Neural Network Poisson Models for Behavioural and Neural Spike Train Data

Amir Dezfouli[1*] **Richard Nock**[12] **Ehsan Arabzadeh**[2] **Peter Dayan**[3]

[1]Data61, CSIRO [2]Australian National University [3]Max Planck Institute
[*]amir.dezfouli@data61.csiro.au

Abstract

It is now possible to monitor the activity of a large number of neurons across the brain as animals perform behavioural tasks. A primary aim for modeling is to reveal (i) how sensory inputs are represented in neural activities and (ii) how these representations translate into behavioural responses. Predominant methods apply rather disjoint techniques to (i) and (ii); by contrast, we suggest an end-to-end model which jointly fits both behaviour and neural activities and tracks their covariabilities across trials using inferred noise correlations. Our model exploits recent developments of flexible, but tractable, neural network point-process models to characterize dependencies between stimuli, actions and neural data. We apply the framework to a dataset collected using Neuropixel probes in a visual discrimination task and analyse noise correlations to gain novel insights into the relationships between neural activities and behaviour.

1 Introduction

Recent developments in neural recording techniques such as Neuropixel probes allow the activity of large numbers of neurons across the brain to be monitored as animals perform behavioural tasks [Jun et al., 2017]. Such recordings allow us to study how the brain represents past and present sensory inputs and how these representations evolve over time and across areas, ultimately leading to behavioural outputs. Very coarsely, there are two main, and highly successful, approaches to linking activity and behaviour [Paninski et al., 2007, Mante et al., 2013, Ganguli and Sompolinsky, 2012, Kass et al., 2014]: *encoding* models that take mechanistic process accounts of tasks and the stimuli and responses on individual trials and try to account for neural activity on those trials; and *decoding* models that reveal what stimulus and response information is represented in the activity of individual neurons or (often low-dimensional projections of) the activity of populations of neurons.

Encoding models offer an attractive, high-level, interpretation of activity [e.g., Meyer et al., 2017, Kohn et al., 2016, Aljadeff et al., 2016]. However, simple reflections of the computational constraints of the task are often inadequate to detect and characterise aspects of the neural representations and computations that operate according to different rules. This is particularly important given that the neural representations of the sensory inputs and actions are distributed across different brain regions [Steinmetz et al., 2019] and can also evolve over time in neural modes that have no behavioural consequence. Exactly same is true of *replicator* models that are based on trying to build high-dimensional feedforward or recurrent neural network (RNN) models of the behaviour [Sussillo, 2014, Richards et al., 2019].

Decoding models provide a more transparent window into what aspects of the external world neurons are representing. Flexible substrates such as artificial neural networks, which, in principle, can match the complexity of any underlying neural processes, have been shown to be effective in modelling detailed neural dynamics [e.g., Pandarinath et al., 2018, Shenoy et al., 2013, Kennedy and Schwartz,

Preprint. Under review.

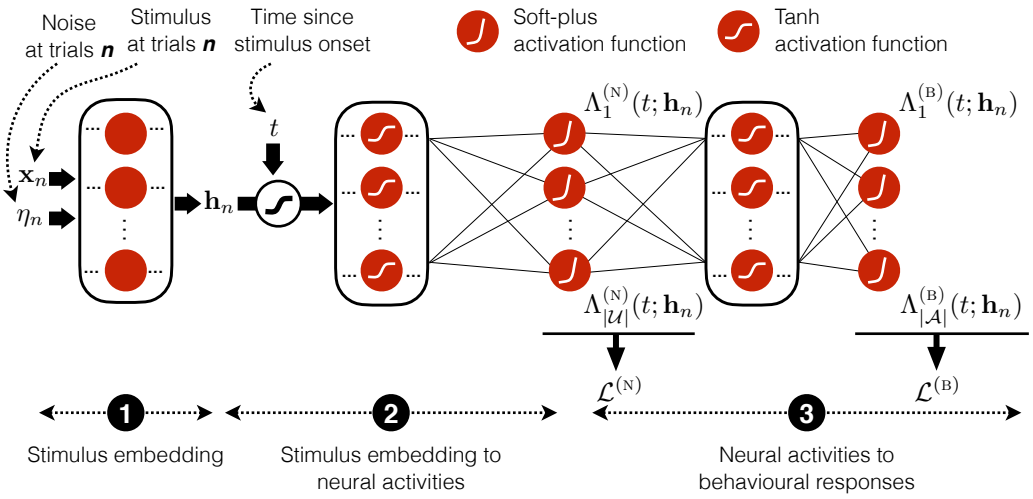

Figure 1: Network architecture. ① Embedding of input stimulus and trial noise into vector $\mathbf{h}_n$. ② Transformation of embedding $\mathbf{h}_n$ into neural activity of each region for each time point t since the stimulus onset. Neural activities are characterised by the *cumulative* intensity function of spike train for each region, denoted by $\Lambda_u^{(\mathrm{N})}(t;\mathbf{h}_n)$ for regions $u \in \mathcal{U}$. The intensity function $\lambda_u^{(\mathrm{N})}(t;\mathbf{h}_n)$ is obtained by differentiating $\Lambda_u^{(\mathrm{N})}(t;\mathbf{h}_n)$ with respect to t. Component ② structurally ensures that $\Lambda_u^{(\mathrm{N})}(t;\mathbf{h}_n)$ increases with t, so $\lambda_u^{(\mathrm{N})}(t;\mathbf{h}_n) \geq 0$. ③ Neural activities are mapped to behavioural responses which are represented by the cumulative intensity function $\Lambda_a^{(\mathrm{B})}(t;\mathbf{h}_n)$ for making each action $a \in \mathcal{A}$ at each time t since stimulus onset (t). The data likelihood is computed over both recorded spike trains and behavioural responses. This yields a neural log-likelihood function ($\mathcal{L}^{(\mathrm{N})}$) and a behavioural log-likelihood function ($\mathcal{L}^{(\mathrm{B})}$) which are combined to train all the weights.

2019]. However, the computational constraints of appropriate behaviour on the tasks typically play second fiddle in these approaches, leaving us unclear about how the neural activity is structured to deliver appropriate performance.

Recently, Dezfouli et al. [2018] suggested a combined approach, with an RNN being trained to tie directly fMRI BOLD activity across the brain with ongoing behaviour. fMRI data allow for a form of model inversion, pinning down the RNN state and so implying how behaviour would be realized neurally. However, this approach requires the invertibility that comes from the dimensionality of fMRI – something that is not currently available on a trial-by-trial basis in neural recordings.

Here, building upon these previous approaches and recent neural network point-process models [Omi et al., 2019], we provide a novel neural network Poisson process model (Figure 1) which: (i) flexibly learns the connections between environmental stimuli and neural representations, and between neural representations and behavioural responses, (ii) jointly fits both behavioural and neural data, (iii) tracks between-trial variabilities in neural and behavioural processes using noise correlations, which (iv) allows us to trace different stimulus-driven behavioural patterns back to their neural substrates. The framework allows efficient training of the model without making assumptions about the functional form of how neural and behavioural processes are driven by input stimuli and noise correlations. We apply the method to a dataset collected using Neuropixel probes [Steinmetz et al., 2019] in a visual discrimination task, and show, by analysing noise correlations, that the method is able to link behavioural data with their underlying neural processes and input stimuli. In the following, we start by introducing the input data and a mathematical description of the model, we then present the architecture of the neural network, and finish by showing the results and elaborating on how to interpret the model using noise correlations.

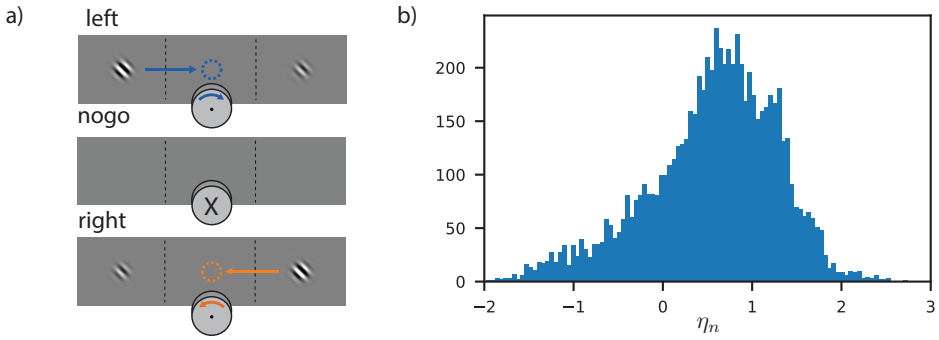

Figure 2: **a)** The visual discrimination task. On each trial, visual stimuli were potentially presented on the left and right sides, and subjects should act ($a \in \{\text{LEFT}, \text{RIGHT}\}$) according to their relative contrasts. If both sides had zero contrast, reward was earned by making no response. Otherwise, for equal contrasts, animals were rewarded at random. The figure is adapted from Steinmetz et al. [2019]. **b)** Histogram of noise correlations learned by the model.

2 The model

2.1 The data

We model the canonical visual discrimination experiment of Steinmetz et al. [2019] (Figure 2a). On each trial, subjects are presented with a stimulus (visual contrast on the left or right side) and had to react with a simple movement (moving a wheel left or right or not moving). The whole experiment comprises multiple sessions, each of which consists of multiple trials, with different neurons being recorded on each session.

We formalize this as follows: the total number of trials in the whole experiment is denoted by $|\mathcal{N}|$; the stimuli on trial $n \in \{1 \dots |\mathcal{N}|\}$ are denoted by vector $\mathbf{x}_n$; in response to the stimuli, animals either can take an action or choose not to make any response. If an animal made a response on trial n, we denote it by $a_n \in \mathcal{A}$ (where, here, $\mathcal{A} = \{\text{LEFT}, \text{RIGHT}\}$. The time at which an action was taken *relative to the stimulus onset* is denoted by r_n.

On trial n, the activities of a set $\mathcal{U}_n$ of neurons are monitored; different neurons in different areas and even different animals may be recorded on separate trials. In total, $|\mathcal{S}_{u,n}|$ spikes are recorded at times $\{s^i_{u,n}\}_{i=1\dots|\mathcal{S}_{u,n}|}$, relative to stimulus onset in the corresponding trials. Following previous work [Steinmetz et al., 2019], we limit the observation window to $W = 400$ms after the stimulus onset, to focus on spikes closely associated with action.

We first discuss how we model the neural data; and then how we couple this to behaviour.

2.2 Neural network Poisson process models

We make the simplification that the spikes of each neuron u can be modelled as the outputs of an inhomogeneous Poisson processes [Daley and Vere-Jones, 2006] with a latent intensity function $\lambda^{(\text{N})}_u(t; \mathbf{h}_n)$ (the superscript (N) indicates that the intensity function is for the (N)eural data). This can be interpreted as the instantaneous probability of observing a spike at time t, where $\mathbf{h}_n$ is not only a function of the stimulus $\mathbf{x}_n$ but can also be affected by other factors such as noise (η_n). The major assumption is that successive spike times are independent, given $\lambda^{(\text{N})}_u(t; \mathbf{h}_n)$.

2.2.1 Spike train models

On trial n stimulus embedding $\mathbf{h}_n$ arises, and spike times $\{s^i_{u.n}\}_{i=1\dots|\mathcal{S}_{u.n}|}$ are observed from neuron u. We consider spikes either until either a response a_n was made at time r_n, or to the end of time window W, whichever comes first, i.e., up to $W_n = \min(W, r_n)$. The joint probability density of observing spike trains from neuron u in trial n is then

$$f^{(\text{N})}_{u,n}(s^1_{u,n} \dots s^{|\mathcal{S}^{u,n}|}_{u,n}) = \prod_{i=1}^{|\mathcal{S}_{u,n}|} \lambda^{(\text{N})}_u(s^i_{u,n}; \mathbf{h}_n) \exp\left(-\int_0^{W_n} \lambda^{(\text{N})}_u(t; \mathbf{h}_n) dt\right). \quad (1)$$

Intuitively, the term $\lambda_u^{(\mathrm{N})}(s_{u,n}^i; \mathbf{h}_n)$ represents the probability of observing a spike at time $s_{u.n}^i$ and the exponential terms represents the probability of not observing spikes in the observation period.

The dependence of the intensity function on the embedding $\mathbf{h}_n$ plays a crucial role in determining how neural activities are shaped by the past stimulus and has to be characterized in a flexible manner. To achieve this, one option is to use a multi-layer feed-forward network which takes t and $\mathbf{h}_n$ as inputs, and outputs $\lambda_u^{(\mathrm{N})}(t; \mathbf{h}_n) \geq 0$. Unfortunately, it is then intractable to calculate the integral in equation 1. An elegant solution to this problem suggested by Omi et al. [2019] is to parameterize the cumulative intensity function $\Lambda_u^{(\mathrm{N})}(t; \mathbf{h}_n)$ instead of $\lambda_u^{(\mathrm{N})}(t; \mathbf{h}_n)$. This is

$$\Lambda_u^{(\mathrm{N})}(t; \mathbf{h}_n) = \int_0^t \lambda_u^{(\mathrm{N})}(\tau; \mathbf{h}_n) d\tau. \tag{2}$$

We represent $\Lambda_u^{(\mathrm{N})}(t; \mathbf{h}_n)$ using a feed-forward network. The intensity function itself can be obtained by differentiating the cumulative intensity function w.r.t to t,

$$\lambda_u^{(\mathrm{N})}(t; \mathbf{h}_n) = \frac{\partial \Lambda_u^{(\mathrm{N})}(t; \mathbf{h}_n)}{\partial t}, \tag{3}$$

which can be performed by automatic methods. Note that the gradient of the cumulative intensity function w.r.t to t must always be non-negative (see Section 3). Given these two related functions, the data log-likelihood implied by equation 1 is

$$\mathcal{L}_{u.n}^{(\mathrm{N})} = \log f_{u.n}^{(\mathrm{N})}(s_{u.n}^1 \dots s_{u.n}^{|\mathcal{S}^{u.n}|}) = \sum_{i=1}^{|\mathcal{S}_{u.n}|} \left[\log \frac{\partial \Lambda_u^{(\mathrm{N})}(t = s_{u.n}^i; \mathbf{h}_n)}{\partial t} \right] - \Lambda_u^{(\mathrm{N})}(W_n; \mathbf{h}_n), \tag{4}$$

which retains the required flexibility, while obviating the calculation of the intractable integral. The total data likelihood for all the trials and neurons is then

$$\mathcal{L}^{(\mathrm{N})} = \sum\nolimits_{t=1}^{|\mathcal{N}|} \sum\nolimits_{u \in \mathcal{U}_n} \mathcal{L}_{u.n}^{(\mathrm{N})}.$$

In principle, each recorded neuron in the experiment could be assigned a separate intensity function. However, given the experimental methodology of only recording some neurons on some trials, the problem of missing data would be radically acute, and the model would be uninterpretable. The computational cost would also be penal. Instead, we make the simplifying assumption that all the neurons in each of the 42 brain regions identified by [Steinmetz et al., 2019] share a common intensity function. As such, we consider $|\mathcal{U}| = 42$.

2.2.2 Behavioural response models

Having provided a way of characterizing the neural response to the embedding $\mathbf{h}_n$, we next need to model the link between neural representations and behaviour. We assume that the probability of making a behavioural response at each point in time depends on the activity of the neurons at that time. In turn, these are driven by the stimulus $\mathbf{h}_n$ on that trial. That is, the behavioural responses are indirectly affected by the stimulus via neural activities. However, rather than model this dependence explicitly, which is hard given the punctate nature of the response, we approximate it implicitly, via smooth intensity functions that in turn depend on $\Lambda^{(\mathrm{N})}$. This amounts to capturing a form of what are sometimes called signal correlations between the brain and behaviour. We discuss noise correlations later.

The intensity function for an action a is denoted by $\lambda_a^{(\mathrm{B})}(t; \mathbf{h}_n)$, which specifies the instantaneous probability of taking the action at time t relative to stimulus onset. The superscript (B) indicates that the intensity function is for the (B)ehavioural data. A key simplification is to allow for the theoretical possibility that the animal performs the same action more than once on a trial; or performs both actions. However, that actions are actually sparse implies that this approximation is not too costly. We write the behavioural cumulative intensity function as a function of the neural cumulative intensity functions,

$$\Lambda_a^{(\mathrm{B})}(t; \mathbf{h}_n) = \Phi_a(\Lambda_1^{(\mathrm{N})}(t; \mathbf{h}_n), \dots, \Lambda_{|\mathcal{U}|}^{(\mathrm{N})}(t; \mathbf{h}_n)), \tag{5}$$

Function $\Phi_a(.)$ can be realised using a deep feed-forward network which can represent arbitrary dependencies between neural activities and behavioural responses. Then, differentiating:

$$\lambda_a^{(\mathrm{B})}(t; \mathbf{h}_n) = \sum\nolimits_{u=1}^{|\mathcal{U}|} \lambda_u^{(\mathrm{N})}(t; \mathbf{h}_n) \frac{\partial \Phi_a(.)}{\partial \Lambda_u^{(\mathrm{N})}}, \tag{6}$$

which implies that the behavioural response probability at each point in time is indirectly dependent on the stimulus through the spike rate of different neural activities, as desired. Function $\Phi(.)$ is also designed to be increasing in t to ensure that $\lambda_a^{(\mathrm{B})}(t; \mathbf{h}_n) > 0$ (see Section 3).

The subjects can either act or not on a trial; the latter is determined by a censoring window W. Write $\mathcal{N}_a$ as the set of trials on which action a was taken before W, at reaction time r_n. The simplified joint probability distribution of the behavioural observations is then:

$$f_a^{(\mathrm{B})}(\{r_n\}_{t\in\mathcal{N}_a}) = \left(\prod_{t\in\mathcal{N}_a} \lambda_a^{(\mathrm{B})}(r_n; \mathbf{h}_n)\right) \prod_{t=1}^{|\mathcal{N}|} \exp\left(-\int_0^W \lambda_a^{(\mathrm{B})}(t; \mathbf{h}_n)dt\right), \tag{7}$$

and, taking logs, the log-likelihood for those observations is:

$$\mathcal{L}_a^{(\mathrm{B})} = \sum_{t\in\mathcal{N}_a} \log \frac{\partial \Lambda_a^{(\mathrm{B})}(t = r_n; \mathbf{h}_n)}{\partial t} - \sum_{t=1}^{|\mathcal{N}|} \Lambda_a^{(\mathrm{B})}(W; \mathbf{h}_n). \tag{8}$$

Over the whole experiment and actions, the behavioural likelihood can be defined as,

$$\mathcal{L}^{(\mathrm{B})} = \sum_{a\in\mathcal{A}} \mathcal{L}_a^{(\mathrm{B})}, \tag{9}$$

in which $\mathcal{A}$ is the set of available actions.

2.3 Signal and noise correlations

One apparent limitations of this model is that it is purely feedforward - and so, unlike the RNN characterization in Dezfouli et al. [2018], does not directly represent the causal relationships between neural activity at one time and activity later, or between neural activity and later behaviour. Instead, it only captures signal correlations – i.e., the general relationships, averaged across trials with the same stimuli.

To help mitigate this, we included a noise term η_n in the input. This can capture certain noise correlations - i.e., it can change the embedding $\mathbf{h}_n$ to capture, for instance, the general pattern of neural activity associated with early versus late responses for a given stimulus condition $\mathbf{x}_n$. Of course, we do not know the value of η_n on a trial; rather we have to infer it using the network directly (under suitable regularization conditions). Nevertheless, even with η_n, some noise correlations are still omitted – the model only characterizes the *rate* functions for the neurons (and behaviour), and so is not able to capture fully the effects of the noise associated with individual spiking.

3 Network architecture

We implement the model using the neural network architecture shown in Figure 1. This has three components. The first maps the stimulus $\mathbf{x}_n$ that was presented, and the noise added η_n through a series of fully connected layers to realize an input embedding denoted by $\mathbf{h}_n$. We infer η_n by backpropagating all the way through the network. Since there is one η_n per trial, the model can potentially make trial-level predictions without using $\mathbf{x}_n$. To address this, we used a drop-out layer after η_n and in each training iteration the η_n of a certain portion of trials is replaced with zero. This encourages the model also to use stimulus information for making predictions.

The second component of the architecture takes the embedding $\mathbf{h}_n$ and t and outputs the modelled activity of each neural region u at time t in the form of cumulative intensity functions for $\Lambda_u^{(\mathrm{N})}(t; \mathbf{h}_n)$. This component is designed such that the outputs of the network, i.e., $\Lambda_u^{(\mathrm{N})}(t; \mathbf{h}_n)$s, are monotonic functions of t to ensure that their gradients with respect to t (which are neural intensity functions) are always positive. To achieve this, following ideas from Sill [1998], Chilinski and Silva [2018], Omi et al. [2019], the weights of the network are constrained to be positive and tanh activation functions are used in the middle layers and soft-plus in the output layers.

The third component of the model takes the neural cumulative intensity functions and maps them to the behavioural cumulative intensity functions (function Φ in equation 5). We used the same method as for the second component to ensure that the gradient of $\Lambda_a^{(\mathrm{B})}(t; \mathbf{h}_n)$ with respect to t is positive. See Supplementary Materials for details.

Following ideas from multi-task learning [e.g., Liao et al., 2016, Kokkinos, 2017], the weights of the networks are trained using a weight sum of the behavioural and neural log-likelihood functions. We

defined a joint objective function as follows,

$$\mathcal{L}^{\text{joint}} = \alpha^{(\text{N})}\mathcal{L}^{(\text{N})} + \alpha^{(\text{B})}\mathcal{L}^{(\text{B})}, \tag{10}$$

in which $\alpha^{(\text{B})}$ and $\alpha^{(\text{N})}$ are hyper-parameters that determined the weight of behavioural and neural data in the total likelihood function. The gradients were obtained using automatic differentiation in Tensorflow [Abadi et al., 2015]. The details of the network architecture are presented in section 5.2.

4 Related works

There is a large body of studies for modelling the relationships between sensory inputs, neural spike trains and behaviour. The main focus has been on representing stimulus features that drive neural activities over time using methods such as spike-triggerd average/covariance and generalized-linear models [e.g., Pillow et al., 2008, Marmarelis and Naka, 1972, Van Steveninck and Bialek, 1988] and transforming the representations to the neural activities based on linear/non-linear functions (such as exponential, sigmoid) and Poisson distribution [e.g., Truccolo et al., 2005, Paninski, 2004]. Our proposed method extends these approaches by automatically learning useful stimulus features and also incorporating a flexible non-linear transformation for mapping stimulus features to neural activities and eventually to behaviour.

Another influential approach in this area is based on statistical modelling of population of recorded neurons using methods such as dimensionality reduction [Cunningham and Byron, 2014]. This includes using traditional methods such as Principal component analysis (PCA) to recent techniques such as recurrent neural networks [e.g., Mante et al., 2013, Pandarinath et al., 2018]. The benefit of these methods over the proposed method is capturing the representation of spike dynamics and dependencies across recorded neurons. That is, the current method represents the dependency of neural activities on stimulus and noise, but not on the previous neural activities. This simplifying assumption was because our aim was to study the relationship between neural data, stimuli and behaviour (see Discussion).

5 Results

5.1 Data

We use the data reported in Steinmetz et al. [2019][1]. The experiments consist of 38 sessions of a visual discrimination task (Figure 2a). Activities from 42 brain regions from the left hemisphere of the brain were recorded (not all regions were recorded in all the sessions). Overall the data from 30,000 neurons were recorded and the whole experiment consisted of 10011 trials. We used the data from 12 sessions for testing and the rest for training the model.

On each trial, the animals were presented with stimuli on the left and right and were required to turn a wheel LEFT, RIGHT or keep it fixed, based on the contrast input (four possible levels of contrast on each side: 0, 25, 50, 100; contrast 0 corresponds to zero contrast). We encoded stimulus contrast using one-hot encoding ($\mathbf{x}_n$ of dimension 8). The reaction time r_n corresponds to the beginning of the wheel turn.

5.2 Training process

All the weights in the model were trained using Adam optimiser [Kingma and Ba, 2014]. A high, 90%, drop-out rate for η_n (i.e., in 90% of trials in each round the value of η_n is replaced with zero) proved necessary to prevent the input stimuli $\mathbf{x}_n$ from being ignored.

Following procedures in Dezfouli et al. [2018], we first trained the model by setting the neural weight to zero ($\alpha^{(\text{N})} = 0$) and obtained the best log-likelihood associated with the behavioural data. We then performed a grid-search using different weights and network parameters to find a configuration which (i) can reach the target behavioural likelihood function on the test data (using $\eta_n = 0$), and (ii) has the capacity to model the neural data. We determined the latter using early stopping – selecting the training iteration with the best likelihood on the test data. The selected model had 10 neurons in each

[1] https://github.com/nsteinme/steinmetz-et-al-2019/wiki/data-files

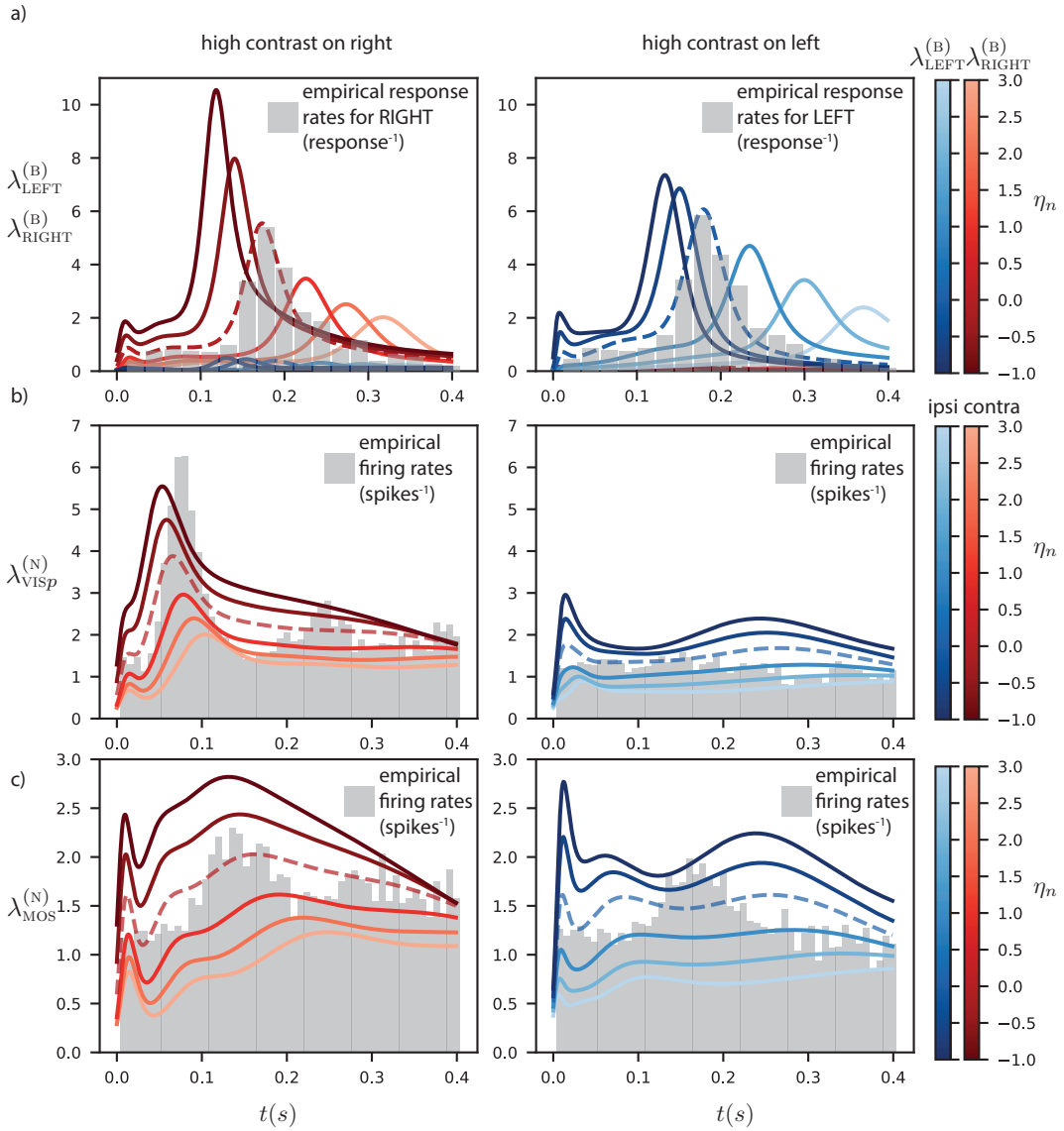

Figure 3: **Linking behavioural and neural processes using noise correlations.** In the left panels, the stimulus input to the model represented a high contrast on the right side and zero contrast on the left side ('high contrast on right'). In the right panels, the stimulus input to the model indicated zero contrast on left and high contrast on right ('high contrast on left'). **a)** Behavioural intensity functions for action LEFT (blue) and RIGHT (red) in each stimulus conditions for different noise values. Grey bars represent empirical response rates estimated from data. **b)** Intensity function of VISp for different stimulus condition and noise values. Note that since VISp is recorded in the left hemisphere, it will be contralateral to the right stimulus (contra; left panel) and ipsilateral to the left stimulus (ipsi; right panel). Grey area represents firing rates for VISp. **c)** Intensity function of MOS for different stimulus inputs and noise values. Dashed lines represent $\eta_n = 0$. Grey area represents empirical firing rates for MOS.

layer (dimensionality of $\mathbf{h}_n$ was 10) with two layers in each of the components shown in Figure 1. $\alpha^{(\text{B})} = 0.01$, $\alpha^{(\text{N})} = 1$. Learning rate was $1e-4$. See Supplementary Materials for more details.

5.3 Experiments

After training the model, we froze the weights and performed simulations in different conditions in order to investigate the neural and behavioural properties of the model. The dashed lines in Figure 3 illustrate these for the case without input noise ($\eta_n = 0$) for cases in which the higher contrast is on the left or right (in their respective columns).

Figure 3a shows, consistent with the data [Steinmetz et al., 2019], that the learned action intensities are compatible with the stimulus contrasts, meaning that animals mainly performed the correct actions (the neural recording sessions were conducted in well trained animals). Grey bars depict empirical response rates estimated from data, which are closely related to the learned intensities.

These action intensities are mediated by neural activities. Figure 3b;c show the intensity functions in two example brain regions: one visual (VISp; primary visual area), which is reported to have the highest portion of visual encoding neurons, and the other in the forebrain (MOS; secondary motor area) as a region which is reported to encode choice related information [Steinmetz et al., 2019]. The activities were all recorded in the left hemisphere, and therefore, the right stimulus/action are contralateral to the recording sites (denoted by contra in the graph), and left stimulus/action are ipsilateral (denoted by ipsi in the graph). As the figure shows, we generally see more activity on the contralateral side in VISp (and to a lesser extent in MOS), which is consistent with previous analyses [Steinmetz et al., 2019]. The relative timing of the activities in the areas are also consistent with expectations. The grey shaded areas in the panels show estimated firing rates, which are consistent with the data in particular for VISp which had a higher peak firing rate (see below).

Given that the model captures signal correlations, we turned to consider variation across externally equivalent trials through the medium of η_n. First, Figure 2b shows the histogram of the values across different trials (η_n). These have generally been dragged away from their starting value of 1 implying that they have been learned/updated in the process of model fitting.

The solid lines in Figure 3 show the behavioural and neural influence of η_n by synthesizing both whilst changing η_n. Figure 3a shows that η_n captures the common mode of variation in the timing of action; in this case, higher values are related to later responses. There is also an asymmetry between how η_n interacts with LEFT and RIGHT actions (although they were similar at the baseline $\eta_n = 0$), which could be because the model required to characterise RIGHT with a higher precision to explain relevant neural activities (which were contralateral to the RIGHT action).

The neural activities shown in Figure 3b bear out this interpretation of η_n. As the figure shows, sooner responses (with higher probabilities) are strongly related to higher peak intensities in VISp, with a small variation in the peak time compared to the action time. A similar pattern is observed in lateral visual area (VISl), which had the second highest portion of visual encoding neurons (see Figure S1 in Supplementary Materials). Figure 3c shows that similarly to VISp, the peak of the intensities for MOS at different noise values are related to response time, however, the peak times closely follow the action times, which suggests the role of this region in choice encoding as reported before [Steinmetz et al., 2019], but here we were also able to tie the variations to the behavioural response. Similar results were obtained using other forebrain regions, prelimbic (PL) and primary motor area (MOp; Figure S2 in Supplementary Materials). Note that the shown empirical firing rates are averaged across all responses, while the intensities correspond to different response times, which helps explain the differences between the two measures. For example, in very fast responses, there is an initial burst in the firing rates, which is captured by the initial sharp rise in the intensities seen in MOS (panel c), consistent with the data (see Figure S3), but this is invisible in the averaged firing rates.

6 Discussion

We presented a novel framework for linking neural spike trains to sensory inputs and behaviour. The framework extended previous works on fMRI data [Dezfouli et al., 2018] by using flexible Point process frameworks. The model was able to automatically learn suitable encoding of stimulus and noise representations that could provide a joint explanation for both behavioural and neural that could be used to recover correlational links between neural and behavioural data.

There are many important directions for future work: capturing richer aspects of behaviour that are known to couple to neural activity [Balleine and O'doherty, 2010]; integrating and/or substituting spiking activity with calcium imaging; like auto-regressive linear-nonlinear-Poisson (LNP) models [Chichilnisky, 2001], incorporating information about past spiking into $\mathbf{h}_n$ to allow full noise correlations to be modeled; using a higher dimension η_n; differentiating more finely the activity in different regions (including neurons with opposite stimulus coding). Nevertheless, we suggest that our method casts brain and behaviour interactions in a compelling new light.

Broader Impact

Developing links between brain and behavioural can ultimately have benefit in the area of mental health and behavioural change. The failure of the framework can lead to wrong predictions, which might affect practitioners in the area of experimental neuroscience.

References

James J Jun, Nicholas A Steinmetz, Joshua H Siegle, Daniel J Denman, Marius Bauza, Brian Barbarits, Albert K Lee, Costas A Anastassiou, Alexandru Andrei, Çağatay Aydın, et al. Fully integrated silicon probes for high-density recording of neural activity. *Nature*, 551(7679):232–236, 2017.

Liam Paninski, Jonathan Pillow, and Jeremy Lewi. Statistical models for neural encoding, decoding, and optimal stimulus design. *Progress in brain research*, 165:493–507, 2007.

Valerio Mante, David Sussillo, Krishna V Shenoy, and William T Newsome. Context-dependent computation by recurrent dynamics in prefrontal cortex. *nature*, 503(7474):78–84, 2013.

Surya Ganguli and Haim Sompolinsky. Compressed sensing, sparsity, and dimensionality in neuronal information processing and data analysis. *Annual review of neuroscience*, 35:485–508, 2012.

Robert E Kass, Uri T Eden, and Emery N Brown. *Analysis of neural data*, volume 491. Springer, 2014.

Arne F Meyer, Ross S Williamson, Jennifer F Linden, and Maneesh Sahani. Models of neuronal stimulus-response functions: elaboration, estimation, and evaluation. *Frontiers in systems neuroscience*, 10:109, 2017.

Adam Kohn, Ruben Coen-Cagli, Ingmar Kanitscheider, and Alexandre Pouget. Correlations and neuronal population information. *Annual review of neuroscience*, 39:237–256, 2016.

Johnatan Aljadeff, Benjamin J Lansdell, Adrienne L Fairhall, and David Kleinfeld. Analysis of neuronal spike trains, deconstructed. *Neuron*, 91(2):221–259, 2016.

Nicholas A Steinmetz, Peter Zatka-Haas, Matteo Carandini, and Kenneth D Harris. Distributed coding of choice, action and engagement across the mouse brain. *Nature*, 576(7786):266–273, 2019.

David Sussillo. Neural circuits as computational dynamical systems. *Current opinion in neurobiology*, 25:156–163, 2014.

Blake A Richards, Timothy P Lillicrap, Philippe Beaudoin, Yoshua Bengio, Rafal Bogacz, Amelia Christensen, Claudia Clopath, Rui Ponte Costa, Archy de Berker, Surya Ganguli, et al. A deep learning framework for neuroscience. *Nature neuroscience*, 22(11):1761–1770, 2019.

Chethan Pandarinath, Daniel J O'Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D Stavisky, Jonathan C Kao, Eric M Trautmann, Matthew T Kaufman, Stephen I Ryu, Leigh R Hochberg, et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature methods*, 15(10):805–815, 2018.

Krishna V Shenoy, Maneesh Sahani, and Mark M Churchland. Cortical control of arm movements: a dynamical systems perspective. *Annual review of neuroscience*, 36:337–359, 2013.

Scott D Kennedy and Andrew B Schwartz. Distributed processing of movement signaling. *Proceedings of the National Academy of Sciences*, 116(52):26266–26273, 2019.

Amir Dezfouli, Richard Morris, Fabio T Ramos, Peter Dayan, and Bernard Balleine. Integrated accounts of behavioral and neuroimaging data using flexible recurrent neural network models. In *Advances in Neural Information Processing Systems*, pages 4228–4237, 2018.

Takahiro Omi, Kazuyuki Aihara, et al. Fully neural network based model for general temporal point processes. In *Advances in Neural Information Processing Systems*, pages 2120–2129, 2019.

D.J. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes: Volume I: Elementary Theory and Methods*. Probability and Its Applications. Springer New York, 2006. ISBN 9780387215648. URL `https://books.google.com.au/books?id=6Sv4BwAAQBAJ`.

Joseph Sill. Monotonic networks. In *Advances in neural information processing systems*, pages 661–667, 1998.

Pawel Chilinski and Ricardo Silva. Neural likelihoods via cumulative distribution functions. *arXiv preprint arXiv:1811.00974*, 2018.

Yiyi Liao, Sarath Kodagoda, Yue Wang, Lei Shi, and Yong Liu. Understand scene categories by objects: A semantic regularized scene classifier using convolutional neural networks. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 2318–2325. IEEE, 2016.

Iasonas Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6129–6138, 2017.

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

Jonathan W Pillow, Jonathon Shlens, Liam Paninski, Alexander Sher, Alan M Litke, EJ Chichilnisky, and Eero P Simoncelli. Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature*, 454(7207):995–999, 2008.

Panos Z Marmarelis and Ken-Ichi Naka. White-noise analysis of a neuron chain: an application of the wiener theory. *Science*, 175(4027):1276–1278, 1972.

R De Ruyter Van Steveninck and William Bialek. Real-time performance of a movement-sensitive neuron in the blowfly visual system: coding and information transfer in short spike sequences. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, pages 379–414, 1988.

Wilson Truccolo, Uri T Eden, Matthew R Fellows, John P Donoghue, and Emery 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.

Liam Paninski. Maximum likelihood estimation of cascade point-process neural encoding models. *Network: Computation in Neural Systems*, 15(4):243–262, 2004.

John P Cunningham and M Yu Byron. Dimensionality reduction for large-scale neural recordings. *Nature neuroscience*, 17(11):1500, 2014.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Bernard W Balleine and John P O'doherty. Human and rodent homologies in action control: corticostriatal determinants of goal-directed and habitual action. *Neuropsychopharmacology*, 35(1):48–69, 2010.

EJ Chichilnisky. A simple white noise analysis of neuronal light responses. *Network: Computation in Neural Systems*, 12(2):199–213, 2001.

Neural network Poisson Models for Behavioural and Neural Spike Train Data

Supplementary Materials

Amir Dezfouli, Richard Nock, Ehsan Arabzadeh, Peter Dayan

S1 Model architecture

In this section we explain the details of the model architecture.

The trial-level noise η_n is first passed through a drop-out layer (%90 rate) and then through a dense layer with 10 units (SOFTPLUS activation). Parallel to this, the input stimuli $\mathbf{x}_n$ is passed through a dense layer with 10 units (SOFTPLUS activation). The outputs of these two dense layers are summed together and the result is passed through a dense layer with 10 units (SOFTPLUS activation) followed by a linear dense layer with 10 units to build stimulus embedding $\mathbf{h}_n$. There is no constraint on the weights in these layers. Note that η_n is a variable which is learned from the data.

The elapsed time t (since the stimulus) is first passed through a linear layer (with 10 units) with the weights constrained to be non-negative and the result is summed with $\mathbf{h}_n$ and passed through a TANH activation function. The output is passed through a dense layer with 10 units and TANH activation function and then through another dense layer with 42 units and SOFTPLUS activation function. The weights of these layers are all constrained to be non-negative. The final output of this layer corresponds to $\Lambda_u^{(\text{N})}(t; \mathbf{h}_n)$ for $u = 1 \ldots 42$ (we refer to these as neural output).

The neural output ($\Lambda_u^{(\text{N})}(t; \mathbf{h}_n)$) is then passed through a dense layer with 10 units (TANH activation) and then through another dense layer with 1 unit (SOFTPLUS activation) to produce ${\Lambda_{\text{RIGHT}}}^{(\text{B})} t; \mathbf{h}_n$. The weights are all constrained to be non-negative.

The neural output ($\Lambda_u^{(\text{N})}(t; \mathbf{h}_n)$) is also passed through a dense layer with 10 units (TANH activation) and then through another dense layer with 1 unit (SOFTPLUS activation) to produce ${\Lambda_{\text{RIGHT}}}^{(\text{B})}(t; \mathbf{h}_n)$. The weights of all the layers are constrained to be non-negative.

Note that the path from t to neural and behavioural outputs only contains positive weights which is a sufficient condition to guarantee that the outputs (neural and behavioural) are monotonic functions of t [Sill, 1998, Chilinski and Silva, 2018, Omi et al., 2019].

The learning rate for training was 0.0001 and $\alpha^{(\text{B})} = 0.01$ and $\alpha^{(\text{N})} = 1$. To make the training scalable, we used stochastic gradient descent and optimised the model using the data of each region separately in each optimisation iteration (iterating over the 42 regions).

The above architecture was the optimal one among different configurations that were considered. These included $\alpha^{(\text{B})} \in \{0.1, 0.01\}$ (keeping $\alpha^{(\text{N})}$ fixed at 1), learning rate $\in \{0.001, 0.0001\}$, and using two/four dense layers in each component of the architecture.

S2 Calculation of empirical rates

For the empirical firing rates, we divided the observation period into 50 equally-sized periods (each 0.008s width). Then, we calculated the total number of spikes in each period and normalized that by the total number of neurons and trials contributed to the spike set and also by the period duration (0.008s). Note that only spikes which were made *before* the response were included in the analysis.

For the empirical behavioural rates, we divided the observation period into 18 equally-sized periods (each $\sim$0.02 width), and calculated the total number of responses made in each period, normalised by the trials contributed to each period and also by the period duration ($\sim$0.02 width).

References

Joseph Sill. Monotonic networks. In *Advances in neural information processing systems*, pages 661–667, 1998.

Pawel Chilinski and Ricardo Silva. Neural likelihoods via cumulative distribution functions. *arXiv preprint arXiv:1811.00974*, 2018.

Takahiro Omi, Kazuyuki Aihara, et al. Fully neural network based model for general temporal point processes. In *Advances in Neural Information Processing Systems*, pages 2120–2129, 2019.

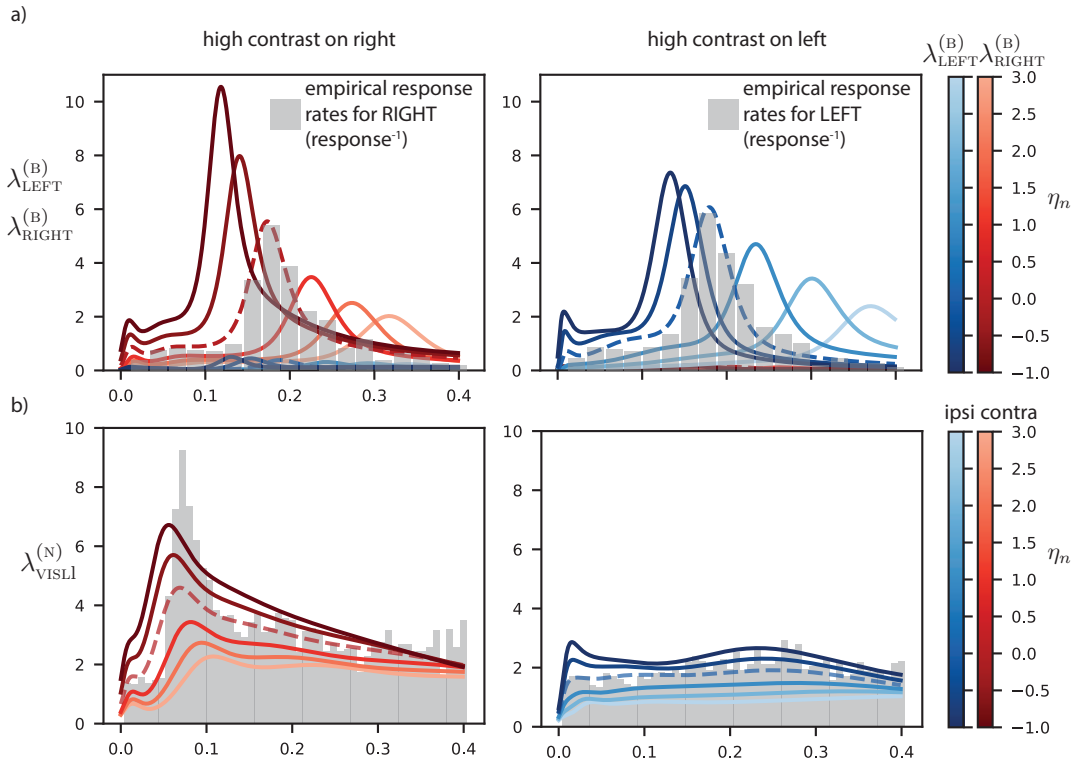

Figure S1: **Linking behavioural and neural processes using noise correlations.** In the left panels, the stimulus input to the model represented a high contrast on the right side and zero contrast on the left side ('high contrast on right'). In the right panels, the stimulus input to the model indicated zero contrast on left and high contrast on right ('high contrast on left'). **a)** Behavioural intensity functions for action LEFT (blue) and RIGHT (red) in each stimulus conditions for different noise values. **b)** Intensity function of VISl for different stimulus condition and noise values. Note that since VISl is recorded in the left hemisphere, it will be contralateral to the right stimulus (contra; left panel) and ipsilateral to the left stimulus (ipsi; right panel).

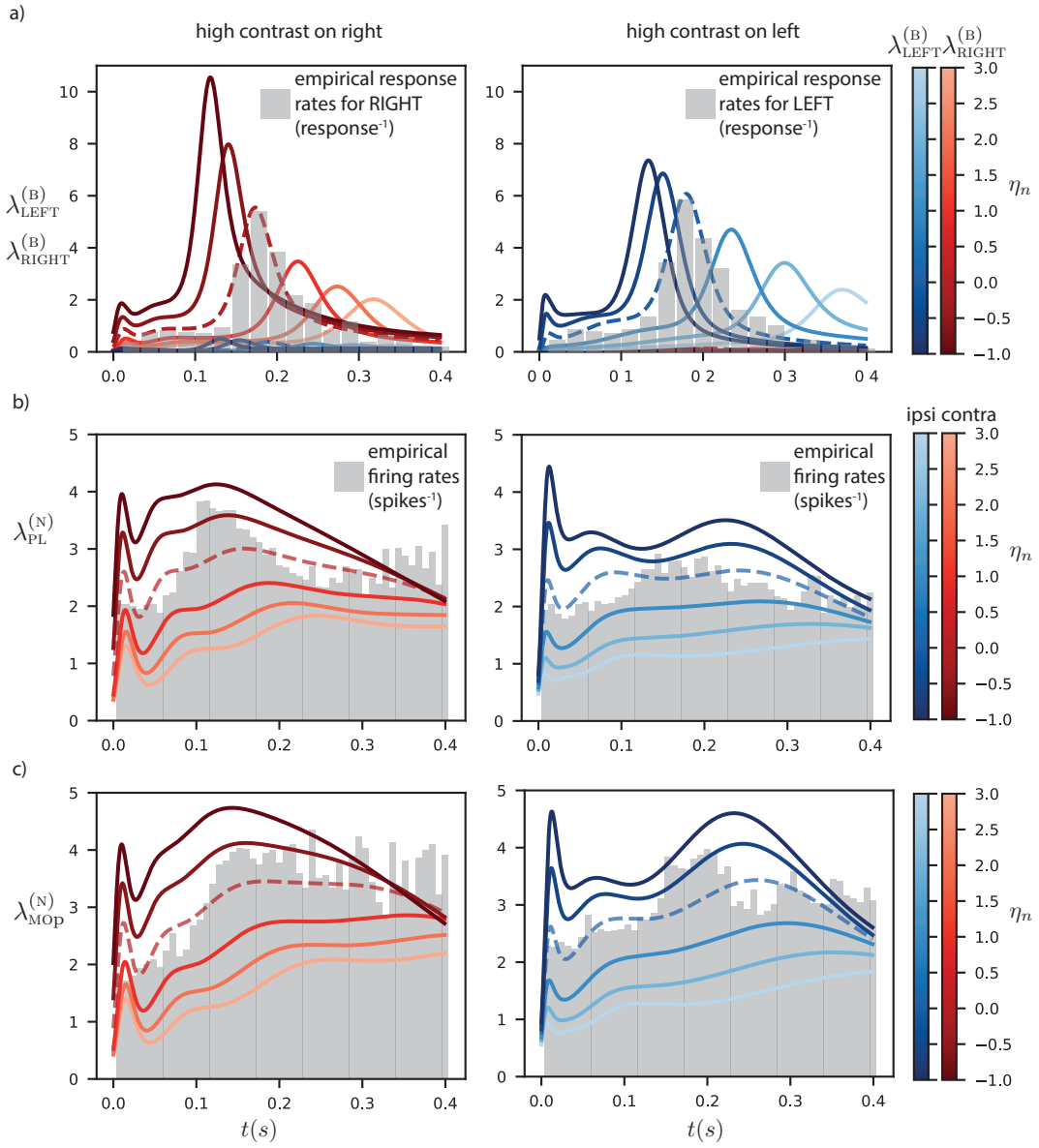

Figure S2: **Linking behavioural and neural processes using noise correlations.** In the left panels, the stimulus input to the model represented a high contrast on the right side and zero contrast on the left side ('high contrast on right'). In the right panels, the stimulus input to the model indicated zero contrast on left and high contrast on right ('high contrast on left'). **a)** Behavioural intensity functions for action LEFT (blue) and RIGHT (red) in each stimulus conditions for different noise values. **b)** Intensity function of PL for different stimulus condition and noise values. Note that since PL is recorded in the left hemisphere, it will be contralateral to the right stimulus (contra; left panel) and ipsilateral to the left stimulus (ipsi; right panel). **c)** Intensity function of MOp for different stimulus inputs and noise values. Dashed lines represent $\eta_n = 0$.

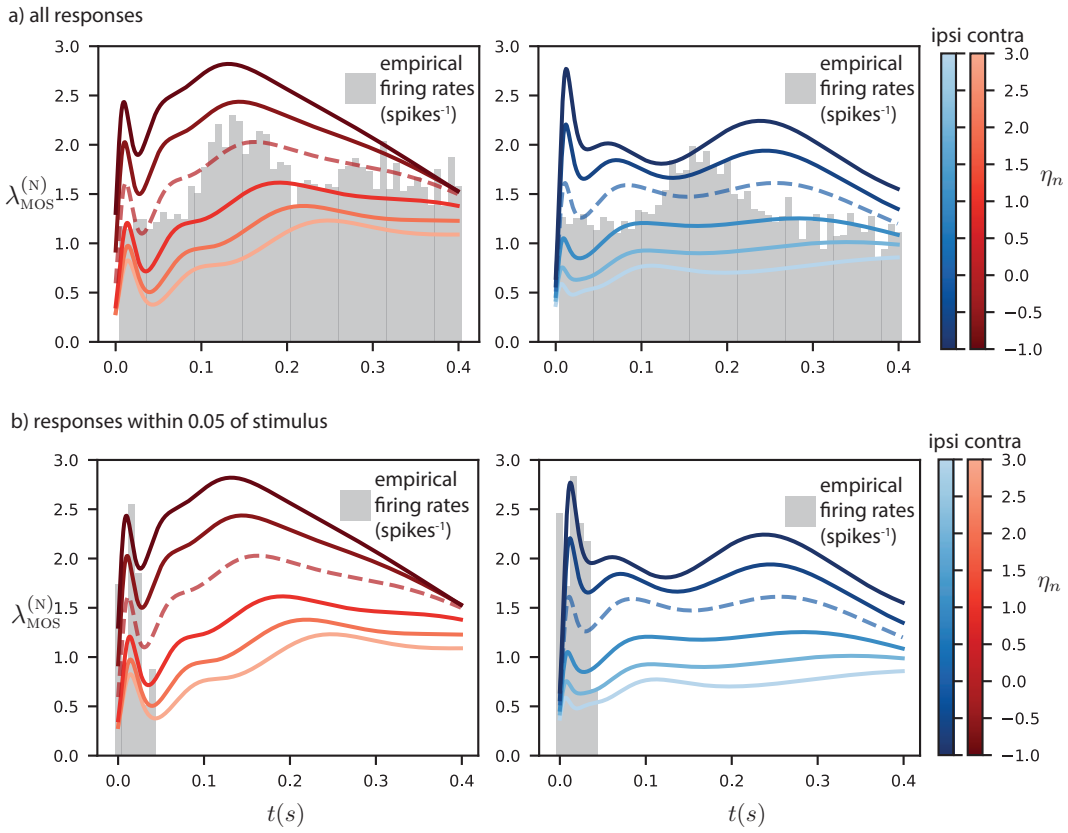

Figure S3: **Linking behavioural and neural processes using noise correlations.** In the left panels, the stimulus input to the model represented a high contrast on the right side and zero contrast on the left side ('high contrast on right'). In the right panels, the stimulus input to the model indicated zero contrast on left and high contrast on right ('high contrast on left'). **a)** Intensity function of MOS for different stimulus condition and noise values. Note that since MOS is recorded in the left hemisphere, it will be contralateral to the right stimulus (contra; left panel) and ipsilateral to the left stimulus (ipsi; right panel). Grey bars represent empirical firing rates using all the responses in the corresponding stimulus condition. **b)** Intensity function of MOS for different stimulus inputs and noise values. Empirical firing rates were calculated only for responses which were made within 0.05s of stimulus presentation. Dashed lines represent $\eta_n = 0$.