A Flow-based latent state generative model of neural population responses to natural images

Anonymous Author(s)
Affiliation
Address
email

Abstract

We present a joint deep neural system identification model for two major sources of neural variability: stimulus-driven and stimulus-conditioned fluctuations. To this end, we combine (1) state-of-the-art deep networks for stimulus-driven activity and (2) a flexible, normalizing flow-based generative model to capture the stimulus-conditioned variability including noise correlations. This allows us to train the model end-to-end without the need for sophisticated probabilistic approximations associated with many latent state models for stimulus-conditioned fluctuations. We train the model on the responses of thousands of neurons from multiple areas of the mouse visual cortex to natural images. We show that our model outperforms previous state-of-the-art models in predicting the distribution of neural population responses to individual stimuli, including stimulus-conditioned changes in the distribution. Furthermore, it learns interesting latent factors of the population response including factors that capture behavioral variables such as pupil dilation, and other factors that vary systematically with brain area or retinotopic location. Overall, our model accurately accounts for two critical sources of neural variability while avoiding several complexities associated with many existing latent variable models. It thus provides a useful tool for uncovering the interplay between different factors that contribute to variability in neural activity.

1 Introduction

Characterizing the activity of sensory neurons is a major goal of neural system identification. While neural responses in the visual cortex vary with visual stimuli, they also exhibit significant variability even to the repeated presentations of identical stimuli [1–4]. This stimulus-conditioned variability has significant and sophisticated correlations commonly referred to as noise correlations [4–6] and exhibits dependency on various factors such as the stimulus [7–9], the behavioral task [10, 11], attention [12–14], and the general brain state [15, 16]. Understanding the nature of this correlated variability and its functional implication in the processing of sensory stimuli requires models that account for both stimulus-driven and shared stimulus-conditioned variability. The goal is thus to model the stimulus-conditioned response distribution $p(\mathbf{r}|\mathbf{x})$ of population activities $\mathbf{r} \in \mathbb{R}^n$ over n neurons responding to a stimulus $\mathbf{x}$. However, models that account for stimulus-driven and stimulus-conditioned correlated variability have been developed largely independently.

In the recent decade, we have seen significant progress in **modeling stimulus-driven activity**, largely driven by the use of deep neural networks (DNNs) [17–22]. In a typical DNN-based system identification, a deep network $\mathbf{f}_\theta$ is trained to capture the expected response of the neurons conditioned on the stimulus: $\mathbf{f}_\theta(\mathbf{x}) = \mathbb{E}[\mathbf{r}|\mathbf{x}]$, where θ is the trainable parameters of the network. Typically, these networks are trained using Poisson-loss. This is equivalent to assuming that the population activity $\mathbf{r}$ is distributed around the stimulus-conditioned mean $\mathbf{f}_\theta(\mathbf{x})$ with an independent Poisson distribution.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

Therefore, the existing state-of-the-art networks typically ignore correlation among neural responses, and impose strong assumptions about the form of the marginal distribution (i.e. Poisson) for each neuron. As sensory populations are known to exhibit noise correlations and deviate from Poisson distribution [4, 23, 24], this conditional independence assumption might substantially limit the ability of these models to accurately capture $p(\mathbf{r}|\mathbf{x})$.

On the other hand, many of the existing **models for stimulus-conditioned variability** capture the variations in the population activity through some—typically lower dimensional—shared latent state $\mathbf{z}$: $p(\mathbf{r}|\mathbf{x}) = \int p(\mathbf{r}|\mathbf{x}, \mathbf{z})p(\mathbf{z}|\mathbf{x})\,\mathrm{d}\mathbf{z}$ [16, 23–29]. Among these are hierarchical generative—doubly stochastic—models that can capture more sophisticated relationships between the stimulus and noise correlations, as well as deviations from Poisson, such as over-dispersion [23, 24, 27, 29]. Unfortunately, their more complex form of distribution usually requires a substantially more involved probabilistic machinery to make latent state inference and parameter fitting feasible. This usually limits the ability of these models to capture complex nonlinear stimulus-response relationships such as the response of cortical populations to arbitrary natural images or videos. As a consequence, most published latent state models for neural data either ignore the stimulus-driven variability all together [25–27, 29], or employ a very simple model of the stimulus-driven variations in the responses [16, 23, 24].

Here, we propose a new model that closes the gap between these two approaches by combining DNN-based models of stimulus-driven activity with a latent state model that accounts for shared stimulus-conditioned fluctuations. While DNNs can be trained effectively via gradient-based optimization, the challenge is to avoid the need for complex probabilistic machinery associated with existing latent state models. To this end, we combine normalizing flows [30–35] with Gaussian factor analysis (FA) models [36] for which inference and learning is simple. FA models are special cases of multi-variate Gaussian distributions with a particular low-rank structure of the covariance matrix. Since neuronal responses are not Gaussian, in particular for low firing rates, FA models are not directly applicable to neural responses. To circumvent this problem, variance-stabilizing transformations have been used in past work to make the responses more Gaussian-distributed [16, 25]. A common choice for this transformation is the square-root function. However, there may be other transformations that capture the response distribution more accurately. Ideally, this transformation would be learned. To achieve this flexibility, we allow our model to learn this transformation with a marginal normalizing flow.

Normalizing flow models are density estimators that use a series of diffeomorphisms to transform a source density given by data into a simple distribution—typically an isotropic Gaussian of the same dimension. While individual transformations are usually chosen to have an efficient-to-compute log-determinant, they typically act on the entire input vector to capture any statistical dependencies between the dimensions. Here, we replace the isotropic Gaussian by an FA model to capture dependencies and only use diffeomorphisms that act on each dimension separately. This choice has two important advantages: (1) The generative model is easy to train while combining state-of-the-art deep networks with flexible latent state models, and (2) the use of a marginal flows allows for an easy mechanism to compute conditional distributions of one neuron based on other neurons that would not necessarily be easy to obtain with non-marginal flow models.

In summary, we make the following contributions. We (1) combine state-of-the-art DNN-based models to predict changes in neural activity due to stimulus, with simple, yet flexible, flow-based latent state models to account for shared fluctuations in neural activity. We (2) apply our method on the activity of thousands of neurons from multiple areas of the mouse visual cortex in response to natural images, and demonstrate that it performs on par with current state-of-the-art methods in modelling response distributions from two-photon recordings. Finally, we (3) show that our model infers interesting latent state structures that exhibit meaningful relations to anatomy and other functional properties of visual sensory neurons.

2 Methods

2.1 Flow-based Factor Analysis model

For a given stimulus $\mathbf{x}$ and population response $\mathbf{r} \in \mathbb{R}^n$, where n is the number of neurons, we define our normalizing flow-based Factor Analysis (FlowFA) model of the stimulus-conditioned population

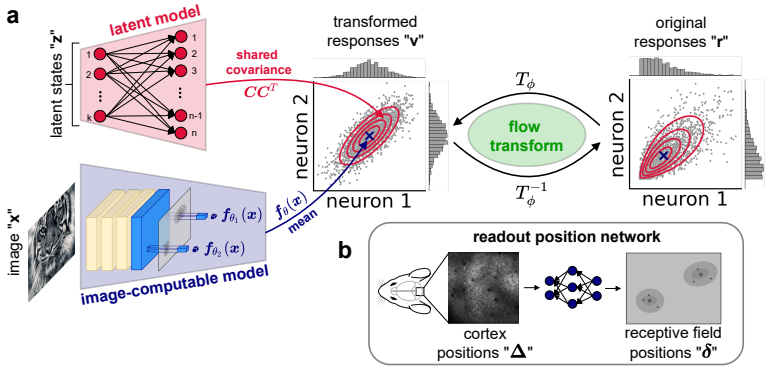

Figure 1: Flow-based Factor Analysis (FlowFA) model. **a:** Schematic of the FlowFA model relating all relevant variables in the study. **b:** Schematic of the sub-network used by the image-computable model to map cortical positions into receptive field positions. Refer to section 2.1 for the details.

activity $p(\mathbf{r}|\mathbf{x})$ as

$$p(\mathbf{r}|\mathbf{x}, \theta, \phi) = \mathcal{N}(T_\phi(\mathbf{r}); \mathbf{f}_\theta(\mathbf{x}), \mathbf{C}\mathbf{C}^\top + \Psi) \cdot |\det \nabla_\mathbf{r} T_\phi(\mathbf{r})| \,. \tag{1}$$

It has two major parts: (1) A flow model T_ϕ with learnable parameters ϕ that transforms the population responses $\mathbf{r}$ such that the transformed responses $\mathbf{v} = T_\phi(\mathbf{r})$ are well modelled by a (2) Gaussian factor analysis (FA) model $\mathcal{N}(\mathbf{v}; \mathbf{f}_\theta(\mathbf{x}), \mathbf{C}\mathbf{C}^\top + \Psi)$ (Fig. 1a). Here, $\mathcal{N}(\mathbf{v}; \mu, \Sigma)$ denotes a Gaussian distribution with mean μ and covariance Σ. According to the FA model, the random variable $\mathbf{v}$ is generated via $\mathbf{v} = \mathbf{f}_\theta(\mathbf{x}) + \mathbf{C}\mathbf{z} + \varepsilon$ where $\mathbf{z} \in \mathbb{R}^k$ is a low-dimensional latent state with $k \ll n$ and an isotropic Gaussian prior $\mathbf{z} \sim \mathcal{N}(0, I_k)$ whose samples map to $\mathbf{v}$ via the *factor loading matrix* $\mathbf{C} \in \mathbb{R}^{n \times k}$. The stimulus $\mathbf{x}$ enters via an additional mean component $\mathbf{f}_\theta(\mathbf{x}) \in \mathbb{R}^n$ that we model by a deep system identification network with learnable parameters θ (Fig. 1a,b). The noise is modelled by $\varepsilon \sim \mathcal{N}(0, \Psi)$ where $\Psi \in \mathbb{R}^{n \times n}$ is a diagonal covariance matrix.

Since the flow model is a learnable change of variables, it introduces the absolute determinant $|\det \nabla_\mathbf{r} T_\phi(\mathbf{r})|$ of the Jacobian ∇ of T_ϕ with respect to $\mathbf{r}$ into Eq. (1). The transform itself is a diffeomorphism, i.e. an invertible differentiable mapping $T_\phi : \mathbb{R}^n \mapsto \mathbb{R}^n$ allowing us to evaluate the exact likelihood of each data point and easily draw samples from the model. Therefore, the model serves as a fully generative model from which samples of the stimulus-conditioned population responses can easily be generated.

Furthermore, we choose T_ϕ to act on each single dimension separately, i.e.

$$T_\phi(\mathbf{r}) = \left[T_{\phi_1}^{(1)}(r_1), ..., T_{\phi_n}^{(n)}(r_n) \right]^\top .$$

This choice not only substantially simplifies the form of the determinant

$$\det \nabla_\mathbf{r} T_\phi(\mathbf{r}) = \prod_{i=1}^{n} \frac{\partial T_{\phi_i}^{(i)}}{\partial r_i},$$

but it also allows us to easily compute conditionals and marginals, by transforming $\mathbf{r}$ to the "transformed" space via T_ϕ, computing the marginal or conditional there and only using the determinant term on the remaining variables (see appendix B for the full derivation). This would not generally be possible for diffeomorphisms with a non-diagonal Jacobian.

2.2 Model components

Deep convolutional neural network $\mathbf{f}_\theta$ We model the stimulus-driven activity $\mathbf{f}_\theta(\mathbf{x})$ using a deep convolutional neural network with the same architecture as used by Lurz et al. [22]. Briefly, the network consists of two parts: (1) A shared four-layer core network, where each layer consists of a standard or depth-separable convolution operation resulting in 64 feature channels, followed by batch normalization and ELU nonlinearities (Fig. 1a), and (2) a neuron-specific readout mechanism (referred to as "Gaussian readout") that learns the position of the neuron's receptive field (RF) and

computes a weighted sum of the features at this position along the channel dimension (Fig. 1b). In contrast to Lurz et al. [22] where the RF positions $\boldsymbol{\delta}$ in image space were obtained by applying a shared affine transformation on the cortical coordinates $\boldsymbol{\Delta}$ of the neurons available from the experiment, here we allow this mapping to take on a non-linear form to allow flips in the visual field as a function of cortical position (Fig. 1b). This is crucial to model cortex-to-visual space mappings for multiple brains areas as the retinotopy of some areas are mirror images of each other across the area boundary. We also apply L1 regularization to the readout feature weights and L2 regularization on the Laplace filtered weights of the first layer.

Normalizing flow T_ϕ We construct the marginal flow model T_ϕ from a set of monotonic functions $\tau_k \in \{\text{Affine}, \text{ELU}, \log, \exp\}$, of which only the affine transformation has learnable parameters. To keep the parameters per single neuron minimal, we share the same transformation $T_\phi(r)$ among all neurons: $T_\phi = \text{Affine} \circ \log \circ \text{Affine} \circ \text{ELU} \circ \text{Affine} \circ \text{ELU} \circ \text{Affine} \circ \exp \circ \text{Affine}$. We compare the flow transformation against two common fixed transformations: square-root [16, 25] and Anscombe [37]. These two transformations can be expressed by the general form $u = \exp(a \log(y + b) + c)$ which is a series of Affine, $\log$, Affine, and $\exp$ transformations, with $a = 0.5$, $b = 0$, and $c = 0$ for square-root, and $a = 0.5$, $b = \frac{3}{8}$, and $c = \log(2)$ for Anscombe. We chose the components of T_ϕ such that common fixed transformations exist as special cases, ensuring that the flow transformations are strictly more flexible than any choice of fixed transformations commonly found in the literature.

2.3 Control models

We compare the FA-based models against two control models that assume independence among neurons with specific form of marginal distributions inspired by existing work: (1) Poisson distribution [18, 22] and (2) Zero-inflated Gamma (ZIG) distributions [38]. To enable capturing continuous neural responses measured with imaging, the discrete Poisson distribution is "relaxed" into continuous distribution by assuming $r = \hat{r} + \epsilon$ where $\hat{r} \sim \text{Poisson}(\lambda)$ and $\epsilon \sim \text{Uniform}[0, 1)$. The corresponding likelihood function is then:

$$p_{\text{poiss}}(\mathbf{r}|\mathbf{x}) = \prod_i^n \frac{\lambda_i(\mathbf{x})^{\lfloor r_i \rfloor} e^{-\lambda_i(\mathbf{x})}}{\lfloor r_i \rfloor !}, \tag{2}$$

where $\lambda(\mathbf{x})$ is the predicted firing rate of the neurons to input image $\mathbf{x}$ modeled as a DNN.

The ZIG distribution is a mixture of a uniform and a gamma distribution separated at the value ρ with no overlap [38]:

$$p_{\text{ZIG}}(\mathbf{r}|\mathbf{x}) = \prod_i^n \left(\frac{1 - q_i(\mathbf{x})}{\rho} + \frac{q_i(\mathbf{x})}{\Gamma(\kappa_i)\theta_i(\mathbf{x})^{\kappa_i}} r_i^{\kappa_i - 1} \exp\left(-\frac{r_i}{\theta_i(\mathbf{x})} \right) \right). \tag{3}$$

In order to combine the ZIG distribution with an image computable model, we let $q_i(\mathbf{x})$ and the scale parameter for the gamma distribution $\theta_i(\mathbf{x})$ be a function of the CNN. Similar to Wei et al. [38], we let the shape parameter κ_i be neuron specific, but independent of the input.

2.4 Neural and Behavioral Data

The data used in our experiments consists of pairs of gray-scale natural images and the corresponding neural population responses in mouse visual cortices. The stimulus presentation and data collection followed the protocol from Walker et al. [39]. Briefly, the neural responses were recorded from layer L2/3 of the mouse visual cortex, and measured using the genetically encoded calcium indicator GCaMP6s and a wide field two photon microscope [40]. In this study, we use two scans from two mice spanning three visual areas: scan 1 covers a field of $2000 \times 1800 \mu m^2$ over primary visual cortex (V1) and lateromedial area (LM); scan 2 covers two $600 \times 600 \times 25 \mu m^3$ volumes centered at V1 and posteromedial area (PM), respectively, each with 6 imaging planes spaced $5\mu m$ apart. The imaging planes in scan 1 and the most superficial plane in scan 2 were positioned in L2/3 at around $200\mu m$ from the surface of the cortex. Imaging was performed at approximately 9.7Hz for scan 1 and 7.2Hz for scan 2. The recorded visual areas were identified based on retinotopic maps generated using two-photon wide field imaging as previously described [41, 42]. Imaging data were automatically segmented into functional units and deconvolved using the CNMF algorithm [43]. Neurons were then selected by a classifier trained to detect somata based on segmented masks. 2,867 V1 neurons and

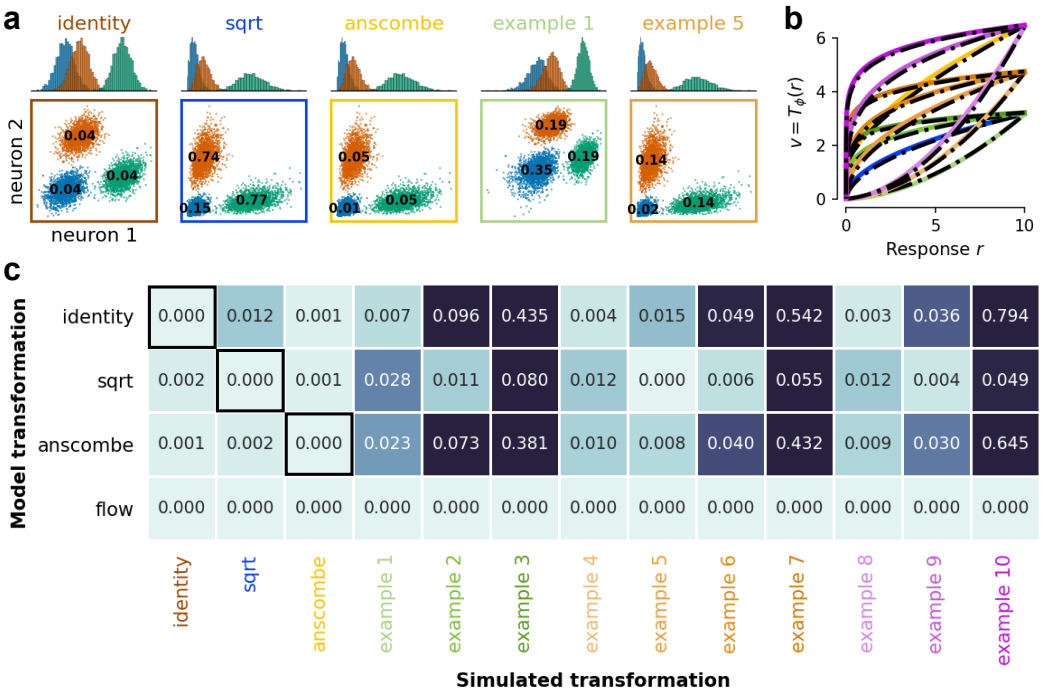

Figure 2: Flow model recovers the underlying transformation function. **a:** Simulated responses for 2 neurons under various transform functions. Across all transformations, transformed responses were sampled from Gaussian distributions with differing means (indicated by color of the samples) but identical covariance. Pearson correlation between the two neurons are shown in black. **b:** Transformations learned by the flow model is shown in black, overlaid on the ground-truth transformations. **c:** Performance of models with fixed or flow transformations (rows) trained on responses simulated with variety of transformations (columns). Cases where the simulating and trained transformations are the same are indicated by black outlines. Performance is measured as the KL divergence between the modeled and ground-truth simulating distributions, where 0 would correspond to a perfect fit.

907 LM neurons were detected in scan 1; 5,029 V1 neural responses and 3,343 PM neural responses were detected in scan 2. For training the models on scan 1 we used 1,000 V1 and 907 LM neurons; for scan 2 we used 1,000 V1 and 1,000 PM neurons, selected at random.

The natural image stimuli were sampled from ImageNet [44], cropped to fit a 16:9 monitor aspect ratio, and presented to the mice at a resolution of 0.53 ppd (pixels per degree of visual angle). The 64×36 px downsampled images were saved for later model fitting. Each image was presented for 500 ms followed by a blank screen with a duration uniformly distributed between 300 and 500 ms. A total of 6,000 images were shown in each scan, of which 1,000 images consist of 100 unique images repeated 10 times each to allow for an estimate of the neural response variability. We used the repeated images for testing, and split the remaining images into 4,500 training and 500 validation images. The neural responses were extracted as the accumulated activity of each neuron between 50 and 550 ms after stimulus onset. All behavior traces were extracted using the same temporal offset and integration window. The neural responses traces were normalized by their standard deviation computed on the training set.

2.5 Model fitting

We trained all models end-to-end via gradient-based optimization to maximize the log-likelihood obtained from Eq. (1), (2) or (3) for the corresponding model, optimizing over all parameters of the model. To ensure that Ψ, the diagonal covariance matrix, stays positive-valued, we re-parameterize $\Psi = e^{\nu}$ and optimized ν instead. For optimization, we use Adam [45] with (i) an early stopping mechanism [46] that would stop the training if a correlation-based score (refer to section 2.6) does

not improve for twenty training iterations, and (ii) a learning rate scheduler that reduces the learning rate by a factor of 0.3 if the score did not improve for ten training iterations. To find the best image-based models, we used Bayesian optimization [47] to find hyper-parameters that optimized the final score of the trained model. Hyper-parameters include the learning rate and regularization coefficient on the readout weights. The metric used for both early stopping and Bayesian optimization was computed on the validation set. Each instance of the model with a specific choice of hyper-parameters was trained on a workstation with a single NVIDIA GeForce RTX 2080 Ti GPU. A single model takes approximately 20–30 minutes to train. The hyperparameter search was completed using one GPU for a total of ~20 hours. All code for model definition, training, and evaluation were implemented in Python 3.8 using PyTorch [48] and NumPy [49] packages. The code can be found in the supplementary material.

2.6 Model evaluation

We compared the FA-based models to the control models based on likelihood and leave-neuron-out prediction correlation on the test set. For the former, we computed the likelihood of the responses in bits per neuron per image under each model, based on equations Eq. (1), (2), and (3), accordingly. For the correlation measure, we computed the conditional correlation between a single neuron and the measured responses on the test set, conditioned on the responses of all other neurons recorded simultaneously on the trial. Our FA-based model learns to capture conditional-dependence among neurons by learning the covariance matrix. We take advantage of this and compute the prediction of responses for a neuron r_i using all other neurons $\mathbf{r}_{\setminus i}$ by computing the posterior mean of the neuron's transformed response as $\mathbb{E}[v_i|\mathbf{x}, \mathbf{v}_{\setminus i}] = f_{\theta,i}(\mathbf{x}) + \Sigma_{i,\setminus i}\mathbf{\Sigma}^{-1}_{\setminus i,\setminus i}(T_\phi(\mathbf{v}_{\setminus i}) - \mathbf{f}_{\theta,\setminus i}(\mathbf{x}))$ where $\Sigma = \mathbf{C}\mathbf{C}^\top + \Psi$. We then compute the Spearman correlation between the predicted posterior mean with the experimentally observed single-trial transformed responses. We used Spearman correlation as its value computed on the transformed responses is identical to that on the original responses for the FA-based models with a stimulus-independent covariance (see appendix A for the proof).

3 Results

FlowFA model faithfully recovers invertible transformations on synthetic data We first use synthetic data to illustrate that our FlowFA model with a learnable transformation can adequately learn and recover a wide variety of transformation that underlies a distribution. To this end, we sampled 5,000 data points for a model with 100 neurons from several ground-truth model with different transforms (see appendix C for details on data generation). Even though the samples from the FA model have fixed covariance, non-linearly transforming them can yield responses that exhibit correlation structures that vary with the mean response and thus with the stimulus (Fig. 2a). The invertible transforms for Examples 1–10 had the general form $\exp(a\log(y+b)+c)$ with differing values of a, b, and c (Fig. 2b). We trained FA-based models with either a fixed transformation (FixedFA) or learnable flow (FlowFA). As expected, the models with a fixed transformation perform well if the data was simulated with a similar transformation, but the performance suffers when the transformations differ significantly (Fig. 2c, first three rows). In contrast, the FlowFA is able to flexibly learn every underlying transformation (Fig. 2b) and captures all distributions across all simulations (Fig. 2c, last row).

3.1 Modeling responses of mouse visual cortex

FlowFA model captures cortical response distribution well As we have demonstrated that the flow-based model can effectively fit a wide range of distributions, we now use it to capture distributions of the mouse visual cortex population responses to natural images, recorded in two different two-photon scans from two mice (scan 1 and scan 2, refer to section 2.4 for details). We train both FixedFA and FlowFA models for different values of latent dimensions $k \in \{0, 1, 2, 3\}$. When $k = 0$, the model assumes the responses to be independent conditioned on the stimulus, similar to the assumption used in the control models (i.e. Poisson and ZIG). We measure the model performance by computing the likelihood as well as the conditional Spearman correlations (see section 2.6).

The FlowFA model outperforms other FixedFA models (Fig. 3a) across all values of k. For $k = 0$ latent dimensions, the ZIG model outperforms all FA-based models in terms of likelihood (Fig. 3a).

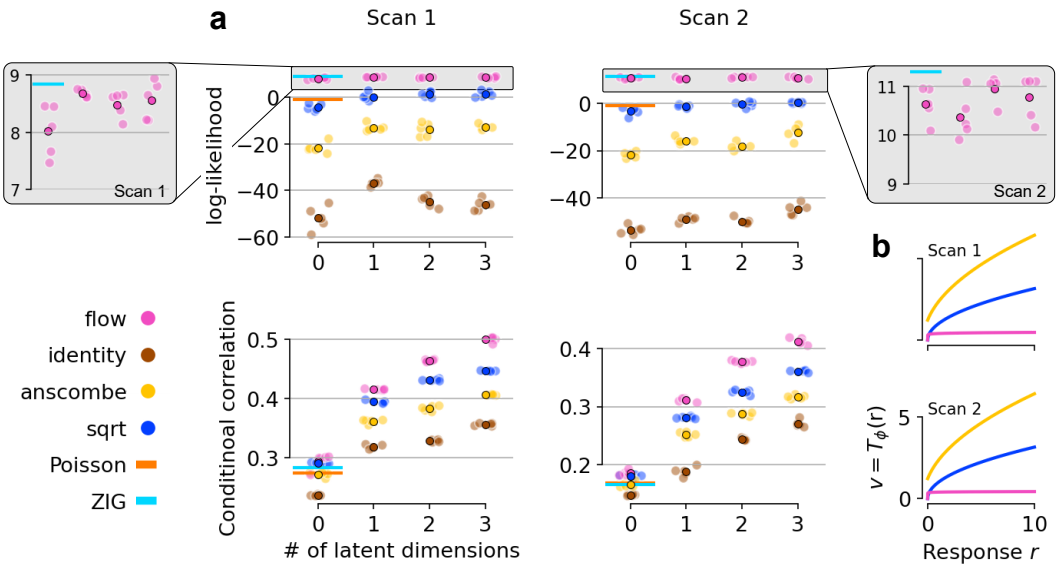

Figure 3: Comparison of models trained on the mouse visual cortical population responses to natural images. **a**: log-likelihood (top row) and conditional Spearman correlation (bottom row) computed for models trained on scan 1 (left half) and scan 2 (right half). Values for both individual (ligther shade) and average (darker shade) model trained under various random seeds are shown. Gray block provides a zoomed-in view of the FlowFA and Zero-Inflated-Gamma (ZIG) models. **b**: Transformation learned (pink) by the flow model shown in comparison to fixed transformations.

However, with increasing latent dimensions, the FlowFA model outperforms the ZIG model in terms of correlations and approaches the ZIG model in terms of likelihood (Fig. 3a, gray insets). This means that the ZIG model better captures the marginal distributions, while the FA-based models better capture dependencies between neurons, which the ZIG model cannot by construction. The FlowFA model outperforms the Poisson model in terms of both likelihood and correlation. The FlowFA model also outperforms all FixedFA models, which is corroborated by the fact the learned transformation markedly differs from all fixed transformation (Fig. 3b).

FlowFA model learns latent states related to anatomy and behavior In this section, we briefly explore the kind of neuroscientific analyses and insights our model can provide.

As mentioned above, our model is equipped with a DNN the predicts the RF location δ of each neuron in visual space from its cortical location Δ (Fig. 1b). This network can be used to infer cortical areas of neurons, as the retinotopic map flips at the area border [41]. To detect this flip, we computed the sign of the determinant of the Jacobian of the predicted receptive locations with respect to the cortical positions $\operatorname{sign} \det \frac{\partial \delta}{\partial \Delta}$ (Fig. 4a, right). When we compare our area map to the experimentally measured area assignment, we find a very good match (Fig. 4a, left vs. right). This means that our model principally allows neuroscientists to infer the area assignment from natural image responses only, without the need to run an extra protocol for area assignments.

Knowing both cortical and learned RF positions of the visual sensory neurons, we then set out to explore whether the effect of latent states on the neurons is related to their cortical or RF positions. To this end, we plotted the sign and magnitude of the corresponding weight from latent state to neuron in the factor loading matrix $\mathbf{C}$ of the FA model in Eq. (1). We observed that the effect of some latent states vary systematically across brain areas (scan 1 latent dimension 1) , or as a function of receptive field locations/retinotopy (scan 1 latent dimensions 2 and 3; scan 2 latent dimension 1) (Fig. 4b,c)

The FA-based model also allows us to infer the most likely latent state correspoding to a particular trial (Fig. 4d). We observed that some of these inferred latent states correlated strongly with behavioral variables such as pupil dilation, as expected from previous work that uses pupil dilation as a proxy to quantify the effect of arousal and attention on neural responses [50–54] .

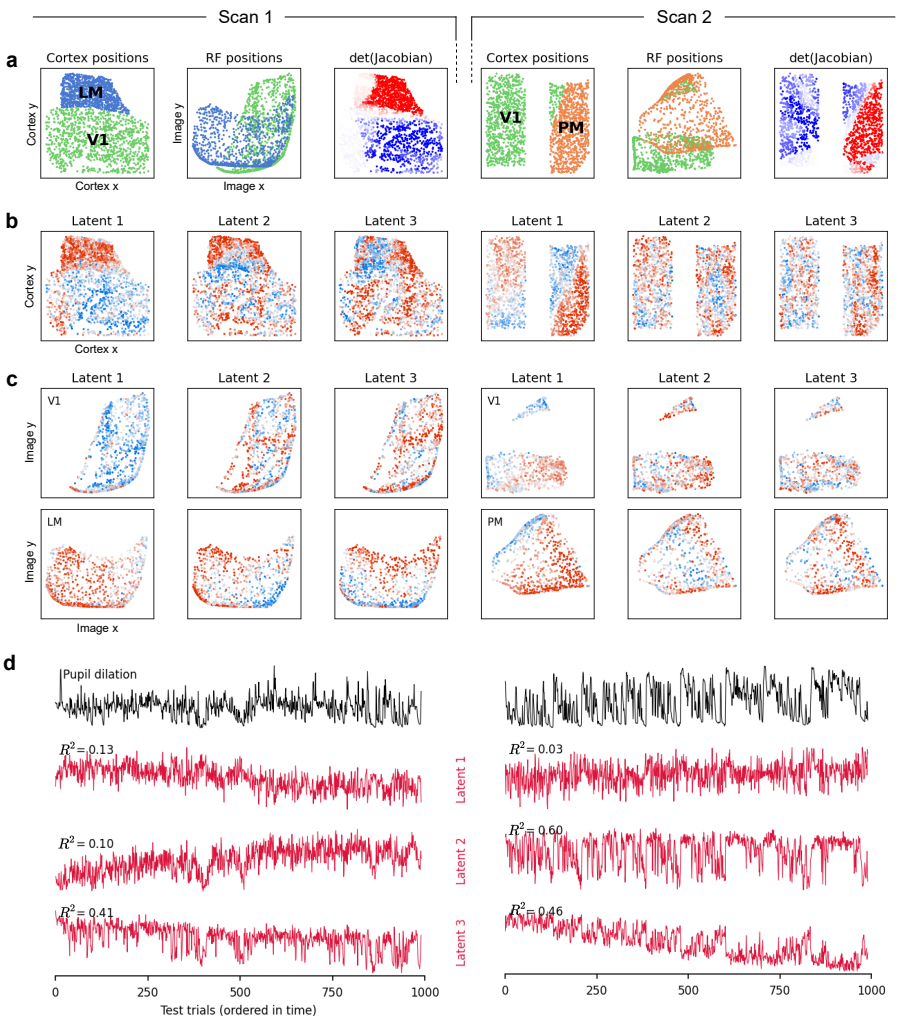

Figure 4: Analysis of flow-based latent state model. **a**: Brain area identification from responses of visual sensory neurons to natural images. From left to right for each scan: cortical position of recorded neurons; learned receptive field position for each neuron as a function of cortical positions; determinant of the relative changes in visual field with respect to changes in cortical position (i.e. Jacobian). **b–c**: Distribution of the latent-to-neuron weights across cortical positions (b) and receptive field positions (c), shown for 3 distinct latent dimensions under the FlowFA model with $k = 3$. **d**: The values of pupil dilation (black) and the inferred latent states (red) across trials from the test set. R^2 values are computed between the inferred latent state and the pupil dilation trace of the corresponding scan across all trials from the test set.

However, we also found other latent states, uncorrelated with pupil dilation (scan 1 latent dimension 2; scan 2 latent dimension 1). Interestingly, these latent dimensions seem to be retinotopically organized. These observations illustrate that the latent states inferred by our model can be a useful tool for uncovering the functional and structural implications of the behavioral or internal processes associated with the inferred latent states.

4 Discussion

Two major components of cortical neural variability are the variability due to stimulus and the variability due to unobserved or internal processes such as behavioral tasks or general brain states that affect population of neurons in similar ways giving rise to correlated variability among neurons. We presented a model that combines a state-of-the-art DNN-based models to predict changes in neural activity due to stimulus with a simple, yet flexible, flow-based factor analysis to account for

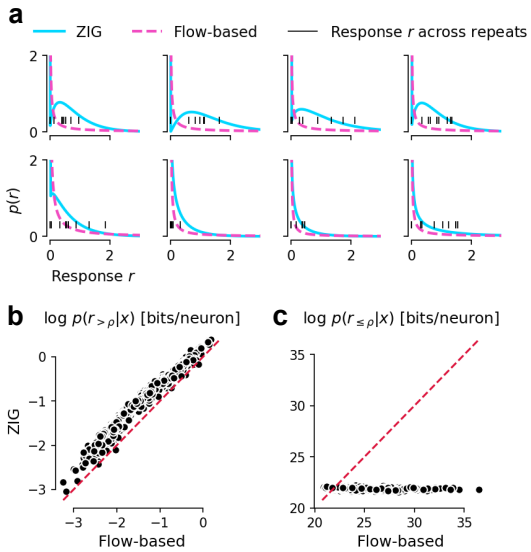

Figure 5: Marginals in the ZIG vs. FAFlow model. **a**: Example marginal distribution of responses of 8 sample neurons from the test set and the corresponding fits of ZIG and FAFlow. While both model the peak at zero, the FAFlow puts too little probability mass on positive responses. **b–c**: ZIG vs. FAFlow log-likelihood in bits/neuron for responses $\mathbf{r}_{>\rho} = \{r_i \| r_i(\mathbf{x}) > \rho\}$ and $\mathbf{r}_{\leq\rho}$, respectively. Each point is a single trial. The fact that most points are above the diagonal shows that the ZIG model puts more mass on the positive responses (**b**) while the FAFlow puts more mass on the responses $\leq \rho$, where ρ is the location parameter of the ZIG model.

correlated neural activity. This formulation allows us to compute exact likelihood evaluation of neural responses, easily sample stimulus-conditioned responses, and efficiently compute conditional and marginal distributions of subsets of neurons. By fitting this model to the activity of thousands of neurons from multiple areas of mouse visual cortex in response to natural images, we were able to demonstrate that our models performs similarly to current state-of-the-art models of neural response while additionally yielding latent states that exhibit meaningful relations to anatomy and functional properties of visual sensory neurons.

Our approach is related to copulas, in that we aim to capture dependencies between neurons by transforming the marginal distributions. However, while copulas have been used previously to model dependencies between spiking activity of neurons [55] they make an assumption about the marginal distribution in the original space. In contrast, while achieving the same goal, our approach does not require such assumption. In that sense, our approach can be seen as a generalization of copula-based methods.

One of the strengths of our model is also its limitation: while the normalizing flow in combination with the FA model allows us to compute exact likelihoods and conditionals of neural responses, flow models use diffeomorphisms to map one distribution into another and this cannot separate the peak at 0—typical for the response of neurons—from the positive response. Thus, our model does not capture the marginal responses as well as the ZIG model (Fig. 5a). In particular, the shared marginal transformation between all neurons leads to a stronger emphasis on the zero component and less emphasis on the positive response part (Fig. 5b,c). As the ZIG model has two components for that, it has an easier time fitting the marginals. Despite this, our model still infers meaningful shared latent states and outperforms the ZIG model in per neuron prediction conditioned on others. In future work, we will extend our flow model to separately deal with the positive and the zero component in order to more accurately reflect the marginals.

Accurate models of neural variability such as the one presented here can not only lead to deeper scientific insights and understanding of how a healthy, functional brain perceives and computes with sensory information, but can eventually also provide insights into how neurological and psychological disorders disturb these functions. In particular, a more accurate model that relates internal brain states, stimulus driven responses, and anatomical features such as retinotopy or memberships to certain brain areas might provide deeper insights into the computational principles of cortex. Naturally, our model requires data from animal experiments to be trained. However, we used existing datasets with very general protocols that can be used in several analyses to make efficient scientific use of data from animal experiments. Furthermore, models such as the one presented here, do help to reduce the amount of animal experiments as faithful models allow us to explore the functional principles of neural populations *in silico*.

References

[1] A F Dean. The variability of discharge of simple cells in the cat striate cortex. *Exp. Brain Res.*, 44(4):437–440, 1981.

[2] D J Tolhurst, J A Movshon, and A F Dean. The statistical reliability of signals in single neurons in cat and monkey visual cortex. *Vision Res.*, 23(8):775–785, 1983.

[3] George J Tomko and Donald R Crapper. Neuronal variability: non-stationary responses to identical visual stimuli. *Brain research*, 79(3):405–418, 1974.

[4] Michael N Shadlen and William T Newsome. The variable discharge of cortical neurons: implications for connectivity, computation, and information coding. *Journal of neuroscience*, 18(10):3870–3896, 1998.

[5] 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.

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

[7] Krešimir Josić, Eric Shea-Brown, Brent Doiron, and Jaime de la Rocha. Stimulus-dependent correlations and population codes. *Neural computation*, 21(10):2774–2804, 2009.

[8] Adrián Ponce-Alvarez, Alexander Thiele, Thomas D Albright, Gene R Stoner, and Gustavo Deco. Stimulus-dependent variability and noise correlations in cortical mt neurons. *Proceedings of the National Academy of Sciences*, 110(32):13162–13167, 2013.

[9] Mihály Bányai, Andreea Lazar, Liane Klein, Johanna Klon-Lipok, Marcell Stippinger, Wolf Singer, and Gergő Orbán. Stimulus complexity shapes response correlations in primary visual cortex. *Proceedings of the National Academy of Sciences*, 116(7):2723–2732, 2019.

[10] Marlene R Cohen and William T Newsome. Context-dependent changes in functional circuitry in visual area mt. *Neuron*, 60(1):162–173, 2008.

[11] Ralf M Haefner, Pietro Berkes, and József Fiser. Perceptual decision-making as probabilistic inference by neural sampling. *Neuron*, 90(3):649–660, 2016.

[12] Marlene R Cohen and John HR Maunsell. Attention improves performance primarily by reducing interneuronal correlations. *Nature neuroscience*, 12(12):1594, 2009.

[13] Jude F Mitchell, Kristy A Sundberg, and John H Reynolds. Spatial attention decorrelates intrinsic activity fluctuations in macaque area v4. *Neuron*, 63(6):879–888, 2009.

[14] Farran Briggs, George R Mangun, and W Martin Usrey. Attention enhances synaptic efficacy and the signal-to-noise ratio in neural circuits. *Nature*, 499(7459):476–480, 2013.

[15] Cristopher M Niell and Michael P Stryker. Modulation of visual responses by behavioral state in mouse visual cortex. *Neuron*, 65(4):472–479, 2010.

[16] 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.

[17] David A Klindt, Alexander S Ecker, Thomas Euler, and Matthias Bethge. Neural system identification for large populations separating "what" and "where". *Adv. Neural Inf. Process. Syst.*, November 2017.

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

[19] Lane T McIntosh, Niru Maheswaranathan, Aran Nayebi, Surya Ganguli, and Stephen A Baccus. Deep learning models of the retinal response to natural scenes. In *Advances in neural information processing systems*, volume 29, pages 1369–1377, February 2016.

[20] Fabian Sinz, Alexander S Ecker, Paul Fahey, Edgar Walker, Erick Cobos, Emmanouil Froudarakis, Dimitri Yatsenko, Zachary Pitkow, Jacob Reimer, and Andreas Tolias. Stimulus domain transfer in recurrent models for large scale cortical population prediction on video. In *Advances in Neural Information Processing Systems 31*, pages 7199–7210, 2018.

[21] 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. *PLoS computational biology*, 15(4):e1006897, 2019.

[22] Konstantin-Klemens Lurz, Mohammad Bashiri, Konstantin Friedrich Willeke, Akshay Kumar Jagadish, Eric Wang, Edgar Y Walker, Santiago Cadena, Taliah Muhammad, Eric Cobos, Andreas Tolias, et al. Generalization in data-driven models of primary visual cortex. *bioRxiv*, 2020.

[23] Adam S Charles, Mijung Park, J Patrick Weller, Gregory D Horwitz, and Jonathan W Pillow. Dethroning the fano factor: a flexible, model-based approach to partitioning neural variability. *Neural computation*, 30(4):1012–1045, 2018.

[24] Robbe LT Goris, J Anthony Movshon, and Eero P Simoncelli. Partitioning neuronal variability. *Nature neuroscience*, 17(6):858–865, 2014.

[25] Byron M Yu, 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. *J. Neurophysiol.*, 102(1):614–635, July 2009.

[26] Jakob H Macke, Lars Buesing, John P Cunningham, Byron M Yu, Krishna V Shenoy, and Maneesh Sahani. Empirical models of spiking in neural populations. In J Shawe-Taylor, R S Zemel, P L Bartlett, F Pereira, and K Q Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1350–1358. Curran Associates, Inc., 2011.

[27] Evan W Archer, Urs Koster, Jonathan W Pillow, and Jakob H Macke. Low-dimensional models of neural population activity in sensory cortical circuits. In *Advances in Neural Information Processing Systems 27: 28th Conference on Neural Information Processing Systems (NIPS 2014)*, pages 343–351, 2015.

[28] Yuan Zhao and Il Memming Park. Variational latent gaussian process for recovering Single-Trial dynamics from population spike trains. *Neural Comput.*, 29(5):1293–1316, May 2017.

[29] Anqi Wu, Nicholas A Roy, Stephen Keeley, and Jonathan W Pillow. Gaussian process based nonlinear latent structure discovery in multivariate spike train data. *Adv. Neural Inf. Process. Syst.*, 30:3496–3505, December 2017.

[30] E G Tabak. A family of non-parametric density estimation algorithms. `https://www.math.nyu.edu/~tabak/publications/Tabak-Turner.pdf`, 2000. Accessed: 2021-5-25.

[31] Esteban G Tabak and Eric Vanden-Eijnden. Density estimation by dual ascent of the log-likelihood. *CMS Books Math./Ouvrages Math. SMC*, 8(1):217–233, March 2010.

[32] Oren Rippel and Ryan Prescott Adams. High-Dimensional probability estimation with deep density models. February 2013.

[33] J P Agnelli, M Cadeiras, E G Tabak, C V Turner, and E Vanden-Eijnden. Clustering and classification through normalizing flows in feature space. *Multiscale Model. Simul.*, 8(5): 1784–1802, January 2010.

[34] L Dinh, J Sohl-Dickstein, and S Bengio. Density estimation using real NVP. Technical report, 2017.

[35] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. *1505.05770*, 2015.

[36] Sam Roweis and Zoubin Ghahramani. A unifying review of linear gaussian models. *Neural Comput.*, 11(2):305–345, 1999.

[37] Shaul K Bar-Lev and Peter Enis. On the classical choice of variance stabilizing transformations and an application for a poisson variate. *Biometrika*, 75(4):803–804, 1988.

[38] Xue-Xin Wei, Ding Zhou, Andres Grosmark, Zaki Ajabi, Fraser Sparks, Pengcheng Zhou, Mark Brandon, Attila Losonczy, and Liam Paninski. A zero-inflated gamma model for deconvolved calcium imaging traces. June 2020.

[39] Edgar Y Walker, Fabian H Sinz, Erick Cobos, Taliah Muhammad, Emmanouil Froudarakis, Paul G Fahey, Alexander S Ecker, Jacob Reimer, Xaq Pitkow, and Andreas S Tolias. Inception loops discover what excites neurons most using deep predictive models. *Nature neuroscience*, 22(12):2060–2065, 2019.

[40] Nicholas James Sofroniew, Daniel Flickinger, Jonathan King, and Karel Svoboda. A large field of view two-photon mesoscope with subcellular resolution for in vivo imaging. *Elife*, 5:e14472, 2016.

[41] Marina E Garrett, Ian Nauhaus, James H Marshel, and Edward M Callaway. Topography and areal organization of mouse visual cortex. *Journal of Neuroscience*, 34(37):12587–12600, 2014.

[42] Emmanouil Froudarakis, Uri Cohen, Maria Diamantaki, Edgar Y Walker, Jacob Reimer, Philipp Berens, Haim Sompolinsky, and Andreas S Tolias. Object manifold geometry across the mouse cortical visual hierarchy. August 2020.

[43] Eftychios A Pnevmatikakis, Daniel Soudry, Yuanjun Gao, Timothy A Machado, Josh Merel, David Pfau, Thomas Reardon, Yu Mu, Clay Lacefield, Weijian Yang, et al. Simultaneous denoising, deconvolution, and demixing of calcium imaging data. *Neuron*, 89(2):285–299, 2016.

[44] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

[45] D P Kingma and J Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, pages 1–13, 2014.

[46] Lutz Prechelt. Early stopping — but when? In Grégoire Montavon, Geneviève B Orr, and Klaus-Robert Müller, editors, *Neural Networks: Tricks of the Trade: Second Edition*, pages 53–67. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[47] Facebook. Adaptive experimentation platform, 2019. URL `https://ax.dev/`.

[48] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

[49] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL `https://doi.org/10.1038/s41586-020-2649-2`.

[50] Jacob Reimer, Emmanouil Froudarakis, Cathryn R R Cadwell, Dimitri Yatsenko, George H H Denfield, and Andreas S S Tolias. Pupil fluctuations track fast switching of cortical states during quiet wakefulness. *Neuron*, 84(2):355–362, 2014.

[51] Martin Vinck, Renata Batista-Brito, Ulf Knoblich, and Jessica A Cardin. Arousal and locomotion make distinct contributions to cortical activity patterns and visual encoding. *Neuron*, 86(3): 740–754, May 2015.

[52] Matthew J McGinley, Martin Vinck, Jacob Reimer, Renata Batista-Brito, Edward Zagha, Cathryn R Cadwell, Andreas S Tolias, Jessica A Cardin, and David A McCormick. Waking state: Rapid variations modulate neural and behavioral responses. *Neuron*, 87(6):1143–1161, September 2015.

[53] Jacob Reimer, Matthew J McGinley, Yang Liu, Charles Rodenkirch, Qi Wang, David A McCormick, and Andreas S Tolias. Pupil fluctuations track rapid changes in adrenergic and cholinergic activity in cortex. *Nat. Commun.*, 7:13289, November 2016.

[54] Siddhartha Joshi, Yin Li, Rishi M. Kalwani, and Joshua I. Gold. Relationships between Pupil Diameter and Neuronal Activity in the Locus Coeruleus, Colliculi, and Cingulate Cortex. *Neuron*, 89(1):221–234, 2016. ISSN 0896-6273. doi: 10.1016/j.neuron.2015.11.028.

[55] Pietro Berkes, Frank Wood, and Jonathan Pillow. Characterizing neural dependencies with copula models. `https://pillowlab.princeton.edu/pubs/Berkes09_Copulas_NIPS.pdf`. Accessed: 2021-5-22.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Please refer to section 4 Discussion.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Implications in animal studies is discussed in section 4 Discussion.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] Any assumption of the model is stated in the derivation e.g. section 2.1 Flow-based Factor Analysis model
   (b) Did you include complete proofs of all theoretical results? [Yes] Most derivations are provided directly in the paper. Some additional proofs and derivations are submitted as Appendix
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The code, an example dataset, as well as instructions to reproduce the results will be included in the supplemental material. The data from mouse visual cortex that was used in this work would be made available upon request.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See section 2.5 Model fitting and Appendix
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Results for all seeds are shown in Fig. 3
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See section 2.5 Model fitting
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] All assets used in this work are cited.
   (b) Did you mention the license of the assets? [N/A]

(c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] No human subject was involved in this work. All data was collected from behaving mice.

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]