Latent Variable and Implicit Models for Neural System Identification

Dissertation
der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von
Mohammad Bashiri
aus Tonekabon, Iran

Tübingen
2023

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	06.02.2024
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatter:	Prof. Dr. Fabian Sinz
2. Berichterstatter:	Prof. Dr. Martin Giese

Ich erkläre, dass ich die zur Promotion eingereichte Arbeit mit dem Titel “Latent Variable and Implicit Models for Neural System Identification” selbstständig verfasst, nur die angegebenen Quellen und Hilfsmittel benutzt und wörtlich oder inhaltlich übernommene Stellen als solche gekennzeichnet habe. Ich versichere an Eides statt, dass diese Angaben wahr sind und dass ich nichts verschwiegen habe. Mir ist bekannt, dass die falsche Abgabe einer Versicherung an Eides statt mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft wird.

Ort/Place, Datum/Date

Unterschrift/Signature

Acknowledgements

Throughout my doctoral journey, I've been privileged to be guided by remarkable mentors and to be part of a community that has not only advanced my scientific understanding but has also made this challenging endeavor a memorable experience. Although I cannot mention every individual who enriched my journey through graduate studies—my apologies for this—I remain indebted and grateful for each interaction during these transformative years.

First of all, I would like to thank my main PhD advisor Prof. Fabian Sinz. His mentorship has been the cornerstone of my academic progress, offering a rare blend of intellectual rigor and empathetic guidance. He has not only shaped this research but, more importantly, has also had a lasting impact on my approach to scientific inquiry. I also wish to thank Prof. Martin Giese, Dr. Anna Lavina, and Prof. Georg Martius who served as part of my Thesis Advisory Committee. Their thoughtful critiques and engaging discussions have not only enriched this work but also provided me with a broader perspective on the subject matter.

I extend a heartfelt thank you to my fellow colleagues in the Sinz Lab and Ecker Lab. Your contributions have made this journey both stimulating and enjoyable. Mara, your support, especially in navigating bureaucratic challenges, and your resilience in making it to our Africa trip with an injured ankle has made you a special colleague and friend. Michaela, our late-stage collaboration has been incredibly rewarding. I only wish we had started working together sooner. Konstantin, you've been by my side throughout this journey more closely than any other colleague, making significant contributions to both the research and my overall PhD experience. For that, I am deeply grateful. And Arne, sharing an office with you has been a delight—our discussions have been enlightening, and co-creating the ML crash course together with you is perhaps my favorite adventure during this time.

This journey has not been without difficult times, and I could not have gotten through it without the support of my friends. Annika, Amir, Akshay, Sina, Cveti, and Phrueksa, your unwavering encouragement and companionship made those challenges surmountable. I am profoundly grateful to each of you for being there when I needed it the most.

Finally, I must express my deepest gratitude to my parents, Shirin and Mehdi. Your faith in me, your love, and the sacrifices you've made have paved the path to where I stand today. Thank you for everything.

Abstract

One of the major goals of neural system identification is to understand the underlying neural mechanisms that give rise to visual perception and sensation. While the quest for understanding visual perception goes back many centuries, with the technological advancements in the past decades, machine learning methods have been increasingly used to analyze and model neural responses recorded from various visual sensory areas. In particular, deep neural networks (DNNs) have achieved state-of-the-art performance in predicting the activity of neurons in these regions. These networks have also been shown to learn representations of stimuli that closely match those found in the brain.

Besides their utility as hypotheses about the functional and structural properties of the brain, these powerful predictive models of sensory neurons, sometimes called *digital twins*, allow us to conduct experiments that are not feasible to conduct with their biological counterpart. Importantly, the findings of the experiments conducted with these digital twins have been verified in-vivo, providing further evidence that these models do indeed capture the complex functional properties of visual sensory neurons.

In this thesis, I will discuss three projects that leverage recent advancements in using DNNs to model the responses of visual sensory neurons. The first project focuses on a hybrid model that combines DNNs with latent variable models. This model aims to accurately predict the distribution of neural responses to unseen stimuli. It also infers latent state structures that have meaningful relations to behavioral variables, such as pupil dilation, as well as to the functional and anatomical properties of visual sensory neurons. The second project discusses a model that learns a reparameterization of the stimulus and, combined with DNN-based predictive models, learns a manifold in the stimulus space that visual sensory neurons are equally and maximally responsive to. The third and final project addresses an essential aspect of model development: finding better ways to quantify how good these models are in capturing neural responses.

Overall, this thesis focuses on three important aspects of neural system identification: (1) developing models that account for multiple driving factors of neural responses, (2) showcasing how these models can be used to generate insight into functional and structural properties of visual sensory neurons, and (3) developing metrics that assess the quality of such models.

Zusammenfassung

Eines der Hauptziele der neuronalen Systemidentifikation besteht darin, die zugrundeliegenden neuronalen Mechanismen zu verstehen, auf denen die visuelle Wahrnehmung und Empfindung basieren. Obwohl das Forschungsfeld, das sich mit den Mechanismen der visuellen Wahrnehmung beschäftigt, schon mehrere Jahrhunderte alt ist, hat vor allem der technologische Fortschritt der letzten wenigen Jahrzehnte im Bereich Machine Learning dazu beigetragen, neuronale Antworten aus verschiedenen visuellen Hirnarealen zu analysieren und zu modellieren. Insbesondere haben tiefe neuronale Netzwerke (DNNs) Spitzenleistungen bei der Vorhersage der Aktivität von Neuronen in visuellen sensorischen Bereichen erzielt. Es hat sich auch gezeigt, dass diese Netzwerke Repräsentationen von Stimuli erlernen, die denen im Gehirn sehr ähnlich sind.

Neben ihrem Nutzen als Hypothesen über die funktionalen und strukturellen Eigenschaften des Gehirns ermöglichen diese leistungsstarken Vorhersagemodelle für sensorische Neuronen, manchmal als *digitale Zwillinge* bezeichnet, Experimente auszuführen, die mit ihren biologischen Gegenstücken nicht machbar wären. Wichtig ist, dass die Ergebnisse der mit diesen digitalen Zwillingen durchgeführten Experimente in-vivo verifiziert wurden, was weitere Belege dafür liefert, dass diese Modelle tatsächlich die komplexen funktionalen Eigenschaften visueller sensorischer Neuronen erfassen.

In dieser Dissertation werde ich drei Projekte diskutieren, die aktuelle Fortschritte bei der Verwendung von DNNs zur Modellierung der Antworten visueller sensorischer Neuronen nutzen. Das erste Projekt konzentriert sich auf ein Hybridmodell, das DNNs mit latenten Variablenmodellen kombiniert. Dieses Modell zielt darauf ab, die Verteilung neuronaler Antworten auf ungesehene Reize genau vorherzusagen. Es erschließt auch latente Zustandsstrukturen, die sinnvolle Beziehungen zu Verhaltensvariablen wie Pupillenerweiterung sowie zu den funktionalen und anatomischen Eigenschaften visueller sensorischer Neuronen aufweisen. Das zweite Projekt behandelt ein Modell, das eine Reparametrisierung des Reizes erlernt und in Kombination mit DNN-basierten Vorhersagemodellen eine Mannigfaltigkeit im Stimulusraum erlernt, auf die visuelle sensorische Neuronen gleichmäßig und maximal reagieren. Das dritte und letzte Projekt befasst sich mit einem wesentlichen Aspekt der Modellentwicklung: der Suche nach besseren Möglichkeiten zur Quantifizierung der Qualität dieser Modelle bei der Erfassung neuronaler Antworten.

Insgesamt konzentriert sich diese Dissertation auf drei wichtige Aspekte der neuronalen Systemidentifikation: (1) die Entwicklung von Modellen, die für mehrere treibende Faktoren neuronaler Antworten verantwortlich sind, (2) die Darstellung, wie diese Modelle verwendet werden können, um Einblicke in funktionale und strukturelle Eigenschaften visueller sensorischer Neuronen zu gewinnen, und (3) die Entwicklung von Metriken zur Bewertung der Qualität solcher Modelle.

Contents

1 Introduction

1.1 Computational modeling of visual sensory neurons

Visual sensation and perception refer to the process of sensing, organizing, identifying, and interpreting visual information received through the eyes [1]. Importantly, the human visual system performs these tasks in a robust and energy-efficient manner [2, 3]. Such qualities spark curiosity and a desire to understand the visual system to build machines and algorithms (i.e., computer vision) that mimic the useful properties of their biological counterpart. Additionally, understanding how visual perception is implemented and performed by the brain can have significant implications for reversing visual disorders.

The quest for understanding visual perception goes back many centuries to Newton who laid the foundations of modern work on color vision in 1704, Franz Boll who discovered rhodopsin (which he called "visual purple" at the time) in frog retina in 1876, Hartline who described receptive fields, as well as ON, OFF, and ON-OFF responses in 1938, and many others [4]. However, while by the mid-20th century important discoveries were made about the retina, it was still unclear how the cortex processes visual stimuli at the level of single neurons. It was in 1959 that Hubel and Wiesel changed the course of visual neuroscience research with their experiment on cats' visual cortex [5]. Along with a follow-up paper in 1962 [6], they showed the presence of orientation and direction tuning in the primary visual cortex, described simple and complex cells, characterized orientation columns, and proposed a model for orientation selectivity (Fig. 1.1).

Up until this point, many facts were known about the anatomical, neurophysiological, and psychophysical aspects of sensation and perception, and scientists were starting to wonder how these various components work together to give rise to perception. In other words, as Barlow puts it, they needed "ideas about what operations are performed by the various structures we have examined" [7]. The initial question that Barlow attempted to tackle in 1961 was: How does the brain, and specifically the retina, deal with the enormous amount of incoming sensory signals to extract behaviorally relevant information? Borrowing ideas from information theory, Barlow argued that sensory neurons *recode* the incoming signal to find the most relevant (i.e., least redundant) internal representation of the outer world (what he referred to as the "redundancy-reducing hypothesis") [7]. His influential work predicted that the visual system should have filters that are evolved to extract informative statistics from the natural world.

Building upon these ideas, Olshausen and Field [8] developed a learning algorithm that, when trained on natural images, generated filters resembling the orientation-selective

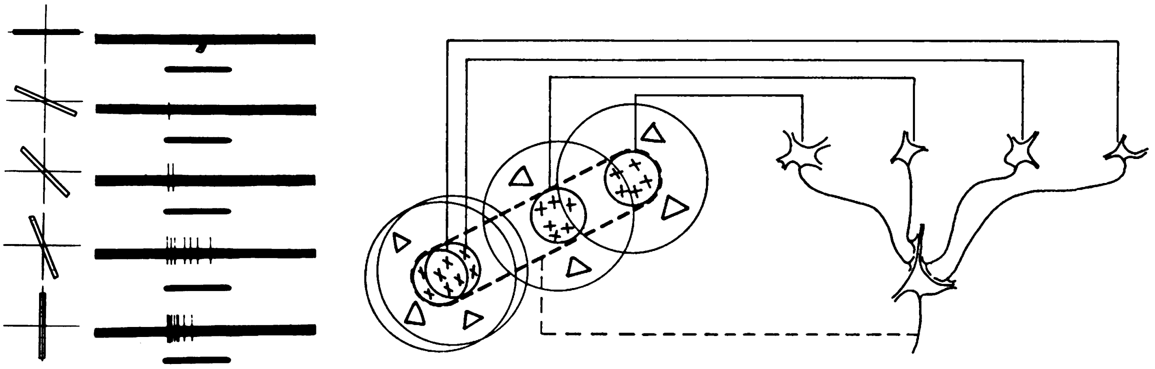


Figure 1.1: **Left**: Response of an example neuron from a cat's visual cortex to different stimulus orientations [5]. **Right**: Model suggested by Hubel and Wiesel [6] for explaining the organization of simple receptive fields. Both illustrations are adapted with permission from Hubel and Wiesel's original papers in the Journal of Physiology [5, 6].

receptive fields observed in the primary visual cortex (V1) by Hubel and Wiesel [5]. Their findings highlighted that sparseness was an important characteristic of the filters to decorrelate features of the natural images, resulting in a more efficient (i.e., less redundant) signal for the higher visual areas. Subsequent technological advancements enabled the simultaneous recording of neural populations across multiple visual areas, along with experimental and behavioral variables, resulting in an abundance of research and discoveries pertaining to various aspects of visual sensory areas. Notable directions of investigation include characterizing the filters employed by visual sensory neurons to extract relevant stimulus features, determining whether their operations are linear or more intricate and nonlinear, quantifying the influence of behavioral variables and cognitive processes (e.g., attention) on the coding properties of sensory neurons, and uncovering population-level computational principles [9–15].

As the data became larger in number of samples, features, and recorded neurons, machine learning (ML) methods have been increasingly used to analyze and model the responses of sensory neurons. One specific approach that has taken the field by storm in the past decade is the use of deep neural networks (DNN). As models of the brain, they have achieved state-of-the-art performance in predicting neural responses [16–22], and have been shown to learn representations of the stimulus that closely match those in the brain [11, 23].

Besides providing hypotheses about the functional and structural properties of the brain, these powerful predictive models allow us to conduct experiments that are not feasible to conduct with their biological counterpart. For instance, let us consider that we want to find out what patterns in a 100×100 image maximize the activity of a specific neuron. If we try all combinations of pixel values, even in the case where they can only take on binary values (i.e., 0 or 1), we will have to show 2^{10000} images to the subject, which is simply impossible. However, by using a predictive model of visual sensory neurons, several studies have recently shown that these models can be *reversed* to find maximally exciting stimuli (MEI) for a target neuron [9, 10, 24]. Importantly, when these MEIs were shown back to the subject, the corresponding neurons in the visual cortex elicited similarly high responses. These findings provide further evidence that these models, often referred to as *digital twins*, do indeed capture the complex functional properties of visual sensory neurons.

Building upon recent advancements in neural system identification, this thesis aims to address several open questions and technical challenges in the field. While deep neural networks (DNNs) have shown promise in predicting neural responses and mimicking the functional properties of visual sensory neurons, there remains a gap in our understanding of how internal processes and behavioral variables modulate these responses. Additionally, with the increasing complexity of these models (e.g. using more complex distributions to capture neural responses), there is a need for methods to quantify the performance of these predictive models on multiple aspects.

To address these challenges, this thesis discusses three projects conducted during my PhD research, each designed to fill a specific gap in current knowledge. The first project introduces a model that combines DNNs with latent variable models. This model not only accurately predicts the distribution of neural responses to unseen stimuli but also infers latent state structures that have meaningful relationships with behavioral variables, such as pupil dilation. This work addresses the technical gap in modeling the activity of sensory neurons by simultaneously taking into account both sensory input and internal processes, thereby providing a foundational step in understanding how internal states and behavioral variables affect neural responses.

The second project presents a model that learns to reparameterize the stimulus. When combined with DNN-based predictive models, this approach identifies a manifold in the stimulus space to which visual sensory neurons are equally and maximally responsive. This work contributes to ongoing efforts to characterize the encoding properties of visual sensory neurons by elucidating the complexity of their invariances. The third project focuses on applying improved metrics for evaluating the performance of predictive models in capturing neural responses, addressing a crucial aspect that has often been overlooked in the field.

By tackling these specific challenges, this thesis aims to advance our understanding of the functional and structural properties of visual sensory neurons and to provide new tools and metrics for future research in the field.

1.2 List of publications and contributions

While these projects are described in detail in their corresponding chapter, a list of their corresponding publications can be found below where I layout the details of my and other authors' contributions. Additionally, I contributed to multiple other projects which are also listed below.

1.2.1 Publications included in this thesis

The list below contains the publications corresponding to the main projects that constitute parts of this thesis, accompanied by a detailed description of the contributions that were made to each publication. The description of the contributions is based on the Contributor Roles Taxonomy (CRediT) guidelines [25–27] with minor adaptations. The adaptations were made such that the categories better capture different aspects of a method paper, which is mainly the case for these papers. The symbol * denotes equal contribution.

- **Mohammad Bashiri***, Edgar Walker*, Konstantin-Klemens Lurz, Akshay Jagadish, Taliah Muhammad, Zhiwei Ding, Zhuokun Ding, Andreas Tolias, and Fabian Sinz. A flow-based latent state generative model of neural population responses to natural images. *Advances in Neural Information Processing Systems*, 34:15801–15815, 2021.
  Initial idea: MB and AJ
  Methodology and theoretical development: MB, EW, FS, AJ
  Software: mainly MB with the help of AJ and KKL
  Method validation and experiments: mainly MB with the help of AJ
  Data collection and preprocessing: TM, ZhiD, ZhuD
  Figures and visualization: MB
  Writing (original draft): MB, EW, FS
  Writing (review and editing): MB, EW, FS, KKL
  Funding acquisition: FS and AT
  Remarks on the shared authorship: While MB implemented almost the entire software and conducted most of the experiments, EW had significant contributions to the ideas, provided valuable feedback, and closely supervised the project. Therefore, an equal first authorship was deemed fair.

- Luca Baroni*, **Mohammad Bashiri***, Konstantin F Willeke, Ján Antolík, and Fabian H Sinz. Learning invariance manifolds of visual sensory neurons. In *NeurIPS Workshop on Symmetry and Geometry in Neural Representations*, pages 301–326. PMLR, 2023.
  Initial idea: FS, JA, LB, MB
  Methodology and theoretical development: LB, MB, FS
  Software: LB and MB
  Method validation and experiments: mainly LB with the help of MB
  Data collection and preprocessing: publicly available data was used
  Figures and visualization: MB and LB
  Writing (original draft): LB and MB
  Writing (review and editing): all authors
  Funding acquisition: JA and FS
  Remarks on the shared authorship: Other than the method validation and experiments, which were mainly conducted by LB, MB and LB had an equal contribution to the project.

- Konstantin-Klemens Lurz*, **Mohammad Bashiri***, Edgar Y. Walker, and Fabian H Sinz. Bayesian oracle for bounding information gain in neural encoding models. In *International Conference on Learning Representations (ICLR)*, 2023.
  Initial idea: FS, EW, MB
  Methodology and theoretical development: FS, KKL, MB, EW
  Software: KKL and MB
  Method validation and experiments: mainly KKL with the help of MB
  Data collection and preprocessing: publicly available data was used
  Figures and visualization: MB and KKL
  Writing (original draft): KKL, MB, FS
  Writing (review and editing): all authors
  Funding acquisition: FS
  Remarks on the shared authorship: Other than the method validation and experiments, which were mainly conducted by KKL, MB and KKL had an equal contribution to the project.

1.2.2 Other publications

Below is a list of publications that are a result of the projects, conducted during the PhD, to which I contributed. For these publications, I will only explain my specific contribution, instead of providing a comprehensive list of all authors' contributions.

- 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. In *International Conference on Learning Representations (ICLR)*, 2021.
  Contributions: MB contributed to discussions about the methodology, figures and visualization, as well as writing (review and editing).

- Konstantin F Willeke*, Paul G Fahey*, **Mohammad Bashiri**, Laura Pede, Max F Burg, Christoph Blessing, Santiago A Cadena, Zhiwei Ding, Konstantin-Klemens Lurz, Kayla Ponder, et al. The sensorium competition on predicting large-scale mouse primary visual cortex activity. *arXiv preprint arXiv:2206.08666*, 2022.
  Contributions: This is a white paper about a competition that was conducted as part of NeurIPS 2022. MB was one of the main organizers of the competition and contributed to the development of the starting kit, competition metrics, website infrastructure, and the evaluation of submissions.

- Paweł A Pierzchlewicz, R James Cotton, **Mohammad Bashiri**, and Fabian H Sinz. Multi-hypothesis 3d human pose estimation metrics favor miscalibrated distributions. *arXiv preprint arXiv:2210.11179*, 2022.
  Contributions: MB contributed to discussions about the methodology, figures and visualization, as well as writing (review and editing).

- Paweł A Pierzchlewicz, **Mohammad Bashiri**, R James Cotton, and Fabian H Sinz. Optimizing mpjpe promotes miscalibration in multi-hypothesis human pose lifting. In *International Conference on Learning Representations (ICLR) as a Tiny Paper*, 2023.
  Contributions: MB contributed to discussions about the methodology, figures and visualization, as well as writing (review and editing).

- Jiakun Fu, Pawel A Pierzchlewicz, Konstantin F Willeke, **Mohammad Bashiri**, Taliah Muhammad, George H Denfield, Fabian Hubert Sinz, and Andreas S Tolias. Heterogeneous orientation tuning across sub-regions of receptive fields of v1 neurons in mice. *Under Review*.
  Contributions: MB contributed to the methodology by providing a method that finds optimal Gabor filters via gradient-based optimization methods [34].

- Polina Turishcheva, Paul G Fahey, Laura Hansel, Rachel Froebe, Kayla Ponder, Michaela Vystrčilová, Konstantin F Willeke, **Mohammad Bashiri**, Eric Wang, Zhiwei Ding, Andreas S. Tolias, Fabian Sinz, and Alexander S. Ecker. The dynamic sensorium competition for predicting large-scale mouse visual cortex activity from videos. *arXiv preprint arXiv:2305.19654*, 2023.
  Contributions: This is a white paper about a competition that was conducted as part of NeurIPS 2023. MB contributed to the preprocessing and packaging of the dataset for the competition.

2 Background

This chapter serves as a reference guide to aid in understanding subsequent chapters. While not essential to read upfront, it may be useful to refer back to relevant sections as you progress through the thesis. Each section specifies its relevance to other chapters.

2.1 Deep neural networks

This section offers context for DNN-based models of visual cortical neurons, relevant to chapters 3, 4, and 5.

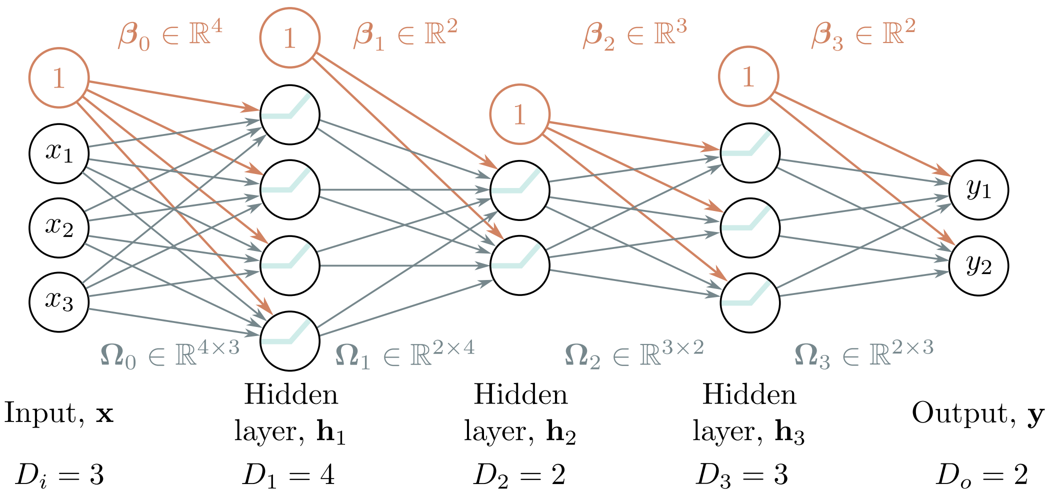
Deep neural networks (DNNs) are parametric machine learning models that consist of multiple linear-nonlinear transformations before yielding an output [36, 37]. Each transformation, commonly referred to as a *hidden layer*, is a linear transformation of its input followed by a simple nonlinearity such as ReLU or sigmoid [38]. Each output dimension of such linear-nonlinear transformation is called a *hidden unit.* Neural networks that contain a higher number of hidden units are said to be *wider* and those that contain more hidden layers are referred to as *deeper*. On the lower extreme where there is only a single hidden layer, the network is called a *shallow* network.

Feedforward neural networks While there is much flexibility in the specific architecture of DNNs (i.e. how hidden units are connected), the most common type of architecture is the forward deep network, in which the computations are applied sequentially (Fig. 2.1):

$$\mathbf{y} = f(\mathbf{x}) = (f^L \circ f^{L-1} \circ \cdots \circ f^2 \circ f^1),$$

where $f^\ell(\mathbf{z}) = \sigma(\mathbf{W}^\ell \mathbf{z} + \mathbf{b}^\ell)$ is a linear-nonlinear transformation, performed by the ℓth hidden layer, with $\mathbf{W}$ being the weight matrix and $\mathbf{b}$ the bias term.

Convolutional neural networks Convolutional neural networks are a sub-class of DNNs that consist of a series of convolutional layers [39, 40]. Convolutional layers are linear-nonlinear transformations based on the convolution operation, where each output dimension is a weighted sum of the nearby dimensions in the input. In the case of a 2D input (e.g. an image) the same weight matrix, so-called *kernel* or *filter*, slides through the 2D input and is applied to all spatial locations, resulting in a single *channel* in the output. Most commonly, each convolutional layer consists of multiple kernels resulting in a 2D output with multiple channels corresponding to the number of kernels. Importantly, applying the same kernel at every spatial location, referred to as *weight sharing*, results in a reduction in the number of model parameters, facilitating model training and generally resource allocation. 2D convolutional models are widely applied to image data where nearby pixels are statistically related. This allows the model to use local computation resulting in a model that performs well while at the same time having fewer parameters.



$\boldsymbol{\beta}_0 \in \mathbb{R}^4$
$\boldsymbol{\beta}_1 \in \mathbb{R}^2$
$\boldsymbol{\beta}_2 \in \mathbb{R}^3$
$\boldsymbol{\beta}_3 \in \mathbb{R}^2$
1
x_1
x_2
x_3
y_1
y_2
$\boldsymbol{\Omega}_0 \in \mathbb{R}^{4\times3}$
$\boldsymbol{\Omega}_1 \in \mathbb{R}^{2\times4}$
$\boldsymbol{\Omega}_2 \in \mathbb{R}^{3\times2}$
$\boldsymbol{\Omega}_3 \in \mathbb{R}^{2\times3}$
Input, $\mathbf{x}$
Hidden layer, $\mathbf{h}_1$
Hidden layer, $\mathbf{h}_2$
Hidden layer, $\mathbf{h}_3$
Output, $\mathbf{y}$
$D_i = 3$
$D_1 = 4$
$D_2 = 2$
$D_3 = 3$
$D_o = 2$

Figure 2.1: Matrix notation for a neural network with a 3-dimensional input, a 2-dimensional output, and 3 hidden layers. The figure is reprinted from [41].

Objective function and model training Model training refers to the process of optimizing the parameters of the model to achieve the best possible performance on a specific task. A specific task (e.g. classification) enforces a certain choice of the objective function (e.g. cross-entropy loss) which is most commonly formulated such that it is minimized during training. The objective function $\mathcal{L}$, also commonly referred to as *loss function*, typically yields a single scalar value which is computed by taking the average over all training samples $\mathbf{x}$ and the dimensions of the output $\mathbf{y}$:

$$\theta^* = \arg\min_{\theta} \frac{1}{ND} \sum_{i=1}^{N} \sum_{d=1}^{D} \mathcal{L}(f_\theta(\mathbf{x}_i), y_i^d),$$

where θ represents the parameters of the DNN f, θ^* is the optimal set of parameters resulting in the minimum loss value, N is the number of samples, and D is the number of output dimensions. Such cases where the objective function contains model outputs $f_\theta(\mathbf{x})$ as well as the target values $\mathbf{y}$ are referred to as *supervised* tasks. In contrast, *unsupervised* tasks use an objective function that does not include any target values and is only applied on (a representation of) the input samples with the goal of finding meaningful and abstract structures in the data.

2.1.1 Models of visual cortical neurons

The primary goal of neural system identification is to construct a model that describes how a neuron responds to arbitrary stimuli, viewing sensory neurons as computational units that implement a certain function $f(x)$ on the sensory input x. Importantly, such a model can then be used to conduct experiments *in silico* that are not feasible *in vivo*, serving as a *digital twin* of its biological counterpart.

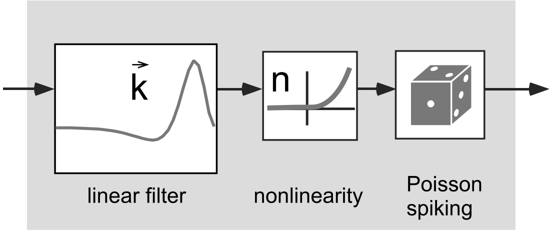


Figure 2.2: A generic schematic of a GLM. The figure is reprinted from [42].

Single-neuron models The traditional approach to modeling the stimulus-response relationship of sensory neurons involves building separate models for each neuron.

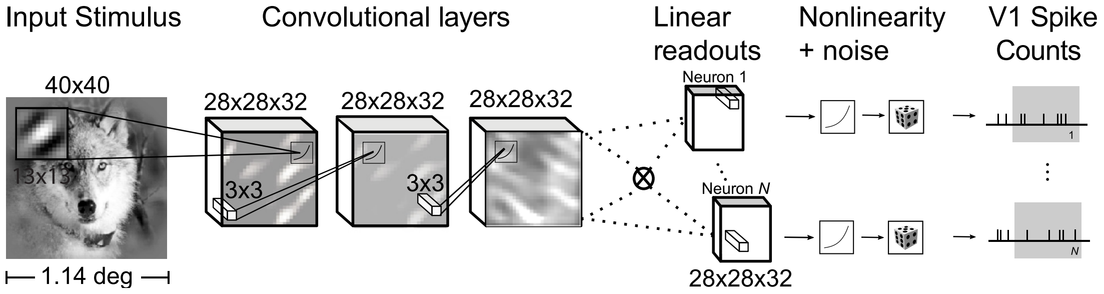


Figure 2.3: An example of a DNN-based model of visual sensory neurons, which consists of multiple convolutional layers, outputting the shared nonlinear representation of the input, followed by a neuron-specific spatially-localized linear readout. The figure is reprinted from [23].

Generalized Linear Models (GLMs) and related methods have been commonly used for this task [42–44]. These models typically consist of a linear filter followed by a nonlinearity, along with a noise model (e.g. Poisson noise) to account for the stochasticity in neural responses (Fig. 2.2). GLMs can be formulated as convex optimization problems, allowing for global optimum solutions, and can be extended to incorporate spatiotemporal relationships and coupling between neurons [45, 46]. However, the linear mapping followed by a nonlinearity imposes a strong assumption on the discoverable space of functions, limiting the capacity of GLMs to identify complex nonlinear computations. An alternative approach is to feed a nonlinear representation of the original input to the GLM. One challenging aspect of this alternative is selecting the appropriate feature space since it requires a deeper understanding of the cell's nonlinear behavior. However, obtaining the appropriate feature space can be considered as part of the model and can be learned using advanced machine learning methods (e.g. DNNs). While this runs into the potential disadvantage of losing the interpretability offered by GLMs, it provides more powerful predictive models that can capture novel and non-trivial nonlinear behaviors of visual sensory neurons.

DNN-based models DNNs are the state-of-the-art models for learning feature spaces that are relevant for the data and the task. In the last decade, these models have been increasingly applied to neural responses recorded across multiple species, modalities, and brain areas, and resulted in considerable gain in predictive power compared to previous models [11, 15–21, 23]. DNN-based models, in their general form, consist of a DNN (so-called *core*) that learns a nonlinear representation of the input and a GLM (commonly referred to as *readout*) that uses the output of DNN as input to yield predictions of neural responses. In the context of visual sensory neurons, since we are dealing with a 2D input (i.e. images or videos), CNNs have been the dominant choice as the architecture used for the core (Fig. 2.3). Most of these models are applied to a population of neurons as opposed to a single neuron, where the nonlinear representation (i.e. output of the CNN) is shared among all neurons, followed by a neuron-specific readout. The motivation behind this design choice is that many neurons in the visual cortex are known to perform similar computations but at different spatial locations [8].

Such a model that allows simultaneous modeling of many neurons combined with the segregation of the shared nonlinear computations (via the *core*) and the neuron-specific computations (via the *readout*) offers the flexibility for innovative design choices on several ends. First, modeling many neurons simultaneously allows for capturing dependencies between neurons, for instance by adding a latent variable model to the DNN-based models [22]. Second, the core and the readout can take advantage of known properties of the sensory neurons, such as orientation selectivity [18] and localized receptive fields [17, 21] to not only reduce the number of model parameters but also yield better-performing models. Additionally, as mentioned earlier, power predictive models of sensory neurons allow us to conduct experiments *in silico* that are not feasible *in vivo*. For instance, they can be used to find an input that maximizes the activity of a single neuron or a whole population of neurons [9, 10], or to find the types of invariances that visual sensory neurons exhibit [28, 47, 48]. Many of these findings have been verified in vivo, which provides additional evidence, beyond prediction performance, that these models do indeed capture the complex nonlinear computations performed by visual sensory neurons.

2.2 Latent variable models

The concepts introduced in this section are directly applicable to the methodologies used in Chapter 3.

In many cases, different components of the observed data are not independent of one another. For instance, sensory neurons have been shown to be affected by low-dimensional internal processes such as attention [49–51], which gives rise to statistical dependencies between neurons. By investigating the statistical relationship between different components of the data, latent variable models (LVM) seek to identify the underlying factors and the generative process that gives rise to the observed data. While many LVMs have been applied to neural responses [52–64], here we will focus on two specific models, namely, Factor Analysis (FA) and Normalizing Flows (NF).

2.2.1 Factor analysis

Factor analysis (FA) is a specific form of latent variable models that assumes the observed d-dimensional data $\mathbf{y} \in \mathbb{R}^d$ is generated via the following generative process:

$$\mathbf{y} = \mathbf{b} + \mathbf{C}\mathbf{z} + \varepsilon, \tag{2.1}$$

where $\mathbf{z} \in \mathbb{R}^k$ is a low-dimensional latent state with $k \ll d$ and an isotropic Gaussian prior $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_k)$ whose samples map to $\mathbf{y}$ via the *factor loading matrix* $\mathbf{C} \in \mathbb{R}^{d \times k}$, $\mathbf{b}$ is the mean of the FA model, and $\varepsilon \sim \mathcal{N}(\mathbf{0}, \Psi)$ is an independent noise term with a diagonal covariance matrix $\Psi \in \mathbb{R}^{d \times d}$. Below, I will derive the joint distribution $p(\mathbf{z}, \mathbf{y})$ which we will use later to demonstrate some use-cases of the FA model.

Since the FA model assumes a linear mapping of a Gaussian-distributed latent variable $\mathbf{z}$ onto the observed variable $\mathbf{y}$, the joint distribution $p(\mathbf{z}, \mathbf{y})$ is a Gaussian distribution too:

$$p(\mathbf{z}, \mathbf{y}) = \mathcal{N}(\begin{bmatrix} \mathbf{z} \\ \mathbf{y} \end{bmatrix}; \begin{bmatrix} \mu_{\mathbf{z}} \\ \mu_{\mathbf{y}} \end{bmatrix}, \begin{bmatrix} \mathbf{\Sigma}_{\mathbf{z}} & \mathbf{\Sigma}_{\mathbf{zy}} \\ \mathbf{\Sigma}_{\mathbf{yz}} & \mathbf{\Sigma}_{\mathbf{y}} \end{bmatrix}),$$

where the mean $[\mu_{\mathbf{z}}, \mu_{\mathbf{y}}]^\top$ can be computed as follows:

$$\begin{aligned} &\mu_{\mathbf{z}} = \mathbf{0} \quad \text{(from the prior } p(z) = \mathcal{N}(\mathbf{0}, \mathbf{I}_k)\text{)}, \\ &\mu_{\mathbf{y}} = \mathbb{E}[\mathbf{y}] = \mathbb{E}[\mathbf{b} + \mathbf{C}\mathbf{z} + \varepsilon] = \mathbf{b} + \mathbf{C}\mathbb{E}[\mathbf{z}] + \mathbb{E}[\varepsilon] = \mathbf{b}, \end{aligned}$$

and the terms in the covariance matrix can be computed as follows:

$$\begin{aligned}
\mathbf{\Sigma_z} &= \mathbf{I}_k \quad \text{(from the prior } p(z) = \mathcal{N}(\mathbf{0}, \mathbf{I}_k)\text{)},\\
\mathbf{\Sigma_{zy}} &= \mathbf{\Sigma_{yz}}^\top = \mathbb{E}[(\mathbf{z} - \mu_\mathbf{z})(\mathbf{y} - \mu_\mathbf{y})^\top] = \mathbb{E}[\mathbf{z}(\mathbf{Cz} + \varepsilon)^\top] = \mathbf{I}_k\mathbf{C}^\top + \mathbf{0} = \mathbf{C}^\top,\\
\mathbf{\Sigma_y} &= \mathbb{E}[(\mathbf{y} - \mu_\mathbf{y})(\mathbf{y} - \mu_\mathbf{y})^\top] = \mathbb{E}[(\mathbf{Cz} + \varepsilon)(\mathbf{Cz} + \varepsilon)^\top] = \mathbf{CC}^\top + \Psi,
\end{aligned}$$

resulting in the following joint distribution:

$$p(\mathbf{z}, \mathbf{y}) = \mathcal{N}(\begin{bmatrix}\mathbf{z}\\ \mathbf{y}\end{bmatrix}; \begin{bmatrix}\mathbf{0}\\ \mathbf{b}\end{bmatrix}, \begin{bmatrix}\mathbf{I}_k & \mathbf{C}^\top\\ \mathbf{C} & \mathbf{CC}^\top + \Psi\end{bmatrix}).$$

Note that the marginal distribution $p(\mathbf{y}) = \mathcal{N}(\mathbf{y}; \mathbf{b}, \mathbf{CC}^\top + \Psi)$ over $\mathbf{y}$ does not involve the latent variable $\mathbf{z}$ and only contains the parameters $\mathbf{b}$, $\mathbf{C}$, and the diagonal entries of the covariance matrix Ψ, which can be learned via common gradient-based optimization methods to maximize the marginal likelihood $p(\mathbf{y}|\mathbf{b}, \mathbf{C}, \text{diag}(\Psi))$.

Inferring latent states from observed data Once the FA model is optimized, using the joint distribution and the conditional properties of the Gaussian distribution, we can infer the latent variables that correspond to the observed data $\mathbf{y}$. This involves computing the conditional distribution $p(\mathbf{z}|\mathbf{y})$ parameterized by the conditional mean $\mu_{\mathbf{z}|\mathbf{y}}$ and the conditional covariance $\mathbf{\Sigma}_{\mathbf{z}|\mathbf{y}}$:

$$p(\mathbf{z}|\mathbf{y}) = \mathcal{N}(\mathbf{z}; \mu_{\mathbf{z}|\mathbf{y}}, \mathbf{\Sigma}_{\mathbf{z}|\mathbf{y}}),$$

where the distribution parameters can be computed using the conditional properties of the Gaussian distribution:

$$\begin{aligned}
\mu_{\mathbf{z}|\mathbf{y}} = \mathbb{E}(\mathbf{z}|\mathbf{y}) &= \mu_\mathbf{z} + \mathbf{\Sigma_{zy}}\mathbf{\Sigma_y}^{-1}(\mathbf{y} - \mu_\mathbf{y})\\
&= \mathbf{0} + \mathbf{\Sigma_{zy}}\mathbf{\Sigma_y}^{-1}(\mathbf{y} - \mu_\mathbf{y})\\
&= \mathbf{C}^\top(\mathbf{CC}^\top + \Psi)^{-1}(\mathbf{y} - \mathbf{b}), \qquad (2.2)\\
\mathbf{\Sigma}_{\mathbf{z}|\mathbf{y}} &= \mathbf{\Sigma_z} - \mathbf{\Sigma_{zy}}\mathbf{\Sigma_y}^{-1}\mathbf{\Sigma_{zy}}^\top\\
&= \mathbf{I}_k - \mathbf{C}^\top(\mathbf{CC}^\top + \Psi)^{-1}\mathbf{C}. \qquad (2.3)
\end{aligned}$$

Here, $\mu_{\mathbf{z}|\mathbf{y}}$ represents the expected value of the latent variables knowing about the observed samples of $\mathbf{y}$, while $\mathbf{\Sigma}_{\mathbf{z}|\mathbf{y}}$ reflects the uncertainty over this expectation. Note that computing the conditional distribution involves inverting a $d \times d$ matrix with rank d, which depending on the dimensionality of the data can be computationally expensive. However, due to the special structure of the covariance matrix, which contains a rank-k matrix $\mathbf{CC}^\top$ and a diagonal matrix Ψ, this inversion can be computed cheaply using matrix inversion lemmas. Using the Woodbury matrix identity, we can re-write the inversion $(\mathbf{CC}^\top + \Psi)^{-1}$ in Eqs. 2.2 and 2.3 as follows:

$$(\mathbf{CC}^\top + \Psi)^{-1} = \Psi^{-1} - \Psi^{-1}\mathbf{C}(\mathbf{I}_k + \mathbf{C}^\top\Psi^{-1}\mathbf{C})^{-1}\mathbf{C}^\top\Psi^{-1}.$$

With this reformulation, the inversion is applied on a $k \times k$ instead of a $d \times d$ matrix, which in most cases is computationally much cheaper. Furthermore, since Ψ is a diagonal matrix, its inverse can be easily computed by replacing the diagonal elements of the matrix with their reciprocals.

Inferring interpretable latent states Despite a well-defined relationship $\mathbf{y} = \mathbf{b} + \mathbf{Cz} + \varepsilon$ between observed samples $\mathbf{y}$ and the latent variable $\mathbf{z}$, interpreting the inferred latent states $\mathbb{E}(\mathbf{z}|\mathbf{y})$ is difficult and quite arbitrary. The reason is that the factor loading matrix $\mathbf{C}$ can only be uniquely determined up to an arbitrary orthogonal transformation. That is, we can transform $\mathbf{C}$ and $\mathbf{z}$ by any arbitrary orthogonal transform matrix $\mathbf{R}$ to yield $\mathbf{C}' = \mathbf{CR}$ and $\mathbf{z}' = \mathbf{R}^\top \mathbf{z}$, and the original relationship is preserved since $\mathbf{C}'\mathbf{z}' = \mathbf{CRR}^\top \mathbf{z} = \mathbf{Cz}$. Furthermore, since a permutation matrix is an orthogonal transformation, the inferred latent states are not necessarily ordered by how much variability in the data they account for.

To address this issue, a similar approach to Yu et al. [65] can be used. Briefly, we orthonormalize the columns of $\mathbf{C}$ by applying singular value decomposition to the learned $\mathbf{C}$ which yields $\mathbf{C} = \mathbf{UDV}^\top$. As a result, $\mathbf{Cz}$ can be re-written as $\mathbf{Cz} = \mathbf{U}(\mathbf{DV}^\top \mathbf{z}) = \mathbf{U}\tilde{\mathbf{z}}$ where $\tilde{\mathbf{z}} \equiv \mathbf{DV}^\top \mathbf{z}$ is the *orthonormalized latent state*. Consequently, instead of inferring the MAP of $\mathbf{z}$, $\mathbb{E}[\mathbf{z}|\mathbf{y}]$, we would infer $\mathbf{DV}^\top \mathbb{E}[\mathbf{z}|\mathbf{y}]$. 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 the data, 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 any arbitrary orthogonal transformation, since $\mathbf{Cz} = \mathbf{UDV}^\top \mathbf{z} = \mathbf{UDV}^\top \mathbf{RR}^\top \mathbf{z} = \mathbf{CRR}^\top \mathbf{z} = \mathbf{C}'\mathbf{z}' = \mathbf{U}\tilde{\mathbf{z}}$, where the resulting orthonormalized latent state $\tilde{\mathbf{z}} \equiv \mathbf{DV}^\top \mathbf{z}$ stays the same regardless of using transformed $\mathbf{C}' = \mathbf{CR}$ and $\mathbf{z}' = \mathbf{R}^\top \mathbf{z}$.

Improved prediction by conditioning on other dimensions Another use-case of the FA model is to leverage the learned statistical dependencies in the data to make better predictions about certain dimensions given the other dimensions. Similar to the latent states, we can use the conditional distribution $p(y_i|\mathbf{y}_{\setminus i}) = \mathcal{N}(y_i; \mu_{i|\setminus i}, \mathbf{\Sigma}_{i|\setminus i})$ to compute the MAP estimate $\mathbb{E}(y_i|\mathbf{y}_{\setminus i})$ for a target dimension i given all the other dimensions $\setminus i$:

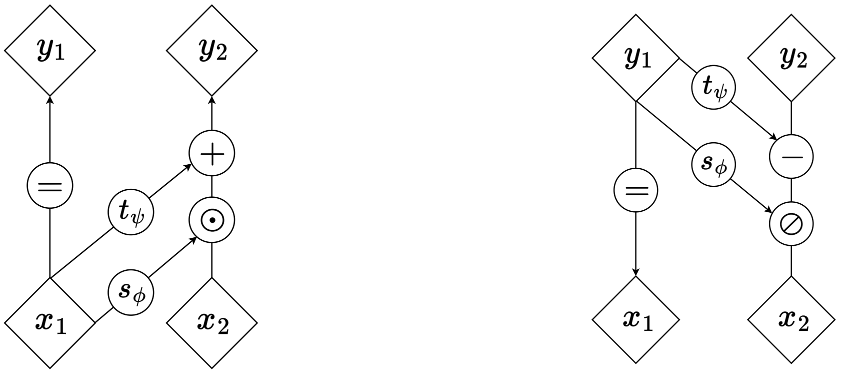
$$\begin{aligned} \mu_{i|\setminus i} &= \mathbb{E}(y_i|\mathbf{y}_{\setminus i}) = b_i + \mathbf{\Sigma}_{i,\setminus i}\mathbf{\Sigma}^{-1}_{\setminus i,\setminus i}(\mathbf{y}_{\setminus i} - \mu_{\setminus i}) \\ \mathbf{\Sigma}_{i|\setminus i} &= \mathbf{\Sigma}_{i,i} - \mathbf{\Sigma}_{i,\setminus i}\mathbf{\Sigma}^{-1}_{\setminus i,\setminus i}\mathbf{\Sigma}^\top_{i,\setminus i}, \end{aligned}$$

where $\mathbf{\Sigma} = \mathbf{CC}^\top + \Psi$, and the subscript i and $\setminus i$ correspond to the entries of the corresponding variable for the target dimension and other dimensions, respectively.

2.2.2 Normalizing flows

One of the major goals of probabilistic machine learning is to model unknown probability distributions given the samples drawn from that distribution. Learning the distribution not only allows us to evaluate likelihoods and detect outliers but also generate new samples for simulations or create larger datasets for training models. This generally falls under the category of unsupervised methods and is also sometimes called *generative modeling*.

Normalizing Flows (NF) [66–68] are a family of generative models with a tractable distribution where both density evaluation is exact and new samples can be efficiently generated. This is an important advantage compared to other generative models such as generative adversarial networks (GANs) [69] and variational auto-encoders (VAEs) [70], where exact density evaluation of new samples is not possible.



(a) Forward propagation

(b) Inverse propagation

Figure 2.4: Schematic of the forward and inverse computations performed by the coupling layer. The figure is adapted from [71].

NFs learn an unknown probability density $p_x(\mathbf{x})$ given samples of a random variable $\mathbf{x} \in \mathbb{R}^D$, leveraging the change of variables formula:

$$p_x(\mathbf{x}) = p_z(f_\theta(\mathbf{x})) \cdot |\det \nabla_x f_\theta(\mathbf{x})|, \tag{2.4}$$

where $p_z(\cdot)$ is a known probability density (e.g. standard Gaussian) and f_θ is an invertible and differentiable mapping, parameterized with θ. The parameters of the invertible transformation can be learned via gradient-based optimization algorithms such that the likelihood $p_x(\mathbf{x})$ is maximized. The absolute determinant $|\det \nabla_x f_\theta(\mathbf{x})|$ of the Jacobian ∇ of f_θ with respect to $\mathbf{x}$ preserves the volume of the density, such that $\int_\mathbf{x} p_x(\mathbf{x})\ d\mathbf{x} = \int_\mathbf{z} p_z(f_\theta(\mathbf{x}))\ d\mathbf{z}$, ensuring that the probability density stays valid under $\mathbf{x}$.

Joint normalizing flows Real-world application of Normalizing Flows often involves high dimensional data such as images. While a multi-dimensional transformation can be efficiently performed using f_θ, computing the absolute determinant of the Jacobian $|\det \nabla_x f_\theta(\mathbf{x})|$ can be computationally very expensive due to the determinant computation. If the data is D-dimensional, the resulting Jacobian is a $D \times D$ matrix resulting in a computational complexity of $\mathcal{O}(D^3)$. Consequently, much of the research in developing NF models has mainly focused on designing transformations f_θ where the determinant can be more efficiently computed [71–73]. One common approach, the so-called *coupling layer* introduced by Dinh et al. [71], is to split the input dimensions $\mathbf{x} = \{\mathbf{x}_{1:d}, \mathbf{x}_{d+1:D}\}$, leave one part untouched while updating the other part using a function which is simple to invert, but which depends on the untouched dimensions in a complex way (Fig. 2.4):

$$\mathbf{y}_{1:d} = \mathbf{x}_{1:d} \tag{2.5}$$

$$\mathbf{y}_{d+1:D} = \mathbf{x}_{d+1:D} \odot \exp(s_\phi(\mathbf{x}_{1:d})) + t_\psi(\mathbf{x}_{1:d}), \tag{2.6}$$

where s_ϕ and t_ψ stand for scale and translation, and are learnable functions from $R^d \mapsto R^{D-d}$, and $\odot$ is an element-wise product. The resulting Jacobian is a triangular matrix, where the determinant can be efficiently computed as $\exp\left[\sum_j s_\phi(\mathbf{x}_{1:d})_j\right]$. Note that, since computing the Jacobian determinant does not involve computing the Jacobian of s_ϕ or t_ψ, these functions can be arbitrarily complex (e.g. DNNs).

As mentioned before, the goal of the NF model is to transform samples of a complex distribution into a *latent* space where they are distributed according to a known simple distribution. However, a single coupling layer leaves some dimensions unchanged, preserving their original distribution. This can be addressed by composing a chain of coupling layers in an alternating pattern, such that the components that are left unchanged in one coupling layer are updated in the next one. Alternating dimensions for each coupling layer, combined with the transformation of a set of dimensions as a function of the other dimensions, allows the NF models to leverage the statistical dependencies between different dimensions and transform a complex distribution into the target (also commonly called *base*) distribution with a known density function.

Since the NF models that I described so far operate on all dimensions jointly (i.e. transformation of some dimensions depends on other dimensions), I refer to these models as *joint normalizing flows*. In the next part, I will describe *marginal normalizing flows*, where the transformation of each dimension is independent of all other dimensions. While this formulation incurs some limitations, it also provides some advantages which we will discuss.

Marginal normalizing flows Marginal NFs are a specific formulation of NF models where the transformation f_θ operates on each dimension independent of all other dimensions such that $f_\theta(\mathbf{x}) = [f_{\theta_1}(x_1), \ldots, f_{\theta_D}(x_D)]^\top$. For the transformation f_θ to be invertible, it must be a monotonic function. Consequently, it can be constructed from a sequence of 1D monotonic functions (e.g. ELU, exp, log, or an affine transformation). While an affine transformation alone does not affect the distribution, using it in combination with other functions is complementary. Specifically, when using fixed transformations, a specific region of the transformation function could be more applicable to the data, where an affine transformation could be used to shift and scale the data accordingly. Another alternative is to use piecewise functions, potentially allowing for more expressive 1D transformations to be learned [74]. Note that, in contrast to joint NF models, in marginal NF models the parameters of dimension-specific transformations are not a function of other dimensions. Therefore, they are defined as parameters that can be learned via gradient-based optimization methods.

The main limitation of marginal NFs is that they cannot take advantage of the statistical dependencies between dimensions to transform the high-dimensional data into a simple distribution such as a standard normal distribution. This limitation can be mitigated by making the parameters of the target distribution learnable such that it best fits the transformed samples. While this formulation is not as expressive as the joint normalizing flow models, it offers some attractive advantages. First, since the transformations operate independently across dimensions, the Jacobian matrix is diagonal. This property allows for an efficient computation of the determinant $\det\nabla_{\mathbf{x}} f_\theta(\mathbf{x}) = \prod_{i=1}^{n} \frac{\partial f_{\theta_i}}{\partial x_i}$ by simply multiplying the diagonal entries.

Second, due to the dimension-specific transformations, by construction, the marginal NF model cannot account for statistical structure in the data. This provides the opportunity to push all the statistical dependencies to be captured by the base distribution and allows us to easily compute conditionals and marginals. Briefly, the conditional distribution $p(\mathbf{x}^{(1)}|\mathbf{x}^{(2)})$, can be computed by first transforming the samples into the latent space, performing the conditioning in the latent space $p(f_{\theta_1}(\mathbf{x}^{(1)})|f_{\theta_2}(\mathbf{x}^{(2)}))$, and finally normalize via the determinant of the Jacobian to yield the conditional distribution in the original space:

$$p(\mathbf{x}^{(1)}|\mathbf{x}^{(2)}) = p(f_{\theta_1}(\mathbf{x}^{(1)})|f_{\theta_2}(\mathbf{x}^{(2)})) \cdot \left|\det\nabla_{\mathbf{x}^{(1)}} f_{\theta_1}\left(\mathbf{x}^{(1)}\right)\right|. \tag{2.7}$$

Importantly, when using joint NF models, where the transformation is not separable across dimensions, computing marginal and conditional densities is generally not possible. For a more detailed derivation see Appendix A of Manuscript 1.

2.3 Implicit neural representations

The material in this section is particularly relevant for understanding the methodologies employed in Chapter 4.

In machine learning, data is traditionally represented on a discrete grid. For instance, images are represented by 1D (grayscale) or 3D (RGB) values at each pixel coordinate. However, the underlying signal is often continuous. Representing data on a discrete grid not only ties the data to a certain resolution but also can be an inefficient use of memory. This issue can be addressed by representing the data with continuous functions, which has gained a lot of popularity lately. For example, an image can be represented by a continuous function mapping the 2D pixel coordinates to their corresponding RGB or grayscale value. When such a mapping is parameterized by a neural network it is typically referred to as an *implicit neural representation* (INR) [75].

Motivated by the abstraction and the efficient coding that is observed in nature, an equivalent approach was introduced by Stanley [76] called *compositional pattern producing networks* (CPPN). One example of such abstraction and complexity in nature is the human genome. While it consists of a finite set of genes, the intricate interactions among these genes can encode a much larger set of structural and functional components, such as the entire human body. CPPNs were introduced as models that capture such abstraction as well as the efficient coding through a composition of functions (e.g. neural networks). If we consider images as an example modality, feeding pixel coordinates x and y as input to the CPPN f results in a pattern that can be conceived as a phenotype whose genotype is f. While CPPNs and INRs are essentially equivalent concepts, for the remaining parts I will use INR for referring to these models.

INRs have been applied on a wide range of modalities and have been shown to successfully represent images [76, 77], 3D shapes [78, 79], 3D scenes [80], videos [81], and audios [82]. In their generic form, these models take the grid points $\mathbf{x}$ (e.g. pixel coordinates) as input and are trained to produce their corresponding feature value $\mathbf{f}$ (e.g. RGB values) as their output:

$$\theta^* = \arg\min_{\theta} \mathcal{L}(f_\theta, \{\mathbf{x}_i, \mathbf{f}_i\}_{i \in \mathcal{I}}), \tag{2.8}$$

where f_θ is the INR with parameters θ, $\mathcal{I}$ represents the set of all grid points in the data (e.g. all pixel locations in the image), and $\mathcal{L}$ is the loss function that is chosen based on the specific task and optimization criterion.

Representing a single image As generative models, INRs can be used to learn an implicit representation of one or multiple images (Fig. 2.5). To learn the implicit representation of a single image, we can feed in the pixel coordinates $\mathbf{x}$ as input to the model and optimize the model parameters θ such that a reconstruction loss (e.g. mean squared error) is minimized. Depending on whether the image pixels have RGB or grayscale values, the final layer is either 3-dimensional or 1-dimensional, respectively.

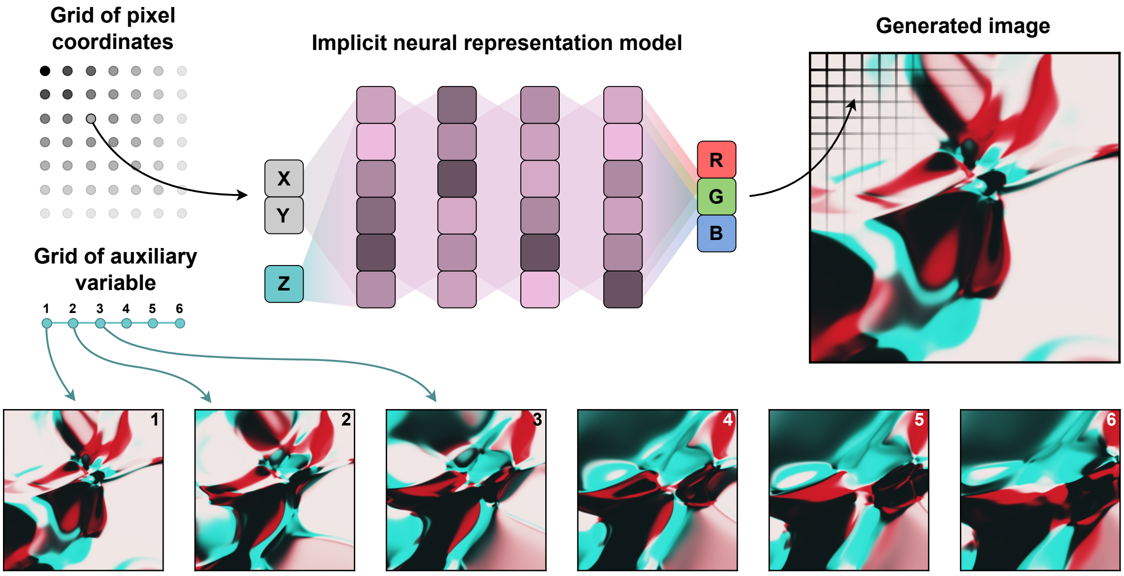

Grid of pixel coordinates
Implicit neural representation model
Generated image
X
Y
Z
R
G
B
Grid of auxiliary variable
1
2
3
4
5
6
1
2
3
4
5
6

Figure 2.5: Demo of implicit neural representations (INR) as generative models of images. To generate a single image, the model takes pixel coordinates as input and outputs RGB (or grayscale) values. Since the resolution of the pixel grid is defined by the user, the image can be generated at arbitrary resolutions. To generate multiple images with the same INR model, we introduce an *auxiliary* variable as an additional input that induces variability in the output even though the model is being fed the same set of pixel coordinates.

Representing multiple images via modulation INRs can also be used to learn an implicit representation of multiple images. However, since multiple images are all defined on the same grid of pixels, to induce variations in the generated images we need to introduce additional variables such that for different values of these *auxiliary variables* the model generates different images even though it is being fed the same set of pixel coordinates (Fig. 2.5 bottom row). Using auxiliary variables to allow INRs to capture multiple data samples is commonly referred to as modulation. Modulations can be implemented in different ways. One approach is to simply add these variables as inputs to the model [28], similar to what is shown in Fig. 2.5. Alternatively, they can be used to apply affine transformations (shift and scale) to the activations of the neural network layers [83–85]. While the INR model learns a representation of the shared data structure across all samples, these modulations encode sample-specific information that can be used for a variety of downstream tasks (e.g. classification).

In the following chapters, I will discuss several projects that use the concepts outlined in this chapter to enhance predictive models of visual cortical neurons by better characterizing their response distribution and deepen our understanding of their functional properties.

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

This chapter is based on the following publication:

- **Mohammad Bashiri***, Edgar Walker*, Konstantin-Klemens Lurz, Akshay Jagadish, Taliah Muhammad, Zhiwei Ding, Zhuokun Ding, Andreas Tolias, and Fabian Sinz. A flow-based latent state generative model of neural population responses to natural images. *Advances in Neural Information Processing Systems*, 34:15801–15815, 2021.

3.1 Motivation: characterizing neural responses beyond stimulus-driven factors

This project aims to model visual sensory neuron activity beyond sensory input-driven factors. While neural responses in the visual cortex vary with visual stimuli, they also exhibit variability to repeated presentations of the same stimuli [86–89], referred to as stimulus-conditioned variability. This stimulus-conditioned variability is often shared (i.e. correlated) across neurons, giving rise to noise correlations [89–91], which has been shown to be related to various factors such as the specific stimulus [92–94], behavioral tasks [95, 96], attention [49–51], and the overall brain state [56, 97]. Recently, [13] showed that around 21% of the total variability in neural responses of mouse visual areas are driven by behavioral variables such as motor information.

In the presence of such non-stimulus-related factors, to gain a comprehensive understanding of the nature of such correlated stimulus-conditioned variability and its functional implications in sensory stimulus processing, it is crucial to develop models that can account for both stimulus-driven and shared stimulus-conditioned variability. Our objective is to capture the stimulus-conditioned response distribution $p(\mathbf{r}|\mathbf{x})$ for n neurons, which encapsulates both population activity $\mathbf{r} \in \mathbb{R}^n$ and the underlying noise correlations when responding to arbitrary sensory stimuli $\mathbf{x}$. However, existing models have predominantly focused on either stimulus-driven activity or stimulus-conditioned correlated variability independently, limiting our ability to accurately capture the joint distribution of neural responses given a stimulus.

Existing Deep Neural Networks (DNNs) excel in modeling stimulus-driven activity but often neglect stimulus-conditioned correlations [17, 19, 21, 23, 98, 99], and can even generate stimuli that yield desirable neural responses [9, 10]. However, current state-of-the-art DNN-based models commonly neglect stimulus-conditioned correlations among

*Equal contribution

neural responses, assuming independence and imposing strong assumptions about the form of the marginal distribution for each neuron, typically a Poisson distribution.

On the other hand, many methods exist for modeling stimulus-conditioned variability. A common approach has been to capture the stimulus-conditioned variability via a, typically much lower-dimensional, shared latent space: $\mathbf{z}$: $p(\mathbf{r}|\mathbf{x}) = \int p(\mathbf{r}|\mathbf{x},\mathbf{z})p(\mathbf{z}|\mathbf{x})\,\mathrm{d}\mathbf{z}$ [56, 57, 61, 65, 100–104]. While these approaches present powerful methods to capture stimulus-conditioned variability, they often suffer from several limitations. One of the limitations is that many of these models capture the conditional distribution $p(\mathbf{r}|\mathbf{x})$ separately for each stimulus $\mathbf{x}$ [56, 57, 61, 104, 105]. These models, therefore, require multiple stimulus presentations for fitting and struggle to generalize to novel stimuli. Furthermore, due to their complexity, these models often either ignore stimulus-driven variability altogether [65, 100, 103] or use simple stimuli such as gratings [56, 57, 61, 101], resulting in additional limitations in their generalization, especially to more complex stimuli such as natural images.

In an attempt to close the gap between these two approaches, in this project, we presented a new model that combines DNN-based and latent variables models to simultaneously capture stimulus-driven as well as stimulus-conditioned variability. We show that the resulting model accurately predicts the distribution of neural responses to unseen stimuli, without the need for repeated presentations to learn stimulus-conditioned variability. Furthermore, we 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.

3.2 Method

To account for the shared variability between neurons we employ Gaussian Factor Analysis (FA) as a latent variable model, while the stimulus-dependence is captured via a DNN that learns to shift the mean of the FA model based on the stimulus. As described in section 2.2.1, FA model assumes the data is distributed according to a Gaussian distribution with a particular low-rank covariance matrix structure. However, neural responses are not Gaussian-distributed, which limits the direct application of FA model to population activity. To address this limitation, variance-stabilizing transformations, such as the square-root function, have been used in the past to make the responses more Gaussian-distributed [56, 65]. However, other transformations, which are not necessarily identical among neurons, may capture the response distribution more accurately. To this end, here we employ Normalizing Flows (section 2.2.2) to learn a marginal transformation such that the transformed responses are marginally distributed according to a Gaussian distribution (Figure 3.1a).

3.2.1 Flow-based factor analysis model

Considering population response $\mathbf{r} \in \mathbb{R}^n$ to a given stimulus x, we define our normalizing flow-based factor analysis (FlowFA) model as:

$$p(\mathbf{r}|\mathbf{x},\theta,\phi) = \mathcal{N}(\overbrace{T_\phi(\mathbf{r})}^{\text{transformed responses}}; \underbrace{\mathbf{f}_\theta(\mathbf{x})}_{\text{output of the CNN}}, \underbrace{\mathbf{C}\mathbf{C}^\top}_{\text{low-rank shared covariance}} + \overset{\text{independent noise}}{\Psi}) \cdot |\mathrm{det}\nabla_{\mathbf{r}} T_\phi(\mathbf{r})| \,. \tag{3.1}$$

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 by a deep network $\mathbf{f}_\theta(\mathbf{x}) \in \mathbb{R}^n$ with learnable parameters θ (Fig. 3.1a,b). And the statistical dependencies between neurons (i.e. noise correlations) are captured via $\mathbf{C}\mathbf{C}^\top$ where $\mathbf{C} \in \mathbb{R}^{n \times k}$ is the so-called *factor loading* matrix that maps samples from the k-dimensional latent space onto the n-dimensional neural space. Note that the FA model is applied on the transformed responses $\mathbf{v} = T_\phi(\mathbf{r})$, where T_ϕ is a marginal normalizing flow: we learn a simple learnable monotonic transformation for each neuron separately. Following the change of variables formula, to construct a proper density over $\mathbf{r}$ we introduce the absolute determinant $|\det \nabla_\mathbf{r} T_\phi(\mathbf{r})|$ of the jacobian ∇ of T_ϕ with respect to $\mathbf{r}$ into Eqn. 3.1.

3.2.2 Zero-inflated flow-based factor analysis model

Neural responses recorded via two-photon Calcium imaging often contain a large portion of zeros, leading to a zero-inflated distribution [106]. To avoid potential problems (e.g. overfitting to the peak at zero) due to this phenomenon, we extend FlowFA to a mixture model that models neural responses below and above a threshold value ρ with two separate, non-overlapping distributions. The peak at zero is captured by modeling the responses below the threshold (i.e. "zero" responses) using a uniform distribution, while FlowFA is used to capture responses above the threshold:

$$p(\mathbf{r}|\mathbf{x}) = \left(\prod_{\{i: r_i \leq \rho\}} \frac{1 - q_i(\mathbf{x})}{\rho} \right) \cdot \left(\prod_{\{i: r_i > \rho\}} q_i(\mathbf{x}) \right) \cdot \mathcal{N}(T_\phi(\mathbf{r}_+); f_{\theta,+}(\mathbf{x}), \mathbf{C}_+\mathbf{C}_+^\top + \Psi_+) \cdot |\nabla T_\phi(\mathbf{r}_+)|, \quad (3.2)$$

where $q_i(\mathbf{x})$ is the probability of the response being above the threshold ρ and is modeled as a function of the stimulus via a DNN $\mathbf{f}_\theta$. $\mathbf{r}_+$ 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 in Eq. (3.1).

3.3 Results

First, we applied the FlowFA model to synthetic data and showed that it faithfully recovers invertible transformations. While these results and their detailed description can be found in Manuscript 1, here I will focus on the findings when we applied our model to real neuronal responses. These responses were recorded via a two-photon microscope, from two mice while they were presented with grayscale natural images, spanning three visual areas: primary visual cortex (V1) and lateromedial area (LM) in one mouse (referred to as "scan 1"); V1 and posteromedial area (PM) in another mouse (referred to as "scan 2"). More details about the dataset and how the models were fitted to the data can be found in Manuscript 1.

3.3.1 Capturing cortical response distribution

We trained the flow-based models (ZIFFA, FlowFA) on population responses for different values of latent dimensions $k \in \{0, 1, 2, 3, 10\}$, and compared the results against multiple control models: 1) flow-based model with fixed transformation (FixedFA) to assess the importance of learnable transformation, and 2) models based on Poisson [21, 98] and Zero-Inflated Gamma [106] distributions to show the importance of the

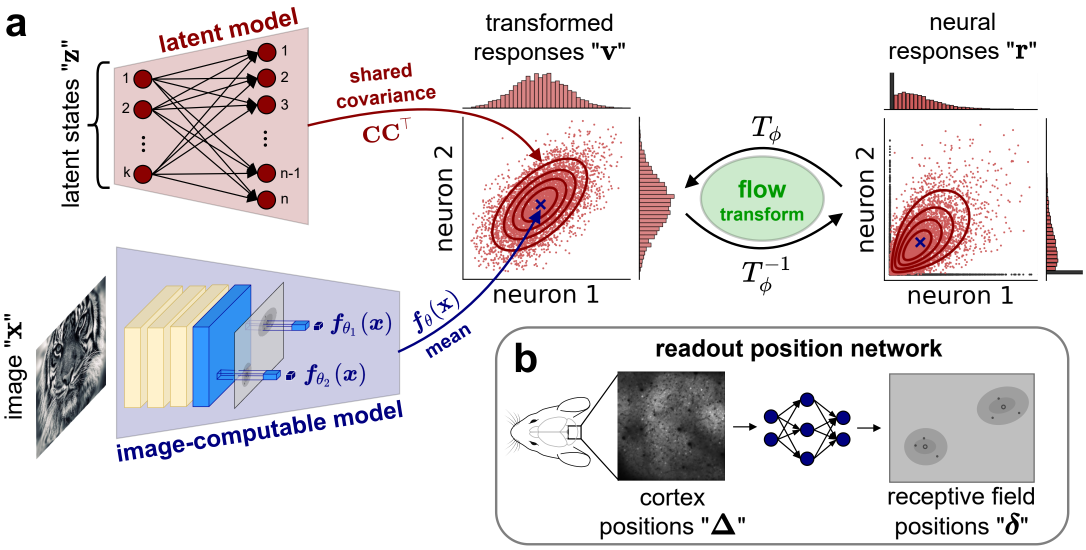


Figure 3.1: Flow-based Factor Analysis model. **a:** Schematic of the flow-based model relating all relevant variables. **b:** Schematic of the sub-network used by the image-computable model (i.e. DNN) to map cortical positions into receptive field positions. Refer to the Method section of Manuscript 1 for the details. The figure is reprinted from [22].

distribution choice. These controls have previously been successfully applied on neural responses [21, 106] and have an important distinction compared to the Flow-based models: they assume independence among neurons. Importantly, when the number of latent dimensions in the flow-based models is set to 0, all models assume independence among neurons. We measured the model performance by computing the log-likelihood as well as the conditional correlations. Importantly, the prediction used for computing conditional correlations not only was conditioned on the image but also on the responses of all the other neurons to that same image. This allows the model to take advantage of the dependencies between neurons learned by the latent variable model.

The ZIFFA model outperformed all other models in terms of log-likelihood and with an increasing number of latent dimensions its performance consistently and considerably improved beyond the control models that assume independence among neurons (Fig. 3.2a). Interestingly, we observed that the ZIFFA model exhibited slightly lower correlation performance compared to models with fixed transformations, reflecting that a higher likelihood does not necessarily correspond to a higher correlation. Additionally, the ZIFFA model outperformed all FixedFA models in terms of likelihood, which underscores the importance of neuron-specific learnable transformations in accurately capturing the distribution of neural responses (Fig. 3.2b). Overall, the results suggest that the ZIFFA model is able to capture the (marginal) neural response distributions more accurately than other models while at the same time learning and taking advantage of the statistical dependencies between neurons.

3.3.2 Uncovering biological insights from the model

We now explore the utility of our model in uncovering potential biological insights. All following analyses were performed on the trained ZIFFA model with a 3-dimensional latent state.

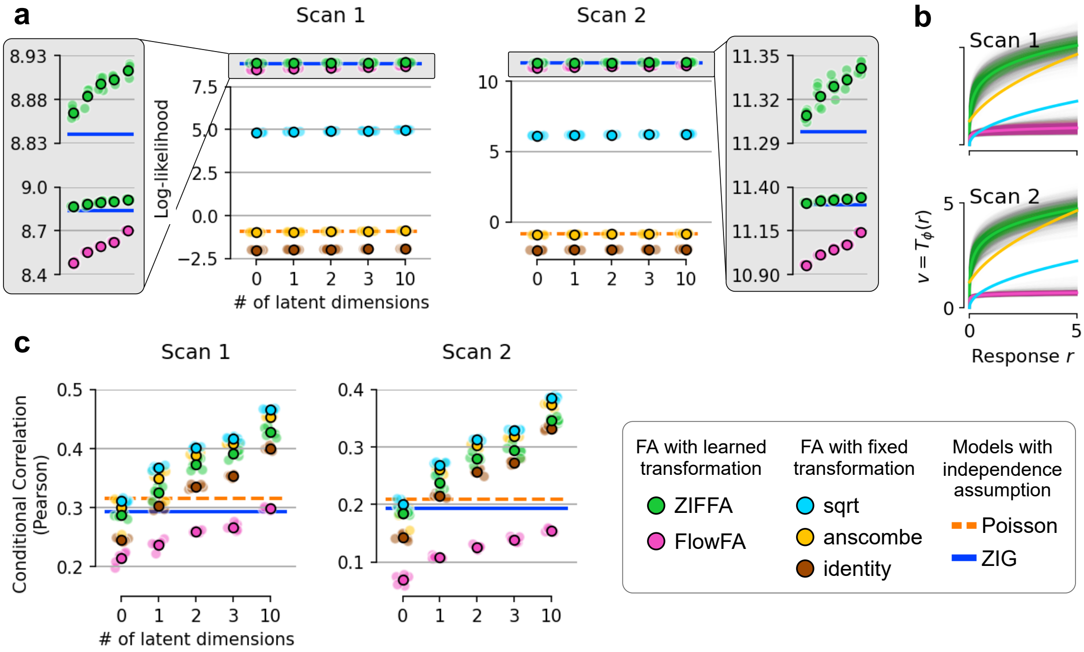


Figure 3.2: Comparison of models trained on the mouse visual cortical population responses to natural images. **a**: log-likelihood computed for models trained on scan 1 (left panel) and scan 2 (right panel). Values for both individual (lighter shade) and average (darker shade) performance of a model trained under various random seeds are shown. The 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. The format is similar to **a**. The figure is reprinted from [22].

Model-based visual area identification Multiple visual areas in mice exhibit retinotopies that demonstrate a flipped relationship between each other [107]. In simple terms, this implies that when a point traverses the cortical surface and crosses the boundary between two flipped areas, its corresponding point in visual space reverses its direction of movement. Our model incorporates a component network, called *readout position network*, that effectively predicts the location of each neuron's receptive field (RF) δ as a function of its cortical location Δ (Fig. 3.1b). In other words, the readout position network learns the mapping from neurons' location on the cortex to their RF location in the visual field. Here we show that this learned mapping can be used to detect distinct visual cortical areas. To this end, we examined the sign of the determinant of the Jacobian matrix det $\frac{\partial \delta}{\partial \Delta}$, which describes the relationship between RF positions and cortical positions. If the determinant of the jacobian has a negative sign it means the direction of movement in the visual field is flipped w.r.t the changes in the cortical position, and it is not flipped if the sign is positive. Finally, comparing the resulting sign across different areas reveals distinct areas identified by the learned mapping. A comparison between the areas identified by our model and the experimentally identified areas reveals a highly accurate correspondence (see Fig. 3.3a, left vs. right panels). Notably, these findings suggest that our model could enable the identification of distinct visual areas solely through the analysis of responses to natural images, eliminating the need for additional experiments dedicated to area identification.

Latent variables are related to behavioral, functional, and structural factors In the subsequent analyses, we delved into the latent states and their associations with various behavioral variables, as well as the anatomical and functional characteristics of visual sensory neurons. First, we inferred *orthonormalized latent states* which are uniquely ordered based on the extent to which each latent dimension accounts for response variability (see 2.2.1 and Manuscript 1 for detailed explanations) and compared them to the behavioral variables recorded during the experiment. Notably, the orthonormalized latent states inferred from the ZIFFA model exhibited strong correlations with behavioral variables, including pupil dilation (Fig. 3.3d), which aligns with prior research using pupil dilation as a surrogate measure for arousal and attention [108–112]. Interestingly, pupil dilation correlated most strongly with the second latent dimension which was consistent not only across models initialized and trained with different random seeds but also across the two mice with R^2 values of 0.53 ($p < 0.001$, two-tailed test for significance of correlation [113]) and 0.63 ($p < 0.001$) for scan 1 and scan 2, respectively, comparable to values previously reported [13]. The substantial correlation observed between the latent states and established indicators of global brain state, such as pupil dilation, implies that the latent model effectively captures meaningful dependencies and shared factors within the neural population.

Next, we explored whether the effect of the latent states on the neurons, captured by the factor loading matrix $\mathbf{C}$, is related to their cortical or RF positions. To this end, we plotted the sign and magnitude of the weights mapping from the latent state to each neuron on the cortical position (Fig. 3.3b) or the RF positions of the neurons (Fig. 3.3c). We observed that while some dimensions exhibit a rather global effect across different visual areas (Fig. 3.3b: dimension 1 for both scans), the effect of some other latent dimensions varies systematically across brain areas where the latent dimension has generally opposite effect on different areas (Fig. 3.3b: dimension 2 for both scans). Additionally, we observed a differential effect of some latent dimensions where the effect seemed to vary as a function of RF or cortex position within each area (Fig. 3.3c: dimension 3 for both scans).

While conclusive biological interpretations would require additional rigorous experiments and analyses, our results illustrate the utility of our model for uncovering the functional and structural implications of the behavioral or internal processes associated with the inferred latent states.

3.4 Discussion

Bridging the Gap in Sensory Neuron Modeling In this project, we developed a model that addresses an important gap in the field: a predictive model that simultaneously accounts for both stimulus-driven and stimulus-conditioned variability in neural responses. Our model combines state-of-the-art DNN-based models with a flow-based factor analysis model, allowing 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. Using the activity of thousands of neurons from multiple areas of the mouse visual cortex in response to natural images, we trained a model that achieves state-of-the-art performance in capturing the distribution of neural responses. Importantly, it also yields latent states that have meaningful relations to behavioral variables, as well as anatomical and functional properties of visual sensory neurons. This directly addresses a gap in modeling the activity of sensory neurons by integrating both sensory input and internal processes into a unified predictive framework, thereby providing a foundational step in understanding how internal states and behavioral variables affect neural responses.

Limitations and future extensions Our proposed method has two potential limitations. Firstly, we used simple learnable marginal transformations to transform neural responses, aiming to align their distribution more closely with a Gaussian distribution. However, alternative approaches, such as neural spline flows [74], could potentially yield improved outcomes by employing more expressive transformations. Secondly, while the learned transformation implicitly incorporates a relationship between the noise correlation structure (covariance matrix) and the stimulus through the learned transformation, an explicit dependency could be a valuable future extension of our methodology. It is worth considering that using a joint transformation that operates on all neurons simultaneously may enhance overall performance, but it may also impose limitations on other aspects of the model, such as the ability to compute conditional densities. Additionally, employing a joint flow could potentially account for some of the dependencies among neurons, thereby impacting the information content and interpretability of the accompanying latent variable model in capturing statistical dependencies.

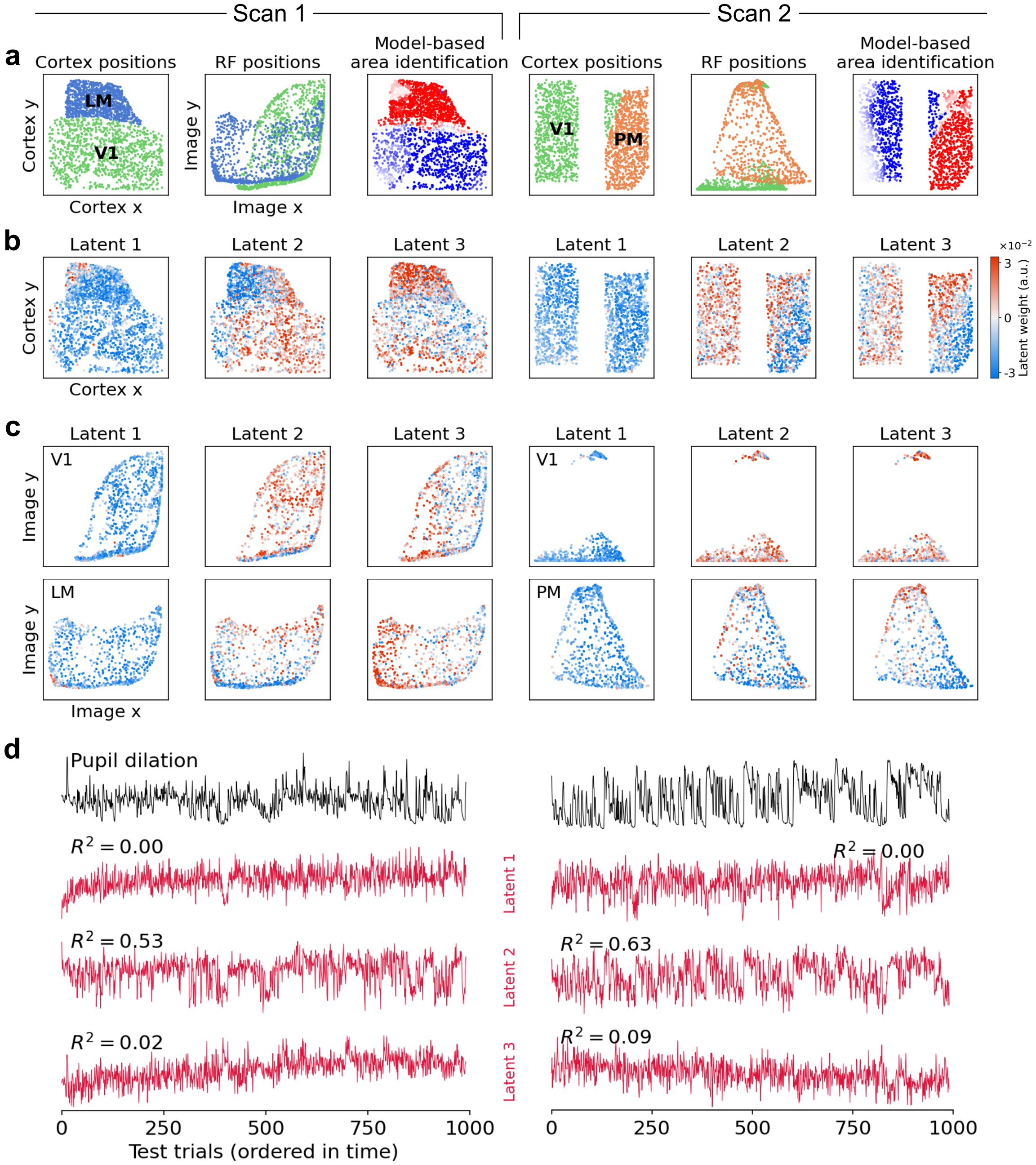


Figure 3.3: Analysis of the ZIFFA model with 3-dimensional latent state ($k = 3$). **a**: Model-based area identification from responses of visual sensory neurons to natural images. Left panel (Cortex positions): cortical position of the recorded neurons color-coded by experimentally identified areas (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. The figure is reprinted from [22].

4 Learning invariance manifolds of visual sensory neurons

This chapter is based on the following publication:

- Luca Baroni[*], **Mohammad Bashiri**[*], Konstantin F Willeke, Ján Antolík, and Fabian H Sinz. Learning invariance manifolds of visual sensory neurons. In *NeurIPS Workshop on Symmetry and Geometry in Neural Representations*, pages 301–326. PMLR, 2023.

4.1 Motivation: neural system identification through the lens of invariances

In the previous chapter, I described how we used a latent variable model applied to neural responses to capture their distribution more accurately and used the resulting model to gain biologically-relevant insights about the visual sensory neurons. In this project, we adopt an alternative latent variable model tailored to the stimulus space, aiming to elucidate the feature selectivity and invariances that visual sensory neurons exhibit in response to varying stimuli.

Reliable and robust object recognition is believed to require neural mechanisms that exhibit selectivity towards certain stimulus features while maintaining invariance to others, such as spatial location or rotation. Such mechanisms enable animals to generalize their visual capabilities to various transformations such as different viewing conditions and contexts. In order to gain a better understanding of the computational mechanisms that underlie the robustness and generalizability of the biological visual systems, it is thus important to identify the features that strongly drive neural activity and uncover the specific transformations of these features that do not alter neural responses, known as *single cell invariances.* A prominent example of a transformation that many visual sensory neurons (i.e. complex cells) are invariant to is phase transition [6]. Importantly, this discovery, like many other discoveries of invariances in the past [114–116], has typically relied on a hypothesis-driven approach that involves presenting carefully chosen stimuli based on the intuition of the experimenter. Considering the immense dimensionality of images and the constraints of experimental time, this approach quickly becomes impractical especially when dealing with higher-level areas where the encoding of visual information becomes more complex.

An alternative approach is to take advantage of the powerful DNN-based predictive models of the visual system. Several studies have recently used these models to find single [9, 10, 24] or multiple [47] maximally exciting inputs (MEIs) for visual sensory neurons. However, existing methods solely identify a discrete collection of stimuli

[*] Equal contribution

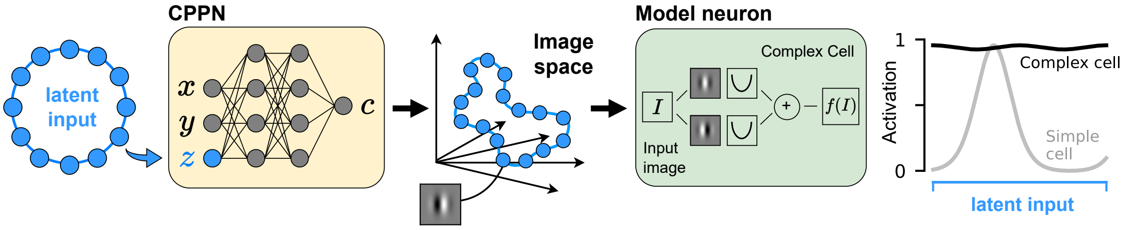


Figure 4.1: Our method uses a CPPN to map a simple low-dimensional latent space onto a complex high-dimensional manifold in the image space. Images from this manifold result in diverse but maximally exciting stimuli for a model neuron. Here we show a schematic of this method applied to a complex cell. Corresponding activations for a simple cell are also added as reference. The figure is reprinted from [28].

from the underlying invariance manifold, which limits their utility for generating new images that are different from the other images but still belong to the same manifold. Considering the high dimensionality of images, understanding how such discrete points in the image space are connected can be highly non-trivial, especially in higher visual areas as there may exist multiple transformations that neurons are invariant to.

In this project, we developed a data-driven method that identifies a manifold in the stimulus space along which all images maximally, and equally, activate a target visual sensory neurons. We refer to this manifold as the MEI invariance manifold (or simply *invariance manifold*). Our method is based on Implicit Neural Representation (INR) models [77, 80, 117] which provide a reparameterization of an image, and contrastive learning [118] which encourages the resulting manifold to capture the true underlying manifold. The main advantages of our method compared to existing methods are two-fold: our method 1) is a **data-driven** approach which allows us to identify unexpected and novel invariances, and 2) uses a **generative** approach which allows us to generate and experiment with new images that lie on the same invariance manifold.

4.2 Method

While previous approaches directly optimized pixel values to identify MEIs, here we use INRs to optimize a reparameterized version of the image (Fig. 4.1). In the context of image generation, INRs are artificial neural networks mapping pixel positions (x, y) to pixel RGB (or grayscale) values. These models have recently gained a lot popularity in the computer vision community as implicit representations of shapes and radiance fields [77, 78, 80]. While using INRs allows us to learn a manifold in the image space, there are no guarantees that the generated images are diverse and the resulting manifold spans a reasonable extent of the true underlying manifold. In other words, the INR can collapse into a single point in the image space or learn a limited range of the true underlying manifold. To circumvent this limitation we use a contrastive learning objective to encourage diversity among the generated images.

4.2.1 Learning invariance manifold via implicit neural representations

Our goal is to use an INR as a generative model that generates multiple images. To facilitate this, we extend the INR's input space by incorporating additional latent dimensions, denoted by $\mathbf{z}$. This allows the model to generate different images for different values of $\mathbf{z}$ even though it is being fed the same pixel coordinates [77]. We implemented the INR as a fully connected network with 8 hidden layers, where each layer contained 15 units and was followed by a batch normalization and a leaky ReLU nonlinearity. The output layer of the INR model had a single unit followed by a Tanh nonlinearity resulting in grayscale images bounded between -1 and $+1$.

Positional encoding To allow control over the characteristic spatial frequency of the patterns generated via INR model, instead of directly using pixel positions as input to the model we used positional encoding of the pixel positions, which we obtained via Fourier mapping [80, 119] (for details please refer to Manuscript 2).

Latent state topology Similar to the diversity of patterns encoded by neurons the transformations that the neural population is invariant to can be diverse. For instance, one neuron could be invariant to phase (e.g. phase invariance in complex cells) while another neuron is invariant to the angle (e.g. rotation invariance). Additionally, similar to mixed selectivity observed in higher visual areas, neurons can show invariance to multiple transformations (e.g. phase and angle). In this work, we investigated the manifold's topology in two aspects: 1) the number of transformations it accommodates, and 2) the periodic or non-periodic nature of the manifold's geometry. In particular, we considered 1D and 2D latent spaces as well as non-periodic (corresponding to a line or sheet topology) and periodic (corresponding to a circle or torus topology) boundary conditions for the latent space.

4.2.2 Encouraging diverse MEIs via contrastive learning

Images generated by the INR model g_ϕ must satisfy two criteria: 1) they all should maximally activate a target neuron, and 2) they should be as different as possible from each other to ensure that the learned manifold reasonably spans the true underlying invariance manifold. Therefore, the full objective contains two terms:

$$\mathcal{L} = \mathcal{L}_{act} + \mathcal{L}_{contrastive}$$

The first term is the activation of the target neuron as predicted by the neural encoding model f_θ:

$$\mathcal{L}_{act} = f_\theta(g_\phi(\mathbf{z}_i)) \cdot \frac{1}{\alpha_{\mathrm{MEI}}} \tag{4.1}$$

where $\mathbf{z}_i \in \mathbb{R}^D$ is a single point from a grid of values covering a D-dimensional latent space. We divide the resulting activation by the target neuron's MEI activation, resulting in a maximum value of 1 for $\mathcal{L}_{act}$. Note that the only learnable parameters here are ϕ, the parameters of the INR model. By maximizing this objective we ensure that the generated images maximally activate the target neuron.

The second term, which is based on soft nearest neighbor contrastive objective [120, 121],

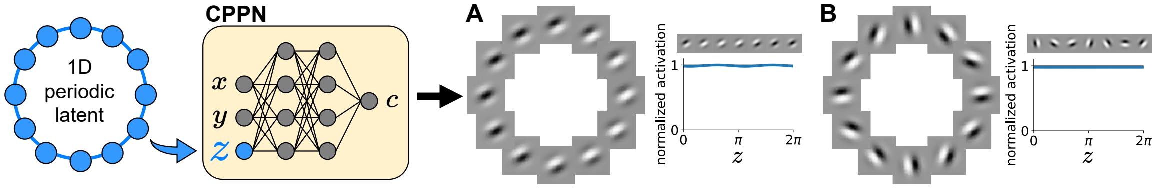


Figure 4.2: Invariances generated from equally spaced points in a periodic 1D latent space in the case of a complex cell (A) and of an orientation invariant neuron (B) and activation values to different images in the corresponding learned manifold. The figure is reprinted from [28].

encourages diversity across the generated images:

$$\mathcal{L}_{\text{contrastive}} = c \cdot \log \frac{\frac{1}{N_+} \sum_{\mathbf{z}_j \in \mathcal{Z}_+} \exp(\text{sim}(g_\phi(\mathbf{z}_i), g_\phi(\mathbf{z}_j))/\tau)}{\frac{1}{N_-} \sum_{\mathbf{z}_k \in \mathcal{Z}_-} \exp(\text{sim}(g_\phi(\mathbf{z}_i), g_\phi(\mathbf{z}_k))/\tau)}, \tag{4.2}$$

where $\mathcal{Z}_+$ is a set of *positive* images generated using the points in the latent space that are close to the anchor grid point $\mathbf{z}_i$ and $\mathcal{Z}_-$ is a set of *negative* images that are generated from points further from $\mathbf{z}_i$ (see Manuscript 2 for details). We used cosine similarity to measure the similarity between two images, where τ is the temperature parameter that controls the diversity of the generated images, and c controls the contribution of the contrastive objective to the complete objective. Optimizing this objective ensures that images in the positive set resemble each other, while those in the negative set diversify, thereby encompassing a broader section of the invariance manifold. As the final step, we average across all grid points in the latent space, resulting in the complete objective function which we maximize during training:

$$\mathcal{L} = \frac{1}{N^D} \sum_{z_i \in \mathcal{Z}} \left(\frac{f_\theta(g_\phi(\mathbf{z}_i))}{\alpha_{\text{MEI}}} + c \cdot \log \frac{\frac{1}{N_+} \sum_{\mathbf{z}_j \in \mathcal{Z}_+} \exp(\text{sim}(g_\phi(\mathbf{z}_i), g_\phi(\mathbf{z}_j))/\tau)}{\frac{1}{N_-} \sum_{\mathbf{z}_k \in \mathcal{Z}_-} \exp(\text{sim}(g_\phi(\mathbf{z}_i), g_\phi(\mathbf{z}_k))/\tau)} \right), \tag{4.3}$$

where D is the number of latent dimensions and N is the number of points per dimension, resulting in a total of N^D points in the latent space.

4.3 Results

We tested our method on simple Gabor-based model neurons with known and exact[1] invariances as well as DNN-based predictive models of neural responses in macaque primary visual cortex. For simulated neurons, we considered Gabor-based models that are either invariant to a single transformation (e.g. phase or rotation) or to two transformations (i.e. phase and rotation). While here I discuss the main results of the project, additional complementary results and analyses can be found in Manuscript 2.

4.3.1 Learning invariance manifolds with 1D and 2D latent spaces

Invariance to a single transformation To test whether the method can capture single invariances we applied it to model neurons that elicit either phase-invariance

[1] neurons with "exact" invariances show the same level of responsiveness to all the images generated from the invariance manifold.

or rotation-invariance. These model neurons were constructed by max-pooling across the responses of several simple cell models. For instance, the rotation-invariant model was constructed by max-pooling across multiple model neurons where each neuron was tuned to a different angle. For these neuron models, our method identifies the invariance manifold almost perfectly (Fig. 4.2) and the values of the latent input correspond to the angle characterizing the invariance.

Note that, in a real scenario, where we apply the method to biological neurons, the number of transformations that a neuron is invariant to is not known. Consequently, it is possible that a neuron is invariant to more transformations than the model can capture. For instance, consider learning the invariance manifold of a neuron that is invariant to changes in both phase and rotation with a model that has a 1D latent space. What invariance manifold would the model identify in this case? While in this case, it is not possible to learn the complete underlying invariance manifold, in Manuscript 2 Fig. S4 we show that our method still learns a meaningful submanifold of the underlying higher-dimensional invariance manifold.

Invariance to multiple transformations Next we considered an INR model with non-periodic and periodic 2D latent space, corresponding to a sheet and a torus topology, respectively. We applied both of these latent space topologies on neuron models with no invariances as well as 1D and 2D invariance manifolds (Fig. 4.3). Ideally, when the dimensionality of the latent space is larger than the number of transformations that a neuron is invariant to, the model should ignore the additional latent dimensions and constraint the manifold to a number of latent dimensions that are necessary to capture the invariance manifold. For instance, in the case where a neuron has no invariance then a model with a 2D latent space should ignore both dimensions and collapse the resulting images into a single point in the image space.

Our results show that not only the model learns to ignore latent dimensions that are not needed for capturing the underlying invariance manifold, but when the dimensionality of the latent space matches the dimensionality of the true invariance topology it disentangles the latent space nearly perfectly. That is, it learns to associate a single dimension in the latent space with a specific transformation in the image space. These results are particularly relevant as they show the utility of our method in providing a clear identification of the invariances and control over them via the latent space.

4.3.2 Learning invariance manifolds of macaque V1 complex cells

Lastly, we applied our method with a 2D latent space on a DNN-based predictive model of macaque V1 neurons (see Manuscript 2 for details about the neural encoding model). We used an ensemble of ANNs as a model of macaque V1 neurons and applied our method on complex cells that we identified using a nonlinearity index [16]. Fig. 4.4 shows that the INR identified phase invariance in the selected neurons: it generated a variety of maximally exciting images resembling Gabor filters and parameterized their phase transformation with one of the latent space dimensions while ignoring the other dimension (Fig. 4.4E).

Note that, in contrast to the Gabor-based neuron models with exact invariances, biological neurons cannot be expected to present exact invariances over maximally exciting stimuli. Furthermore, our experiments here are performed on neural network

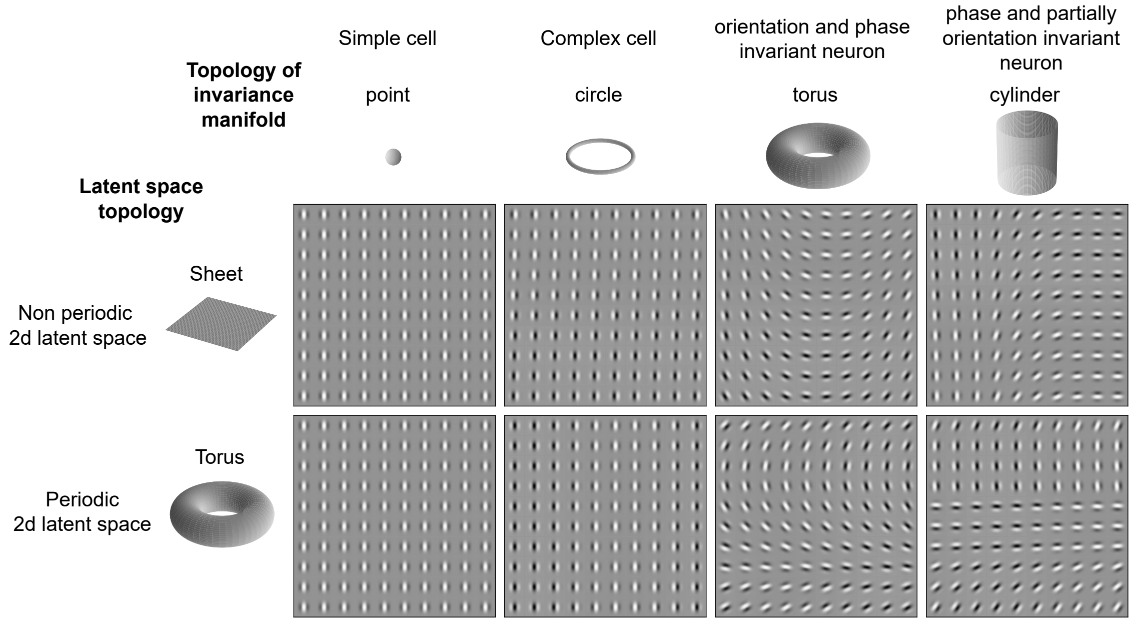

Simple cell
Complex cell
orientation and phase invariant neuron
phase and partially orientation invariant neuron
Topology of invariance manifold
point
circle
torus
cylinder
Latent space topology
Sheet
Non periodic 2d latent space
Torus
Periodic 2d latent space

Figure 4.3: Invariances learned with 2D latent space for different configurations of latent space topology and topologies of the ground truth invariance manifold. The figure is reprinted from [28].

models fitted to neural responses, which despite achieving high predictive performance, are not perfect. Therefore, in the case of the biological neurons, a meaningful definition of the MEI invariance manifold should be more forgiving. Nevertheless, overall, our results demonstrate the utility of our method in identifying invariances of biological neurons.

4.4 Discussion

Data-driven identification of invariance manifold In this project, we developed a data-driven method that directly contributes to closing the gap in understanding the complexity of encoding properties in visual sensory neurons. Our method combines implicit neural representation models with a contrastive objective to learn an invariance manifold in the image space. This manifold maximally excites a target neuron, allowing us to characterize the complexity of their invariances. We tested our approach on both simulated neurons and predictive models of macaque V1 complex cells and showed that it successfully uncovers the invariance manifold in both cases. In contrast to previous approaches, our method learns a smooth reparameterization of the invariance manifold that allows generating new images from the manifold, and when a neuron exhibits multiple invariances it learns to disentangle each transformation and associates it with a different latent dimension. Furthermore, when there is a mismatch between dimensionality D_{model} of the latent space and the dimensionality D_{true} of the underlying invariance manifold it still yields meaningful results by either learning a submanifold (when $D_{\text{model}} < D_{\text{true}}$) or ignoring unnecessary latent dimensions (when $D_{\text{model}} > D_{\text{true}}$).

Limitations and future extensions Our study is limited by its focus on macaque V1 complex cells, chosen to demonstrate proof of concept through well-established

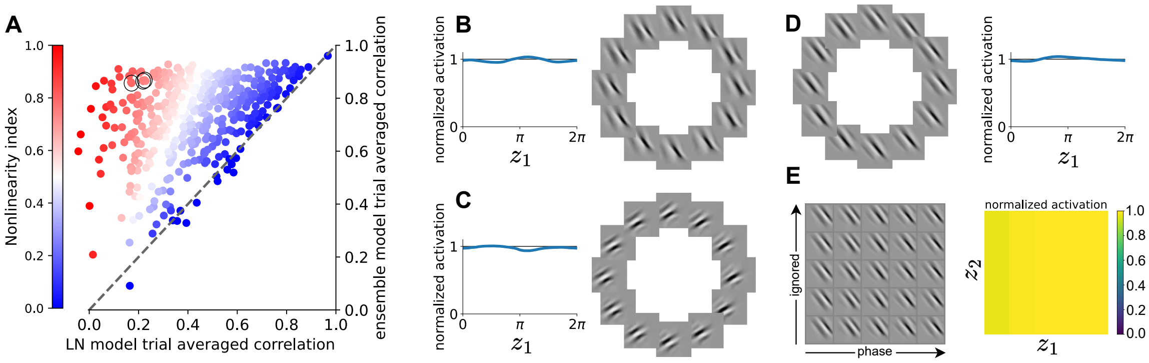


Figure 4.4: **A**: Nonlinearity index of macaque V1 neurons. Black circles highlight the neurons shown in panels **B**–**E**. **B**–**E**: Phase invariances identified with periodic 1D latent (**B**–**D**) and with 2D non-periodic latent(**E**) and corresponding activations. For visualization purposes, MEIs are cropped around the receptive field of the neurons. The figure is reprinted from [28].

invariances such as phase invariance. Despite this narrow scope, our systematic approach can be adapted for more diverse neural populations and higher-level visual areas, presenting exciting avenues for future research. One direction is to apply the method on multiple neurons instead of a single neuron. For instance, by associating a unique learnable fingerprint to each neuron, neurons could be clustered based on their functional properties in this potentially low-dimensional fingerprint space. Similarly, by modifying the objective function, which here was focused on learning invariances, we can learn tuning direction in the image space where many neurons are tuned to a specific input generated from the same manifold shared across all neurons. Furthermore, applying our method to higher visual areas where neurons are potentially invariant to multiple, and more complex, transformations allows us to compare areas based on their invariances and investigate how simple invariances in primary visual areas give rise to complex invariances in higher visual areas.

5 Bayesian oracle for bounding information gain in neural encoding models

This chapter is based on the following publication:

- Konstantin-Klemens Lurz*, **Mohammad Bashiri***, Edgar Y. Walker, and Fabian H Sinz. Bayesian oracle for bounding information gain in neural encoding models. In *International Conference on Learning Representations (ICLR)*, 2023.

5.1 Motivation: from correlation to full likelihood-based evaluation metrics

In recent years, neural system identification has seen many advancements in building neural encoding models, i.e. predictive models of neural activity [9, 11, 15, 16, 18, 21, 23, 122]. However, these models are most commonly evaluated using mean-based measures, such as correlation or fraction of explainable variance explained (FEVE), which are mainly focused on how well the model captures the conditional mean (i.e. stimulus-driven variations in neural responses). Consequently, when evaluating these models other aspects such as how well they characterize the stimulus-conditioned variability are ignored. Importantly, this stimulus-conditioned variability is not just noise and is often correlated among neurons giving rise to noise correlations [89–91], which are known to be related to multiple behavioral and cognitive variables [13, 22, 49–51, 56, 95–97].

Furthermore, many normative theories that link first principles to neural response properties, such as the probabilistic population code [123] and neural sampling [124, 125], make predictions about the variability of neural responses around the mean [96, 126, 127]. This calls for both developing models that capture more than just the conditional mean, and devising evaluation metrics capable of appropriately assessing these advanced models. Therefore, in this project, we focused on alternative measures that allow us to evaluate neural encoding models not just based on how well they capture the stimulus-driven variations in neural responses but also on other aspects such as stimulus-conditioned variability.

Specifically, we focused on a likelihood-based evaluation metric which allows model evaluation based on complete response distribution. In contrast to correlation which is an interpretable measure since it is naturally bounded between -1 and $+1$, it is not trivial to interpret likelihood values without putting them into context. Ideally, we would want to normalize model likelihood such that it falls within an interpretable range of values. There are additional reasons besides interpretability that make a normalization to a bounded and interpretable scale desirable: 1) Assessing whether a

*Equal contribution

model has achieved the *best possible* performance for a given dataset, and 2) comparing models that are trained on different datasets, which due to noise can exhibit different levels of achievable performance. To this end, Normalized Information Gain (NInGa) [128] can be used, which uses an estimated lower and upper bound to put the model likelihood on a meaningful and interpretable range. The challenge, however, is to obtain such bounds for noisy neural responses.

Here, we propose a method for robustly estimating such bounds for neural responses. We show how a naïve Point Estimate (PE) approach fails to yield a robust estimate, especially for higher moments beyond the mean, due to several common characteristics of recorded neural responses, namely few samples, sparsity of responses, and low signal-to-noise ratio. To address this shortcoming, we propose a generalization of the PE approach to a full Bayesian approach by using the posterior predictive distributions. We show that the resulting approach is robust to all the above-mentioned complexities of neural responses and we provide derivations for a variety of common distributions including the state-of-the-art zero-inflated mixture models. Finally, by applying our approach on zero-inflated mixture models of neural responses, we show that they achieve around 90% of the maximum achievable performance.

5.2 Method

Consider neural response y to stimulus x with the conditional distribution $p(y|x)$. In order to evaluate a model likelihood $\hat{p}(y|x)$ in an interpretable fashion we can normalize it using Normalized Information Gain (NInGa):

$$\mathrm{NInGa} = \frac{\langle \log \hat{p}(y \mid x)\rangle_{y,x} - \langle \log p_0(y)\rangle_{y,x}}{\langle \log p_*(y \mid x)\rangle_{y,x} - \langle \log p_0(y)\rangle_{y,x}}, \tag{5.1}$$

where the *Null model* $p_0(y)$ is a marginal distribution over response y which does not account for any stimulus-related information, and the *Gold Standard (GS) model* $p_*(y|x)$ reflects the best possible approximation of the true conditional distribution $p(y|x)$. Compared to the model under evaluation the GS model has access to more information, such as responses to repeated presentations of the same stimulus. Using these responses to the same stimulus we estimate the parameters of the GS model in a leave-one-out fashion: given a set of n repeats, the GS parameters of a target repeat i are estimated using the $n-1$ other repeats $\backslash i$.

As we will discuss below, the parameters of the GS model can be computed as point estimates via moment matching. However, such a point estimate approach fails to yield a robust estimate of the upper bound, mainly due to multiple characteristics of the recorded neural responses: 1) low number of repeats because of limited experimental time, 2) sparsity of responses, and 3) high signal-to-noise ratio. Note that the Null model is not noticeably influenced by these limitations because it has many more samples as it is not conditioned on the stimulus. In our specific experimental setup, where each of the 1000 stimuli is presented 10 times, the Null model parameters are estimated on 1000 samples, whereas each GS model is confined to a mere 10 samples for its parameter estimation.

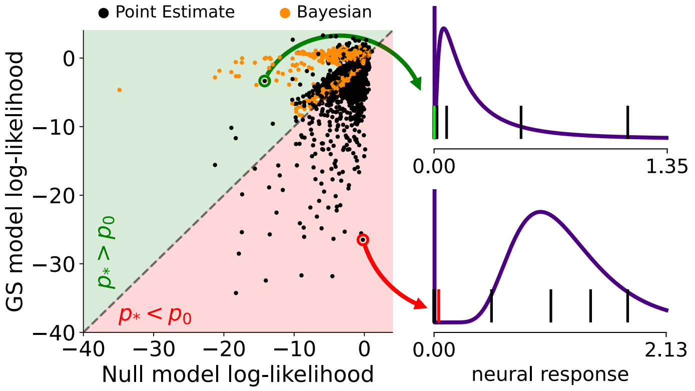


Figure 5.1: Comparison of lower and upper bound likelihood estimates (Null vs GS) per neuron. **Left:** For many neurons, the PE approach yields worse GS than the Null score, while the Bayesian method results in the expected outcome. **Right:** Two example neurons where the PE method fails (red) or succeeds (green). The figure is reprinted from [29].

5.2.1 Naïve point estimate of the upper bound

Considering the upper bound estimate $p_*(y_i|\mathbf{y}_{\setminus i}, x) = p(y_i|\theta_i)$ for a target repeat i, parameterized by θ_i, we obtain a point estimate of θ using the $n-1$ other repeats as $\theta_i = f(\mathbf{y}_{\setminus i})$, where f represents moment matching. The conditioning on stimulus x arises because the GS model's parameters are estimated per stimulus, relying exclusively on the corresponding neural responses $\mathbf{y}$. For brevity, however, we will drop the conditioning on x for the remainder of this manuscript.

To test the PE approach, we chose the model distribution $\hat{p}(y|x)$ to be a zero-inflated Log-Normal (ZIL) distribution over real neural responses:

$$\hat{p}(y|x) = (1 - q(x)) \cdot \underbrace{p_u(y)}_{\text{uniform}} + q(x) \cdot \underbrace{\text{Lognormal}(y|\mu(x), \sigma^2(x), loc)}_{\text{positive distribution on } [\tau, \infty)}, \tag{5.2}$$

where the mixing proportion q and the parameters μ, σ depend on x. Note that the location parameter loc is the minimum value of the support range for the Log-Normal distribution which is fixed $loc = \tau$. Our goal is to obtain the GS model by estimating the parameters $\theta = \{q, \mu, \sigma^2\}$ of the distribution using the PE approach (see [22, 106] for details on zero-inflated distributions, and see Manuscript 3 for data description and moment matching derivations). By definition, since the GS model is an estimate of the upper bound, it should yield higher likelihood values compared to the Null model. However, as depicted in Fig. 5.1 (black points), the PE approach yields an upper bound (GS model likelihood) that for many neurons is lower than the lower bound (i.e. Null model likelihood). The main reason for this effect is the sparsity of neural responses. When coupled with the small sample size, this leads to an overconfident, and thus biased, estimation of the GS model parameters, as illustrated by the two example neurons in Fig. 5.1.

5.2.2 Bayesian to the rescue

To mitigate the issue of overconfidence inherent in the PE approach, we imposed uncertainty (i.e. a prior) over the estimated parameters, and estimated the GS model in a fully Bayesian manner via the full posterior predictive distribution:

$$p_*(y_i|\mathbf{y}_{\setminus i}) = \int_{-\infty}^{\infty} \underbrace{p(y_i|\theta)}_{\text{likelihood}} \underbrace{p(\theta|\mathbf{y}_{\setminus i})}_{\text{posterior}} \, \mathrm{d}\theta \tag{5.3}$$

Note that this is a generalized formulation where the PE approach corresponds to a special case with $p(\theta|\mathbf{y}_{\setminus i}) = \delta(\theta - f(\mathbf{y}_{\setminus i}))$. While for certain choices of likelihood, with an appropriate choice of prior, the posterior predictive distribution has a closed-form solution, in general the integral is intractable and solving it requires numerical approximation. Here, we derived the posterior predictive distribution for zero-inflated distributions and show that it boils down to a one-dimensional integral over q if the posterior predictive distribution of the positive part $p(y_i|\mathbf{y}^1_{\setminus i})$ is known:

$$p(y_i|\mathbf{y}_{\setminus i}) = \begin{cases} p(y_i|\mathbf{y}^0_{\setminus i}) \cdot \int_q (1-q) \cdot p(q|\mathbf{y}_{\setminus i})\, dq & \text{if } y_i < \tau \\ p(y_i|\mathbf{y}^1_{\setminus i}) \cdot \int_q q \cdot p(q|\mathbf{y}_{\setminus i})\, dq & \text{if } y_i \geq \tau \end{cases}$$

where $\mathbf{y}^0_{\setminus i}$ and $\mathbf{y}^1_{\setminus i}$ denote the set of zero and non-zero responses in $\mathbf{y}_{\setminus i}$, respectively. For detailed derivations refer to Manuscript 3.

5.3 Results

Here we provide a more thorough comparison between the PE and the Bayesian approach by assessing their performance under different conditions, namely, different number of samples as well as different levels of signal-to-noise ratio (SNR). As a final step, we applied the complete Normalized Information Gain as an evaluation metric to evaluate neural encoding models trained on responses recorded from the mouse visual cortex.

5.3.1 Point estimate versus the Bayesian estimate

The analyses here were conducted on both simulated and real neuronal responses recorded from the mouse visual cortex (refer to Manuscript 3 for a detailed description of the data). Similar to the previous analyses, we chose the model distribution to be a zero-inflated Log-Normal distribution with parameters $\theta = \{\mu, \sigma^2, q\}$ and $\tau = \exp(-10)$.

First, we assessed which parameters profit the most from the Bayesian approach by comparing GS models where individual parameters of the distribution were either estimated via the PE approach or the Bayesian approach. This analysis is especially insightful when each parameter of the distribution corresponds to a different moment, which is the case for the Log-Normal distribution. Our results show that the **Bayesian approach yields better estimates of the higher order moments** (Fig. 5.2a). Notably, estimating the variance σ^2 via the Bayesian approach (Fig. 5.2a light blue bar) results in a significant improvement compared to when both parameters are estimated via the PE approach, while the contribution of Bayesian approach for the mean μ is much less pronounced.

Next, we compared the two approaches for different numbers of samples and show that the **Bayesian approach is more data-efficient** than the PE approach (Fig. 5.2b). That is, as we decrease the number of samples the Bayesian approach incurs a smaller decrease in the likelihood compared to the PE approach. The difference in the upper bound likelihoods of these two approaches is especially apparent for lower number of repeats, where the Bayesian approach outperforms the PE approach by a significant margin.

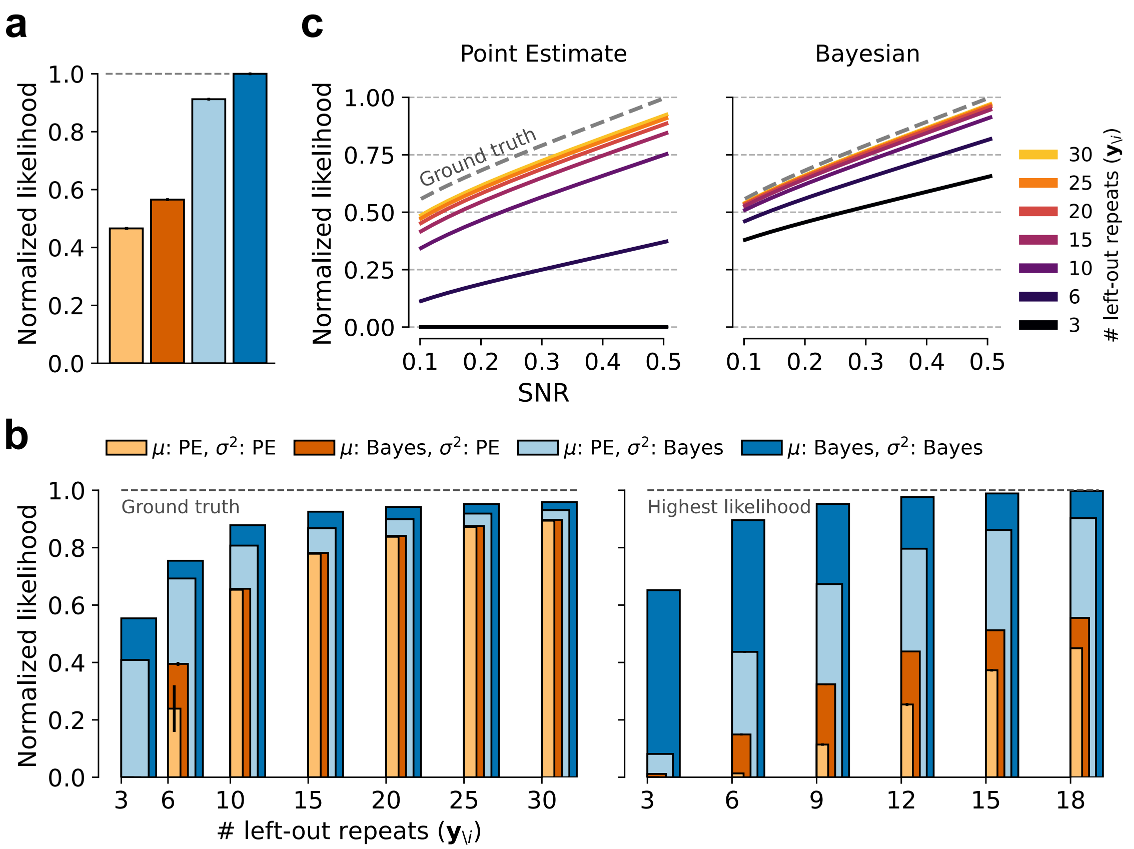


Figure 5.2: Comparison of the PE and Bayesian GS models. **a:** Different GS models where the individual parameters are either estimated via the PE or the Bayesian approach. The number of other repeats $\mathbf{y}_{\setminus i}$ is 19. Colors are the same as in **b**. **b:** Similar to **a** but for different numbers of left-out repeats $\mathbf{y}_{\setminus i}$. **Left:** Simulated data. **Right:** Neural responses. **c:** Upper bound likelihood scores for different signal-to-noise ratios and different number of left-out repeats $\mathbf{y}_{\setminus i}$. In all panels, the likelihood values are averaged over stimuli and neurons, and the error bars and shaded areas show SEM over 5 random selections of the left-out repeats. The figure is reprinted from [29].

As different datasets can exhibit different levels of noise, we also tested the two approaches on responses with varying levels of SNR. As shown in Fig. 5.2c, the **Bayesian GS is more robust to different SNRs**. Similar to other results, the difference between the two approaches is more pronounced when there are few repeats. It is worth noting that in real experiments a commonly used number of repeats is 10, which emphasizes the advantage of using our proposed method when dealing with recorded neural responses.

5.3.2 Likelihood-based evaluation of neural encoding models

Now that we have established the superiority of the Bayesian approach compared to the PE approach, we will only use the Bayesian approach when referring to the GS model. Being able to estimate a robust upper bound, we are now equipped to use the lower and upper bound estimates to evaluate neural encoding models via Normalized Information Gain (Eq. 5.1).

To this end, we trained a model on responses recorded from mouse primary visual

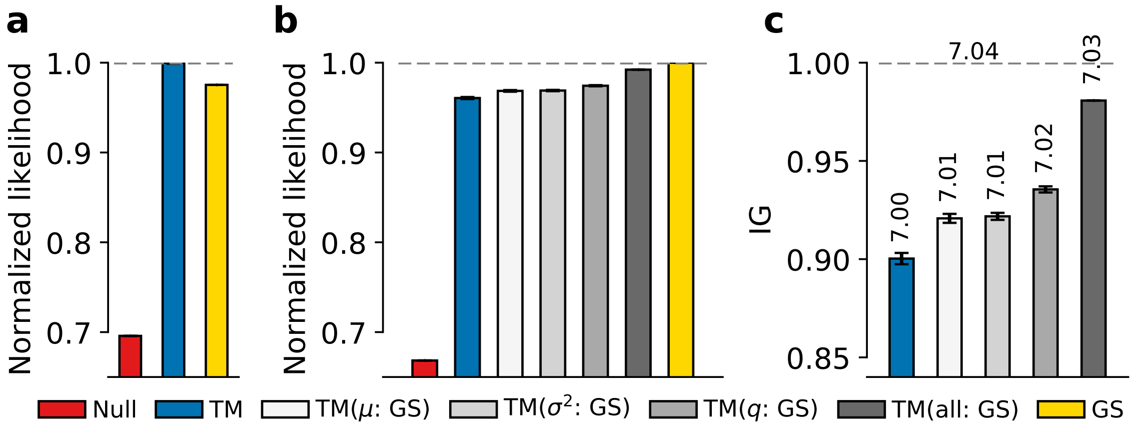


Figure 5.3: Evaluation of a zero-inflated Log-Normal neural encoding model trained on real neural responses. **a:** A sub-optimal prior yields a GS model that performs worse than the trained model (TM). **b:** An optimized prior results in a GS model that outperforms the TM. The TM estimates all parameters similarly well. Greyscale colors indicate models from the same distribution as the TM (ZIL) but with one or all parameters matching the GS model. **c:** Normalized Information Gain (NInGa) for the TM and the models (grey bars) from **b**. Values on top of the bars indicate the likelihood per image and per neuron in bits. In all panels, the likelihood values are averaged over stimuli and neurons, and the error bars and shaded areas show SEM over 5 random initializations of the TM parameters. There are no errorbars on the red, yellow, and dark grey bars since they do not involve a TM. The figure is reprinted from [29].

cortex to natural images. The dataset contained a test set with 100 images each repeated 10 times, resulting in $n-1=9$ other repeats $\mathbf{y}_{\setminus i}$, which were used to estimate the GS model parameters. Similar to previous sections we assume a zero-inflated Log-Normal (ZIL) distribution over the responses and trained the model with a negative log-likelihood objective (see Manuscript 3 for a detailed description of the data, model, and training procedure).

Computing the posterior predictive likelihood requires a prior over the parameters of the response distribution. First, we obtained the prior parameters simply by optimizing a marginal distribution over all responses irrespective of which stimulus they belong to. Since the GS model is an upper bound and has access to more information (i.e. multiple responses to the same stimulus) compared to the trained model (TM), it must yield a higher likelihood. However, we observed that the resulting GS model results in a lower performance compared to the trained model (Fig. 5.3a). An alternative approach is to optimize the prior such the likelihood of the GS model is maximized. Optimizing the prior yields a GS model that outperforms the trained model (Fig. 5.3b), as expected.

In order to investigate which parameters of the response distribution are captured well by the trained model, we conducted an analysis similar to the one shown in Fig. 5.2a. Specifically, we compared the likelihood of the trained model to cases where we matched either one or all parameters to the GS model (Fig. 5.3b, blue vs. grey bars). We observed that matching each parameter resulted in a slight improvement in the performance of the trained model. Notably, the improvement was very similar for all the parameters, implying that the **neural encoding model captures all**

parameters equally well. As expected, matching all three parameters resulted in a performance beyond matching any of the parameters individually. However, even though we matched all the parameters, the performance did not match the GS model, which can be explained by the difference in the distributional shape of the positive part: Log-Normal for ZIL vs. Log-Student-t for the GS model.

Finally, we evaluated the trained model using the Normalized Information Gain and show that the **encoding model performs at** 90% **NInGa** (Fig. 5.3c). When using NInGa, the difference between the contribution of different parameters is more pronounced and, in this case, it seems that future models can benefit from predicting the parameter q better. We also performed additional analyses to show that using NInGa enables model comparison across different datasets (see Manuscript 3).

5.4 Discussion

Advancing metrics for evaluating neural encoding models In this project, we focused on applying improved metrics for evaluating the performance of neural encoding models. With the increasing complexity of these models (e.g. using more complex distributions to capture neural responses), there is a need for metrics that quantify the performance of these predictive models on multiple aspects. Here, we argued for an interpretable likelihood-based metric: Normalized Information Gain (NInGa), a metric that puts model performance on an interpretable scale. Specifically, we focused on the challenges of obtaining lower and upper bounds for NInGa. Our Bayesian approach to estimating the upper bound provides a data-efficient and robust Generalized Sigmoid (GS) model, particularly useful in high Signal-to-Noise Ratio (SNR) scenarios. Using this robust upper bound, we evaluated current neural encoding models and found that they capture the response distribution remarkably well, achieving up to 90% NInGa. This project provides helpful practical steps to facilitate the adoption of such likelihood-based metrics in the field to evaluate future neural encoding models.

Limitations and future extensions While our results show that the current neural encoding models capture the response distribution well, the high NInGa scores could potentially be due to a sub-optimal upper bound estimator, pointing to a need for future work in refining this estimator. Additionally, our current approach assumes independence across neurons and is not directly applicable to models that account for statistical dependencies, such as the latent variable model discussed in chapter 3. However, the flexibility of the NInGa metric allows for future extensions to cover such cases, offering a pathway for more comprehensive evaluations of neural encoding models.

6 Discussion and conclusion

In recent years, neural system identification has seen great advancements in building predictive models of cortical population activity. Many of these advancements are owed to the recent developments in machine learning, especially deep learning and deep neural networks. In the past decade, these models have been utilized to obtain powerful predictive models of the sensory neurons [21], characterize the structural aspects of the visual system [11], and run experiments *in silico* that are infeasible to conduct with the biological system, but whose results can be evaluated *in vivo*. Following this research direction, in this thesis, I discussed three projects that show how DNN-based models of visual sensory neurons can be used to generate insights about the functional and structural properties of these neurons, as well as how internal processes and behavioral variables affect the responses of these neurons beyond the sensory input. Additionally, I discussed an important aspect of developing these models which is concerned with finding better ways to evaluate such models based on how well they capture the full distribution of neural responses as opposed to only focusing on the mean.

While the primary goal of the visual sensory neurons is to encode visual stimuli, many studies have shown that other factors such as behavioral variables or cognitive processes (e.g. attention) can affect how these neurons respond to the sensory input [13, 49–51, 56, 97]. Based on these experimental observations, in the first project we aimed at developing a model that captures the variations is neural responses beyond those that are induced by the visual stimulus. To this end, we combined state-of-the-art DNN-based models with a simple, yet flexible, flow-based factor analysis model to account for two major sources of variability in neural responses: stimulus-driven and stimulus-conditioned. We showed that the resulting model not only captures the responses of visual sensory neurons well but also, through learning the statistical dependencies across these neurons, yields latent states that exhibit meaningful relations to behavioral variables as well as anatomical and functional properties of visual sensory neurons. Such models that can capture multiple aspects of cortical population responses can lead to deeper scientific insights and a better 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.

One of the main goals of studying and understanding the visual system is to build machines and algorithms (i.e. computer vision) that mimic the useful properties of their biological counterpart. A key characteristic of visual perception is its robustness and its ability to generalize despite significant variations in the environment. Such ability is believed to require neural mechanisms that exhibit selectivity towards certain stimulus features while maintaining invariance to others, such as spatial location or rotation. Therefore, in order to gain a better understanding of the computational mechanisms that underlie the robustness and generalizability of the biological visual systems, it is important to identify the features that strongly drive neural activity and uncover the specific transformations of these features that do not alter neural responses, known as single cell invariances. Focusing on invariances, in the second project we set out to develop a model that learns a manifold in the stimulus space such that images along this manifold equally and maximally excite a target neuron. While in the past identification of invariances has commonly been a hypothesis-driven process relying on the presentation of carefully selected stimuli, our method is designed to be data-driven, which allows us to identify unexpected and novel invariances. In addition, it uses a generative approach which allows us to generate and experiment with new images that lie on the same invariance manifold. We believe that such an approach not only can yield important insights into the coding properties of visual sensory neurons but also provides ideas that can inspire more robust computer vision algorithms.

Finally, the utility of DNN-based models as *digital twins* of the biological visual system depends on how well they capture and represent neural responses recorded from the brain. Therefore, prior to using these models as tools for generating insights about the brain, it is crucial to ensure that they represent the functional properties of their biological counterpart well. This requires the development of evaluation metrics that take into account as many aspects of the neural responses as possible. As I discussed in chapter 5 a good candidate for this purpose is Normalized Information Gain (NInGa) which is an interpretable likelihood-based evaluation metric. In the third project, we focused on the theoretical and practical aspects of using NInGa, in a robust and data-efficient manner, for evaluating predictive models of visual sensory neurons.

Overall, this thesis focused on three important aspects of neural system identification: 1) the development of models that consider multiple factors influencing neural responses, 2) the demonstration of how these models can generate insights into the functional and structural properties of visual sensory neurons, and 3) the development of robust metrics for better model evaluation. These contributions extend beyond the immediate field, offering valuable insights for both neuroscience and machine learning. By refining models and metrics, we take meaningful steps toward a better understanding of the complex computations carried out by the brain, as well as how these can be approximated in computational frameworks. These advancements have the potential to inform a range of applications, from enhancing neurological diagnostics to advancing intelligent systems. My hope is that the methods and insights presented in this thesis will serve as a useful resource for future research at the intersection of neuroscience and machine learning.

References

[1] Bing Wei, Yudi Zhao, Kuangrong Hao, and Lei Gao. Visual sensation and perception computational models for deep learning: State of the art, challenges and prospects. *arXiv preprint arXiv:2109.03391*, 2021.

[2] Samuel Dodge and Lina Karam. A study and comparison of human and deep learning recognition performance under visual distortions. In *2017 26th international conference on computer communication and networks (ICCCN)*, pages 1–7. IEEE, 2017.

[3] Vishal Saxena, Xinyu Wu, Ira Srivastava, and Kehan Zhu. Towards neuromorphic learning machines using emerging memory devices with brain-like energy efficiency. *Journal of Low Power Electronics and Applications*, 8(4):34, 2018.

[4] Stuart Trenholm and Arjun Krishnaswamy. An annotated journey through modern visual neuroscience. *Journal of Neuroscience*, 40(1):44–53, 2020.

[5] David H Hubel and Torsten N Wiesel. Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148(3):574, 1959.

[6] David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106, 1962.

[7] Horace B Barlow et al. Possible principles underlying the transformation of sensory messages. *Sensory communication*, 1(01):217–233, 1961.

[8] Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.

[9] 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.

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

[11] Daniel LK Yamins, Ha Hong, Charles F Cadieu, Ethan A Solomon, Darren Seibert, and James J DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the national academy of sciences*, 111(23):8619–8624, 2014.

[12] Carsen Stringer, Marius Pachitariu, Nicholas Steinmetz, Matteo Carandini, and Kenneth D Harris. High-dimensional geometry of population responses in visual cortex. *Nature*, 571(7765):361–365, 2019.

[13] 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.

[14] David J Heeger. Normalization of cell responses in cat striate cortex. *Visual neuroscience*, 9(2):181–197, 1992.

[15] Max F Burg, Santiago A Cadena, George H Denfield, Edgar Y Walker, Andreas S Tolias, Matthias Bethge, and Alexander S Ecker. Learning divisive normalization in primary visual cortex. *PLOS Computational Biology*, 17(6):e1009028, 2021.

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

[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] Alexander S Ecker, Fabian H Sinz, Emmanouil Froudarakis, Paul G Fahey, Santiago A Cadena, Edgar Y Walker, Erick Cobos, Jacob Reimer, Andreas S Tolias, and Matthias Bethge. A rotation-equivariant convolutional neural network model of primary visual cortex. *arXiv preprint arXiv:1809.10504*, 2018.

[19] 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.

[20] Yimeng Zhang, Tai Sing Lee, Ming Li, Fang Liu, and Shiming Tang. Convolutional neural network models of v1 responses to complex patterns. *Journal of computational neuroscience*, 46(1):33–54, 2019.

[21] 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. In *International Conference on Learning Representations (ICLR)*, 2021.

[22] **Mohammad Bashiri***, Edgar Walker*, Konstantin-Klemens Lurz, Akshay Jagadish, Taliah Muhammad, Zhiwei Ding, Zhuokun Ding, Andreas Tolias, and Fabian Sinz. A flow-based latent state generative model of neural population responses to natural images. *Advances in Neural Information Processing Systems*, 34:15801–15815, 2021.

[23] 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.

[24] Carlos R Ponce, Will Xiao, Peter F Schade, Till S Hartmann, Gabriel Kreiman, and Margaret S Livingstone. Evolving images for visual neurons using a deep generative network reveals coding principles and neuronal preferences. *Cell*, 177 (4):999–1009, 2019.

[25] Liz Allen, Jo Scott, Amy Brand, Marjorie Hlava, and Micah Altman. Publishing: Credit where credit is due. *Nature*, 508(7496):312–313, 2014.

[26] Liz Allen, Alison O'Connell, and Veronique Kiermer. How can we ensure visibility and diversity in research contributions? how the contributor role taxonomy (credit) is helping the shift from authorship to contributorship. *Learned Publishing*, 32(1):71–74, 2019.

[27] Alex O Holcombe. Contributorship, not authorship: Use credit to indicate who did what. *Publications*, 7(3):48, 2019.

[28] Luca Baroni*, **Mohammad Bashiri***, Konstantin F Willeke, Ján Antolík, and Fabian H Sinz. Learning invariance manifolds of visual sensory neurons. In *NeurIPS Workshop on Symmetry and Geometry in Neural Representations*, pages 301–326. PMLR, 2023.

[29] Konstantin-Klemens Lurz*, **Mohammad Bashiri***, Edgar Y. Walker, and Fabian H Sinz. Bayesian oracle for bounding information gain in neural encoding models. In *International Conference on Learning Representations (ICLR)*, 2023.

[30] Konstantin F Willeke*, Paul G Fahey*, **Mohammad Bashiri**, Laura Pede, Max F Burg, Christoph Blessing, Santiago A Cadena, Zhiwei Ding, Konstantin-Klemens Lurz, Kayla Ponder, et al. The sensorium competition on predicting large-scale mouse primary visual cortex activity. *arXiv preprint arXiv:2206.08666*, 2022.

[31] Paweł A Pierzchlewicz, R James Cotton, **Mohammad Bashiri**, and Fabian H Sinz. Multi-hypothesis 3d human pose estimation metrics favor miscalibrated distributions. *arXiv preprint arXiv:2210.11179*, 2022.

[32] Paweł A Pierzchlewicz, **Mohammad Bashiri**, R James Cotton, and Fabian H Sinz. Optimizing mpjpe promotes miscalibration in multi-hypothesis human pose lifting. In *International Conference on Learning Representations (ICLR) as a Tiny Paper*, 2023.

[33] Jiakun Fu, Pawel A Pierzchlewicz, Konstantin F Willeke, **Mohammad Bashiri**, Taliah Muhammad, George H Denfield, Fabian Hubert Sinz, and Andreas S Tolias. Heterogeneous orientation tuning across sub-regions of receptive fields of v1 neurons in mice. *Under Review.*

[34] **Mohammad Bashiri**. Learning gabor filters via gradient descent, 2020. URL `https://github.com/mohammadbashiri/fitgabor`.

[35] Polina Turishcheva, Paul G Fahey, Laura Hansel, Rachel Froebe, Kayla Ponder, Michaela Vystrčilová, Konstantin F Willeke, **Mohammad Bashiri**, Eric Wang, Zhiwei Ding, Andreas S. Tolias, Fabian Sinz, and Alexander S. Ecker. The dynamic sensorium competition for predicting large-scale mouse visual cortex activity from videos. *arXiv preprint arXiv:2305.19654*, 2023.

[36] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521 (7553):436–444, 2015.

[37] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[38] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011.

[39] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980.

[40] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10): 1995, 1995.

[41] Simon J.D. Prince. *Understanding Deep Learning*. MIT Press, 2023.

[42] Jonathan W Pillow, Liam Paninski, Valerie J Uzzell, Eero P Simoncelli, and EJ Chichilnisky. Prediction and decoding of retinal ganglion cell responses with a probabilistic spiking model. *Journal of Neuroscience*, 25(47):11003–11013, 2005.

[43] John Ashworth Nelder and Robert WM Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, 135(3):370–384, 1972.

[44] Edward H Adelson and James R Bergen. Spatiotemporal energy models for the perception of motion. *Josa a*, 2(2):284–299, 1985.

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

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

[47] Santiago A. Cadena, Marissa A. Weis, Leon A. Gatys, Matthias Bethge, and Alexander S. Ecker. Diverse feature visualizations reveal invariances in early layers of deep neural networks, 2018. URL `https://arxiv.org/abs/1807.10589`.

[48] Zhiwei Ding, Dat T Tran, Kayla Ponder, Erick Cobos, Zhuokun Ding, Paul G Fahey, Eric Wang, Taliah Muhammad, Jiakun Fu, Santiago A Cadena, et al. Bipartite invariance in mouse primary visual cortex. *bioRxiv*, 2023.

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

[50] 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.

[51] 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.

[52] Pietro Berkes, Frank Wood, and Jonathan Pillow. Characterizing neural dependencies with copula models. *Advances in neural information processing systems*, 21, 2008.

[53] Byron M Yu, John P Cunningham, Gopal Santhanam, Stephen Ryu, Krishna V Shenoy, and Maneesh Sahani. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. *Advances in neural information processing systems*, 21, 2008.

[54] Jakob H Macke, Lars Buesing, John P Cunningham, Byron M Yu, Krishna V Shenoy, and Maneesh Sahani. Empirical models of spiking in neural populations. *Advances in neural information processing systems*, 24, 2011.

[55] Evan W Archer, Urs Koster, Jonathan W Pillow, and Jakob H Macke. Low-dimensional models of neural population activity in sensory cortical circuits. *Advances in neural information processing systems*, 27, 2014.

[56] 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.

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

[58] 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 computation*, 29(1):50–93, 2017.

[59] Yuan Zhao and Il Memming Park. Variational latent gaussian process for recovering single-trial dynamics from population spike trains. *Neural computation*, 29(5):1293–1316, 2017.

[60] Anqi Wu, Nicholas A Roy, Stephen Keeley, and Jonathan W Pillow. Gaussian process based nonlinear latent structure discovery in multivariate spike train data. *Advances in neural information processing systems*, 30, 2017.

[61] 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.

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

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

[64] Stephen Keeley, Mikio Aoi, Yiyi Yu, Spencer Smith, and Jonathan W Pillow. Identifying signal and noise structure in neural population activity with gaussian process factor models. *Advances in neural information processing systems*, 33: 13795–13805, 2020.

[65] 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.

[66] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

[67] Esteban G Tabak and Eric Vanden-Eijnden. Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, 8(1):217–233, 2010.

[68] Esteban G Tabak and Cristina V Turner. A family of nonparametric density estimation algorithms. *Communications on Pure and Applied Mathematics*, 66 (2):145–164, 2013.

[69] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[70] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[71] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[72] Jakub M Tomczak and Max Welling. Improving variational auto-encoders using convex combination linear inverse autoregressive flow. *arXiv preprint arXiv:1706.02326*, 2017.

[73] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

[74] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. *Advances in neural information processing systems*, 32, 2019.

[75] Emilien Dupont, Hyunjik Kim, SM Ali Eslami, Danilo Jimenez Rezende, and Dan Rosenbaum. From data to functa: Your data point is a function and you can treat it like one. In *International Conference on Machine Learning*, pages 5694–5725. PMLR, 2022.

[76] Kenneth O Stanley. Compositional pattern producing networks: A novel abstraction of development. *Genetic programming and evolvable machines*, 8(2): 131–162, 2007.

[77] David Ha. Generating large images from latent vectors. *blog.otoro.net*, 2016. URL `https://blog.otoro.net/2016/04/01/generating-large-images-from-latent-vectors/`.

[78] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019.

[79] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019.

[80] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[81] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6498–6508, 2021.

[82] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33:7462–7473, 2020.

[83] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[84] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5799–5809, 2021.

[85] Ishit Mehta, Michaël Gharbi, Connelly Barnes, Eli Shechtman, Ravi Ramamoorthi, and Manmohan Chandraker. Modulated periodic activations for generalizable local functional representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14214–14223, 2021.

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

[87] 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.

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

[89] 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.

[90] 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.

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

[92] 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.

[93] 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.

[94] 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.

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

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

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

[98] 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.

[99] 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.

[100] 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.

[101] 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.

[102] 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.

[103] 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.

[104] 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.

[105] 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.

[106] 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.

[107] 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.

[108] 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.

[109] 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.

[110] 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.

[111] 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.

[112] 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.

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

[114] Jean-Rene Duhamel, Frank Bremmer, Suliann Ben Hamed, and Werner Graf. Spatial invariance of visual receptive fields in parietal cortex neurons. *Nature*, 389(6653):845–848, 1997.

[115] Reto Wyss, Peter König, and Paul FMJ Verschure. Invariant representations of visual patterns in a temporal population code. *Proceedings of the National Academy of Sciences*, 100(1):324–329, 2003.

[116] Matteo Carandini. Melting the iceberg: contrast invariance in visual cortex. *Neuron*, 54(1):11–13, 2007.

[117] Alexander Mordvintsev, Nicola Pezzotti, Ludwig Schubert, and Chris Olah. Differentiable image parameterizations. *Distill*, 2018. doi: 10.23915/distill.00012. https://distill.pub/2018/differentiable-parameterizations.

[118] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005.

[119] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020.

[120] Ruslan Salakhutdinov and Geoff Hinton. Learning a nonlinear embedding by preserving class neighbourhood structure. In *Artificial Intelligence and Statistics*, pages 412–419. PMLR, 2007.

[121] Nicholas Frosst, Nicolas Papernot, and Geoffrey Hinton. Analyzing and improving representations with the soft nearest neighbor loss. In *International conference on machine learning*, pages 2012–2020. PMLR, 2019.

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

[123] Wei Ji Ma, Jeffrey M Beck, Peter E Latham, and Alexandre Pouget. Bayesian inference with probabilistic population codes. *Nature neuroscience*, 9(11):1432–1438, 2006.

[124] József Fiser, Pietro Berkes, Gergő Orbán, and Máté Lengyel. Statistically optimal perception and learning: from behavior to neural representations. *Trends in cognitive sciences*, 14(3):119–130, 2010.

[125] Pietro Berkes, Gergő Orbán, Máté Lengyel, and József Fiser. Spontaneous cortical activity reveals hallmarks of an optimal internal model of the environment. *Science*, 331(6013):83–87, 2011.

[126] Richard D Lange, Ankani Chattoraj, Jeffrey M Beck, Jacob L Yates, and Ralf M Haefner. A confirmation bias in perceptual decision-making due to hierarchical approximate inference. *PLoS Computational Biology*, 17(11):e1009517, 2021.

[127] Adrian G Bondy, Ralf M Haefner, and Bruce G Cumming. Feedback determines the structure of correlated variability in primary visual cortex. *Nature neuroscience*, 21(4):598–606, 2018.

[128] Matthias Kümmerer, Thomas SA Wallis, and Matthias Bethge. Information-theoretic model comparison unifies saliency metrics. *Proceedings of the National Academy of Sciences*, 112(52):16054–16059, 2015.

Appendix

This chapter contains the publications discussed in chapters 3–5.

Manuscript 1

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.

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

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, 31, 34], or employ a very simple model of stimulus-driven variations [16, 25, 26, 32].

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.

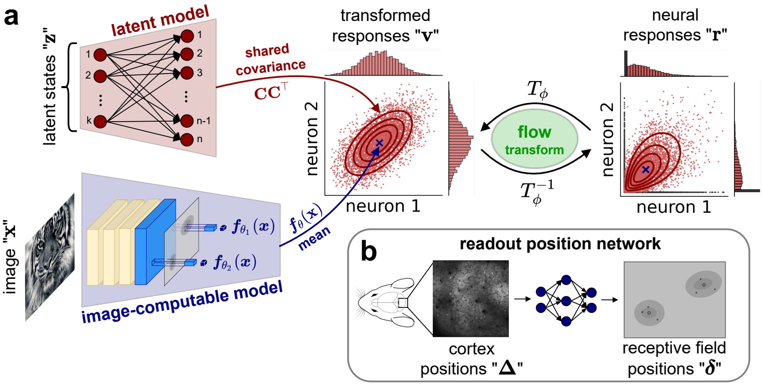


Figure 1: Flow-based Factor Analysis model. **a:** Schematic of the flow-based model relating all relevant variables in the study. **b:** Schematic of the sub-network used by the image-computable model to map cortical positions into receptive field positions. Refer to section 2 Methods for the details.

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

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

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 $\mathcal{N}(\mathbf{v}; \mathbf{f}_\theta(\mathbf{x}), \mathbf{CC}^\top + \Psi)$ (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 $\mathbf{v} = \mathbf{f}_\theta(\mathbf{x}) + \mathbf{Cz} + \varepsilon$ 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 $|\text{det}\nabla_\mathbf{r} T_\phi(\mathbf{r})|$ 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 $\text{det}\nabla_\mathbf{r} T_\phi(\mathbf{r}) = \prod_{i=1}^n \frac{\partial T_{\phi_i}}{\partial r_i}$, 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:

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

where $q_i(\mathbf{x})$ is the probability of the response being above the threshold ρ 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_ϕ = affine $\circ$ exp $\circ$affine $\circ$ ELU $\circ$ affine $\circ$ ELU $\circ$ affine $\circ$ log $\circ$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 indexed by i, 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).

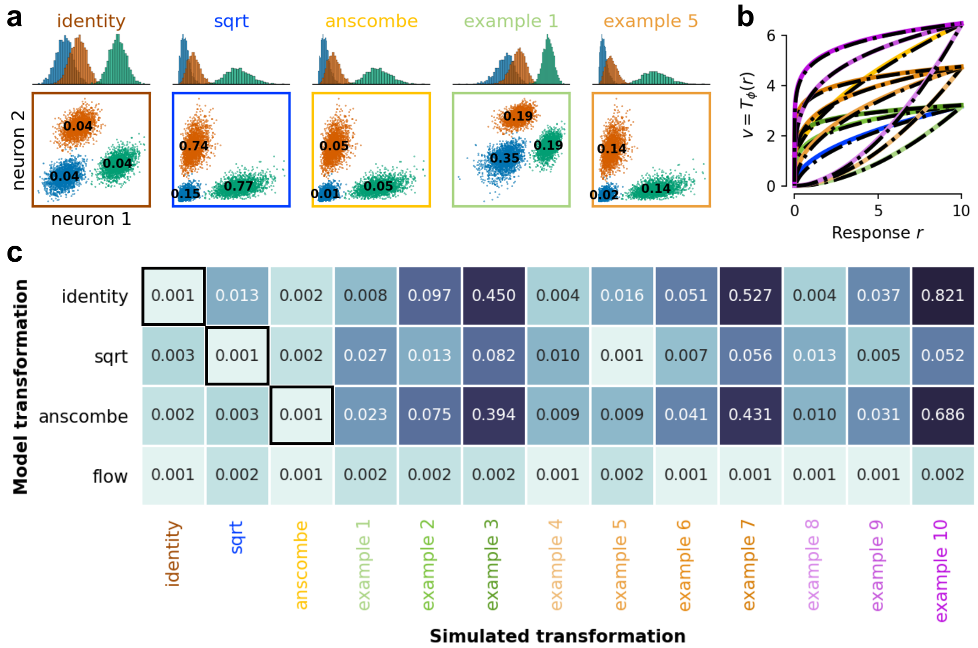


Figure 2: FlowFA model recovers the underlying transformation. **a:** Simulated responses for 2 neurons under various transformations. Across all transformations, *transformed* responses were sampled from Gaussian distributions with differing means (indicated by the color of the samples) but identical 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).

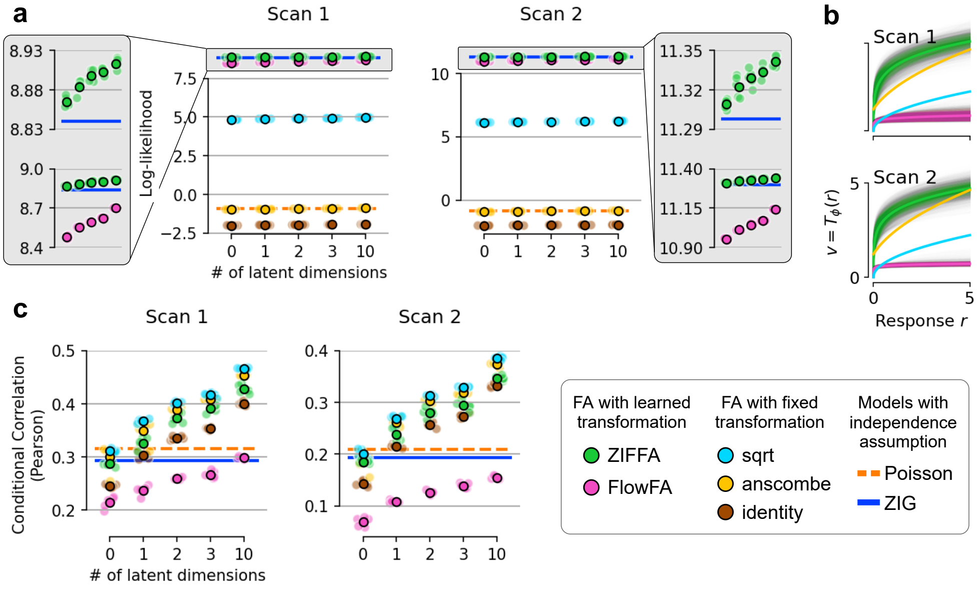


Figure 3: Comparison of models trained on the mouse visual cortical population responses to natural images. **a**: log-likelihood computed for models trained on scan 1 (left panel) and scan 2 (right panel). Values for both individual (lighter shade) and average (darker shade) performance of a model trained 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

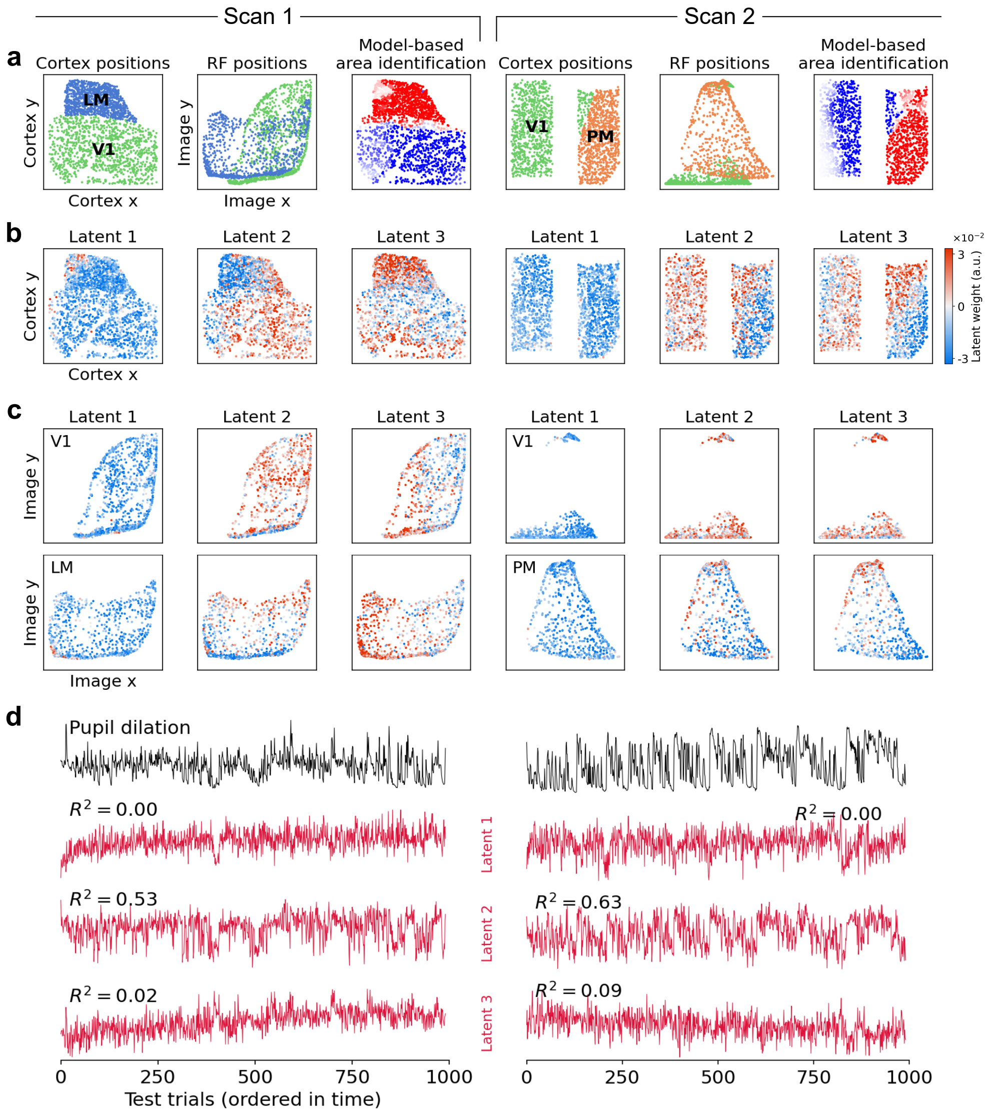


Figure 4: Analysis of the ZIFFA model with 3-dimensional latent state ($k = 3$). **a**: Model-based area identification from responses of visual sensory neurons to natural images. Left panel (Cortex positions): cortical position of the recorded neurons color-coded by experimentally identified areas (green: 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.

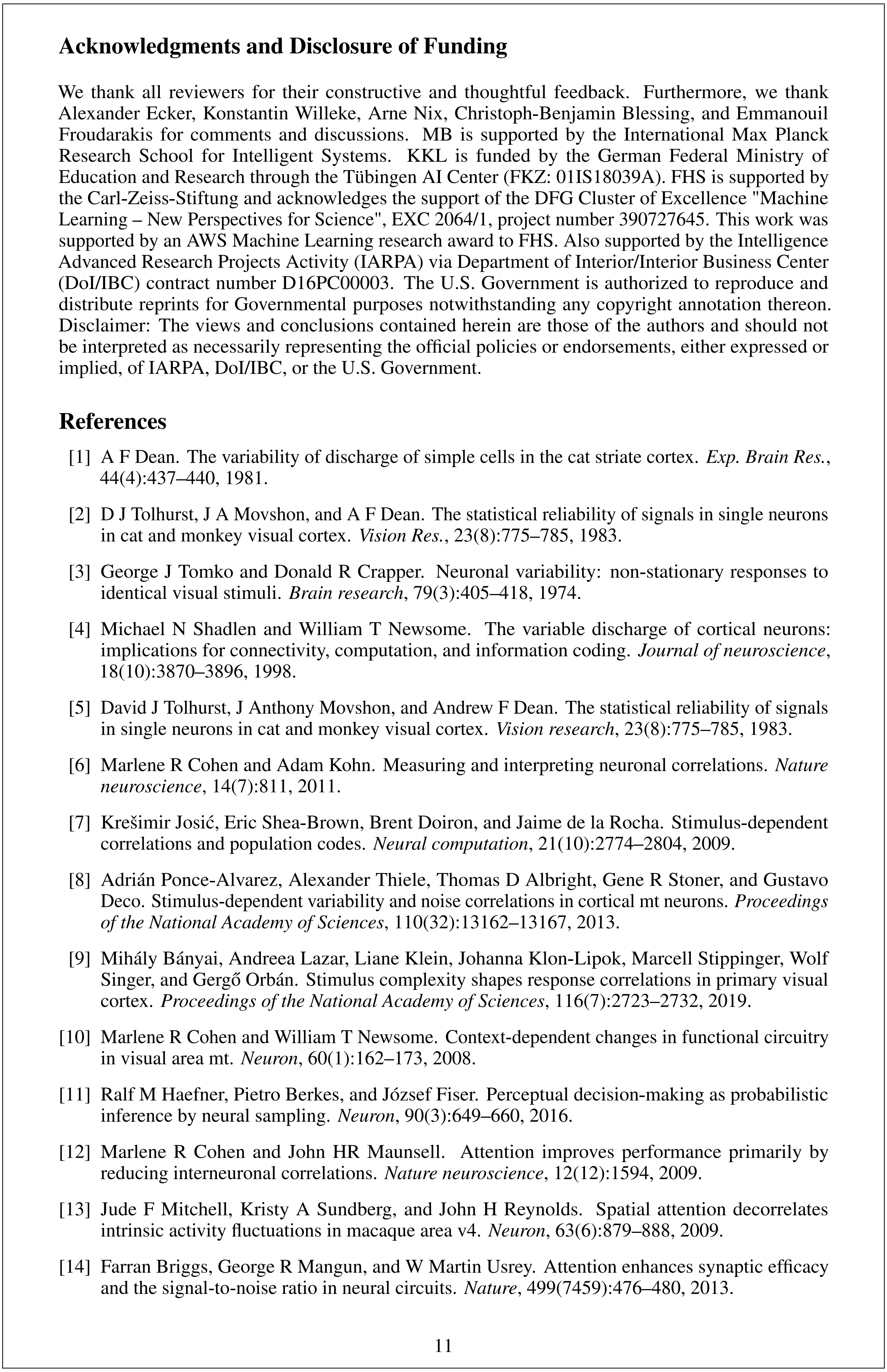
Modeling zero-inflated response distribution Flow models use diffeomorphisms to map one distribution into another. However, diffeomorphisms cannot transform a single peak at 0—typically observed in neural responses recorded via Calcium imaging—into a smooth distribution such as Gaussian used in our model. 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 Froudarakis 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. Also supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DoI/IBC) contract number D16PC00003. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/IBC, or the U.S. Government.

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:

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

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

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

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 $\mathbb{E}[r_i|\mathbf{x}, \mathbf{r}_{\setminus i}] = \frac{1}{N}\sum_j^N T_{\phi,i}^{-1}(\mathbf{s}_i^{(j)})$ 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:

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

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

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 $C \in \mathbb{R}^{100 \times 4}$, 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

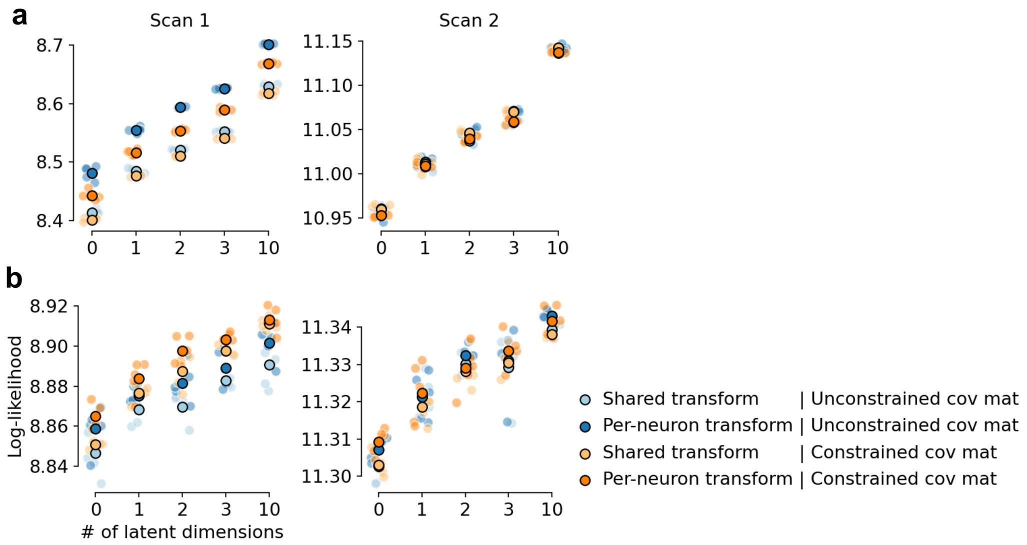


Figure S1: Comparison of flow-based models with different model configurations. These configurations include: 1) using a shared vs neuron-specific flow transformation, and 2) unconstrained vs constrained covariance matrix of the FA. The transformation T_ϕ could be defined such that a single flow transformation 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.

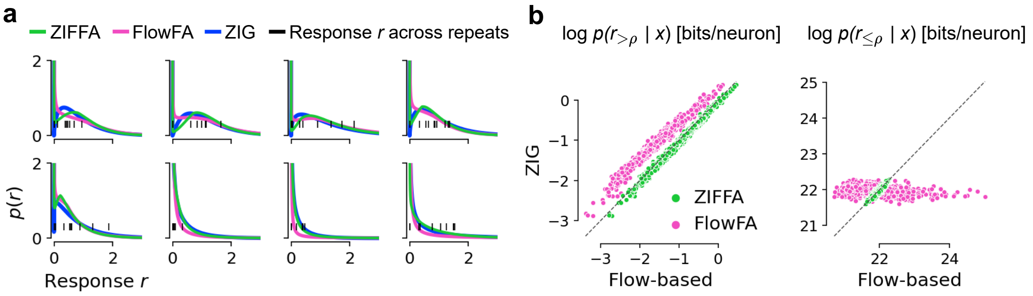


Figure S2: Comparison of the learned density by the ZIFFA, FlowFA, and ZIG models. **a**: Example marginal distribution of responses of 8 sample neurons to the repeated presentations of an image from the test set and the corresponding fits of ZIFFA, FlowFA, and ZIG. While all three models peak at zero, 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).

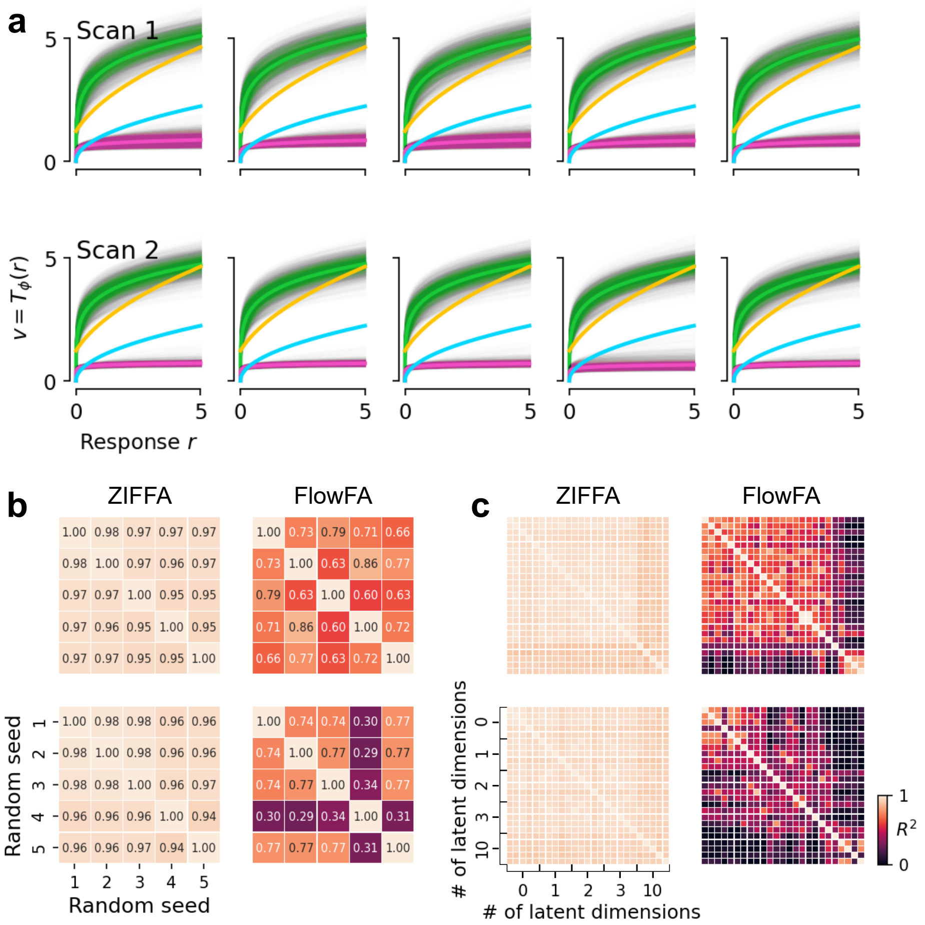


Figure S3: Consistency of the learned transformation across models initialized and trained with different random seeds, and across different number of latent dimensions. **a**: The learned flow transformation for both ZIFFA (green) and FlowFA (pink) models with 0-dimensional latent. Square-root (blue) 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.

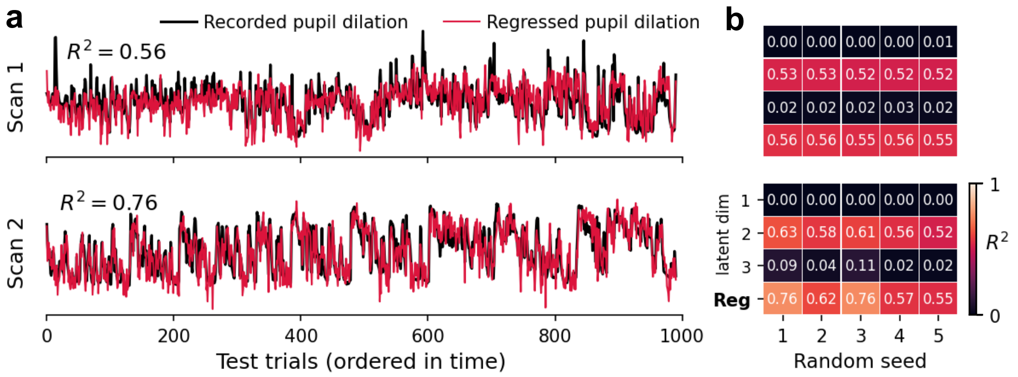


Figure S4: Correlation and regression analysis between inferred latent states and the pupil dilation. **a**: The regressed pupil dilation vs the recorded pupil dilation for the same model as in Fig. 4. **b**: First three rows: The R^2 values between orthonormalized latent states and pupil dilation across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

Manuscript 2

Learning Invariance Manifolds of Visual Sensory Neurons

Luca Baroni[†] BARONI@KSVI.MFF.CUNI.CZ
Faculty of Mathematics and Physics, Charles University, Prague, Czechia

Mohammad Bashiri[†] MOHAMMAD.BASHIRI@UNI-TUEBINGEN.DE
Institute for Bioinformatics and Medical Informatics, University of Tübingen, Germany

Konstantin F. Willeke KONSTANTIN-FRIEDRICH.WILLEKE@UNI-TUEBINGEN.DE
Institute for Bioinformatics and Medical Informatics, University of Tübingen, Germany

Ján Antolík ANTOLIKJAN@GMAIL.COM
Faculty of Mathematics and Physics, Charles University, Prague, Czechia

Fabian H. Sinz SINZ@CS.UNI-GOETTINGEN.DE
Campus Institute Data Science, University Göttingen, Germany
Institute for Bioinformatics and Medical Informatics, University of Tübingen, Germany

[†] *denotes equal contribution*

Editors: Sophia Sanborn, Christian Shewmake, Simone Azeglio, Arianna Di Bernardo, Nina Miolane

Abstract

Robust object recognition is thought to rely on neural mechanisms that are selective to complex stimulus features while being invariant to others (e.g., spatial location or orientation). To better understand biological vision, it is thus crucial to characterize which features neurons in different visual areas are selective or invariant to. In the past, invariances have commonly been identified by presenting carefully selected hypothesis-driven stimuli which rely on the intuition of the researcher. One example is the discovery of phase invariance in V1 complex cells. However, to identify novel invariances, a data-driven approach is more desirable. Here, we present a method that, combined with a predictive model of neural responses, learns a manifold in the stimulus space along which a target neuron's response is invariant. Our approach is fully data-driven, allowing the discovery of novel neural invariances, and enables scientists to generate and experiment with novel stimuli along the invariance manifold. We test our method on Gabor-based neuron models as well as on a neural network fitted on macaque V1 responses and show that 1) it successfully identifies neural invariances, and 2) disentangles invariant directions in the stimulus space [*].

Keywords: neural invariances, invariance manifold, MEI, disentanglement, contrastive learning, visual cortex, CPPN

1. Introduction

Visual sensory areas enable animals to identify objects robustly under different viewing conditions and contexts. Such ability is thought to require neural mechanisms that are selective to complex stimulus features but invariant to others (e.g., spatial location or rotation). To better understand biological vision, it is thus crucial to characterize which features strongly drive neural activity and identify which transformations of such features

[*]. Code is available at https://github.com/sinzlab/cppn_for_invariances.

leave neural responses unchanged – i.e. single cell invariances. In the past, identification of invariances in visual sensory systems have commonly been a hypothesis-driven process relying on presentation of carefully selected stimuli. One example of this is the discovery of phase invariance in complex cells of primary visual cortex (Hubel and Wiesel, 1962). However, such an approach heavily relies on the intuition of the experimenter or serendipity. Since the dimensionality of images is enormous and experimental time is limited, this approach quickly becomes infeasible when encoding of visual information becomes more complex in higher areas.

In recent years, artificial neural networks trained on large datasets of neural responses to natural images have proven to be powerful predictive models of neural responses (Yamins et al., 2014; Kriegeskorte, 2015; Antolík et al., 2016; Yamins and DiCarlo, 2016; Klindt et al., 2017; Cadena et al., 2019; Kubilius et al., 2019; Sinz et al., 2018; Lurz et al., 2021; Zhuang et al., 2021). An alternative approach might thus be to systematically explore the invariance space of visual sensory neurons via optimization using these predictive models. A large body of research in the field of interpretable machine learning has focused on feature visualization, a set of techniques to identify which inputs highly activate the network units or layers (Olah et al., 2017). These techniques have already been successfully used to find single (Walker et al., 2019; Bashivan et al., 2019; Ponce et al., 2019) or multiple (Cadena et al., 2018; Ding et al., 2022) maximally exciting stimuli for visual sensory neurons. However, all current methods predict only a discrete set of stimuli from the invariance manifold. Considering the high dimensionality of images, understanding how such stimuli are connected in the image space can be non-trivial, especially when neurons are invariant to multiple transformations, as it is expected to be more and more the case along the visual hierarchy.

Here, we present a systematic data-driven approach based on implicit image representations and contrastive learning, that allows the identification and parameterization of the manifold of highly activating stimuli. We refer to this manifold as MEI invariance manifold (or just invariance manifold for simplicity). We first tested our method on simple Gabor-based toy models that exhibit multiple invariances and different invariant manifold topologies. We found that our method correctly identifies and disentangles different invariance directions. We then validated our method on selected macaque V1 neurons where it identifies an almost exact phase invariance. Taken together, our results show that our approach can capture invariance manifolds in a meaningful way and can be potentially used to discover novel invariances in visual sensory neurons.

2. Related work

Most Exciting Image (MEI) via pixel optimization Artificial neural networks have been recently used to synthesize images that maximize the response of a given neuron in the visual system of mice and monkeys (Walker et al., 2019; Bashivan et al., 2019; Ponce et al., 2019). Such MEIs were commonly identified via direct optimization of pixel values. This is a well established technique in the field of interpretable machine learning for inspecting the units and their function in artificial neural networks (Erhan et al., 2009; Olah et al., 2017). Importantly, Walker et al. (2019); Bashivan et al. (2019); Ponce et al. (2019) demonstrated that these MEIs indeed activate biological neurons stronger than control stimuli, such as Gabors, in most cases. These results thus demonstrate the utility of these models as digital twins of the biological brain, allowing neuroscientists to conduct analyses *in-silico* that are infeasible to perform on the biological system, but whose predictions can be verified *in-vivo*.

Diverse feature visualization Previous works have mostly focused on identifying a single MEI for a single (Walker et al., 2019) or a population of neurons (Bashivan et al., 2019; Ponce et al., 2019). However, it is not clear whether there exist only a single MEI or rather a manifold of maximally exciting images. To inspect the presence of such invariances, Cadena et al. (2018) expanded on the same technique, optimizing for multiple images (diverse MEIs) while enforcing diversity with an additional objective. Such an approach allows the identification of multiple distant points in the manifold of maximally exciting images. Given that the space of images is very high dimensional, the question remains how to connect such points to construct an invariance manifold. For instance, different phases of an optimal Gabor stimulus of a complex cell cannot be connected by straight lines in image space. The mid point between two 180 degree shifted Gabors would be a flat image, which is certainly not strongly driving a complex cell. Instead, the maximally exciting curve between the two Gabors forms a circle in high dimensions.

Differentiable Image Parameterization Recent developments in feature visualization techniques show that smooth, semantically meaningful, transition between images can be obtained via differentiable parameterization methods (Mordvintsev et al., 2018; Ha, 2016; Mildenhall et al., 2021). Such methods are, however, yet to be applied to characterize the invariances of biological neurons.

3. Methods

In contrast to previous approaches to identify MEIs, i.e. directly optimizing pixel values, we use Compositional Pattern Producing Networks (CPPNs) to optimize a reparameterized version of the image. CPPNs (Stanley, 2007) are artificial neural networks mapping pixel positions (x, y) to pixel RGB (or grayscale) values. They have recently gained a lot of attention in the computer vision community as implicit representations of shapes and radiance fields (Ha, 2016; Mescheder et al., 2019; Mildenhall et al., 2021). A vanilla CPPN is a differentiable implicit representations of a single image in arbitrary resolution.

3.1. CPPN as an implicit representation of the invariance manifold

Our goal is to use a single CPPN as an implicit representation of not a single image but the whole manifold of images that equally maximize the activation of a target neuron. For this, a single CPPN needs to produce a variety of images. This can be achieved by extending the inputs of the CPPN to include an additional input variable z belonging to a low-dimensional bounded latent space. This allows the CPPN to output different images while being fed the same set of pixel positions (Ha, 2016). In the context of learning the invariance manifold, different values of z should result in different images that maximally excite a target neuron. If this is achieved, z captures a latent parameterization of the MEI invariance manifold and a specific value of it represents a single point on the manifold. We implemented the CPPN as a simple fully-connected neural network of 8 hidden layers each with 15 units. Each hidden layer was followed by a batch normalization and leaky ReLU nonlinearity. As we considered only grayscale images, the output layer of the CPPN contains a single unit with a

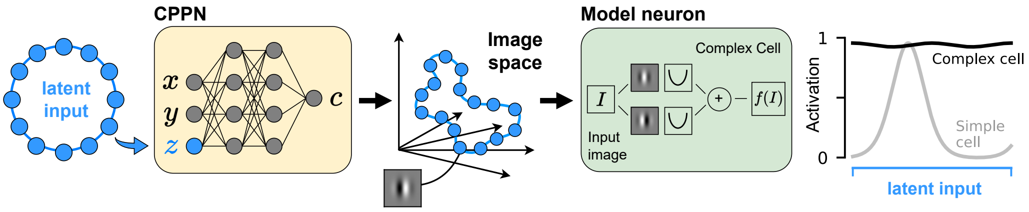


Figure 1: Our method uses a CPPN to map a simple low-dimensional latent space onto a complex high-dimensional manifold in the image space. Images from this manifold result in diverse but maximally exciting stimuli for a model neuron. Here we show a schematic of this method applied on a complex cell. Corresponding activations for a simple cell are also added as reference.

tanh nonlinearity, resulting in a 1D output with values between -1 and 1. To allow control over the characteristic spatial frequency of the patterns generated via CPPN, instead of directly using pixel positions as inputs to the CPPN, we used positional encoding of the pixel positions via random Fourier mapping (Tancik et al., 2020; Mildenhall et al., 2021)[1].

As the topology of the MEI invariance manifold might vary from neuron to neuron, we explored different dimensionalities and boundary conditions for the latent space ("latent input" in Fig. 1). In particular, we considered 1D and 2D latent spaces, non-periodic (corresponding to a line or sheet topology) or periodic (corresponding to circle or torus topology).

3.2. Constrastive objective for image diversification

After the CPPN maps the low-dimensional latent to a manifold in image space, these images are fed to a predictive model of neural responses (Fig. 1). The parameters of the CPPN are then optimized to (i) maximally excite a target model neuron and (ii) produce diverse images. To enforce the latter objective, we train the CPPN with a contrastive objective function (Chopra et al., 2005) which encourages image diversification. Specifically, for each point $z_i \in \mathbb{R}^D$, belonging to a grid of values covering the D-dimensional latent space, the objective function to be maximized is composed of two terms:

$$\mathcal{L} = \mathcal{L}_{\text{act}} + \mathcal{L}_{\text{contrastive}} \tag{1}$$

The first term $\mathcal{L}_{\text{act}}$ represents the resulting neural activation from the generated image $I(z_i)$, and encourages the CPPN to generate images that highly activate the neuron:

$$\mathcal{L}_{\text{act}} = \frac{\alpha_i}{\alpha_{MEI}},$$

[1] Each position (x, y) gets mapped to a k-dimensional space followed by $\sin(\cdot)$ and $\cos(\cdot)$ transformations: $[\sin(\mathbf{b}[x, y]^\top), \cos(\mathbf{b}[x, y]^\top)]$ where $\mathbf{b} \sim \mathcal{N}(\mathbf{0}, \sigma\mathbf{I})$ is randomly sampled from a k-dimensional normal distribution. Here, we used $k = 10$ and $\sigma = 1$.

where α_i is the model neuron's response to image $I(z_i)$ and α_{MEI} is the neuron's MEI activation obtained through standard pixel optimization (see Appendix A for implementation details of MEI generation via pixel optimization). The normalization by the neuron's MEI activation results in a maximal objective value around 1. The second term $\mathcal{L}_{\text{contrastive}}$ is based on soft nearest neighbor contrastive objective (Salakhutdinov and Hinton, 2007; Frosst et al., 2019). It uses positive and negative images to encourage a manifold of generated images to expand and be meaningfully parameterized by the latent coordinates:

$$\mathcal{L}_{\text{contrastive}} = c \cdot \log \frac{\frac{1}{N_+} \sum_{z_j \in \mathcal{Z}_+} \exp(\text{sim}(I(z_i), I(z_j))/\tau)}{\frac{1}{N_-} \sum_{z_k \in \mathcal{Z}_-} \exp(\text{sim}(I(z_i), I(z_k))/\tau)}. \tag{2}$$

Specifically, for each latent grid point z_i a set of "positive" neighboring points $\mathcal{Z}_+$ is defined on the grid. The rest of the grid points that are further from z_i are treated as "negative" points and are denoted as $\mathcal{Z}_-$. We use cosine similarity as a similarity measure on the corresponding generated images. The numerator of the logarithm in Eq. (2) thus enforces images corresponding to close-by points to look similar, while the denominator forces images corresponding to distant points to look different. A temperature parameter τ regularizes this term (Wang and Liu, 2021) to control the diversity of images generated by the CPPN. We also used a scaling factor c to control the strength of the $\mathcal{L}_{\text{contrastive}}$ contribution to the full objective in Eq. (1). Finally, we average the single terms given by Eq. (1) across all grid points resulting in the complete objective function to maximize during training:

$$\mathcal{L} = \frac{1}{N^D} \sum_{z_i \in \mathcal{Z}} \left(\mathcal{L}_{\text{act}} + \mathcal{L}_{\text{contrastive}}\right). \tag{3}$$

Here, N^D denotes the total number of grid points, D the number of latent dimensions, and N the number of grid points per dimension.

3.3. Training the CPPN

At each step, a grid of N^D evenly spaced points covering values between 0 and 2π (in each dimension) is constructed in the latent space. To allow the CPPN to learn meaningful representations not only at discrete positions, but on the whole latent space, a random jitter $\epsilon \in [-\frac{a}{2}, \frac{a}{2}]^D$ is added to the entire grid, where a is the spacing between grid points in each latent dimension. If required, periodicity on the latent space is enforced by applying $\sin(\cdot)$ and $\cos(\cdot)$ functions on the grid points before passing them to the CPPN (i.e. $z \to [\cos(z), \sin(z)]$). The CPPN generates a grid of images corresponding to the latent grid points. Subsequently these images are rescaled to have a fixed mean (luminance) and standard deviation (contrast) and passed to the ANN model predicting neural activation. The constraint on the luminance and contrast allows for the comparison between the responses across multiple images and forces highly driving features to appear in the receptive field of the neuron, while flattening the rest of the image (for training details refer to Appendix A).

3.4. Predicting neural responses of macaque V1

Neuronal data The neuronal data have been described previously in (Cadena et al., 2022). In brief, responses of neurons in medial primary visual cortex at eccentricities ranging from 1.4 to 3.0 degrees of visual angle were recorded from two rhesus macaque monkeys. Using 32-channel linear silicon probes, a total of 458 neurons were isolated in 15 (monkey 1) and 17 (monkey 2) sessions. Neural activity was recorded in response to natural images from ImageNet (Deng et al., 2009) while the monkeys were fixating on a central fixation spot. Each image was shown for 120ms, and spikes were extracted from 40 to 160 ms after image onset. Per recording session, between 10,000 and 15,000 unique images out of a pool of 24075 ImageNet images were presented in blocks of 15. All images were displayed in grayscale, with a resolution of 63 pixels per degree (ppd), covering 6.7 degrees visual angle on the monitor.

Predictive model The artificial neural network (ANN) model we used for predicting neural responses from macaque V1 is inspired by previous deep network models (Cadena et al., 2019; Lurz et al., 2021). It consists of a nonlinear core which captures general image representations, and a readout that maps the core representations onto scalar neuronal responses via regularized regression. As core, we used a CNN with depth separable convolutions (all layers except the first), with 3 layers and 32 feature channels per layer. After each convolutional layer, a batch normalization followed by an ELU nonlinearity are applied. From the last layer, a pyramid readout (Sinz et al., 2018) extracts the features at a learned spatial location (x, y) as well as at the same location in two progressively downsampled versions of the last layer's output. We used average pooling with a kernel size of 3 in each downsampling step. $n = 96$ weights per neuron are then learned to linearly combine the features from the last layers and its two downsampled versions. The resulting outputs are then passed through an ELU$+1$ nonlinearity to finally obtain the scalar positive firing rate for each neuron.

Training the ANN on monkey V1 responses We first cropped the images to the central 2.65 degrees (from the original 6.7 degrees) and subsequently downsampled the resolution to 35 pixels per degree, leading to an input size of 93×93 pixels for the ANN model. Prior to model training, we split all stimuli into 19200 training, 4800 validation, and 75 test images, and z-scored all images based on the mean and standard deviation across the training and validation set. We trained our ANN by minimizing the Poisson loss $\frac{1}{m}\sum_{i=1}^{m}\left(\hat{r}^{(i)} - r^{(i)}\log\hat{r}^{(i)}\right)$, where m denotes the number of neurons, r the observed neuronal firing rate, and $\hat{r}$ the predicted firing rate. We then optimized the parameters of the ANN using the Adam (Kingma and Ba, 2014) optimizer with a learning rate of 0.0042. We decreased the learning rate by a factor of 0.3 when the validation loss did not decrease for three consecutive epochs for a maximum of 3 times before stopping the training altogether.

4. Results

We tested our method on simple Gabor-based model neurons with known (and exact) invariances and on neural network models predicting the responses of macaque V1 neurons. On synthetic data, we tested our approach on model neurons with a variety of invariances. While the method can be applied to arbitrarily high-dimensional invariance manifolds, here we considered model neurons with 1D and 2D invariances to easily visualize the results and facilitate interpretation of the learned invariances. Specifically, we considered a simple cell

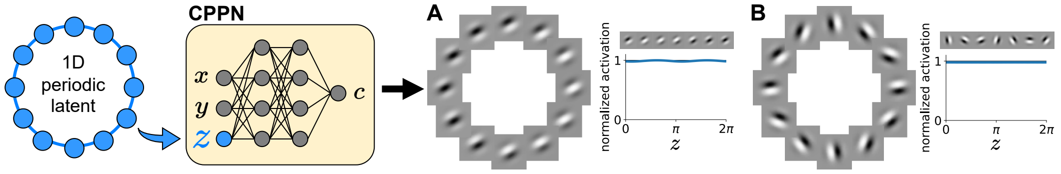


Figure 2: Invariances generated from equally spaced points in a periodic 1D latent space in the case of a complex cell (A) and of a orientation invariant neuron (B) and activation values to different images in the corresponding learned manifold

corresponding to a single point (i.e. no invariance), a complex cell (phase invariance) as well as an orientation-invariant neuron corresponding to a circle, and a phase-and-orientation-invariant neuron corresponding to a torus[2]. In the case of phase-and-orientation-invariant neuron, we additionally considered a partial orientation invariance covering only 90 degrees, resulting in a cylinder invariance topology (see Appendix B for implementation details). This variety of topologies allowed us to test how robustly our method parameterizes the entire invariance manifold, whether the parameterization associates meaningful directions to the axes of the latent space, and how it behaves when the topology of the latent space does not match the one of the invariance manifold.

Learning invariance manifolds with 1D latent spaces First, we explored how our method parameterizes 1D invariances in the case of complex and orientation-invariant model neurons. Since both phase and orientation represent periodic transformations, the invariance manifold of these cells have the topology of a circle. We therefore first tested a 1D periodic latent variable z as input to the CPPN. Our method identified the invariance manifold almost perfectly (Fig. 2). Specifically, the latent space input represents a parameter that corresponds to the angle characterizing the invariance. Next, we considered a non-periodic 1D latent space – topology of a line. In this case, the temperature parameter of the contrastive objective seems to affect the extend of the invariance manifold captured by the CPPN. The reason for this behavior is a mismatch between the true invariance topology (a circle) and the fitted topology (a line): for a line topology, opposite boundaries in the latent space correspond to negative samples, and the contrastive objective encourages them to look dissimilar discouraging the model to complete a full circle (see Appendix C for a more thorough analysis). Nonetheless, even in this case, the generated invariance manifold well adheres to a part of the ground truth invariance manifold and achieves a meaningful parameterization of the invariance (see Fig. S2).

In the scenarios we considered so far the latent space dimensionality matched the dimensionality of the invariance manifold. However, it is possible that the underlying invariance manifold is higher-dimensional than the CPPN's latent space. To see how it behaves in such a scenario, we applied a CPPN with a 1D latent space on a phase-and-orientation-invariant neuron (2D invariance). While it is not possible to capture the complete invariance mani-

[2] To be more precise, the MEI invariance manifold of a phase-and-orientation-invariant neuron has the topology of torus that touches itself for phases corresponding to even Gabors and rotations of 180 degrees.

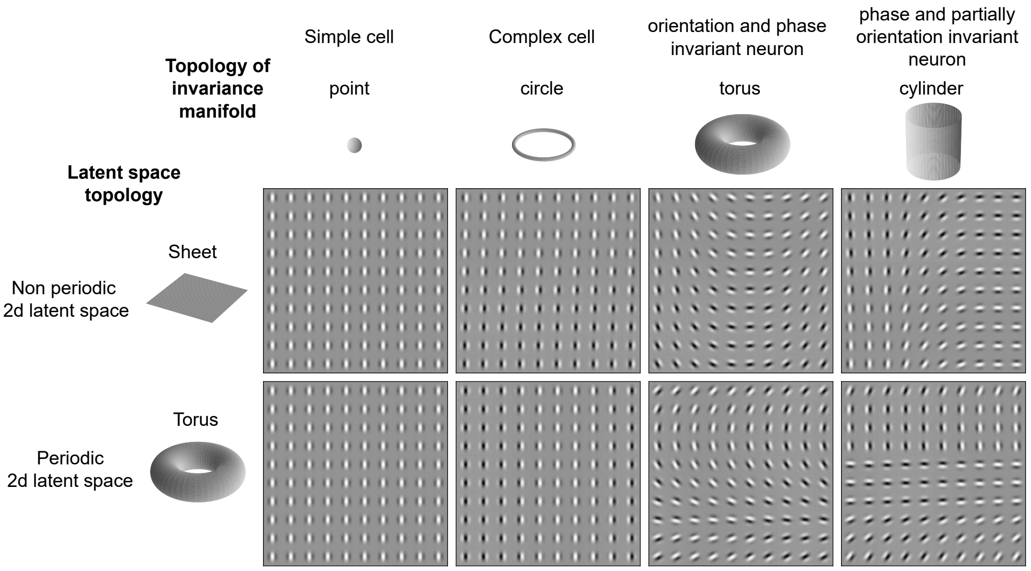

Simple cell
Complex cell
orientation and phase invariant neuron
phase and partially orientation invariant neuron
Topology of invariance manifold
point
circle
torus
cylinder
Latent space topology
Sheet
Non periodic 2d latent space
Torus
Periodic 2d latent space

Figure 3: Invariances learned with with 2D latent space for different configurations of latent space topology and topologies of the ground truth invariance manifold. Mean and standard deviation of the activations corresponding to the shown images are reported in Appendix G.

fold in this case, in Appendix D we show that our method still learns a submanifold of the higher-dimensional invariance manifold.

Our method implicitly assumes the invariance manifold to be continuous and so far we tested it on smooth ground truth invariances. As a final test, we also assessed how well it can capture discontinuous manifolds. Our results (Fig. S5) show that our method can indeed learn to approximate discontinuous invariance manifolds successfully (for details refer to Appendix E).

Learning invariance manifolds with 2D latent spaces Subsequently, we considered a 2D latent space with non-periodic (sheet topology) and periodic (torus topology) invariances and trained a CPPN to identify the invariances of all the neuron models mentioned above (Fig. 3). In the case of a simple cell, the CPPN learned to ignore the invariance latent variable z and collapsed the predicted invariance manifold onto a single point, matching the Gabor filter corresponding to the MEI of the cell (Fig. 3 left column). In the case of a complex cell, the CPPN learned to ignore one latent dimension and associated the invariance transformation with the other (Fig. 3 second column from left), as it would be expected in the ideal case. In a similar fashion to the 1D case (Fig. 2), the CPPN with non-periodic latent learned an incomplete yet meaningful parameterization of the invariance, whereas the CPPN with periodic latent learned the full invariance.

In the case of the jointly orientation-and-phase-invariant neuron, the CPPN learned both invariances and disentangled them in the latent space nearly perfectly (Fig. 3 second column

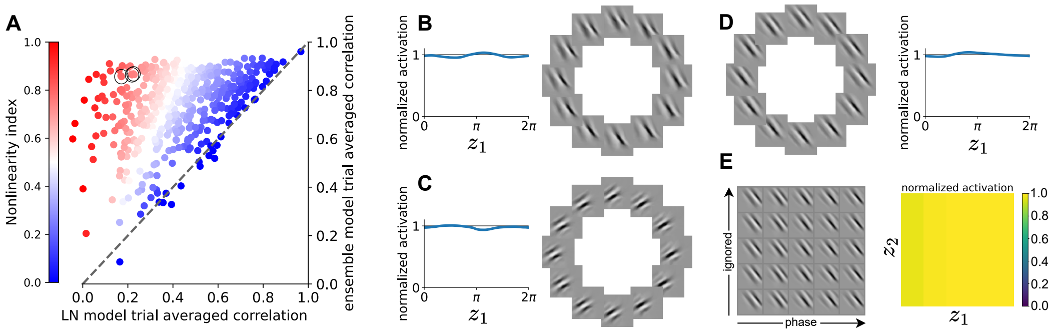


Figure 4: **A**: Nonlinearity index of macaque V1 neurons. Black circles highlight the neurons shown in panels **B**–**E**. **B**–**E**: Phase invariances identified with periodic 1D latent (**B**–**D**) and with 2D non-periodic latent(**E**) and corresponding activations. For visualization purposes, MEIs are cropped around the receptive field of the neurons.

from right). This result is particularly relevant, as it allows both clear identification of the invariance and control over it via the latent space. Again, the CPPN mapping from non-periodic latent space parameterized only half of the true periodic invariance transformations. We then explored the scenario of a phase-and-partially-orientation-invariant model neuron. The invariance manifold of this neuron model has the topology of a cylinder. In this case, the CPPN with a non-periodic latent space learned both invariance transformations and disentangled them along the two latent space dimensions (Fig. 3 right column top row). In line with the previous results, the partial orientation invariance transformation was learned fully, whereas the phase transformation was learned up to a 180 degree phase shift (Fig. 3 right column top row). Fitting a periodic latent space on a non-periodic invariance manifold topology is more complex. Specifically, to deal with non-periodic invariances, the CPPN mapping from periodic latent can either learn the full invariance transformation twice, or introduce sudden jumps in the invariance manifold (Fig. 3 right column bottom row). Our experiments indicate that both scenarios can happen (see Fig. S6) and that the CPPN can still learn to disentangle the two transformations.

Learning the invariance manifold of macaque V1 complex cells Lastly, we set out to validate our method on a model for a population of macaque V1 neurons (see section 3.4). The Gabor-based model neurons above presented (almost) exact invariances, with no fluctuations over activation levels. In a biological neuron, however, a meaningful definition of the MEI invariance manifold should be more forgiving, allowing for a broader variety of images to be considered maximally exciting, for the following reasons: First, it is not to be expected that biological neurons present exact invariances over maximally exciting stimuli. Second, the data collected and analyzed in neurophysiological experiments are intrinsically noisy and limited in size. Third, our experiments here are performed on neural network models fitted to neural responses, which despite achieving high predictive performance, are not perfect. We used an ensemble model of ANNs (see section 3.4) as a model of of macaque V1 neurons and applied our method on complex cells that were identified using a nonlinearity index (Antolík et al., 2016). See Appendix H for details on selecting complex cells. Fig. 4 shows that the CPPN found phase invariance in the selected neurons: it generated a variety of maximally exciting images resembling Gabor filters and parameterized their phase transformation with one of the latent space dimension (see Appendix I for a more thorough analysis of the learned phase invariance). This demonstrates the ability of the method to identify invariances in biological neural representations.

5. Discussion

We presented a data-driven method that combines a CPPN with a contrastive learning objective to map from a low-dimensional latent space to a manifold in the space of images that describes the MEI invariances of a given neuron. We tested our approach on synthetic neural responses, where multiple ground truth exact invariance manifolds were known, as well as on predictive models of macaque V1 complex cells. We showed that our approach successfully uncovers MEI invariance manifolds in both scenarios. In contrast to previously presented approaches, our method allows a smooth parameterization of the invariance manifold and, when multiple invariances are present, it disentangles them along the axes of the latent space. When the dimensionality of the latent space is higher than the dimensionality of the invariance, the CPPN learns to ignore unnecessary latent dimensions.

In the future, our approach can be extended to learn implicit representation of MEI invariances for multiple neurons, for instance by associating to each neuron a learnable embedding used as a fingerprint. Such multi-neuron implementation, in combination with a regularization of the space of fingerprints, could, in principle, allow us to classify neurons in functional clusters, according to their invariances. Similarly, a multi-neuron implementation could allow us to study interesting tuning directions, i.e. directions in image space to which certain neurons are selective, whereas others are invariant. We believe that the approach presented here will prove to be a valuable tool to advance our understanding of visual sensory coding, especially in the higher visual areas, such as V4, that potentially exhibit more invariances, both in terms of quantity and complexity.

Acknowledgments

We thank Alexander Ecker and the members of Sinz lab for their helpful feedback and discussions. The authors thank Tolias lab for providing recordings from macaque V1. We also thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for funding MB and supporting KFW. This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Maria Sklodowska-Curie grant agreement No 861423. FHS was 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.

References

Edward H Adelson and James R Bergen. Spatiotemporal energy models for the perception of motion. *Josa a*, 2(2):284–299, 1985.

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

Mohammad Bashiri. Learning gabor filters via gradient descent, 2020. URL https://github.com/mohammadbashiri/fitgabor.

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

Lukas Biewald. Experiment tracking with weights and biases, 2020. URL https://www.wandb.com/. Software available from wandb.com.

Santiago A Cadena, Marissa A Weis, Leon A Gatys, Matthias Bethge, and Alexander S Ecker. Diverse feature visualizations reveal invariances in early layers of deep neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 217–232, 2018.

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.

Santiago A Cadena, Konstantin F Willeke, Kelli Restivo, George Denfield, Fabian H Sinz, Matthias Bethge, Andreas S Tolias, and Alexander S Ecker. Diverse task-driven modeling of macaque v4 reveals functional specialization towards semantic tasks. *bioRxiv*, 2022.

Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Zhiwei Ding, Dat Tran, Erick Cobos, Taliah Muhammad, Kayla Ponder, Santiago Cadena, Alexander Ecker, Xaq Pitkow, and Andreas Tolias. Inception loops reveal novel spatially-localized phase invariance in mouse primary visual cortex, 2022. URL https://static1.squarespace.com/static/6102ca347474c263c40150cd/t/62325b5f6dbf95289c4472e3/1647467367870/Cosyne2022_program_book.pdf#page=221. COSYNE conference 2022 booklet page 205, 3-034.

Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009.

Nicholas Frosst, Nicolas Papernot, and Geoffrey Hinton. Analyzing and improving representations with the soft nearest neighbor loss. In *International conference on machine learning*, pages 2012–2020. PMLR, 2019.

David Ha. Generating large images from latent vectors. *blog.otoro.net*, 2016. URL https://blog.otoro.net/2016/04/01/generating-large-images-from-latent-vectors/.

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.

David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106, 1962.

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

David Klindt, Alexander S Ecker, Thomas Euler, and Matthias Bethge. Neural system identification for large populations separating "what" and "where". *Advances in Neural Information Processing Systems*, 30, 2017.

Nikolaus Kriegeskorte. Deep neural networks: a new framework for modelling biological vision and brain information processing. *biorxiv*, page 029876, 2015.

Jonas Kubilius, Martin Schrimpf, Kohitij Kar, Rishi Rajalingham, Ha Hong, Najib Majaj, Elias Issa, Pouya Bashivan, Jonathan Prescott-Roy, Kailyn Schmidt, et al. Brain-like object recognition with high-performing shallow recurrent anns. *Advances in neural information processing systems*, 32, 2019.

Konstantin-Klemens Lurz, Mohammad Bashiri, Konstantin Willeke, Akshay K Jagadish, Eric Wang, Edgar Y Walker, Santiago A Cadena, Taliah Muhammad, Erick Cobos, Andreas S Tolias, et al. Generalization in data-driven models of primary visual cortex. *BioRxiv*, pages 2020–10, 2021.

Dirk Merkel. Docker: lightweight linux containers for consistent development and deployment. *Linux journal*, 2014(239):2, 2014.

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Alexander Mordvintsev, Nicola Pezzotti, Ludwig Schubert, and Chris Olah. Differentiable image parameterizations. *Distill*, 2018. doi: 10.23915/distill.00012. https://distill.pub/2018/differentiable-parameterizations.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017. doi: 10.23915/distill.00007. https://distill.pub/2017/feature-visualization.

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 *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.

Carlos R Ponce, Will Xiao, Peter F Schade, Till S Hartmann, Gabriel Kreiman, and Margaret S Livingstone. Evolving images for visual neurons using a deep generative network reveals coding principles and neuronal preferences. *Cell*, 177(4):999–1009, 2019.

Ruslan Salakhutdinov and Geoff Hinton. Learning a nonlinear embedding by preserving class neighbourhood structure. In *Artificial Intelligence and Statistics*, pages 412–419. PMLR, 2007.

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. *Advances in neural information processing systems*, 31, 2018.

Kenneth O Stanley. Compositional pattern producing networks: A novel abstraction of development. *Genetic programming and evolvable machines*, 8(2):131–162, 2007.

Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020.

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.

Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2495–2504, 2021.

Daniel LK Yamins and James J DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nature neuroscience*, 19(3):356–365, 2016.

Daniel LK Yamins, Ha Hong, Charles F Cadieu, Ethan A Solomon, Darren Seibert, and James J DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the national academy of sciences*, 111(23):8619–8624, 2014.

Chengxu Zhuang, Siming Yan, Aran Nayebi, Martin Schrimpf, Michael C Frank, James J DiCarlo, and Daniel LK Yamins. Unsupervised neural network models of the ventral visual stream. *Proceedings of the National Academy of Sciences*, 118(3):e2014196118, 2021.

Appendix A. Additional training details

A.1. CPPN training details

The CPPN was trained via gradient-based optimization to maximize the objective in Eq. 3. During training, the contrastive objective requires a set of positive and negative input images for each latent input. This was achieved with the construction of masks identifying positive and negative points. Positive neighboring areas are as squares surrounding the point considered, and extending from it in each direction for $0.1 * N$ number of points (Fig. S1). The periodicity condition of the latent space is reflected in the masks of points close to the boundaries. Regularization strengths were rescaled as $c = \bar{c} \times \frac{\tau}{2}$ to normalize the maximum possible contribution coming from a single point in the contrastive term depending on temperature. We observed that strong initial regularization seems to disentangle the invariant directions and to avoid sudden jumps in image space as a function of the latent space, but can concurrently deteriorate the MEIs generated. This is due to the fact that the objective in this case has to balance between maximizing activation and satisfying comparably strong regularization conditions. For this reason during training we decrease $\bar{c}$.

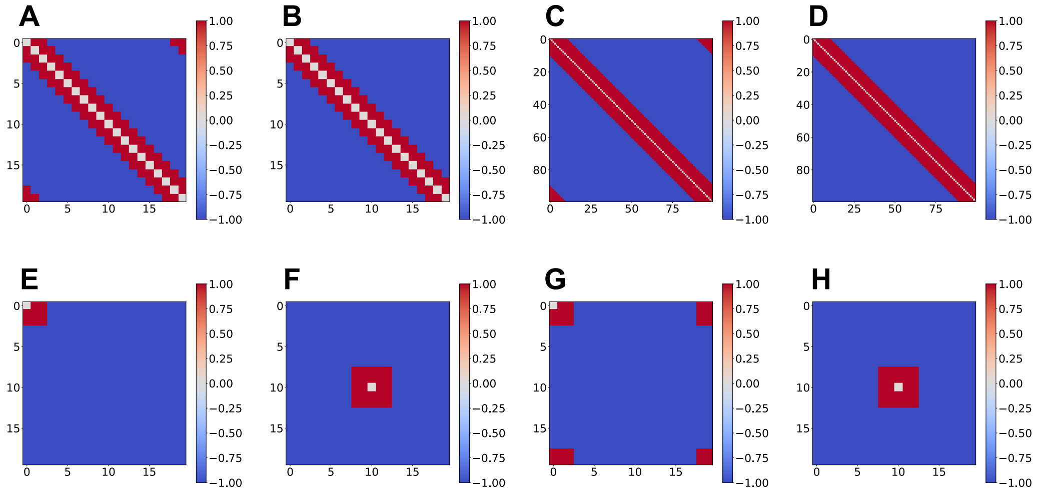


Fig. S1: Masks to determine positive (red) and negative (blue) examples on the grid for different conditions. Neighbouring size set to $s_n = 0.1$. (**A-D**): Masks for all points in 1D grid (rows of each matrix) under periodic conditions of latent (**A**,**C**): non periodic conditions (**B**,**D**), and grids with different number of points, $n_p = 20$ (**A**,**B**)) and $n_p = 100$ (**C**,**D**). (**E-F**): Mask for single points in 2D grids close to latent space boundaries (**E**,**G**) and away from them (**F**,**H**), under non periodic latent space (**E**,**F**) and non periodic latent space (**G**,**H**).

Synthetic 1D case Temperature was set to $\tau = 1$, contrastive regularization strength coefficient to $\bar{c} = [2, 0.5]$. Number of points per dimension of the grid was set to $N = 100$.

Number of batches (each corresponding to a grid) per epoch was set to 100 and models were trained for 10 epochs per each value of $\bar{c}$. Learning rate was set to 0.01. Before being presented to the model neurons, all images where rescaled to have a mean of 0 and a standard deviation of 0.2. Images were generated with a 30×30 pixel resolution, matching the resolution of the Gabor based neuron models.

Synthetic 2D case Temperature was set to $\tau = 1$ in the case of non-periodic latent $\tau = 0.3$ in case of periodic latent (see Appendix C). Constrastive regularization strength coefficients were set to $\bar{c} = [1, 0.5]$. Number of points per grid dimension was set to 20, resulting in grids of 400 points. Learning rate was set to 0.01, number of batches (each corresponding to a grid) per epoch was set to 100 and models were trained for 100 or 120 epochs per each value of $\bar{c}$ respectively in the case of non periodic latent space and periodic latent space. Nonetheless, CPPNs appeared to converge much faster for each regularization strength coefficient considered (20 epochs being sufficient). Before being presented to the model neurons all images where rescaled to have a mean of 0 and a standard deviation of 0.2. Images were generated with a 30×30 pixel resolution, matching the resolution of the Gabor based neuron models.

Macaque complex cell Temperature was set to $\tau = 1$, contrastive regularization strength coefficient to $\bar{c} = [1, 0.5]$. Number of points per dimension of the grid was set to $N = 20$. Number of batches (each corresponding to a grid) per epoch was set to 50 and models were trained for 10 and 20 epochs per each value of $\bar{c}$ in the 1D and 2D case, respectively. Learning rate was set to 0.05 and 0.01 in 1D case and 2D case, respectively. Before being presented to the ensemble model all images where rescaled to have a mean 0.2019 (corresponding to the mid grayscale value of the images on which the ANNs in the ensemble model were fitted), to have a standard deviation 0.15, and if necessary pixel values were clipped to the values corresponding to the extremes of the grayscale on the images on which the macaque model was fitted $[2.1919, -1.7876]$. The standard deviation value was selected to allow clear identification of maximally exciting features while avoiding excessive clipping.

A.2. Pixel-based MEI optimization

In this method, the pixels of an image are defined as learnable parameters and are learned via gradient-based optimization such that the activation of a target neuron is maximized. Specifically, we defined an input image of size 93×93 for monkey V1 neuron and 30×30 for Gabor based neurons and used the Adam optimizer with learning rate of 0.01 to obtain an MEI after 2000 training steps.

A.3. Software and hardware specifications

All code for model definition, training, evaluation and experiment tracking were implemented in Python 3.9 using PyTorch (Paszke et al., 2019), NumPy (Harris et al., 2020), Weights & Biases (Biewald, 2020), and Docker (Merkel, 2014) packages. All CPPN models were trained using the Adam (Kingma and Ba, 2014) optimizer on a Tesla V100-SXM2-32GB GPU, and took a few minutes to train.

Appendix B. Gabor-based model neurons

On synthetic data, we tested our approach on a simple cell, a complex cell, an orientation-invariant neuron and two phase-and-orientation-invariant neurons.

- The simple cell was implemented as a Gabor filter followed by ReLU nonlinearity.
- The complex cell was implemented as an energy model (Adelson and Bergen, 1985).
- The orientation-invariant neuron was implemented as a set of simple cells with different orientations, followed by a max-pooling operation.
- The phase-and-orientation-invariant neurons were implemented as sets of complex cells with different orientations, followed by a max-pooling operation.

Appendix C. The effect of temperature on invariance manifold learning

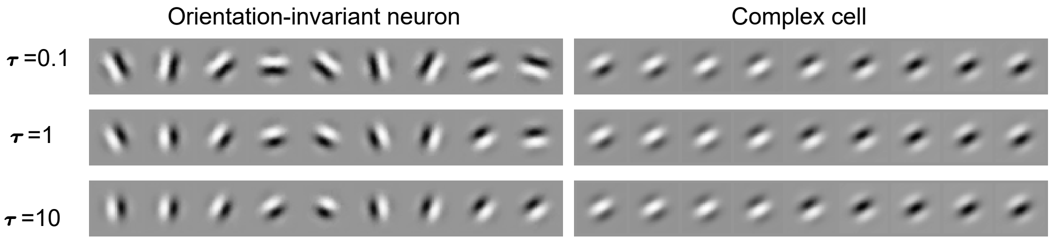


Fig. S2: Images in invariance manifolds corresponding to equally spaced points in 1D non periodic latent space for different temperature values (row) and invariances (columns)

Different choice of temperature can have a important effect on the extent of the invariance manifold that is learned. Fig. S2 shows how in the case of a non-periodic 1D latent space the extent depends as well on the type of invariance of the model neuron. This is due to the fact that images belonging to different invariant transformations have different similarity values and because the temperature parameter controls how strongly to encourage and penalize similarity to positive samples and negative samples, respectively.

In the case of 1D non-periodic latent space, the invariance manifold corresponding to an orientation invariant odd Gabor neuron (Fig. S2 left column) ranges from being almost complete transformation (for low temperatures) to be slightly more than half (high temperature). The phase invariance of complex cell (Fig. S2 right column), on the contrary, is learned up to half a transformation (180 degrees) for all the temperature values considered.

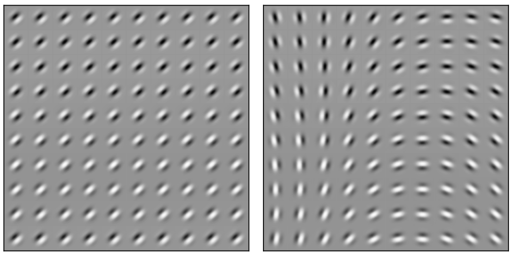


Fig. S3: Effect on temperature on learning 2D invariances. Both figures correspond to an phase-and-orientation-invariant neuron whose invariances are learned with a CPPN with 2D periodic latent space, but with different temperature, respectively $\tau = 10$ and $\tau = 0.3$ from left to right.

When a neuron presents multiple invariances, temperature has an effect also on *which* invariance transformations are learned. See Fig. S3 as example. For a phase-and-orientation-invariant neuron, a CPPN mapping from a 2D periodic latent space is able to capture orientation invariance only at low temperatures. At high temperature only phase invariance is learned and one of the latent space axes is ignored. This results shows how the optimization objective can, for specific invariances and temperature values, be maximized in the scenario in which the CPPN selectively learns only one of the invariances.

Appendix D. Learning 2D invariances with 1D latent space

In this appendix we assess the outcome of our method when the invariance manifold of the neuron considered is higher dimensional than the latent space from which the CPPN maps. For this purpose we considered the scenario in which a CPPN mapping from a 1D periodic latent space is trained to identify the invariance of a phase-and-rotation-invariant neuron (2D invariance manifold). We performed the same experiment for high temperature ($\tau = 10$), low temperatures ($\tau = 0.3$) and multiple seeds. With the exception of temperature, training details match the ones reported for 1D synthetic case. Results are shown in Fig. S4.

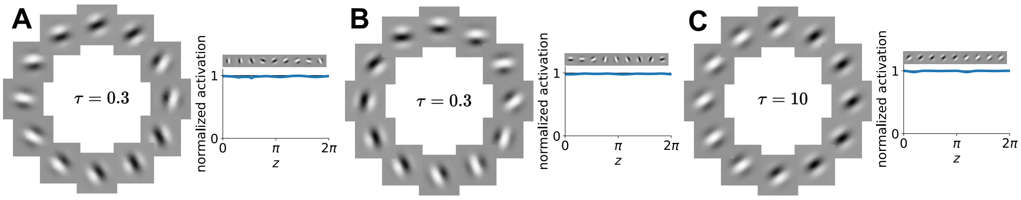


Fig. S4: A CPPN mapping from a 1D periodic latent space is trained to capture the invariance manifold of a phase-and-rotation-invariant-neuron. Images corresponding to equidistant points in latenst space are shown, together with activation. (**A**-**B**) corresponds to different seed, (**B**-**C**) to different values of temperature.

As can be seen from the diversity of images smoothly varying and from the activation being maximized for all images, in all cases considered the CPPN learns a 1D submanifold of the 2D invariance manifold. Fig. S4 further illustrates however how the nature of the learned submanifold might depend on multiple factors such as the nature of the invariances in higher dimensional invariance manifold (e.g. phase vs orientation), the CPPN initialization (inter-seed variability), and training details (e.g., optimization objective). In the case considered, the only invariance learned in the case of high temperature is phase (similarly to what happens in S3). This results shows how the optimization objective can, for specific parameter configurations, be maximized in the scenario in which the CPPN ignores one of the invariances.

Appendix E. Approximating a discontinuous invariance manifold with continuous parameterization

The method presented implicitly assumes the MEI invariance manifold to learn to be continuous, as many interesting biological neurons' invariances are smooth. A given neuron could however present a discontinuous invariance. In this appendix we illustrate how the CPPN can address this situation learning to approximate a discrete invariance manifold with a continuous latent space, thanks to the introduction of jumps. Specifically we considered a polarity invariant neuron obtained max pooling the responses of ON and OFF centered even simple cells (same parameters, except for phase). Such neuron presents an invariance manifold that consists in two point in the image space, corresponding to the ON and OFF centered linear filters of the simple cells from which it is composed. We trained to CPPN mapping from a 1D periodic latent space to approximate the discontinuous invariance manifold of such polarity invariant neuron. Temperature was set to $\tau = 0.3$. Remaining training details match the ones reported for the 1D synthetic case in appendix A.

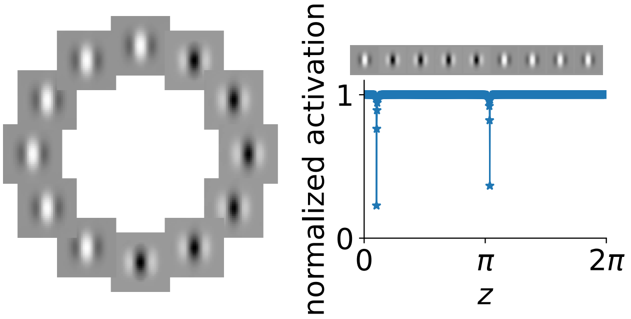


Fig. S5: Learned invariance manifold and corresponding activations in the case of a CPPN mapping from a 1D periodic latent space and trained to learn the discrete manifold of a polarity invariant neuron MEIs are reported for 12 equally spaced points and activation for 1000 equally spaced points in the latent space. Two domains, each of them corresponding to one of the two MEIs, appear in latent space. The necessity of connecting such domains via a continuous parameterization corresponds to the arising of sudden jumps in image space connecting such domains, during which activation drops. The small fraction of points corresponding to activation sensibly deviating from MEI activation gives a measure of how localized jumps are in latent space.

Fig. S5 demonstrates that CPPN can learn to approximate the two-point invariance manifold mapping large domains of the latent space to the same maximally exciting image and introducing sudden jumps between such domains. This approximate learning of a discontinuous and discrete manifold with a continuous latent space is possible thanks to the decreasing strength of the contrastive objective during training. In the first part of the training, the high regularization strength c of the contrastive objective tends to be predominant and the CPPN learns a smooth manifold of images that are particularly diverse, that tend to highly activate the neuron but that are not exactly maximally exciting. When the regularization strength decreases, however, the activation objective becomes predominant over the contrastive objective that ensures smoothness and the CPPN learns to introduce jumps between the domains in which the generated images look the same.

Appendix F. Periodic 2D latent space on cylinder invariance topology

This appendix displays some of the results obtained when fitting a 2D periodic manifold on a non-periodic invariance (cylinder topology).

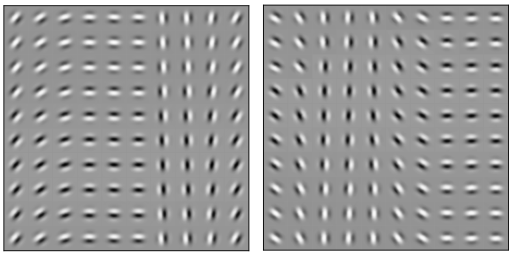


Fig. S6: Two different instantiations (same hyperparameters, different seeds) of the same CPPN learning a phase-and-partially-orientation-invariant neuron invariance manifold with a 2D periodic latent. Fitting a 2D periodic latent on a non-periodic invariance manifold forces either sudden jumps (left figure, jump on the orientation axis) or to learn the same transformation twice (right figure, rotation of 90 degrees is learned first clockwise and than anticlockwise)

Appendix G. Activations corresponding to images in Fig. 3

	simple cell	complex cell	phase and orientation invariant neuron	phase and partially orientation invariant neuron
2D non periodic latent	1 ± 8e-8	0.991 ± 0.006	0.991 ± 0.007	0.991 ± 0.006
2D periodic latent	1 ± 9e-8	0.990 ± 0.007	0.985 ± 0.007	0.985 ± 0.006

Table 1: Mean and standard deviation of the activations corresponding to images generated from the learned invariance manifold when using a 2D latent space (corresponding to images shown in Fig. 3).

Appendix H. Selection of complex cells

We identified complex cells following these steps:

1. We created an ensemble model, averaging the predictions of $n = 3$ ANNs implemented and trained as described in 3.4.

2. We trained multiple instances of a linearized version (LN models) of the ANN model obtained by simply dropping the nonlinearities in the model (except the last one which ensures positive-values firing rates).

3. We computed a nonlinearity index for each neuron by comparing the correlation of the ensemble model predictions with trial-averaged neural responses with the highest correlation achieved by any of the trained LN models: $I_{nl} = (c_{ens} - \max(c_{lin}, 0))/c_{ens}$.

4. Among neurons with high I_{nl} we selected the ones with $c_{ens} > 0.8$ to consider only nonlinear neurons well modelled by our ensemble.

5. We performed direct pixel optimization to identify one MEI per neuron

6. We selected the neurons whose MEI visually resembled a Gabor filter.

Appendix I. Analysis of the MEIs generated via CPPN

To better show that our method has captured the previously shown phase invariance in monkey V1 complex cells, for each image generated via the CPPN (Fig. 4**B**–**D**) we learned a Gabor filter that results in the least mean squared error (Bashiri, 2020), and assessed the phase of the learned Gabor filters. Fig. S7 shows that the learned Gabor filters are a close match to the MEIs both qualitatively (i.e. visually) and quantitatively (i.e. resulting activations). Importantly, as we go along the learned invariance manifold the phase of the learned Gabor smoothly changes and the manifold covers the complete 2π cycle.

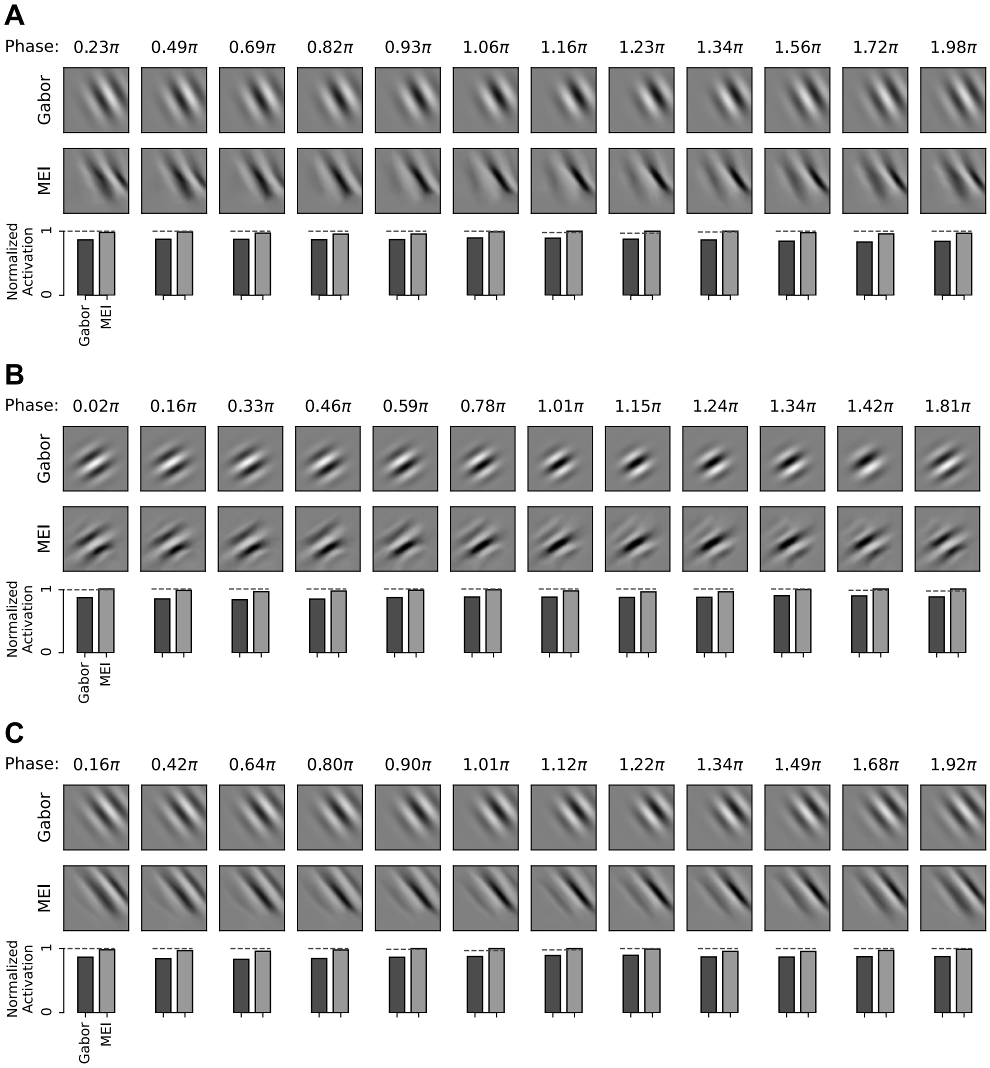
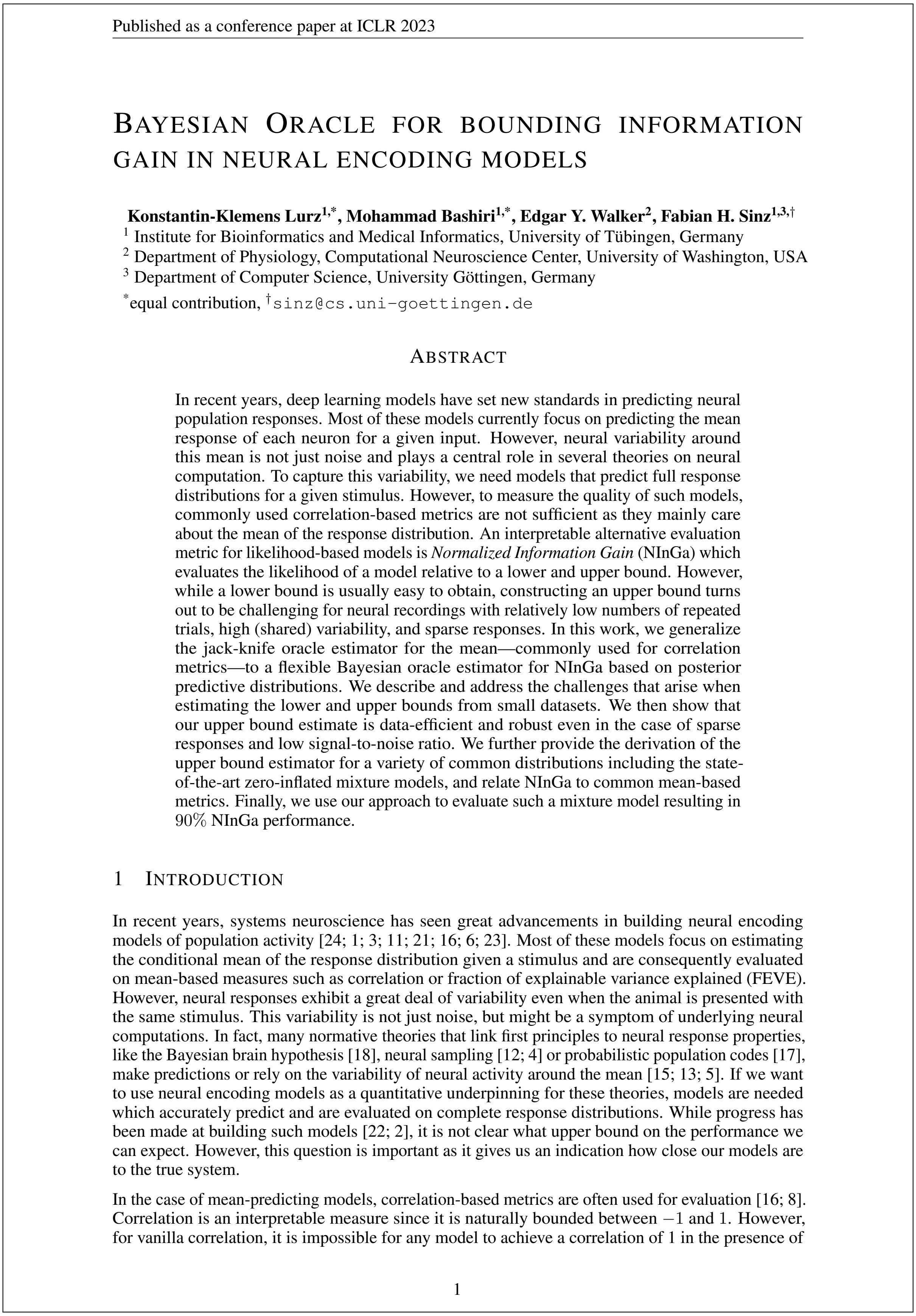


Fig. S7: Analysis of the MEIs generated via CPPN using fitted Gabors. **A**, **B**, and **C** correspond to the neurons shown in Fig. 4B, 4C, and 4D, respectively. For visualization purposes, MEIs and Gabors are cropped around the receptive field of the neurons.

Manuscript 3



Bayesian Oracle for bounding information gain in neural encoding models

Konstantin-Klemens Lurz[1,*], **Mohammad Bashiri**[1,*], **Edgar Y. Walker**[2], **Fabian H. Sinz**[1,3,†]
[1] Institute for Bioinformatics and Medical Informatics, University of Tübingen, Germany
[2] Department of Physiology, Computational Neuroscience Center, University of Washington, USA
[3] Department of Computer Science, University Göttingen, Germany
[*]equal contribution, [†]`sinz@cs.uni-goettingen.de`

Abstract

In recent years, deep learning models have set new standards in predicting neural population responses. Most of these models currently focus on predicting the mean response of each neuron for a given input. However, neural variability around this mean is not just noise and plays a central role in several theories on neural computation. To capture this variability, we need models that predict full response distributions for a given stimulus. However, to measure the quality of such models, commonly used correlation-based metrics are not sufficient as they mainly care about the mean of the response distribution. An interpretable alternative evaluation metric for likelihood-based models is *Normalized Information Gain* (NInGa) which evaluates the likelihood of a model relative to a lower and upper bound. However, while a lower bound is usually easy to obtain, constructing an upper bound turns out to be challenging for neural recordings with relatively low numbers of repeated trials, high (shared) variability, and sparse responses. In this work, we generalize the jack-knife oracle estimator for the mean—commonly used for correlation metrics—to a flexible Bayesian oracle estimator for NInGa based on posterior predictive distributions. We describe and address the challenges that arise when estimating the lower and upper bounds from small datasets. We then show that our upper bound estimate is data-efficient and robust even in the case of sparse responses and low signal-to-noise ratio. We further provide the derivation of the upper bound estimator for a variety of common distributions including the state-of-the-art zero-inflated mixture models, and relate NInGa to common mean-based metrics. Finally, we use our approach to evaluate such a mixture model resulting in 90% NInGa performance.

1 Introduction

In recent years, systems neuroscience has seen great advancements in building neural encoding models of population activity [24; 1; 3; 11; 21; 16; 6; 23]. Most of these models focus on estimating the conditional mean of the response distribution given a stimulus and are consequently evaluated on mean-based measures such as correlation or fraction of explainable variance explained (FEVE). However, neural responses exhibit a great deal of variability even when the animal is presented with the same stimulus. This variability is not just noise, but might be a symptom of underlying neural computations. In fact, many normative theories that link first principles to neural response properties, like the Bayesian brain hypothesis [18], neural sampling [12; 4] or probabilistic population codes [17], make predictions or rely on the variability of neural activity around the mean [15; 13; 5]. If we want to use neural encoding models as a quantitative underpinning for these theories, models are needed which accurately predict and are evaluated on complete response distributions. While progress has been made at building such models [22; 2], it is not clear what upper bound on the performance we can expect. However, this question is important as it gives us an indication how close our models are to the true system.

In the case of mean-predicting models, correlation-based metrics are often used for evaluation [16; 8]. Correlation is an interpretable measure since it is naturally bounded between -1 and 1. However, for vanilla correlation, it is impossible for any model to achieve a correlation of 1 in the presence of

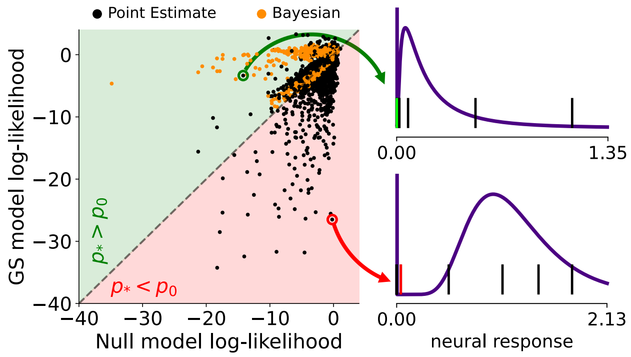


Figure 1: Comparison of lower and upper bound likelihood estimates (Null vs GS) per neuron. **Left:** For many neurons, the PE approach yields worse GS than the Null score. The Bayesian method results in the expected outcome of upper bound scores being higher than lower bound scores. **Right:** Two example neurons demonstrating where the PE method fails (red) or succeeds (green).

trial-to-trial fluctuations. Therefore, model correlation is often normalized by an upper bound oracle estimator [19; 16], which is commonly obtained by computing point estimates of the conditional mean using the responses to repeated presentations of the same stimulus. For a likelihood-based metric, a similar normalization to a bounded and interpretable scale would be desirable, especially for: 1) Assessing whether a model has achieved its “best possible” performance for a given dataset, and 2) comparing models that are trained on different datasets, which can exhibit different levels of achievable performance. To this end, one can use Normalized Information Gain (NInGa) [14], which uses an estimate of both upper and lower bound, to put the likelihood between two meaningful values. However, the challenge lies in how these bounds can be obtained for noisy neural responses.

In this work, we develop a robust way to estimate such lower and upper bounds for NInGa on neuronal responses. We show that a point estimate approach for obtaining the upper bound fails and demonstrate that this is caused by the lack of robustness for the estimate of moments beyond the mean. This is especially pronounced when dealing with data that have few samples, sparse responses, and low signal-to-noise ratios which are common characteristics of neural responses. To mitigate this problem, we propose a generalization of the point estimate approach to a full Bayesian treatment using posterior predictive distributions. Our approach yields lower and upper bounds which are proven to be robust to all the above-mentioned complexities in neural data. We derive a general expression for the Bayesian estimator for zero-inflated distributions that can be efficiently estimated under very general conditions by solving only a single one-dimensional integral on a bounded interval. These distributions capture the sparse nature of neural responses, in particular for 2-photon recordings, and include state-of-the-art zero-inflated mixture models [22; 2]. Using this full-likelihood-based metric, we then evaluate a zero-inflated mixture model and find that it performs remarkably well at 90% NInGa. Finally we experimentally and mathematically relate NInGa to other common metrics for the performance of neural prediction models which are based on the mean and derive general conditions under which likelihood and correlation as a metric identify the same predictive function.

2 Methods

2.1 Normalized Information Gain and problems with point estimates

Information Gain Let $p(y|x)$ denote the distribution of a neuron's response y to a stimulus x. In order to evaluate and interpret the modeled distribution $\hat{p}(y|x)$ we use Normalized Information Gain (NInGa) [14; 20] which sets the model likelihood on an interpretable scale between an estimated lower and upper bound:

$$\text{NInGa} = \frac{\langle \log \hat{p}(y \mid x) \rangle_{y,x} - \langle \log p_0(y) \rangle_{y,x}}{\langle \log p_*(y \mid x) \rangle_{y,x} - \langle \log p_0(y) \rangle_{y,x}} \tag{1}$$

using a Null distribution $p_0(y)$ and a Gold Standard distribution $p_*(y|x)$. This method of computing NInGa can be interpreted as a normalized comparison of lower bounds of mutual information [20, and Appendix E]. In general, IG is completely flexible in the choice of Null, Gold Standard, and trained model distribution. Therefore, it can be used for model comparison across different distribution families as long as the GS and Null model are kept the same. The *Null model* should reflect basic aspects of the response. Here, we choose a Null model that does not account for any stimulus-related information, resulting in the marginal distribution of responses $p_0(y)$. The *Gold Standard (GS) model* $p_*(y|x)$, on the other hand, should be the best possible approximation of the true conditional distribution $p(y|x)$. Here, we use an oracle model that has more information than the model under evaluation, such as access to repeated presentations of the same stimulus. Importantly, we do this in a leave-one-out fashion: given a set of n repeats, the GS parameters of a target repeat i are estimated from $n-1$ left-out repeats $\setminus i$. However, as we demonstrate below, estimating a robust GS model can be challenging.

Point Estimate (PE) GS model The parameters θ of the upper bound estimator can be obtained as point estimates (PE) from $n-1$ left-out repeats:

$$p_*(y_i|\mathbf{y}_{\setminus i}, x) = p(y_i|\theta_i) \quad \text{with } \theta_i = f(\mathbf{y}_{\setminus i}),$$

where f is a function used to obtain the point estimate of θ. In our case, f represents moment matching. For correlation-based performance metrics of neural prediction models, a jack-knifed mean estimator over repeated presentations of the same stimulus $E[y_i|x] = \frac{1}{n-1}\sum_{y_j \in y_{\setminus i}} y_j$ is commonly used as an oracle predictor for the conditional mean to obtain an upper bound on the achievable performance in the presence of noise [19; 16]. While the posterior predictive distribution is generally conditioned on stimulus x because it requires responses to the repeated presentations of the same stimulus, for the remaining of this manuscript we will drop the conditioning on x for brevity.

Problems with the PE approach To demonstrate the problems with point estimate GS models, we modeled neural responses with a zero-inflated Log-Normal likelihood and estimated the upper bound using the PE approach (see Appendix A.1 for details on data, [22; 2] for details on zero-inflated distributions, and Appendix B for the moment matching derivations). Since the GS model estimates parameters per stimulus, it should yield higher likelihood values than the Null model whose parameters are not stimulus-specific. However, applying the PE approach to neural data, we observed that the Null model outperforms the GS model for the majority of neurons (Fig. 1, black points). The reason for this effect is that the PE approach is sensitive to the sparse distribution of the data, which combined with few responses per stimulus results in an overconfident estimation of the GS parameters (see Fig. 1 on the right; Appendix D for a more detailed analysis on where the PE fails).

2.2 Bayesian Gold Standard Model

Gold Standard Model based on posterior predictive distributions To avoid an overconfident GS model, we add uncertainty to the parameter estimation and estimate the GS model in a fully Bayesian fashion via the full posterior predictive distribution:

$$p_*(y_i|\mathbf{y}_{\setminus i}) = \int_{-\infty}^{\infty} \underbrace{p(y_i|\theta)}_{\text{likelihood}} \underbrace{p(\theta|\mathbf{y}_{\setminus i})}_{\text{posterior}} \, \mathrm{d}\theta \tag{2}$$

Note that the PE approach is a special case within this framework for which $p(\theta|\mathbf{y}_{\setminus i}) = \delta(\theta - f(\mathbf{y}_{\setminus i}))$ is collapsed onto a delta distribution.

Efficient estimation for zero-inflated distributions In general, the integral in equation 2 is intractable and the posterior predictive distribution can only be evaluated using numerical approximations. Only for certain choices of likelihood the integral can be solved analytically if the right conjugate prior was chosen.

Here, we show that the posterior predictive distribution can be efficiently computed for the class of zero-inflated distributions [22, Fig. 2]. In principle, such a distribution is a mixture of a delta-distribution at zero and a density of a positive part. In practice, we replace the delta-distribution with a uniform distribution in a small interval $[0, \tau)$ and shift the positive part to the interval $[\tau, \infty)$ so that the two mixture components do not overlap

The probability density for a single stimulus x is then defined as:

$$\hat{p}(y|x) = (1 - q(x)) \cdot \underbrace{p_u(y)}_{\text{uniform}} + q(x) \cdot \underbrace{p(y|\theta_1(x))}_{\text{positive distribution on } [\tau, \infty)}, \tag{3}$$

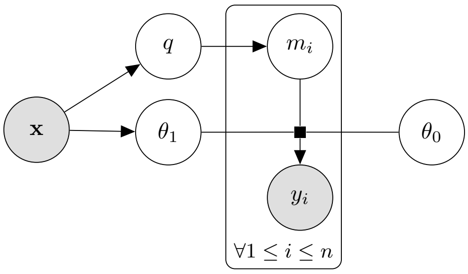


Figure 2: Graphical model for a zero-inflated distribution. A stimulus $\mathbf{x}$ determines the probability q whether a neurons fires or not and the parameters θ_1 of the response distribution of the non-zero response distribution. A Bernoulli random variable m_i determines whether a neuron fires on a particular trial i. If $m_i = 1$ a response is drawn from $p(y_i|\theta_1)$, otherwise from $p(y_i|\theta_0)$.

where the mixing proportion q and the parameters θ_1 are dependent on x. When using a predictive model, q and θ_1 are typically predicted from the stimulus x using an encoding model. Zero-inflated likelihoods are the basis of current state-of-the-art likelihood-based neural encoding models [22; 2].

Here we show that the posterior predictive distribution for a zero-inflated likelihood boils down to a one dimensional integral over q if the posterior predictive distribution of the positive part is known.

Lemma 1. *The posterior predictive distribution of a zero-inflated distribution as defined in equation 3 is given by*

$$p(y_i|\mathbf{y}_{\setminus i}) = \begin{cases} p(y_i|\mathbf{y}^0_{\setminus i}) \cdot \int_q (1-q) \cdot p(q|\mathbf{y}_{\setminus i})\, dq & \textit{if } y_i < \tau \\ p(y_i|\mathbf{y}^1_{\setminus i}) \cdot \int_q q \cdot p(q|\mathbf{y}_{\setminus i})\, dq & \textit{if } y_i \geq \tau \end{cases}$$

where $\mathbf{y}^0_{\setminus i}$ *and* $\mathbf{y}^1_{\setminus i}$ *denote the set of zero and non-zero responses in* $\mathbf{y}_{\setminus i}$*, respectively.*

Proof. See Appendix C. □

This means that the posterior predictive distribution can be computed efficiently for a large class of positive distributions, including the zero-inflated Log-Normal or even zero-inflated Flow models [2]. In the next section, we demonstrate that this Bayesian treatment yields a GS model that is more robust against outliers and yields higher likelihoods than the Null model, as expected (Fig. 1, orange points).

3 Experiments

3.1 Analysis of Gold Standard Models

In this section, we investigate why the Bayesian approach outperforms the PE and test its robustness under different numbers of left-out repeats $\mathbf{y}_{\setminus i}$ and different signal-to-noise ratios. For this analysis, we used responses from 7672 neurons to 360 stimuli, each repeated 20 times as well as simulated data. Details about the datasets are provided in Appendix A.1 and Appendix A.2, respectively. We base our posterior predictive distribution on a zero-inflated Log-Normal distribution with zero-threshold $\tau = \exp(-10)$ and parameters $\theta_1 = (\mu, \sigma^2)$, referring to the first and second moment of the log-transformed positive responses (Appendix C). The conjugate prior for the Log-Normal part of the distribution is a Normal-Inverse-Gamma distribution $p(\theta_1) = \mathcal{N}G^{-1}(\mu, \lambda, \alpha, \beta)$ in log space whose parameters can be chosen freely. We discuss choices of prior parameters later in section 3.2.

Bayesian GS estimates higher order moments better In order to determine which parameters of the likelihood profits most from the Bayesian estimation, we compared GS models where the individual parameters are either estimated via the PE or the Bayesian approach (Fig. 3**a**). For this we used the largest number of left-out repeats $n - 1 = 19$ available in our real neuron dataset. First, we observe that the likelihood improves with the Bayesian estimation of μ (orange vs. yellow bar) as well as σ^2 (light blue vs. yellow bar) individually. Consequently, the highest performance is achieved when both parameters are estimated via the Bayesian approach (dark blue vs. yellow bar). Interestingly, the relative gain in performance is much higher for σ^2 than for the μ, reflecting a lower robustness of the higher moments compared to the first moment in log space.

Bayesian GS is data-efficient Datasets can vary in how many repeats per stimulus they contain. Since a metric should be comparable across datasets, NInGa ought to yield consistent estimates for

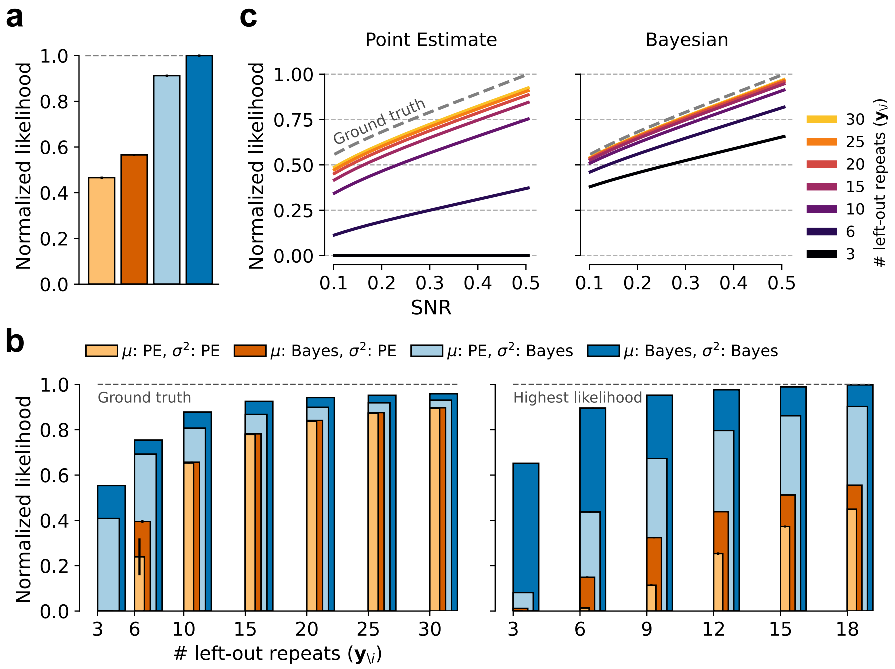


Figure 3: Comparison of the point estimate and Bayesian GS model. **a:** Comparison of different GS models where the individual parameters are either estimated via the PE or the Bayesian approach. The number of left-out repeats $\mathbf{y}_{\setminus i}$ is 19. Colors are the same as in **b**. Normalized wrt. the max. likelihood value, i.e. the dark blue bar. **b:** Similar to **a** but for different numbers of left-out repeats $\mathbf{y}_{\setminus i}$. **Left:** Simulated data. The likelihood is normalized w.r.t. the ground truth likelihood. **Right:** Neural responses. Likelihood is normalized w.r.t. the maximum likelihood value, i.e. the dark blue bar at 18 left-out repeats. **c:** Upper bound likelihood scores for different signal-to-noise ratios and different number of left-out repeats $\mathbf{y}_{\setminus i}$. **Left:** Point Estimate. **Right:** Bayesian. Likelihood is normalized w.r.t. the ground truth likelihood. In all panels the likelihood values are averaged over stimuli and neurons, and the error bars and shaded areas show SEM over 5 random selections of the left-out repeats.

different numbers of left-out repeats $\mathbf{y}_{\setminus i}$, in particular in the regime of low $n-1$. The right panel of Fig. 3**b** shows this on neural data where we first observe that the results of Fig. 3**a** are qualitatively consistent for different numbers of repeats. The effect of the Bayesian parameter estimation on the likelihood performance, however, is much more pronounced in the low $n-1$ regime: As the number of left-out repeats decreases the PE approach suffers much more than the Bayesian and it completely fails at $n=3$ (vanishing yellow bar). To test the higher $n-1$ regime, we simulated neural responses since the real neural dataset contained maximally 20 repeats. In the left panel of Fig. 3**b** we explored the differences between the two approaches for up to 30 repeats and observed that the Bayesian treatment consistently yields a better likelihood than the PE estimate (yellow bar does not completely converge to dark blue bar). The probabilistic treatment of μ, however, seems to become less important in the high $n-1$ regime than that of σ^2, reflecting the higher robustness of the first moment compared to the second moment in log space (compare the difference between orange and yellow for high vs. low $n-1$).

Bayesian GS is robust to different SNRs Apart from different numbers of repeats, datasets can also vary in terms of signal-to-noise ratio. We therefore simulated neural data with different underlying means and variances per stimulus, resulting in different SNR values (see Appendix A.2 for details). We then tested Bayesian and point estimate GS models on this data (Fig. 3 **c**) and observed that the Bayesian approach consistently outperforms the PE approach across all SNRs (data for Fig. 3**b** left panel had an SNR of 0.42).

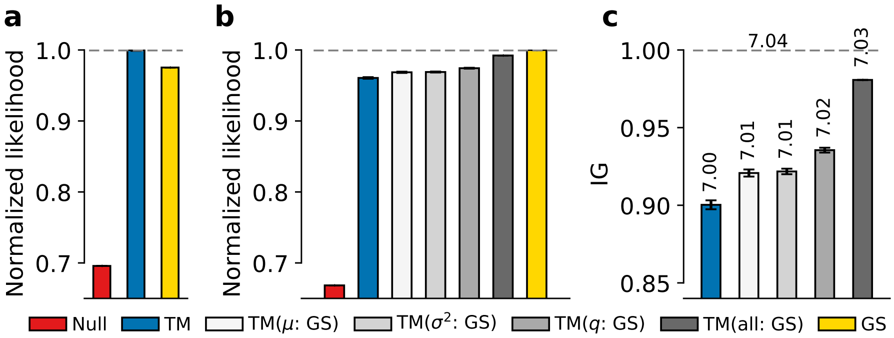


Figure 4: Real neural data evaluation of a neural encoding model trained on a zero-inflated Log-Normal likelihood (ZIL) using the Bayesian upper bound estimate (GS). **a:** A sub-optimal prior for the GS model results in a lower performance than the trained model (TM). Normalized wrt. the max. likelihood value, i.e. the blue bar. **b:** The GS model outperforms the TM when its prior was optimized. The TM estimates all parameters similarly well. Grey scale colors indicate models from the same distribution as the TM (ZIL) for which all parameters are estimated by the trained model except for the parameters in parentheses. Normalized wrt. the max. likelihood value, i.e. the yellow bar. **c:** Normalized Information Gain (NInGa) for the TM and the models grey scale models from **b**. Values on top of bars indicate the likelihood per image and per neuron in bits. In all panels the likelihood values are averaged over stimuli and neurons, and the error bars and shaded areas show SEM over 5 random initializations of the TM weights. There are no errorbars on the red, yellow and dark grey bars since they do not involve a TM.

3.2 Evaluating neural encoding models via NInGa

In Section 3.1 we demonstrated that the Bayesian approach for obtaining an upper bound estimator greatly outperforms the point estimate (PE) approach in several aspects. From now on, we therefore only use the Bayesian estimator when referring to the Gold Standard Model. In this section, we use the lower and upper bound estimates to evaluate neural encoding models via Normalized Information Gain.

To this end, we trained an encoding model on a dataset which is publicly available [16] containing the responses of 5335 neurons in mouse primary visual cortex to 5094 unique natural images. The dataset was split into 4472 stimulus-response pairs for training, 522 for validation, and 100 for testing. The stimuli in the test set were each repeated 10 times resulting in $n - 1 = 9$ repeats for the GS model to be computed on. The neural encoding model and the training are the same as in the model provided by Lurz et al. [16]. Briefly, the encoding model consists of two parts: (1) A core network which is shared across neurons with four convolutional layers (some of them depth-separable [9]) resulting in 64 feature channels, followed by batch normalization and ELU nonlinearity. And (2) a neuron-specific Gaussian readout mechanism [16] 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. While Lurz et al. [16] used Poisson loss to train the models, we chose the negative loglikelihood of a zero-inflated Log-Normal (ZIL) distribution as a loss function. This means that the model needs to predict three parameters (q, μ, σ^2) instead of a single mean firing rate λ as in Lurz et al. [16]. The readout thus learns three weight vectors to combine features extracted via the core network, at neuron's learned RF position. Since the metric that the model will be evaluated on is NInGa, the early-stopping criterion was changed from correlation to likelihood. The results of these experiments with neural encoding models are summarized in Fig. 4 and will be explained in detail below.

Choice of prior is crucial for the GS model Up to this point, we chose the prior hyper-parameters neuron-independently: We chose the parameters of the Normal-Inverse-Gamma prior based on the average conditional mean $\mathrm{E}_x[\mathrm{E}_y[y|x]]$ and average conditional variance $\mathrm{E}_x[\mathrm{Var}_y[y|x]]$ of our dataset. This left us with number of neurons samples which we fit the prior on, resulting in one identical prior for every neuron. However, we observed that the resulting GS model (yellow bar) was outperformed by the trained encoding model (TM model, blue bar) and did not provide a good estimate of a performance upper bound (Fig. 4 **a**). To obtain a better upper bound oracle model, we therefore optimized the prior hyper-parameters directly on the sum of leave-one-out GS likelihoods via gradient descent for each single neuron. This is analogous to other oracle models that have access to more information than the predictive model under evaluation, thus providing an upper bound. Note that this approach results in a more conservative estimate of the model performance (Fig. 4 **b**, compare the blue and yellow bars). We also tried other approaches (e.g. MAP estimate) to obtain a better GS model but none of them outperformed the Posterior Predictive GS (see Appendix I).

Encoding model captures all parameters similarly well In order to investigate which parameters of the response distribution the encoding model predicts well, we conducted an experiment similar to the one shown in Fig. 3 **a**: We compared the likelihood of the trained model (TM, blue) to cases where we matched one or all of its three parameters (q, μ, σ^2) to the GS model (see Fig. 4 **b** blue bar vs. grey bars). While we could match the parameter q directly, we used moment matching to obtain the parameters of the non-zero part of the trained model (Log-Normal distribution) from the non-zero part of the GS model (Log-Student-t). We observed that the likelihood of the trained model improved when each of the three parameters were matched with the GS model individually (three lighter grey bars vs. blue bar), where q yielded the highest increase. Matching all three parameters, however, did not result in the same performance as the GS model itself (darkest grey vs. yellow). Since all parameters of the ZIL model were fitted on the GS model, the reason must be the difference in distributional shape of the positive part: Log-Normal for ZIL vs. Log-Student-t for the GS model.

Encoding model performance is at 90% NInGa The final NInGa value of the trained model using the Null model and the GS model with optimized prior hyper-parameters can be seen in Fig. 4 **c**. It performs remarkably well at 90% NInGa (blue bar), which corresponds to a likelihood of 7.00 bits per image and neuron (printed value above the bar). The effect of the parameter matching (grey bars) is more emphasized and suggests that the largest performance gain can be achieved in future models by improving the prediction of the parameter q. We performed additional analyses on multiple datasets to show how NInGa facilitates model comparison across different datasets (see Appendix J).

4 Relation between NInGa and other metrics

Neural predicting models have so far been mostly evaluated using metrics such as fraction of explainable variance explained (FEVE) [7], correlation, and fraction oracle [19; 16]. While we cannot expect that there is a one-to-one relationship between NInGa and these metrics, as NInGa is sensitive to the entire response distribution whereas the commonly used metrics mostly focus on the mean response of a neuron for a given stimulus, we can nevertheless relate NInGa to these metric under certain assumptions. In this section we provide a summary of the relationships to these metrics. Details and proofs can be found in Appendices F and G.

Generally, we will demonstrate the relations in two steps: First we show that NInGa linearly depends on the expected Kullback-Leibler (KL) divergence between the true distribution $p(y|x)$ and the model distribution $\hat{p}(y|x)$. Then we derive how this KL divergence relates to other metrics.

NInGa is a linear function of $\langle D_{KL}[p(y|x), \hat{p}(y|x)]\rangle_x$ The NInGa described in Eq. 1 can be re-written in terms of KL divergences between estimated/fitted distributions and the true conditional distribution $p(y|x)$ (see Appendix E for the detailed derivation):

$$NInGa = \frac{\langle D_{KL}[p(y|x)||p_0(y)]\rangle_x - \langle D_{KL}[p(y|x)||\hat{p}(y|x)]\rangle_x}{\langle D_{KL}[p(y|x)||p_0(y)]\rangle_x - \langle D_{KL}[p(y|x)||p_*(y|x)]\rangle_x} \tag{4}$$

This shows that the Normalized Information Gain is a linear function of $\langle D_{KL}[p(y|x)||\hat{p}(y|x)]\rangle_x$ with a negative slope. It is maximized when average Kullback-Leibler divergence between the true distribution and the model distribution is minimized.

NInGa vs. FEVE Using this fact, we now show that NInGa is linear in another commonly used metric, fraction of explainable variance explained (FEVE), for a Gaussian likelihood. For a fixed model noise variance $\hat{\sigma}_\epsilon^2$ the KL-divergence $\langle D_{KL}[p(y|x)||\hat{p}(y|x)]\rangle_x$ is a linear function of FEVE:

$$\langle D_{KL}[p(y|x)||\hat{p}(y|x)]\rangle_x = f(\hat{\sigma}_\epsilon) + \frac{1}{2}(1 - FEVE) \times \frac{\sigma_s^2}{\hat{\sigma}_\epsilon^2}, \tag{5}$$

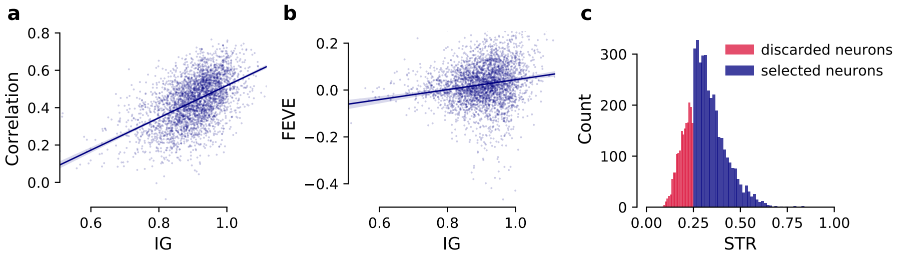


Figure 5: Per neuron comparison of NInGa with correlation and FEVE. The data comes from the TM in Fig. 4 (one of the 5 models from different initializations). **a:** Correlation vs. NInGa. **b:** FEVE vs. NInGa. **c:** The neurons displayed in panel **a** and **b** were selected based on whether their signal-to-total-variance ratio (STR) was above 0.25. The blue lines depict linear regressions with 95% confidence intervals.

where $f(\hat{\sigma}_\epsilon) = \log\left(\frac{\hat{\sigma}_\epsilon}{\sigma_\epsilon}\right) - \frac{1}{2} + \frac{\sigma_\epsilon^2}{2\hat{\sigma}_\epsilon^2}$ and σ_s^2 is the signal variance. Since $\langle D_{KL}[p(y|x)||\hat{p}(y|x)]\rangle_x$ and NInGa are also linearly related, then NInGa too is a linear function of FEVE. Note that when the estimated noise variance is the same as the true noise variance, then Eq. 5 becomes:

$$\langle D_{KL}[p(y|x)||\hat{p}(y|x)]\rangle_x = \frac{1}{2}(1 - FEVE) \times SNR$$

See Appendix F for detailed derivations. If the estimated noise variance $\hat{\sigma}_\epsilon^2$ is not fixed, as is the general case, this trend will still be true but the linear relationship between the NInGa and FEVE is more noisy. In Fig. 5**b**, we show that by evaluating the trained model from Fig. 4 on NInGa and FEVE per neuron. Note that we only display neurons whose signal-to-total-variance ratio $STR = \sigma_s^2/\sigma_y^2 = \sigma_s^2/(\sigma_s^2 + \sigma_\epsilon^2)$ exceeds a threshold which we set to 0.25 (see Fig. 5**c**).

NInGa vs. correlation Another commonly used metric is the correlation $\rho(\hat{\mu}_x, \mu_x)$ between the model prediction $\hat{\mu}_x = \langle\hat{y}\rangle_{\hat{y}|x}$ and trial averaged responses $\mu_x = \langle y\rangle_{y|x}$. Assuming a Gaussian likelihood, like in the relation to FEVE, we show that for a fixed model noise and signal variance $\langle D_{KL}[p(y|x)||\hat{p}(y|x)]\rangle_x$ is a linear function of the correlation between μ_x and $\hat{\mu}_x$:

$$\langle D_{KL}[p(y|x)||\hat{p}(y|x)]\rangle_x = f(\hat{\sigma}_\epsilon) + \frac{1}{2}\left(1 + \frac{\hat{\sigma}_s^2}{\sigma_s^2} - \frac{2\hat{\sigma}_s}{\sigma_s}\rho(\hat{\mu}_x, \mu_x)\right) \times \frac{\sigma_s^2}{\hat{\sigma}_\epsilon^2} \tag{6}$$

As before, if the model noise variance matches the true noise variance, $\sigma_\epsilon^2 = \hat{\sigma}_\epsilon^2$, we have:

$$\langle D_{KL}[p(y|x)||\hat{p}(y|x)]\rangle_x = \frac{1}{2}\left(1 + \frac{\hat{\sigma}_s^2}{\sigma_s^2} - \frac{2\hat{\sigma}_s}{\sigma_s}\rho(\hat{\mu}_x, \mu_x)\right) \times SNR$$

Assuming further that the model's signal variance matches the true signal variance, $\hat{\sigma}_s^2 = \sigma_s^2$, we get:

$$\langle D_{KL}[p(y|x)||\hat{p}(y|x)]\rangle_x = (1 - \rho(\hat{\mu}_x, \mu_x)) \times SNR$$

It is worth noting that the assumptions for having a linear relationship with KL divergence are stronger in the case of correlation than that of the FEVE. Specifically, there is an additional dependence on the signal variance in the case of correlation. This makes sense as FEVE is inherently sensitive to signal variance while correlation is not (see Appendix G for the detailed derivation). As in the case of FEVE, we empirically show (Fig. 5 **a**) the relation between NInGa and correlation on our trained model from Fig. 4. While the general linear trend can be observed, this relation is very noisy, as expected. Note again, that we only show neurons that pass the threshold of $STR \geq 0.25$.

Does a high likelihood always correspond to a high correlation? Correlation only reflects good estimation of the mean of a distribution (see Appendix H). NInGa, on the other hand is a likelihood-based metric and as such depends on the correct estimation of all parameters of the likelihood. So does a high likelihood correspond to a high correlation? The answer is: it depends.

Correlation is high if the conditional mean is accurately estimated. However, since the mean of the distribution is in general not directly a parameter of the likelihood it is not necessarily estimated

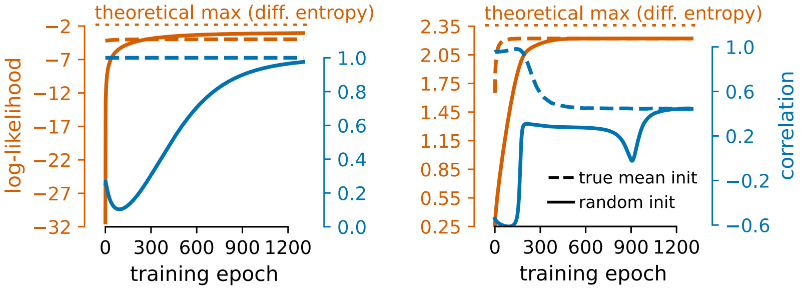


Figure 6: Comparison of log-likelihood and correlation of a model which is optimized via maximum likelihood estimation. **Left:** The model distribution is from the exponential family and has the mean as a sufficient statistic. The optima of the two metrics coincide but an improvement in likelihood does not necessarily mean an improvement in correlation. Data distribution: Normal. Model distribution: Gamma. **Right:** The model distribution does *not* fulfill these criteria. No similar behavior of likelihood and correlation can be guaranteed. If initialized with optimal correlation, the correlation after training has decreased (dotted line). Data distribution: Gamma. Model distribution: Chi-squared.

accurately even if optimal likelihood is achieved. Therefore, there is in general no guarantee that a model that outperforms another model on likelihood also outperforms this other model on correlation. However, if the distribution is 1) a member of the exponential family and 2) has the mean as a sufficient statistic [10], an optimal likelihood also implies an optimal correlation. A distribution that fulfills these two criteria guarantees that for optimal likelihood, the mean is optimally estimated and with it the correlation. We demonstrate this effect using simulated data (see Appendix A.2) in Fig. 6 in the left panel: A model of such a distribution is being optimized on toy data via maximum likelihood estimation and at the end of training the optima of likelihood and correlation coincide. Note however that during training, i.e. for non-optimal likelihood, a training step which improves likelihood does not necessarily yield an improvement of correlation (compare decreasing blue line vs. increasing orange line). If the distribution does not fulfill the second criterion, i.e. if the mean is not a sufficient statistic, no relation between the correlation and likelihood can be guaranteed, not even at the optimum. This can be seen in Fig. 6 in the right panel: An improvement in likelihood (orange) does not guarantee an improvement in correlation (blue). If the model was initialized such that the correlation is optimal (dotted lines), the maximum likelihood estimation will even result in a model that has lower correlation than it had at the time of initialization (dotted blue line at epoch 0 vs 1200).

5 Conclusion

In this work, we discussed the challenges in obtaining the lower and upper bound estimates for likelihood-based measures like Normalized Information Gain (NInGa) in order to put the performance on an interpretable scale. We introduced a robust way of obtaining such an upper bound by using the Bayesian framework of posterior predictive distributions. Equipped with this metric, we showed that current neural encoding models are able to predict full response distributions to up to 90% NInGa and we examined which parameters of the distribution the model still fails at predicting. We also gave a detailed derivation for obtaining upper bound estimates for the state-of-the-art family of distributions, the zero-inflated distributions. Finally, we related likelihood-based metrics like NInGa to other metrics which are commonly used in Computational Neuroscience like correlation and FEVE.

There are, of course, also some limitations to the current work. While we were able to fix the catastrophic failure of naive PE estimates (Fig. 1), our estimates of NInGa or the GS model are not perfect as some neurons yield NInGa > 1 (Fig. 5) which is not mathematically impossible but would ideally not happen. Better NInGa or GS estimators will mitigate this issue. Finally, our approach is for single neuron likelihoods. However, the neural variation is structured on a population level, for instance through brain states. Since the general approach of NInGa still works in that situation, our framework is flexible enough for a robust GS model of full populations to be derived in future work.

Acknowledgments

We thank all reviewers for their constructive and thoughtful feedback.

Konstantin-Klemens Lurz is funded by the German Federal Ministry of Education and Research through the Tübingen AI Center (FKZ: 01IS18039A). Mohammad Bashiri is supported by the International Max Planck Research School for Intelligent Systems. Fabian H. Sinz 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 the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DoI/IBC) contract number D16PC00003. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/IBC, or the U.S. Government.

References

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

[2] Mohammad Bashiri, Edgar Walker, Konstantin-Klemens Lurz, Akshay Jagadish, Taliah Muhammad, Zhiwei Ding, Zhuokun Ding, Andreas Tolias, and Fabian Sinz. A flow-based latent state generative model of neural population responses to natural images. *Advances in Neural Information Processing Systems*, 34:15801–15815, 2021.

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

[4] Pietro Berkes, Gergő Orbán, Máté Lengyel, and József Fiser. Spontaneous cortical activity reveals hallmarks of an optimal internal model of the environment. *Science*, 331(6013):83–87, 2011.

[5] Adrian G Bondy, Ralf M Haefner, and Bruce G Cumming. Feedback determines the structure of correlated variability in primary visual cortex. *Nature neuroscience*, 21(4):598–606, 2018.

[6] Max F Burg, Santiago A Cadena, George H Denfield, Edgar Y Walker, Andreas S Tolias, Matthias Bethge, and Alexander S Ecker. Learning divisive normalization in primary visual cortex. *PLOS Computational Biology*, 17(6):e1009028, 2021.

[7] 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.

[8] Santiago A Cadena, Konstantin F Willeke, Kelli Restivo, George Denfield, Fabian H Sinz, Matthias Bethge, Andreas S Tolias, and Alexander S Ecker. Diverse task-driven modeling of macaque v4 reveals functional specialization towards semantic tasks. *bioRxiv*, 2022.

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

[10] Justin Domke. Moment-matching conditions for exponential families with conditioning or hidden data. *arXiv preprint arXiv:2001.09771*, 2020.

[11] Alexander S Ecker, Fabian H Sinz, Emmanouil Froudarakis, Paul G Fahey, Santiago A Cadena, Edgar Y Walker, Erick Cobos, Jacob Reimer, Andreas S Tolias, and Matthias Bethge. A rotation-equivariant convolutional neural network model of primary visual cortex. *arXiv preprint arXiv:1809.10504*, 2018.

[12] József Fiser, Pietro Berkes, Gergő Orbán, and Máté Lengyel. Statistically optimal perception and learning: from behavior to neural representations. *Trends in cognitive sciences*, 14(3): 119–130, 2010.

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

[14] Matthias Kümmerer, Thomas SA Wallis, and Matthias Bethge. Information-theoretic model comparison unifies saliency metrics. *Proceedings of the National Academy of Sciences*, 112 (52):16054–16059, 2015.

[15] Richard D Lange, Ankani Chattoraj, Jeffrey M Beck, Jacob L Yates, and Ralf M Haefner. A confirmation bias in perceptual decision-making due to hierarchical approximate inference. *PLoS Computational Biology*, 17(11):e1009517, 2021.

[16] 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.

[17] W J Ma, J M Beck, P E Latham, and A Pouget. Bayesian inference with probabilistic population codes. *Nat. Neurosci.*, 9:1432–1438, 2006.

[18] Wei Ji Ma, Jeffrey M Beck, Peter E Latham, and Alexandre Pouget. Bayesian inference with probabilistic population codes. *Nature neuroscience*, 9(11):1432–1438, 2006.

[19] 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. *Advances in neural information processing systems*, 31, 2018.

[20] Lucas Theis, Andrè Maia Chagas, Daniel Arnstein, Cornelius Schwarz, and Matthias Bethge. Beyond glms: a generative mixture modeling approach to neural system identification. *PLoS computational biology*, 9(11):e1003356, 2013.

[21] 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.

[22] 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. *arXiv preprint arXiv:2006.03737*, 2020.

[23] Konstantin F Willeke, Paul G Fahey, Mohammad Bashiri, Laura Pede, Max F Burg, Christoph Blessing, Santiago A Cadena, Zhiwei Ding, Konstantin-Klemens Lurz, Kayla Ponder, et al. The sensorium competition on predicting large-scale mouse primary visual cortex activity. *arXiv preprint arXiv:2206.08666*, 2022.

[24] Daniel LK Yamins, Ha Hong, Charles F Cadieu, Ethan A Solomon, Darren Seibert, and James J DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the national academy of sciences*, 111(23):8619–8624, 2014.

A Data

A.1 Neural data

Data from real neural activity is used in Fig. 1, in Fig. 3 panel **a** and panel **b** on the right side, and in Fig. 4. Responses were obtained via two-photon calcium imaging of layer L2/3 of the primary visual cortex (area V1) of the mouse. Recordings, experimental paradigm and pre-processing was similar to [16]. The data for Fig. 1 and Fig. 3 consists of the responses of 7672 neurons to 360 images where each image was presented 20 times. Since 7 trials were missing, this makes for a total of 7193 trials per neuron. The data for Fig. 4 consists of the responses of 5335 neurons to 4472 images in the training set, 522 images in the validation set and 100 images in the test set. The images of the test set were repeated 10 times which makes for 999 test trials per neuron since one trial was missing.

A.2 Simulated neural data

Simulated neural data is used in Fig. 3 in the left part of panel **b** and both parts of panel **c**. We generated samples for 100 neurons, 360 stimuli, and 31 repeats per stimulus. Briefly, we simulated the data assuming a zero-inflated Log-Normal distribution where the parameters μ and σ^2 of the Log-Normal part are normal-gamma distributed. The complete description of the simulation is as follows:

$$\begin{aligned}
y &\sim ZIL(\mu, \sigma^2, q, \tau)\\
\mu &\sim \mathcal{N}(\mu_\mu, \sigma^2/\nu)\\
\sigma^2 &\sim Gamma(\alpha_{\sigma^2}, \beta_{\sigma^2})\\
q &\sim Beta(21, 117)\\
\tau &= \exp(-10)\\
\nu &\sim Gamma(8.29, 7.32)\\
\alpha_{\sigma^2} &\sim Gamma(27.81, 0.8)\\
\beta_{\sigma^2} &= \alpha_{\sigma^2}/\overline{\sigma}^2_{noise}\\
\begin{bmatrix}\mu_\mu\\ \overline{\sigma}^2_{noise}\end{bmatrix} &\sim \mathcal{N}\left(\begin{bmatrix}-3.13\\ 0.36\end{bmatrix}, \begin{bmatrix}0.158 & -0.017\\ -0.017 & 0.003\end{bmatrix}\right)
\end{aligned}$$

The parameter values were chosen such that the resulting simulated data resembles the real neural activity.

Simulating data with different SNRs In order to simulate data with different SNR values, we first generated samples y as described above and then transformed the data into the log space $z = \log(y)$. Next, to extract the noise we subtracted the mean per stimulus, scaled the noise and then added the mean back. This results in a set of samples where the mean stays the same while the noise level has been scaled. Finally, we transformed the data back into the original space by applying the exp function on the resulting samples:

$$y = \exp\left((z - \overline{z}) * c + \overline{z}\right),$$

where $\overline{z}$ is the average across repeats and c is the scaling factor for the noise across repeats. The SNR of the (simulated) responses is computed as $\frac{\text{Var}_x(\mathbb{E}_y[y|x])}{\mathbb{E}_x[\text{Var}_y(y|x)]}$ where $\text{Var}_x(\mathbb{E}_y[y|x])$ is the variance of averaged responses and $\mathbb{E}_x[\text{Var}_y(y|x)]$ is the average noise level.

Data for comparison of likelihood and correlation The artificial data used in Fig. 6 is not intended to simulate accurate patterns of neural activity. It is thus simply drawn from a normal and a gamma distribution, in the left and right plot respectively. We sampled 5000 train and 500 test repeats to three "stimuli" for two "neurons", resulting in three 2D distributions. In the case of gamma data (left plot):

$$\begin{bmatrix} y_{11} \\ y_{12} \end{bmatrix} \sim Gamma\left(\begin{bmatrix} .56 \\ .43 \end{bmatrix}, \begin{bmatrix} .28 \\ .34 \end{bmatrix}\right)$$
$$\begin{bmatrix} y_{21} \\ y_{22} \end{bmatrix} \sim Gamma\left(\begin{bmatrix} .32 \\ .27 \end{bmatrix}, \begin{bmatrix} .13 \\ .39 \end{bmatrix}\right)$$
$$\begin{bmatrix} y_{31} \\ y_{32} \end{bmatrix} \sim Gamma\left(\begin{bmatrix} .45 \\ .34 \end{bmatrix}, \begin{bmatrix} .35 \\ .37 \end{bmatrix}\right)$$

And in the case of normal data (right plot):

$$\begin{bmatrix} y_{11} \\ y_{12} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 15. \\ 21. \end{bmatrix}, \begin{bmatrix} 1. & 0. \\ 0. & 1. \end{bmatrix}\right)$$
$$\begin{bmatrix} y_{21} \\ y_{22} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 16.8 \\ 16.3 \end{bmatrix}, \begin{bmatrix} 4.2 & 0. \\ 0. & .5 \end{bmatrix}\right)$$
$$\begin{bmatrix} y_{31} \\ y_{32} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 21.0 \\ 10.1 \end{bmatrix}, \begin{bmatrix} 2.5 & 0. \\ 0. & 5.3 \end{bmatrix}\right)$$

The fitted likelihoods are a gamma and a χ^2 distribution, respectively.

B MOMENT MATCHING FOR ZERO-INFLATED LIKELIHOOD

In this section we demonstrate how to compute the moments of each component of a zero-inflated mixture model. Since the uniform zero part does not have any parameters, we express the moments of the non-zero part as a function of the moments of the entire data, under the assumption of a zero-inflated distribution. We then use those for moment-matching the parameters of the non-zero part. The detailed step-by-step derivation is as follows:

We first express the total mean μ_{total} and total variance σ^2_{total} in terms of the means and variances of each component of the mixture model. We then solve for mean μ_1 and variance σ_1^2 of the positive distribution:

$$\mu_{total} = \mathbb{E}_y[y] = (1-q)\cdot\mu_0 + q\cdot\mu_1$$

To compute the total variance we make use of the law of total variance $\text{Var}_y(y) = \mathbb{E}_m[\text{Var}_{y|m}(y)] + \text{Var}_m(\mathbb{E}_{y|m}[y])$ where

$$\begin{aligned}\mathbb{E}_m[\text{Var}_{y|m}(y)] &= \mathbb{E}_m[\{\text{Var}_{y|m=0}(y), \text{Var}_{y|m=1}(y)\}]\\ &= (1-q)\cdot\sigma_0^2 + q\cdot\sigma_1^2\end{aligned}$$

and

$$\begin{aligned}\text{Var}_m(\mathbb{E}_{y|m}[y]) &= \mathbb{E}_m[\mathbb{E}_{y|m}[y]^2] - \mathbb{E}_m[\mathbb{E}_{y|m}[y]]^2\\ &= \mathbb{E}_m[\{\mathbb{E}_{y|m=0}[y]^2, \mathbb{E}_{y|m=1}[y]^2\}] - \mathbb{E}_m[\{\mathbb{E}_{y|m=0}[y], \mathbb{E}_{y|m=1}[y]\}]^2\\ &= (1-q)\cdot\mu_0^2 + q\cdot\mu_1^2 - ((1-q)\cdot\mu_0 + q\cdot\mu_1)^2\\ &= (1-q)\cdot\mu_0^2 + q\cdot\mu_1^2 - (1-q)^2\cdot\mu_0^2 - q^2\cdot\mu_1^2 - 2q(1-q)\cdot\mu_0\mu_1\\ &= ((1-q)-(1-q)^2)\cdot\mu_0^2 + (q-q^2)\cdot\mu_1^2 - 2q(1-q)\cdot\mu_0\mu_1\\ &= q(1-q)\cdot\mu_0^2 + q(1-q)\cdot\mu_1^2 - 2q(1-q)\cdot\mu_0\mu_1\\ &= q(1-q)\cdot(\mu_0-\mu_1)^2.\end{aligned}$$

Notation $\mathbb{E}[\{a, b, ...\}]$ is used to denote that the expectation involves the terms in the set $\{a, b, ...\}$. The total variance can then be computed as

$$\begin{aligned}\sigma^2_{total} = \text{Var}_y[y] &= \mathbb{E}_m[\text{Var}_{y|m}(y) + \text{Var}_m(\mathbb{E}_{y|m}[y])\\ &= (1-q)\cdot\sigma_0^2 + q\cdot\sigma_1^2 + q(1-q)\cdot(\mu_0-\mu_1)^2\end{aligned}$$

The mean and variance of the non-zero part can thus be computed as

$$\begin{aligned}\mu_1 &= \frac{\mu_{total} - (1-q)\cdot\mu_0}{q}\\ \sigma_1^2 &= \frac{\sigma^2_{total} - (1-q)\cdot\sigma_0^2 - q(1-q)(\mu_0-\mu_1)^2}{q}\end{aligned}$$

The parameters of the non-zero part can then be obtained by moment matching with μ_1 and σ_1^2. Note, however, that the mean of the non-zero distribution is not μ_1 itself but $\mu_1 - \tau$. In a case where there are no responses above the zero-threshold τ, μ_1 and σ_1^2 are not defined because of the denominator $q = 0$. In this case we assign a small value of 0.1 to the mean and 0.3 to the variance. We chose these values because they resulted in the best GS model performance for the PE approach.

B.1 ZERO-INFLATED LOG-NORMAL LIKELIHOOD

In the case of a Log-Normal non-zero part, the parameters μ_{LogN} and σ^2_{LogN} evaluate to

$$\begin{aligned}\mu_{LogN} &= \log\left(\frac{\mu_1-\tau}{\sqrt{\frac{\sigma_1^2}{(\mu_1-\tau)^2}+1}}\right)\\ \sigma^2_{LogN} &= \log\left(\frac{\sigma_1^2}{(\mu_1-\tau)^2}+1\right)\end{aligned}$$

Note that μ_0, μ_1, σ_0^2 and σ_1^2 are the means and variances of the zero and non-zero part of the distribution, respectively. The parameters μ_{LogN} and σ^2_{LogN} are *not* the mean and variance of the Log-Normal distribution but of the underlying Normal distribution in log space.

C Posterior Predictive for zero-inflated likelihood

Our goal is to probabilistically infer the parameters of the distribution per image, in a leave-one-out manner. That is, to compute $p(y_i|\mathbf{y}_{\setminus i}, x)$. For brevity, we drop the conditioning on x in the following derivations. Following the graphical model in Fig. 2, let's define some of the density functions that will be used later on:

$$\begin{aligned}
p(y, \theta, m, q) &= p(y|\theta, m)p(m|q)p(\theta)p(q) \\
p(m|q) &= q^m \cdot (1-q)^{1-m} \\
p(y|\theta, m) &= p(y|\theta_0)^{1-m} \cdot p(y|\theta_1)^m \\
p(q) &= q^{\alpha-1} \cdot (1-q)^{\beta-1} \cdot \frac{1}{\mathrm{B}(\alpha, \beta)}
\end{aligned}$$

Marginalizing over m:

$$\begin{aligned}
p(y, \theta, q) &= \sum_{m \in \{0,1\}} p(y, \theta, m, q) \\
&= p(\theta)p(q) \sum_{m \in \{0,1\}} p(y|\theta, m)p(m|q) \\
&= p(\theta)p(q) \left[p(y|\theta, m=0)p(m=0|q) + p(y|\theta, m=1)p(m=1|q)\right] \\
&= p(\theta)p(q) \left[p(y|\theta_0)(1-q) + p(y|\theta_1) \cdot q\right] \\
&= p(\theta)p(q)p(y|\theta, q)
\end{aligned}$$

Our goal is to compute the posterior predictive distribution $p(y_i|\mathbf{y}_{\setminus i})$:

$$\begin{aligned}
p(y_i|\mathbf{y}_{\setminus i}) &= \int_{\theta,q} p(y_i, \theta, q|\mathbf{y}_{\setminus i}) d\theta dq \\
&= \int_{\theta,q} \underbrace{p(y_i|\theta, q, \mathbf{y}_{\setminus i})}_{=p(y_i|\theta,q) \text{ since } y_i \perp\!\!\!\perp \mathbf{y}_{\setminus i}|\theta,q} p(\theta, q|\mathbf{y}_{\setminus i}) d\theta dq \\
&= \int_{\theta,q} \underbrace{p(y_i|\theta, q)}_{\text{likelihood}} \underbrace{p(\theta, q|\mathbf{y}_{\setminus i})}_{\text{posterior}} d\theta dq
\end{aligned}$$

Let us now compute the quantities we need for the posterior predictive $p(y_i|\mathbf{y}_{\setminus i})$, for a single neuron and a single image.

We know the likelihood: $p(y|\theta, q) = (1-q) \cdot p(y|\theta_0) + q \cdot p(y|\theta_1)$. Since the two distributions of our mixture model are not overlapping we can re-write the likelihood as follows:

$$p(y|\theta, q) = \begin{cases} (1-q) \cdot p(y|\theta_0) & \text{if } y \leq \tau \ (m=0) \\ q \cdot p(y|\theta_1) & \text{otherwise } (m=1) \end{cases}$$

The posterior can be derived as follows:

$$\begin{aligned}
p(\theta, q|\mathbf{y}_{\setminus i}) &\propto p(\mathbf{y}_{\setminus i}|\theta, q)p(\theta)p(q) \\
&\propto \left(p(\theta_0) \cdot \prod_{y_j \in \mathbf{y}^0_{\setminus i}} (1-q) \cdot p(y_j|\theta_0)\right) \cdot \left(p(\theta_1) \cdot \prod_{y_j \in \mathbf{y}^1_{\setminus i}} q \cdot p(y_j|\theta_1)\right) \cdot p(q) \\
&\propto \left(p(\theta_0) \cdot \prod_{y_j \in \mathbf{y}^0_{\setminus i}} p(y_j|\theta_0)\right) \cdot \left(p(\theta_1) \cdot \prod_{y_j \in \mathbf{y}^1_{\setminus i}} p(y_j|\theta_1)\right) \cdot (1-q)^{n_0} \cdot q^{n_1} \cdot p(q) \\
&\propto p(\theta_0)p(\mathbf{y}^0_{\setminus i}|\theta_0) \cdot p(\theta_1)p(\mathbf{y}^1_{\setminus i}|\theta_1) \cdot (1-q)^{n_0} \cdot q^{n_1} \cdot q^{\alpha-1} \cdot (1-q)^{\beta-1} \cdot \frac{1}{\mathrm{B}(\alpha, \beta)} \\
&\propto p(\theta_0)p(\mathbf{y}^0_{\setminus i}|\theta_0) \cdot p(\theta_1)p(\mathbf{y}^1_{\setminus i}|\theta_1) \cdot (1-q)^{n_0+\beta-1} \cdot q^{n_1+\alpha-1} \cdot \frac{1}{\mathrm{B}(\alpha, \beta)},
\end{aligned}$$

where $\mathbf{y}^0_{\setminus i}$ are the zero responses, $\mathbf{y}^1_{\setminus i}$ are the positive responses, and n_0 and n_1 are the number of zero and positive responses, respectively. Since the joint distribution factorizes, the whole posterior factorizes (because it is just a re-scaled version of the joint). Normalizing each factor by its own constant, respectively, we get:

$$\begin{aligned} p(\theta, q|\mathbf{y}_{\setminus i}) &= \frac{p(\theta_0)p(\mathbf{y}^0_{\setminus i}|\theta_0)}{Z_1} \cdot \frac{p(\theta_1)p(\mathbf{y}^1_{\setminus i}|\theta_1)}{Z_2} \cdot \frac{(1-q)^{n_0+\beta-1} \cdot q^{n_1+\alpha-1}}{\mathrm{B}(n_1+\alpha, n_0+\beta)} \\ &= p(\theta_0|\mathbf{y}^0_{\setminus i}) \cdot p(\theta_1|\mathbf{y}^1_{\setminus i}) \cdot \mathrm{Beta}(n_1+\alpha, n_0+\beta) \end{aligned} \tag{7}$$

Note that in the case of the posterior over q since the distribution takes the form of a beta distribution we can simply adjust the denominator to the appropriate normalization factor for a beta distribution $B(n_1+\alpha, n_0+\beta)$.

Let us now combine these two components of the posterior predictive to compute $p(y_i|\mathbf{y}_{\setminus i})$:

$$\begin{aligned} p(y_i|\mathbf{y}_{\setminus i}) &= \int_{\theta,q} p(y_i|\theta, q)p(\theta, q|\mathbf{y}_{\setminus i})\, d\theta dq \\ &= \int_{\theta,q} p(y_i|\theta, q)p(\theta_0|\mathbf{y}^0_{\setminus i})p(\theta_1|\mathbf{y}^1_{\setminus i})p(q|\mathbf{y}_{\setminus i})\, d\theta dq \\ &= \int_q \underbrace{\left(\int_\theta p(y_i|\theta, q)p(\theta_0|\mathbf{y}^0_{\setminus i})p(\theta_1|\mathbf{y}^1_{\setminus i})\, d\theta\right)}_{=p(y_i|q,\mathbf{y}_{\setminus i})} p(q|\mathbf{y}_{\setminus i})\, dq \\ &= \int_q p(y_i|q, \mathbf{y}_{\setminus i})p(q|\mathbf{y}_{\setminus i})\, dq \end{aligned} \tag{8}$$

The posterior predictive can then be evaluated depending on whether the target response y_i is below the zero-threshold τ or above it:

If $y_i < \tau$:

$$\begin{aligned} p(y_i|\mathbf{y}_{\setminus i}) &= \int_q \int_\theta p(y_i|\theta, q)p(\theta_0|\mathbf{y}^0_{\setminus i})p(\theta_1|\mathbf{y}^1_{\setminus i})\, d\theta\; p(q|\mathbf{y}_{\setminus i})\, dq \\ &= \int_q \int_\theta p(y_i|\theta_0, q)p(\theta_0|\mathbf{y}^0_{\setminus i})p(\theta_1|\mathbf{y}^1_{\setminus i})\, d\theta\; p(q|\mathbf{y}_{\setminus i})\, dq \\ &= \int_q \int_{\theta_0} p(y_i|\theta_0, q)p(\theta_0|\mathbf{y}^0_{\setminus i})\, d\theta_0 \underbrace{\int_{\theta_1} p(\theta_1|\mathbf{y}^1_{\setminus i})\, d\theta_1}_{=1} p(q|\mathbf{y}_{\setminus i})\, dq \\ &= \int_q \int_{\theta_0} p(y_i|\theta_0, q)p(\theta_0|\mathbf{y}^0_{\setminus i})\, d\theta_0\; p(q|\mathbf{y}_{\setminus i})\, dq \\ &= \int_q p(y_i|q, \mathbf{y}^0_{\setminus i})\; p(q|\mathbf{y}_{\setminus i})\, dq \\ &= \int_q (1-q) \cdot p(y_i|\mathbf{y}^0_{\setminus i})\; p(q|\mathbf{y}_{\setminus i})\, dq \\ &= p(y_i|\mathbf{y}^0_{\setminus i}) \int_q (1-q) \cdot p(q|\mathbf{y}_{\setminus i})\, dq \end{aligned}$$

And if $y_i \geq \tau$:

$$\begin{aligned} p(y_i|\mathbf{y}_{\setminus i}) &= \int_q \int_{\theta_1} p(y_i|\theta_1, q)p(\theta_1|\mathbf{y}^1_{\setminus i})\, d\theta_1\, p(q|\mathbf{y}_{\setminus i})\, dq \\ &= \int_q p(y_i|q, \mathbf{y}^1_{\setminus i})\, p(q|\mathbf{y}_{\setminus i})\, dq \\ &= \int_q q \cdot p(y_i|\mathbf{y}^1_{\setminus i})\, p(q|\mathbf{y}_{\setminus i})\, dq \\ &= p(y_i|\mathbf{y}^1_{\setminus i}) \int_q q \cdot p(q|\mathbf{y}_{\setminus i})\, dq \end{aligned}$$

This means that depending on the target response y_i we either need to compute the posterior predictive of the zero distribution (i.e., Uniform) or positive distribution (i.e., Log-Normal).

Finally, the complete posterior predictive distribution is estimated via numerical integration over q. Numerical integration in this particular case is feasible since q only takes values between 0 and 1.

C.1 Zero-inflated Log-Normal likelihood

We now apply the generic derivation in the previous section to zero-inflated Log-Normal distribution and derive the posterior predictive distribution for it. Let us start by assuming that the target response y_i is below the zero-threshold τ. In this case, the response falls into the Uniform distribution whose parameters are fixed and do not depend on the other zero responses. Therefore, the posterior predictive stays a uniform distribution: $p(y_i|\mathbf{y}_{\setminus i}) = 1/\tau$.

Alternatively, the target response y_i could be higher than the zero-threshold τ falling into the Log-Normal distribution. In this case, we first transform the responses via the $\log$ function into the Gaussian space, then compute the posterior predictive distribution, and finally normalize the resulting distribution to go back into the log space:

$$\begin{aligned} p(y_i|\mathbf{y}_{\setminus i}) &= p(\log(y_i)|\log(\mathbf{y}_{\setminus i})) \cdot |\det \nabla_{y_i} \exp(y_i)| \\ &= p(\log(y_i)|\log(\mathbf{y}_{\setminus i})) \cdot \frac{1}{y_i} \end{aligned} \tag{9}$$

We now focus on computing the posterior predictive in the Gaussian space. For brevity let us assign $\log(y)$ to a new variable $z = \log(y)$. To compute the posterior predictive distribution we need to specify a prior over our likelihood parameters, in this case μ and σ^2. For a Gaussian distribution with unknown μ and σ^2 the conjugate prior is the Normal-inverse gamma distribution with parameters μ_0, ν, α, and β. These parameters are estimated form the data. Once the prior parameters are known, we can then compute the posterior predictive distribution, which is a t-distribution in the case of a Gaussian likelihood:

$$p(z_i|z_{\setminus i}) = t_{2\alpha'}\left(z_i|\mu^{'}, \frac{\beta^{'}(\nu^{'}+1)}{\nu^{'}+\alpha^{'}}\right), \tag{10}$$

where

$$\begin{aligned} \mu^{'} &= \frac{\nu\mu_0 + n\bar{z}_{\setminus i}}{\nu + n} \\ \nu^{'} &= \nu + n \\ \alpha^{'} &= \alpha + \frac{n}{2} \\ \beta^{'} &= \beta + \frac{1}{2}\sum_{z_j \in z_{\setminus i}}^{n} (z_j - \bar{z}_{\setminus i})^2 + \frac{n\nu(\bar{z}_{\setminus i} - \mu_0)^2}{2(\nu + n)} \end{aligned}$$

with n being the number of left-out repeats $z_{\setminus i}$ and $\bar{z}_{\setminus i}$ being the mean of the left-out repeats. As the final step, to compute the posterior predictive in the original $\log$ space, we plug Eq. 10 back into Eq. 9:

$$p(y_i|q, \mathbf{y}_{\setminus i}) = t_{2\alpha'}\left(\log(y_i)|\mu^{'}, \frac{\beta^{'}(\nu^{'}+1)}{\nu^{'}+\alpha^{'}}\right)\cdot\frac{1}{y_i}$$

D In which cases does the Bayesian GS outperform the PE

The PE approach to obtain a Gold Standard model fails in many cases which is the reason behind using the Bayesian Posterior Predictive approach instead. Fig. S1 provides insight into why and in which cases the PE fails compared to the Bayesian GS. In summary: This is due to the combination of low-valued responses that fall into the positive distribution and the sparsity in the data:
Target responses that are slightly greater than the threshold value are not covered by the uniform distribution of the zero part but are an extreme value for the positive part (left panel). When this coincides with overfitting due to sparse data, i.e. low proportion q of positive responses (right panel), the Point Estimate results in a low log-likelihood. Note that the reason for this not being visible for the smallest value of q ($q = 0$) is that in this case no positive responses were available to estimate the PE parameters on. Since the target trial could still be positive, we needed to assign the PE parameters of the positive part of the distribution as a hyper parameter. This is equivalent to applying a delta-peak prior and results in a quasi-Bayesian approach for the PE in these rare cases.

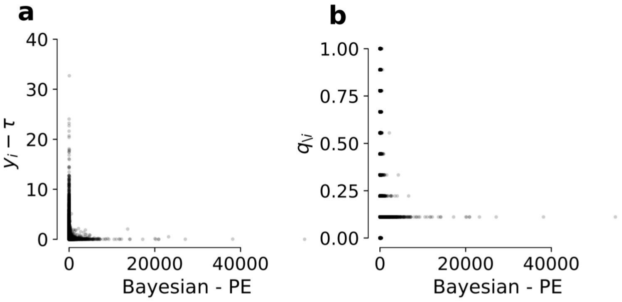


Fig. S1: Comparison between Bayesian and Point Estimate (PE) Gold Standard models. Since the two GS models share the same zero distribution, this analysis was only performed on the responses that fall into the positive distribution ($y \geq \tau$). **a:** Distance of the positive response from the zero-threshold τ as a function of the difference between Bayesian and the PE GS models. **b:** Fraction $q_{\backslash i}$ of positive leave-one-out responses $\mathbf{y}_{\backslash i}$ as a function of the difference between Bayesian and PE GS models. Data is per neuron, per repeat, and per stimulus).

E Normalized Information Gain in terms of KL Divergences

Here we provide the normalized information gain formulated in terms of KL divergence and derive the estimate presented in Eq.1:

$$
\begin{aligned}
\text{NInGa} =& \frac{\langle D_{KL}\left[p(y \mid x) \| p_0(y)\right]\rangle_x - \langle D_{KL}[p(y \mid x) \| \hat{p}(y \mid x)]\rangle_x}{\langle D_{KL}\left[p(y \mid x) \| p_0(y)\right]\rangle_x - \langle D_{KL}\left[p(y \mid x) \| p_*(y \mid x)\right]\rangle_x} \\
=& \frac{\left\langle\left\langle \log \frac{p(y|x)}{p_0(y)}\right\rangle_{y|x}\right\rangle_x - \left\langle\left\langle \log \frac{p(y|x)}{\hat{p}(y|x)}\right\rangle_{y|x}\right\rangle_x}{\left\langle\left\langle \log \frac{p(y|x)}{p_0(y)}\right\rangle_{y|x}\right\rangle_x - \left\langle\left\langle \log \frac{p(y|x)}{p_*(y|x)}\right\rangle_{y|x}\right\rangle_x} \\
=& \frac{\left\langle \langle \log \hat{p}(y \mid x)\rangle_{y|x}\right\rangle_x - \left\langle \langle \log p_0(y)\rangle_{y|x}\right\rangle_x}{\left\langle \langle \log p_*(y \mid x)\rangle_{y|x}\right\rangle_x - \left\langle \langle \log p_0(y)\rangle_{y|x}\right\rangle_x} \\
\approx& \frac{\sum_i \left(\log \hat{p}(y_i \mid x_i) - \log p_0(y_i)\right)}{\sum_i \left(\log p_*(y_i \mid x_i) - \log p_0(y_i)\right)}
\end{aligned}
$$

F Relation between Normalized Information Gain and FEVE

Here we go through the complete derivation underlying Eq. 5. Let us start by defining FEVE:

$$FEVE = 1 - \frac{\left\langle (\mu_x - \hat{\mu}_x)^2 \right\rangle_x}{\sigma_s^2}, \tag{11}$$

where μ_x is the true conditional mean, $\hat{\mu}_x$ the estimated conditional mean by the model, and σ_s^2 is the signal variance. An estimator of FEVE was previously used by Cadena et al. [7] (which we also use to compute FEVE in Fig. 5**b**):

$$\begin{aligned}
FEVE &= 1 - \frac{\left\langle (\mu_x - \hat{\mu}_x)^2 \right\rangle_x}{\sigma_s^2} \\
&= 1 - \frac{\left\langle (\mu_x - \hat{\mu}_x)^2 \right\rangle_x}{\sigma_y^2 - \sigma_\epsilon^2} \\
&= 1 - \frac{\sigma_\epsilon^2 + \left\langle (\mu_x - \hat{\mu}_x)^2 \right\rangle_x - \sigma_\epsilon^2}{\sigma_y^2 - \sigma_\epsilon^2} \\
&= 1 - \frac{\left\langle (y - \mu_x)^2 \right\rangle_{x,y} + \left\langle (\mu_x - \hat{\mu}_x)^2 \right\rangle_x - \overbrace{2\left\langle (y - \mu_x)(\mu_x - \hat{\mu}_x) \right\rangle_{x,y}}^{=0} - \sigma_\epsilon^2}{\sigma_y^2 - \sigma_\epsilon^2} \\
&= 1 - \frac{\left\langle (y - \mu_x + \mu_x - \hat{\mu}_x)^2 \right\rangle_{x,y} - \sigma_\epsilon^2}{\sigma_y^2 - \sigma_\epsilon^2} \\
&= 1 - \frac{\left\langle (y - \hat{\mu}_x)^2 \right\rangle_{x,y} - \sigma_\epsilon^2}{\sigma_y^2 - \sigma_\epsilon^2},
\end{aligned} \tag{12}$$

where $\sigma_y^2 = \text{Var}(y)$ and $\sigma_\epsilon^2 = \mathbb{E}_x[\text{Var}(y|x)]$ are estimated from the data. Now let us expand $\langle D_{KL}[p(y|x)||\hat{p}(y|x)] \rangle_x$ in the case of a Gaussian likelihood:

$$\begin{aligned}
\langle D_{KL}[p(y|x)||\hat{p}(y|x)] \rangle_x &= \langle \log\left(p(y|x)\right) - \log(\hat{p}(y|x)) \rangle_{x,y} \\
&= \left\langle \log\left(\left(2\pi\sigma_x^2\right)^{-1/2} \right) - \frac{(y_x - \mu_x)^2}{2\sigma_x^2} - \log\left(\left(2\pi\hat{\sigma}_x^2\right)^{-1/2} \right) + \frac{(y_x - \hat{\mu}_x)^2}{2\hat{\sigma}_x^2} \right\rangle_{x,y} \\
&= \left\langle \log\left(\frac{\hat{\sigma}_x}{\sigma_x}\right) - \frac{(y_x - \mu_x)^2}{2\sigma_x^2} + \frac{(y_x - \hat{\mu}_x)^2}{2\hat{\sigma}_x^2} \right\rangle_{x,y} \\
&= \left\langle \log\left(\frac{\hat{\sigma}_x}{\sigma_x}\right) - \frac{(y_x - \mu_x)^2}{2\sigma_x^2} \right\rangle_{x,y} + \left\langle \frac{(y_x - \mu_x + \mu_x - \hat{\mu}_x)^2}{2\hat{\sigma}_x^2} \right\rangle_{x,y} \\
&= \left\langle \log\left(\frac{\hat{\sigma}_x}{\sigma_x}\right) - \frac{1}{2} \right\rangle_x + \left\langle \frac{(y_x - \mu_x)^2 + (\mu_x - \hat{\mu}_x)^2 + 2(y_x - \mu_x)(\mu_x - \hat{\mu}_x)}{2\hat{\sigma}_x^2} \right\rangle_{x,y} \\
&= \left\langle \log\left(\frac{\hat{\sigma}_x}{\sigma_x}\right) - \frac{1}{2} \right\rangle_x + \left\langle \frac{\left\langle (y_x - \mu_x)^2 \right\rangle_{y|x} + (\mu_x - \hat{\mu}_x)^2 + 2\overbrace{\left\langle (y_x - \mu_x)(\mu_x - \hat{\mu}_x) \right\rangle_{y|x}}^{=0}}{2\hat{\sigma}_x^2} \right\rangle_x \\
&= \left\langle \log\left(\frac{\hat{\sigma}_x}{\sigma_x}\right) - \frac{1}{2} \right\rangle_x + \left\langle \frac{\sigma_x^2 + (\mu_x - \hat{\mu}_x)^2}{2\hat{\sigma}_x^2} \right\rangle_x \\
&= \left\langle \log\left(\frac{\hat{\sigma}_x}{\sigma_x}\right) - \frac{1}{2} + \frac{\sigma_x^2 + (\mu_x - \hat{\mu}_x)^2}{2\hat{\sigma}_x^2} \right\rangle_x \\
&= \left\langle \log\left(\frac{\hat{\sigma}_x}{\sigma_x}\right) - \frac{1}{2} + \frac{\sigma_x^2}{2\hat{\sigma}_x^2} + \frac{(\mu_x - \hat{\mu}_x)^2}{2\hat{\sigma}_x^2} \right\rangle_x \\
&= \langle f(\hat{\sigma}_x) \rangle_x + \frac{1}{2} \left\langle \frac{(\mu_x - \hat{\mu}_x)^2}{\hat{\sigma}_x^2} \right\rangle_x
\end{aligned} \tag{13}$$

where $\hat{\sigma}_x$ is the noise estimated by the model, σ_x is the true noise, and $f(\hat{\sigma}_x) = \log\left(\frac{\hat{\sigma}_x}{\sigma_x}\right) - \frac{1}{2} + \frac{\sigma_x^2}{2\hat{\sigma}_x^2}$. Note that if $\hat{\sigma}_x = \sigma_x$ then $f(\hat{\sigma}_x) = 0$, and we would have:

$$\langle D_{KL}[p(y|x)||\hat{p}(y|x)]\rangle_x = \frac{1}{2}\left\langle \frac{(\mu_x - \hat{\mu}_x)^2}{\hat{\sigma}_x^2} \right\rangle_x$$

The term with the expectation can further be simplified. If the noise variance $\hat{\sigma}_x^2$ is not dependent on the stimulus x, which is the case for a Gaussian distribution, then $\hat{\sigma}_x^2 = \hat{\sigma}_\epsilon^2 = \langle \hat{\sigma}_x^2 \rangle_x$ and we can simply bring it outside the expectation:

$$\left\langle \frac{(\mu_x - \hat{\mu}_x)^2}{\hat{\sigma}_x^2} \right\rangle_x = \frac{\langle (\mu_x - \hat{\mu}_x)^2 \rangle_x}{\hat{\sigma}_\epsilon^2} \tag{14}$$

This would also mean that $f(\hat{\sigma}_x) = f(\hat{\sigma}_\epsilon) = \log\left(\frac{\hat{\sigma}_\epsilon}{\sigma_\epsilon}\right) - \frac{1}{2} + \frac{\sigma_\epsilon^2}{2\hat{\sigma}_\epsilon^2}$. However, if the noise variance is stimulus-dependent then the term with the expectation can be approximated via first-order Taylor expansion around the expected values of the numerator and denominator $\mathbf{c} = (\langle (\mu_x - \hat{\mu}_x)^2 \rangle_x, \langle \hat{\sigma}_x^2 \rangle_x)$:

$$\begin{aligned}\left\langle \frac{(\mu_x - \hat{\mu}_x)^2}{\hat{\sigma}_x^2} \right\rangle_x &\approx \frac{\langle (\mu_x - \hat{\mu}_x)^2 \rangle_x}{\langle \hat{\sigma}_x^2 \rangle_x} \\ &\approx \frac{\langle (\mu_x - \hat{\mu}_x)^2 \rangle_x}{\hat{\sigma}_\epsilon^2}\end{aligned}$$

Since we are dealing with a Gaussian distribution we will continue with the case where noise variance is not stimulus-dependent. However, the derivation applies to the approximate case too. Let us now relate Eq. 11 and Eq. 13:

$$\begin{aligned}\langle D_{KL}[p(y|x)||\hat{p}(y|x)]\rangle_x &= f(\hat{\sigma}_\epsilon) + \frac{1}{2}\frac{\langle (\mu_x - \hat{\mu}_x)^2 \rangle_x}{\hat{\sigma}_\epsilon^2} \\ &= f(\hat{\sigma}_\epsilon) + \frac{1}{2}\frac{\langle (\mu_x - \hat{\mu}_x)^2 \rangle_x}{\sigma_\epsilon^2} \times \frac{\sigma_s^2}{\sigma_s^2} \\ &= f(\hat{\sigma}_\epsilon) + \frac{1}{2}\frac{\langle (\mu_x - \hat{\mu}_x)^2 \rangle_x}{\sigma_s^2} \times \frac{\sigma_s^2}{\hat{\sigma}_\epsilon^2} \\ &= f(\hat{\sigma}_\epsilon) + \frac{1}{2}(1 - FEVE) \times \frac{\sigma_s^2}{\hat{\sigma}_\epsilon^2} \end{aligned} \tag{15}$$

Note that when estimated noise variance matches the true noise variance, then Eq. 15 becomes:

$$\langle D_{KL}[p(y|x)||\hat{p}(y|x)]\rangle_x = \frac{1}{2}(1 - FEVE) \times SNR$$

G RELATION BETWEEN NORMALIZED INFORMATION GAIN AND CORRELATION

Another commonly used metric is the trial-averaged correlation between the model prediction $\hat{\mu}_x = \langle \hat{y}|x \rangle_{\hat{y}|x}$ and true responses $\mu_x = \langle y|x \rangle_{y|x}$:

$$\rho(\hat{\mu}_x, \mu_x) = \frac{\text{Cov}(\hat{\mu}_x, \mu_x)}{\sqrt{\hat{\sigma}_s^2 \cdot \sigma_s^2}}$$

To relate this quantity to $\langle D_{KL}[p(y|x)||\hat{p}(y|x)] \rangle_x$, we start by expanding Eq. 14. Specifically, we add and subtract $\mu = \langle \mu_x \rangle = \langle \hat{\mu}_x \rangle$ in the numerator:

$$\begin{aligned}
\frac{\left\langle (\mu_x - \hat{\mu}_x)^2 \right\rangle_x}{\hat{\sigma}_\epsilon^2} &= \frac{\left\langle ((\mu_x - \mu) - (\hat{\mu}_x - \mu))^2 \right\rangle_x}{\hat{\sigma}_\epsilon^2} \\
&= \frac{\left\langle (\mu_x - \mu)^2 + (\hat{\mu}_x - \mu)^2 - 2(\mu_x - \mu)(\hat{\mu}_x - \mu) \right\rangle_x}{\hat{\sigma}_\epsilon^2} \\
&= \frac{\sigma_s^2 + \hat{\sigma}_s^2 - 2\text{Cov}(\hat{\mu}_x, \mu_x)}{\hat{\sigma}_\epsilon^2} \\
&= \frac{\sigma_s^2 + \hat{\sigma}_s^2 - 2\text{Cov}(\hat{\mu}_x, \mu_x)}{\hat{\sigma}_\epsilon^2} \times \frac{\sigma_s^2}{\sigma_s^2} \\
&= \frac{\sigma_s^2 + \hat{\sigma}_s^2 - 2\text{Cov}(\hat{\mu}_x, \mu_x)}{\sigma_s^2} \times \frac{\sigma_s^2}{\hat{\sigma}_\epsilon^2} \\
&= \left(1 + \frac{\hat{\sigma}_s^2}{\sigma_s^2} - \frac{2\hat{\sigma}_s}{\sigma_s} \rho(\hat{\mu}_x, \mu_x) \right) \times \frac{\sigma_s^2}{\hat{\sigma}_\epsilon^2}
\end{aligned}$$

Putting this back into Eq. 13:

$$\langle D_{KL}[p(y|x)||\hat{p}(y|x)] \rangle_x = f(\hat{\sigma}_\epsilon) + \frac{1}{2} \left(1 + \frac{\hat{\sigma}_s^2}{\sigma_s^2} - \frac{2\hat{\sigma}_s}{\sigma_s} \rho(\hat{\mu}_x, \mu_x) \right) \times \frac{\sigma_s^2}{\hat{\sigma}_\epsilon^2} \tag{16}$$

Again, if the model's noise variance matches the true noise variance ($\sigma_\epsilon^2 = \hat{\sigma}_\epsilon^2$), we have:

$$\langle D_{KL}[p(y|x)||\hat{p}(y|x)] \rangle_x = \frac{1}{2} \left(1 + \frac{\hat{\sigma}_s^2}{\sigma_s^2} - \frac{2\hat{\sigma}_s}{\sigma_s} \rho(\hat{\mu}_x, \mu_x) \right) \times SNR$$

If we further assume that the model's signal variance matches the true signal variance, $\hat{\sigma}_s^2 = \sigma_s^2$, we get:

$$\langle D_{KL}[p(y|x)||\hat{p}(y|x)] \rangle_x = (1 - \rho(\hat{\mu}_x, \mu_x)) \times SNR$$

H Optimizing correlation only focuses on matching trial-averaged responses

In addition to trial-averaged correlation, neural encoding models are also evaluated via single-trial correlation [23]. While in the trial-averaged case the correlation obviously only focuses on conditional means, here we show this is the case even for single-trial correlation. That is, optimizing single-trial correlation only focuses on matching the conditional means:

$$\begin{aligned}
\rho_{st}(\hat{\mu}_x, y) &= \frac{\text{Cov}(\hat{\mu}_x, y)}{\sqrt{\hat{\sigma}_s^2 \sigma_y^2}} \\
&= \frac{\text{Cov}(\hat{\mu}_x, y)}{\sqrt{\hat{\sigma}_s^2 \sigma_y^2}} \\
&= \frac{\text{Cov}(\mathbb{E}[\hat{\mu}_x|x], \mathbb{E}[y|x]) + \overbrace{\mathbb{E}[\text{Cov}(\hat{\mu}_x, y|x)]}^{=0}}{\sqrt{\hat{\sigma}_s^2 \sigma_y^2}} \\
&= \frac{\text{Cov}(\hat{\mu}_x, \mu_x)}{\sqrt{\hat{\sigma}_s^2 \sigma_y^2}}
\end{aligned}$$

where $\hat{\mu}_x$ is the predicted conditional mean, μ_x is the trial-averaged response, $\hat{\sigma}_s^2$ is the model signal variance, and σ_y^2 is the total data variance computed across all trials. Note that this quantity is invariant to affine transformations of the predicted conditional mean.

I Other approaches for a GS estimate

I.1 Maximum a posteriori estimate

Instead of using the full posterior predictive to obtain a good GS model, one can use the maximum a posteriori (MAP) estimate of the distribution parameters. Here, we show the derivation of the MAP estimate for the zero-inflated Log-Normal distribution. However, it does not perform as well as the posterior predictive approach, see Fig. S2.

The maximum a posteriori estimator of a parameter $\phi \in \{\theta_0, \theta_1, q\}$ of a zero inflated likelihood can be computed as

$$\begin{aligned}\hat{\phi}_{MAP} &= \arg\max_{\phi} p(\mathbf{y}_{\setminus i}|\theta, q)p(\theta)p(q)\\ &= \arg\max_{\phi} p(\mathbf{y}^0_{\setminus i}|\theta_0)p(\theta_0)p(\mathbf{y}^1_{\setminus i}|\theta_1)p(\theta_1) \cdot q^{n_1}(1-q)^{n_0}p(q)\\ &= \arg\max_{\phi} \log\left(p(\mathbf{y}^0_{\setminus i}|\theta_0)p(\theta_0)\right) + \log\left(p(\mathbf{y}^1_{\setminus i}|\theta_1)p(\theta_1)\right) + \log\left(q^{n_1}(1-q)^{n_0}p(q)\right)\end{aligned}$$

where the second step is analogous to Eq. 7.

MAP estimate for q. In order to obtain the maximum a posteriori estimator for q we set the derivative with respect to q to zero. As a prior for q we choose a Beta distribution $p(q) = \mathrm{Beta}(q; \alpha'', \beta'')$:

$$\begin{aligned}\hat{q}_{MAP} &= \arg\max_{q} \log\left(q^{n_1}(1-q)^{n_0}p(q)\right)\\ &= \arg\max_{q} \log\left(q^{n_1}(1-q)^{n_0}\frac{q^{\alpha''-1}(1-q)^{\beta''-1}}{B(\alpha'', \beta'')}\right)\\ &= \arg\max_{q} \underbrace{\log\left(q^{n_1+\alpha''-1}(1-q)^{n_0+\beta''-1}\right)}_{:=f(q)}\end{aligned}$$

$$\begin{aligned}\frac{\partial f(q)}{\partial q} &= \frac{\partial}{\partial q}\left[\log\left(q^{n_1+\alpha''-1}(1-q)^{n_0+\beta''-1}\right)\right]\\ &= \frac{\partial}{\partial q}\left[(n_1+\alpha''-1)\log(q) + (n_0+\beta''-1)\log(1-q)\right]\\ &= (n_1+\alpha''-1)\frac{1}{q} - (n_0+\beta''-1)\frac{1}{1-q}\\ &\overset{!}{=} 0\\ \hat{q}_{MAP} &= \frac{n_1+\alpha''-1}{n_0+n_1+\alpha''+\beta''-2}\end{aligned}$$

MAP estimate for θ_1. The parameters θ_1 are all parameters of the non-zero part of the distribution. In the case of a LogNormal distributions, this is $\theta_1 \in \{\mu, \sigma^2\}$ and we assume a Normal-Inverse-Gamma prior $p(\theta_1) = \mathcal{N}G^{-1}(\mu'', \lambda'', \alpha'', \beta'')$. The posterior then follows a Normal-Inverse-Gamma distribution as well

$$\begin{aligned}p(\mathbf{y}^1_{\setminus i}|\theta_1)p(\theta_1) &\approx p(\theta_1|\mathbf{y}^1_{\setminus i})\\ &= \mathcal{N}G^{-1}(\mu', \lambda', \alpha', \beta')\\ &= \frac{\sqrt{\lambda'}\beta'^{\alpha'}}{\sqrt{2\pi}\Gamma(\alpha')}\frac{1}{\sigma}\left(\frac{1}{\sigma^2}\right)^{\alpha'+1}\exp\left(-\frac{2\beta' + \lambda'(\mu-\mu')^2}{2\sigma^2}\right)\end{aligned}$$

with

$$
\begin{aligned}
\mu' &= \frac{\mu''\nu'' + n_1\overline{y}}{\nu'' + n_1} \\
\nu' &= \nu'' + n_1 \\
\alpha' &= \alpha'' + \frac{n_1}{2} \\
\beta' &= \beta'' + \frac{1}{2}\sum_{y_j \in y^1_{\setminus i}}^{n_1} (y_j - \overline{y})^2 + \frac{n_1\nu''(\overline{y} - \mu'')^2}{2(\nu'' + n_1)}
\end{aligned}
$$

where $\overline{y} := 1/n_1 \sum_{y_j \in \mathbf{y}^1_{\setminus i}} y_j$

The maximum a posteriori estimator of μ can then be obtained as follows:

$$
\begin{aligned}
\hat{\mu}_{MAP} &= \arg\max_{\mu} \log\Big(p(\mathbf{y}^1_{\setminus i}|\theta_1)p(\theta_1)\Big) \\
&= \arg\max_{\mu} \log\left(\exp\left(-\frac{2\beta' + \lambda'(\mu - \mu')^2}{2\sigma^2}\right)\right) \\
&= \arg\max_{\mu} -\frac{\lambda'(\mu - \mu')^2}{2\sigma^2} \\
&= \mu'
\end{aligned}
$$

And for σ^2:

$$
\hat{\sigma}^2_{MAP} = \arg\max_{\sigma^2} \underbrace{\log\left(\frac{1}{\sigma}\left(\frac{1}{\sigma^2}\right)^{\alpha'+1} \exp\left(-\frac{2\beta' + \lambda'(\mu - \mu')^2}{2\sigma^2}\right)\right)}_{:=f(\sigma^2)}
$$

$$
\begin{aligned}
\frac{\partial f(\sigma^2)}{\partial \sigma^2} &= \frac{\partial}{\partial \sigma^2}\left(-\frac{1}{2}\log\sigma^2 - (\alpha' + 1)\log\sigma^2 - \frac{2\beta' + \lambda'(\mu - \mu')^2}{2}\frac{1}{\sigma^2}\right) \\
&= (-\alpha' - \frac{3}{2})\frac{1}{\sigma^2} + \frac{2\beta' + \lambda'(\mu - \mu')^2}{2}\frac{1}{\sigma^4} \\
&\overset{!}{=} 0 \\
\hat{\sigma}^2_{MAP} &= \frac{2\beta' + \lambda'\left(\hat{\mu}_{MAP} - \mu'\right)^2}{2\alpha' + 3}
\end{aligned}
$$

MAP estimate for θ_0 In general, the maximum a posteriori estimator for θ_0 can be obtained analogously. In our case we model the zero part of the response distribution with a uniform distribution which does not have a parameter θ_0.

I.2 Gold Standard model as a mixture of Null and Posterior Predictive distributions

For some individual neurons and images the null model performs better than the Gold Standard model because the prior of the GS model is fitted per neuron but not per image. In cases with few positive responses where the GS model has to rely heavily on the prior, the performance can thus be sub-optimal for individual images. One idea, suggested by one of the reviewers, to circumvent this is to build a mixture model p_{**} between the GS p_* and Null p_0 model:

$$p_{**}(y_i|\mathbf{y}_{\setminus i}, y) = w_i \cdot p_*(y_i|\mathbf{y}_{\setminus i}) + (1 - w_i) \cdot p_0(y),$$

where $w \in [0, 1]$. We optimized w in a leave-one-out manner, just like p_* itself is obtained in a leave-one-out-manner: we obtained a w for each target repeat (per neuron per image) by optimizing p_{**} with respect to w on the other repeats. However, the resulting GS mixture model does not outperform the Bayesian GS model, see Fig. S2.

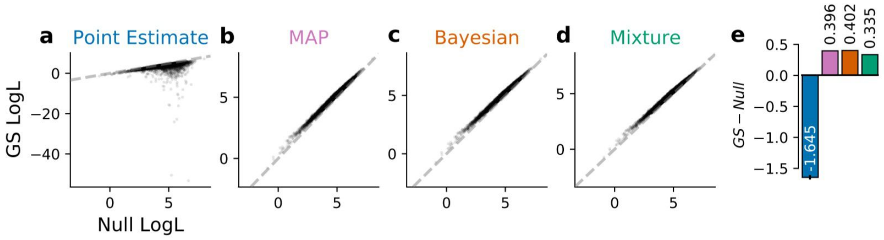


Fig. S2: Comparison of different methods to obtain an upper bound (GS): **a–d:** Various GS model log-likelihood vs. the Null model log-likelihood. Data is per neuron (averaged over repeats and stimuli). **e:** The full Bayesian Posterior Predictive outperforms the Point Estimate, the Maximum a posteriori (MAP), and the Mixture model. Each bar is the difference between the corresponding GS model and the Null model, averaged over repeats, stimuli, and neurons. Error bars correspond to the SEM and evaluate to ±0.03 for the PE and ±0.002 for the other GS models.

J NInGa across different datasets

We performed an analysis similar to Fig. 4c (blue bar) but for multiple datasets. We trained the same model described in section 3.2 on five different additional datasets from [23]. Our results show that using NInGa allows a better comparison of models that are trained on different datasets which can exhibit different levels of achievable performance (Fig. S3, compare left vs. right) . When models with the same architecture are trained on different datasets the resulting performances are more similar in NInGa (Eq. 1) than in the unnormalized IG (i.e. the numerator of Eq. 1), because the performance of the model is reported relative to the Null and Gold Standard model.

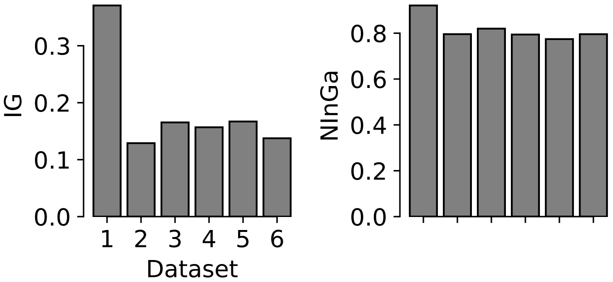


Fig. S3: Comparison of Trained Model performance on different datasets. **Left:** Models evaluated on simple Information Gain (IG), i.e. the numerator of Eq. 1. **Right:** Models evaluated on Normalized Information Gain (NInGa), i.e. the full Eq. 1.