize the likelihood by reassigning the patterns m between the two classes, while keeping the parameters β_0 and β fixed, e.g. to perform the M step. To this end, it suffices to maximize each term in the sum over m independently.

Thus consider this component of the log likelihood

$$\begin{aligned} \log \mathcal{L}_m^{\pm} &= \sum_{t \in T_m} \log P_m(t) \\ &= \sum_{t \in T_m} \log \left[\frac{e^{\pm[\beta_0 + X(t)\beta]}}{1 + e^{\pm[\beta_0 + X(t)\beta]}} \frac{T_m}{T_\pm} \right] \end{aligned} \tag{28}$$

If we assume, without loss of generality, that the covariate matrix X has been standardized so that each column has mean zero, then the fraction of observations belonging to each class $\pm$ is given by the bias parameter β_0

$$\frac{T_\pm}{T} = \frac{e^{\pm\beta_0}}{1 + e^{\pm\beta_0}}; \tag{29}$$

Substituting this into equation 28 and canceling terms, we obtain

$$\begin{aligned} \log \mathcal{L}_m^{\pm} &= \sum_{t \in T_m} \log \left[\frac{e^{\pm[\beta_0 + X(t)\beta]}}{1 + e^{\pm[\beta_0 + X(t)\beta]}} \frac{1 + e^{\pm\beta_0}}{e^{\pm\beta_0}} \frac{T_\pm}{T} \frac{T_m}{T_\pm} \right] \\ &= \sum_{t \in T_m} \pm X(t)\beta + \log \left[\frac{1 + e^{\pm\beta_0}}{1 + e^{\pm[\beta_0 + X(t)\beta]}} \right] + \log \left[\frac{T_m}{T} \right] \end{aligned} \tag{30}$$

The node into which pattern m should be placed is the one which maximizes equation 30 . The third term is the same for both nodes. The first depends upon the sign of $\sum_{t \in T_m} X(t)\beta$. We now show that the second term does as well. Multiplying the top

and bottom of the second term by $e^{\mp[\beta_0+X(t)\beta]/2}$ we obtain

$$\begin{aligned}
\log & \left[\frac{1+e^{\pm\beta_0}}{1+e^{\pm[\beta_0+X(t)\beta]}}\frac{e^{\mp[\beta_0+X(t)\beta]/2}}{e^{\mp[\beta_0+X(t)\beta]/2}}\right] = \\
\log & \left[\frac{e^{\mp\beta_0/2}+e^{\pm\beta_0/2}}{e^{\mp[\beta_0+X(t)\beta]/2}+e^{\mp[\beta_0+X(t)\beta]/2}}e^{\mp X(t)\beta/2}\right] = \\
\log & \left[\frac{e^{\mp\beta_0/2}+e^{\pm\beta_0/2}}{e^{\mp[\beta_0+X(t)\beta]/2}+e^{\mp[\beta_0+X(t)\beta]/2}}\right] \mp \frac{X(t)\beta}{2}
\end{aligned} \tag{31}$$

Inserting equation 31 into equation 30 and combining terms we obtain

$$\begin{aligned}
\log \mathcal{L}_m^{\pm} &= \sum_{t\in T_m} \pm\frac{X(t)\beta}{2} \\
&+ \sum_{t\in T_m}\left[\log\left[\frac{e^{\mp\beta_0/2}+e^{\pm\beta_0/2}}{e^{\mp[\beta_0+X(t)\beta]/2}+e^{\mp[\beta_0+X(t)\beta]/2}}\right]+\log\left[\frac{T_m}{T_\pm}\right]\right]
\end{aligned} \tag{32}$$

Only the first term is different between the two nodes.

Thus so as to maximize the likelihood, pattern m should be placed in the first $(-)$ child node if $\sum_{t\in T_m} X(t)\beta < 0$ and in the second $(+)$ child node if $\sum_{t\in T_m} X(t)\beta > 0$. This rule can be recursively applied for splitting each node of the tree.

6 Appendix B: Algorithmic Benchmarking

To test the regression tree algorithm over a range of data lengths, true numbers of clusters, and numbers of patterns comprising each cluster, we generated multinomial processes from models of the form

$$P_m(t) = P_c(t)P_{m|c} \tag{33}$$

where $P_c(t)$ is a multinomial logit model

$$P_c(t) = \frac{e^{\beta_0^c+X(t)\beta^c}}{\sum_{c=1}^{C} e^{\beta_0^c+X(t)\beta^c}} \tag{34}$$

and

$$P_{m|c} = \frac{N_m}{\sum_{m\in c} N_m} \tag{35}$$

The parameters $\{\beta_0^c, \beta^c\}$ for each cluster were randomly chosen within $[-1, 1]$. Likewise the covariate matrix $\mathbf{X(t)}$ was random with values drawn from within $[-1, 1]$. We

chose $\mathbf{X}(\mathbf{t})$ to have 24 columns (24 plus 1 bias parameter gives 25 parameters total for each class) and and a variable number T of observations (rows). Each class had a variable number of "patterns" M labeled by ordinal numbers. The probabilities of drawing any given pattern $m \in 1 \ldots M$ given the cluster c were equal. That is $P_{m|c} = 1/M$. In the absence of Hamming regularization, the regression tree algorithm can operate on any multinomial process, not only one where the observations are defined as patterns of spikes.

We then simulated data for different numbers of clusters $C \in [5, 10, 20, 50]$ and different numbers of patterns per class $M \in [5, 10, 20, 50]$, and also different numbers of observations $T \in [10, 20, 100, 200] \times 10^3$. These numbers of observations were chosen to reflect experiments at 1 ms resolution which are 10, 20, 100 and 200 seconds long respectively. For each triplet $\{C, M, T\}$ 20 simulated data sets were generated, and the logistic tree algorithm fit to the data.

We calculated three measures of "goodness of fit". The first was the number of leaf nodes found by the algorithm normalized by the true number of clusters.

$$F = \frac{N_{leaf}}{C} \tag{36}$$

The second measure was one of pattern mis-assignment. The individual patterns comprising each leaf node can, due to insufficient data, belong to different "true" clusters. We assigned each leaf node to a true equivalence class C by choosing the class to which the majority of patterns in the leaf node belong to, and calculated the "mixing fraction as"

$$G = \frac{N_{m \notin C}}{N_{m \in \text{leaf}}} \tag{37}$$

Third we calculated the percentage of log likelihood which the regression tree accounted for compared to the true model. We used the null model, for which the probability of any given pattern m is simply its occupation probability, to define a baseline log likelihood.

$$\log \mathcal{L}^{null} = \sum_m N_m \log \frac{N_m}{T} \tag{38}$$

and calculated

$$\Delta L = \frac{\log \mathcal{L}^{tree} - \log \mathcal{L}^{null}}{\log \mathcal{L}^{true} - \log \mathcal{L}^{null}} \tag{39}$$

where $\log \mathcal{L}^{tree}$ is computed with the time varying probability of each pattern m as implicitly defined by the logistic regression tree, and $\log \mathcal{L}^{true}$ is the log likelihood of the true model used to generate the simulated data. A value of ΔL close to 1 implies that the tree model is almost as good as the true model, while a value of ΔL close to 0 implies a poor fit.

The results are shown in Supplementary Figure S1. A few trends are apparent. First, the number of leaves (first column) tends to be slightly higher than the true number of clusters. This trend increases with the total number of patterns and does not diminish as the simulation time grows. In contrast, the mixing fraction (second column) does decrease as the data size grows, indicating that the leafs, although more numerous than they should be, are comprised of patterns which should be grouped together. These trends arise because if patterns are mis-assigned by an initial branching they can not be remerged. Thus the algorithm has to make additional leaves to achieve a well fit model. We often find two or more leaves containing patterns which belong to the same cluster, but only that cluster. Although not as parsimonious as the multinomial logit model used for the simulation, the regression tree does describe the data reasonably well. The log likelihood fraction (third column) is high for all simulations, although it does decrease moderately as the number of clusters grows.

7 Appendix C: Experimental Methods

Anesthesia was induced with ketamine and maintained with a mixture of 70% N2O and 30% O2, supplemented with halothane (0.40.6%). The animal was paralyzed with pancuronium bromide (Pancuronium, Organon, 0.15 mg kg^{-1} h^{-1}). All the experiments were conducted according to the guide-lines of the American Physiological Society and German law for the protection of animals, approved by the local governments ethical committee and overseen by a veterinarian.

Multi-unit activity was recorded from a region of area 17 corresponding to the central part of the visual field. Single-unit activity was extracted by offline sorting. For recording we used two SI-based multi-electrode probes (16-channels per electrode) supplied by the Centre for Neural Communication Technology at the University of Michigan (Michigan probes) with an inter-contact distance of 200 μm (0.3 - 0.5 MΩ

impedance at 1,000 Hz). The probes were inserted in the cortex approximately perpendicular to the surface to record from neurons at different cortical depths and along an axis tangential to the cortical surface. The software used for visual stimulation was Active-STIM (www.ActiveSTIM.com). The stimulus consisted of a drifting sinusoidal grating, spanning 15 degrees of visual angle (spatial frequency, 3 degrees/cycle; temporal frequency of the drift, 3.6 degrees/s), which was sufficient to cover the receptive fields of all the recorded cells simultaneously and to stimulate also the regions surrounding the receptive fields.

8 Appendix D: Parameterizing the Grating Stimulus

As discussed in the main text, neurons in area 17 were stimulated by sinusoidal grating of a fixed temporal frequency and in different directions over repeated trials. These neurons mean firing rates were strongly modulated by grating direction and also periodically at the frequency of the grating. This periodic modulation had both amplitudes and phase shifts that were also grating direction dependent. Thus we had to account for all of these effects in parameterizing the stimulus as a covariate matrix X linearly weighted by a parameter vector β. Defining the argument of the logistic regression models as $\Lambda = X\beta$, we used the form

$$\Lambda(\phi, t) = \mu + A\cos(\phi - \phi_1) + B\cos(2\phi - 2\phi_2) + C(\phi)\cos(\omega t - \Psi(\phi)) \qquad (40)$$

μ is a fitted parameter that quantifies the mean firing rate. The second and third terms model how this rate is modulated by grating direction ϕ and A, B, ϕ_1 and ϕ_2 are parameters to be fit. These terms can be rewritten using a trigonometric identity so that they are linear in fitted parameters, e.g.

$$\begin{aligned} A\cos(\phi - \phi_1) &= A(\cos(\phi_1)\cos(\phi) + A\sin(\phi_1)\sin(\phi) \\ &= \alpha_1\cos(\phi) + \alpha_2\cos(\phi) \end{aligned} \qquad (41)$$

where $\alpha_1 = A\cos(\phi_1)$ and $\alpha_2 = A\sin(\phi_2)$. The third term can similarly be linearized. This model form is commonly used to model the effect of angular variables such as grating direction.

The fourth term modulates the response as a sinusoidal function of time since stimulus onset. The phase shift of this sinusoid $\Psi(\phi)$depends upon the grating direction, as

does the magnitude $C(\phi)$ of the sinusoidal response. We first split this term analogously to the above.

$$C(\phi)\cos(\omega t - \Psi(\phi)) \quad = \quad [C(\phi)\cos(\Psi(\phi))]\cos(\omega t) + [C(\phi)\sin(\Psi(\phi)]\sin(\omega t) \tag{42}$$

We then parameterized each bracketed term similarly to the directional dependence of the firing rate.

$$C(\phi)\cos(\Psi(\phi)) = \gamma_1\cos(\phi) + \gamma_2\sin(\phi) + \gamma_3\cos(2\phi) + \gamma_4\sin(2\phi) \tag{43}$$

and $C(\phi)\cos(\Psi(\phi))$ is similar. Λ is now written as a linear sum of weighted functions dependent upon the grating direction and time since stimulus onset, and can be recast as a 13 column covariate matrix X multiplied by a parameter vector β.

References

L. F. Abbott and P. Dayan. The effect of correlated variability on the accuracy of a population code. *Neural computation*, 11(1):91–101, 1999.

B. B. Averbeck, P. E. Latham, and A. Pouget. Neural correlations, population coding and computation. *Nat Rev Neurosci*, 7(5):358–66, 2006.

D. Ba, S. Temereanca, and E.N. Brown. Algorithms for the analysis of ensemble neural spiking activity using simultaneous-event multivariate point-process models. *unpublished*.

R. Barbieri, L.M. Frank, D.P. Nguyen, M.C. Quirk, V. Solo, M.A. Wilson, and E.N. Brown. Dynamic analyses of information encoding in neural ensembles. *Neural Computation*, 16:277–307, 2004.

L. Breiman. Bagging predictors. *Machine Learning*, 26:123–140, 1996.

L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth, Boca Raton, FL, 1984.

E. N. Brown, D. P. Nguyen, L. M. Frank, M. A. Wilson, and V. Solo. An analysis of neural receptive field plasticity by point process adaptive filtering. *Proceedings of*

the National Academy of Sciences of the United States of America, 98(21):12261–6, 2001.

E. N. Brown, R. E. Kass, and P. P. Mitra. Multiple neural spike train data analysis: state-of-the-art and future challenges. *Nature neuroscience*, 7(5):456–61, 2004.

D.V. Buonomano and W. Maass. State-dependent computations: spatiotemporal processing in cortical networks. *Nature Reviews Neuroscience*, 10:113–125, 2009.

A. Clauset, C. R. Shalizi, and M. J. Newman. Powerlaw distributions in empirical data. *SIAM Review*, 51661–703, 2009.

M. I. Chelaru and V. Dragoi. Efficient coding in heterogeneous neuronal populations. *Proceedings of the National Academy of Sciences of the United States of America*, 105(42):16344–9, 2008. Chelaru, Mircea I Dragoi, Valentin Proc Natl Acad Sci U S A. 2008 Oct 21;105(42):16344-9. Epub 2008 Oct 14.

Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121:256–285, 1995.

Y. Freund and R. Schapire. A decision-theoretic generalization of online learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139, 1997.

J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting (with discussion). *Annals of Statistics*, 28:337–407, 2000.

E. Ganmor, R. Segev, and E. Schneidman. Sparse low-order interaction network underlies a highly correlated and learnable neural population code. *Proceedings of the National Academy of Sciences of the United States of America*, 2011. Proc Natl Acad Sci U S A. 2011 May 20.

A.P. Georgopoulos, A.B Schartz, and R.E. Kettner. Neuronal population encoding of movement direction. *Science*, 233:1416–1419, 1986.

F. Gerhard, G. Pipa, B. Lima, S. Neuenschwander, and W. Gerstner. Extraction of network topology from multi-electrode recordings is there a small-world effect? *Frontiers in Computational Neuroscience*, 5(4):1–13, 2011.

G.L. Gerstein and D.H. Perkel. Simultaneously recorded trains of action potentials: analysis and functional interpretation. *Science*, 164:828–830, 1969.

R. Haslinger, K.L. Klinkner, and C.R. Shalizi. The computational structure of spike trains. *Neural Computation*, 22:121–157, 2010.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference and Prediction.* Springer-Verlag, New York, 2nd. edition edition, 2009.

A. L. Jacobs, G. Friedman, R. M. Douglas, N. M. Alam, P. E. Latham, G. T. Prusky, and S. Nirenberg. Ruling out and ruling in neural codes. *Proceedings of the National Academy of Sciences of the United States of America*, 106(14):5936–41, 2009.

J. E. Kulkarni and L. Paninski. Common-input models for multiple neural spike-train data. *Network*, 18(4):375–407, 2007.

V. Lawhern, W. Wu, N. Hatsopoulos, and L. Paninski. Population decoding of motor cortical activity using a generalized linear model with hidden states. *Journal of neuroscience methods*, 189(2):267–80, 2010.

M. London, A. Roth, L. Beeren, M. Hausser, and P. E. Latham. Sensitivity to perturbations in vivo implies high noise and suggests rate coding in cortex. *Nature*, 466 (7302):123–7, 2010.

J. Macke, M. Opper, and M. Bethge. Common input explains higher-order correlations and entropy of neural population activity. *Physical Review Letters*, 106208102-5, 2011.

Z. F. Mainen and T. J. Sejnowski. Reliability of spike timing in neocortical neurons. *Science*, 268(5216):1503–6, 1995.

L. Martignon, G. Deco, K. Laskey, M. Diamond, W. Freiwald, and E. Vaadia. Neural coding: higher-order temporal patterns in the neurostatistics of cell assemblies. *Neural computation*, 12(11):2621–53, 2000.

J McCullagh, P; Nelder. *Generalized linear models*. Chapman and Hall, New York, 1989.

MAL Nicolelis. *Methods for neural ensemble recordings*. Frontiers in Neuroscience, Boca Raton, 2 edition, 2008.

S. Nirenberg, S. M. Carcieri, A. L. Jacobs, and P. E. Latham. Retinal ganglion cells act largely as independent encoders. *Nature*, 411(6838):698–701, 2001.

M. Okatan, M. A. Wilson, and E. N. Brown. Analyzing functional connectivity using a network likelihood model of ensemble neural spiking activity. *Neural Computation*, 17(9):1927–1961, 2005. Times Cited: 63.

M. W. Oram, N. G. Hatsopoulos, B. J. Richmond, and J. P. Donoghue. Excess synchrony in motor cortical neurons provides redundant direction information with that from coarse temporal measures. *Journal of neurophysiology*, 86(4):1700–16, 2001.

A. Orlitsky, N. P. Santhanam, and J. Zhang. Always Good Turing: Asymptotically Optimal Probability Estimation. *Science*,302(5644):427–31, 2003.

R. S. Petersen, S. Panzeri, and M. E. Diamond. Population coding of stimulus location in rat somatosensory cortex. *Neuron*, 32(3):503–14, 2001.

J. W. Pillow, J. Shlens, L. Paninski, A. Sher, A. M. Litke, E. J. Chichilnisky, and E. P. Simoncelli. Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature*, 454(7207):995–9, 2008.

J. W. Pillow, Y. Ahmadian, and L. Paninski. Model-based decoding, information estimation, and change-point detection techniques for multineuron spike trains. *Neural computation*, 23(1):1–45, 2011.

G. Pipa, D. W. Wheeler, W. Singer, and D. Nikolic. Neuroxidence: reliable and efficient analysis of an excess or deficiency of joint-spike events. *Journal of computational neuroscience*, 25(1):64–88, 2008.

Y. Paitwan *In All Likelihood* Oxford University Press, New York, NY, 2001.

J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufman, San Mateo, CA, 1993.

B. D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge, England, 1996.

Y. Roudi, S. Nirenberg, and P. E. Latham. Pairwise maximum entropy models for studying large biological systems: When they can work and when they can't. *Plos Computational Biology*, 5(5), 2009.

R. Schapire. The strength of weak learnability. *Machine Learning*, 5:197–227, 1990.

P.H. Schiller, B.L. Finlay, and S.F. Volman. Short-term response variability of monkey striate neurons. *Brain Research*, 105:347–349, 1976.

E. Schneidman, 2nd Berry, M. J., R. Segev, and W. Bialek. Weak pairwise correlations imply strongly correlated network states in a neural population. *Nature*, 440(7087): 1007–12, 2006.

M. N. Shadlen and W. T. Newsome. Noise, neural codes and cortical organization. *Current opinion in neurobiology*, 4(4):569–79, 1994.

C. R. Shalizi and K. L. Klinkner. Blind construction of optimal nonlinear recursive predictors for discrete sequences. In M. Chickering and J. Y. Halpern, editors, *Uncertainty in artificial intelligence: Proceedings of the twentieth conference (UAI 2004)*, pages 504–511, 2004.

J. Sohl-Dickstein, P. B. Battaglino and M. R. DeWeese. New method for parameter estimation in probablistic models: minimum probability flow. *Physical Review Letters*, 107(22):220601–220604, 2011.

A. Tang, D. Jackson, J. Hobbs, W. Chen, J. L. Smith, H. Patel, A. Prieto, D. Petrusca, M. I. Grivich, A. Sher, P. Hottowy, W. Dabrowski, A. M. Litke, and J. M. Beggs. A maximum entropy model applied to spatial and temporal correlations from cortical networks in vitro. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 28(2):505–18, 2008.

D.J. Tolhurst, J.A. Movshon, and A.F. Dean. The statistical reliability of signals in single neurons in cat and monkey visual cortex. *Vision Research*, 23:775–785, 1983.

W. Truccolo, U. T. Eden, M. R. Fellows, J. P. Donoghue, and E. N. Brown. A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *Journal of Neurophysiology*, 93(2):1074–1089, 2005. Times Cited: 170.

W. Truccolo, L. R. Hochberg, and J. P. Donoghue. Collective dynamics in human and monkey sensorimotor cortex: predicting single neuron spikes. *Nature Neuroscience*, 13(1):105–U275, 2010.

Q. H. Vuong. Likelihood ratio tests for model selection and non-nested hypotheses. *Econometrica*, 57307-333, 1989.

R. Vogels, W. Spileers, and G.A. Orban. The response variability of striate cortical neurons in the behaving monkey. *Experimentelle Hirnforschung Experimentation cerebrale*, 77:432–436, 1989.

S. Yu, D. Huang, W. Singer, and D Nikolić. A small world of neuronal synchrony. *Cerebral Cortex*, 18:2891–2901, 2008.

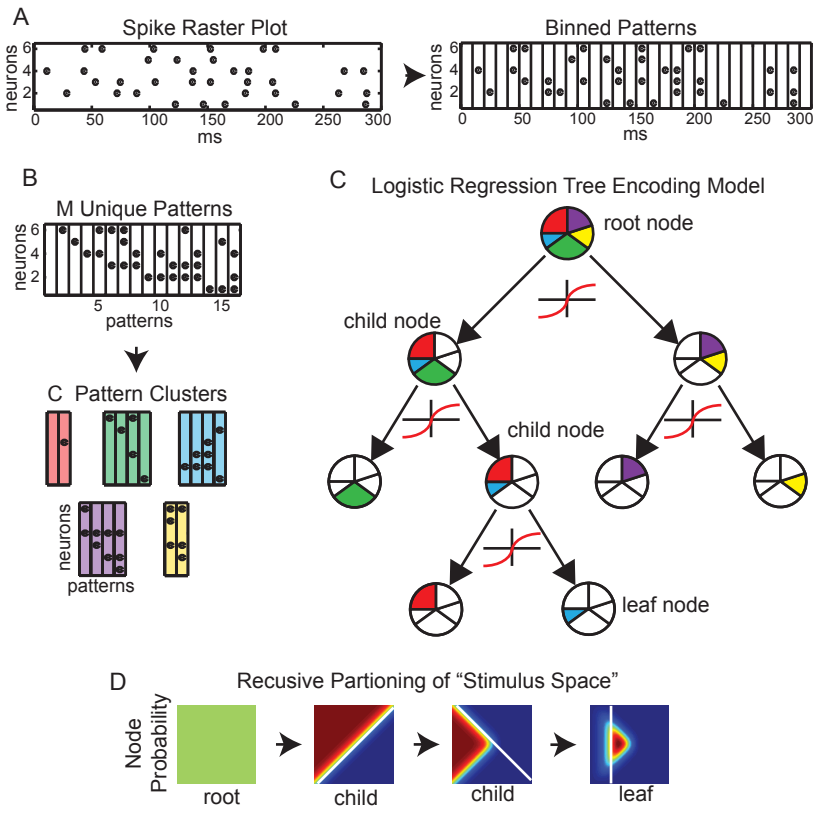

Figure 1: **Schematic of the tree based pattern encoding model.** A) Spikes are binned into discrete time (here 10 ms) patterns across neurons. B) The encoding model partitions the M unique observed patterns into C clusters. The member patterns of these clusters have similar stimulus encoding properties. C) Partitioning is accomplished using a regression tree which constitutes a probability model for each pattern. The root node contains all patterns (colors denote patterns with similar encoding properties) and is split into two child nodes. The probability of each child node is described by a stimulus dependent logistic regression model. The child nodes are themselves split, until the BIC is minimized. The leafs on the tree are the final clusters, comprised of patterns with the same probabilistic dependence upon the stimuli. D) Each split in the tree defines a soft partitioning (white line) of the stimulus space into subspaces. The leaf has high probability in the final subspace (denoted by red).

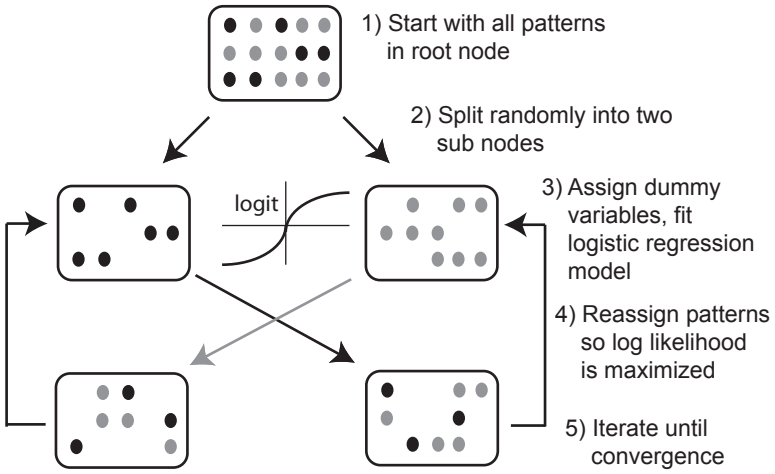

Figure 2: **Expectation maximization type node splitting algorithm.** Patterns are randomly assigned to two child nodes. A logistic regression model is fit (E step), and then the patterns are reassigned to maximize the likelihood (M step).

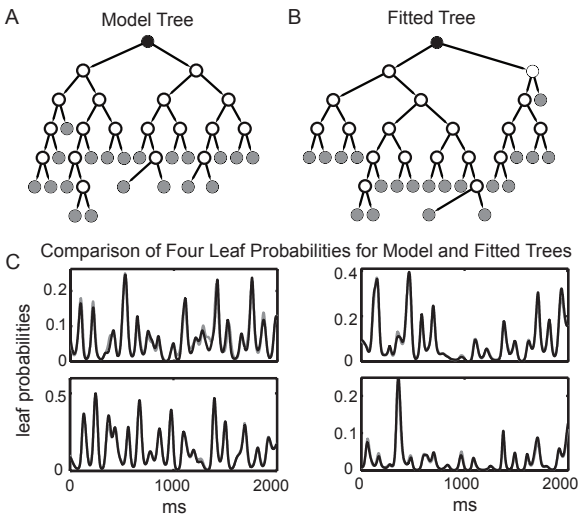

Figure 3: **Algorithm recovers correct pattern groupings and probabilities.** A) Logistic regression tree used to generate ordinal patterns with time varying probabilities. The 20 leaves are in grey, each is comprised of 20 patterns. B) Fitted logistic regression tree. Although the branchings of the tree are different than the model, each leaf is isotropic to (comprised of the same patterns as) a leaf of the model tree. C) Time varying probabilities of four leaves. Grey is the model. Black is the fitted regression tree. The plots are nearly identical.

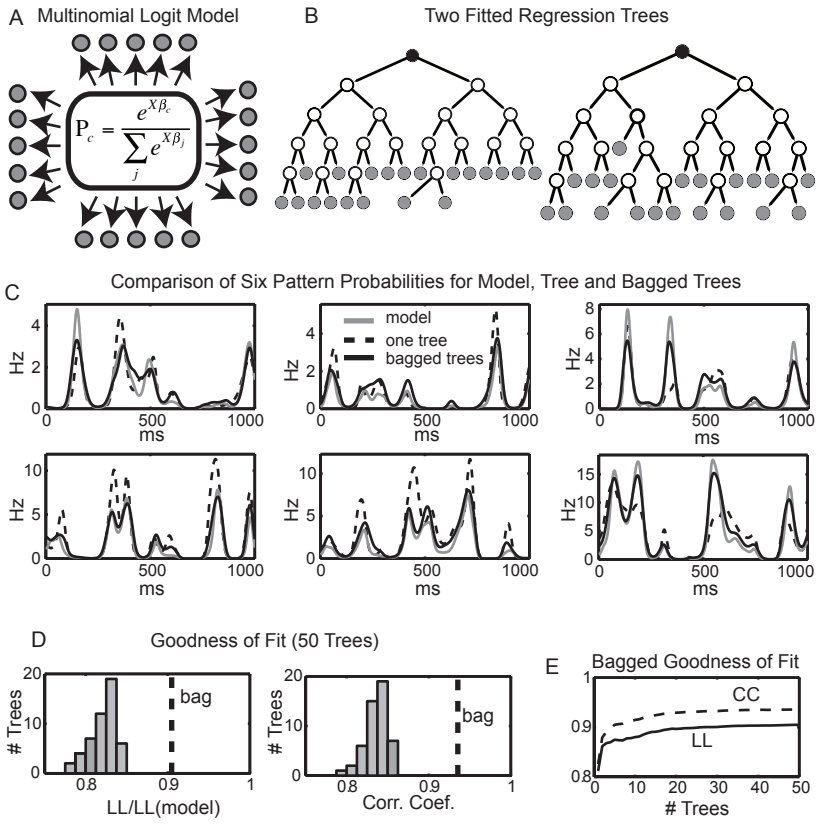

Figure 4: **Bagging can substantially improve goodness of fit.** A) A multinomial logit model with 20 pattern groupings was used to generate data. B) Two different regression trees, each with 20 leaves isotropic to the original model. C) Time varying probabilities of six patterns (from different groups) according to the original model (grey), the first tree shown in B) (dashed black) and 50 bagged trees (black). D)Left panel: Histogram of the log likelihood ratio (compared to original model) accounted for by all 50 fitted trees. Right panel: the mean correlation coefficient between the true and fitted pattern probabilities. Bagging substantially improved these goodness of fit measures (vertical dashed lines). E) Log likelihood ratio and mean correlation coefficients as a function of the number of trees used for bagging.

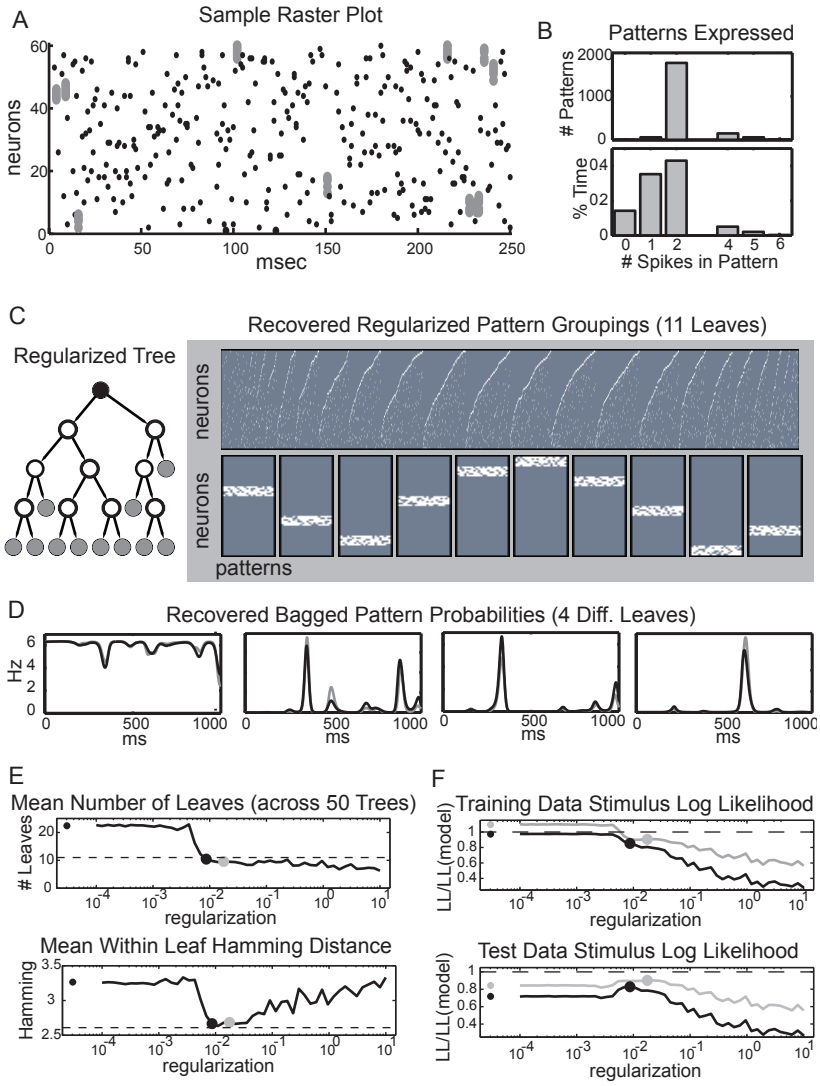

Figure 5: **Regularization improves generalizability.** A) Sample raster plot of 60 simulated neurons firing independently, except when groups of 4 or more cells fire collectively as a "cell assembly" (large grey dots). B) Number of unique patterns, as a function of number of spikes contained, and percentage of time expressed. C) Regularized regression tree recovering correct pattern groupings. Panels show patterns contained in each leaf. White dots denote spiking neurons. Top panel = independent neuron group bottom 10 panels = cell assembly groups. D) Bagged pattern probabilities (black) for 4 patterns (from different leaves). Grey = original model. E) Mean number of leaves, and mean within leaf Hamming distance versus regularization parameter. Dashed lines are original model. Small dots are un-regularized result. F) Stimulus driven log likelihoods for training and test data. Black: average over 50 trees. Grey: bagged trees. Black and grey dots are maxima of test data log likelihood. Dashed line is original model.

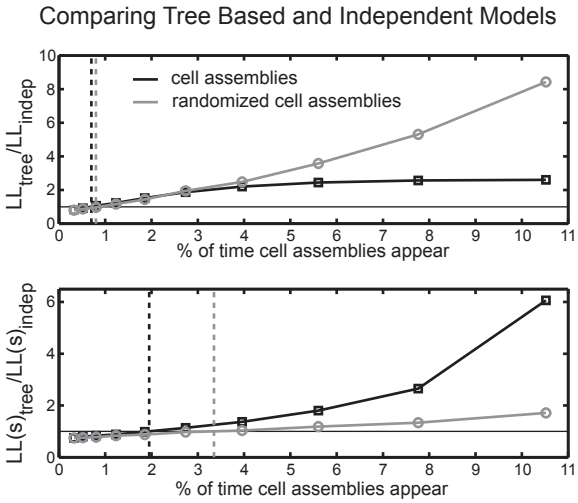

Figure 6: **Algorithm detects the presence of weak patterns.** Upper panel: ratio of full test data log likelihoods for bagged tree and independent neuron models. Lower panel: ratio of stimulus driven log likelihoods. Black squares: cell assembly model of Figure 5. Grey circles, same model but with patterns randomized across cell assemblies (see text). Horizontal lines denote equality for the bagged tree and independent neuron model log likelihoods. Vertical dashed lines denote percentages at which patterns can no longer be detected as present (upper panel) and no longer detected as collectively driven by the stimulus (lower panel).

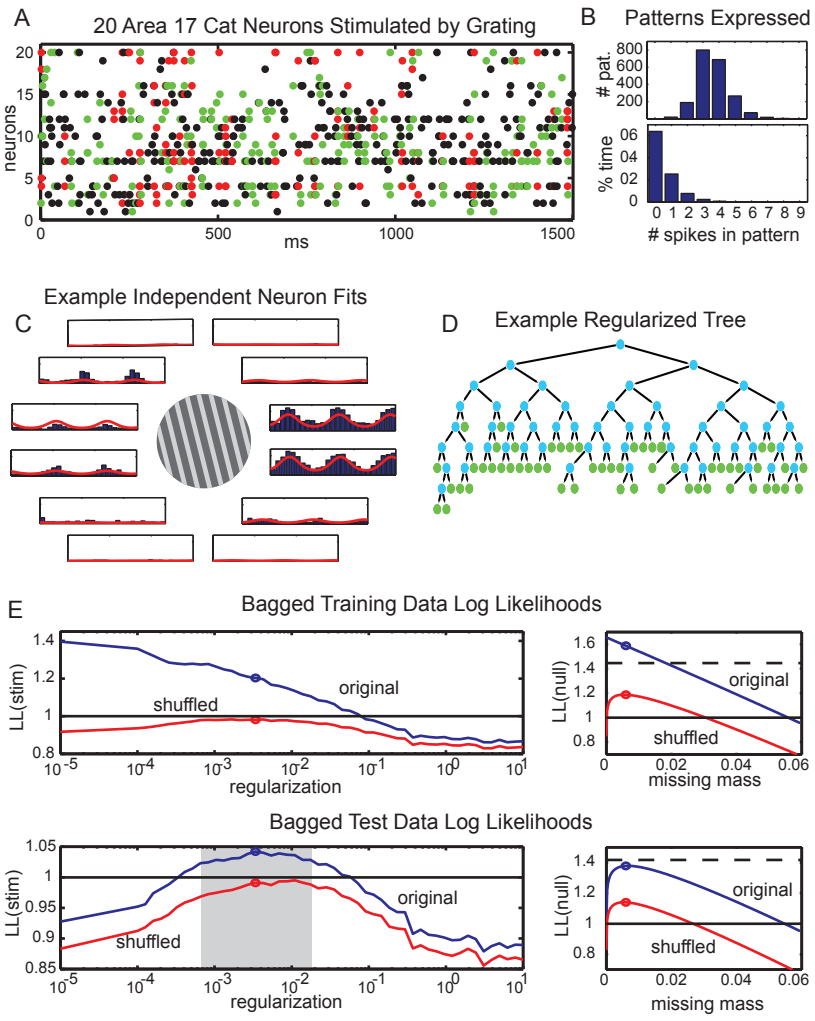

Figure 7: **Application of regression trees to a 20 V1 cat neuron population.** A) Example raster plot. Black dots correspond to patterns with one spike, green: 2 spikes, red: 3 or more spikes. B) Number of patterns and percentage of time expressed. C) Neurons were stimulated by a sinusoidal grating moving in 12 different directions. Histogram is PSTH for each direction, red line is independent neuron model fit. D) An optimally regularized regression tree. E) *Left:* Stimulus driven log likelihoods (blue) versus regularization for training and test data. Horizontal black line is independent neuron model. Circles denote optimal regularization. Grey region denotes significant improvement over independent model. Red lines are log likelihoods of shuffled and jittered data. *Right:* Null log likelihood (blue) as a function of missing mass used for generalization. Circle is Good-Turing estimate. Red is shuffled and jittered data. Horizontal black line is independent neuron model, horizontal dashed line is Ising model.

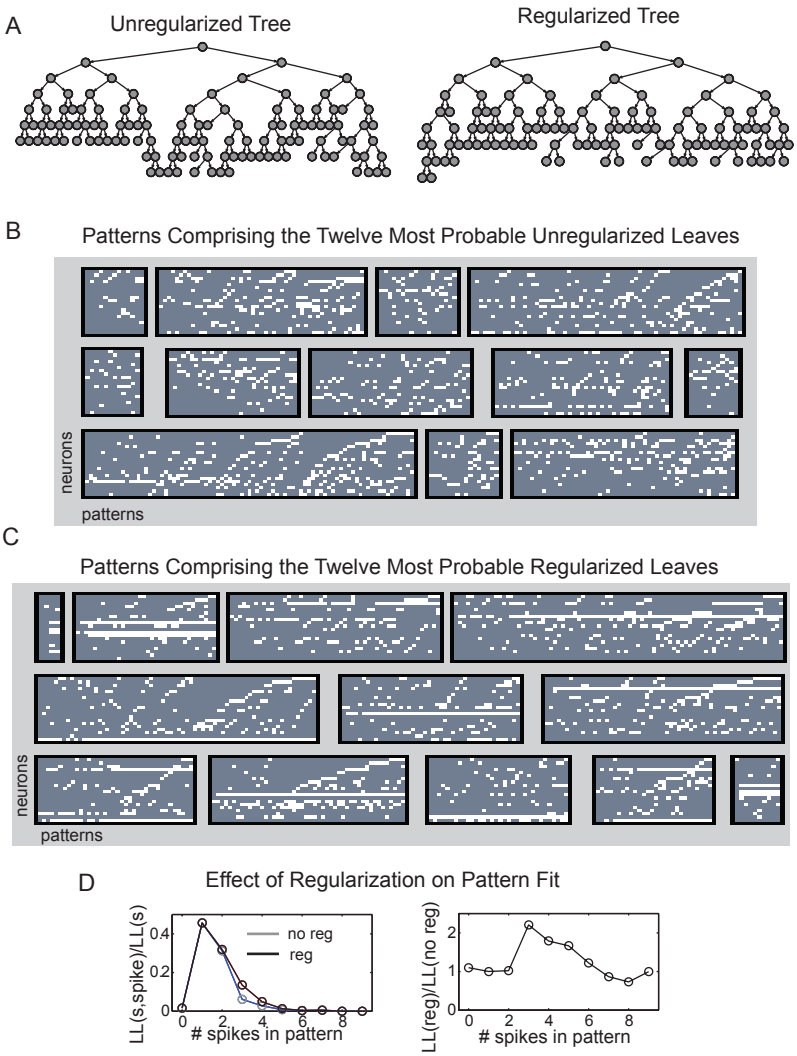

Figure 8: **Regularization improves the fit of cat V1 patterns with high numbers of spikes.** A) Unregularized and regularized regression trees. B) Member patterns for the 12 most probable leaves of the unregularized tree, C) and for a regularized tree. Singlets doublets and triplets are apparent in the leaves of the regularized tree. D) *Left:* Percentage of the log likelihood accounted for by patterns with different numbers of spikes. Grey = no regularization, black=regularized. *Right:* ratio of the plots in the upper panel. Regularization improves the fit of 3 spike patterns by more than a factor of two.

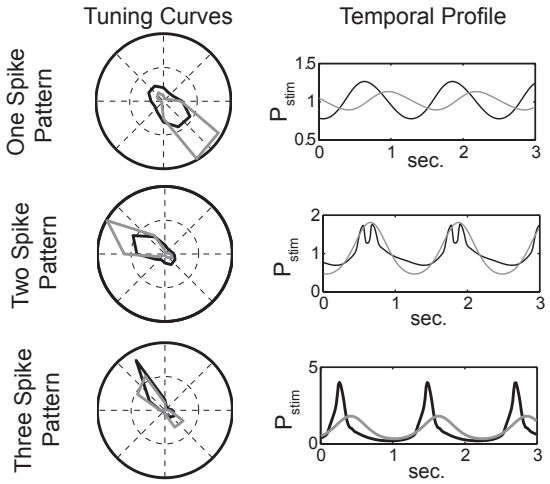

Figure 9: **Tuning curves and temporal profiles of three cat V1 patterns with different numbers of spikes.** Black=bagged regression trees, grey=independent neuron models.

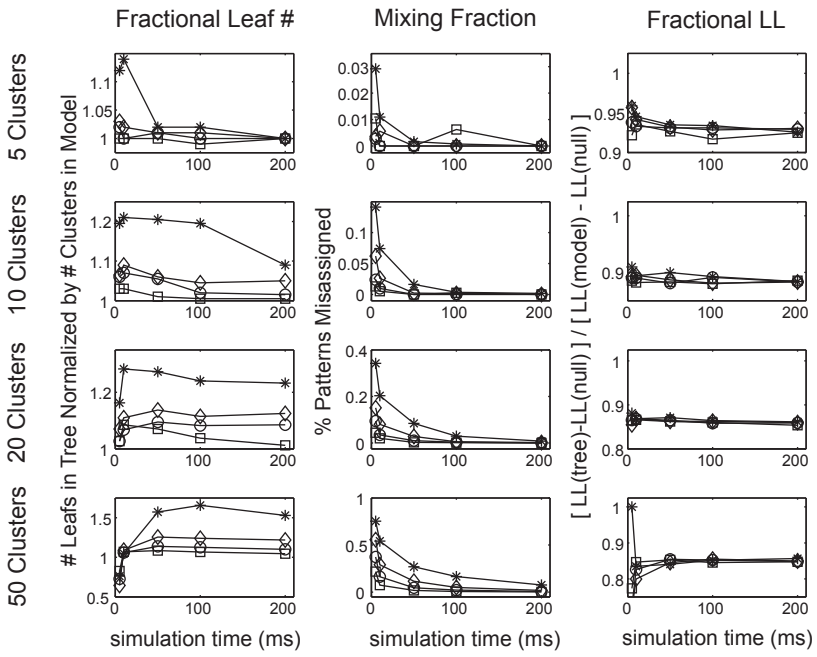

Figure 10: **Performance of regression tree algorithm on data generated from a multinomial logit model, with different numbers of classes, patterns per class, and data lengths (1 msec bins).** Squares: 5 patterns per class, circles: 10 patterns, diamonds: 20 patterns, asterixes 50 patterns. See Appendix B for details.

ARTICLE Communicated by Ron Meir

Discrete Time Rescaling Theorem: Determining Goodness of Fit for Discrete Time Statistical Models of Neural Spiking

Robert Haslinger
robhh@nmr.mgh.harvard.edu
Martinos Center for Biomedical Imaging, Massachusetts General Hospital, Charlestown, MA 02129, U.S.A., and Massachusetts Institute of Technology, Department of Brain and Cognitive Sciences, Cambridge, MA 02139, U.S.A.

Gordon Pipa
mail@g-pipa.com
Massachusetts Institute of Technology, Department of Brain and Cognitive Sciences, Cambridge, MA 02139, U.S.A.; Max-Planck Institute for Brain Research, Department of Neurophysiology, 60528 Frankfurt am Main, Germany; and Frankfurt Institute for Advanced Studies, 60438 Frankfurt am Main, Germany

Emery Brown
enb@neurostat.mit.edu
Massachusetts Institute of Technology, Department of Brain and Cognitive Sciences, Cambridge, MA 02139, U.S.A., and Massachusetts General Hospital, Department of Anesthesia and Critical Care, Boston, MA 02114, U.S.A.

One approach for understanding the encoding of information by spike trains is to fit statistical models and then test their goodness of fit. The time-rescaling theorem provides a goodness-of-fit test consistent with the point process nature of spike trains. The interspike intervals (ISIs) are rescaled (as a function of the model's spike probability) to be independent and exponentially distributed if the model is accurate. A Kolmogorov-Smirnov (KS) test between the rescaled ISIs and the exponential distribution is then used to check goodness of fit. This rescaling relies on assumptions of continuously defined time and instantaneous events. However, spikes have finite width, and statistical models of spike trains almost always discretize time into bins. Here we demonstrate that finite temporal resolution of discrete time models prevents their rescaled ISIs from being exponentially distributed. Poor goodness of fit may be erroneously indicated even if the model is exactly correct. We present two adaptations of the time-rescaling theorem to discrete time models. In the first we propose that instead of assuming the rescaled times to be exponential, the reference distribution be estimated through direct simulation by the fitted model. In the second, we prove a discrete time version of the time-rescaling theorem that analytically corrects for the effects of finite

resolution. This allows us to define a rescaled time that is exponentially distributed, even at arbitrary temporal discretizations. We demonstrate the efficacy of both techniques by fitting generalized linear models to both simulated spike trains and spike trains recorded experimentally in monkey V1 cortex. Both techniques give nearly identical results, reducing the false-positive rate of the KS test and greatly increasing the reliability of model evaluation based on the time-rescaling theorem.

1 Introduction

One strategy for understanding the encoding and maintenance of information by neural activity is to fit statistical models of the temporally varying and spike history–dependent spike probability (conditional intensity function) to experimental data. Such models can then be used to deduce the influence of stimuli and other covariates on the spiking. Numerous model types and techniques for fitting them exist, but all require a test of model goodness of fit, which is crucial to determine a model's accuracy before making inferences from it. Any measure of goodness of fit to spike train data must take the binary nature of such data into account (e.g., discretized in time, a spike train is a series of zeros and ones). This makes standard goodness-of-fit tests, which often rely on assumptions of asymptotic normality, problematic. Further, typical distance measures such as the average sum of squared deviations between recorded data values and estimated values from the model often cannot be computed for point process data.

One technique, proposed by Brown, Barbieri, Ventura, Kass, and Frank (2001) for checking the goodness of fit of statistical models of neural spiking, makes use of the time-rescaling theorem. This theorem states that if the conditional intensity function is known, then the interspike intervals (ISIs) of any spike train (or indeed any point process) can be rescaled so that they are Poisson with unit rate, that is, independent and exponentially distributed. Checking goodness of fit is then easily accomplished by comparing the rescaled ISI distribution to the exponential distribution using a Kolmogorov-Smirnov (KS) test (Press, Teukolsky, Vetterling, & Flannery, 2007; Massey, 1951). The beauty of this approach is not only its theoretical rigor, but also its simplicity, as the rescaling requires only the calculation of a single integral. Further, a second transformation takes the exponentially distributed rescaled times to a uniform distribution, and the KS test can then be performed graphically using a simple plot of the cumulative density function (CDF) of the rescaled times versus the CDF of the uniform distribution to determine if the rescaled times lie within analytically defined confidence bounds. Due to its many appeals, the time rescaling theorem has been extensively used to test model goodness of fit to spike train data (Frank, Eden, Solo, Wilson, & Brown, 2002; Truccolo, Eden, Fellows, Donoghue, & Brown, 2005; Czanner et al., 2008; Song et al., 2006).

There are, however, certain neurophysiological situations in which the standard time-rescaling approach can give misleading results, indicating poor goodness of fit when model fit may in fact be very good. This is a consequence of the practical numerical consideration that when a statistical model is fit to spike data one almost always discretizes time into bins. The time-rescaling theorem applies exactly to a continuous time point process (e.g., if we have infinite temporal precision and if the events, that is the spikes, are instantaneous). In a practical neuroscience setting, however, we usually do not have infinite temporal precision. First, a spike is an event that lasts for a finite ($\sim$1 msec) period of time, and any temporal resolution far below this lacks physical relevance.[1] Second, from a computational perspective, the fitting of statistical models requires much less computer time when the temporal discretization is coarser. Temporal discretization therefore imposes both physical and practical numerical constraints on the problem.

Often the probability per bin of a spike is small, and the distinction between continuous and discrete time of no concern, because the width of a spike is very short compared to the average interspike interval. Nevertheless, there are cases for which firing rates can be very high due to strong stimuli and ISIs short due to burst-type dynamics, and here the the per bin spike probability can be large even at 1 msec resolution or less. Such situations can arise in, for example, primate visual experiments where neurons can be extremely active (De Valois, Yund, & Hepler, 1982; MacEvoy, Hanks, & Paradiso, 2007; also see section 3 of this article), exhibiting firing rates of up to 100 Hz or more. In such situations, it is important to ensure that the rescaled ISIs are still (approximately) exponentially distributed and if not, to determine the correct distribution before performing the KS test.

Our aim in this article is to develop simple and easily applied goodness-of-fit tests for the discrete time case. We first restate the standard, continuous time form of the time-rescaling theorem for point processes and then demonstrate the discretization problem using a simple homogeneous Bernoulli (discretized homogeneous Poisson) process. We show theoretically that the discrete nature of the Bernoulli process results in first a lower bound on the smallest possible rescaled ISI, and second, because there can be only one spike per bin, a spike probability less than that which would be estimated by a continuous time model. These differences lead to biases in the KS plot caused by fundamental differences in the shapes of the geometric and exponential distributions, not by poor spike sampling or poor numerical integration techniques. We demonstrate further that these biases persist for more complicated simulated neural data with inhomogeneous

[1] This statement applies if one considers the spike as an event, as we do here. If one instead is interested in the shape and timing of the spike waveform—for example, the exact time of the waveform peak—then temporal resolutions of $\ll$1 msec may be physically relevant.

firing rates and burst-type spike history effects. We show that the biases increase when spike history effects are present.

We then propose two computationally tractable modifications to the time-rescaling theorem applicable to discrete time data. The first is similar in spirit to a bootstrap and involves direct simulation of confidence bounds on the rescaled ISI distribution using the statistical model being tested. In the second method, by randomly choosing exact spike times within each bin and introducing a correction to the fitted discrete spike probabilities, we define an analytic rescaled time that is exponentially distributed at arbitrary temporal discretizations. Use of this analytical method gives results nearly identical to the numerical approach. In this article, we use generalized linear models (GLMs) with logistic link functions (McCullagh & Nelder, 1989; Wasserman, 2004). However, we emphasize that both procedures will apply to any discrete time statistical model of the time-varying spike probability, not only GLMs. We demonstrate both approaches using simulated data and also data recorded from real V1 neurons during monkey vision experiments. In all our examples, the KS plot biases are eliminated. Models for which the original KS plots originally lay outside 95% confidence bounds are demonstrated to in fact be very well fit to the data, with the modified KS plots lying well within the bounds. In addition to providing more accurate statistical tests for discrete time spiking models, our approaches allow the use of larger time bin sizes and therefore can substantially decrease the computation time required for model fitting.

2 Theory

The time-rescaling theorem states that the ISIs of a continuous time point process can be transformed, or rescaled, so that the rescaled process is Poisson with unit rate (e.g., the rescaled ISIs are independent and exponentially distributed). This variable transform takes the form

$$\tau_i = \int_{t_{i-1}}^{t_i} \lambda(t \mid H_t)\, dt, \tag{2.1}$$

where $\{t_i\}$ is the set of spike times and $\lambda(t \mid H_t)$ is the conditional intensity function: temporally varying and history-dependent spike probability. Although we henceforth drop the H_t in our notation, such conditioning on the previous spiking history is always implied. Intuitively, the ISIs are stretched or shrunk as a function of total spike probability over the ISI interval so that the rescaled ISIs are centered about a mean of 1. Several proofs of this theorem exist (Brown et al., 2001). Here we present a simple proof of the exponential distribution of the rescaled ISIs. A proof of their independence is in appendix A.

The proof proceeds by discretizing time into bins of width Δ, writing down the probability for each discrete ISI, and then taking the continuous time limit: $\Delta \to dt$. The discrete time bins are indexed as k, and the bins within which the spikes occur are denoted as k_i. Further, we define p_k as the discrete probability mass of a spike in bin k, and like $\lambda(t)$, it should be taken as conditionally dependent on the previous spiking history.

The probability of the ith ISI is the probability that there is a spike in bin k_i given that the preceding spikes were located in bins $k_1, k_2, \ldots, k_{i-1}$:

$$P(ISI_i) = P(k_i \mid k_1, k_2, \ldots, k_{i-1}) = \left[\prod_{l=1}^{L_i-1}(1 - p_{k_{i-1}+l})\right] p_{k_{i-1}+L_i}, \qquad (2.2)$$

where L_i is defined such that $k_{i-1} + L_i = k_i$. This is simply the product of the probabilities that there are no spikes in bins $k = \{k_{i-1} + 1, k_{i-1} + 2, \ldots, k_i - 1\}$ multiplied by the probability that there is a spike in bin $k = k_i$. For simplicity, we now drop the i subscripts.

In preparation for taking the small bin size limit, we note that when Δ becomes small, so does p: $p \ll 1$ for all bins. This implies that $1 - p \approx e^{-p}$, allowing the above equation to be rewritten as

$$P(ISI) = P(k + L) \approx \exp\left[-\sum_{l=1}^{L} p_{k+l}\right] p_{k+L}. \qquad (2.3)$$

Note that the upper limit of the sum has been changed from $L - 1$ to L with the justification that we are in a regime where all the p's are very small. We define the lower and upper spike times as $t_k = k\Delta$ and $t = t_{k+L} = (k + L)\Delta$, define $\lambda(t_{k+l})$ such that $p_{k+l} = \lambda(t_{k+l})\Delta$,[2] and also define the ISI probability density $P(t)$ such that $P(k + L) = P(t)\Delta$. After substituting these into equation 2.3 and converting the sum to an integral, we obtain

$$P(t)\,dt = e^{-\int_{t_k}^{t} \lambda(t')\,dt'} \lambda(t)\,dt. \qquad (2.4)$$

Consulting equation 2.1, we note that the integral in the exponent is, by definition, τ. Further, applying the fundamental theorem of calculus to this

[2] $\lambda(t_{k+l}) = \langle \lambda(t) \rangle_{k+l}$, where the average is taken over the time bin k. This definition holds only when the bin size is very small. We will show that for moderately sized bins, $p_{k+l} \neq \lambda(t_{k+l})\Delta$, and that this leads to biases in the KS plot.

integral gives $d\tau = \lambda(t)\,dt$.[3] Changing variables from t to τ, we finally obtain

$$P(\tau)d\tau = e^{-\tau}d\tau, \tag{2.5}$$

which is now exponentially distributed and completes the proof.

Although the τ_i can be compared to the exponential distribution, it is useful to note that a second variable transform will make the rescaled ISIs uniformly distributed:

$$z_i = 1 - e^{-\tau_i}. \tag{2.6}$$

General practice is to sort the rescaled ISIs z_i into ascending order and plot them along the y-axis versus the uniform grid of values $b_i = \frac{i-0.5}{N}$, where N is the number of ISIs and $i = 1, \ldots, N$. If the rescaled ISIs z_i are indeed uniformly distributed, then this plot should lie along the 45 degree line. Essentially the cumulative density function (CDF) of the rescaled ISIs z_i is being plotted against the CDF of the uniform distribution (the b_i's). We show an example of such a plot in Figure 1. Such a plot can be thought of as a visualization of a KS test, which compares two CDFs and is usually referred to as a KS plot. Formally we can state the null hypothesis H_0 of this test as follows:

H_0: Given a model of the conditional intensity function that is statistically adequate, the experimentally recorded ISIs can be rescaled so that they are distributed in the same manner as a Poisson process (exponentially distributed) with unit rate.

Under the null hypothesis, the maximum distance between the two CDFs will, in 95% of cases, be less than $\frac{1.36}{\sqrt{N}}$, where N is the number of rescaled ISIs (Brown et al., 2001; Johnson & Kotz, 1970). Equivalently, the plotted line of rescaled ISIs will lie within the bounds $b_k \pm \frac{1.36}{\sqrt{N}}$ in 95% of cases under the null hypothesis. It should be kept in mind that this is not equivalent to saying that the line of rescaled ISIs lying within these bounds implies a 95% chance of the model being correct.

2.1 Temporal Discretization Imposes KS Plot Bias. The time-rescaling theorem applies exactly to a point process with instantaneous events (spikes) and infinite temporal precision (i.e., continuous time). As a practical matter, one generally discretizes time when fitting a statistical model.

[3] Specifically,

$$\frac{d\tau}{dt} = \frac{d}{dt}\int_{t_k}^{t} \lambda(t')\,dt' = \lambda(t),$$

and therefore $d\tau = \lambda(t)\,dt$.

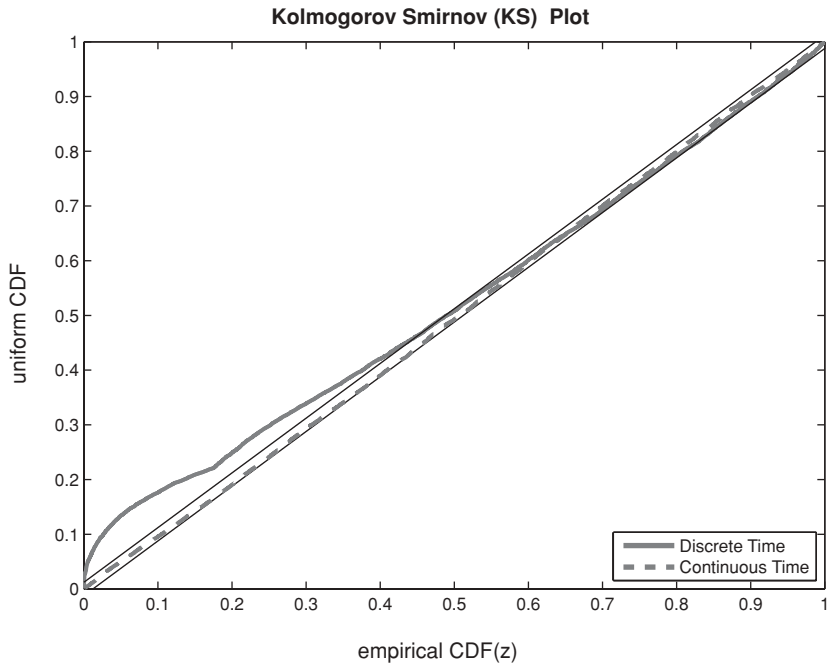

Figure 1: Two simple KS plots demonstrating that temporal discretization induces biases even if the conditional intensity function used to calculate the rescaled times is exactly correct. CDF of the rescaled times z is plotted along the x-axis versus the CDF of the uniform (reference) distribution along the y-axis. Spikes were generated from an inhomogeneous Poisson process with a maximum firing rate of 50 Hz. Thick gray dashed line: KS plot of rescaled ISIs generated by a continuous time model. Thick gray solid line: KS plot of rescaled ISIs calculated from the same model discretized at 5 msec resolution. The discretization was deliberately enhanced to emphasize the effect. Thin black 45 degree lines are 95% confidence bounds on the KS plots.

For discrete time, the integral of equation 2.1 is naively replaced by

$$\tau_i = \sum_{k=k_{i-1}+1}^{k_i} p_k. \tag{2.7}$$

If $p_k \ll 1 \; \forall k$ (i.e., situations where either the bin size is very small $\Delta \to 0$ or the firing rate is very low), the time-rescaling theorem will apply approximately even if a discrete time model is used. However, it often happens that p_k is in fact large. For example, 50 Hz spiking sampled at 1 msec implies $p \approx 0.05$, and under many conditions, the firing rate can be much

higher, at least over some subset of the recording (e.g., during bursting). In such cases, the rescaled times τ_i will not be exponentially distributed, and the KS plot will exhibit significant biases (divergences from the 45% line) even if the discrete time model for p_k is exactly correct. We demonstrate this in Figure 1 where two KS plots generated using the exact same spikes and time-varying firing rate are shown, but a temporal discretization was imposed for one of the plots.

These biases originate in two distinct consequences of discretizing a continuous process. First, there is a lower bound on the smallest possible ISI (one bin), which leads to a lower bound on the smallest possible rescaled time z. Second, because only a single spike per bin is allowed, using a discrete time model to estimate the firing rate of a continuous time process results in an underestimation of the firing rate. To demonstrate these issues fully, we now consider the simple case of a homogeneous Bernoulli process with a constant spike probability $p_k = p$ per bin for which the the CDF of the z's can be calculated analytically and the KS plot determined exactly.

For a discrete time process, only a discrete set of ISIs is possible—specifically $\{n\Delta\}$, where n is an integer greater than zero and Δ is the bin width. In the case of a homogeneous Bernoulli process, the rescaled ISIs are $\tau(n) = pn$ and

$$z(n) = 1 - e^{-pn}, \tag{2.8}$$

and the discrete probability distribution of interspike interval times (and rescaled times) is

$$P_B(n) = (1 - p)^{n-1} p. \tag{2.9}$$

As in equation 2.2, this is merely the product of the probability of no spike for $n - 1$ bins, followed by the probability of a spike in the last (nth) bin. The B subscript indicates the Bernoulli process. $P_B(n)$ is not an exponential distribution, as would be expected for a homogeneous Poisson process. It is a geometric distribution, although in the limit of small p it reduces to an exponential distribution.[4] The CDF of this ISI distribution is easily calculated by summing the geometric series and combining terms:

$$\begin{aligned} CDF_B(n) = \sum_{j=1}^{n} P_B(j) &= \frac{p}{1-p} \sum_{j=1}^{n} (1-p)^j \\ &= 1 - (1-p)^n. \end{aligned} \tag{2.10}$$

[4]Setting $p = \lambda\Delta$ and $t = n\Delta$, $P_B(t) = (1-p)^{n-1} p = \frac{\lambda\Delta}{1-\lambda\Delta}(1-\lambda\Delta)^{t/\Delta} \to \lambda e^{-\lambda t} dt = P_P(t)\, dt$, when the limit $\Delta \to dt$ is taken.

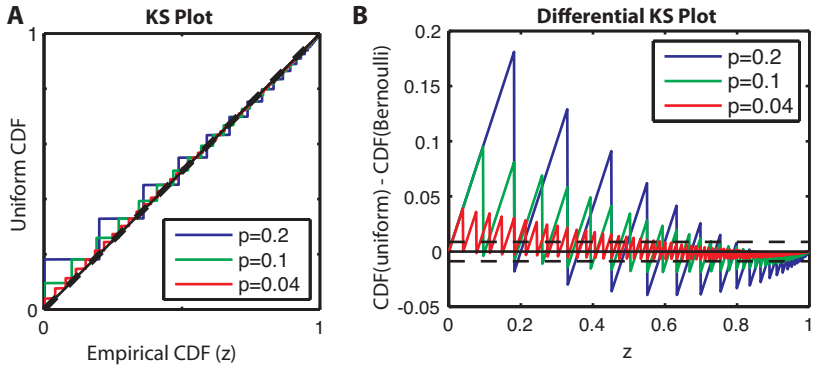

Figure 2: Illustration of KS plot bias induced when a homogeneous Poisson process is discretized to a homogeneous Bernoulli process. (A) KS plot for various spike per bin probabilities p. Blue: $p = 0.2$, green: $p = 0.1$, red: $p = 0.04$ (40 Hz at 1 msec discretization). The rescaled times are not uniformly distributed but have positive bias at rescaled ISIs close to 0 and negative bias at rescaled ISIs close to 1. (B) Differential KS Plot: $CDF_{uniform} - CDF(z)_{Bernoulli}$. Biases are easier to see if the difference between the expected CDF (uniform) and the actual CDF of the rescaled times is plotted. The colors indicate the same spike per bin probabilities p as in A. The horizontal dashed lines are the 95% confidence region assuming 10 minutes of a 40 Hz Bernoulli process (24,000 spikes).

To get the CDF of the rescaled ISIs z, equation 2.8 is inverted to get $n = -\frac{\log(1-z(n))}{p}$ and substituted into equation 2.10:

$$CDF_B(z) = 1 - (1-p)^{-\frac{\log(1-z(n))}{p}} \qquad z(n-1) \le z \le z(n). \tag{2.11}$$

In Figure 2 we use equation 2.11 to generate the KS plot for various spikes per bin probabilities p. Even at $p = 0.04$, which would correspond to 40 Hz firing at 1 msec discretization, the CDF is highly nonuniform with a steplike structure caused by the discrete values that the rescaled ISIs can take. Such "steps" will be smoothed out if an inhomogeneous Bernoulli process is used instead. There is, however, another more serious divergence from uniformity: a distinct positive bias at low (close to 0) rescaled ISIs and a distinct negative bias at high (close to 1) rescaled ISIs. This bias will not disappear if an inhomogeneous Poisson process is used.

The dashed lines, which are barely visible, denote the 95% confidence region of the KS plot assuming 10 minutes of 40 Hz spiking, which translates into 24,000 spikes on average. Since the confidence bounds are so close to the 45 degree line, and will be for any spike train with a long recording time and appreciable firing rate, we introduce a new type of plot in Figure 2,

which we term a differential KS plot. This is simply a plot of the difference between the distribution we hypothesize that the CDF of rescaled times should follow (in this case uniform) and the CDF of the experimentally recorded rescaled ISIs (in this case the rescaled ISIs of the Bernoulli process):

$$CDF_{hyp}(z) - CDF_{exp}(z). \tag{2.12}$$

The differential KS plot displays the same information as the KS plot, but does so in a different and more visually accessible manner. The confidence bounds (the horizontal dashed lines in Figure 2) are now simply given by $\pm\frac{1.36}{\sqrt{N}}$, where N is again the number of rescaled ISIs. Plotted this way, one can clearly see the positive bias at low values of the rescaled ISIs and the negative bias at high values of the rescaled ISIs. We emphasize that since these KS and differential KS plots are calculated using the exact homogeneous Bernoulli distribution, the biases are not finite sampling effects.

The positive bias at low ISIs is easily understood by noting that the smallest possible rescaled time is not zero but

$$z(1) = 1 - e^{-p} = p - \frac{p^2}{2} + \cdots > 0. \tag{2.13}$$

What about the negative bias at large (z close to 1) rescaled ISIs? Consider a homogeneous Poisson process with a firing rate λ. Upon discretizing time into bins of width Δ, one might naively expect the probability of a spike per bin to be $p = \lambda\Delta$. However, it is in fact slightly less than this, as we now show. Assume a spike at time $t = 0$. Then for a homogeneous Poisson process, the probability density for the waiting time t_w until the next spike is $\rho(t_w) = \lambda e^{-\lambda t_w}$. Integrating, the probability that the next spike lies within any interval $t < t_w \leq t + \Delta$ can be obtained:

$$P(t < t_w \leq t + \Delta) = \int_t^{t+\Delta} \lambda e^{-\lambda t'} dt' = e^{-\lambda t}(1 - e^{-\lambda\Delta}). \tag{2.14}$$

Defining the bin index n such that $t = (n-1)\Delta$ and discretizing, we get

$$\begin{aligned} P(n_w = n) &= e^{-\lambda\Delta(n-1)}(1 - e^{\lambda\Delta}) \\ &= [1 - (1 - e^{-\lambda\Delta})]^{n-1}(1 - e^{-\lambda\Delta}) \\ &= (1-p)^{n-1} p, \end{aligned} \tag{2.15}$$

where we have defined $p = 1 - e^{-\lambda\Delta}$ in the last line. Discretizing time transforms the homogeneous Poisson process into a homogeneous Bernoulli process, but with a per bin probability of a spike $p \neq \lambda\Delta$. In fact, by expanding

the exponential as a Taylor series, it can be seen that

$$p = 1 - e^{-\lambda\Delta} = \lambda\Delta - \frac{(\lambda\Delta)^2}{2} + \cdots < \lambda\Delta. \tag{2.16}$$

The continuous Poisson process still has an expected number of spikes per interval of width Δ of $\int_0^\Delta \lambda dt = \lambda\Delta$, but such an interval could have more than one spike in it. In contrast, the discrete Bernoulli process can have only 0 or 1 spikes per bin. Therefore the per bin "spike probability" p calculated above is not the expected number of spikes of the continuous point process within an interval Δ. It is the expected number of first spikes in an interval Δ, which is, of course, less than the total number of expected spikes. Any chance of there being more than one spike in a time window Δ has been eliminated by discretizing into bins.

The breakdown of the first-order expansion of the exponent is the source of the negative KS plot bias at high (z close to 1) rescaled ISIs. It is a fundamental consequence of discretizing a continuous time point process and is closely connected to how the conditional intensity function is generally defined, that is, as the small bin size limit of a counting process (see Snyder, 1975). More specifically the conditional intensity function is the probability density of single spike in an infinitesimal interval $[t, t + \Delta)$. As shown above, this probability density is actually $p/\Delta = (1 - e^{-\lambda\Delta})/\Delta < \lambda$, and the equality holds only in the limit. Thus, p/Δ is not a good approximation for λ when the bin size is too large, and this causes the time-rescaling theorem to break down.

2.2 Inhomogeneous Bernoulli Processes. The same positive (negative) bias in the KS plot at low (high) rescaled ISIs remains when the spiking process is not homogeneous Bernoulli. We now we define three inhomogeneous spiking models in continuous time and subsequently discretize them. We use these inhomogeneous discrete time models to simulate spikes and then calculate the rescaled ISIs using the exact discrete time model used to generate the spikes in the first place. The goal is to show that even if the exact discrete time generative model is known, the continuous time-rescaling theorem can fail for sufficiently coarse discretizations.

The first model is an inhomogeneous Bernoulli process. One second of the inhomogeneous firing probability is shown in Figure 3A. The specific functional form was spline based, with knots spaced every 50 msec and the spline basis function coefficients chosen randomly. This model firing probability was repeated 600 times for 10 minutes of simulated time. The second and third models were the homogeneous and inhomogeneous Bernoulli models, respectively, but with the addition of a spike history–dependent renewal process shown in Figure 3B. We used a multiplicative model for

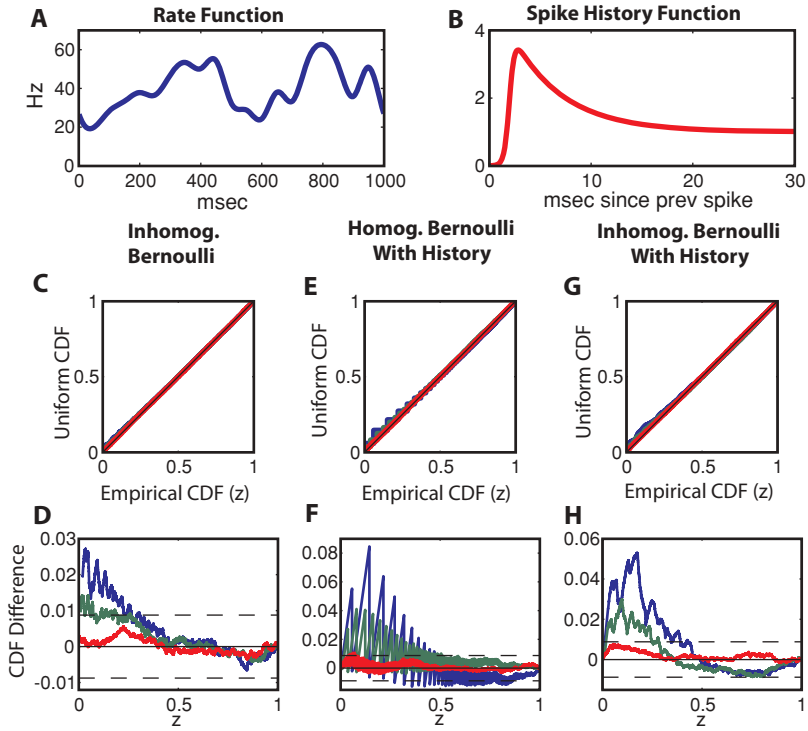

Figure 3: KS and differential KS plots for 10-minute-long 40 Hz mean firing rate simulated spike trains. Three continuous time models of the conditional intensity function were used for simulation: inhomogeneous Poisson process, homogeneous Poisson with a renewal spike history process, and inhomogeneous Poisson with a renewal spike history process. (See the text.) The continuously defined processes were discretized at various values Δ and used to simulate spikes. (A) 40 Hz mean inhomogeneous Bernoulli firing rate. (B) Spike history term λ_{hist} as a function of time since the most recent spike. (C, D) KS and differential KS plots for inhomogeneous Bernoulli process. Blue: $\Delta = 1$ msec, green: $\Delta = 0.5$ msec, red: $\Delta = 0.1$ msec. Horizontal dashed lines are 95% confidence bounds. (E, F) Homogeneous Bernoulli process with spike renewal history term. (G, H) Inhomogeneous Bernoulli process with spike renewal history term. Note that when spike history effects are present, the biases are larger at both short and long rescaled ISIs.

the history–dependent firing probabilities of the form

$$\lambda(t) = \lambda_0(t)\lambda_{hist}(t - t_{ls}), \tag{2.17}$$

where $\lambda_0(t)$ is the time-dependent firing probability independent of spike history effects and λ_{hist} is the spike history–dependent term, which is a

function of the time since the last spike ($t' = t - t_{ls}$). The functional form of the spike history–dependent term was a renewal process, specifically

$$\lambda_{hist}(t') = \frac{1 + 3e^{-(t'-2)/5}}{1 + e^{-4(t'-2)}}, \tag{2.18}$$

where $t' = t - t_{ls}$ is in msec. This form was chosen to mimic a brief refractory period and subsequent rebound. For comparison purposes, all three of these models were constructed so that the mean firing rate remained approximately 40 Hz. Thus, the inhomogeneous Bernoulli firing probability had a 40 Hz mean. In the spike history–dependent cases, the history-independent firing probabilities $\lambda_0(t)$ were adjusted downward so that when history effects were included, the mean firing rate remained approximately 40 Hz. Specifically, the history-independent firing probability of the homogeneous Bernoulli process was reduced to 29 Hz, and a similar reduction was made for the inhomogeneous Bernoulli model.

In Figure 3, we demonstrate the effect on the KS and differential KS plots when these models are subjected to various temporal discretizations. Specifically, we discretized the models at 1, 0.5, and 0.1 msec resolution by averaging $\lambda_0(t)$ over these bin widths: $p_{k,0} = \langle \lambda_0(t) \rangle_k$. The spike history–dependent term is a function of $t' = t - t_{ls}$, which was also partitioned into bins. Similar averaging was then employed so that $p_{k',hist} = \langle \lambda_{hist}(t') \rangle_{k'}$. The full discrete conditional spike probability is then $p_k = p_{k,0}\, p_{k-k_{ls},hist}$, where k_{ls} is defined as the most recent bin prior to bin k that has a spike in it. We then simulated 10 minutes worth of spikes for each model and discretization.[5] After generating the spikes, we then calculated the rescaled times and CDF difference plots according to

$$z_i = 1 - e^{-\sum_{k=k_{i-1}}^{k_i} p_k}. \tag{2.19}$$

Figures 3C and 3D show the results for the inhomogeneous Bernoulli model. Comparison with Figure 2 reveals that the main effect of inhomogeneity is to smooth out the steps. The positive (negative) biases at low (high) rescaled times remain, and, as expected, they are smaller for finer temporal discretizations. Figures 3E to 3H show the results when the spike history–dependent term is added to both the homogeneous and inhomogeneous Bernoulli models. The important point is that the biases are worse for both models when spike history effects are included, even though the models are constructed so that the mean firing rate remains 40 Hz. The

[5] For the spike history–dependent models, the generation of a spike in bin k modifies the firing probabilities in bins $k' > k$. Thus, the simulation proceeded bin by bin, and on generation of a spike, the firing probabilities in the following bins were updated according to equation 2.18 before generating the next observation (spike or no spike) in bin $k + 1$.

reason is that the history-dependent term is constructed so that the spike train exhibits burstlike behavior. Specifically, after a short (2 msec) refractory period there is an increased probability of a spike. This increases the number of short ISIs. It also increases the smallest possible rescaled ISI z because the probability of a spike goes up immediately following a prior spike, and this ends up shifting distributional weight to short ISIs. This is an important point because it implies that in real experimentally recorded spike trains, which may exhibit burst-type behavior, the bias in the KS plot will be worse than would be expected by a simple estimate based on the mean firing rate, as given in equation 2.13.

2.3 Unbiased Discrete Time Rescaling Test Using Model Simulation. In a previous section, we showed analytically that when discrete time models are used, the rescaled ISIs may not be exponentially distributed even if the model is exactly correct and that this manifests in the KS plot as systematic biases. Our first proposed solution (we present a second in the following section) to the bias problem is not to assume that the rescaled ISIs are exponentially (or uniformly) distributed, but to instead use a procedure similar to bootstrapping. This proceeds by noting that if a candidate model accurately describes the recorded spikes, then the rescaled ISI distribution of the spikes and the rescaled ISI distribution expected by the fitted model should be statistically indistinguishable. If instead the model form is inappropriate to describe the spiking data, then the rescaled ISI distribution expected by the candidate model will not match that of the experimentally recorded spikes, because the model does not describe the recorded spikes accurately. Although the expected distribution of rescaled ISIs is implicitly defined by the fitted model, in practice an explicit analytical form for this distribution may be hard to come by. It can, however, be sampled numerically using the fitted model to generate spikes and rescaling the resulting ISIs as a function of the model used to generate them.[6]

Specifically, after a candidate model is proposed and fit to the recorded spike train data (any type of candidate model may be used as long as it provides an estimate of the conditional intensity function λ), we use the model to simulate spikes, rescale the resulting simulated ISIs, and then use a two-sample KS test to determine if the sample of estimated rescaled ISIs $\{z_{est}\}$ and the sample of experimentally recorded rescaled ISIs $\{z_{exp}\}$ are consistent with being drawn from the same underlying distribution (Press et al., 2007). Formally, the null hypothesis of the KS test has been changed

[6]Most generally, the conditional intensity function will have the form $\lambda(t_k) = \lambda(x(t_k) \mid H(t_k))$, where $x(t_k)$ is the set of time-varying external covariates and $H(t_k)$ is the previous spiking history. As the originally recorded spike train was of length T and the external covariates were defined over this time interval, it is simplest to simulate multiple spike trains the length of the original recording time T. For each spike train simulation, $x(t)$ remains the same, but $H(t)$ will differ depending on the exact spike times.

from that stated in section 2 and adapted to the case of discrete time data using a model-based approach:

H_0(estimated): Given a model of the conditional intensity function that is statistically adequate, the set of experimentally recorded ISIs can be rescaled so that they are distributed in the same manner as a sample of rescaled ISIs generated by the statistical model itself.

To determine the number of spikes (or length of time) that must be simulated, we use the analytical expression for the confidence bounds of the two-sample KS test. These bounds are a function of the sample sizes of the distributions being compared—in our case, the size of the empirical distribution N_{exp} dictated by experiment and the size of the simulated distribution N_{sim}, which we can chose. The two-sample KS test determines the maximum difference between the CDFs of the experimentally recorded rescaled ISIs $\{z_{exp}\}$ and the set of rescaled ISIs simulated using the model $\{z_{sim}\}$. If the maximum difference between the two CDFs is less than a certain value, specifically,

$$\max |C\,DF_{sim}(z) - C\,DF_{exp}(z)| < 1.36\sqrt{\frac{N_{exp} + N_{sim}}{N_{exp}N_{sim}}}, \tag{2.20}$$

then the null hypothesis is confirmed at the $\alpha = 0.05$ significance level (Press et al., 2007). Alternatively a differential KS plot (as already discussed) will have 95% confidence bounds of

$$\pm 1.36\sqrt{\frac{N_{exp} + N_{sim}}{N_{exp}N_{sim}}} = \pm\frac{1.36}{\sqrt{N_{exp}}}\sqrt{\frac{1+\gamma}{\gamma}}, \tag{2.21}$$

where we have written $N_{sim} = \gamma N_{exp}$.

Since N_{exp} is fixed by experiment, the test will be strictest (tightest confidence bounds) when $N_{sim} \to \infty$ or, equivalently, as γ increases. Formally, increasing N_{sim} increases the power of the KS test and reduces the number of false positives (false rejections of the null hypothesis). Fortunately N_{sim} need not be overly large. Already at $\gamma = 20$, the confidence bounds are only a factor of 1.02 wider than they would be in the infinite limit ($\pm 1.36/\sqrt{N_{exp}}$) in which the exact distribution would be known. This implies that a simulated spike train 20 times longer than the original, experimentally recorded, spike train provides a test power close to optimal and is sufficient to approximate the confidence bounds extremely well. In section 3, we use simulated spike trains 100 times the original experimental length ($\gamma = 100$), which widens the confidence bounds by a factor of only 1.0005.

Specifically, this technique proceeds as follows:

Procedure for Numerical Correction

1. Discretize the spike train into bins of width Δ, and fit a discrete time statistical model.
2. Rescale the experimentally recorded ISIs using equation 2.7 to obtain the set of rescaled ISIs $\{z_{exp}\}$.
3. Use the statistical model to estimate the rescaled ISI distribution. Simulate $\gamma > 20$ spike trains of the same length as the original experimentally recorded spike train, and rescale the simulated ISIs $\{z_{sim}\}$.
4. Construct the CDFs of both z_{exp} and z_{sim}, take the difference, and plot it on the interval [0, 1] with the confidence bounds $\pm \frac{1.36}{\sqrt{N_{exp}}} \sqrt{\frac{1+\gamma}{\gamma}}$.

2.4 Discrete Time Version of the Time-Rescaling Theorem. We now prove a discrete time version of the time-rescaling theorem that corrects for both sources of KS plot bias. Specifically, we demonstrate how to write a new rescaled time ξ, which is exponentially distributed for arbitrary temporal discretization. The proof given here assumes that an underlying continuous time point process $\lambda(t \mid H_t)$ is sampled at finite resolution Δ.

Proposition. *Suppose a continuous time point process $\lambda(t \mid H_t)$ is sampled at finite resolution so that the observation interval from $(0 \mid T]$ is partitioned into bins of width Δ. Denote the bin in which the ith spike is located as k_i and that of the next spike as bin $k_{i+1} = k_i + L_i$. Let $p_{k_i+l} = p(k_i + l \mid H_{k_i+l})$ be the discrete time conditional spike probabilities evaluated in bins k_{i+l} for $l = 1, \ldots, L_i$. Define the random variable*

$$\xi_i = \sum_{l=1}^{L_i-1} q_{k_i+l} + q_{k_i+L_i} \frac{\delta_i}{\Delta}, \tag{2.22}$$

where

$$q_{k_i+l} = -log(1 - p_{k_i+l}), \tag{2.23}$$

and $\delta_i \in [0, \Delta]$ is a random variable determined by first drawing a uniform random variable $r_i \in [0, 1]$ and then calculating

$$\delta_i = -\frac{\Delta}{q_{k_i+L}} log[1 - r_i(1 - e^{-q_{k_i+L_i}})]. \tag{2.24}$$

Then ξ_i has an exponential PDF with unit rate. For clarity of notation we drop the subscript i in the following proof. It should be taken as implicit.

Proof. Assume the last spike was located in bin k and the next spike in bin $k + L$. If we knew the underlying continuous time conditional intensity function $\lambda(t \mid H_t)$ and the exact spike time $t_\delta = (k + L - 1)\Delta + \delta$ in bin $k + L$ ($\delta \in [0, \Delta]$) then using the continuous time version of the time-rescaling theorem, we could write the probability of this event as

$$P(t_\delta)\,dt = e^{\int_{k\Delta}^{t_\delta} \lambda(u)\,du} \lambda(t_\delta)\,dt = e^{-\tau} d\tau, \tag{2.25}$$

which is exponentially distributed in τ. Since we know neither $\lambda(t \mid H_t)$ nor t_δ precisely, we must recast τ in terms of what we do know: the discrete bin-wise probabilities p_{k+l}.

The p_{k+l}'s can be written in terms of the underlying continuous process $\lambda(t \mid H_t)$. Consider any bin $k + l$. Since discretization enforces at most one spike per bin, p_{k+l} does not equal the integral of $\lambda(t \mid H_t)$ over the bin, but rather the probability (measured from the start of the bin) that the first spike waiting time is less than Δ:

$$\begin{aligned} p_{k+l} &= \int_{(k+l-1)\Delta}^{(k+l)\Delta} e^{\int_{(k+l-1)\Delta}^{t} \lambda(u)\,du} \lambda(t)\,dt \\ &= 1 - e^{-\int_{k\Delta}^{(k+1)\Delta} \lambda(u)\,du}. \end{aligned} \tag{2.26}$$

Partitioning the integral in the exponent of equation 2.25 into a sum of integrals over each bin allows $P(t_\delta)$ to be written as

$$P(t_\delta)\,dt = \exp\left[-\sum_{l=1}^{L-1} q_{k+l} - \int_{(k+L-1)\Delta}^{t_\delta} \lambda(u)\,du \right] \lambda(t_\delta)\,dt, \tag{2.27}$$

where we have introduced $q_{k+l} = \int_{(k+l-1)\Delta}^{(k+l)\Delta} \lambda(u)\,du$ as shorthand. By inverting equation 2.26, q_{k+l} can be written directly in terms of p_{k+l}:

$$q_{k+l} = -\log(1 - p_{k+l}). \tag{2.28}$$

Since we have no information about how $\lambda(t \mid H_t)$ varies over bin $k + L$, we can pick any functional form as long as it obeys the constraint that its integral over the bin equals $q_{k+L} = -\log(1 - p_{k+L})$. One choice is $\lambda(t \mid H_t) = \frac{q_{k+L}}{\Delta}$.[7] It then follows that

$$P(t_\delta)\,dt = \exp\left[-\sum_{l=1}^{L-1} q_{k+l} - \frac{q_{k+L}}{\Delta}\delta \right] \frac{q_{k+L}}{\Delta} dt = e^{-\xi} d\xi. \tag{2.29}$$

[7] In fact any form for λ within bin $k + L$ could be chosen. Choosing it to be constant merely allows easier random sampling.

$P(t_\delta)\,dt$ has now been rewritten in terms of what we know: the q_{k+l}'s (implicitly the p_{k+l}'s). We have defined the rescaled time as

$$\xi = \sum_{l=1}^{L-1} q_{k+l} + \frac{q_{k+L}}{\Delta}\delta \tag{2.30}$$

(where $d\xi = \frac{q_{k+L}}{\Delta} d\delta$) to distinguish it from the rescaled time of the continuous time version of the theorem, which directly sums the p_{k+l} and does not require random sampling of the exact spike time δ.

Random sampling of $\delta \in [0, \Delta]$ must respect the choice of $\lambda(t \mid H_t) = \frac{q_{k+L}}{\Delta}$ and the fact that we know the spike is in the bin somewhere. For our choice of λ, the probability density of δ conditioned on there being a spike in the bin is a truncated exponential:

$$P(\delta \mid \text{spike})d\delta = \frac{e^{-\frac{q_{k+L}}{\Delta}\delta}}{1 - e^{-q_{k+L}}} \frac{q_{k+L}}{\Delta} d\delta. \tag{2.31}$$

The numerator is simply the event time probability measured from the start of bin $k+L$. The denominator is a normalization obtained by integrating the numerator between 0 and Δ.[8] To draw from this distribution, we integrate it to obtain its CDF,

$$\text{CDF}(\delta \mid \text{spike}) = \frac{1 - e^{-q_{k+L}\delta/\Delta}}{1 - e^{-q_{k+L}}}; \tag{2.32}$$

set this cdf equal to a uniform random variable r; and then solve for δ

$$\delta = -\frac{\Delta}{q_{k+L}} \log[1 - r(1 - e^{-q_{k+L}})]. \tag{2.33}$$

This completes the proof.[9]

There are two differences between the discrete and continuous time versions of the theorem. The first, and more fundamental, difference is that $p = 1 - \exp[-\int_0^\Delta \lambda(t)\,dt]$, not $\int_0^\Delta \lambda(t)\,dt$. The latter is true only when Δ is small. Expanding the logarithm of equation 2.28, we obtain

$$q = \int_0^\Delta \lambda(t)\,dt = -\log(1 - p) = p - \left[\frac{p^2}{2} - \frac{p^3}{3} + \cdots\right]. \tag{2.34}$$

[8] Or Bayes' rule could be used: $P(\delta \mid \text{spike}) = P(\delta)/P(\text{spike}) = \{\exp[-q_{k+L}\delta/\Delta]\frac{q_{k+L}}{\Delta}\}/p_{k+L} = \{\exp[-q_{k+L}\delta/\Delta]\frac{q_{k+L}}{\Delta}\}/(1 - e^{-q_{k+L}})$.

[9] Since δ is chosen randomly, rescaling will give slightly different results if performed multiple times. For all results presented in this article, such variation was negligible when considered at the scale of the KS plot's 95% confidence bounds. Further, 95% confidence bounds on the variability can be calculated analytically for a discretized homogeneous Poisson process. These bounds are given by $\pm\min(p, 1.36\sqrt{2p/N})$ and are always smaller than $\pm 1.36\sqrt{2/N}$, the bounds of a two-sample KS test.

To properly rescale the ISIs when Δ is large, all the terms in the Taylor series must be kept. This can be thought of as introducing a correction term (in the brackets) for the finite bin size. Equivalently, the approximation $1 - p \approx e^{-p}$ used in the continuous time version of the proof is not valid for large p (or Δ). The second difference is that we randomly choose an exact spike time $t_\delta = (k + L - 1)\Delta + \delta$ according to the distribution given in equation 2.31. This is done because there is no information about where exactly in bin $k + L$ the spike is located, and for the rescaled time ξ to be exponentially distributed, it must be continuously valued. In the continuous time limit, both of these distinctions vanish.

The hypothesis for testing goodness of fit is now exactly the same as that of the original time-rescaling theorem, except that the rescaling is modified to take into account the discretization. Reintroducing the subscript i to denote the individual spike times k_i, the procedure for performing the KS test is simply described.

Procedure for Analytical Correction

1. Discretize the spike train into bins of width Δ with the spikes in bins $\{k_i\}$ and fit a discrete time statistical model resulting in the spike per bin probabilities p_k.
2. Generate a new time series of discrete values q_k according to

$$q_k = -\log(1 - p_k). \tag{2.35}$$

3. For each interspike interval, calculate the rescaled ISI ξ_i according to

$$\xi_i = \sum_{l=1}^{L_i - 1} q_{k_i + l} + q_{k_i + L_i} \frac{\delta_i}{\Delta}, \tag{2.36}$$

where δ is a random variable determined by first drawing a uniform random variable $r_i \in [0, 1]$ and then calculating

$$\delta_i = -\frac{\Delta}{q_{k_i + L_i}} \log[1 - r_i(1 - e^{-q_{k_i + L_i}})]. \tag{2.37}$$

4. Make a final transform to the random variables y_i:

$$y_i = 1 - e^{-\xi_i}. \tag{2.38}$$

If the discrete time statistical model is accurate, the y_i will be uniformly distributed. Therefore, the y_i can be used to make a KS or differential KS plot.

3 Results

In this section we fit GLMs to spike trains both simulated and experimentally recorded in awake monkey V1 cortex during visual stimulation. We then check goodness of fit using both the standard KS test and our methods.

We demonstrate dramatic and nearly identical improvement in KS test accuracy for both techniques. Although we emphasize that any discrete time statistical model may be used, we chose a GLM, specifically the logistic regression (logit link function) form, because of the discrete binary nature of spike train data. Standard linear regression assumes continuous variables and is therefore inappropriate for the problem. Further reasons for using GLMs are their already wide application to the analysis of neural spiking activity (Frank et al., 2002; Truccolo et al., 2005; Czanner et al., 2008; Paninski, 2004a; Kass & Ventura, 2001), their optimality properties (Pawitan, 2001), and the ease of fitting them via maximum likelihood. Methods for fitting GLMs exist in most statistical packages, including Matlab and R.

3.1 Simulated Data. Using the three continuous time point process models of the previous section (inhomogeneous Poisson, homogeneous Poisson with spike history dependence, and inhomogeneous Poisson with spike history dependence), we simulated 10 minutes of spikes from each model at very fine 10^{-10} msec discretization, essentially continuous time. These spike trains are the experimental data. We emphasize that all of our simulated data used a realistic mean firing rate of 40 Hz, and that many experimental situations exist for which the mean firing rates are much higher (De Valois et al., 1982; MacEvoy et al., 2007). The spikes were then discretized into 1 msec bins, and a GLM was fit to each simulated spike train. This procedure mimics the usual approach taken in fitting a GLM to real data. We used a logistic regression type GLM (logit link function) appropriate for discrete time binary data. Each model's spike train was fit using one of the following GLM forms:

- Inhomogeneous Bernoulli GLM:

$$\log\left[\frac{\lambda(k)}{1-\lambda(k)}\right] = \sum_{j=1}^{J} \beta_j \mathcal{B}_j(k) \tag{3.1}$$

- Homogeneous Bernoulli with spike history GLM:

$$\log\left[\frac{\lambda(k)}{1-\lambda(k)}\right] = \beta_0 + \sum_{r=1}^{R} \theta_r g(k-r) \tag{3.2}$$

- Inhomogeneous Bernoulli with spike history GLM:

$$\log\left[\frac{\lambda(k)}{1-\lambda(k)}\right] = \sum_{j=1}^{J} \beta_j \mathcal{B}_j(k) + \sum_{r=1}^{R} \theta_r g(k-r) \tag{3.3}$$

The $\mathcal{B}_j(k)$ are periodic B-spline basis functions with knots spaced 50 msec apart. These are continuously defined (even though we use discrete time

bins), temporally localized basis functions, similar in shape to gaussians, which can be computed recursively (Wasserman, 2007). Their use allows for a PSTH-like fit, but one that is cubic polynomial smooth. The $g(k-r)$ are indicator functions equal to 1 if the most recent spike is r bins prior to k and 0 otherwise. This functional form of λ_{hist} is standard (Truccolo et al., 2005; Czanner et al., 2008). The $\beta's$ and $\theta's$ are parameters to be fit via maximum likelihood.

Next, following the first procedure described in section 2, we used the fitted GLM to simulate 100 10 minute spike trains, rescaled both the experimental and simulated ISIs, and constructed both the KS and CDF difference plots. The results are shown in Figure 4, where the blue lines correspond to the comparison of the experimental CDF with the uniform CDF and the red lines to the comparison of the experimental CDF with the CDF estimated from the GLM, as described in section 2.3. For all three models, the differential KS plots reveal strong biases when the experimental rescaled ISIs are compared with the uniform distribution and a complete elimination of the bias when the distribution simulated from the GLM is used. Further, use of the GLM simulated distribution makes the difference between the differential KS plot lying within or outside the 95% confidence bounds. This was true even when spike history effects were included and KS plot biases much worse than in their absence. Finally we applied the analytical discrete time-rescaling theorem described in section 2.4 and plotted the results in green. The analytically corrected differential KS plot is nearly identical to the numerically simulated one. This indicates that the analytical correction, which is simpler to apply, is sufficient to test model goodness of fit.

3.2 Monkey V1 Receptive Field Data. Next we used spiking data recorded in V1 of two adult female rhesus monkeys (*Macaca mulatta*) during a fixation task. (See appendix B for details on the experimental procedure.) The visual stimuli consisted of a high-contrast light bar (50 cd/m^2; bar width, 0.2° or 5 pixels) moving with a constant velocity ($v = 14.9°/s$ or 425 pixels/s). The bar was presented in a square aperture of size (21.8° × 21.8° or 600 × 600 pixels centered over the receptive fields of the neurons being recorded. During stimulus presentation, the monkey was required to maintain fixation within a virtual window (window size, 1°) centered on the fixation point.

In this article, we show data from two monkeys. For each monkey, we selected two examples in which the recorded cells exhibited high average firing rates (first column of Figure 5). The data shown were recorded over nine trials, each of which lasted 2 seconds, during which the bar moved in a single direction. As with the simulated data, we used a GLM-based logistic regression form (logit link function) for the conditional intensity function,

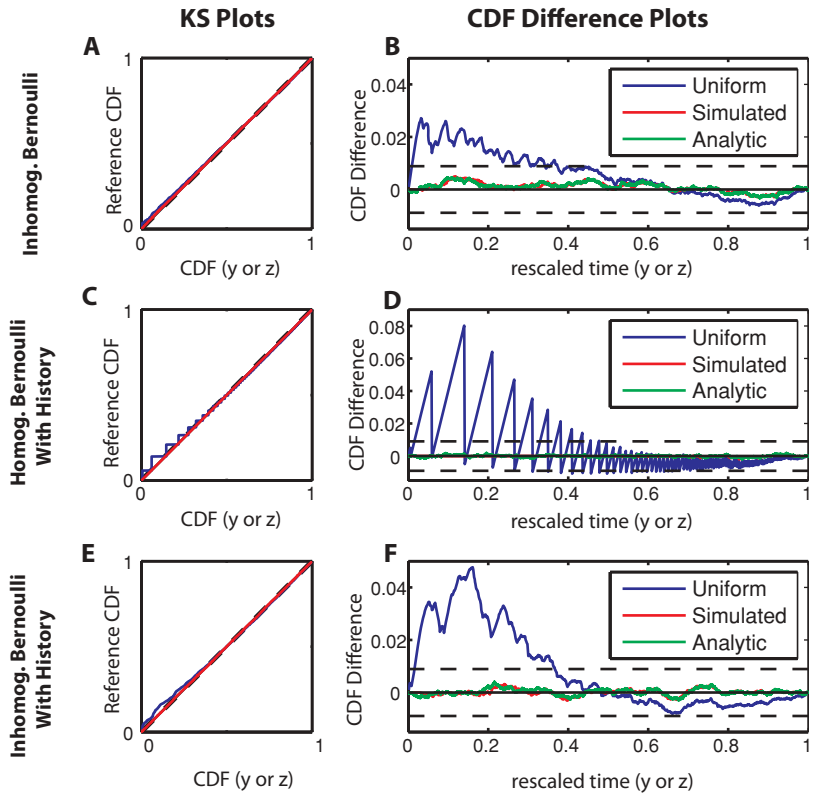

Figure 4: Comparison of standard KS test, KS test using simulated rescaled ISI distribution, and KS test using the analytically corrected rescaled time. Spike trains were simulated using the same three models as in Figure 2 at fine 10^{-10} msec temporal precision and then discretized at $\Delta = 1$ msec resolution. Logistic GLM models were fit and used to estimate the rescaled ISI distributions (See the text.) (A, C, E) KS plots for inhomogeneous Bernoulli, homogeneous Bernoulli with spike history, and inhomogeneous Bernoulli with spike history, respectively. (B, D, F) Differential KS plots for the same. Blue lines correspond to the standard KS test, which plots the CDF of the rescaled time z versus the CDF of the uniform distribution; red lines to the numerical simulation method, which plots the CDF of the rescaled time z versus the CDF of the numerically simulated reference distribution; and green lines to the analytical method, which plots the CDF of the analytically corrected rescaled time y versus the CDF of the uniform distribution. The red and green lines essentially overlap in the plots. For all three spike train models, strong KS and differential KS plot bias was eliminated when the numerically estimated distribution or the analytical correction was used.

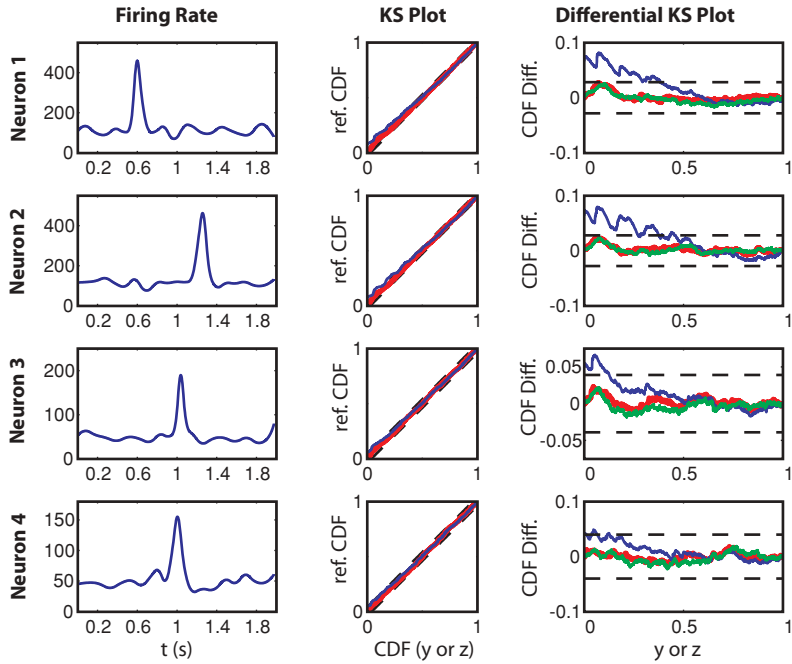

Figure 5: Four examples of neurons from two different monkeys (top two rows, monkey 1; bottom two rows, monkey 2) for which goodness of fit appears to be poor when the standard KS test is used but revealed to be good when either the numerically estimated reference distribution or the analytically corrected rescaled time y is used. Left column: firing rate. Middle column: KS plot. Right column: differential KS plots. Blue: standard KS test. Red: KS test with numerical simulation of reference distribution. Green: KS test with analytically corrected rescaled time y. As with the simulated spike trains of Figure 4, the KS and differential KS plot biases are eliminated when either the rescaled ISI distribution (z) is simulated using the fitted model or the analytically corrected rescaled time y is used.

with a temporal discretization of $\Delta = 1$ msec:

$$\log\left[\frac{\lambda(k)}{1-\lambda(k)}\right] = \underbrace{\sum_{j=1}^{J}\xi_j x_j(k)}_{} + \underbrace{\sum_{r=1}^{R}\theta_r g(k-r)}_{},$$

$$\stackrel{\text{def}}{=} \psi_{\text{stim}}(k) + \psi_{\text{hist}}(k) \tag{3.4}$$

where x_j represents the jth covariate that encodes the stimulus input (which may be in the form of either a feature of the visual stimulus, a PSTH profile,

or a specific basis function). The $g(k-r)$ is, as in the previous section, an indicator function representing whether the most recent spike was in the past rth temporal window, and θ_r represents the associated weight coefficient (a negative θ_r implies an inhibitory effect that might account for the refractory period of neuronal firing, and a positive θ_r implies an excitatory effect). The first term of the right-hand side of equation 2.1 is summarized by a stimulus-dependent response factor denoted by ψ_{stim}, and the last term represents a spiking history–dependent factor denoted by ψ_{hist}.

Similar to the simulated inhomogeneous Bernoulli process data in the previous section, we used semiparametric cubic B-spline functions (piecewise smooth polynomials) to model the stimulus-induced spiking activity ψ_{stim},

$$\psi_{\text{stim}}(k) = \sum_{j=1}^{J} \xi_j \mathcal{B}_j(k), \tag{3.5}$$

where J denotes the number of knots or control points. Note that the values of control points affect only the shape of ψ_{stim} locally due to the piecewise definition. For the data shown here, 12 control points are nonevenly placed on the 2 s time interval

As with our simulated data, we see in Figure 5 that when the standard KS test was used, the KS and differential KS plots lay outside the 95% confidence bounds. However, when temporal discretization was taken into account and our two techniques were used, the plots lay well within confidence bounds, and the GLM model was shown to be very well fit to the data. Thus, again, the simple analytical method is found to be sufficient to account for discretization-induced KS plot bias.

4 Discussion

It is vital to check a model's goodness of fit before making inferences from it. The time-rescaling theorem provides a powerful yet simple-to-implement statistical test applicable to a spike train, or other point process, for which the data are binary rather than continuously valued. The theorem states that the ISIs of a continuous time point process can be rescaled (through a variable transformation) so that they are independent and exponentially distributed. The rescaled ISIs can then be compared to the exponential distribution using a KS test or further rescaled to a uniform distribution and the KS test performed graphically (Brown et al., 2001). Each ISI is rescaled as a function of the time-varying spike probability over that particular ISI. Thus, time rescaling considers the probabilities of individual ISIs and provides a much stronger statistical test than, for example, tests based on the unscaled ISI distribution. Practical numerical considerations dictate that

the fitting of a statistical model usually requires the discretization of time into bins. For the purposes of the time-rescaling theorem, if the spike rate is low, ISIs long, and the probability per bin of a spike small, the distinction between discrete and continuous time will often not be important. In this article, we addressed the case where the spike rate is high, ISIs short, and the probability per bin of a spike large so that the distinction between discrete and continuous time matters.

When the probability per time bin of a spike is not sufficiently small, the standard, continuous time KS plot exhibits biases at both low and high rescaled ISIs. The source of these biases is twofold and originates in the consequences of discretizing a continuous time point process. First, the uncertainty as to where exactly in a bin a spike is located causes discrete time models to place a lower bound on the size of the smallest rescaled ISI z This leads to a positive KS plot bias at low z. Second, because discrete binary models allow only for a single spike per bin, they estimate per bin spike probabilities p_k that are less than $\int_0^{\Delta} \lambda(t)\,dt$ with the integral over bin k. We demonstrated both of these points theoretically using a homogeneous Poisson process, which we discretized into a homogeneous Bernoulli process, and also in our proof of the discrete time version of the theorem. These biases can be numerically relevant even at moderate spike rates and reasonable temporal discretizations. In this article, we considered mainly 40 Hz spiking at 1 msec discretization, ($p = 0.04$), but under some neurophysiological conditions, the spike rate can be much higher. For example, the awake monkey data presented in section 2 exhibited firing rates that at times exceeded 100 Hz.

Under such conditions, KS plots will exhibit biases at both low and high rescaled ISIs, which cannot be removed through more accurate numerical integration techniques or increased data sampling. In fact, sampling a longer spike train will make the issue more critical because the 95% confidence bounds on the KS plot scale as $1/\sqrt{N_{exp}}$, where N_{exp} is the number of experimentally recorded ISIs. In cases of long recording times, the confidence bounds can be quite tight, and it can be difficult to see variations in the fit using the standard KS plot even if those variations are statistically significant. We therefore introduced a new type of plot, the differential KS plot, in which we plot the difference between the CDFs of the empirical and simulated ISI distributions along with analytical 95% confidence bounds. This new type of plot displays the same information as the original KS plot but in a more visually accessible manner.

To handle KS plot bias, we proposed and implemented two procedures, both capable of testing the statistical sufficiency of any model that provides a measure of the discrete time conditional intensity function. The first procedure operates purely in discrete time and uses numerical simulation, in a manner similar in spirit to a bootstrap, to estimate the distribution of rescaled ISIs directly from a fitted statistical model. Model goodness of fit is tested by comparing the estimated and experimentally recorded rescaled

ISI distributions using a KS test. The confidence bounds on this two-sample KS test scale as $\sqrt{N_{exp}N_{sim}/(N_{exp}+N_{sim})}$. This procedure is therefore computationally tractable because a simulated spike train 20 times longer than the original experimentally recorded spike train will result in a KS test with confidence bounds only 1.02 times as wide as if the exact rescaled ISI distribution were known. For the second technique, we presented and proved a discrete time version of the time-rescaling theorem. This presumes an underlying continuous time point process that is sampled at finite resolution Δ, analytically corrects for the discretization, and defines a rescaled time ξ that is exponentially distributed at arbitrary temporal discretizations. We applied these two techniques to both simulated spike trains and spike trains recorded in awake monkey V1 cortex and demonstrated an elimination of KS plot bias when our techniques were used. The performance of both techniques was nearly identical, revealing high goodness of fit even when the fitted model failed the standard continuous time application of the KS test. Therefore, either method might be used, although the analytical method is perhaps preferable, if only because it is quicker to compute.

The discrete time-rescaling theorem is appropriate for point process type data such as spike trains, which are equally well described by either their spike times or their interspike intervals. It is, however, a test of model sufficiency, namely, whether a proposed statistical model is sufficient to describe the data. It does not, in and of itself, address issues of model complexity (overfitting) or whether the model form chosen is appropriate for describing the data in the first place.[10] Overfitting can be guarded against by splitting one's data into training and test data. After fitting the model parameters using the training data, the fitted model and the discrete time-rescaling theorem can be applied to the test data. Of course, we do not mean to imply that the discrete time-rescaling theorem is the only statistical test that should be employed for selecting and validating an appropriate model. Other statistical tests and criteria—for example, log likelihood ratio tests and the Akaike and Bayesian information criteria—should also be employed to rigorously judge goodness of fit and model complexity.

One might reasonably ask, Why not simply fit a statistical model with extremely fine temporal discretization so that the time-rescaling theorem applies in its standard form? There are several issues. First, spikes are not instantaneous events but are spread out in time on the order of 1 msec or slightly less. Second, experimenters often exclude apparent spikes that occur less than a msec (or thereabouts) apart in a recording, as it is difficult to distinguish spike wave forms that essentially lie on top of each other. For both of these reasons, defining spikes as instantaneous events is

[10] In this article, we used GLMs. Such models are widely applied to the analysis of spike train data. They also have an interpretation as a sort of integrate-and-fire neuron (see, e.g., Paninski, 2004b). However, nothing in the discrete time-rescaling theorem precludes its use for testing the fit of a statistical model of the spiking probability that is not a GLM.

physically problematic. Although the continuous time point process framework is theoretically appealing, there is usually no reason not to consider the data in discrete time, fit a discrete time model, and perform a discrete time goodness-of-fit test. Finally, there is the important issue of computation time and computer memory. When recording times are long and the number of spikes large, confidence bounds on the KS test will be very tight. Extremely fine temporal discretization will then be required for the biases to be less than the width of the confidence bounds. The amount of memory and computation time required under these conditions can rapidly become prohibitive. Further, since using the discrete time-rescaling theorem is almost as quick and simple a procedure as the standard KS test, we can see no reason not to use it. In closing, a failure of the standard KS test does not immediately imply poor model fit. Biases induced by temporal discretization may be a factor and should be considered before rejecting the model.

Appendix A: Independence of Rescaled Times

We prove that the rescaled times ξ_i (and in the continuous time limit τ_i) are not only exponentially distributed but also independent. To establish this result, it suffices to show that the joint CDF of the ξ_i's can be written as the product of the individual CDFs and that these CDFs are those of independent exponential random variables with rate 1. The CDF of an exponential random variable with rate 1 is

$$F(\xi) = 1 - e^{-\xi}. \tag{A.1}$$

We recall that the rescaled times ξ_i are defined as

$$\xi_i = \sum_{k=k_{i-1}+1}^{k_i-1} q_k + \frac{q_{k_i}}{\Delta}\delta_i, \tag{A.2}$$

where $\delta_i \in [0, \Delta]$ is a random variable determined by first drawing a uniform random variable $r_i \in [0, 1]$ and then calculating

$$\delta_i = -\frac{\Delta}{q_{k_i}} \log[1 - r_i(1 - e^{-q_{k_i}})]. \tag{A.3}$$

This definition of the rescaled times ξ_i implicitly defines a spike time $t_i = (k_i - 1)\Delta + \delta_i$. Because the transformation from the spike times t_i (or the spike bins k_i) to the ξ_i's is one-to-one, we have that the following two events are equivalent:

$$\{\Xi_1 < \xi_1, \Xi_2 < \xi_2, \ldots, \Xi_N < \xi_N\} = \{T_1 < t_1, T_2 < t_2, \ldots, T_N < t_n\}. \tag{A.4}$$

Therefore, the joint CDF of the ξ_i's is

$$\begin{aligned} F(\xi_1, \xi_2, \ldots, \xi_N) &= P\{\Xi_1 < \xi_1, \Xi_2 < \xi_2, \ldots, \Xi_N < \xi_N\} \\ &= P\{T_1 < t_1, T_2 < t_2, \ldots, T_N < t_n\} \\ &= F(t_1, t_2, \ldots, t_N) \\ &= \prod_{i=2}^{N} F(t_i \mid t_1, \ldots, t_{i-1}) F(t_1 \mid 0). \end{aligned} \tag{A.5}$$

The last line follows from the multiplication rule of probability (Miller, 2006).

The conditional CDFs $F(t_i \mid t_1, \ldots, t_{i-1})$ can be calculated by noting that the probability of any given ISI is equal to 1 minus the probability that there was at least one spike within the epoch defined by the ISI. Formally, this can be written as

$$\begin{aligned} &P(\text{no spike in } (k_{i-1}\Delta, (k_i - 1)\Delta + \delta_i)) \\ &\quad = 1 - P(\text{at least one spike in } (k_{i-1}\Delta, (k_i - 1)\Delta + \delta_i)) \\ &\quad = 1 - \int_{k_{i-1}\Delta}^{t_i} P(t_i' \mid k_1, k_2, \ldots, k_{i-1})\, dt_i' \\ &\quad = 1 - F(t_i \mid k_1, k_2, \ldots, k_{i-1}). \end{aligned} \tag{A.6}$$

The right-hand side has the CDF we wish to calculate. It remains to determine the left-hand side. But this is simply the ISI probability of equation 2.29:

$$P(t = (k_i - 1)\Delta + \delta_i \mid k_1, k_2, \ldots, k_{i-1}) d\delta_i = P(\xi_i) d\xi_i = e^{-\xi_i} d\xi_i. \tag{A.7}$$

Thus,

$$F(t_i \mid k_1, k_2, \ldots, k_{i-1}) = F(\xi_i) = 1 - e^{-\xi_i}. \tag{A.8}$$

Inserting this result into equation A.5, we get

$$\begin{aligned} F(\xi_1, \xi_2, \ldots \xi_N) &= \prod_{i=1}^{N} 1 - e^{\xi_i} \\ &= \prod_{i-1}^{N} F(\xi_i), \end{aligned} \tag{A.9}$$

which establishes the proof. A similar argument can be made for the continuous time-rescaled time τ_i. An intuitive way to understand the independence of the rescaled ISIs is that these were calculated using either $\lambda(t \mid H_t)$ or $p(k \mid H_t)$, which are conditioned on the previous spiking history. This preconditioning before calculation of the rescaled times enforces the independence of ξ or τ, or both. This independence of the rescaled times is useful because testing for independence provides an additional statistical significance test beyond the testing for exponentiality by a KS or CDF difference plot. The test is identical to the continuous time case (we refer readers to Czanner et al., 2008, for a discussion).

Appendix B: Experimental Procedures

Experimental procedures were approved by the National Committee on Animal Welfare (Regierungspraesidium Hessen, Darmstadt) in compliance with the guidelines of the European Community for the care and use of laboratory animals (European Union directive 86/609/EEC). Neuronal spiking activities were recorded in awake and head-fixed monkeys in opercular region of V1 (RFs centers, 2–5° of eccentricity) and, on some occasions, from the superior bank of the calcarine sulcus (8–12° of eccentricity).

Quartz-insulated tungsten-platinum electrodes (diameter 80 μm, 0.3–1.0 MΩ impedance; Thomas Recording) were used to record the extracellular activities from three to five sites in both superficial and deep layers of the striate cortex (digitally bandpass filtered, 0.7–6.0 kHz; Plexon Inc.). Spikes were detected by amplitude thresholding, which was set interactively based on online visualization of the spike waveforms (typically, 2–3 SD above the noise level). Trials with artifacts were rejected during which the monkey did not maintain fixation or showed no response or incorrect behavior.

Acknowledgments

We thank Sergio Neuenschwander and Bruss Lima for the generous use of their data. This work was supported by NIH grants K25 NS052422-02, DP1 OD003646-01, MH59733-07, the Hertie Foundation, the Max Planck Society, and EU grant FP6-2005-NEST-Path-043309.

References

Brown, E. N., Barbieri, R., Ventura, V., Kass, R. E., & Frank, K. V. (2001). The time-rescaling theorem and its application to neural spike train data analysis. *Neural Computation*, *14*, 325–346.

Czanner, G., Eden, U. T., Wirth, S., Yanike, M., Suzuki, W. A., & Brown, W. A. (2008). Analysis of between-trial and within-trial neural spiking dynamics. *Journal of Neurophysiology*, *99*, 2672–2693.

De Valois, R. L., Yund, E. W., & Hepler, N. (1982). The orientation and direction selectivity of cells in macaque visual cortex. *Vision Research*, *22*, 531–544.

Frank, L. M., Eden, U. T., Solo, V., Wilson, M. A., & Brown, E. N. (2002). Contrasting patterns of receptive field plasticity in the hippocampus and the entorhinal cortex: An adaptive filtering approach. *Journal of Neuroscience*, *22*, 3817–3830.

Johnson, A., & Kotz, S. (1970). *Distributions in statistics: Continuous univariate distributions* (2nd ed.). New York: Wiley.

Kass, R. E., & Ventura, V. (2001). A spike train probability model. *Neural Computation*, *13*, 1713–1720.

MacEvoy, S. P., Hanks, T. D., & Paradiso, M. A. (2007). Macaque V1 activity during natural vision: Effects of natural scenes and saccades. *Journal of Neurophysiology*, *99*, 460–472.

Massey, F. J. (1951). The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American Statistical Association*, *46*, 68–77.

McCullagh, P., & Nelder, J. A. (1989). *Generalized linear models* (2nd ed.). New York: Chapman and Hall.

Miller, G. K. (2006). *Probability*. Hoboken, NJ: Wiley.

Paninski, L. (2004a). Maximum likelihood estimation of cascade point process neural encoding models. *Network*, *4*, 243–262.

Paninski, L. (2004b). Maximum likelihood estimation of a stochastic integrate-and-fire neural encoding model. *Neural Computation*, *16*, 2533–2561.

Pawitan, Y. (2001). *In all likelihood: Statistical modeling and inference using likelihood*. New York: Oxford University Press.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (2007). *Numerical recipes* (3rd ed.). Cambridge: Cambridge University Press.

Song, D., Chan, R. H., Marmarelis, V. Z., Hampson, R. E., Deadwyler, S. A., & Berger, T. W. (2006). Physiologically plausible stochastic non-linear kernel models of spike tain to spike train transformation. In *Conf. Proc. IEEE Eng. Med. Biol. Soc.* (Vol. 1, pp. 6129–6132). Piscataway, NJ: IEEE.

Snyder, D. (1975). *Random point processes*. Hoboken, NJ: Wiley.

Truccolo, W., Eden, U. T., Fellows, M. R., Donoghue, J. P., & Brown, E. N. (2005). A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *Journal of Neurophysiology*, *93*, 1074–1089.

Wasserman, L. (2004). *All of statistics*. Berlin: Springer-Verlag.

Wasserman, L. (2007). *All of nonparametric statistics*. Berlin: Springer-Verlag.

Received May 7, 2009; accepted February 20, 2010.