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

Mohammad Bashiri,[1,*] Edgar Y. Walker,[1,*] Konstantin-Klemens Lurz,[1]
Akshay Kumar Jagadish,[1,2], Taliah Muhammad,[3-4] Zhiwei Ding,[3-4] Zhuokun Ding,[3-4]
Andreas S. Tolias,[3-4] Fabian H. Sinz[5,1,†]
[1] Institute for Bioinformatics and Medical Informatics, University of Tübingen, Germany
[2] Max Planck Institute for Biological Cybernetics, Tübingen, Germany
[3] Department for Neuroscience, Baylor College of Medicine, Houston, TX, USA
[4] Center for Neuroscience and Artificial Intelligence, Baylor College of Medicine, Houston, TX, USA
[5] Department of Computer Science, University Göttingen, Germany

[*]equal contribution, [†]sinz@cs.uni-goettingen.de

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 novel stimuli, including shared stimulus-conditioned variability. Furthermore, it successfully learns known latent factors of the population responses that are related to 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 state 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 variability to the repeated presentations of identical stimuli [1–4]. This stimulus-conditioned variability has significant and sophisticated correlations among neurons 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 activity $\mathbf{r} \in \mathbb{R}^n$ over n neurons responding to an arbitrary sensory stimulus $\mathbf{x}$. However, models that account for stimulus-driven and stimulus-conditioned correlated variability have been developed largely independently.

Preprint. Under review.

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]. Typically, the expected response of the neurons conditioned on the stimulus is captured as a function of the stimulus via a deep network $\mathbf{f}_\theta(\mathbf{x}) = \mathbb{E}[\mathbf{r}|\mathbf{x}]$ with learnable parameters θ. These models can therefore predict how population responses depend on an arbitrary stimulus, and could even be used to derive stimuli that would yield desirable responses [23, 24]. Typically, these networks are trained using Poisson-loss, assuming that the population activity $\mathbf{r}$ is distributed around the stimulus-conditioned mean $\mathbf{f}_\theta(\mathbf{x})$ with an independent Poisson distribution. Therefore, existing state-of-the-art networks commonly ignore stimulus-conditioned correlations 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 distributions [4, 25, 26], this conditional independence assumption might 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 by specifically modeling the responses to repeated presentations of an identical stimulus. Many of these approaches employ statistical techniques such as maximum-entropy or copula distributions to reduce the number of parameters needed to fit the target distribution [27–29]. A popular approach has been to describe the stimulus-conditioned variability in terms of a 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, 25, 26, 30–35]. Among these are hierarchical generative models that can capture more sophisticated relationships between the stimulus and noise correlations, as well as deviations from Poisson, such as over-dispersion [25, 26, 32, 34, 35]. While these approaches present powerful methods to capture stimulus-conditioned variability, they often fit $p(\mathbf{r}|\mathbf{x})$ separately for each unique stimulus and require responses to repeated presentations of the stimulus [16, 25, 26, 29, 35]. This limits their ability to yield predictions to a novel stimulus without requiring some stimulus-specific parameters to be learned. Furthermore, the increased complexity of the distribution usually requires a substantially more involved probabilistic machinery to make latent state inference and parameter fitting feasible. Consequently, most latent state models for neural data either ignore stimulus-driven variability altogether [30–32, 34], or employ a very simple model of stimulus-driven variations [16, 25, 26].

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 variability. While DNNs can be trained effectively via gradient-based optimization, the challenge is to avoid the complex probabilistic machinery associated with existing latent state models, particularly those that require stimulus-specific parameters to be learned over repeated presentations of identical stimuli. To this end, we combine normalizing flows [36–41] with Gaussian Factor Analysis (FA) models [42], where the stimulus-dependence occurs through a DNN that learns to shift the mean of the FA distribution based on the stimulus. FA models make use of multivariate Gaussian distributions with a particular low-rank structure of the covariance matrix. While the use of FA in capturing shared variability greatly simplifies inference and learning, it is not directly applicable to neural responses because neural responses are not Gaussian-distributed, particularly for low firing rates. To circumvent this problem, variance-stabilizing transformations, such as the square-root function, have been used in the past to make the responses more Gaussian-distributed [16, 30]. However, there may be other transformations that capture the response distribution more accurately. Furthermore, since the transformation for one neuron may not be applicable to other neurons, ideally it would be learned for each neuron separately. To achieve this flexibility, we allow our model to learn neuron-specific transformations with a marginal normalizing flow.

Normalizing flow models are density estimators that use a series of diffeomorphisms to transform the source density underlying the data into a simple distribution—typically an isotropic Gaussian of the same dimension. These transformations are usually chosen to have efficient-to-compute log-determinants, and typically act on the entire variable vector to capture any statistical dependencies between the dimensions. Here, we replace the isotropic Gaussian with an FA model to capture dependencies among dimensions and only use diffeomorphisms that act on each dimension separately, i.e. apply flow-based transformations on the marginals only. While this choice places certain restrictions on the complex dependencies between neurons that may be captured (refer to section 4 Discussion for details), it 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 marginal flows allows for an easy mechanism to compute conditional distributions of one neuron given responses of other neurons that would not be easy to obtain with non-marginal flow models.

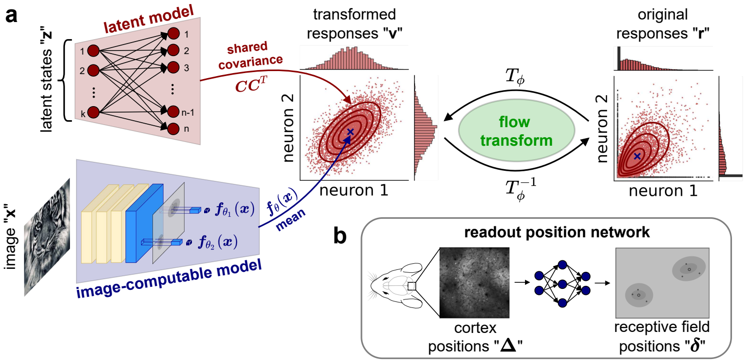


Figure 1: Flow-based Factor Analysis model. **a:** Schematic of the flow-based model relating all relevant variables in the study. **b:** Schematic of the sub-network used by the image-computable model to map cortical positions into receptive field positions. Refer to section 2 Methods for the details.

In summary, we make the following contributions. We (1) combine state-of-the-art DNN-based models with flow-based latent state models to jointly account for stimulus-driven and shared stimulus-conditioned variability in neural population activity. Our model can predict the distribution of neural responses to unseen stimuli, without the need for repeated presentations to learn stimulus-conditioned variability. We (2) apply our method on the activity of thousands of neurons in response to natural images, recorded via two-photon Calcium imaging from multiple areas of the mouse visual cortex. We demonstrate that our model outperforms current state-of-the-art methods in capturing the distribution of responses. Finally, we (3) show that our model infers latent state structures with meaningful relations to behavioral variables such as pupil dilation as well as other functional and anatomical properties of visual sensory neurons.

2 Methods

2.1 Models

Flow-based Factor Analysis model (FlowFA) 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 activity $p(\mathbf{r}|\mathbf{x})$ as

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

FlowFA 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 over $\mathbf{v}$ 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 effect of the stimulus $\mathbf{x}$ on the responses is captured by the mean of the FA distribution that depends on the stimulus, modeled as a deep network $\mathbf{f}_\theta(\mathbf{x}) \in \mathbb{R}^n$ with learnable parameters θ (Fig. 1a,b). We further include neuron-specific, independent noise $\varepsilon \sim \mathcal{N}(0, \Psi)$ where $\Psi \in \mathbb{R}^{n \times n}$ is a diagonal covariance matrix.

Since the flow model is a trainable 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 for an arbitrary stimulus.

In the model formulation presented here, we choose T_ϕ to act on each single dimension separately, i.e. $T_\phi(\mathbf{r}) = [T_{\phi_1}(r_1), ..., T_{\phi_n}(r_n)]^\top$. This choice results in a diagonal Jacobian which not only substantially simplifies the form of the determinant to $\det \nabla_\mathbf{r} T_\phi(\mathbf{r}) = \prod_{i=1}^n \frac{\partial T_{\phi_i}}{\partial r_i}$, but also allows us to easily compute conditionals and marginals (see appendix A for the details). This would not generally be possible for diffeomorphisms with a non-diagonal Jacobian.

Zero-Inflated Flow-based Factor Analysis model (ZIFFA) For two-photon Calcium imaging, a significant portion of inferred neural activity is zero, resulting in a sharp peak at zero in the response distribution (i.e. zero-inflated distribution) [43]. This zero-inflation is potentially a problem for the FlowFA model since the model would attempt to generate the peak at zero by mapping a large proportion of the Gaussian probability mass onto the "zero" responses, resulting in a poor fit to the response distribution. To avoid this, we extend FlowFA by modeling the zero responses with a separate peak (similar to Wei et al. [43]) and applying the FlowFA model to capture only the positive responses. We refer to this model as Zero-Inflated Flow-based Factor Analysis (ZIFFA). More specifically, ZIFFA is a mixture model that models neural responses below and above a threshold value ρ with two separate, non-overlapping distributions. To capture the peak at zero, the responses below the threshold (i.e. "zero" responses) are modeled by a uniform distribution, while FlowFA is used to capture responses above the threshold:

$$p(\mathbf{r}|\mathbf{x}) = \left(\prod_{\{i: r_i \leq \rho\}} \frac{1 - q_i\left(\mathbf{x}\right)}{\rho} \right) \cdot \left(\prod_{\{i: r_i > \rho\}} q_i\left(\mathbf{x}\right) \right) \cdot \mathcal{N}(T_\phi(\mathbf{r}_+); f_{\theta,+}(\mathbf{x}), \mathbf{C}_+\mathbf{C}_+^\top + \Psi_+) \cdot |\nabla T_\phi(\mathbf{r}_+)| \,, \tag{2}$$

where $q_i(\mathbf{x})$ is the probability of the response being above the threshold $\mathbf{x}$ modeled, jointly with the mean of the FA, as a function of the stimulus via a DNN $\mathbf{f}_\theta$ with learnable parameters θ. $\mathbf{r}_+$ and $f_{\theta,+}(\mathbf{x})$ are the sub-vectors, and $\mathbf{C}_+$ and Ψ_+ are the sub-matrices corresponding to responses above the threshold, and θ, $\mathbf{C}$, Ψ are the same as defined in Eq. (1). Refer to appendix B for the derivation.

Control models We compare the FA-based models against two control models used for neural system identification that assume independence among neurons with specific forms of marginal distributions inspired by existing work: (1) Poisson [18, 22] and (2) Zero-inflated Gamma (ZIG) [43]. To capture continuous neural responses measured with Calcium imaging, we relax the discrete Poisson distribution into a continuous distribution by assuming $r = \hat{r} + \epsilon$ where $\hat{r} \sim \text{Poisson}(\lambda)$ and $\epsilon \sim \text{Uniform}[0, 1)$. This yields the likelihood function

$$p_{\text{poiss}}(\mathbf{r}|\mathbf{x}) = \prod_i^n \frac{\lambda_i(\mathbf{x})^{\lfloor r_i \rfloor} e^{-\lambda_i(\mathbf{x})}}{\lfloor r_i \rfloor !}, \tag{3}$$

where $\lambda(\mathbf{x}) = \mathbf{f}_\theta(\mathbf{x})$ is the predicted firing rate of the neurons to input image $\mathbf{x}$ modeled as a DNN $\mathbf{f}_\theta$ with learnable parameters θ. The ZIG distribution is a mixture of a uniform and a gamma distribution separated at the value ρ with no overlap [43]:

$$p_{\text{ZIG}}(\mathbf{r}|\mathbf{x}) = \prod_i^n \left(\frac{1 - q_i(\mathbf{x})}{\rho} + \frac{q_i(\mathbf{x}) r_i^{\kappa_i - 1}}{\Gamma(\kappa_i)\nu_i(\mathbf{x})^{\kappa_i}} \exp\left(-\frac{r_i}{\nu_i(\mathbf{x})} \right) \right), \tag{4}$$

where $\nu_i(\mathbf{x})$ is the scale parameter of the gamma distribution, and $q_i(\mathbf{x})$ is same as in Eq. (2). To formulate ZIG as an image-computable model, $\nu_i(\mathbf{x})$ and $q_i(\mathbf{x})$ are jointly modeled using a DNN $\mathbf{f}_\theta$ with learnable parameters θ. Similar to Wei et al. [43], we let the shape parameter κ_i be neuron-specific, but independent of the input. Importantly, we used the same value for ρ in both ZIG and ZIFFA models.

Note that when the covariance matrix of the FA-based models is diagonal (i.e. 0-dimensional latent state), these models assume independence among neurons and their performance is directly comparable to the control models.

2.2 Model components

Deep convolutional neural network $\mathbf{f}_\theta$ We capture the stimulus-driven changes in the neuronal response distribution using a deep convolutional neural network $\mathbf{f}_\theta(\mathbf{x})$ 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 [44] convolution operation resulting in 64 feature channels, followed by batch normalization and ELU nonlinearity, 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. 1a). 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 experimentally measured cortical positions $\boldsymbol{\Delta}$ of the neurons, here we allow this mapping to take on a non-linear form to allow flips in the representation of the visual field as a function of cortical position (Fig. 1b). This is crucial to model cortex-to-visual space mappings for multiple brain areas, as the retinotopy of some areas are mirrored with respect to each other. During training, we apply L1 regularization to the readout feature weights and L2 regularization on the Laplace-filtered weights of the first convolution layer.

Normalizing flow T_ϕ We construct the marginal flow model $T_\phi = \text{affine} \circ \log \circ \text{affine} \circ \text{ELU} \circ \text{affine} \circ \text{ELU} \circ \text{affine} \circ \exp \circ \text{affine}$ from a set of monotonic functions $\{\text{affine}, \text{ELU}, \log, \exp\}$, of which only the affine transformation has learnable parameters. We restricted all the affine transformation layers to have positive scale, and additionally restricted the first affine layer to have a positive offset. For each neuron, we learn a separate marginal transformation T_{ϕ_i}. We compare the flow transformation against two common fixed transformations: square-root [16, 30] and Anscombe [45]. 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 specifically chose the components of T_ϕ such that these 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. For ZIFFA, we adjusted the formulation of the marginal flow T_ϕ such that the predicted neuronal responses remain above ρ, the boundary between the uniform and the FlowFA components of the mixture model, by replacing the first affine transformation in T_ϕ with a layer that only shifts by $-\rho$.

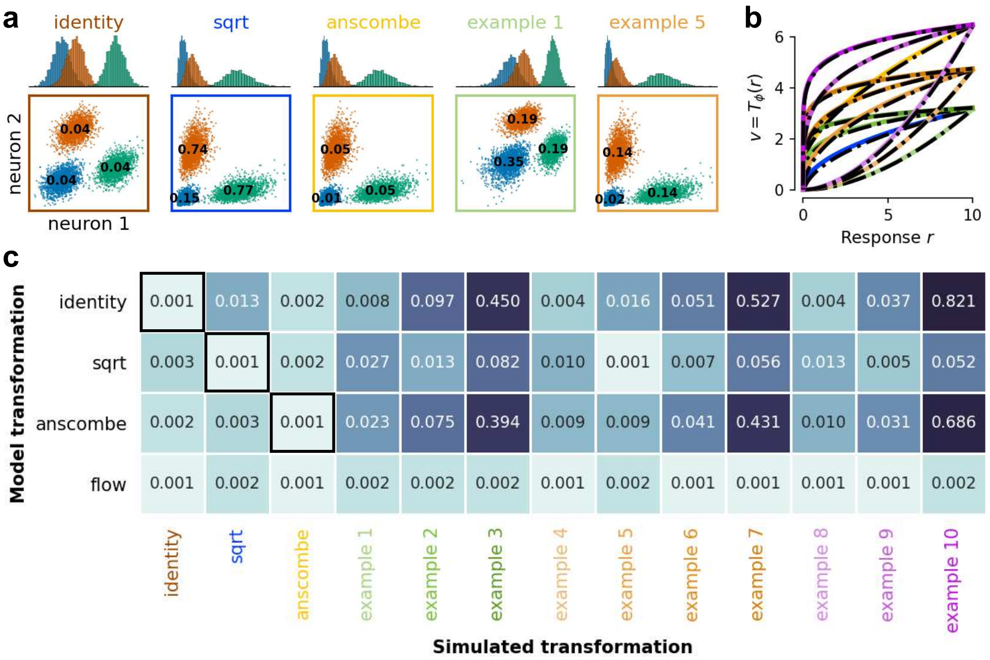
2.3 Neural and behavioral data

We recorded the response of neurons in mouse visual cortices (layer L2/3) to gray-scale natural images using a wide-field two-photon microscope [46] (see appendix C for details). In this study, we used two scans from two mice spanning three visual areas: primary visual cortex (V1) and lateromedial area (LM) in scan 1; V1 and posteromedial area (PM) in scan 2. A total of 2,867 V1 neurons and 907 LM neurons were recorded in scan 1; 5,029 V1 neurons and 3,343 PM neurons were recorded in scan 2. Among these, we used 1,000 V1 and 907 LM neurons from scan 1, and 1,000 V1 and 1,000 PM neurons from scan 2. For both scans, neurons were randomly selected if the area contained more than 1,000 neurons. We also recorded behavioral variables such as pupil dilation, simultaneously. The natural image stimuli were sampled from ImageNet [47], cropped to fit a monitor with 16:9 aspect ratio, and presented to the mice at a resolution of 0.53 ppd (pixels per degree of visual angle). A total of 6,000 images were shown in each scan, of which 1,000 images consist of 100 unique images each repeated 10 times 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.

2.4 Model fitting and evaluation

Fitting We trained all models end-to-end via gradient-based optimization to maximize the log-likelihood obtained from Eqs. (1), (2), (3) or (4) for the corresponding model, optimizing over all learnable parameters. To ensure that Ψ, the diagonal covariance matrix, stays positive-valued, we re-parameterized $\Psi = e^\nu$ and optimized ν instead. To find the best image-computable DNN models, we used Bayesian optimization [48] to find hyper-parameters that maximized the final log-likelihood of the trained model. Hyper-parameters include the learning rate and regularization coefficient on the readout weights. The log-likelihood used for scheduling learning rate, early stopping, and finding hyper-parameters was computed on the validation set. Additional details about training can be found in appendix D. The code can be found at `https://github.com/sinzlab/bashiri-et-al-2021`.

Evaluation We compared the FA-based models (ZIFFA, FlowFA, and FA with fixed transformations) 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 Eqs. (1), (2), (3), and (4), accordingly. For the correlation measure, we computed the Pearson correlation between the predicted and the measured responses of each neuron on the test set. For the FA-based models that may capture the statistical dependency (i.e. covariance) between neurons, we predicted the response of a given neuron conditioned on the responses of all other neurons recorded simultaneously on the trial. More specifically, given an image $\mathbf{x}$ and the response of all other neurons $\mathbf{r}_{\setminus i}$, we estimated the response of a neuron r_i to the image by computing the posterior mean of the neuron's response $\mathbb{E}[r_i|\mathbf{x}, \mathbf{r}_{\setminus i}]$. We refer to this measure as *conditional correlation* (see appendix E for details).



Model transformation	identity	sqrt	anscombe	example 1	example 2	example 3	example 4	example 5	example 6	example 7	example 8	example 9	example 10
identity	0.001	0.013	0.002	0.008	0.097	0.450	0.004	0.016	0.051	0.527	0.004	0.037	0.821
sqrt	0.003	0.001	0.002	0.027	0.013	0.082	0.010	0.001	0.007	0.056	0.013	0.005	0.052
anscombe	0.002	0.003	0.001	0.023	0.075	0.394	0.009	0.009	0.041	0.431	0.010	0.031	0.686
flow	0.001	0.002	0.001	0.002	0.002	0.002	0.001	0.002	0.001	0.001	0.001	0.001	0.002

Figure 2: FlowFA model recovers the underlying transformation. **a:** Simulated responses for 2 neurons under various transformations. Across all transformations, *transformed* responses were sampled from Gaussian distributions with differing means (indicated by the color of the samples) but identical covariance. The covariance between the two neurons is shown in black text. **b:** Transformations learned by the flow model are shown in black, overlaid on the ground-truth transformations. **c:** Performance of models with fixed or learned (flow) transformations (rows) trained on responses simulated with a 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 distributions, where 0 would correspond to a perfect fit.

3 Results

3.1 Model performance

FlowFA model faithfully recovers invertible transformations on synthetic data We first used synthetic data to illustrate that our FlowFA model with a learnable transformation can adequately learn and recover a wide variety of transformations resulting in different response distributions. To this end, we sampled 5,000 data points for 100 neurons from models with different ground-truth transformations (see appendix F for details on data generation). The invertible transformations (Example 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 (FixedFA) or a learnable flow-based (FlowFA) transformation. As expected, the models with a fixed transformation performed well if the data was generated with a similar transformation, but the performance suffered when the transformations differed (Fig. 2c, first three rows). In contrast, the FlowFA model was able to flexibly learn every underlying transformation (Fig. 2b) and effectively captured all distributions across all simulations (Fig. 2c, last row).

Flow-based models capture cortical response distribution well After demonstrating that the flow-based model can effectively fit a wide range of distributions, we used 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 2.3 for details). We trained the FA-based models (ZIFFA, FlowFA, and FixedFA) for different values of latent dimensions $k \in \{0, 1, 2, 3, 10\}$. We measured the model performance by computing the log-likelihood as well as the conditional correlations (see section 2.4).

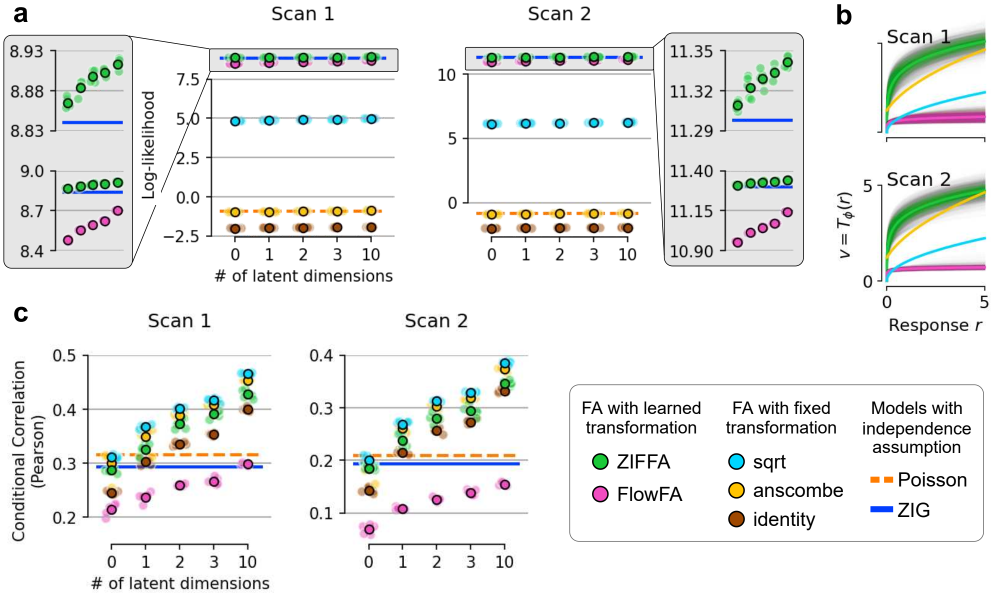


Figure 3: Comparison of models trained on the mouse visual cortical population responses to natural images. **a**: log-likelihood computed for models trained on scan 1 (left panel) and scan 2 (right panel). Values for both individual (lighter shade) and average (darker shade) performance of a model trained under various random seeds are shown. Gray block provides a zoomed-in view of the ZIFFA, FlowFA, and Zero-Inflated-Gamma (ZIG) models. **b**: Neuron-specific transformations learned by the flow-based models (ZIFFA in green, average across neurons in light green; FlowFA in pink, average across neurons in light pink) shown in comparison to fixed transformations. **c**: Conditional correlation. Format is similar to **a**.

The ZIFFA model outperformed all other models across all numbers of latent dimensions k in terms of log-likelihood (Fig. 3a). Furthermore, with increasing latent dimensions, the conditional correlation of the ZIFFA model improved significantly beyond the control models (Fig. 3c). Interestingly, we observed that the ZIFFA model exhibited slightly lower correlation performance compared to models with fixed transformations, reflecting that fitting models on likelihood does not necessarily yield optimal correlation. Importantly, the flow-based models outperformed all FixedFA models in terms of likelihood, which is corroborated by the fact that the learned transformation markedly differs from all fixed transformations and from one neuron to the other (Fig. 3b). Overall, the results suggest that the ZIFFA model is able to capture the (marginal) neural response distributions more accurately than other models (Fig. S2) while at the same time it learns and takes advantage of the statistical dependencies between neurons.

3.2 Uncovering biological insights from the trained model

Here, we explore the utility of our model in uncovering potential biological insights. All analyses were performed on the trained ZIFFA model with 3 latent dimensions.

Model-based visual area identification Several visual areas in mice show retinotopies that are "flipped" with respect to each other [49]. Intuitively, this means that if a point moves along the cortical surface, as it crosses the boundary between two "mirrored" areas, its counterpart in visual space would reverse its movement direction. As described in section 2.2, our model is equipped with a component network that predicts the RF location δ of each neuron in visual space as a function of its cortical location Δ (Fig. 1b). This network can be used to infer distinct visual cortical areas by detecting where the retinotopy "flips" with respect to the cortical position. To detect this flip we looked at the sign of the determinant of the Jacobian of the RF positions with respect to cortical positions $\det \frac{\partial \delta}{\partial \Delta}$. The sign can detect changes in the direction because (1) the sign of a determinant flips if one of the column or row vectors of the Jacobian matrix flips and (2) the determinant is invariant under rotation. When we compare distinct areas identified via the model to the experimentally identified areas, we find a very good match (Fig. 4a, left vs. right panels). To assess the quality of the learned

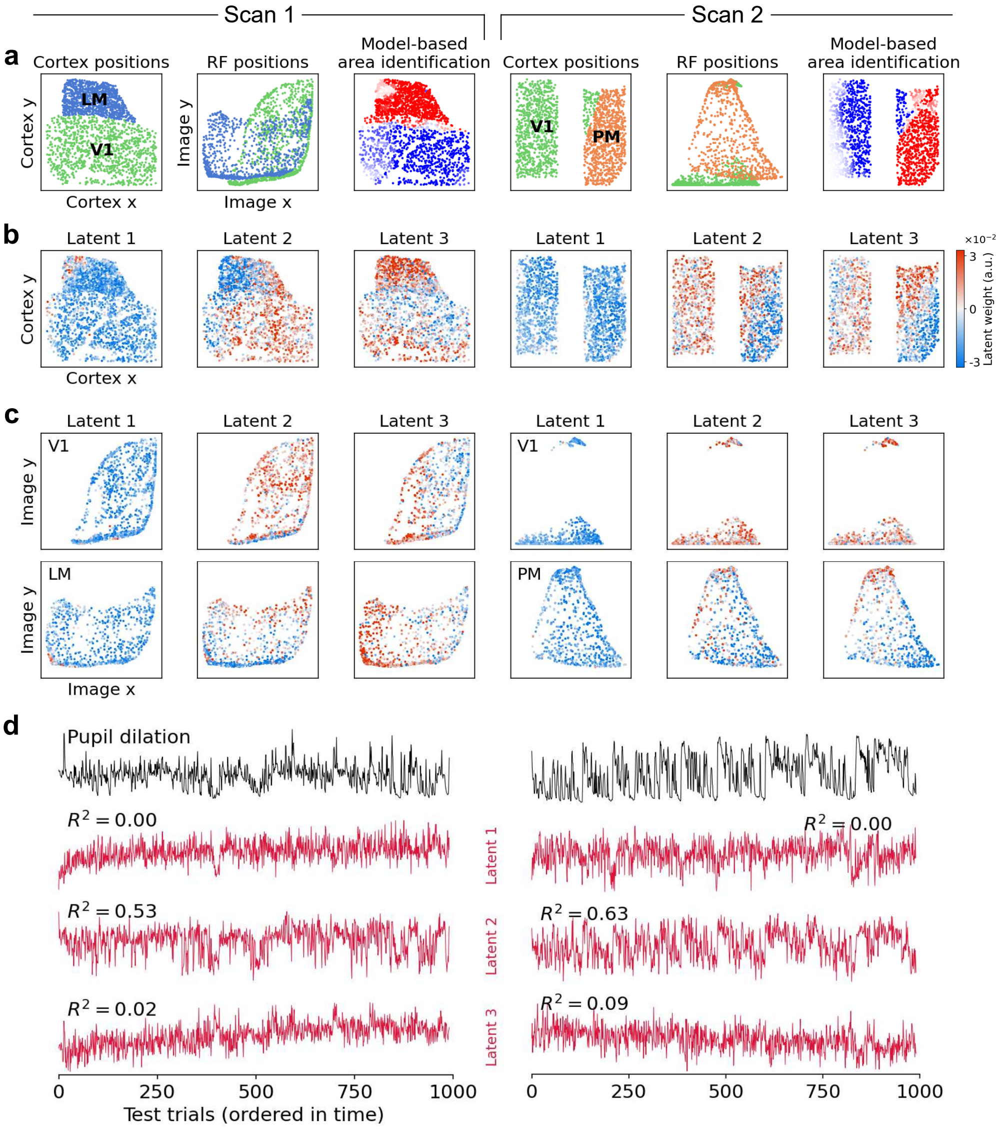


Figure 4: Analysis of the ZIFFA model with 3-dimensional latent state ($k = 3$). **a**: Model-based area identification from responses of visual sensory neurons to natural images. Left panel (Cortex positions): cortical position of the recorded neurons color-coded by experimentally identified areas (green: V1; blue: LM; orange: PM). Middle panel (RF positions): learned receptive field position for each neuron as a function of cortical positions color-coded by experimentally identified areas. Right panel (Model-based area identification): visual areas identified via the model by computing the determinant of the relative changes in RF position with respect to changes in cortical position; blue color shows negative determinant (i.e. mirrored visual field representation) and red color shows positive determinant (i.e. non-mirrored visual field representation). **b–c**: Distribution of the latent-to-neuron weights across cortical positions (**b**) and receptive field positions (**c**). **d**: 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.

mapping, we quantified how well our model can identify distinct visual brain areas via the sign of the determinant. Across models initialized and trained with different random seeds, the sign correctly classifies distinct brain areas with an accuracy of $84\% \pm 3.4\%$ (SEM) and $75\% \pm 7.7\%$ (SEM). Because the experimental methods to determine area assignment that we use as ground truth can be quite coarse, the actual accuracy could even be higher. This suggests that our model could in principle allow neuroscientists to identify distinct visual areas from responses to natural images alone, without the need for an extra experiment for area identification.

Inferred latent states and their functional and anatomical implications We next explored the latent states and how they relate to anatomy or behavior. For any particular trial, the FA-based models allow us to infer the most probable latent state $\mathbf{z}$ (MAP estimate), where the effect of each latent dimension on the neural population is captured by the factor loading matrix $\mathbf{C}$. However, as formulated in Eq. (1) and (2), interpreting the inferred latent states $\mathbf{z}$ can be difficult because the latent dimensions can be arbitrarily permuted and rotated (with corresponding changes in $\mathbf{C}$) without affecting the fit of the model. To facilitate interpretability of the inferred latent states, we follow a similar procedure used by Yu et al. [30] to extract *orthonormalized latent states* which are uniquely ordered by the amount of response variability each latent dimension accounts for (see appendix G for detailed explanation).

The orthonormalized latent states inferred from the ZIFFA model showed strong correlations with behavioral variables such as pupil dilation (Fig. 4d), as expected from previous works that use pupil dilation as a proxy for arousal and attention [50–54]. Interestingly, pupil dilation correlated most strongly with the second latent dimension in both scans with R^2 values of 0.53 ($p < 0.001$, two-tailed test for significance of correlation [55]) and 0.63 ($p < 0.001$) for scan 1 and scan 2, respectively, comparable to values previously reported [56]. To our surprise, this observation was consistent across models initialized and trained with different random seeds (Fig. S4b). To further quantify how well the latent states can jointly predict the pupil dilation, we regressed the pupil dilation against the latent states (Fig. S4a). The resulting R^2 values were 0.56 ($p < 0.001$) and 0.76 ($p < 0.001$) for scan 1 and scan 2, respectively. The high correlation between the latent states and the known surrogates of global brain state such as pupil dilation suggests that the latent model is able to learn meaningful dependencies and common factors in neural population.

Next, we explored whether the effect of the orthonormalized latent states on the neurons is related to their cortical or RF positions. To this end, we plotted the sign and magnitude of the weight mapping from the latent state to each neuron on the cortical position (Fig. 4b) or the RF positions of the neurons (Fig. 4c). We observed that the effect of some latent dimensions vary systematically across brain areas where the latent dimension has generally opposite effect on different areas (Fig. 4b: dimension 2 for both scans). In addition, some latent dimensions seemed to vary as a function of RF positions/retinotopy where a differential effect of the latent dimension is observed for both areas (Fig. 4c: dimension 3 for both scans). Interestingly, the first dimension which accounts for most of the shared variability in neural responses (refer to section G for more details) seemed to have a global effect that does not vary across different visual areas. These observations illustrate that 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.

While the result of the analyses we present here are promising, we would like to point out that all analyses are preliminary, and conclusive biological interpretations would require additional rigorous experiments and analyses.

4 Discussion

Getting the best of both worlds Two major components of the variability in the activity of cortical neurons 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. Here, we presented a model that combines state-of-the-art DNN-based models to predict stimulus-driven changes in neural activity with a simple, yet flexible, flow-based factor analysis model to account for correlated neural activity. This formulation allows us to evaluate the exact likelihood 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 obtained state-of-the-art performance in capturing neural response distribution while additionally yielding latent states that exhibit meaningful relations to anatomy and functional properties of visual sensory neurons.

Modeling zero-inflated response distribution Flow models use diffeomorphisms to map one distribution into another. However, diffeomorphisms cannot transform a single peak at 0—typically observed in neural responses recorded via Calcium imaging—into a smooth distribution such as Gaussian used in our model. The ZIFFA model avoids this problem by only transforming the positive part of the response with a diffeomorphism while explicitly capturing the peak at 0 via a uniform distribution as found in ZIG. Importantly, ZIFFA preserves all properties of the FlowFA model, while capturing the marginal distributions more accurately (Fig. S2), achieving a higher likelihood (Fig. 3), and learning more consistent and less step-like transformations (Fig. S3).

Dependency of noise correlation on the stimulus The presented flow-based models learn a nonlinear transformation between a simple distribution (Gaussian FA) and the neural response distribution. While the learned covariance structure on the "transformed" neural responses captured by the FA model does not vary with the stimulus and the stimulus is only used to shift the mean of the FA model, this is not true for samples from the FA model transformed back into "neural response space" because the nonlinear flow transformation can introduce changes in the covariance as the mean varies (Fig. 2a). This mean-dependent change in the covariance potentially allows the model to capture changes in the covariance structure based on stimulus through the nonlinear transformation. A possible extension of our model is an explicit dependence of the FA's covariance matrix on the stimulus, which would allow the model to capture more complex dependencies between the stimulus and covariance structure.

Comparison to related methods Our approach in capturing stimulus-conditioned variability is related to many existing approaches, or can be seen as a generalization thereof, while being computationally easier to handle at the same time. Recently, Keeley et al. [35] captured the trial-by-trial fluctuations by modeling the stimulus-specific and trial-specific latents via Factor Analysis (FA) models much like in our model. Importantly, while we capture the dependence of the stimulus-specific latents on the stimulus explicitly via a trained DNN, they inferred it from repeated presentations of the stimulus. Furthermore, the final Poisson distribution used to map from the latents to the distribution of neurons can be captured in our model via the flow-based transformation (e.g. inverse Anscombe) that maps Gaussian-distributed latents into a continuous approximation of a Poisson distribution. Moreover, the use of FA in combination with the marginal flow makes our approach related to copula-based distribution approximation and related approaches [28, 29, 57]. However, by explicitly limiting the stimulus dependence to occur via the shift in the mean of the FA model along with flow-based transformation of responses, we avoid the reliance on the repeated presentations of the stimuli [29] or highly constrained forms of the marginal distribution [28].

Limitations and future extensions As discussed above, our flow-based approach generalizes several existing methods to capture stimulus-conditioned variability of neural responses while being computationally more tractable. This allows us to train our models end-to-end directly on the likelihood via common gradient-based optimization algorithms. Within this general framework, we presented a specific case where we learned neuron-specific stimulus-independent transformations, mapping responses into a FA model whose mean varies with the stimulus. As noted earlier, for each stimulus, this approach closely parallels Gaussian copula and thus shares much of the same limitations. Also, the fact that stimulus-dependent changes in the covariance structure only occur through the learned transformation implies that the model can only capture changes in the covariance structure that varies with the mean (a limitation shared with many of the existing models). That being said, we believe that our general approach of flow-based modeling of neural response distributions allows for several generalizations that would overcome these limitations. Examples include an explicit dependence of the FA's covariance matrix on the stimulus, as well as the usage of richer, potentially stimulus-dependent, learnable transformations.

Broader impact Accurate models of neural variability such as the one presented here can lead to deeper scientific insights and understanding of how brains perceive and compute with sensory information, and can eventually also provide insights into how neurological and psychological disorders may 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*.

Acknowledgments and Disclosure of Funding

We thank all reviewers for their constructive and thoughtful feedback. Furthermore, we thank Alexander Ecker, Konstantin Willeke, Arne Nix, Christoph-Benjamin Blessing, and Emmanouil Froudaraki for comments and discussions. MB is supported by the International Max Planck Research School for Intelligent Systems. KKL is funded by the German Federal Ministry of Education and Research through the Tübingen AI Center (FKZ: 01IS18039A). FHS is supported by the Carl-Zeiss-Stiftung and acknowledges the support of the DFG Cluster of Excellence "Machine Learning – New Perspectives for Science", EXC 2064/1, project number 390727645. This work was supported by an AWS Machine Learning research award to FHS. Supported by the Intelligence Advanced Research Projects Activity via Department of Interior/Interior Business Center contract number D16PC00003.

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] 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.

[24] Pouya Bashivan, Kohitij Kar, and James J DiCarlo. Neural population control via deep image synthesis. *Science*, 364(6439):eaav9436, May 2019.

[25] 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.

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

[27] Cian O'Donnell, J Tiago Gonçalves, Nick Whiteley, Carlos Portera-Cailliau, and Terrence J Sejnowski. The population tracking model: A simple, scalable statistical model for neural population data. *Neural Comput.*, 29(1):50–93, January 2017.

[28] 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.

[29] Oleksandr Sorochynskyi, Stéphane Deny, Olivier Marre, and Ulisse Ferrari. Predicting synchronous firing of large neural populations from sequential recordings. *PLoS Comput. Biol.*, 17 (1):e1008501, January 2021.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] Stephen L Keeley, Mikio C Aoi, Yiyi Yu, Spencer L Smith, and Jonathan W Pillow. Identifying signal and noise structure in neural population activity with gaussian process factor models. July 2020.

[36] 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.

[37] 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.

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

[39] 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.

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

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

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

[43] 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.

[44] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[45] 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.

[46] 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.

[47] 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.

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

[49] 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.

[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] Student. Probable error of a correlation coefficient. *Biometrika*, pages 302–310, 1908.

[56] Carsen Stringer, Marius Pachitariu, Nicholas Steinmetz, Charu Bai Reddy, Matteo Carandini, and Kenneth D Harris. Spontaneous behaviors drive multidimensional, brainwide activity. *Science*, 364(6437), 2019.

[57] Jakob H Macke, Philipp Berens, Alexander S Ecker, Andreas S Tolias, and Matthias Bethge. Generating spike trains with specified correlation coefficients. *Neural Comput.*, 21(2):397–423, February 2009.

[58] 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.

[59] 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.

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

[61] 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.

[62] 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`.

[63] 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`.

A Expression for the marginal and conditional distributions

Here we derive and show that the marginal and conditional distributions in the neural response space can be straightforwardly expressed in terms of the corresponding marginal and conditional distributions in the transformed response space when the transformation function T is separable. Consider partitioning neurons into two mutually-exclusive subgroups $\mathbf{r}^{(1)}$ and $\mathbf{r}^{(2)}$. Furthermore assume that the transformation function factorizes over these two subgroups such that $T(\mathbf{r}) = [T_1(\mathbf{r}^{(1)})^\top, T_2(\mathbf{r}^{(2)})^\top]^\top = [\mathbf{v}^{(1)\top}, \mathbf{v}^{(2)\top}]^\top = \mathbf{v}$, for some constituent diffeomorphisms T_1 and T_2. Given this,

$$\begin{aligned} p_r\left(\mathbf{r}|x\right) &= p_r\left(\mathbf{r}^{(1)},\mathbf{r}^{(2)}\middle|x\right)\\ &= p_v\left(T_1\left(\mathbf{r}^{(1)}\right),T_2\left(\mathbf{r}^{(2)}\right)\middle|x\right)\cdot\left|\det\nabla_{\mathbf{r}^{(1)}}T_1\left(\mathbf{r}^{(1)}\right)\right|\cdot\left|\det\nabla_{\mathbf{r}^{(2)}}T_2\left(\mathbf{r}^{(2)}\right)\right|,\end{aligned}$$

where p_r and p_v denote the densities for the respective random variables. Then the marginal over $\mathbf{r}^{(1)}$ can be expressed as follows:

$$\begin{aligned} p_r\left(\mathbf{r}^{(1)}\middle|x\right) &= \int_{\mathbf{r}^{(2)}} p_r\left(\mathbf{r}^{(1)},\mathbf{r}^{(2)}\middle|x\right)\,\mathrm{d}\mathbf{r}^{(2)}\\ &= \int_{\mathbf{r}^{(2)}} p_v\left(T_1\left(\mathbf{r}^{(1)}\right),T_2\left(\mathbf{r}^{(2)}\right)\middle|x\right)\cdot\left|\det\nabla_{\mathbf{r}^{(1)}}T_1\left(\mathbf{r}^{(1)}\right)\right|\cdot\left|\det\nabla_{\mathbf{r}^{(2)}}T_2\left(\mathbf{r}^{(2)}\right)\right|\,\mathrm{d}\mathbf{r}^{(2)}.\end{aligned}$$

We now employ the change of variables with:

$$\begin{aligned} \mathbf{r}^{(2)} &= T_2^{-1}(\mathbf{v}^{(2)})\\ \therefore\ \mathrm{d}\mathbf{r}^{(2)} &= \left|\nabla_{\mathbf{r}^{(2)}}T_2(\mathbf{r}^{(2)})\right|^{-1}\mathrm{d}\mathbf{v}^{(2)},\end{aligned}$$

yielding:

$$\begin{aligned} p_r\left(\mathbf{r}^{(1)}\middle|x\right) &= \int_{\mathbf{v}^{(2)}} p_v\left(T_1\left(\mathbf{r}^{(1)}\right),\mathbf{v}^{(2)}\middle|x\right)\cdot\left|\det\nabla_{\mathbf{r}^{(1)}}T_1\left(\mathbf{r}^{(1)}\right)\right|\,\mathrm{d}\mathbf{v}^{(2)}\\ &= \left|\det\nabla_{\mathbf{r}^{(1)}}T_1\left(\mathbf{r}^{(1)}\right)\right|\cdot\int_{\mathbf{v}^{(2)}} p_v\left(T_1\left(\mathbf{r}^{(1)}\right),\mathbf{v}^{(2)}\middle|x\right)\,\mathrm{d}\mathbf{v}^{(2)}\\ &= \left|\det\nabla_{\mathbf{r}^{(1)}}T_1\left(\mathbf{r}^{(1)}\right)\right|\cdot p_v\left(T_1\left(\mathbf{r}^{(1)}\right)\middle|x\right)\end{aligned}$$

Hence, the marginal over $\mathbf{r}^{(1)}$ can be simply expressed in terms of marginal distribution over the transformed variable $T_1(\mathbf{r}^{(1)})$. Finally, we can write the conditional distribution over original responses in terms of the conditionals over the transformed variables:

$$\begin{aligned} p_r\left(\mathbf{r}^{(1)}\middle|\mathbf{r}^{(2)},x\right) &= \frac{p_r\left(\mathbf{r}^{(1)},\mathbf{r}^{(2)}|x\right)}{p_r\left(\mathbf{r}^{(2)}|x\right)}\\ &= \frac{p_v\left(T_1\left(\mathbf{r}^{(1)}\right),T_2\left(\mathbf{r}^{(2)}\right)|x\right)\cdot\left|\det\nabla_{\mathbf{r}^{(1)}}T_1\left(\mathbf{r}^{(1)}\right)\right|\cdot\cancel{\left|\det\nabla_{\mathbf{r}^{(2)}}T_2\left(\mathbf{r}^{(2)}\right)\right|}}{\cancel{\left|\det\nabla_{\mathbf{r}^{(2)}}T_2\left(\mathbf{r}^{(2)}\right)\right|}\cdot p_v\left(T_2\left(\mathbf{r}^{(2)}\right)|x\right)}\\ &= \left|\det\nabla_{\mathbf{r}^{(1)}}T_1\left(\mathbf{r}^{(1)}\right)\right|\frac{p_v\left(T_1\left(\mathbf{r}^{(1)}\right),T_2\left(\mathbf{r}^{(2)}\right)|x\right)}{p_v\left(T_2\left(\mathbf{r}^{(2)}\right)|x\right)}\\ &= \left|\det\nabla_{\mathbf{r}^{(1)}}T_1\left(\mathbf{r}^{(1)}\right)\right|p_v\left(T_1\left(\mathbf{r}^{(1)}\right)\middle|T_2\left(\mathbf{r}^{(2)}\right),x\right).\end{aligned}$$

Note again that in order for the expressions for the conditionals and marginals to cleanly reduce, it is essential that the transformation $T\left(\cdot\right)$ is separable over the two groups of neurons.

B Zero-Inflated Flow-based Factor Analysis (ZIFFA)

Joint distribution Here, we provide the derivation of the joint distribution $p(\mathbf{r}|\mathbf{x})$ of the ZIFFA model. Let $\mathbf{m} \in \{0,1\}^n$ denote whether a neuron has a response r_i below or above the threshold ρ as indicated by $m_i = 0$ or $m_i = 1$, respectively. For a given assignment of $\mathbf{m}$, we model the density of a response vector $\mathbf{r} \in \mathbb{R}^n_{\geq 0}$ as a product of (1) a uniform distribution between 0 and threshold ρ and (2) a joint FlowFA model for above threshold responses. Accordingly, the conditional distribution can be expressed as follows:

$$
\begin{aligned}
p(\mathbf{r}|\mathbf{x},\mathbf{m}) = & \underbrace{\left(\prod_{\{i:m_i=0\}} [\![0 \leq r_i \leq \rho]\!] \cdot \rho^{-1}\right)}_{\text{Uniform part for all } r_i \text{ with } m_i=0} \cdot \\
& \underbrace{\left(\prod_{\{i:m_i=1\}} [\![\rho < r_i]\!]\right) \cdot \mathcal{N}(T_\phi(\mathbf{r}_+); f_{\theta,+}(\mathbf{x}), \mathbf{C}_+\mathbf{C}_+^\top + \Psi_+) \cdot |\nabla T_\phi(\mathbf{r}_+)|,}_{\text{FlowFA part for all } r_i \text{ with } m_i=1}
\end{aligned}
$$

where $\mathbf{r}_+$ and $f_{\theta,+}(\mathbf{x})$ are the sub-vectors corresponding to responses that are above the threshold. Also, $\mathbf{C}_+$ and Ψ_+ are sub-matrices of $\mathbf{C}$ and Ψ, respectively, only containing entries corresponding to the neurons with above threshold response. We choose T_ϕ such that $T_\phi^{-1}(\mathbf{v}) > \rho$, where $\mathbf{v} = T_\phi(\mathbf{r})$. We use a slight abuse of notation and determine the size of $T_\phi(\mathbf{r}_+)$ by the dimensionality of its input $\mathbf{r}_+$. Here $[\![A]\!]$ denotes the indicator function for the set A. Note that (1) this is a proper density on $\mathbb{R}^n_{\geq 0}$ since it remains non-negative and integrates to one, and that (2) all population responses $\mathbf{r}$ that do not agree with $\mathbf{m}$ (i.e. $m_i = 0$ and $r_i > \rho$, and vice versa) have zero density since one of the indicator functions in the product will be zero (i.e. they enforce $\mathbf{m}$). To get $p(\mathbf{r}|\mathbf{x})$, we marginalize out $\mathbf{m}$. To this end, we model the probability of each m_i independently as a function $q_i(\mathbf{x})$ of the image $\mathbf{x}$. This yields

$$
p(\mathbf{m}|\mathbf{x}) = \prod_{i=1}^{n} q_i(\mathbf{x})^{m_i} \left(1 - q_i(\mathbf{x})\right)^{1-m_i},
$$

and

$$
\begin{aligned}
p(\mathbf{r}|\mathbf{x}) &= \sum_{\mathbf{m}\in\{0,1\}^n} p(\mathbf{r}|\mathbf{x},\mathbf{m}) \cdot p(\mathbf{m}|\mathbf{x}) \\
&= \left(\prod_{\{i:r_i\leq\rho\}} \frac{1-q_i(\mathbf{x})}{\rho}\right) \cdot \\
&\quad \left(\prod_{\{i:r_i>\rho\}} q_i(\mathbf{x})\right) \cdot \mathcal{N}(T_\phi(\mathbf{r}_+); f_{\theta,+}(\mathbf{x}), \mathbf{C}_+\mathbf{C}_+^\top + \Psi_+) \cdot |\nabla T_\phi(\mathbf{r}_+)| \,.
\end{aligned}
$$

Note that all $2^n - 1$ mixture components whose $\mathbf{m}$ are not in agreement with $\mathbf{r}$ are zero, which leaves only one single mixture component in the end.

Conditional distribution The conditional distribution over i^{th} neuron's response r_i given the response of all other neurons $\mathbf{r}_{\setminus i}$, can be computed as:

$$
\begin{aligned}
p(r_i \mid \mathbf{r}_{\setminus i}, \mathbf{x}) &= \frac{p(\mathbf{r} \mid \mathbf{x})}{p(\mathbf{r}_{\setminus i} \mid \mathbf{x})} \\
&= \begin{cases} (1 - q_i(\mathbf{x})) \cdot \rho^{-1} & \text{if } r_i \leq \rho \\ q_i(\mathbf{x}) \cdot \frac{\mathcal{N}(T_\phi(\mathbf{r}_+); f_{\theta,+}(\mathbf{x}), \mathbf{C}_+\mathbf{C}_+^\top + \Psi_+) \cdot |\nabla T_\phi(\mathbf{r}_+)|}{\mathcal{N}(T_\phi(\mathbf{r}_{+\setminus i}); f_{\theta,+\setminus i}(\mathbf{x}), \mathbf{C}_{+\setminus i}\mathbf{C}_{+\setminus i}^\top + \Psi_{+\setminus i}) \cdot \left|\nabla T_\phi(\mathbf{r}_{+\setminus \mathbf{i}})\right|} & \text{if } r_i > \rho, \end{cases}
\end{aligned}
$$

where subscript $+\setminus i$ is used to denote all neurons with responses above threshold except for the i^{th} neuron. While conditioning does not change the distribution over the responses below the threshold ρ, for the responses above the threshold, the conditional distribution is computed as the fraction of joint distribution of all neurons $p(\mathbf{r}|\mathbf{x})$ over the joint distribution of all neurons except the target neuron $p(\mathbf{r}_{\setminus i}, \mathbf{x})$. This fraction of the two Gaussian distributions is equivalent to a Gaussian distribution over the response of the target neuron i where the mean and variance are computed conditioned on other neurons $\setminus i$:

$$\frac{\mathcal{N}(T_\phi(\mathbf{r}_+); f_{\theta,+}(\mathbf{x}), \mathbf{C}_+\mathbf{C}_+^\top + \Psi_+)}{\mathcal{N}(T_\phi(\mathbf{r}_{+\setminus i}); f_{\theta,+\setminus i}(\mathbf{x}), \mathbf{C}_{+\setminus i}\mathbf{C}_{+\setminus i}^\top + \Psi_{+\setminus i})} = \mathcal{N}(T_\phi(r_i); \mu_i, \sigma_i^2),$$

where μ_i and σ_i^2 are the posterior mean and variance, respectively, of the i^{th} neuron's transformed response conditioned on the stimulus $\mathbf{x}$ and transformed responses of other neurons $T_\phi(\mathbf{r}_{+\setminus i})$. These quantities can be straightforwardly computed from the FA model as follows:

$$\begin{aligned}
\mu_i &= f_{\theta,+,i}(\mathbf{x}) + \mathbf{\Sigma}_{+,i,\setminus i}\mathbf{\Sigma}_{+,\setminus i,\setminus i}^{-1}(T_\phi(\mathbf{r}_{+\setminus i}) - \mathbf{f}_{\theta,+,\setminus i}(\mathbf{x})) \\
\sigma_i^2 &= \Sigma_{+,i,i} + \mathbf{\Sigma}_{+,i,\setminus i}\mathbf{\Sigma}_{+,\setminus i,\setminus i}^{-1}\mathbf{\Sigma}_{+,i,\setminus i}^\top,
\end{aligned}$$

where $\mathbf{\Sigma} = \mathbf{C}\mathbf{C}^\top + \Psi$ and $\mathbf{\Sigma}_+ = \mathbf{C}_+\mathbf{C}_+^\top + \Psi_+$.

It is worth noting that the expressions for the conditionals cleanly reduce only when T_ϕ is separable for each neuron (see appendix A for derivations).

C Details on data recording and stimulation

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 as previously described [49, 58]. We selected cells based on a classifier for somata on the segmented cell masks and deconvolved their fluorescence traces using the CNMF algorithm [59].

Images were presented for 500 ms followed by a blank screen with a random duration uniformly distributed between 300 and 500 ms. After spike inference from Calcium data, the neural responses were extracted as the accumulated activity of each neuron between 50 and 550 ms after stimulus onset. All behavior traces (i.e. pupil dilation and running speed) 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.

D Additional details about model training

The models were trained end-to-end via gradient-based optimization to maximize the log-likelihood obtained from Eq. (1), (2), (3) or (4) for the corresponding model, optimizing over all parameters of the model. For optimization, we used Adam [60] with (i) an early stopping mechanism [61] that would stop the training if the log-likelihood 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 log-likelihood does not improve for ten training iterations.

To find the best image-computable model, we used Bayesian optimization [48] to find hyper-parameters that optimized the final log-likelihood (explained in section 2.4) of the trained model. Hyper-parameters included the learning rate and the regularization coefficient on the readout weights. The ZIFFA and ZIG models included the zero-threshold parameter ρ as an additional hyper-parameter. To find ρ, we experimented with several candidate values and chose the value which resulted in the highest score for the ZIG model, and used the same value for the ZIFFA model.

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 ZIFFA model takes approximately 2–3 hours to train whereas all other models take 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 [62] and NumPy [63] packages.

E Computation of conditional response predictions

We estimated the posterior mean of the neuron's responses to an image $\mathbf{x}$ conditioned on the responses of other neurons via Monte Carlo approximation. To achieve this, we first drew samples from the posterior based on the learned FA model, yielding samples in the space of the transformed responses. We then inverse-transformed these samples to yield samples in the space of the neural responses. Subsequently, we computed the average across these samples.

More specifically, for the FA-based models (except ZIFFA, see below), the posterior mean of the neuron's original response to image $\mathbf{x}$ was computed as $\mathbb{E}[r_i|\mathbf{x}, \mathbf{r}_{\setminus i}] = \frac{1}{N}\sum_j^N T^{-1}_{\phi,i}(\mathbf{s}_i^{(j)})$ where $\mathbf{s}_i^{(j)} \sim \mathcal{N}(\mathbb{E}[v_i|\mathbf{x}, \mathbf{v}_{\setminus i}], \sigma_i^2)$. $\mathbb{E}[v_i|\mathbf{x}, \mathbf{v}_{\setminus i}]$ and σ_i^2 are the posterior mean and variance, respectively, of the i^{th} neuron's transformed response conditioned on the stimulus $\mathbf{x}$ and transformed responses of other neurons $\mathbf{v}_{\setminus i} = T_\phi(\mathbf{r}_{\setminus i})$. These quantities can be straightforwardly computed from the FA model as follows:

$$\begin{aligned}
\mathbb{E}[v_i|\mathbf{x}, \mathbf{v}_{\setminus i}] &= f_{\theta,i}(\mathbf{x}) + \Sigma_{i,\setminus i}\boldsymbol{\Sigma}^{-1}_{\setminus i,\setminus i}(T_\phi(\mathbf{r}_{\setminus i}) - \mathbf{f}_{\theta,\setminus i}(\mathbf{x})), \\
\sigma_i^2 &= \Sigma_{i,i} + \Sigma_{i,\setminus i}\boldsymbol{\Sigma}^{-1}_{\setminus i,\setminus i}\Sigma^{\top}_{i,\setminus i},
\end{aligned}$$

where $\boldsymbol{\Sigma} = \mathbf{C}\mathbf{C}^\top + \Psi$.

For the ZIFFA model, the procedure for posterior mean computation is almost identical to the procedure explained above with two differences: 1) when computing the posterior mean and variance of the neuron's transformed response, we condition only on other neurons who exhibit above threshold responses $\mathbf{r}_{+\setminus i}$ (refer to appendix B for details), and 2) the posterior mean in the neural response space is computed as the mixture of the mean of the two mixture model components:

$$\mathbb{E}[r_i|\mathbf{x}, \mathbf{r}_{\setminus i}] = (1 - q_i(\mathbf{x})) \cdot \frac{\rho}{2} + q_i(\mathbf{x}) \cdot \mathbb{E}[r_i|\mathbf{x}, \mathbf{r}_{+\setminus i}].$$

F Synthetic data generation

We generated 5,000 samples from a correlated 100-d Gaussian distribution, corresponding to the transformed responses $\mathbf{v}$ of 100 neurons. The covariance matrix of the Gaussian distribution took the form $CC^{\top} + \Psi$, corresponding to that of FA models. $CC^{\top}$ was of rank 4 with $C \in \mathbb{R}^{100\times4}$, where the choice of the rank was arbitrary. To ensure that generated Gaussian samples (1) fall in a range where the transformation is invertible and that they (2) cover the most nonlinear part of the transformation, we kept the variances and covariances relatively small and sampled the mean for each neuron in a transform-specific fashion. The entries of C were sampled uniformly between 0.02 and 0.07, and the diagonal entries of Ψ were sampled uniformly between 0.002 and 0.01. We further imposed stronger or weaker correlations between selected neurons by scaling the corresponding entries of the full covariance matrix either by 1.5 or 0.2. The mean for each neuron (in the transformed response space) was uniformly sampled between a transform-specific minimum and maximum value. The transform-specific minimum value was computed as $T(\epsilon) + \alpha \cdot \max(CC^{\top} + \Psi)$ where ϵ was a small value (10^{-12}) close to zero and α took on a transform-specific value summarized in Table 1. The transform-specific maximum value was computed as $T(10)$. Once the Gaussian samples were generated for each transformation function, the samples were inverse-transformed via the corresponding T^{-1} into the simulated neural responses. The code used to generate simulated data can be found at `https://github.com/sinzlab/bashiri-et-al-2021`.

Table 1: transform-specific α values

T:	identity	sqrt	anscombe	example 1	example 2	example 3	example 4
α:	1.0	3.0	2.0	1.5	3.0	3.0	1.0
T:	example5	example 6	example 7	example 8	example 9	example 10	
α:	3.0	3.0	3.0	1.0	3.0	3.0	

G Computing orthonormalized latent states

We extract latent states from the FA-based model by computing the posterior mean $\mathbb{E}[\mathbf{z}|\mathbf{x},\mathbf{r}]$. While the relationship between the latent states $\mathbf{z}$ and the neural responses $\mathbf{r}$ is well defined via the model relationship $\mathbf{r} = T_\phi^{-1}(\mathbf{f}_\theta(\mathbf{x}) + \mathbf{C}\mathbf{z} + \epsilon)$, the factor loading matrix $\mathbf{C}$ can only be uniquely determined up to an arbitrary orthogonal transformation. That is, given $\mathbf{z} \sim \mathcal{N}(0, I_k)$, we can transform the factor loading matrix $\mathbf{C}$ and $\mathbf{z}$ by any arbitrary orthogonal transform matrix $\mathbf{R}$ to yield $\mathbf{C}' = \mathbf{C}\mathbf{R}$ and $\mathbf{z}' = \mathbf{R}^\top\mathbf{z}$. The resultant alternative definition of $\mathbf{z}'$ along with $\mathbf{C}'$ would yield identical fit to the neural responses since $\mathbf{C}'\mathbf{z}' = \mathbf{C}\mathbf{R}\mathbf{R}^\top\mathbf{z} = \mathbf{C}\mathbf{z}$ and $\mathbf{z}' \sim \mathcal{N}(0, I_k)$. Furthermore, the inferred latent states $\mathbf{z}$ are not necessarily ordered by how much neural variability they account for. In fact, the order of the latent states are arbitrary, and this can be seen by noting that a permutation matrix is an example of an orthogonal transformation. Combined with an additional observation that the columns of $\mathbf{C}$ are not guaranteed to be mutually orthogonal, interpreting the inferred latent states $\mathbf{z}$ is difficult and quite arbitrary.

To address this issue, we follow a similar approach to Yu et al. [30]. Briefly, we orthonormalize the columns of $\mathbf{C}$ by applying the singular value decomposition to the learned $\mathbf{C}$ which yields $\mathbf{C} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$. As a result, $\mathbf{C}\mathbf{z}$ can be re-written as $\mathbf{C}\mathbf{z} = \mathbf{U}(\mathbf{D}\mathbf{V}^\top\mathbf{z}) = \mathbf{U}\tilde{\mathbf{z}}$ where $\tilde{\mathbf{z}} \equiv \mathbf{D}\mathbf{V}^\top\mathbf{z}$ is the *orthonormalized latent state*. Consequently, instead of visualizing the MAP of $\mathbf{z}$, $\mathbb{E}[\mathbf{z}|\mathbf{x},\mathbf{r}]$, we would visualize $\mathbf{D}\mathbf{V}^\top\mathbb{E}[\mathbf{z}|\mathbf{x},\mathbf{r}]$. This approach incurs multiple advantages. Firstly, while the elements of $\mathbf{z}$ (and corresponding columns of $\mathbf{C}$) have no particular order, the elements of $\tilde{\mathbf{z}}$ (and corresponding columns of $\mathbf{U}$) are ordered by the amount of data variance they explain. Therefore, the inferred latent states are ordered by their contribution in explaining the variance observed in neural activity, resulting in more intuitive and interpretable latent states. Secondly, when the singular values are non-zero and non-repeating, the method recovers a unique latent state $\tilde{\mathbf{z}}$ for $\mathbf{C}' \equiv \mathbf{C}\mathbf{R}$ and $\mathbf{z}' \equiv \mathbf{R}^\top\mathbf{z}$ regardless of $\mathbf{R}$. This can be seen from the fact that singular value decomposition of $\mathbf{C}'$ is given by $\mathbf{C}' = \mathbf{U}\mathbf{D}\mathbf{V}'^\top$ where $\mathbf{V}' = \mathbf{R}^\top\mathbf{V}$, therefore

$$\begin{aligned}\tilde{\mathbf{z}}' &\equiv \mathbf{D}\mathbf{V}'^\top\mathbf{z}' \\ &= \mathbf{D}\mathbf{V}^\top\mathbf{R}\mathbf{R}^\top\mathbf{z} \\ &= \mathbf{D}\mathbf{V}^\top\mathbf{z} \\ &= \tilde{\mathbf{z}}.\end{aligned}$$

H Supplementary Figures

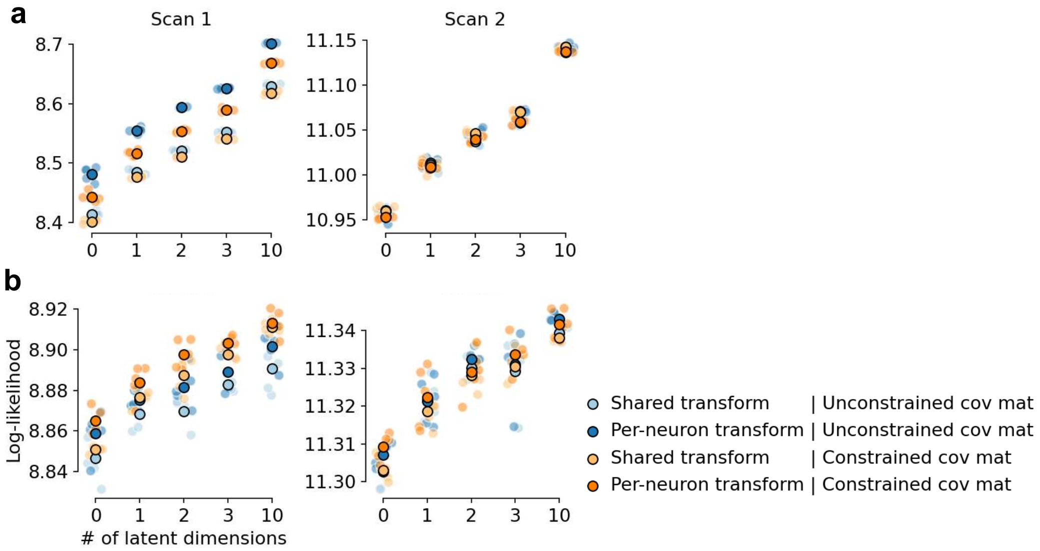


Figure S1: Comparison of flow-based models with different model configurations. These configurations include: 1) using a shared vs neuron-specific flow transformation, and 2) unconstrained vs constrained covariance matrix of the FA. The transformation T_ϕ could be defined such that a single flow transformation is shared among all neurons or it could be defined such that it contains neuron-specific parameters resulting in neuron-specific transformations (for details refer to section 2.2). As expected, per-neuron transformation (darker color) seem to results in a higher likelihood. The constrain imposed on the covariance matrix was used to ensure that the marginals have unit variance (i.e. a correlation matrix). While unconstrained covariance matrix (blue color) works best for the FlowFA model, the ZIFFA model with constrained covariance matrix (orange color) generally results in highest likelihood. **a**: FlowFA model. **b**: ZIFFA model.

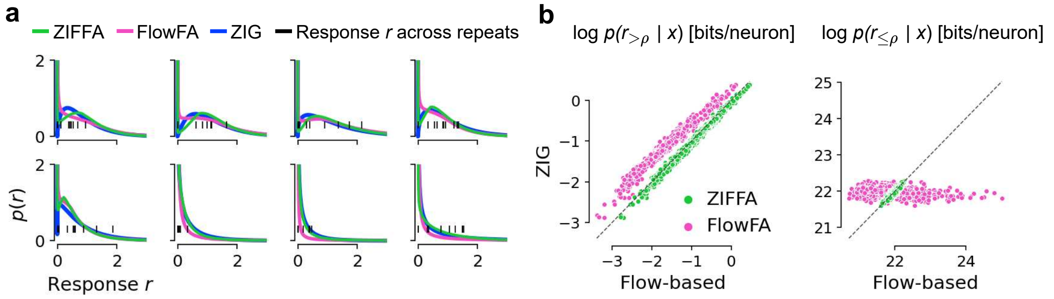


Figure S2: Comparison of the learned density by the ZIFFA, FlowFA, and ZIG models. **a**: Example marginal distribution of responses of 8 sample neurons to the repeated presentations of an image from the test set and the corresponding fits of ZIFFA, FlowFA, and ZIG. While all three models peak at zero, the FlowFA puts relatively little probability mass on positive responses $\mathbf{r}_{>\rho} = \{r_i | r_i(\mathbf{x}) > \rho\}$. **b**: Flow-based models vs ZIG log-likelihood in bits/neuron for positive responses $\mathbf{r}_{>\rho}$ and "zero" responses $\mathbf{r}_{\leq\rho}$, respectively. Each point is a single trial. Compared to ZIFFA and ZIG, FlowFA model seems to put less mass on responses $\mathbf{r}_{>\rho}$ and, for many trials, more mass on responses $\mathbf{r}_{\leq\rho}$. Importantly, while ZIFFA performs very similar to ZIG for responses $\mathbf{r}_{\leq\rho}$, it slightly puts more mass on the responses $\mathbf{r}_{>\rho}$ resulting in a higher likelihood performance as illustrated in (Fig. 3).

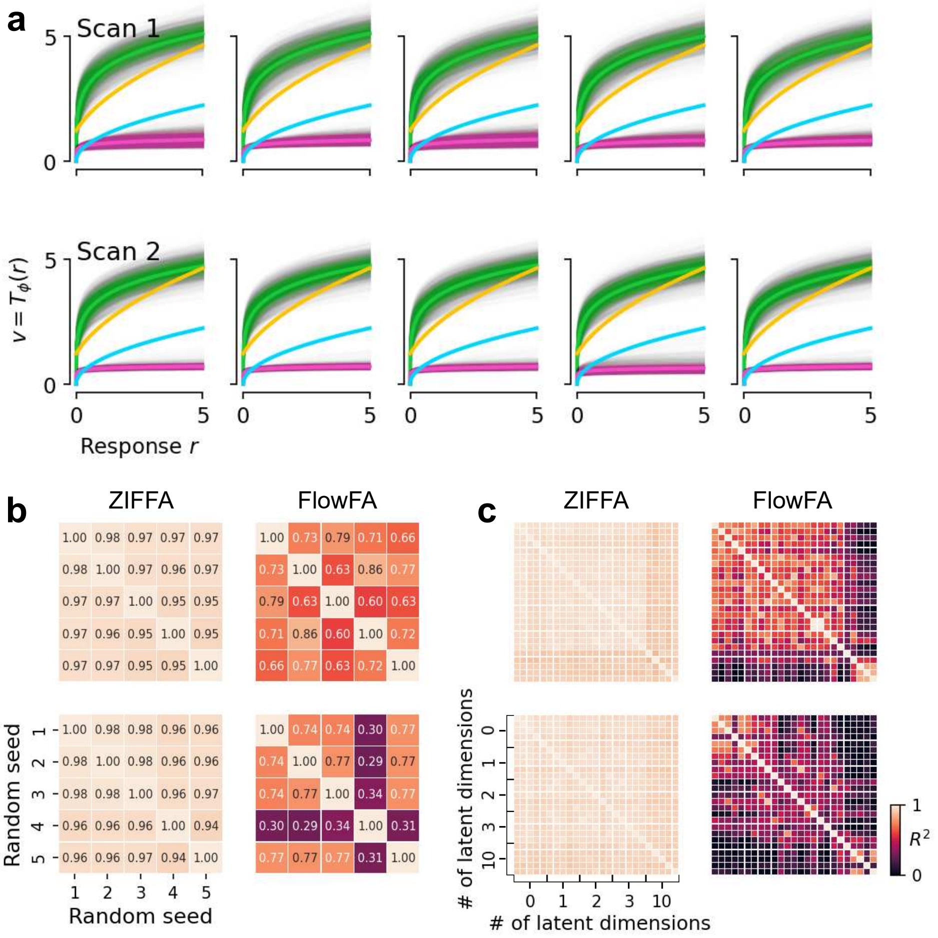


Figure S3: Consistency of the learned transformation across models initialized and trained with different random seeds, and across different number of latent dimensions. **a**: The learned flow transformation for both ZIFFA (green) and FlowFA (pink) models with 0-dimensional latent. Square-root (blue) and Anscombe (yellow) are also visualized for reference. Top row: Scan 1; bottom row: Scan 2. Colors are the same as in Fig. 3. **b**: Quantification of the consistency of learned flow transformations across random seeds, for the same models shown in **a**. To quantify the consistency, we flattened "transformed" responses **v** across all neurons getting a single vector for one seed, and then computed the R^2 between flattened **v** of all pairs of seeds. Higher R^2 value implies more consistency. Top row: Scan 1; bottom row: Scan 2; Left column: ZIFFA; right column: FlowFA. **c**: Same as **b**, but extended to also show the consistency of the learned transformation across models with different number of latent dimensions.

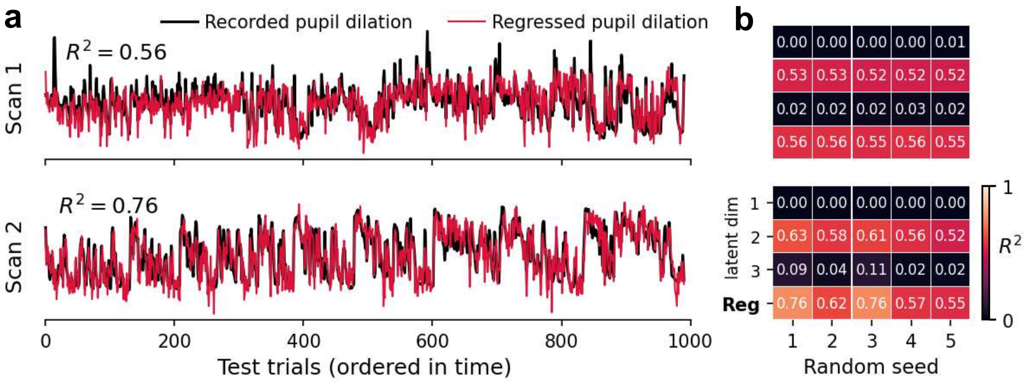


Figure S4: Correlation and regression analysis between inferred latent states and the pupil dilation. **a**: The regressed pupil dilation vs the recorded pupil dilation for the same model as in Fig. 4. **b**: First three rows: The R^2 values between orthonormalized latent states and pupil dilation across all random seeds. Last row: The R^2 values between regressed and recorded pupil dilation. Top: scan 1; bottom: scan 2.