Article

Synchrony-Division Neural Multiplexing: An Encoding Model

Mohammad R. Rezaei [1,2,3], Reza Saadati Fard [4], Milos R. Popovic [2,3], Steven A. Prescott [2,5,6] and Milad Lankarany [1,2,3,5,*]

1 Krembil Research Institute, University Health Network (UHN), Toronto, ON M5T 0S8, Canada
2 Institute of Biomedical Engineering, University of Toronto, Toronto, ON M5S 3G9, Canada
3 KITE Research Institute, Toronto Rehabilitation Institute, University Health Network (UHN), Toronto, ON M5G 2A2, Canada
4 Department of Computer Science, Worcester Polytechnic Institute, Worcester, MA 01609, USA
5 Department of Physiology, University of Toronto, Toronto, ON M5S 1A8, Canada
6 Neurosciences and Mental Health, The Hospital for Sick Children, Toronto, ON M5G 1X8, Canada
* Correspondence: milad.lankarany@uhnresearch.ca

Abstract: Cortical neurons receive mixed information from the collective spiking activities of primary sensory neurons in response to a sensory stimulus. A recent study demonstrated an abrupt increase or decrease in stimulus intensity and the stimulus intensity itself can be respectively represented by the synchronous and asynchronous spikes of S1 neurons in rats. This evidence capitalized on the ability of an ensemble of homogeneous neurons to multiplex, a coding strategy that was referred to as synchrony-division multiplexing (SDM). Although neural multiplexing can be conceived by distinct functions of individual neurons in a heterogeneous neural ensemble, the extent to which nearly identical neurons in a homogeneous neural ensemble encode multiple features of a mixed stimulus remains unknown. Here, we present a computational framework to provide a system-level understanding on how an ensemble of homogeneous neurons enable SDM. First, we simulate SDM with an ensemble of homogeneous conductance-based model neurons receiving a mixed stimulus comprising slow and fast features. Using feature-estimation techniques, we show that both features of the stimulus can be inferred from the generated spikes. Second, we utilize linear nonlinear (LNL) cascade models and calculate temporal filters and static nonlinearities of differentially synchronized spikes. We demonstrate that these filters and nonlinearities are distinct for synchronous and asynchronous spikes. Finally, we develop an augmented LNL cascade model as an encoding model for the SDM by combining individual LNLs calculated for each type of spike. The augmented LNL model reveals that a homogeneous neural ensemble model can perform two different functions, namely, temporal- and rate-coding, simultaneously.

Keywords: neural coding; information representation; multiplexed coding; synchronous and asynchronous spikes; general linear model

1. Introduction

Transmitting multiple signals over a single communication channel increases the channel bandwidth and enhances the coding efficiency [1,2]. Similar to digital communication systems, the brain utilizes different forms of multiplexing—in different brain regions and in regard to different stimuli—to represent multiple features of a stimulus with a neural code [2]. For example, in the auditory sensory system, the frequency and intensity of a periodic stimulus are encoded by the phase-locked spikes and the probability of spiking per stimulus cycle, respectively [3]. Similarly, the frequency and intensity of vibrotactile stimuli are represented by the timing and rate of spikes in the somatosensory cortex [4]. Recently, differentially synchronized spiking in neurons of the primary somatosensory cortex was shown to enable multiplexed coding of low- and high-contrast features of tactile stimuli [5].

Despite various forms of neural multiplexing, a thorough understanding of how the brain achieves multiplexing remains undiscovered. Specifically, the functional characteristics—in the sense of linear or nonlinear filtering properties—of a neural ensemble that multiplexes different features of a stimulus are yet to be uncovered.

Different features of stimuli, such as the intensity, frequency, onset and offset, etc., dictate which multiplexing strategies are the most appropriate [4]. In addition to the properties of stimuli, heterogeneity of the neurons in a population code enables different neurons to encode different stimulus features. The functional properties of a heterogeneous neural ensemble, which includes neurons with different functions, e.g., integrators vs. coincidence detectors, might be fully described by the dynamics of individual neurons. For example, an ensemble of heterogeneous cochlear nuclei in the auditory cortex is composed of two anatomically distinct sub-nuclei, namely, the magnocellular and the angular nucleus, each of which selectively encodes a specific feature of the stimulus. The magnocellular nucleus selectively encodes the stimulus frequency with a temporal code by implementing a high-pass filter, whereas the angular nucleus selectively encodes the stimulus intensity with a rate code by implementing a low-pass filter [6]. In contrast to heterogeneous neural ensembles, the functional characteristics of an ensemble of homogenous neurons, which includes neurons with nearly identical functions, cannot be identified based on the properties of individual neurons solely [4,5]. For example, in synchrony-division multiplexing (SDM) [5], information about slow and fast stimulus features were respectively represented by asynchronous and synchronous spikes of the same neurons. Thus, this form of multiplexing suggests that both slow and fast features of the stimulus can be encoded by homogeneous (identical) neurons that operate in a hybrid mode [5], i.e., neither low-pass nor high-pass filtering of the stimulus [7,8]. Thus, a challenging question is whether multiplexing (such as SDM) in a homogeneous neural ensemble reveals system-level functions beyond those performed by individual neurons [5].

In this paper, we utilize conductance-based and linear nonlinear (LNL) cascade models to establish a theoretical framework to address this question [9–12]. First, we use conductance-based models and construct a homogeneous neural ensemble that multiplexes the slow and fast features of a common stimulus using asynchronous and synchronous spikes, respectively. Using LNL cascade models, we explore whether different linear filters and static nonlinearities are associated with different types of spikes. We show that a low-pass filter followed by a nonlinearity with a mild slope generates asynchronous spikes whereas a high-pass filter followed by a nonlinearity with a steep slope detects fast features of the stimulus by generating synchronous spikes. Then, we develop an augmented LNL model for SDM by integrating the LNL models underlying each type of spike.

2. Results

In the present paper, we developed an augmented LNL cascade model as an encoding model for the SDM [5]. Conductance-based neuron models were used to create an ensemble of homogeneous neurons whose input (mixed stimulus)–output (spikes) relationship was estimated with the augmented LNL model.

As shown in Figure 1A, we construct an ensemble of homogenous neurons with 30 Morris–Lecar (ML) neuron models (see Methods), all of which receive a common mixed signal comprising slow and fast features as well as independent, physiologically realistic conductance noise [5,13]. We divided the simulated data (20 s of data) into two training (the first 50% of data) and test (the remaining 50%) sets. The parameters of the ML model were selected in a way that all neurons operate in a hybrid mode [14]. Spikes generated by an ensemble of ML neurons were used to fit the augmented LNL model in which two separate LNL models were combined to represent the rate and temporal codes simultaneously. This study shows that an ensemble of homogeneous neurons utilizes different strategies to generate synchronous and asynchronous spikes, which enable the simultaneous coding of fast and slow features of a mixed stimulus, respectively. Although the biophysical mechanisms underlying implementation of SDM with an ensemble of

homogeneous neurons is still unknown, the two-stream augmented LNL model provides a system-level understanding of SDM function.
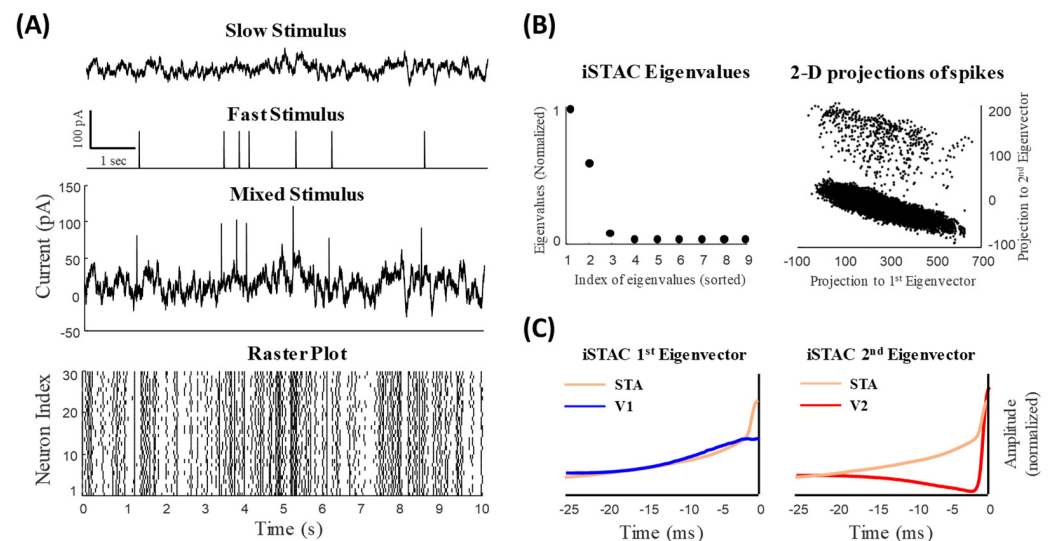

Figure 1. Slow and fast features of a mixed signal can be inferred from responses of a homogeneous ensemble of neurons using the iSTAC method. (**A**) Slow and fast signals comprising a mixed signal. (Bottom) Sample raster plot of 30 model neurons receiving the common mixed signal (and independent noise). Spikes evoked by the fast and slow signals cannot be distinguished visually. (**B**) The iSTAC method was applied to spike-triggered mixed signal and eigenvalues and eigenvectors were obtained (see Methods). (**Left**) The eigenvalues of the iSTAC matrix reveal two significant components of the population code. (**Right**) The projection of spike-triggered mixed signal onto the main eigenvectors of the iSTAC matrix. Two clusters can be visually distinguished. (**C**) The 1st and 2nd eigenvectors of the iSTAC matrix, $V1$ and $V2$, respectively, are shown against the spike-triggered average (STA). $V1$ resembles the STA filter reflecting slowly-varying changes in the signal. Unlike $V1$, $V2$ resembles a high-pass filter (differentiator) that reflects fast features of the mixed signal.

2.1. Different Temporal Filters Map Distinct Features of a Mixed Stimulus

To explore how the slow and fast features of the stimulus are encoded by spikes of an ensemble of neurons, we used well-known feature space estimators such as the spike-triggered average (STA) [15,16] or information-theoretic spike-triggered average and covariance (iSTAC) to reveal the temporal characteristics of neurons in response to a stimulus [17].

The STA filter is a precise and unbiased predictor for a neural population given a stationary and single-dimension stimulus [16]. However, it fails to provide precise predictions when the dimensionality of the stimulus is larger than one. For example, in retinal ganglion cells the STA cannot predict the neural response of both ON and OFF cells given a mixed input comprising more than one feature. To explore other possible subspace features of the neural response, we used the iSTAC method and calculated the optimal subspace features. The iSTAC quantifies the significance of subspaces based on the mutual information between the stimulus and neural response [17]. In this method, we choose the eigenvectors of the spike-triggered stimulus ensemble matrix more precisely by minimizing the Kullback–Leibler (KL) divergence between the eigenvectors of the ensemble matrix and the raw stimulus distributions (see Methods for more details). In fact, the iSTAC maximizes information based on the first two moments of the spike-triggered stimulus ensemble and provides a unifying information-theoretic framework that captures the ensemble neuron activity in different subspaces. This provides an implicit model of the contribution of the nonlinear function mapping the feature space to the neural response. As shown in Figure 1B (left), the iSTAC matrix calculated for the mixed stimulus has two significant eigenvalues whose underlying eigenvectors reveal two distinct temporal filters, namely,

ν_1 and ν_2. The projection of the spike-triggered stimulus ensemble on ν_1 and ν_2, shown in Figure 1B (right), reveals two distinct clusters.

In synchrony-division multiplexing, a mixed-input signal containing slow and fast features drives an ensemble of neurons. The fast component of the stimulus whose neural representation is synchronous and sparse does not appear in the STA, as STA averages out the sparsely-occurring fast features of the stimulus [5]. However, unlike the fast signal, the neural representation of the slow signal is asynchronous and dense; thus, the STA filter mainly contains information of the slow components of the mixed signal [5]. Unlike the STA filter, the most informative subspaces selected by the iSTAC method behave as multi-space feature estimators and illustrate the slow and fast features of the mixed stimulus.

Figure 1C shows that the STA filter calculated for the mixed stimulus mainly captures the slow feature of the signal but cannot truly capture the dynamics of synchronous spikes. Unlike the STA filter, ν_1 and ν_2 of the iSTAC method illustrate the slow and fast features of the mixed stimulus, respectively. As can be observed in Figure 1C, ν_1 is similar to the STA filter and represents the slow component of the stimulus, and ν_2 describes the fast features of the stimulus (note that the STA filter was duplicated in Figure 1C (left and right) and compared with both ν_1 and ν_2).

2.2. Low-Dimensional Feature Space of the Neural Response Can Be Characterized by the STAs of Synchronous and Asynchronous Spikes

Recently, it has been shown that synchronous and asynchronous spikes encode information about the fast and slow features of a mixed stimulus (equivalent to that used in the present study), respectively [13,18,19]. Using an information-theoretic approach, it was shown that synchronous and asynchronous spikes carry information in different time scales. By classifying the spikes of a population of neurons into synchronous and asynchronous spikes, it was demonstrated that the STA filters underlying these spikes, namely, μ_{Async} and μ_{Sync}, reflect the fast and slow features of the stimulus, respectively. Figure 2A shows the classified synchronous (red) and asynchronous (blue) spikes in the raster plot.
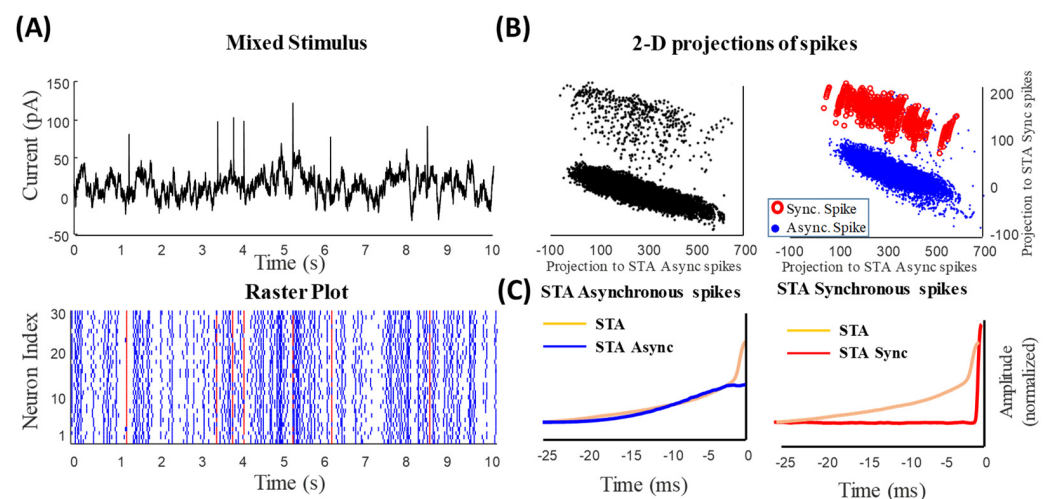

Figure 2. Synchronous and asynchronous spikes represent slow and fast features of the mixed signal, respectively. (**A**) Synchronous (red) and asynchronous (blue) spikes are distinguished by setting a threshold on the instantaneous firing rate calculated by a narrow kernel (see Methods). Synchronous spikes evoked by the fast signals can be distinguished visually. (**B**) The projection of spike-triggered mixed signal onto the STA$_{Sync}$ and STA$_{Async}$. Two (visually) distinguishable clusters belong to asynchronous spikes representing the slow feature of the signal (blue dots) and synchronous spikes representing the fast features (red circles). (**C**) The spike-triggered average of synchronous (red) and asynchronous (blue) spikes, namely, STA$_{Sync}$ and STA$_{Async}$, respectively, is shown against the STA of all spikes (similar to Figure 1C).

Here, we compared these filters with those obtained using the iSTAC method. First, we tested if the projection of the spike-triggered stimulus ensemble on μ_{Async} and μ_{Sync} creates two distinct clusters similar to that projected on ν_1 and ν_2. As shown in Figure 2B (left), two distinct and separable clusters were generated by μ_{Async} and μ_{Sync}. More importantly, one can distinguish between these clusters by projecting the synchronous- and asynchronous-spike-triggered stimulus ensemble on μ_{Async} and μ_{Sync}. Figure 2B (right) reveals that these stimulus ensembles are separable and mutually exclusive. Figure 2C shows the temporal patterns of μ_{Async} and μ_{Sync} versus the STA filter. As expected, μ_{Async} resembles the STA filter, indicating the slow features of the stimulus, and μ_{Sync} (similar to ν_2) describes abrupt changes in the stimulus.

Furthermore, to investigate the functional roles of the above filters, we tested how they contribute to signal reconstruction. The reconstructed signal was obtained by the convolution of spikes—either all spikes for STA (Figure 3A) or ν_1 and ν_2 (Figure 3B) or asynchronous and synchronous spikes for μ_{Async} and μ_{Sync}, respectively (Figure 3C). Figure 3 illustrates a 10 s sample of the reconstructed signal using these methods. As is clear in Figure 3B, the signal reconstructed with ν_1 and ν_2 (iSTAC method) resembles that generated with μ_{Async} and μ_{Sync}, and both of these signals better capture the fast features than that obtained by the STA filter, indicating the functional relevance between these filters.
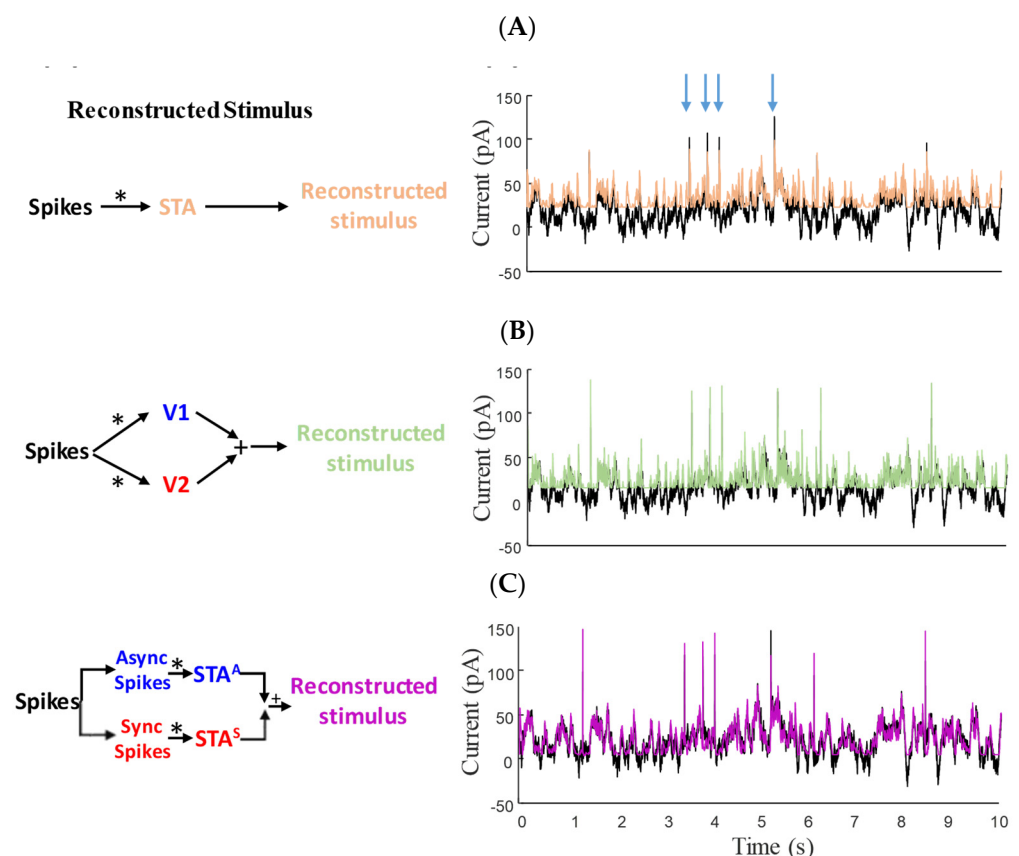

Figure 3. Block diagram of decoding the mixed signal from spikes using (**A**) the STA filter (light brown), (**B**) a weighted sum of the 1st and 2nd eigenvectors of the iSTAC method (green), and (**C**) a weighted sum of filtered asynchronous spikes (by STA$_{Async}$) and filtered synchronous spikes (by STA$_{Sync}$) (purple). Original mixed signal (black) is overlaid with reconstructed signal (color) in the plots. As can be seen in these figures, the reconstructed signal based on STA$_{Sync}$ and STA$_{Async}$—similar to that obtained by eigenvectors of iSTAC method—can capture both slow and fast components of the signal accurately. "*" indicates the spiking signals.

2.3. Different Nonlinear Functions Are Associated with Synchronous and Asynchronous Spikes

Given different temporal filters underlying synchronous and asynchronous spikes, we sought how these filters map the fast and slow features of the mixed stimulus to the firing rate of an ensemble of conductance-based model neurons. Moreover, since the dynamics of a neural ensemble is not fully linear, these linear filters are not sufficient to project the stimulus to spikes. We utilized a well-known phenomenological model, namely, the LNL cascade model, which uses a linear stimulus filter followed by a static nonlinear transformation, to estimate the firing rate of an ensemble of neurons. Figures 4 and 5 (panel A for both figures) show the LNL diagram for asynchronous and synchronous spikes, respectively. We tested if the linear filter and static nonlinearity are different for synchronous and asynchronous spikes given a common mixed signal. We obtained static nonlinearity functions for synchronous and asynchronous spikes by applying μ_{Sync} and μ_{Async} filters to the mixed stimulus (s) and mapping their outputs (through the nonlinearity) to the peri-stimulus time histogram (PSTHs) of the synchronous and asynchronous spikes, respectively:

$$PSTH_{Async} = f_{Async}\left(\mu_{Async} * s\right) \tag{1a}$$

$$PSTH_{Sync} = f_{Sync}\left(\mu_{Sync} * s\right) \tag{1b}$$

where $f_{Async}(x)$ and $f_{Sync}(x)$ are the nonlinearities associated with the asynchronous and synchronous spikes, respectively.

Figures 4B and 5B show, respectively, the raw nonlinearities for asynchronous and synchronous spikes that correspond to the mapping of every single point of the output of the linear filters (x-axis) to the values of the PSTHs (y-axis). For the nonlinearities underlying the asynchronous and synchronous spikes, we fitted ReLU nonlinearity and sigmoid functions, respectively [20]. The nonlinearity associated with the asynchronous spikes, $f_{Async}(x)$, has a shallow slope and broad dynamic range, enabling rate-modulated coding. In contrast, the nonlinearity underlying the synchronous spikes, $f_{Sync}(x)$), has a steep slope and narrow dynamic range, enabling detection of events (i.e., abrupt changes). Although more sophisticated nonlinear functions could provide better fits, we chose simple and well-established nonlinear functions to highlight the difference in the shapes of the nonlinearities underlying rate versus temporal codes in the context of SDM. The instantaneous firing rates of each type of spike can be constructed by passing the output of the temporal filter through the fitted nonlinearities. These firing rates were estimated and drawn against the PSTHs of the asynchronous and synchronous spikes for the test data in Figure 4C,D and Figure 5C,D, respectively. As shown in these figures, the nonlinear functions and estimated PSTHs underlying the temporal filters obtained using the iSTAC (*V1* and *V2*) and classified spikes (μ_{Async} and μ_{Sync}) are nearly identical.
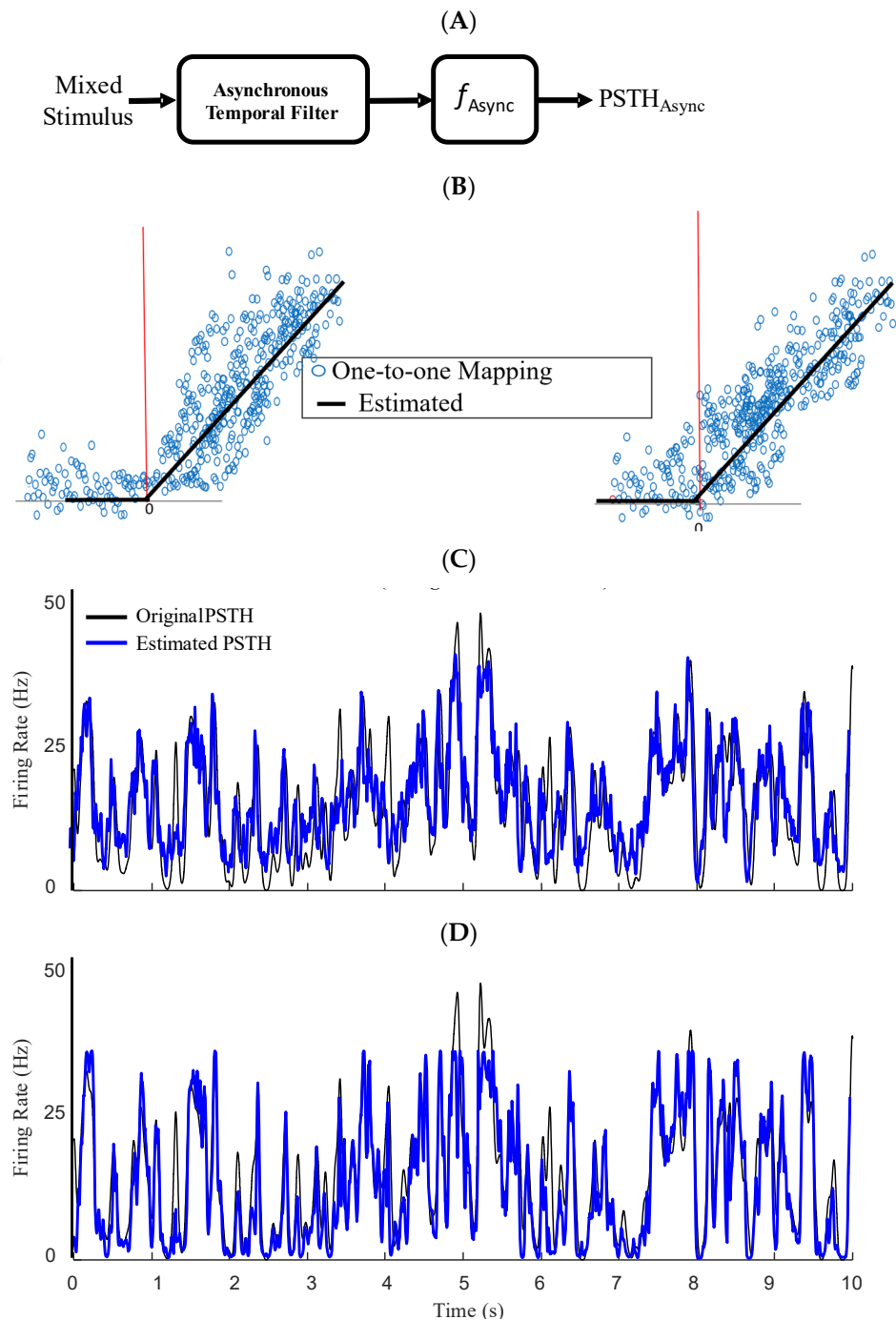
Figure 4. Static nonlinearities underlying asynchronous spikes. (**A**) Block diagram of LNL model for asynchronous spikes. (**B**) Static nonlinearity calculated for asynchronous spikes is obtained by mapping the output of filtered stimulus to the instantaneous rate of asynchronous spikes (calculated with a wide kernel, σ = 25 msec). Static nonlinearity calculated based on 1st eigenvectors of the iSTAC method, v_1, (**Left**) and STA_{Async} (**Right**). The solid black shows fitted rectifiers. (**C**) The PSTHs constructed using the fitted nonlinearities based on v_1 were drawn against the PSTH of asynchronous spikes. (**D**) The PSTHs constructed using the fitted nonlinearities based on STA_{Async} were drawn against the PSTH of asynchronous spikes.
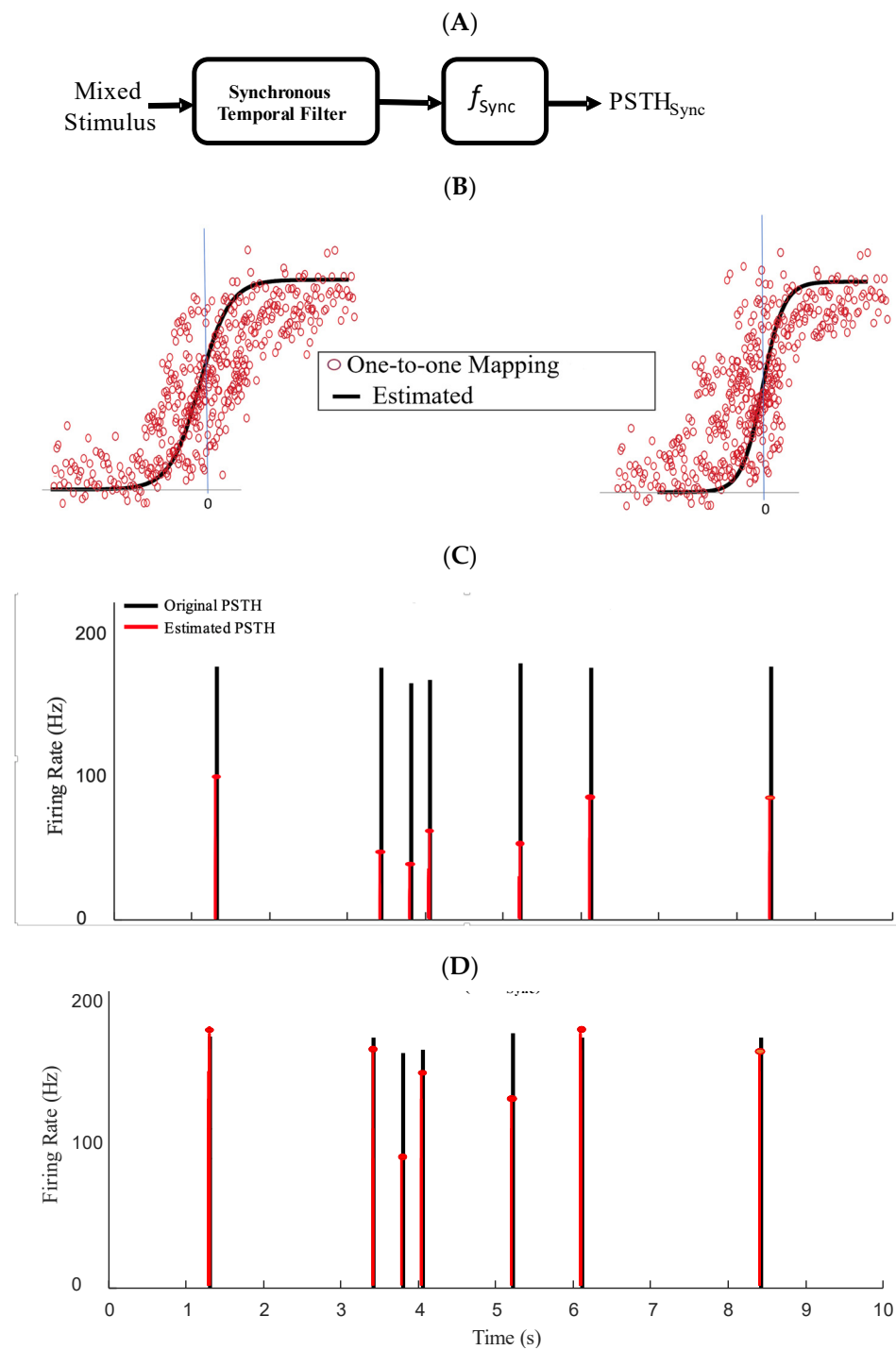
Figure 5. Static nonlinearities underlying synchronous spikes. (**A**) Block diagram of LNL model for synchronous spikes. (**B**) Static nonlinearity calculated for the synchronous spikes is obtained by mapping the output of filtered stimulus to the instantaneous synchronous events (calculated with a narrow kernel, σ = 1 msec). Static nonlinearity calculated based on 2nd eigenvectors of the iSTAC method, v_2, (**Left**) and STA_{Sync} (**Right**). The solid black shows fitted sigmoid functions. (**C**) The PSTHs constructed using the fitted nonlinearities based on v_2 were drawn against the PSTH of synchronous spikes. (**D**) The PSTHs constructed using the fitted nonlinearities based on STA_{Sync} were drawn against the PSTH of synchronous spikes.

2.4. An Augmented LNL Cascade Model for Synchrony-Division Multiplexing

The LNL cascade models were utilized to encode specific features of a mixed stimulus with synchronous or asynchronous spikes. As shown in the previous sections, temporal filters and nonlinear transformations of either type of spike was distinct and estimated using separate LNL cascade models. Here, we sought whether a combination of these cascade models, i.e., an augmented LNL model, could accurately encode different features of a mixed stimulus through different types of spikes. We developed a two-stream LNL cascade model that combines the PSTHs of the synchronous and asynchronous spikes to reconstruct the mixed PSTH of both types of spikes, as:

$$PSTH_{total} = \sum_{i \in \{Sync, Async\}} \omega_i \times G_i * f_i(\mu_i * s) \tag{2}$$

where $\omega_i s$ are the combination weights for each stream of reconstructed PSTHs. To reduce the model complexity and promote smoothness in the output, we applied parameterized Gaussian kernels, $G_{Async} = Gaussian\left(0, \sigma_{Async}\right)$ and $G_{Sync} = Gaussian\left(0, \sigma_{Sync}\right)$, to the reconstructed PSTHs in each stream [21,22]. Figures 4B and 5B show, respectively, the raw nonlinearities for the asynchronous (f_Async) and synchronous (f_Sync) spikes that correspond to the mapping. The augmented LNL model simultaneously encodes the slow and fast features of the stimulus using asynchronous and synchronous spikes, respectively. Figure 6A shows the block diagram of the augmented LNL model. This model implies that the low-pass filter and shallow non-linearity underlying the asynchronous spikes are required to produce the rate code. In contrast, the high-pass filter and sigmoid nonlinearity for synchronous spikes are necessary to preserve the reliable spike times underlying fast features of the stimulus. Taken together, the augmented LNL model allows rate and temporal to coexist and represent distinct features of the mixed signal. the coexistence of the rate and temporal codes happen to encode multiple features of a mixed stimulus. To capitalize on the significance of temporal filters and nonlinearity transformations of each type of spike in estimating the total firing rate of a neural ensemble, we compared the performance of the augmented LNL model with that of a conventional one-stream LNL. Figure 6B–D shows the firing rate estimated by the three methods, namely, Poisson GLM and augmented LNL models (Figure 6C,D) (see Section 2.3), against the PSTH of ensemble of neurons for the test data. We also quantitatively measured the performance of the augmented LNL and Poisson models in Table 1. As can be observed, the firing rate estimated using the augmented LNL model can better capture both the rate of asynchronous spikes and the timing of synchronous events compared to those estimated using the one-stream Poisson GLM.

Table 1. Mean absolute error (MAE) and root mean squared error (RMSE) performance measure comparison between Poisson GLM and augmented LNL model on test data.

	(MAE)			RMSE		
	Sync	Async	Mixed	Sync	Async	Mixed
Poisson GLM	0.245	0.223	0.228	0.338	0.313	0.326
Augmented LNL (STA_{V_1}, STA_{V_2})	0.101	0.104	0.102	0.137	0.134	0.135
Augmented LNL (STA_{ASync}, STA_{Sync})	0.103	0.107	0.106	0.140	0.151	0.149

(A)
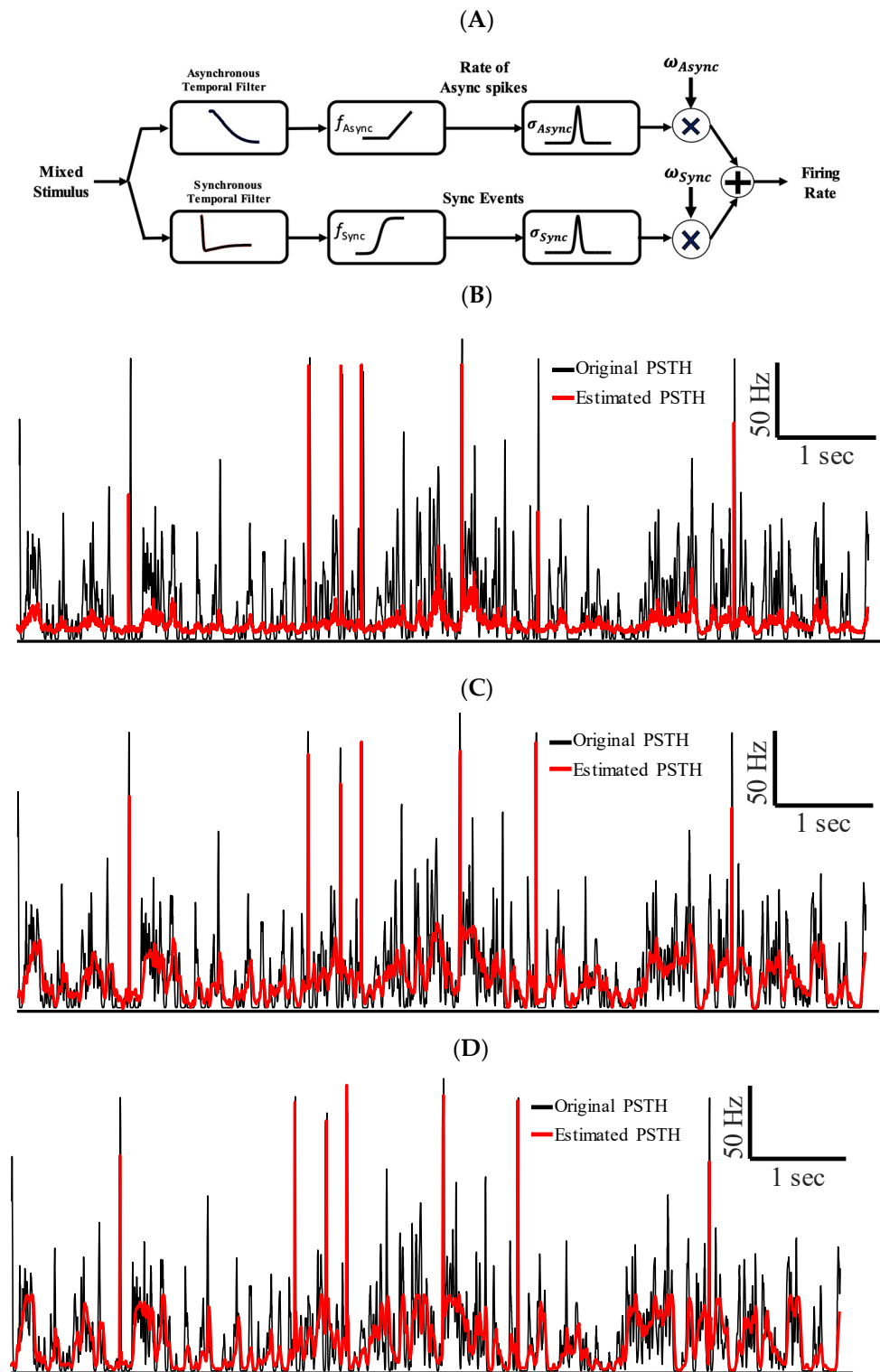

(B)

(C)

(D)

Figure 6. Two-stream LNL model, referred to as augmented LNL model, enables co-existence of temporal and rate codes. (**A**) Block diagram of the augmented LNL model for combining rate of asynchronous and synchronous spike events. (**B**) The PSTHs estimated using a conventional Poisson GLM (red) are shown against the original PSTH (calculated with a 1 msec Gaussian kernel). (**C**) The PSTHs estimated using the segmented LNL using temporal filters of iSTAC method. (**D**) The PSTHs estimated using LNL using the segmented LNL using STA_{Async} and STA_{Sync}.

3. Discussion

The ability of an ensemble of homogeneous cortical neurons to multiplex multiple features of a mixed stimulus was studied in [5]. The specific mechanism by which these neurons encode different features remains to be determined. In this paper, we presented a computational framework to provide a system-level understanding of the encoding mechanism underlying SDM. We used conductance-based neuron models to construct a homogenous neural ensemble that encodes the slow and fast features of a mixed stimulus through asynchronous and synchronous spikes, respectively. To elucidate the contribution of slow and fast features of the mixed stimulus to the spikes generated by the model neurons, we calculated the most significant subspaces (eigenvectors) of the spike-triggered stimulus matrix using the iSTAC method. We demonstrated that the calculated first and second eigenvectors resemble the slow and fast features of the stimulus, respectively. Furthermore, the projection of the spike-triggered stimulus matrix on these eigenvectors created two distinct clusters. We tested whether these clusters can be characterized by synchronous and asynchronous spikes. By computing the spike-triggered average (STA) filters of the synchronous and asynchronous spikes and projecting this matrix on these filters, we clearly separated those clusters. Furthermore, we fitted an LNL model to each type of spike. Similar to distinct temporal filters for synchronous and asynchronous spikes, their static nonlinearities are different. We found that the nonlinearity associated with the asynchronous spikes is very shallow and can be approximated with a linear function. On the other hand, the nonlinearity associated with synchronous spikes has a very large slope and can be approximated using highly nonlinear functions such as the sigmoid function. Finally, we developed an augmented LNL model both to capture the dynamical characteristics of the synchronous and asynchronous spikes and to reconstruct the PSTH of all spikes.

3.1. Subspace Feature Extractors: iSTAC vs. STC

To explore more than one subspace feature for stimulation-evoked neural responses, we compared the performance of the STC and iSTAC methods. One can find the most informative subspaces that maximize the mutual information between stimulus and response [23,24]. Nevertheless, an accurate estimation of mutual information requires a large amount of data, although no guarantee for optimal estimation can be necessarily expected [24]. A conventional way to find these subspaces is to calculate those related to the most significant eigenvectors of the spike-triggered covariance (STC) matrix [16]. The eigenvectors of the STC matrix provide analytic expressions for filter estimation using the moments of the stimulus and spike-triggered stimulus distribution [16,17]. However, this method does not incorporate joint information between the mean and the variance, and there is also no specific measurement for selecting the most significant subspaces based on that information. We calculated the most important eigenvectors of the STC matrix underlying the mixed stimulus and neural response (see Figure 1). As one can expect, the first eigenvector of the STC matrix resembled that obtained using the iSTAC method and was similar to the STA of the asynchronous spikes. Unlike the first eigenvector of the STC matrix, the second eigenvector comprised both slow and fast features of the stimulus. Therefore, the 2D projection of the spike-triggered stimulus matrix on the eigenvectors of the STC matrix cannot be clearly separated into two distinct clusters.

To avoid this problem, we used the iSTAC method, which allows us to choose the eigenvectors of the spike-triggered stimulus ensemble matrix more precisely by minimizing the Kullback–Leibler (KL) divergence between the eigenvectors of this matrix and that obtained using raw stimulus distributions [17]. It is to be noted that the whitening transformation is usually used before finding subspaces of the spike response. One can use the whitening transformation to calculate the uncorrelated and normalized subspaces (for both STC and iSTAC methods), which represent an unbiased estimate of the neurons' temporal features. However, due to the type of mixed stimulus (i.e., structured and not a random process), we found that eliminating this transformation results in more representative sub-

space features, as shown in Figure 1, but it should be noted that these subspace features are biased to the choice of input stimuli. We compared the 2D projection of the spike-triggered stimulus matrix on the eigenvectors of the iSTAC method with and without whitening. It can be clearly observed that the iSTAC without whitening can better separate the 2D space.

3.2. Choice of Static Nonlinearity in the LNL Model

The static nonlinearities obtained in the augmented LNL model explains why the synchronous and asynchronous spikes are associated with different functions. For example, the smoothness and linear behavior of $f_{Async}(x)$, for $x > 0$, generates a smooth PSTH for the asynchronous spikes, which linearly encodes to the intensity of the stimulus. In contrast, the sigmoid-like nonlinearity of the synchronous spikes, $f_{Sync}(x)$, maintains the sparse PSTH of synchronous spikes because thanks to the steep nonlinearity. It is worth mentioning that more flexible nonlinear functions could provide better fits for representing the PSTH of the synchronous and asynchronous spikes. Of note, one can use deep neural networks (DNNs) to give more flexibility to the models. A DNN is simply a high-dimensional non-linear function estimator that gives a multilayer nonlinear function in the form of a neural network [25,26]. However, the main challenge with DNNs is their large parameter set, which demands a relatively larger dataset compared to what we used here to optimize the model parameters. Not only that, but also due to the large degree of freedom given by a DNN, we lose the interpretability of the modeling mechanism. In cases where these two points are concerned, one can replace our augmented LNL with a DNN to carry out the multiplexed encoding.

3.3. Generalized Linear Model (GLM) for Augmented LNL

The proposed augmented LNL can also be interpreted in the GLM framework. From this point of view, synchronous and asynchronous PSTHs are modeled with two separated GLMs with different random processes that eventually combine their PSTHs linearly. The first GLM filters the mixed stimulus with the first eigenvector of iSTAC and then, by passing it through a nonlinearity and then a Gaussian random process (with a linear nonlinearity as its link function), it models the PSTH related to the asynchronous spikes. Likewise, the second GLM filters the mixed stimulus with the second eigenvector of iSTAC and then, by passing it through a nonlinearity and then a Bernoulli random process (with a sigmoid nonlinearity as its link function), it models the PSTH related to the synchronous spikes (see Methods for more details about GLMs). Alternatively, we could simply interpret the augmented LNL as a single Poisson GLM with two input filters (the first two eigenvectors of iSTAC) and a Poisson random process at the end (see Appendix A for more details).

To reach the optimal parameter set for the model and avoid computational complexity, we use simple parametric models for the static nonlinearities [9]. We also can make the model more flexible by considering a flexible link function. We expect this to lead to a better fit for our model when using more complex models of neurons and eventually to lead to a better performance in encoding the fast and slow signals. For example, we can use a more flexible parametric function (with parameter set θ) such as the ex-quadratic function $f_\theta(x)$ as the static nonlinearities. To use the ex-quadratic function as nonlinearities we eventually need to optimize a convex cost function, which gives the optimum parameter set θ for the nonlinearity and can be optimized using a maximum-likelihood (ML) algorithm (details in Appendix B) [9]. The downside of this modeling is its computational complexity and the harder interpretation of the resulting model due to more parameters used in comparison with what we showed in this research.

4. Materials and Methods

4.1. Stimulated Mixed Input

According to the feasibility of neural systems to multiplexed coding, we simulated the activity of a homogeneous neural ensemble in response to a mixed stimulus to explore how much information can be encoded by different patterns of spikes. Each neuron

received a mixed signal (I_{mixed}) that consists of a fast signal (I_{fast}) and a slow signal (I_{slow}). I_{fast} stands for the timing of the fast events or abrupt changes in the stimulus and was generated by convolving a randomly (Poisson) distributed Dirac-delta function with a synaptic waveform (normalized to the peak amplitude), $\tau_{rise} = 0.5$ ms and $\tau_{fall} = 3$ ms. The fast events occurred at a rate of ~ 1 Hz and were scaled by $a_{fast} = 85$ pA.

I_{slow} was generated using an Ornstein-Uhlenbeck process as follows.

$$\frac{dI_{slow}}{dt} = -\frac{I_{slow}(t) - \mu}{\tau} + \sigma\frac{\sqrt{2}}{\tau}\xi(t), \ \ \xi \sim (0,1) \tag{3}$$

where ξ is a random number drawn from a Gaussian distribution and $\tau = 100$ ms is the time constant of the slow signal that produces a slow-varying random walk with an average of $\mu = 15$ pA and a standard deviation of $\sigma = 60$ pA. The mixed signal (I_{mixed}) was obtained by adding I_{fast} and I_{slow}, which were generated independently.

An independent noise (equivalent to the background synaptic activity) was added to each neuron; thus, each neuron receives a mixed signal plus noise. Similarly, the noise (I_{noise}) was generated using an Ornstein-Uhlenbeck process of $\tau = 5$ ms, $\mu = 0$ pA, and $\sigma = 10$ pA.

4.2. Simulated Neural Ensemble and Its Response to the Mixed Input

The neural ensemble consists of 30 neurons, each of which was modeled with Morris–Lecar equations [13,27]. The equations of a single model neuron receiving a mixed signal plus noise can be written as follows.

$$C\frac{dV}{dt} = I_{mixed}(t) \quad +I_{noise}(t) - \bar{g}_{Na}m_\infty(V)(V - E_{Na}) - \bar{g}_K w(V - E_K) - g_L(V - E_L) \\ -\bar{g}_{AHP}z(V - E_K) - g_{exc}(V - E_{exc}) - g_{inh}(V - E_{inh}) \tag{4}$$

where

$$\frac{dw}{dt} = \phi\frac{w(V) - w}{\tau_{W(V)}} \tag{5}$$

$$\frac{dz}{dt} = \frac{\frac{1}{1+e^{(\beta_z - V)/\gamma}} - z}{\tau_z} \tag{6}$$

$$m_\infty(V) = 0.5\left[1 + tanh\left(\frac{V - \beta_m}{\gamma_m}\right)\right] \tag{7}$$

$$w_\infty(V) = 0.5\left[1 + tanh\left(\frac{V - \beta_w}{\gamma_w}\right)\right] \tag{8}$$

$$\tau_w(V) = \frac{1}{cosh\left(\frac{V-\beta_w}{2\beta_w}\right)} \tag{9}$$

where $\{\bar{g}_{NA} = 20, \bar{g}_k = 20, \bar{g}_L = 20, \bar{g}_{AHP} = 25, g_{exc} = 1.2, g_{inh} = 1.9\}\frac{mS}{cm^2}$, $\beta_m = -1.2$ mV, $\gamma_m = 18$ mV, $\beta_w = -19$ mV, $\gamma_w = 10$ mV, $\beta_z = 0$ mV, $\gamma_z = 2$ mV, $\tau_a = 20$ ms, $\phi = 0.15$, and $C = 2\frac{\mu F}{cm^2}$. These parameters were set to ensure that a neuron operates in a hybrid mode [28], i.e., an operating mode between integration and coincidence detection [5,29]. The surface area of the neuron was set to 200 μm^2 so that I_{mixed} is reported in pA, rather than as a density [30,31]. Figure 1A shows the mixed stimulus and the spiking activities of the ensemble of neurons in response to this stimulus.

4.3. Generalized Linear Model (GLM) Details

A GLM is a generalization of traditional linear models that gives the neural encoding models more flexibility to capture the nonlinear dynamics of neural activity. GLMs contain three stages. The first stage is a linear mapping that consists of a set of d linear filters. Let

us assume $K = [k_1, \ldots, k_D]$, which maps a high-dimensional sensory stimulus $s \in R^M$ onto a low-dimensional stimulus feature map $x \in R^D$:

$$x = K^T s \tag{10}$$

The second stage is a pointwise nonlinearity, $f : R^D \rightarrow R$, which maps the linear features of d dimensions into a nonnegative spike rate:

$$\lambda = f(x) \tag{11}$$

In the final stage, the number of spikes is generated by a random process:

$$P_\theta(Y = r | s) \tag{12}$$

where Y is a random variable related to spike occurrence, r is the instantaneous firing rate, and θ is the parameter set of the random process

In simple words, by using a GLM we approximate the instantaneous firing rate by considering features from D dimensions instead of M dimensions:

$$P(Y | s) \sim P(Y | K^T s) \tag{13}$$

Thus, there are two sets of parameters, the estimators (K) and the pointwise nonlinearity (f), which can be optimized to reach the desired model.

4.4. STA and STC Estimators

If we assume that $p(s)$ has zero mean, then the STA can be defined as the average of the stimulus given the instantaneous firing rate:

$$\mu = \frac{1}{n_{sp}} \sum_{\{s_i | spike\}} s_i, \; n_{sp} = \sum_{t=1}^{N} r_t \tag{14}$$

where N is the total number of time points. The STA is an unbiased, consistent estimation that gives the direction in the stimulus space along which the means of $P(s|spike)$ and $P(s)$ differ the most. The problem is the STA gives a single direction in stimulus space and leads to a single estimator, which is not sufficient to capture all the information in the stimulus space (as previously discussed, we have a mixed stimulus in this research). To consider other possible directions with maximal differences in the variances between $P(s|spike)$ and $P(s)$ we can use the eigenvectors of the STC matrix, defined as:

$$\Lambda = \frac{1}{n_{sp}} \sum_{\{s_i | spike\}} (s_i - \mu)(s_i - \mu)^T \tag{15}$$

The STA and eigenvectors of the STC matrix can provide a basis, K, for a reduced-dimensional model of the neural response.

4.5. iSTAC Estimator

There are major problems with STA/STC. The measure we use to select the eigenvectors of STC is based on eigenvalues, which do not truly represent the most informative directions. As we mentioned before, the objective in iSTAC is to reduce the KL divergence between the Gaussian approximations to the spike-triggered and raw stimulus distributions. Therefore, we define Q as a Gaussian approximation of $P(s|spike)$ based on the information contained only in the mean and covariance of P as:

$$Q(s) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Lambda|^{\frac{1}{2}}} e^{-\frac{1}{2}(s-\mu)^T \Lambda^{-1}(s-\mu)} \tag{16}$$

where n is the dimensionality of the stimulus space. Now we drive KL the divergence between Q and P as:

$$D(Q,P) = \int Q(s) \, \log\left(\frac{Q(s)}{P(s)}\right) ds \tag{17}$$

By considering that the mean of P and $P \circ Q$ is zero and have identity covariance (if not, we can use a "whitening" technique) we can rewrite D in a simpler form as:

$$D(Q,P) = \frac{1}{2}\left(Tr(\Lambda) - \log|\Lambda| + \mu^T\mu - n\right) \tag{18}$$

where $Tr(.)$ and $|.|$ indicate the trace and determinant, respectively.

Based on the fact that we are interested in d subspaces we can approximate the D with:

$$D_{[K]}(P,Q) = \frac{1}{2}\left(Tr\left[K^T\left(\Lambda + \mu^T\mu\right)K - loglog\left|K^T\Lambda K\right| - d\right]\right) \tag{19}$$

where d is the dimension of the interested subspaces.

Thus, in terms of finding the d most informative subspaces decomposed by STA and the eigenvectors of STC, we need to find $D_{[K]}(P,Q)$ for all subspaces and select the first d ones.

An important advantage of the iSTAC approach over traditional STA/STC analysis is that it makes statistically efficient use of changes in both the mean and covariance of the response spaces.

Appendix A

We assume the number of spikes in time are discrete events. By dividing the time horizon of the experiment, $(0, T]$, into K (K is a large number) subintervals $(t_{k-1}, t_k]_{k=1}^K$ and considering N_k as the number of events in the time interval $(t_{k-1}, t_k]$, we can model the spike observation process with a point process by inhomogeneous Poisson distribution and parameter λ_k as:

$$P(N_k|s_k) = \frac{1}{N_k!}(\Delta\lambda_k)^{N_k}e^{-\Delta\lambda_k} \tag{A1}$$

where Δ is the width of the time bins. By assuming that the number of spikes is conditionally independent in time, we can write the whole observation process as

$$P(N_{1:K}|s_{1:K}) = \prod_{k=1}^K \frac{1}{N_k!}(\Delta\lambda_k)^{N_k}e^{-\Delta\lambda_k} \tag{A2}$$

On the other hand, we can model λ_k in a way that captures the effects of two features of the stimulus as a linear combination of their effects in λ_k (discussed in the Results section), as

$$\lambda_k = \sum_{i=1}^{D} \omega_i f_i(\mu_i * s_k) \tag{A3}$$

where $\theta = \{\omega_1, \ldots, \omega_D\}$ is our parameter set and D is the dimension of the feature map. Finally, using maximum-likelihood estimation, we can tune the model parameters

$$\hat{\theta} = arglog\ P(N_{1:K}|s_{1:K}) \rightarrow$$
$$log\ P(N_{1:K}|s_{1:K}) = \sum_{k=1}^{K} log\frac{1}{N_k!} + \sum_{k=1}^{K} N_k log\ \Delta + N_k\left(\sum_{i=1}^{D} \omega_i f_i(\mu_i * s_{1:k})\right) \tag{A4}$$
$$-\Delta \sum_{i=1}^{D} \omega_i f_i(\mu_i * s_{1:k})$$

Based on Jensen's inequality and considering that $loglog\ x$ is a concave function, we have:

$$log\ P(N_{1:K}|s_{1:K}) \geq C + \sum_{k=1}^{K} N_k\left(\sum_{i=1}^{D}(\omega_i f_i(\mu_i * s_{1:k})) - \Delta \omega_i f_i(\mu_i * s_{1:k})\right) = Q,$$
$$C = \sum_{k=1}^{K} log\ \frac{1}{N_k!} + \sum_{k=1}^{K} N_k log\ \Delta + N_k \tag{A5}$$

where Q is a lower bound for the log-likelihood. Therefore, we can find the θ s by maximizing the Q over them as

$$\hat{\theta_{ML}} = \frac{\partial log\ P(N_{1:K}|s_{1:K})}{\partial \theta} = 0 \rightarrow$$
$$\frac{\partial log\ P(N_{1:K}|s_{1:K})}{\partial \theta} = 0 + \sum_{k=1}^{K}\left(\frac{N_k}{\omega_i} - \Delta f_i(\mu_i * s_{1:k})\right) = \tag{A6}$$
$$\sum_{k=1}^{K} N_k - \frac{1}{\omega_i}\sum_{k=1}^{K} \Delta f_i(\mu_i * s_{1:k}) = 0 \rightarrow \hat{\omega}_i = \frac{\sum_{k=1}^{K} N_k}{\sum_{k=1}^{K} \Delta f_i(\mu_i * s_{1:k})}$$

By estimating the model parameters, we reach a model for encoding the firing rate of the multiplexed spikes.

Appendix B

The estimators perform a dimensionality-reduction task and map a high-dimensional sensory stimulus to a lower-dimensional linear feature map $= K^T s$, where K is the basis of the feature map space. Based on the definition of GLM mentioned above, we still need to find the optimum model for $f(x)$, $f : R^d \rightarrow R$. By considering the motivation in [4], a reasonable way is using an exponential general quadratic function:

$$f = exp\left(\frac{1}{2}x^T Cx + b^T x + a\right) \tag{A7}$$

where C is a symmetric matrix, b is a vector, and a is a scalar. Thus, now we can use maximum likelihood to optimize the parameter set, $\{C, b, a\}$. To do that we need to maximize the log-likelihood of observing a spike given all spikes and the parameters set ($L = loglog\ P(r_{1:N}|s_{1:N}, C, b, a)$). By assuming that the spikes firing in time are independent, we can rewrite it as:

$$L = \frac{1}{n_{sp}}\sum_i log\ P(r_i|s_i, C, b, a) \tag{A8}$$

where n_{sp} is total number of spikes, so our objective is to maximize L by finding the best parameter set:

$$\left\{\hat{C}, \hat{b}, \hat{a}\right\} = argmax_{\{C,b,a\}}\ L \tag{A9}$$

By following the optimization steps in [5] and assuming that the stimulus is drawn from $x \sim N(0, \Phi)$; the maximum-likelihood estimation of the parameters is:

$$\hat{C} = \Phi^{-1} - \Lambda^{-1}, \ \hat{b} = \Lambda^{-1}\mu \tag{A10a}$$

$$\hat{a} = log\left(\frac{nsp}{N}\left|\Phi\Lambda^{-1}\right|^{0.5}\right) - \frac{1}{2}\mu^T\Phi^{-1}\Lambda^{-1}\mu \tag{A10b}$$

References

1. Li, Y.G.; Winters, J.H.; Sollenberger, N.R. MIMO-OFDM for wireless communications: Signal detection with enhanced channel estimation. *IEEE Trans. Commun.* **2002**, *50*, 1471–1477. [CrossRef]
2. Laughlin, S.B.; Sejnowski, T.J.J.S. Communication in neuronal networks. *Science* **2003**, *301*, 1870–1874. [CrossRef] [PubMed]
3. Johnson, D.H. The relationship between spike rate and synchrony in responses of auditory-nerve fibers to single tones. *J. Acoust. Soc. Am.* **1980**, *68*, 1115–1122. [CrossRef] [PubMed]
4. Harvey, M.A.; Saal, H.P.; Dammann, J.F., III; Bensmaia, S.J. Multiplexing stimulus information through rate and temporal codes in primate somatosensory cortex. *PLoS Biol.* **2013**, *11*, e1001558. [CrossRef]
5. Lankarany, M.; Al-Basha, D.; Ratté, S.; Prescott, S.A. Differentially synchronized spiking enables multiplexed neural coding. *Proc. Natl. Acad. Sci. USA* **2019**, *116*, 10097–10102. [CrossRef]
6. Sullivan, W.; Konishi, M. Segregation of stimulus phase and intensity coding in the cochlear nucleus of the barn owl. *J. Neurosci.* **1984**, *4*, 1787–1799. [CrossRef]
7. Saal, H.P.; Harvey, M.A.; Bensmaia, S.J. Rate and timing of cortical responses driven by separate sensory channels. *eLife* **2015**, *4*, e10450. [CrossRef]
8. Saal, H.P.; Bensmaia, S.J. Touch is a team effort: Interplay of submodalities in cutaneous sensibility. *Trends Neurosci.* **2014**, *37*, 689–697. [CrossRef]
9. Paninski, L. Maximum likelihood estimation of cascade point-process neural encoding models. *Netw. Comput. Neural Syst.* **2004**, *15*, 243–262. [CrossRef]
10. Latimer, K.W.; Barbera, D.; Sokoletsky, M.; Awwad, B.; Katz, Y.; Nelken, I.; Lampl, I.; Fairhall, A.L.; Priebe, N.J. Multiple timescales account for adaptive responses across sensory cortices. *J. Neurosci.* **2019**, *39*, 10019–10033. [CrossRef]
11. Churchland, M.M.; Byron, M.Y.; Sahani, M.; Shenoy, K.V. Techniques for extracting single-trial activity patterns from large-scale neural recordings. *Curr. Opin. Neurobiol.* **2007**, *17*, 609–618. [CrossRef] [PubMed]
12. Paninski, L.; Pillow, J.; Lewi, J. Statistical models for neural encoding, decoding, and optimal stimulus design. *Prog. Brain Res.* **2007**, *165*, 493–507.
13. Rezaei, M.R.; Popovic, M.R.; Lankarany, M. A Time-Varying Information Measure for Tracking Dynamics of Neural Codes in a Neural Ensemble. *Entropy* **2020**, *22*, 880. [CrossRef] [PubMed]
14. Prescott, S.A.; Ratté, S.; De Koninck, Y.; Sejnowski, T.J. Pyramidal neurons switch from integrators in vitro to resonators under in vivo-like conditions. *J. Neurophysiol.* **2008**, *100*, 3030–3042. [CrossRef] [PubMed]
15. Paninski, L. Convergence properties of three spike-triggered analysis techniques. *Netw. Comput. Neural Syst.* **2003**, *14*, 437–464. [CrossRef]
16. Schwartz, O.; Pillow, J.W.; Rust, N.C.; Simoncelli, E.P. Spike-triggered neural characterization. *J. Vis.* **2006**, *6*, 13. [CrossRef]
17. Pillow, J.W.; Simoncelli, E.P. Dimensionality reduction in neural models: An information-theoretic generalization of spike-triggered average and covariance analysis. *J. Vis.* **2006**, *6*, 9. [CrossRef]
18. Lankarany, M.; Prescott, S.A. Multiplexed coding through synchronous and asynchronous spiking. *BMC Neurosci.* **2015**, *16*, P198. [CrossRef]
19. Bojak, I.; Nowotny, T. 27th Annual Computational Neuroscience Meeting (CNS* 2018): Part Two. *BMC Neurosci.* **2018**, *19*, 65.
20. Ramachandran, P.; Zoph, B.; Le, Q.V. Searching for activation functions. *arXiv* **2017**, arXiv:1710.05941.
21. Pillow, J.W.; Paninski, L.; Uzzell, V.J.; Simoncelli, E.P.; Chichilnisky, E. Prediction and decoding of retinal ganglion cell responses with a probabilistic spiking model. *J. Neurosci.* **2005**, *25*, 11003–11013. [CrossRef] [PubMed]
22. Pillow, J.W.; Shlens, J.; Paninski, L.; Sher, A.; Litke, A.M.; Chichilnisky, E.; Simoncelli, E.P. Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature* **2008**, *454*, 995–999. [CrossRef] [PubMed]
23. Paninski, L. Estimation of entropy and mutual information. *Neural Comput.* **2003**, *15*, 1191–1253. [CrossRef]
24. Sharpee, T.; Rust, N.C.; Bialek, W. Analyzing neural responses to natural signals: Maximally informative dimensions. *Neural Comput.* **2004**, *16*, 223–250. [CrossRef]
25. Moskovitz, T.H.; Roy, N.A.; Pillow, J.W. A comparison of deep learning and linear-nonlinear cascade approaches to neural encoding. *BioRxiv* **2018**, 463422. [CrossRef]
26. Rezaei, M.R.; Gillespie, A.K.; Guidera, J.A.; Nazari, B.; Sadri, S.; Frank, L.M.; Eden, U.T.; Yousefi, A. A Comparison Study of Point-Process Filter and Deep Learning Performance in Estimating Rat Position Using an Ensemble of Place Cells. In Proceedings

of the 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Honolulu, HI, USA, 18–21 July 2018; pp. 4732–4735.

27. Morris, C.; Lecar, H. Voltage oscillations in the barnacle giant muscle fiber. *Biophys. J.* **1981**, *35*, 193–213. [CrossRef]
28. Ratté, S.; Hong, S.; De Schutter, E.; Prescott, S.A. Impact of neuronal properties on network coding: Roles of spike initiation dynamics and robust synchrony transfer. *Neuron* **2013**, *78*, 758–772. [CrossRef]
29. Pirschel, F.; Kretzberg, J. Multiplexed population coding of stimulus properties by leech mechanosensory cells. *J. Neurosci.* **2016**, *36*, 3636–3647. [CrossRef]
30. Destexhe, A.; Rudolph, M.; Paré, D. The high-conductance state of neocortical neurons in vivo. *Nat. Rev. Neurosci.* **2003**, *4*, 739–751. [CrossRef]
31. Prescott, S.A.; De Koninck, Y.; Sejnowski, T.J. Biophysical basis for three distinct dynamical mechanisms of action potential initiation. *PLoS Comput. Biol.* **2008**, *4*, e1000198. [CrossRef]