bioRxiv preprint doi: https://doi.org/10.1101/407858; this version posted September 4, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Characterizing the nonlinear structure of shared variability in cortical neuron populations using neural networks

Matthew R Whiteway[1,*], **Karolina Socha**[2,3], **Vincent Bonin**[2,3,4], **Daniel A Butts**[1,5]

Affiliations:
[1]Program in Applied Mathematics & Statistics, and Scientific Computation Program
University of Maryland, College Park, MD, United Statess
[2]Neuro-Electronics Research Flanders, Leuven, Belgium
[3]Department of Biology & Leuven Brain Institute, KU Leuven, Leuven, Belgium
[4]VIB, Leuven, Belgium
[5]Department of Biology and Program in Neuroscience and Cognitive Sciences
University of Maryland, College Park, MD, United States
[*]Current: Grossman Center, Columbia University, New York, NY, United States

Abstract

Sensory neurons often have variable responses to repeated presentations of the same stimulus. Simultaneous recordings of neural populations demonstrate that such variability is often shared across many neurons, and thus cannot be simply averaged away. Understanding the effects of this shared variability on neural coding requires an understanding of what the common drivers of variability are, and how they are related to each other and to stimulus processing. Latent variable models offer an approach for characterizing the structure of this shared variability in neural recordings, though most previous modeling approaches have either been linear or had to make very restrictive assumptions about the nonlinear structure. Here we demonstrate the use of an autoencoder neural network as a general means to fit nonlinear latent variable models for analyzing large-scale neural recordings. We first present a general nonlinear latent variable model that is agnostic to the stimulus tuning properties of the individual neurons, and is hence well suited for exploring neural populations whose tuning properties are not well characterized. This motivates a second class of model, the Generalized Affine Model, which simultaneously determines each neuron's stimulus selectivity and a set of latent variables that modulate these stimulus responses both additively and multiplicatively. Despite the general nonlinear relationships that can be uncovered with our approaches, we find that population variability in anesthetized primary visual cortex (V1) is best described by a single additive and single multiplicative latent variable. However, this is not generally the case, and to contrast the V1 results we apply the same models to recordings in awake macaque prefrontal cortex, where more general nonlinearities are needed to compactly describe the population response variability. These results thus demonstrate how nonlinear latent variable models can be used to describe population variability, and suggest that a range of methods is necessary to study different brain regions under different experimental conditions.

bioRxiv preprint doi: https://doi.org/10.1101/407858; this version posted September 4, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

1 Introduction

The activity of sensory cortical neurons is highly variable in response to repeated presentations of the same stimulus [1, 2]. This response variability has the potential to limit the reliability of neural computations, since multiple stimuli could in principle elicit the same number of spikes, and has thus motivated both experimental and theoretical research into understanding the structure and implications of this variability. Data from a wide variety of recording modalities increasingly reveal that this variability is not solely due to noisy biological processes that are independent to each neuron [3], but that a large portion of this variability is actually shared among large populations of neurons [4, 5, 6, 7, 8, 9, 10, 11, 12].

These findings raise questions about the role of shared variability in neural computations, and adequately addressing those questions requires a deeper understanding of the statistical structure of this variability in experimentally recorded neural populations. To this end, shared variability in large-scale neural recordings has been modeled using latent variable methods, where a small number of factors, or latent variables, drive neural activity across the entire population [13]. Until recently, however, most latent variable methods assumed a linear mathematical form, where the activity of each latent variable contributes to neural responses independently of the other latent variables. Such methods are popular because they are easy to apply and interpret, and include techniques such as Principal Component Analysis and Factor Analysis. If there is a complicated nonlinear interaction between latent variables in the system under study, linear methods can still capture much of the variability by adding more independent factors. However, the larger number of resulting latent variables will mix the activity of the true latent variables with approximations of their nonlinear interactions, and be difficult to interpret. In contrast, if the form of the nonlinear interaction were known and incorporated into a model, the observed neural activity could in principle be described more compactly, and would lead to greater insight into what the latent variables are related to. For example, a common form of nonlinear latent variable model has positive "gain-like" latent variables that multiplicatively modulate the response to the stimulus, and has been related to attention and arousal [5, 10]. However, in general the development of these more complex models has been limited due to the difficulty of determining the model parameters in the nonlinear setting.

Much of the recent work analyzing the structure of shared variability in large neural populations (both linear and nonlinear) has been performed on activity recorded from primary visual cortex (V1), where sensory-driven neural responses to drifting gratings are relatively well understood. In this context, shared variability in V1 responses has been modeled using a single additive latent variable that modulates neural activity across the entire population [6, 11], and subsequently modeled using a multiplicative latent variable that acts as a gain on the stimulus-driven response [5]. More recently, the "affine" model of Lin et al. [8] - and the related multi-gain model of Arandia-Romero et al. [12] - combined both additive and multiplicative latent variables to account for neural variability, which described more of the population response than either latent variable alone. However, in all cases restrictive assumptions were necessary to fit these models, and as a result, it is currently unclear whether any of the proposed models are the best model of population response variability in V1, or simply the best under these assumptions. For example, it is possible that another form of nonlinear interaction between the stimulus response and latent variables would be more appropriate than a multiplicative gain, or that multiple gain terms should be included [10].

To address these questions, we introduce a general nonlinear latent variable modeling framework that uses autoencoder neural networks to infer latent variables. We first show how the autoencoder can be used as a general nonlinear latent variable model, which models arbitrary nonlinear interactions between the latent variables. We then show how this framework can be modified to include the affine model, and extended to create a generalized version of the affine model with an arbitrary number of additive and multiplicative latent variables. By fitting these two classes of nonlinear latent variable models to population recordings from anesthetized monkey V1, we find that the population response variability is surprisingly well described by the affine model, where one multiplicative and one additive latent variable can generally capture most of the variance of the more general models. We also fit these models to population recordings from awake macaque prefrontal cortex (PFC), and find that, unlike anesthetized V1, the affine model is not sufficient to capture most of the shared variability. Thus, this novel approach to fitting nonlinear latent variable models provides a new means to compare constrained latent variable models to more general models of population response variability, and takes a step towards elucidating the nonlinear computations that underlie neural activity.

bioRxiv preprint doi: https://doi.org/10.1101/407858; this version posted September 4, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

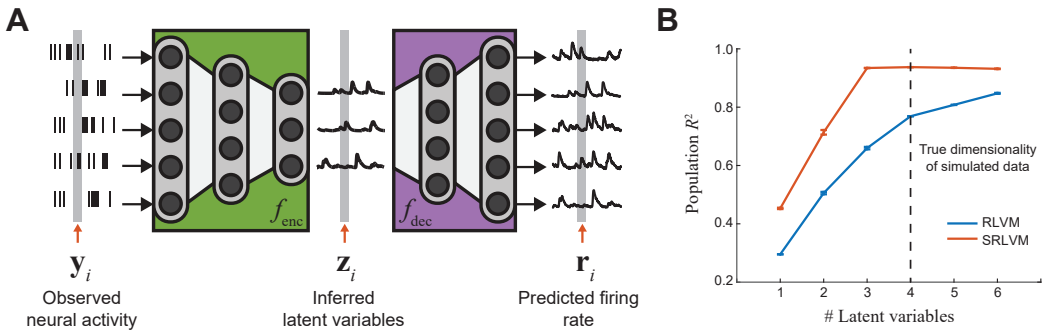

Figure 1: **The structure of the Stacked Rectified Latent Variable Model (SRLVM). A**: The SRLVM is an autoencoder neural network that predicts the population activity $\mathbf{r}_i$ using the measured population activity $\mathbf{y}_i$. The high-dimensional measured population activity is transformed into a lower-dimensional set of latent variables $\mathbf{z}_i$ using an encoding function f_{enc}, which is learned from the data with a neural network (*green box*). A decoding function f_{dec} is simultaneously learned with a separate neural network (*purple box*) to transform the latent variables into the predicted population activity. The RLVM models f_{enc} and f_{dec} as single-layer neural networks, whereas the SRLVM models f_{enc} and f_{dec} as multi-layer neural networks, allowing for more complex, nonlinear transformations of the data. All neural network layers are fully connected. **B**: Fits of the RLVM and SRLVM to simulated data generated using a nonlinear combination of four latent variables. The SRLVM is able to capture most of the variance using four latent variables, whereas the RLVM needs more latent variables to capture the effect of the nonlinear interactions. Error bars are standard error of the mean over ten cross-validation folds.

2 Results

2.1 A general framework for fitting nonlinear latent variable models

Latent variable models aim to describe the high-dimensional activity of many neurons using a low-dimensional set of latent variables. A linear latent variable model predicts the activity of each neuron as a linear combination of the latent variables, so that the vector of predicted neural responses $\mathbf{r}_i$ on trial i is given by

$$\mathbf{r}_i = W\mathbf{z}_i + \mathbf{b} \tag{1}$$

where $\mathbf{z}_i$ is the vector of latent variables, W is a matrix mapping the latent variables to the predicted responses, and $\mathbf{b}$ is a vector of biases. Linear latent variable models such as Principal Component Analysis (PCA) and Factor Analysis (FA) – and variations thereof (e.g. dPCA [14]) – are finding widespread use in neuroscience due to the ease of fitting these models to neural data, their interpretability, and their flexibility [13]. However, relationships between latent variables and the activity of different neurons can be highly nonlinear due to the discrete nature of the spike generation process (among other factors), and linear latent variable models are often used to analyze neural data only after activity has been averaged over similar trials. Thus, while linear approaches have provided insights about the nature of neural computations during particular tasks (e.g. [15, 16]), a deeper understanding of how neural activity is structured during single trials, for example to study neural activity during decision-making tasks or during learning, will require nonlinear latent variable methods.

A nonlinear latent variable model predicts the activity of each neuron as some nonlinear transformation f_{dec} of the latent variables (f_{dec} "decodes" the latent variables), so that the vector of predicted responses is given by

$$\mathbf{r}_i = f_{\text{dec}}(\mathbf{z}_i) \tag{2}$$

The two challenges in fitting nonlinear latent variable models are (1) defining and fitting the nonlinear function f_{dec}; and (2) inferring the latent variables $\mathbf{z}_i$, which will depend on the function f_{dec}. We tackle both of these challenges simultaneously by using an autoencoder neural network (Fig. 1A). The arbitrary nonlinear function f_{dec} in equation 2 is modeled with a multi-layer neural network, which can in principle approximate any high-dimensional nonlinear function [17]. To produce the latent variables $\mathbf{z}_i$, the autoencoder transforms the vector of observed spike counts $\mathbf{y}_i$ into a low-dimensional

bioRxiv preprint doi: https://doi.org/10.1101/407858; this version posted September 4, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

representation using another arbitrary nonlinear function f_{enc} (which "encodes" the neural activity into the latent variables), also modeled with a multi-layer neural network:

$$\mathbf{z}_i = f_{\text{enc}}(\mathbf{y}_i) \tag{3}$$

The parameters of the model - the weights and biases of each neural network layer in f_{enc} and f_{dec} - can then be learned from experimental data by maximizing the log-likelihood of the model predictions $\mathbf{r}_i$ under an appropriate noise distribution (see Methods), and the value of the latent variables on each trial can be computed using equation 3.

If the latent variable model defined in equations 2 and 3 is restricted so that f_{enc} and f_{dec} are simple affine transformations (a linear transformation with an additional bias term), and the $\mathbf{z}_i$ are constrained to be nonnegative through the use of pointwise nonlinearities in the hidden layer, the resulting model is the Rectified Latent Variable Model (RLVM) of Whiteway & Butts [18]. We then denote the more general, nonlinear latent variable model, where f_{enc} and f_{dec} are multi-layer neural networks, as the "Stacked" RLVM, or SRLVM.

One benefit of the added complexity of the SRLVM is that it can recover nonlinear, low-dimensional structure more parsimoniously than the RLVM. To illustrate this point, we generated the activity of a population of 50 neurons using a nonlinear combination of just four latent variables (see Methods), and fit both models to the data (Fig. 1B). The performance of the RLVM increases as more latent variables are added, even past four (the true dimensionality of the data). This behavior arises because the inferred latent variables of the RLVM must account not only for the activity of the true latent variables, but for the nonlinear interactions between them as well. The SRLVM, on the other hand, is able to model those nonlinear interactions with a neural network. As a consequence, it is able to explain more of the population variability than the RLVM without the additional latent variables. The SRLVM is thus able to fully and compactly capture the nonlinear subspace, aiding in further analysis and visualization.

2.2 Nonlinear models better describe V1 population response variability

In order to compare the ability of the RLVM and SRLVM to predict neural responses in experimental data, we fit these models to large population recordings made publicly available by the Kohn Lab [19]. This dataset contains Utah array recordings from primary visual cortex (V1) of three anesthetized macaques (Monkeys 1-3 in Figs. 2, 4, 6) in response to drifting gratings in 12 equally spaced directions (see Methods for details). We chose this dataset because previous work studying trial-to-trial variability in V1 has shown that a significant amount of this variability is shared among neurons, and has been observed in the context of noise correlations between pairs of neurons [20], and using latent variable approaches [6, 5, 11, 8, 12].

The first question that we asked is how many linear latent variables are necessary to describe the variability in the data. Some of the overall variability in spike counts is due to variability in the average responses across stimuli (i.e. a non-flat tuning curve), and the remainder is defined as the trial-to-trial variability (Fig. 2A, B). To determine how much of the trial-to-trial variability our models explain, instead of using R^2 as a measure for goodness of fit we adopt the Quality Index (QI) introduced in Lin et al. The QI is a scaled version of R^2 (see Methods), such that QI = 0 for models that describe as much variability as the tuning curve, and QI = 1 for perfect prediction (Fig. 2C). We found that the RLVM was able to explain a substantial portion of the trial-to-trial variability using a small number latent variables (Fig. 2D, *blue curves*). These results are similar to what we would find with PCA, since the two models share the same cost function [18]. Though the RLVM also incorporates regularization terms which can in principle help with overfitting, we did not find this to be much of a problem given the large number of trials in these datasets (2400 per session).

Despite the good performance of the RLVM, this model will not provide the most compact description of population responses if the data contain nonlinear interactions. To demonstrate this, we then fit the SRLVM. As expected, given the success previous nonlinear approaches, the cross-validated performance of the SRLVM is better than that of the RLVM for a given number of latent variables (Fig. 1D). Similar to the example simulation shown in Figure 1B, the nonlinear latent variables not only offer a more compact description of population response variability, but are able to explain more of that variability as well.

However, one limitation of the SRLVM is that it is agnostic to the stimulus tuning of individual neurons, and as a result the inferred nonlinear latent variables will combine the stimulus response

bioRxiv preprint doi: https://doi.org/10.1101/407858; this version posted September 4, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

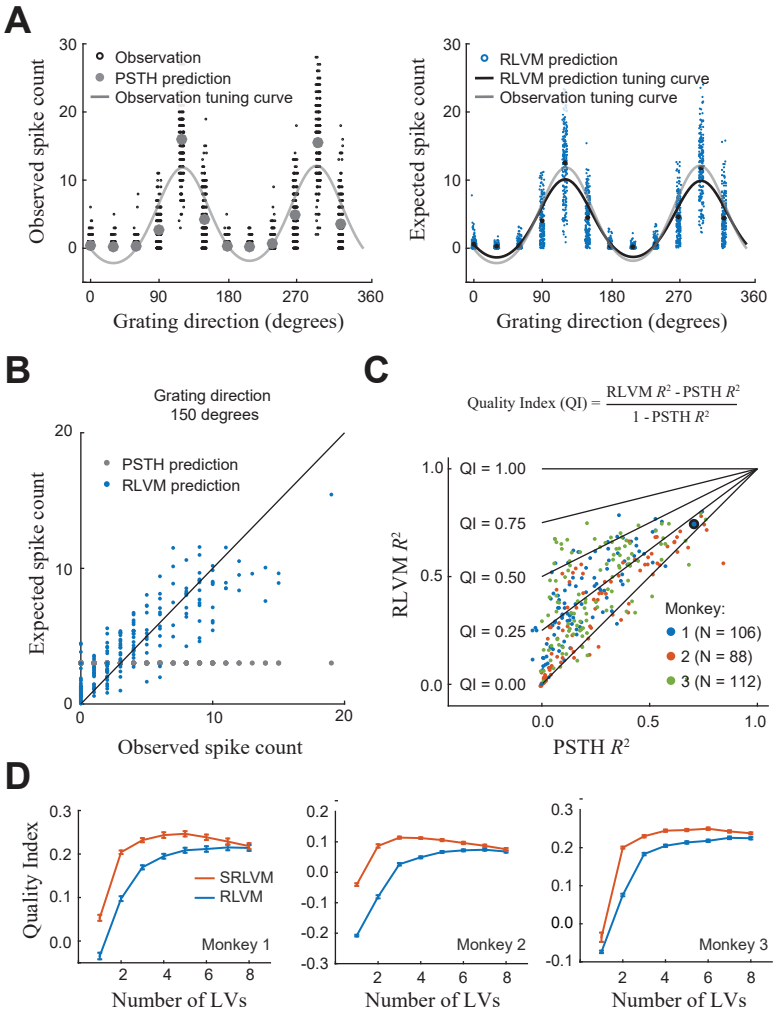

Figure 2: **Performance of the RLVM and SRLVM on population data from anesthetized macaque V1.** **A**: An illustration of the trial-to-trial variability in an example neuron. *Left*: The observed spike counts (*black dots*) versus grating direction, along with the expected spike counts from the PSTH (*gray dots*). The PSTH predictions are not able to capture the large amount of trial-to-trial variability. *Right*: The expected spike counts from the RLVM (with four latent variables; *blue dots*), which display much more variability. Points have been horizontally jittered in both plots for clarity. **B**. The expected spike count versus observed spike count for a single grating direction, demonstrating that the trial-to-trial variability in the observations is well captured by the RLVM. **C**: The Quality Index (QI) compares the predictive ability of a given model (here, the RLVM) to that of the mean stimulus response (PSTH). A high QI results when the model captures a much larger proportion of the variance than the PSTH. A low QI can either result from a poor model, or when the model explains little more than the PSTH. The RLVM R^2 versus the PSTH R^2 demonstrates the ability of the RLVM to capture trial-to-trial variability across all three datasets. Note that many neurons whose activity are poorly captured by the PSTH can be well predicted by the RLVM. The neuron used in **A**, **B** is outlined in black. **D**: Mean QI over neurons and cross-validation folds, plotted as a function of the number of latent variables in the RLVM (*blue line*) or SRLVM (*red line*). All datasets are best described by a relatively small number of latent variables.

bioRxiv preprint doi: https://doi.org/10.1101/407858; this version posted September 4, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

with shared variability that is not stimulus-driven. This mixing leads to issues of interpretability if we are interested in studying the relationship between stimulus tuning and sources of shared variability. The SRLVM is potentially even more difficult to interpret because it then uses a neural network to nonlinearly transform those mixed latent variables into the predicted responses. As a result, even though the SRLVM can better capture the statistical structure of the data, the resulting model yields latent variables that are difficult to relate to experimentally observable or controllable variables. Nevertheless, the superior performance of the SRLVM suggests that there are indeed nonlinearities present in the data, and as a consequence this model can be used to inform hypotheses about the nature of those nonlinearities.

2.3 A new method to fit the affine model of V1 population response variability

One way to improve interpretability of the models is to posit an explicit nonlinear interaction between the stimulus response and the latent variables. One such example of an explicit nonlinearity is multiplication, which previous models have used to capture gain-like modulation of responses to simple stimuli in V1 [5, 8, 12]. This model, which we will refer to as the "multiplicative model", describes the predicted activity of neuron n on trial i, r_i^n, in response to a drifting grating at angle theta, as

$$r_i^n = (1 + w_n g_i) f_n(\theta) \tag{4}$$

where $f_n(\theta)$ is the average stimulus response for neuron n, g_i is the multiplicative gain term that changes from trial to trial (which is shared across the entire population), and w_n is the coupling weight of neuron n to the gain term. When combined with an additive term, the resulting model describes the predicted response as

$$r_i^n = (1 + w_n g_i) f_n(\theta) + v_n h_i \tag{5}$$

where h_i is an additive offset that, like g_i, changes from trial to trial and is shared across the entire population, and v_n is the coupling weight of neuron n to this term. The existence of nonlinearities makes inference of the latent variables much more challenging, and in order to fit these models some of the previous work incorporated particular constraints. For example, Goris et al. introduced a model similar to that in equation 4, with a uniform coupling weight ($w_n = 1$) across all neurons; Lin et al. introduced the model in equation 5 as the "affine" model, and again enforced a uniform coupling to the gain term ($w_n = 1$); finally, Arandia-Romero et al. introduced the "multi-gain" model, where the w_n's and v_n's are unconstrained, but the multiplicative and additive latent variables are constrained to be equal ($g_i = h_i$).

Here we can use the autoencoder framework introduced in the previous section to fit the affine model in equation 5, imposing no constraints on either the coupling terms or the latent variables. [We will henceforth refer to the model of Lin et al. as the "constrained" affine model.] Our method uses neural networks to transform the observed spike count vector $\mathbf{y}_i$ into each of the latent variables g_i and h_i, so that

$$g_i = f_{\text{mult}}(\mathbf{y}_i) \tag{6}$$

$$h_i = f_{\text{add}}(\mathbf{y}_i) \tag{7}$$

where f_{mult} and f_{add} represent (in general) different neural networks, each with an arbitrary number of layers (Fig. 3). As with the SRLVM, all parameters – the weights and biases of the neural networks, and the coupling weights w_n, v_n, for all n - can be fit simultaneously by maximizing the log likelihood of the model (see Methods), and the value of the latent variables can be computed using equations 6 and 7.

With this ability to fit the fully unconstrained affine model, we first asked how this model compared to some of the constrained versions introduced previously in the literature. One possible outcome is that the constrained model performs equivalently to, or better than, the unconstrained model (potentially due to overfitting of the unconstrained model), a result that supports the validity of the constraints. The other possibility, then, is that the constrained model performs worse than the unconstrained one, in which case the constraints limit the model's ability to capture the full structure of the data and are not supported by the experimental observations.

We found that the unconstrained affine model outperformed both the additive (Fig. 4A, $p<5e-10$ for each of the three monkeys, two-sided sign test) and multiplicative models (Fig. 4B, $p<5e-10$ for each of the three monkeys). Furthermore, the unconstrained affine model outperformed the

bioRxiv preprint doi: https://doi.org/10.1101/407858; this version posted September 4, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

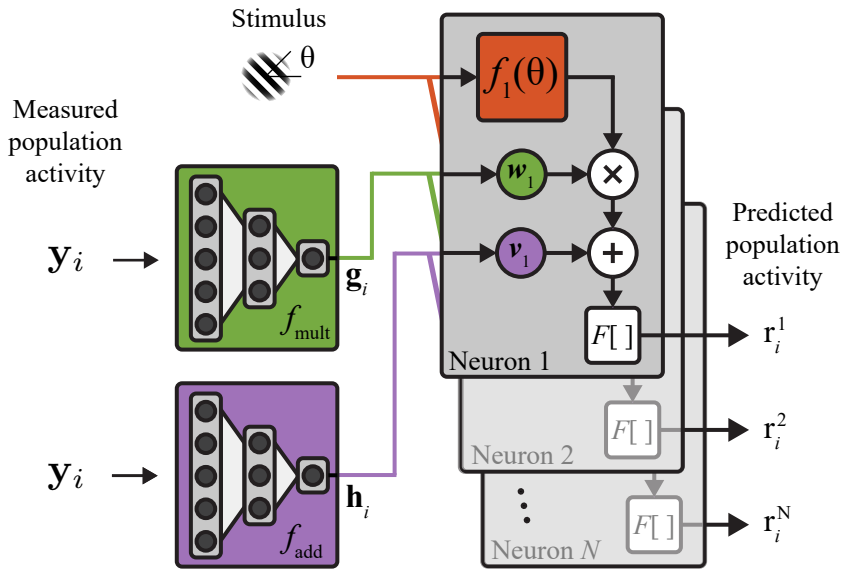

Figure 3: **The structure of the Generalized Affine Model (GAM).** The GAM is a nonlinear latent variable model that explicitly models multiplicative and additive interactions between latent variables and the stimulus response of each neuron. A neural network is used to transform the measured population activity $\mathbf{y}_i$ into multiplicative gain signals $\mathbf{g}_i$ (*green network*). These signals are shared across the entire population and modulate the output of the stimulus response for each individual neuron (*red box*) according to a set of coupling weights w_n for each neuron n. A separate neural network transforms the measured population activity into additive signals $\mathbf{h}_i$ (*purple network*), which are also shared across the population, and added to the modulated stimulus response according to a set of coupling weights v_n for each neuron n. $F[]$ is an optional spiking nonlinearity that can be applied at the output to arrive at the predicted population activity $\mathbf{r}_i$.

constrained affine model (Fig. 4C, p<5e-5 for each of the three monkeys). These results demonstrate that the more general model structure of the unconstrained affine model is indeed able to capture more structure in the data.

Given the superior performance of the unconstrained affine model, we next asked how well it compared to the SRLVM. We first looked at the SRLVM with two latent variables (SRLVM-2), the same number used by the unconstrained affine model. We found that the unconstrained affine model outperformed the SRLVM-2 (figure 4D, p<5e-10 for each of the three monkeys, two-sided sign test). This result is perhaps unsurprising, given that the SRLVM latent variables must capture both the stimulus response and the trial-to-trial variability of the neural population. However, increasing the number of SRLVM latent variables to four (after which the SRLVM begins to overfit; see Fig. 2D) results in no significant difference between the unconstrained affine model and the SRLVM (figure 4E, p>0.05 for each of the three monkeys, two-sided sign test). Therefore, we find that the two latent variables of the affine model provide a more compact description of the data than the SRLVM. This example clearly demonstrates how models incorporating an explicit form of nonlinear interaction can be compared to more general models such as the SRLVM to understand the extent to which the hypothesized nonlinearity (in this case, multiplication) is able to capture the statistical structure of the neural responses.

2.4 Prefrontal cortex is best described by many additive and multiplicative latent variables

The unconstrained affine model (hereafter referred to simply as the affine model) was able to offer a particularly parsimonious description of population response variability in anesthetized V1, even when compared to a more general model such as the SRLVM. To test the generality of this result, we used these methods to analyze a more complex dataset from macaque prefrontal cortex, made publicly available by the Kiani lab [21, 22]. This dataset contains Utah array recordings from three

bioRxiv preprint doi: https://doi.org/10.1101/407858; this version posted September 4, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

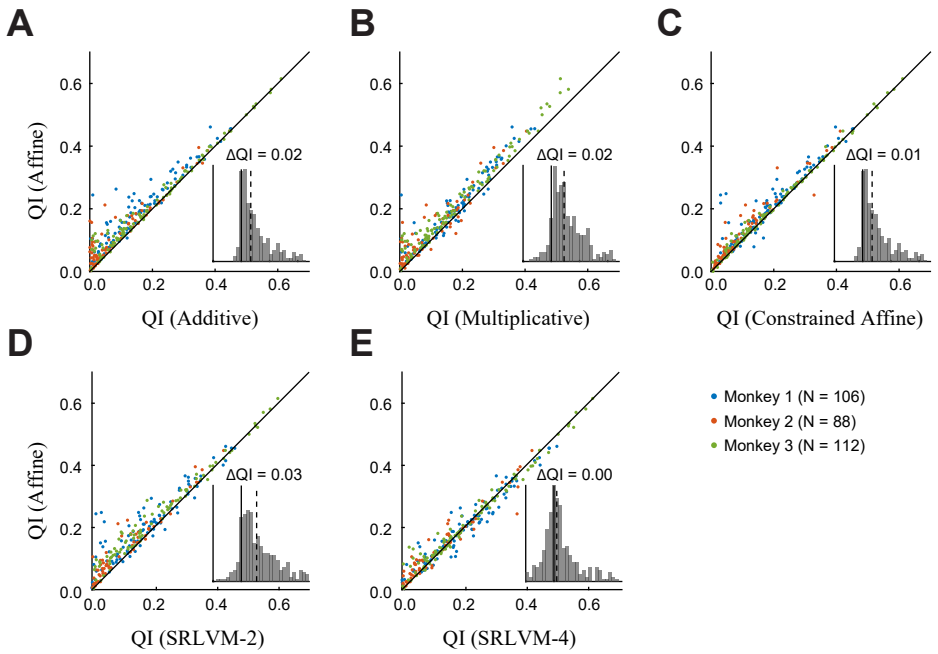

Figure 4: **Performance of additive, multiplicative and affine models in V1 data.** A comparison of the cross-validated performance of the models on individual neurons across three datasets (dataset indicated by color). Each point is the mean QI over 10 cross-validation folds. **A**: Comparison of the affine model, with one multiplicative and one additive latent variable, to the strictly additive model. *Inset*: A histogram of the difference between the QI of the affine model and the QI of the additive model (ΔQI). Zero is indicated with the solid black line, median difference across all neurons is indicated with the dashed black line. **B**: Comparison of the affine model to the strictly multiplicative model. **C**: Comparison of the affine model to the constrained affine model of Lin et al., which requires all neurons to have the same coupling weight to the multiplicative latent variable. **D**(**E**): Comparison of the affine model to the SRLVM with two (four) latent variables.

macaques performing a perceptual discrimination task with a random dot motion stimulus of fixed duration (see Methods for details).

We found that both the RLVM and SRLVM were able to explain significantly more variability than the affine model (Fig. 5A), in stark contrast to the results from V1. A further difference in these results is that the PFC data requires many more latent variables. Even if we restrict the SRLVM to two latent variables, to match the number in the affine model, there is still a large performance gain in two of the three monkeys. It is possible, however, that two additive latent variables, or two multiplicative latent variables, could capture more variability than one of each.

To address this question, we extended the affine model by allowing an arbitrary number of additive and multiplicative latent variables, which we call the Generalized Affine Model (GAM):

$$r_i^n = \left(\sum_{k=1}^{K} w_n^k g_i^k + b_n \right) f_n(\theta) + \sum_{m=1}^{M} v_n^m h_i^m \tag{8}$$

Each neuron is allowed to have its own coupling to each of the M additive latent variables, and a different coupling to each of the multiplicative latent variables; the unconstrained model is recovered from equation 8 when $M = K = 1$. Model fitting is performed in exactly the same way as before, by maximizing the log-likelihood of the model predictions.

We found, using the GAM, that a large number of additive and multiplicative latent variables were necessary to describe the PFC population response variability (Fig. 5B). However, the SRLVM was typically able to capture more variance using the same number of latent variables, especially for small numbers of latent variables (comparing, for example, three SRLVM latent variables to one additive

bioRxiv preprint doi: https://doi.org/10.1101/407858; this version posted September 4, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

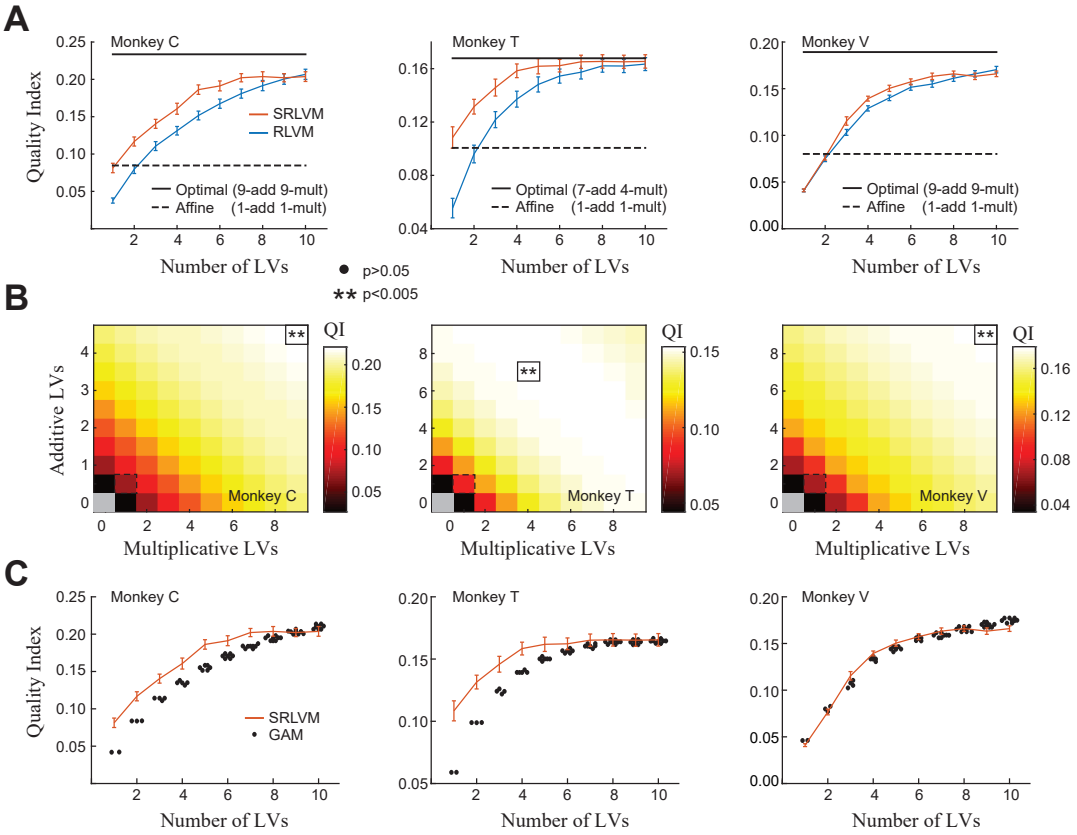

Figure 5: **Performance of GAMs on awake PFC data. A**: Performance of the affine model (*dashed black line*) and the optimal GAM (*solid black line*; see panel **B**), plotted against the RLVM and SRLVM. The affine model does not perform nearly as well as the other models. The optimal GAM performs approximately as well as the SRLVM, but uses many more latent variables. **B**: Mean QI over neurons and cross-validation folds for the GAM across a range of additive and multiplicative latent variable combinations. The affine model is outlined (*dashed black line*) as well as the GAM with the highest QI ("optimal GAM"; *solid black line*; $^{**}p < 0.005$, $\bullet > 0.05$, two-sided sign test on the mean compared to the affine model). A large number of additive and multiplicative latent variables can describe much more variability than the affine model, in contrast to the V1 data (see Fig. 6). **C**: Comparison of the SRLVM and GAM for a given number of latent variables; for example, a GAM with three latent variables could have 3/0, 2/1, 1/2 or 0/3 additive/multiplicative latent variables. Especially for small numbers of latent variables the SRLVM describes population activity better than the GAM, regardless of the latent variable combinations. GAM points are jittered to display overlapping data points, and error bars have been omitted for clarity. The SRLVM curves are the same as those in panel **A**.

and two multiplicative GAM latent variables; Fig. 5C). This result demonstrates that the SRLVM can provide a more compact description of activity than the GAM, and motivates the search for a different explicit form of nonlinear interactions in this data.

2.5 V1 remains well-described by the affine model

We were also interested in fitting the GAM to the V1 data, where we previously found that the affine model performed comparably to the more flexible SRLVM. Would the GAM be able to explain even more variance? Perhaps unsurprisingly, we found that the best cross-validated models included multiple additive and multiplicative latent variables (Fig. 6A). However, the number of additional latent variables in the optimal models was small (1-3 for both types), and the difference was not large

bioRxiv preprint doi: https://doi.org/10.1101/407858; this version posted September 4, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

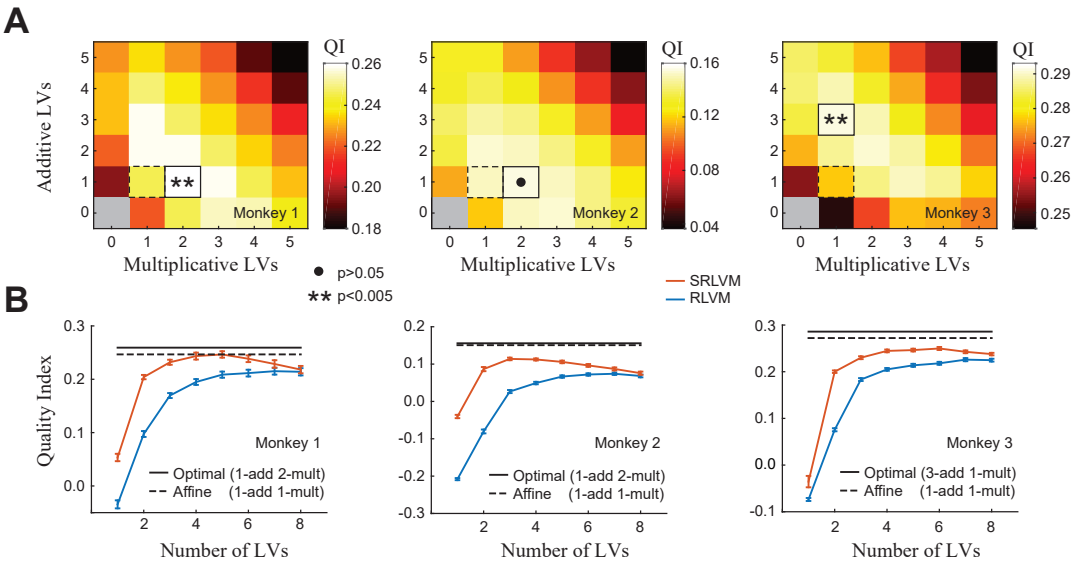

Figure 6: **Performance of GAMs on anesthetized V1 data. A**: Same conventions as Figure 5B. Only a small number of GAM latent variables are needed to describe the variability, unlike the PFC dataset (compare to Figure 5B). **B**: Performance of the affine model (*dashed black line*) and the optimal GAM (*solid black line*), plotted against the RLVM and SRLVM (same curves as Figure 2D). The affine model performs almost as well as the optimal GAM, and as well as or better than the SRLVM across all three monkeys, indicating that this model can provide a parsimonious description of the population response.

in magnitude (Fig. 6B, *solid* and *dashed black lines*). This result indicates that the affine model, with a single additive and multiplicative latent variable, is indeed a good description of population response variability in this anesthetized V1 dataset.

3 Discussion

We introduced a new framework for fitting nonlinear latent variable models to neural data, based on the autoencoder neural network, and developed two new models within this framework. The first model, the Stacked Rectified Latent Variable Model (SRLVM), is a multi-layer autoencoder neural network that we used as a general nonlinear latent variable model, which is agnostic to the stimulus tuning of the neurons and does not require an explicit form of nonlinear interaction among the latent variables (Fig. 1A). The second model, the Generalized Affine Model (GAM), extends several previous models of neural population responses [5, 6, 8, 11, 12]. The GAM allows for an arbitrary number of additive and multiplicative latent variables, which are shared across the entire population, and modulate the stimulus response of each neuron (Fig. 3). For both the SRLVM and the GAM, our model fitting framework allows us to fit all parameters and infer all latent variables simultaneously.

We applied these new models to population recordings from anesthetized macaque V1, and found that the GAM was better able to explain the variance of the population responses with fewer latent variables than the SRLVM (Fig. 6B). The optimal GAM explained only slightly more variance than an affine model with a single additive and single multiplicative latent variable (Fig. 6B), suggesting that the affine model is a good description of population response variability in this dataset. However, we found the opposite situation in awake macaque PFC, where the SRLVM provided a more compact description of the data than the GAM, and far outperformed the affine model (Fig. 5). These results demonstrate that more general, unconstrained nonlinear latent variable models like the SRLVM can provide a benchmark to test how well more structured nonlinear models (like the affine model) fit the data, and can thus aid in the generation and testing of new hypotheses about the nature of computation in populations of neurons. Furthermore, our comparison between V1 and PFC shows that a range of

bioRxiv preprint doi: https://doi.org/10.1101/407858; this version posted September 4, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

nonlinear methods will be necessary to study different brain regions under different experimental conditions.

Several other examples of nonlinear latent variable models based on autoencoder neural networks exist in the literature on neural data analysis, but differ from our models in the prior assumptions placed on the latent variables. The SRLVM and the GAM both assume that the latent variables at time t are strictly a function of the observed neural data at time t, and as such there is no notion of dynamics in our models: the data can be randomly permuted in time with no effect on the model fits. The fLDS model developed in [23] models the latent variables with a linear dynamical system (LDS) prior, such that the value of the latent variables at time t are a linear combination of the latent variables at time $t-1$. LFADS (Latent Factor Analysis via Dynamical Systems) [24] is another model based on the autoencoder, and uses a more general recurrent neural network to model the dynamics of the latent variables. However, our aim in this work was not to study dynamics (either linear or nonlinear), but to study the nonlinear transformations that take place between the latent variables and the observed neural activity. To simplify this endeavor we only modeled spike counts on a per-trial basis, and extending our work to higher time resolution (where dynamics likely play a larger role) is a direction for future investigation.

One of our main findings was that the PFC data required many more dimensions to describe neural variability than the V1 data. This result could be due to several different factors, and likely others we do not consider here. First of all, the PFC data was recorded in awake animals performing a behavioral task, unlike the V1 data. Because of this, we cannot make precise statements about how much the difference is due to the awake versus anesthetized state, and how much is due to the different computational requirements of the corresponding tasks. The PFC data was recorded during the execution of a perceptual decision making task, which intuitively places more demands on neural function than passively viewing a stimulus, and might therefore result in higher dimensional neural activity. [25] formalized this intuition and showed that, at least in the linear setting, increased task complexity does indeed result in higher-dimensional neural activity.

However, it is possible that the differing dimensionalities we observed in these datasets is not related to task complexity at all, and instead related to other non-experiment-related activity present in an awake behaving animal. Recent work has demonstrated that behavioral variability can drive a large proportion of the variability in rodent cortex, enough even to mask the task-related activity [26, 27]. It is possible that much of the variability we captured with our models in the PFC data is linked with behavior not directly related to the task (i.e. orofacial movements), which is absent in the anesthetized V1 data, although it is currently unclear to what extent activity in primate cortex is similarly dominated by behavior. Fortunately, it is straightforward to incorporate external predictors of activity such as behavior into both the SRLVM and GAM, and so both of these models can be used to study variability while controlling for non-task-related behavior.

Another advantage that our modeling framework offers (specifically the GAM) is the ability to fit the stimulus responses simultaneously with the latent variables. In all previous work that we have considered here, as well as this work, the datasets contain neural responses to low-dimensional stimuli that could be characterized with simple tuning curves. The GAM expands the range of stimulus sets that can be used since it can incorporate arbitrarily complex stimulus processing models, including Generalized Linear Models [28] and more complicated, nonlinear models like the Nonlinear Input Model [29] or convolutional neural networks [30, 31, 32, 33, 34, 35]. This ability can, for example, allow for the investigation of neural variability during tasks that include more naturalistic stimuli.

Latent variable models such as the SRLVM and the GAM will be increasingly useful for exploring the increasing number of large-scale datasets provided by multi-electrode recordings and two-photon imaging. These models, when used correctly, can capture meaningful structure in the data that can guide our understanding of the principles that underlie brain structure and function, and suggest additional experimental investigations [36]. To fully describe the richness of neural activity, these models must able to capture nonlinear relationships; to be useful, they must be easily fit to experimental data. The SRLVM and GAM fulfill both of these requirements, and are also easily extendable, allowing them to adapt to changing analysis demands as experimental neuroscience continues to provide datasets of unprecedented size and complexity.

bioRxiv preprint doi: https://doi.org/10.1101/407858; this version posted September 4, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

4 Methods

4.1 Experimental data

V1 dataset. We analyzed electrophysiology data from the Kohn Lab, which has been made publicly available at `http://dx.doi.org/10.6080/K0NC5Z4X`. Spiking activity was recorded with a Utah array in primary visual cortex from three anesthetized macaques, in response to full-contrast drifting gratings with 12 equally-spaced directions, presented for 1280 ms (200 repeats). Full details can be found in [37]. Spike counts were analyzed using a single 500 ms time bin per trial (500 ms to 1000 ms after stimulus onset). Spike counts were square-rooted before fitting the models to stabilize the variance and reduce the influence of neurons with high firing rates, as in [38].

PFC dataset. We analyzed electrophysiology data from the Kiani Lab, which has been made publicly available at `http://www.cns.nyu.edu/kianilab/Datasets.html`. Spiking activity was recorded with a Utah array in area 8Ar of the prearcuate gyrus from three macaques as they performed a direction discrimination task. On each trial the monkey was presented with a random dot motion stimulus for 800 ms, and after a variable-length delay period the monkey reported the perceived direction of motion by saccading to a target in the corresponding direction. The coherence of the dots and their direction of motion varied randomly from trial to trial. Spike counts were analyzed using a single 700 ms time bin per trial during the stimulus presentation (100 ms to 800 ms after stimulus onset). As with the V1 data, spike counts were square-rooted before fitting the models.

4.2 Simulated data

We simulated the firing rate r_t^n for each of $N = 50$ neurons as a weighted sum of three stimulus signals $\{s_t^m\}_{m=1}^3$, multiplicatively modulated by a gain signal g_t

$$r_t^n = (1 + \beta^n g_t) f(\alpha_1^n s_t^1 + \alpha_2^n s_t^2 + \alpha_3^n s_t^3) \tag{9}$$

where $f(x) = \max(0, x)$. This resulted in population activity that was fully described in a four dimensional space. The three stimulus signals and gain signal were each low-pass filtered white noise signals, and each neuron had its own unique coupling weights to the three stimulus signals (α_k^n's) and the gain signal (β^n's). The output of the simulation is a continuous-valued firing rate, rather than the resulting spiking activity or two-photon activity, in order to more clearly elucidate model behavior under ideal conditions.

4.3 Modeling details

We performed all model fitting using 10-fold cross-validation, where the data are divided into 10 equally-sized blocks, with nine used for training and one for testing, with 10 distinct testing blocks. All reported measures of model performance were calculated using testing data. We used L_2 regularization in all of our models to prevent overfitting to the training data, which required fitting a hyperparameter governing the strength of the regularization (e.g. λ in equation 12 below). For each fold, we fit the models to the training data using a range of hyperparameter values (six values logarithmically spaced between 10^{-5} and 10^0), and selected the value that resulted in the best performance on the held-out testing fold.

SRLVM. The SRLVM predicts the observed spike count vector of N neurons on trial i, $\mathbf{y}_i \in \mathbb{R}^N$, using a smaller set of latent variables $\mathbf{z}_i \in \mathbb{R}^M$, where typically $M \ll N$. The SRLVM constrains the latent variables to be some encoding function f_{enc} of the observed spike counts, so that

$$\mathbf{z}_i = f_{\text{enc}}(\mathbf{y}_i) \tag{10}$$

The population activity is then coupled to the latent variables with the decoding function f_{dec}, so that the predicted response vector $\mathbf{r}_i$ of the model is

$$\mathbf{r}_i = f_{\text{dec}}(\mathbf{z}_i) \tag{11}$$

We implemented both f_{enc} and f_{dec} with feed-forward neural networks, which used rectified linear (ReLU) activation functions $f(x) = \max(0, x)$ as the pointwise nonlinearities in all hidden layers. For all SRLVMs we used three hidden layers, with 10-X-10 number of units, where X corresponds to the number of latent variables. The RLVM has only a single hidden layer corresponding to the latent variables (and thus both f_{enc} and f_{dec} are implemented as affine transformations).

bioRxiv preprint doi: https://doi.org/10.1101/407858; this version posted September 4, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

To simultaneously fit the weights θ_W and the biases θ_b in the networks defining f_{enc} and f_{dec}, we maximized the log-likelihood of the predicted responses under the Gaussian distribution (with identity covariance matrix), which is equivalent to minimizing the mean square error [39] between the observed spike counts $\mathbf{y}_i$ and the predicted responses $\mathbf{r}_i$, across all I trials. [Note that it is also possible to use the Poisson log-likelihood as the loss function, which we did not do owing to the large time bins and high spike counts.] We included regularization terms on the network weights to prevent overfitting, so that the final loss function for the model is defined as

$$\mathcal{L}_{\text{SRLVM}} = \frac{1}{2I} \sum_{i=1}^{I} \left\| \mathbf{y}_i - \mathbf{r}_i \right\|_2^2 + \lambda q(\theta_W) \tag{12}$$

where $q(\cdot)$ is the regularization term that we take to be the L_2 norm on the weights, governed by the hyperparameter λ. Equation 12 was optimized using the L-BFGS optimization routine [40].

We performed layer-wise initialization of the weights in the encoding network by using varimax-rotated principal components of the observed responses $\{\mathbf{y}_i\}_{i=1}^{I}$, and initialized weights in the decoding network as transposes of those in the encoding network, which we found to produce models with consistently better cross-validation performance than other initialization schemes (data not shown).

GAM. The GAM models neural responses using three terms: (1) the stimulus response; (2) one or more gain terms that are shared across the population and multiplicatively modulate the stimulus response; and (3) one or more additive terms that are likewise shared across the population. We define the full model for the predicted response r_i^n of neuron n on trial i in several steps.

We first define a stimulus model $f_n(\mathbf{s}_i)$ for each neuron n such that f_n maps $\mathbf{s}_i$, the vector of stimulus values on trial i, to a rate for each i. f_n could be as simple as a one-dimensional tuning curve or a peri-stimulus time histogram (PSTH), or any arbitrary feed-forward stimulus processing model like a GLM [28], NIM [29] or convolutional neural network [30, 31, 32, 33, 34, 35]. In this work, we modeled the stimulus response of each individual neuron in both datasets using an L_2-regularized form of the tuning curve, which can be fit using a penalized form of linear regression (ridge regression). We denote the parameters of the stimulus models $\{f_n\}_{n=1}^{N}$ as θ_{stim}. For this basic stimulus model, then, the GAM estimates the firing rate r_i^n as

$$r_i^n = F[c_n + f_n(\mathbf{s}_i)] \tag{13}$$

where c_n is an overall offset term and $F[\cdot]$ is an optional pointwise spiking nonlinearity (which we take to be linear in our analyses).

Next we define latent variables $\mathbf{g}_i \in \mathbb{R}^K$ that have an explicitly multiplicative effect on the stimulus processing (figure 3 only illustrates a single multiplicative latent variable, but more than one can be fit using this framework). These latent variables are shared across the entire population, though each neuron has its own weight w_n^k to each of the K latent variables g_i^k:

$$r_i^n = F\left[c_n + u\left(\sum_{k=1}^{K} w_n^k g_i^k + b_n\right) f_n(\mathbf{s}_i)\right] \tag{14}$$

where b_n is a bias term and $u(\cdot)$ is a static nonlinearity. We use $u(x) = 1 + x$ to fit the models in this work, though other functions like $u(x) = \exp(x)$ are also suitable [10].

Inference of the latent variables is performed by using a neural network f_{mult} to nonlinearly map the observed population response $\mathbf{y}_i$ into the latent variables:

$$\mathbf{g}_i = f_{\text{mult}}(\mathbf{y}_i) \tag{15}$$

We denote the parameters for the multiplicative latent variables (w_n^k, b_n, and the weights and biases of f_{mult}) as θ_{mult}.

Finally, we allow the GAM to have additive latent variables $\mathbf{h}_i \in \mathbb{R}^M$ as well, in order to capture activity that cannot be accounted for by the modulated stimulus model of equation 15. Like the multiplicative latent variables, these additive latent variables are shared across the population, but each neuron has its own weight v_n^m to each of the M latent variables h_i^m

$$r_i^n = F\left[c_n + u\left(\sum_{k=1}^{K} w_n^k g_i^k + b_n\right) f_n(\mathbf{s}_i) + \sum_{m=1}^{M} v_n^m h_i^m\right] \tag{16}$$

bioRxiv preprint doi: https://doi.org/10.1101/407858; this version posted September 4, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Inference of the additive latent variables likewise uses a neural network,

$$\mathbf{h}_i = f_{\text{add}}(\mathbf{y}_i) \tag{17}$$

and we denote the parameters for the additive latent variables (v_n^m and the weights and biases of f_{add}) as θ_{add}.

We implemented both f_{mult} and f_{add} with affine transformations; including hidden layers, such that the transformation from observed activity to latent variables is nonlinear, did not result in substantive model improvements (data not shown), likely due to the increased number of parameters and relative lack of data.

To fit the parameters $\{\theta_{\text{stim}}, \theta_{\text{mult}}, \theta_{\text{add}}\}$ of the GAM we define the loss function to be the penalized negative log-likelihood $\mathcal{L}_{\text{GAM}}$ under the Gaussian noise model (as with the SRLVM):

$$\mathcal{L}_{\text{GAM}} = \frac{1}{2I}\sum_{i=1}^{I}\left\|\mathbf{y}_i - \mathbf{r}_i\right\|_2^2 + \lambda_{\text{stim}} q(\theta_{\text{stim}}) + \lambda_{\text{add}} q(\theta_{\text{add}}) + \lambda_{\text{mult}} q(\theta_{\text{mult}}) \tag{18}$$

where the $q(\cdot)$ are regularization terms that we take to be the L_2 norm on the weights, governed by the hyperparameter λ. The parameters θ_{mult} and θ_{add} are initialized using the varimax-rotated principal components of the observed responses $\{\mathbf{y}_i\}_{i=1}^I$; θ_{stim} is initialized by fitting the desired stimulus processing model f_n (equation 13) to each neuron individually and using the resulting parameters. To train the full model we hold θ_{stim} fixed and simultaneously optimize θ_{mult} and θ_{add} using the L-BFGS optimization routine [40].

4.4 Evaluating model performance

To quantify the goodness-of-fit of the different models we employed the coefficient of determination across the full population, defined as

$$R^2 = \frac{1}{N}\sum_{n=1}^{N}\left[1 - \frac{\sum_i (y_i^n - r_i^n)^2}{\sum_i (y_i^n - \bar{y}^n)^2}\right] \tag{19}$$

where $\bar{y}^n$ is the average value for neuron n across all trials.

To actually evaluate model performance of the SRLVM we performed a version of the leave-one-out method introduced in [38]: first the full model is fit to all of the training data, and then for cross-validation the estimated population activity r_i^n on trial i for neuron n is calculated as

$$r_i^n = [f_{\text{dec}}(f_{\text{enc}}(\mathbf{y}_i^{-n}))]_n \tag{20}$$

where $\mathbf{y}_i^{-n}$ is the population activity on trial i with the value for neuron n set to zero. The prediction r_i^n is calculated in this way for each value of i and for each neuron n, and all such values are then combined for the final prediction $\{\mathbf{r}_i\}_{i=1}^I$. This procedure results in low cross-validation performance if any single neuron dominates the activity of a latent variable. The same procedure is used for GAMs, so that

$$r_i^n = F\left[c_n + u\left(\mathbf{w}_n^T f_{\text{mult}}(\mathbf{y}_i^{-n}) + b_n\right) f_n(\mathbf{s}_i) + \mathbf{v}_n^T f_{\text{add}}(\mathbf{y}_i^{-n})\right] \tag{21}$$

where $\mathbf{w}_n$ and $\mathbf{v}_n$ are the multiplicative and additive coupling terms for neuron n collected into vectors.

To compare the performance of the SRLVM and GAM, we compare the R^2 of the given latent variable model ("MODEL") with the R^2 of the stimulus model ("STIM"):

$$\text{QI} = \frac{R^2_{\text{MODEL}} - R^2_{\text{STIM}}}{1 - R^2_{\text{STIM}}} \tag{22}$$

This measure will be equal to zero when the latent variable model explains as much variability as the stimulus model, and one when the model predicts the neural activity perfectly.

Acknowledgments

This work was supported by the NSF IIS-1350990 and NIH R21 EY025403-02 (MRW and DAB) and NERF, a joint research initiative of imec, KU Leuven and VIB (KS and VB).

bioRxiv preprint doi: https://doi.org/10.1101/407858; this version posted September 4, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

References

[1] David J Tolhurst, J Anthony Movshon, and Andrew F Dean. The statistical reliability of signals in single neurons in cat and monkey visual cortex. *Vision research*, 23(8):775–785, 1983.

[2] Rufin Vogels, Werner Spileers, and Guy A Orban. The response variability of striate cortical neurons in the behaving monkey. *Experimental brain research*, 77(2):432–436, 1989.

[3] A Aldo Faisal, Luc PJ Selen, and Daniel M Wolpert. Noise in the nervous system. *Nature reviews neuroscience*, 9(4):292, 2008.

[4] Amos Arieli, Alexander Sterkin, Amiram Grinvald, and AD Aertsen. Dynamics of ongoing activity: explanation of the large variability in evoked cortical responses. *Science*, 273(5283): 1868–1871, 1996.

[5] Robbe LT Goris, J Anthony Movshon, and Eero P Simoncelli. Partitioning neuronal variability. *Nature neuroscience*, 17(6):858, 2014.

[6] Alexander S Ecker, Philipp Berens, R James Cotton, Manivannan Subramaniyan, George H Denfield, Cathryn R Cadwell, Stelios M Smirnakis, Matthias Bethge, and Andreas S Tolias. State dependence of noise correlations in macaque primary visual cortex. *Neuron*, 82(1): 235–248, 2014.

[7] Marieke L Schölvinck, Aman B Saleem, Andrea Benucci, Kenneth D Harris, and Matteo Carandini. Cortical state determines global variability and correlations in visual cortex. *Journal of Neuroscience*, 35(1):170–178, 2015.

[8] I-Chun Lin, Michael Okun, Matteo Carandini, and Kenneth D Harris. The nature of shared cortical variability. *Neuron*, 87(3):644–656, 2015.

[9] Marius Pachitariu, Dmitry R Lyamzin, Maneesh Sahani, and Nicholas A Lesica. State-dependent population coding in primary auditory cortex. *Journal of Neuroscience*, 35(5):2058–2073, 2015.

[10] Neil C Rabinowitz, Robbe L Goris, Marlene Cohen, and Eero P Simoncelli. Attention stabilizes the shared gain of v4 populations. *Elife*, 4:e08998, 2015.

[11] Michael Okun, Nicholas A Steinmetz, Lee Cossell, M Florencia Iacaruso, Ho Ko, Péter Barthó, Tirin Moore, Sonja B Hofer, Thomas D Mrsic-Flogel, Matteo Carandini, et al. Diverse coupling of neurons to populations in sensory cortex. *Nature*, 521(7553):511, 2015.

[12] Iñigo Arandia-Romero, Seiji Tanabe, Jan Drugowitsch, Adam Kohn, and Rubén Moreno-Bote. Multiplicative and additive modulation of neuronal tuning with population activity affects encoded information. *Neuron*, 89(6):1305–1316, 2016.

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

[14] Dmitry Kobak, Wieland Brendel, Christos Constantinidis, Claudia E Feierstein, Adam Kepecs, Zachary F Mainen, Xue-Lian Qi, Ranulfo Romo, Naoshige Uchida, and Christian K Machens. Demixed principal component analysis of neural population data. *Elife*, 5:e10989, 2016.

[15] Mark M Churchland, John P Cunningham, Matthew T Kaufman, Justin D Foster, Paul Nuyujukian, Stephen I Ryu, and Krishna V Shenoy. Neural population dynamics during reaching. *Nature*, 487(7405):51–56, 2012.

[16] Misha B Ahrens, Jennifer M Li, Michael B Orger, Drew N Robson, Alexander F Schier, Florian Engert, and Ruben Portugues. Brain-wide neuronal dynamics during motor adaptation in zebrafish. *Nature*, 485(7399):471, 2012.

[17] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

[18] Matthew R Whiteway and Daniel A Butts. Revealing unobserved factors underlying cortical activity with a rectified latent variable model applied to neural population recordings. *Journal of neurophysiology*, 117(3):919–936, 2017.

bioRxiv preprint doi: https://doi.org/10.1101/407858; this version posted September 4, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

[19] Adam Kohn and Matthew A Smith. Utah array extracellular recordings of spontaneous and visually evoked activity from anesthetized macaque primary visual cortex (v1). *http://dx.doi.org/10.6080/K0NC5Z4X*.

[20] Marlene R Cohen and Adam Kohn. Measuring and interpreting neuronal correlations. *Nature neuroscience*, 14(7):811, 2011.

[21] Roozbeh Kiani, Christopher J Cueva, John B Reppas, and William T Newsome. Dynamics of neural population responses in prefrontal cortex indicate changes of mind on single trials. *Current Biology*, 24(13):1542–1547, 2014.

[22] Roozbeh Kiani, Christopher J Cueva, John B Reppas, Diogo Peixoto, Stephen I Ryu, and William T Newsome. Natural grouping of neural responses reveals spatially segregated clusters in prearcuate cortex. *Neuron*, 85(6):1359–1373, 2015.

[23] Yuanjun Gao, Evan W Archer, Liam Paninski, and John P Cunningham. Linear dynamical neural population models through nonlinear embeddings. In *Advances in Neural Information Processing Systems*, pages 163–171, 2016.

[24] David Sussillo, Rafal Jozefowicz, LF Abbott, and Chethan Pandarinath. Lfads-latent factor analysis via dynamical systems. *arXiv preprint arXiv:1608.06315*, 2016.

[25] Peiran Gao, Eric Trautmann, M Yu Byron, Gopal Santhanam, Stephen Ryu, Krishna Shenoy, and Surya Ganguli. A theory of multineuronal dimensionality, dynamics and measurement. *bioRxiv*, page 214262, 2017.

[26] Carsen Stringer, Marius Pachitariu, Nicholas Steinmetz, Charu Bai Reddy, Matteo Carandini, and Kenneth D Harris. Spontaneous behaviors drive multidimensional, brain-wide population activity. *bioRxiv*, page 306019, 2018.

[27] Simon Musall, Matthew T Kaufman, Steven Gluf, and Anne Churchland. Movement-related activity dominates cortex during sensory-guided decision making. *bioRxiv*, page 308288, 2018.

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

[29] James M McFarland, Yuwei Cui, and Daniel A Butts. Inferring nonlinear neuronal computation based on physiologically plausible inputs. *PLoS Comput Biol*, 9(7):e1003143, 2013.

[30] Michael Oliver and Jack Gallant. A deep convolutional energy model of v4 responses to natural movies. *Journal of Vision*, 16(12):876–876, 2016.

[31] Lane McIntosh, Niru Maheswaranathan, Aran Nayebi, Surya Ganguli, and Stephen Baccus. Deep learning models of the retinal response to natural scenes. In *Advances in neural information processing systems*, pages 1369–1377, 2016.

[32] Eleanor Batty, Josh Merel, Nora Brackbill, Alexander Heitman, Alexander Sher, Alan Litke, EJ Chichilnisky, and Liam Paninski. Multilayer recurrent network models of primate retinal ganglion cell responses. 2016.

[33] Ján Antolík, Sonja B Hofer, James A Bednar, and Thomas D Mrsic-Flogel. Model constrained by visual hierarchy improves prediction of neural responses to natural scenes. *PLoS computational biology*, 12(6):e1004927, 2016.

[34] Santiago A Cadena, George H Denfield, Edgar Y Walker, Leon A Gatys, Andreas S Tolias, Matthias Bethge, and Alexander S Ecker. Deep convolutional models improve predictions of macaque v1 responses to natural images. *bioRxiv*, page 201764, 2017.

[35] William F Kindel, Elijah D Christensen, and Joel Zylberberg. Using deep learning to reveal the neural code for images in primary visual cortex. *arXiv preprint arXiv:1706.06208*, 2017.

[36] Liam Paninski and John Cunningham. Neural data science: accelerating the experiment-analysis-theory cycle in large-scale neuroscience. *bioRxiv*, page 196949, 2017.

bioRxiv preprint doi: https://doi.org/10.1101/407858; this version posted September 4, 2018. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

[37] Matthew A Smith and Adam Kohn. Spatial and temporal scales of neuronal correlation in primary visual cortex. *Journal of Neuroscience*, 28(48):12591–12603, 2008.

[38] M Yu Byron, John P Cunningham, Gopal Santhanam, Stephen I Ryu, Krishna V Shenoy, and Maneesh Sahani. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. In *Advances in neural information processing systems*, pages 1881–1888, 2009.

[39] CM Bishop. Pattern recognition and machine learning: springer new york. 2006.

[40] Mark Schmidt. minfunc: unconstrained differentiable multivariate optimization in matlab. *http://www.cs.ubc.ca/ schmidtm/Software/minFunc.html*, 2005.