

Deep Neural Networks for System Identification in Mouse Visual Cortex

Dissertation

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

vorgelegt von
Konstantin-Klemens Lurz
aus Günzburg

Tübingen
2024

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

Tag der mündlichen Qualifikation: 21.10.2024

Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatter:	Prof. Dr. Fabian Sinz
2. Berichterstatter:	Prof. Dr. Philipp Berens

Ich erkläre, dass ich die zur Promotion eingereichte Arbeit mit dem Titel “Deep Neural Networks for System Identification in Mouse Visual Cortex” 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

I consider myself incredibly privileged to have shared paths with remarkable and inspiring people during my entire doctoral studies. Throughout my doctoral journey at the University of Tübingen, the University of Göttingen and the International Max Planck Research School, these people helped turning this challenging and, at times, exasperating endeavor into a memorable experience full of personal anecdotes which I will always look back on fondly. Although it would be impossible to mention every individual who shared a part of my journey during the last years, I want to point out a few exceptional people who I owe special gratitude.

First and foremost, I want to thank my mentor and supervisor as a PhD student, Fabian Sinz. Fabian, your sharp logical thinking accompanied with empathy and social intelligence is a combination that is outstanding in the scientific community. You deeply influenced me with respect to the scientific method, team leading, and an approach to life in general. For instance, I will never forget the mathematical and philosophical advice you gave me when I was desperate about failed applications: "Many things in life are a Bernoulli process. All you can do is optimize p and hope for the best." I also want to thank my other supervisors who, together, formed my thesis advisory committee. Caterina De Bacco, Philipp Berens and Alexander Ecker promoted my scientific progress with their helpful advice, critical questions and refreshing new ideas.

Beyond my supervisors, I want to thank my colleagues without whom the last years could never have been both as successful as well as enjoyable. Mohammad, you were my wingman, co-author, flatmate, travel-partner and, most importantly, my friend. I can say without exaggeration that there is not a single person I have spent so much time with during the past years, and I enjoyed every minute of it. I can only wish that our connection will stay for a lifetime. Arne, Christoph, Konstantin, Mara, Judith, Edgar, Suhas, Dominik, Pawel and Pavithra, you have made my time in the lab very enjoyable both as co-authors as well as co-players at the kicker table. Finally, all the people in Andreas Toliás' lab, without your collaboration on the experimental side my research would never have been possible. I am deeply thankful for that.

Pursuing doctoral studies for so many years is not only a scientific but also an emotional and mental challenge. I could not have endured the ups and downs of the past years without the support of my friends. I want to especially thank Elle, Elina, Oli and Jana, who made me feel like home in Tübingen from the very first minute. Thank you so much for letting me be a part of your wonderful WG.

Finally, I want to express my deepest gratitude to my family. My parents, Klemens and Ortrud, you have always been there for me, showed interest in what I am doing and supported me with everything you had, not only during my PhD but throughout the whole journey that led to it. My girlfriend, Theresa, you believed in me and supported me during the hardest times of my PhD. You were understanding and loving despite having to endure a long distance relationship and uncertainty about our future. Thank you so much for being in my life.

Abstract

The pursuit of a thorough understanding of human visual perception and sensation has occupied many great minds for centuries. In recent years, however, Vision Neuroscience has seen a surge in advancements due to the mutually beneficial symbiosis with the field of Machine Learning. Especially the adaptation of Artificial Neural Networks (ANNs) as system identification models has proven to be of great value for modeling the complex computations that are performed by cortical neurons for visual processing. Such system identification models using ANNs are now the state-of-the-art for predicting neural activity as a response to natural stimuli. Additionally, these models have proven exceptionally useful for traditional biological Neuroscience as well: Experiments whose search space would be unfeasible to investigate directly in an animal (in-vivo) can be first run on a system identification model instead (in-silico). The models can then help to shrink the search space in-silico to a size which is more manageable for the actual in-vivo experiment.

This thesis builds on the latest advancements of ANN system identification models of mouse visual cortex for neural prediction and in-silico experiments. It contributes to the knowledge of the field in three major directions: It first introduces a novel ANN architecture which is exceptionally data-efficient and which can cleverly make use of retinotopy by incorporating anatomical information additionally to the functional data. Combined with a novel method of matching and tracking neurons over separate experimental sessions, the resulting model sets a new state-of-the-art for neural response prediction and, more importantly, yields features that generalize between animals. Since such models, however, only capture a part of the full neural response distribution while many normative theories require knowledge of the full distribution, the second contribution of this thesis sets the stage for full-likelihood models: By mathematically deriving the full posterior predictive distribution of the most popular distributions in Neuroscience, the upper bounds of a likelihood-based performance metric are obtained. This metric makes it possible to accurately evaluate full-likelihood models and, more precisely, get insights into exactly which parts of the model need the most attention for improvement. In the final contribution of the thesis the framework for full-likelihood models and the generalizing state-of-the-art features are then used to obtain neuroscientific insights: The analysis of the variability of neural firing. To this end, novel types of stimuli that aim at driving or suppressing the variance conditioned on a stimulus are generated. The results indicate that the variance of the firing rate might be affected by the size of the stimulus, much like known center-surround effects for the mean firing rate.

Zusammenfassung

Das Streben nach einem gründlichen Verständnis der menschlichen visuellen Wahrnehmung und Empfindung hat viele große Intellektuelle seit Jahrhunderten beschäftigt. In den letzten Jahren jedoch hat die visuelle Neurowissenschaft einen sprunghaften Anstieg an Fortschritten erlebt, was auf die für beide Seiten vorteilhafte Symbiose mit dem Bereich maschinellen Lernens zusammen hängt. Insbesondere die Adaption von Künstlichen Neuronalen Netzen (ANNs) als Systemidentifikationsmodelle hat sich als sehr wertvoll für die Modellierung der komplexen Berechnungen, die von kortikalen Neuronen zur visuellen Verarbeitung durchgeführt werden, erwiesen. Solche Systemidentifikationsmodelle mit ANNs sind heute der Stand der Technik für die Vorhersage neuronaler Aktivität als Reaktion auf natürliche Stimuli. Außerdem haben sich diese Modelle auch für die traditionelle biologische Neurowissenschaft als außerordentlich nützlich erwiesen: Experimente, deren Suchraum sich nicht direkt am Tier (in-vivo) untersuchen ließe, können stattdessen zunächst auf einem Systemidentifikationsmodell (in-silico) durchgeführt werden. Die Modelle können dann helfen, den Suchraum in-silico auf eine Größe zu verkleinern, die für das eigentliche in-vivo Experiment machbar ist. Diese Thesis baut auf den jüngsten Fortschritten der ANN Systemidentifikationsmodelle des visuellen Kortex der Maus für neuronale Vorhersagen und in-silico Experimente auf. Sie trägt in drei Hauptrichtungen zum Wissen auf diesem Gebiet bei: Zunächst wird eine neuartige ANN-Architektur vorgestellt, die außergewöhnlich dateneffizient ist und die Retinotopie geschickt nutzen kann, indem sie zusätzlich zu den funktionellen Daten anatomische Informationen einbezieht. In Kombination mit einer neuartigen Methode zum Abgleich und zur Verfolgung von Neuronen über verschiedene Versuchssitzungen hinweg setzt das resultierende Modell neue Maßstäbe für die Vorhersage neuronaler Reaktionen und, was noch wichtiger ist, es liefert Features, die sich zwischen verschiedenen Tieren verallgemeinern lassen. Da solche Modelle jedoch nur einen Teil der gesamten neuronalen Antwortverteilung erfassen, während viele normative Theorien die Kenntnis der gesamten Verteilung erfordern, schafft der zweite Beitrag dieser Arbeit die Voraussetzungen für Modelle der vollen Wahrscheinlichkeitsverteilung (full-likelihood models): Durch die mathematische Ableitung des vollständigen Posterior Predictive der gängigsten Verteilungen in den Neurowissenschaften werden die oberen Grenzen einer Likelihood-basierten Leistungsmetrik ermittelt. Diese Metrik ermöglicht eine genaue Bewertung von Full-Likelihood-Modellen und gibt Aufschluss darüber, welche Teile des Modells am dringendsten verbessert werden müssen. Im letzten Beitrag der Arbeit werden das Framework für Full-Likelihood-Modelle und die generalisierenden Features verwendet, um neurowissenschaftliche Erkenntnisse zu gewinnen: Die Analyse der Variabilität des neuronalen Feuerns. Zu diesem Zweck werden neuartige Stimuli erzeugt, die darauf abzielen, die auf einen Stimulus konditionierte Varianz zu erhöhen oder zu unterdrücken. Die Ergebnisse deuten darauf hin, dass die Varianz der Feuerrate von der Größe des Stimulus beeinflusst werden könnte, ähnlich wie die bekannten Center-Surround Effekte für die mittlere Feuerrate.

Contents

1	Introduction	7
1.1	A brief history of vision science	7
1.2	Neural system identification	10
1.3	Own contribution	13
1.3.1	Publications and projects constituting a part of this thesis . . .	13
1.3.2	Other publications	14
2	Background	15
2.1	Artificial Neural Networks	15
2.2	Neural system identification ANN architecture	17
2.3	Digital twins and in-silico synthesis of stimuli	19
2.4	Data	20
2.5	Posterior predictive	21
3	Generalization in data-driven models of primary visual cortex	25
3.1	Motivation	25
3.2	Methods	26
3.3	Results	27
3.4	Discussion	29
4	Bayesian oracle for bounding information gain in neural encoding models	31
4.1	Motivation	31
4.2	Methods	32
4.3	Results	34
4.4	Discussion	37
5	Variably exciting stimuli	38
5.1	Motivation	38
5.2	Methods	39
5.3	Results	41
5.4	Discussion	47
6	Discussion and conclusion	48
	References	50
	Manuscripts	60

1 Introduction

1.1 A brief history of vision science

For centuries, vision, as the primary sense that humans rely on, has fascinated countless brilliant minds. Ancient philosophers like Democritus, Plato and Aristotle developed theories of vision debating whether it was the image that entered the eye (intromission theory) or the eye that was sending out a beam to the object like a lantern (emission theory)[1]. The case was decided in the 11th century when the physicist Alhazen discovered the laws of reflection and refraction, an early version of Fermat's principle. He was also the first to realize that vision ultimately takes place in the brain by arguing that visual perception is subjective and influenced by personal experience [2]. It was the artist Leonardo Da Vinci who made the next major contribution to vision science driven by his desire to precisely understand the interplay of light and visual perception that would affect the observer of his artwork. He presumed that space is filled with an infinite number of images of an object that spread in all directions like a wave, rather than a linear corpuscular stream [3], unknowingly touching upon a phenomenon that would later be known as the wave-particle-duality of light in Quantum Mechanics:

Just as the stone thrown into water becomes the centre and cause of various circles, and sound made in the air spreads out in circles, so every body placed within the luminous air spreads itself out in circles and fills the surrounding parts with an infinite number of images of itself, and appears all in all and all in each part.

— The Notebooks of Leonardo Da Vinci

Two centuries later, Newton then used a prism to split light rays which lead him to publish groundbreaking work on the theory of colours in his book *Opticks* [4] presenting 7 primary colors of which white light is composed of. In his book *Zur Farbenlehre* [5], Goethe challenged Newton's analytic research of a physical optical spectrum but concerned himself more with the human perception of color [6], tailoring the topic towards a more brain and neuroscience leaning approach.

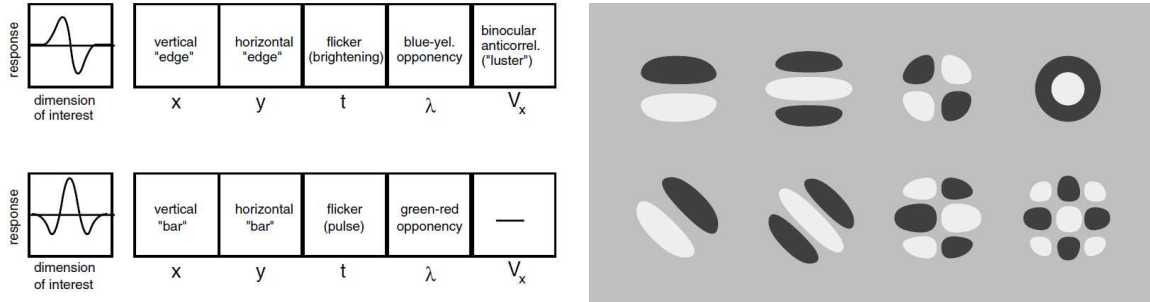
Vision Neuroscience The development of amplifiers and the advent of electrophysiology allowed the recording of neuronal firing rates [7, 8], kick-starting research on vision beyond optics: the apparatus processing the visual signal. The notion of a neuron's receptive field was discovered by Hartline [9] in 1940 and it became increasingly clear that neurons are at the heart of thinking and perception, performing far more complex computations than previously anticipated. Starting with neurons early in the visual pathway (i.e. retina, optic nerve) and successively also analyzing

higher areas in the visual cortex it became apparent that many neurons seem to compute local derivatives on slices of the so called plenoptic function [10]:

The plenoptic function P (defined in their book by Adelson et al. [10]) is a theoretical entity which describes everything that can potentially be seen, drawing inspiration from Leonardo Da Vinci's idea of infinite number of images everywhere in space which he calls "radiant pyramid". It is thus a high dimensional function of the spatial coordinates x and y , the color or wavelength λ at this position, the time t and, in the case of a holographic movie, the viewing position V_x, V_y, V_z .

$$P = P(x, y, \lambda, t, V_x, V_y, V_z) \quad (1.1)$$

The task of early vision is thought to be the extraction of a small set of useful measurements about each observable location in the plenoptic function. Since only a small fraction of the vast information contained in P is of importance to the organism while at the same time the evolutionary pressure is high to keep the energy consumption of the brain economical, the early neurons need to act as strong filters on the original visual stimulus [11]. Given that points in P are locally highly correlated, good filters for extracting non-redundant information are filters of local change, i.e. low order derivatives. Take, for instance the x, y plane of the plenoptic function: a black and white image. The simplest and least redundant representation of an object in this image is a sketch of its edges, i.e. of the points with highest local derivative. Observing the t slice of the plenoptic function results in a movie. Here, measuring the local derivative with respect to t produces filters for motion and acceleration which are vital for the organism. For a schematic overview of local derivative filters being applied on single dimensions of the plenoptic function, see Fig. 1.1.



(a) Derivatives along single dimensions yield some basic visual measurements. (b) Low-order derivatives lead to a few two-dimensional operators (receptive fields)

Figure 1.1: Local derivative filters on slices of the plenoptic function lead to basic measurements of non-redundant features. Source: [10]

The insights about neural processing with respect to several slices of the plenoptic function as described above were found and shown in real neurons during the second half of the 20th century by several neuroscientists: Hubel and Wiesel [12, 13] showed orientation and direction tuning in the cat's visual cortex. The filters found in their experiments were validated by a model of Olshausen and Field [14] trained on natural images. Enroth-Cugell and Robson [15] found contrast sensitivity in cat retinal ganglion cells and modeled neural responses to edges as a difference of Gaussians [15–17]. Enroth-Cugell and Pinto [18] discovered the interplay of center and surround signals. The combination of simple filters or cells in order to form complex cells was

shown for the perception of motion by Adelson and Bergen [19] and the orientation and spatial frequency representation in V1 by De Valois et al. [20].

With the advent of computers, the interest in vision came no longer only from a scientific or medical but also from an engineering perspective. Understanding the way that the brain processes visual information could be used to design algorithms for processing visual data. Barlow et al. [21], for instance, compared visual pattern analysis in machines and animals and Geirhos et al. [22] show the texture versus shape bias in machines and humans. Computers also facilitated the analysis of neural datasets which were becoming larger and more complex due to advanced recording techniques. Especially the advances in Machine Learning (ML) and Deep Learning have resulted in a surge of highly performing models of various stages of the visual pathway in the brain. This way, a highly productive symbiosis of Machine Learning and Neuroscience has emerged in which both fields profit from each other's insights.

This thesis builds on the recent advances in the field of ML models in Vision Neuroscience, specifically mouse visual cortex. It makes contributions to the field in the following ways:

The **first project** (Chapter 3, [23]) tackles the problem of generalization, i.e. the training of representations that are generally characteristic for visual cortex. Such representations should not be subject to the peculiarities of an individual animal but show universal properties that generalize between animals of the same species. The representations were successfully obtained by designing a novel model architecture and by combining data from different sources as well as experimental sessions. The resulting model not only generalizes between animals but also sets a new state-of-the-art in neural response prediction in mouse visual cortex from natural stimuli. Additionally, it is 40% more data efficient than previous models and captures better characteristic cortical features than a competing approach, the task-driven pretraining of representations.

The **second project** (Chapter 4, [24]) explores the necessary theoretical background for a new class of models: Models that predict the full neural response distribution. Traditionally, neural responses were modeled with a focus on the conditional mean of the distribution. Higher moments of the distribution like the variance may, however, carry important information about the stimulus and should not be neglected. Predictions about the variance are described by many normative theories like the Bayesian brain hypothesis, neural sampling or probabilistic population codes. However, when building models that capture the full response distribution of a single neuron, a performance measure is needed that is fully likelihood-based as opposed to the traditional mean dependent correlation-based measures. In this project, all necessary components of such a measure (normalized information gain) were mathematically derived for a variety of commonly used distributions. Then, a data-efficient and robust estimator of the performance upper and lower bound was obtained.

Given the well performing and generalizing model from the first project and the methods to extend this model to full distributions from the second project, the **third project** (Chapter 5) finally aims at gaining neuroscientific insights from these models. By using the models to synthesize new stimuli with the objective to modulate the variance of the distribution, it was possible to find clues to what kind of information neurons might encode via the variance. While the results could not yet be tested back in the animal, they were consistent over various *in-silico* models.

1.2 Neural system identification

As described by [25], "system identification is the field of mathematical modeling of systems from experimental data". In the case of Vision Neuroscience, the system to identify is the visual system or parts thereof, consisting for mammals of the eye, the retina, the optic nerve, the optic tract and the visual cortex. The experimental data usually consists of a visual stimulus x and measured neural activity y in response to this stimulus. The probabilistic relationship between stimulus and neural activity $p(y|x)$ is then approximated by a model distribution $p_\theta(y|x)$ with parameters θ , see Fig.1.2. The choice of model class is a substantial part of system identification and many factors such as fittability, complexity, tractability, interpretability, biological plausibility and many more need to be considered. This chapter will give an overview of the main model classes used in neural system identification for vision as well as their respective advantages and restrictions.

Apart from their evident value for understanding and explaining phenomena in Vision Neuroscience, such models are of significant use for the design of experiments. Due to their predictive power, they can serve as *in-silico* representations of the real brain. Their advantage is that, other than the real organism, they do not pose strong restrictions on the setup, duration or other limiting factors of *in-vivo* experiments. Additionally, since every single one of their parameters is known, they can be analyzed precisely and also be reversed. This way, for instance, they allow to make informed decisions on what stimuli to choose for a certain desired manipulation of neural responses - a task that would be unfeasible in a real experiment if the full search space of possible pixel values of image stimuli was to be scanned in a random manner. The set of possibly insightful stimuli found by the *in-silico* model can then be used in the *in-vivo* experiment and tested, see Chapter 2.3. Using this approach, maximally exciting inputs (MEIs) for neurons have been found (*in-silico*) and validated (*in-vivo*) by various groups [26–30]. Chapter 5 uses this same method to synthesize stimuli that elicit high/low variability in the neural responses.

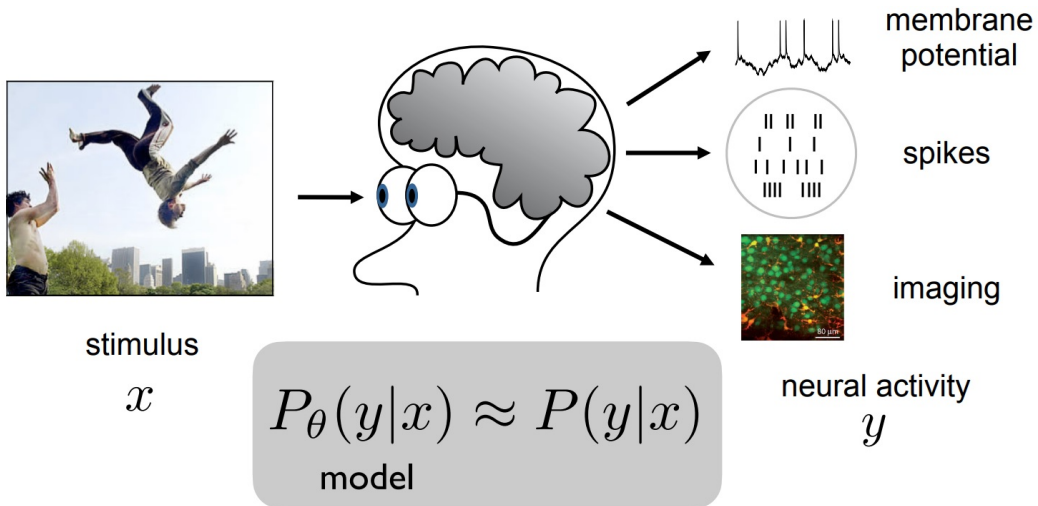


Figure 1.2: The distribution of neural activity provoked by a stimulus is approximated by a model with parameters θ . Source: Talk by Jonathan Pillow at CSHL Summer School "Computational Neuroscience Vision" 2022.

Spike triggered average models and white noise analysis These are arguably the most simple system identification models. The neuron’s receptive field is obtained by showing a stimulus of white noise over a period of time and then averaging the resulting responses, see for example [31–34]. The method works well for simple receptive fields in the early visual system but fails for higher areas.

General Linear Models As a simple parametric model, the General Linear Model models the firing rate λ as the stimulus weighted by a parameterized filter θ : $\lambda = \theta x$. The neural activity y is then simulated by a stochastic process with this firing rate. For neural data, often a Poisson process is chosen for which λ is both the mean and the variance: $y \sim \text{Poi}(\lambda)$. This results in the following encoding model:

$$P(y|x, \theta) = \frac{1}{y!} \lambda^y e^{-\lambda} \quad (1.2)$$

$$= \frac{1}{y!} (\theta x)^y e^{-\theta x} \quad (1.3)$$

The model can be fitted via maximum likelihood estimation, as is described in more detail in Chapter 2.1. In the case that the chosen stochastic process is in the exponential family, the fitted parameters can even be derived as a closed form solution. Given its assumptions of linearity and conditional independence across trials, this model class is a simple yet limited method in neural system identification - neurons are known to be non-linear and strongly correlated across trials.

Generalized Linear Models (GLMs)

This model class, introduced in 1972 by Nelder and Wedderburn [36], builds on and generalizes the General Linear Models. Nelder himself regretted the confusing terminology with respect to the General Linear Model in a 2003 Statistical Science interview. GLMs are thus also often referred to as Linear-Nonlinear Models which describes the fact that they add a non-linearity f to the linear filter of the General Linear Model: $\lambda = f(\theta x)$, $y \sim \text{Poi}(\lambda)$. The model can be extended to multiple neurons by adding coupling filters and the statistical dependency across trials can be taken into account by post-spike filters, see Figure 1.3. The resulting models have shown strong results and applications in Neuroscience, for instance [35, 37–41]. While the function space that GLMs can learn is still quite restricted, their great advantage is that their solutions can be easily interpreted and studied. Giving up this interpretability in exchange for a much richer and powerfull function space leads to modern Deep Neural Network models.

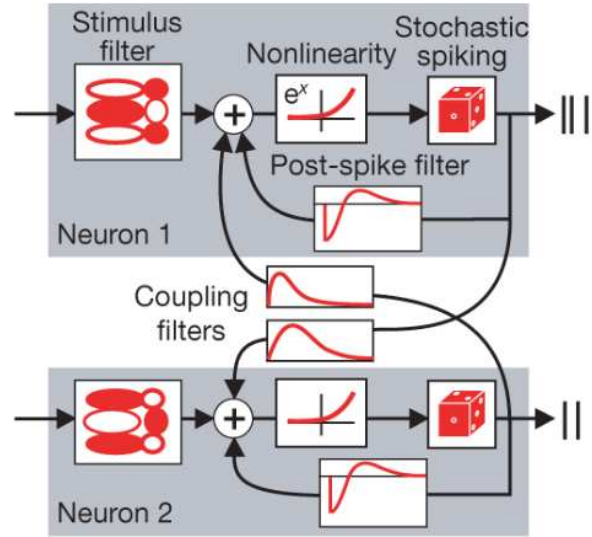


Figure 1.3: A multi-neuron GLM with post-spike and coupling filters. Source: Pillow et al. [35]

Deep Neural Network Models (DNNs) Stacking N layers of linear filters which are each passed through non-linearities yields a neural network model

$$\lambda = (f^N \circ f^{N-1} \circ \dots \circ f^1)(x), y \sim Poiss(\lambda) \quad (1.4)$$

$$f^n(\mathbf{z}) = f(\mathbf{W}^n \mathbf{z} + \mathbf{b}^n) \quad (1.5)$$

where the weights $\mathbf{W}$ and biases $\mathbf{b}$ are the parameters of the model. Again, any stochastic process can be used but the Poisson process is the most common choice. Note, however, that the project in Chapter 4 [24] questions this choice, arguing for alternative probabilistic distributions. For more details on the theory of neural networks, refer to Chapter 2.1.

Given that (big enough) neural network models are universal function approximators [42], they do not restrict the function space and can, in theory, learn any neural representation. An infinite function space, however, comes with the disadvantage that the true neural functions need to be found by fitting large amounts of data to the network - a requirement that is hard to fulfill in a neuroscientific setting where data is scarce and hard to acquire experimentally. This and their reputation to overfit on the noise of the too small datasets they were trained on often raises doubts about their practical use as system identification models in Neuroscience. These issues and doubts are addressed by the first project (Chapter 3, [23]) of this thesis. Finally, DNN models for neural system identification lack the interpretability of the models discussed in the previous sections, which is why they are often referred to as "black box" models.

Despite the reluctance of parts of the neuroscientific community towards them, deep neural network models have led to a surge of state-of-the-art system identification models that often outperform traditional models at predicting neural responses to visual stimuli [43–63]. Moreover, they have allowed the synthesis of stimuli which are tailored for specific response characterizations like strong neural firing (MEIs) [26–30] or invariances in the neural response [64–66]. The project in Chapter 5 contributes to this line of work by synthesizing stimuli that manipulate the variability in the neural firing rates. Finally, these models have provoked interesting comparisons between computer vision and biological vision, showing that the representations in task-driven models from tasks like object recognition resemble the representations in the primate visual system [67–73]. Such approaches can be beneficial for both communities since neuro models can help improve computer vision models and vice-versa. In addition, this offers interesting normative theories that computer vision and biological vision could be aligned by the same computational goals, thus resulting in similar functional representations. For instance, comparisons of the predictive performance for neural data between task-driven models and other models were done by Cadena et al. [69], Schrimpf et al. [74] highlighting the success of task-driven models for the primate visual system. However, Cadena et al. [75] showed that the same can not be claimed for rodents. The project in Chapter 3 [23] addresses this issue and provides an alternative modeling approach for the mouse visual system, building data-driven models that greatly outperform their task-driven counterparts.

For detailed reviews and outlooks of deep neural networks in Vision Neuroscience, refer to Kriegeskorte [76], Kietzmann et al. [77], Sinz et al. [78]. Kay [79] also offers a more critical opinion on the topic.

1.3 Own contribution

The following lists contain the main scientific contributions based on the Contributor Roles Taxonomy (CRediT) guidelines [80–82]. The guidelines were slightly adapted to allow for a better attribution of contribution in the case of method oriented research. The symbol * denotes equal contribution. Publications which earned the "Spotlight" distinction in their respective conferences are marked as such.

1.3.1 Publications and projects constituting a part of this thesis

[Spotlight]

- **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.

Initial idea: KKL and FS

Methodology and theoretical development: KKL, FS, EW, MB

Software: mainly KKL, Gaussian readout prototype by AJ and FS, pipeline help by KW

Method validation and experiments: KKL

Data collection and preprocessing: EW, TM, EC

Figures and visualization: KKL, method figure by MB

Writing (original draft): KKL, FS, EW

Writing (review and editing): KKL, FS, EW, MB, KW, SC

Funding acquisition: FS and AT

- **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, KKL and MB had an equal contribution to the project.

- **Konstantin-Klemens Lurz** and Fabian Sinz. Variably exciting stimuli for mouse primary visual cortex. *Project is not published (yet)*, 2024.

Initial idea: KKL

Methodology and theoretical development: KKL, FS

Software: KKL

Method validation and experiments: KKL

Data collection and preprocessing: publicly available data was used

Figures and visualization: KKL

Funding acquisition: FS

1.3.2 Other publications

[Spotlight]

- 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.

Contributions: KKL contributed to the software, discussions about methodology, writing, figures and editing.

- Suhas Shrinivasan, **Konstantin-Klemens Lurz**, Kelli Restivo, George Denfield, Andreas S Tolias, Edgar Y Walker, and Fabian H Sinz. Taking the neural sampling code very seriously: A data-driven approach for evaluating generative models of the visual system. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Contributions: KKL contributed to the software and review/editing.

- **Konstantin-Klemens Lurz***, Mohammad Bashiri*, and Fabian H Sinz. Bayesian oracle for bounding information gain in neural encoding models. In *NeurIPS Workshop on Information-Theoretic Principles in Cognitive Systems*, 2022.

Contributions: This is a workshop publication on an early version of [24]. For contributions see the respective publication in the section above.

- 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. KKL contributed to the competition metrics and writing.

2 Background

2.1 Artificial Neural Networks

Artificial neural networks (ANNs) belong to the field of Machine Learning (ML) and are a statistical tool for processing and analyzing data. Their original form, the Perceptron and Multilayer Perceptron (MLP) [87, 88] were inspired by the neural circuitry of the brain. Since then, research on ANNs has resulted in many novel architectures and network types, of which some like Convolutional Neural Networks (CNNs) [89, 90] or Recurrent Neural Networks (RNNs) [91] have drawn inspiration from neuroscientific findings. However, other architectures like the state-of-the-art transformer networks [92] have diverged from their biological counterparts. In general, while the inspiration from Neuroscience in the field of ML has certainly been fruitful, one must be careful when comparing the two because of many substantial differences on the computational, algorithmic and implementational level. Especially when applying ANNs in the field of Neuroscience, the conception of the model being a neural model of the brain can be misleading. In this thesis, ANNs are used as a data-precessing tool to model strictly only the *representations* that the brain extracts, not the computation or implementation. The word *neural* in ANN when modeling real neurons is here thus only coincidental.

Multilayer Perceptrons (MLPs)

"Neurons" or "nodes" in this network are non-linear processing units that take as an input the weighted sum of their connected neurons' output. After applying a non-linearity, they then pass the result to the next nodes. If a network only allows connections from one layer of nodes to later layers, it is called a "feed-forward" network. The weights between nodes are adaptive and can be learned during training as shall be described later. Given input x , the MLP then computes the output y as described in Equation 1.4. As for non-linearities, often functions like hyperbolic tangent, sigmoid or ReLu are chosen [94]. The layers between the input and output layers are called *hidden* layers and if a network has more than one hidden layer, it is considered *deep*. Such deep networks have proven to be very powerful but also hard to train giving rise to the ML branch of Deep Learning. For an overview, good practices and tricks refer for instance to the book "Deep learning" by Goodfellow et al. [95].

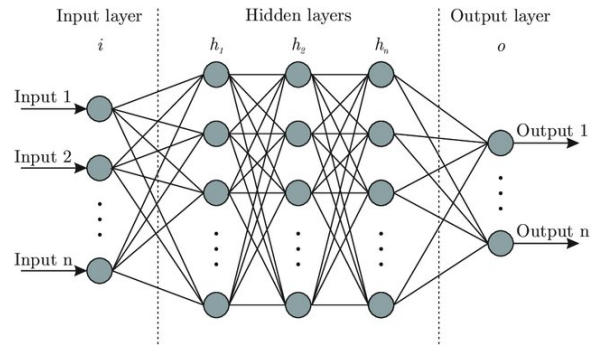


Figure 2.1: Example of a deep feed-forward Multilayer Perceptron. Source: [93].

Convolutional Neural Networks (CNNs) While MLPs are the most general ANN architecture and can, in theory, approximate any function [42], they are very inefficient at doing so and require large amounts of data to train their vast number of parameters. To overcome this, one can restrict the network’s function space to only contain function candidates that are reasonable for the type of data at hand. Such restrictions are called inductive biases and they can for example be used to exploit symmetries in the data. One such network architecture that exploits symmetries in image data is the Convolutional Neural Network (CNN) [89, 90]. An object in an image is the same, no matter where it is in the image. This property is called shift invariance and it is built into the CNN by applying the convolution operation. A weight kernel of relatively small size compared to the image size is convolved over the image, creating an output node that only looks at neighbouring pixels as opposed to the fully connected setup in an MLP. Importantly, the weights in the kernel are shared among all nodes, drastically reducing the number of learnable parameters. This way, the same filter (kernel) is applied at every location in the image instead of creating a new filter for each location. Often, CNNs contain several shared filters in order to extract different features of the input, resulting in multiple *channels*. Between the layers, a pooling operation is often applied in order to distill only the most relevant information. Models using CNNs have been extremely successful and have won many Computer Vision competitions. They were also one of the first ANN models to solve important commercial applications [95].

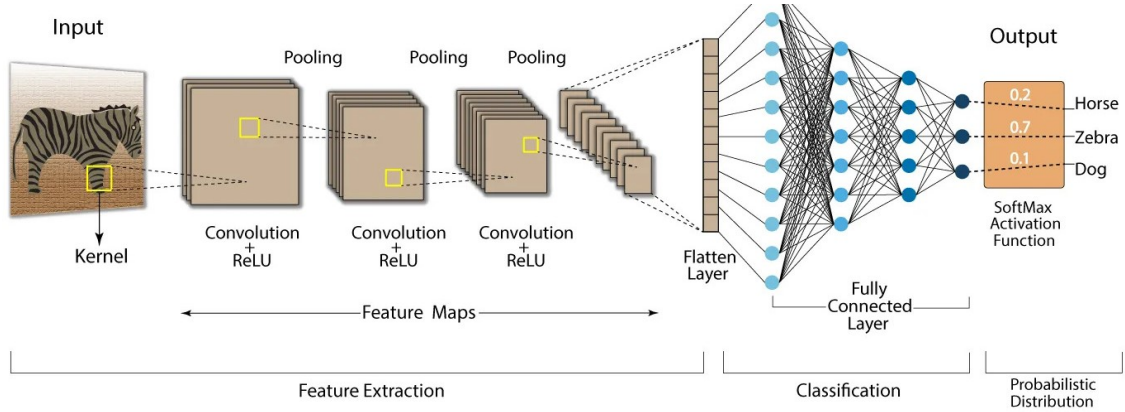


Figure 2.2: Example architecture of a CNN model used for object classification. The extraction of features from the input image is done by a deep CNN model. The flattened output is then fed into a fully connected MLP for classification. Source: [96]

Model training The process of adapting the parameters of the model is called model fitting or training. It is done by changing the parameters θ in such a way that the modeled likelihood $p_{\theta}(\mathbf{y}|\mathbf{x})$ (refer to Chapter 1.2) is maximized, i.e. explains the data best[95]. In a setup where the true values of y or *labels* are known, the learning is *supervised*. While other learning paradigms like *semi-supervised* or *unsupervised* learning have many applications, they are not used in the research of this thesis. In log space, finding the optimal parameters $\hat{\theta}$ then decomposes into a sum over the m

data points:

$$\hat{\theta} = \arg \max_{\theta} \sum_{i=1}^m \log (p_{\theta}(y^{(i)}|x^{(i)})) \quad (2.1)$$

The negative log likelihood is also referred to as the *loss* or *cost function*. and the choice of its form is essential for successfully fitting the model to the task at hand. In many regression problems, the mean squared error (MSE) is used which is a Gaussian loss with unit variance. For classification problems, on the other hand, the (binary) cross entropy is usually the choice of loss function, derived from the Binomial/Multinomial distribution. Neural responses have often been fitted using the Poission loss which has also been dealt with in Chapter 3. However, Chapter 4 challenges this choice, arguing for alternative probabilistic distributions.

For most problems, Equation 2.1 can not be solved analytically. It is then optimized in an iterative manner using techniques like stochastic gradient descent (SGD) as in the Adam optimizer [97]. The weights θ are then updated by passing the gradient of the loss function with respect to θ backwards through the network in a process called *backpropagation* [98].

In order to avoid *overfitting*, an issue where the over-parameterized model does not only fit the signal but also the noise in the data, Equation 2.1 can be extended by a regularization term. In the Bayesian framework, this is identical to adding a prior to the likelihood of the model, building in prior knowledge about the task. Another approach to tackle the problem of overfitting is *early-stopping* where model training on the training dataset is stopped when the loss on the held out validation dataset stops decreasing. In this thesis, both approaches are used in combination. The results of model performance are then always reported on a third held out dataset, the test set.

2.2 Neural system identification ANN architecture

The ANN models used for system identification in this thesis are based on the models introduced by [58, 60, 61, 75], see Figure 2.3. Conceptually, the model consists of four different parts: The *core* and *readout* build the main model and are essential for system identification. The *modulator* and *shifter* networks are optional additions to the model and help improving predictive performance by leveraging behavioral data collected during the experiment, see Chapter 2.4. They are only used in the models of Chapter 5.

Core The task of the core is to extract relevant features from the input image and to build meaningful non-linear representations for neural system identification. Since neurons are known to often perform similar computations and rely on similar features at different spatial locations [14], the core is shared across neurons, emerging as a collection of useful features or basis functions. It consists of a deep Convolutional Neural Network of three or four layers with 64 feature channels per layer where each convolution except for the first one is a depth-separable convolution [99]. The output of each layer is passed through a batch normalization and an ELU non-linearity [100, 101]. The exact hyperparameters of the core architecture were found in architecture searches on the validation dataset and differ slightly in the three projects.

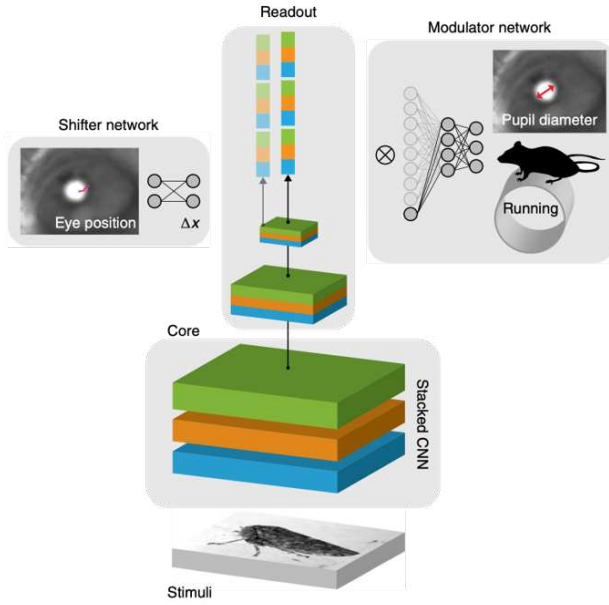


Figure 2.3: **ANN model which system identification in this thesis is based on.** The core extracts representations from the input and is shared across neurons. The readout then selects useful representations per neuron and combines them in a weighted sum to predict the neuron's response. Note that a novel readout architecture differing from the readout depicted in this figure is developed in Chapter 3 and used in subsequent chapters. The modulator and shifter networks are optional and only applied in Chapter 5. Source: Fabian H. Sinz (used in [26]).

Readout Every neuron only depends on a subset of the feature collection extracted by the core. Finding this selection per neuron and weighing the respective features is the task of the readout. Since the number of neurons in a dataset is in the order of 10^3 (refer to Chapter 2.4) and since there are many features to choose from ($channels * height * width$ from the output of the core), this task requires a large number of parameters - too many to be fitted with the limited data available. Several approaches to solve this issue have been proposed, among them the Gaussian readout developed in the first project [23] which shall be discussed in Chapter 3. Once the features are selected, the readout then computes a weighted sum and passes the result through a non-linearity to fulfill the requirements of the parameters of the model distribution. For $y \sim Poisson(\lambda)$, the non-linearity is an "ELU+1" to enforce λ to be positive.

Modulator During the recording of neural responses, the mouse is placed on a treadmill, allowing the simultaneous recording of its running speed. Also, the dilation of its pupils is measured. This behavioral data, apart from the stimulus, is predictive of the neural firing in visual cortical areas [102, 103]. The modulator network maps this data to a gain and offset value using a simple fully connected MLP. These values can then be used to alter the prediction of the stimulus-dependent model (core and readout).

Shifter Apart from running speed and pupil dilation, the pupil positions of the animal are recorded as well. This way, the model can be informed about possible gaze shifts during the experiment. Via a MLP, the shifter network maps the pupil positions to a positional shift in the output tensor of the core ($height * width$ dimension). The readout then reads out from the core using these corrected locations.

2.3 Digital twins and in-silico synthesis of stimuli

Many possibly interesting neuroscientific experiments are not feasible due to experimental limitations. For instance, animals (and humans) can only attend to a sequence of stimuli for a short time before they get tired and distracted. Also, when recording activity in a certain area of the brain, the neurons, synapses and activity of earlier areas leading to the target area are unknown. It is thus not clear what transformations the stimulus has been exposed to before being passed to the target neurons. These are constraints of classical "biological" Neuroscience where the subject is studied *in-vivo*. System identification models can help overcome some of these issues since they can be considered a *digital twin* of their biological counterpart and experiments can be run on them *in-silico*. The model does not have relevant time restrictions on how long stimuli can be presented without a break. Also, all weights and biases in the model are known exactly, making it possible to retrieve what information in the stimulus is eventually responsible for which aspect of the predicted neurons' spiking activity. If one tried to find the image-stimulus that drives a certain neuron the most and only allowed for 100×100 gray-scale stimuli of 10 different shades of gray, one would end up with 10^{10000} possible images - more than the number of stars in the observable universe ($10^{22} - 10^{24}$ [104]). Given that the parameters in an *in-silico* model are known, one does not need to try out all of these pixel combinations but can instead "reverse" the model to compute or optimize the pixel values that drive the target neurons the most. This has been done successfully by several groups [26–30].

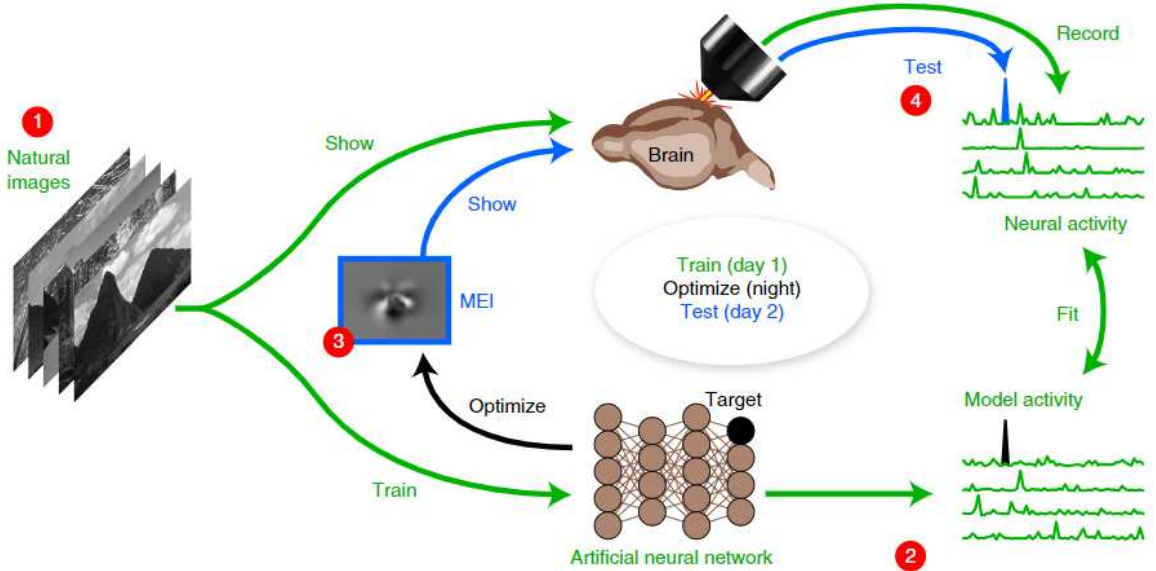


Figure 2.4: **Method for synthesizing and testing stimuli.** 1) Natural images are shown to the animal and neural responses to these images are recorded. 2) An ANN model is trained on these image-response pairs such that the activity predicted by the model best matches the recorded neural activity. 3) A stimulus is optimized such that it maximizes the predicted activity of a target neuron. The stimulus is called maximally exciting input (MEI). 4) The MEI is shown back to the animal and neural responses are recorded from the same neurons as in step 1). Source: Fabian H. Sinz (used in [26])

Walker et al. [26], for instance, synthesized the stimuli in the following way (see also Figure 2.4): They first recorded the neural responses to natural stimuli and trained a system identification model as digital twin of these neurons. Then, they initialized an image tensor with random noise pixels and passed it through the network whose weights were frozen. The prediction of the target neuron was then passed to a gradient ascent algorithm which optimized the input image to increase the neurons predicted firing rate in an iterative manner. Since one can not be sure whether the digital twin is a correct model of the biological neurons, the resulting maximally exciting inputs (MEI) were then tested *in-vivo*: The next day, the resulting synthesized stimuli were shown back to the animal and the responses of the same neurons as day 1 were recorded, showing that indeed the synthesized MEIs triggered the highest response of all tested stimuli in their respective target neurons. Chapter 5 uses this same method to synthesize stimuli with the goal to elicit high/low variability in the neural responses.

2.4 Data

All data used in the projects in this thesis was acquired by our collaborators at Toliaslab in Baylor College of Medicine, Houston. The experimental protocol followed the procedures described by Walker et al. [26]. In short, it consisted of a head-fixed mouse on a treadmill which was shown visual stimuli on a screen, and a two photon microscope to record the neural responses to these stimuli in the visual cortex. Note that while the data is described in the following paragraphs in a general way, some specifics of the data can vary between the projects.

Functional data The functional data consists of pairs of input stimuli and neural population responses. The input stimuli were usually gray-scale natural images which were sampled from ImageNet. The stimuli were then cropped and isotropically down-sampled to obtain a size of 64x36 pixels, corresponding to 0.53 pixels per degree of visual angle (ppd). The neural population responses to the presented visual stimuli were recorded via wide field two photon microscopy [105]. This technique does not measure the neural firing rates themselves but the fluorescence of the genetically encoded calcium indicator GCaMP6s. The calcium trace serves as a proxy for the firing of the neuron and was deconvolved [106] resulting in a value which is, in principle, only proportional to the true neural firing rate. For simplicity, this data will still be referred to as "the neural firing rate" throughout this thesis.

Behavioural data In addition to the stimulus-response pairs, behavioural data was collected during the experiments as well. It consists of the running speed of the head-fixed mouse as measured by the treadmill, the pupil dilation, and the change in pupil dilation over time. These three behavioural variables are known to influence neural firing in the visual cortex apart from the visual stimulus.

Anatomical data The two photon microscope records calcium traces in a three dimensional volume of tissue. From this spatial information one can assign each neuron a position (x, y, z) in the tissue. This location, together with functional data, can be

used to identify and match neurons which were recorded in two different scans. It can also be of value when using the relative positions of neurons to each other in order to predict receptive field locations, see Chapter 3.

Data size and data split Data can be recorded from an animal in several experimental sessions which can consist of several scans. While numbers can vary between scans, one scan usually roughly contains the responses of around 5000-8000 neurons to around 6000 image presentations. The data is split into a train, test and validation set along the image dimension. While the images in the train and validation sets were shown only once, the test set contains repeated presentations of stimuli (usually 10-20 repeats per stimulus). This way an estimate of the reliability of the neurons can be obtained.

2.5 Posterior predictive

While system identification models learn the distribution $p(y \mid x)$ of responses y conditioned on a stimulus x , in Chapter 4 a so called Gold Standard (GS) model is developed which has access to repeated trials of the *same* stimulus. The purpose of such a model is to estimate the upper bound of the performance that the system identification model can achieve. The GS model thus fits the distribution $p(y_{ijn} \mid \mathbf{y}_{i/jn})$ of an unobserved trial y_{ijn} given observed trials of the same stimulus $\mathbf{y}_{i/jn}$ (indexing: image, repeat, neuron). In Bayesian statistics, such a distribution is called the posterior predictive distribution [107]

$$\begin{aligned} p(y_{ijn} \mid D) &= \int_{-\infty}^{\infty} p(y_{ijn}, \theta \mid \mathbf{y}_{i/jn}) d\theta \\ &= \int_{-\infty}^{\infty} p(y_{ijn} \mid \mathbf{y}_{i/jn}, \theta) p(\theta \mid \mathbf{y}_{i/jn}) d\theta \\ &= \int_{-\infty}^{\infty} \underbrace{p(y_{ijn} \mid \theta)}_{\text{likelihood}} \underbrace{p(\theta \mid \mathbf{y}_{i/jn})}_{\text{posterior}} d\theta \end{aligned}$$

where θ are the distribution parameters and the last step assumes i.i.d. data, so $p(y_{ijn} \mid \mathbf{y}_{i/jn}, \theta) = p(y_{ijn} \mid \theta)$.

Gaussian case Here, the posterior distribution of a Gaussian likelihood is derived since the model in Chapter 4 uses a (Log)Normal distribution. The posterior predictive for other distributions from the exponential family can be derived analogously and the generalization to zero-inflated exponential family distributions is derived in the Manuscript of the original publication [24].

We model the responses of a neuron to a given image to come from a Normal distribution with mean μ_{in} and variance σ_{in}^2 , see Figure 2.5. For a Gaussian likelihood with unknown mean and variance, the conjugate prior (and thus the posterior) takes the form of a normal-inverse-gamma (NInvG) distribution such that the posterior predictive becomes

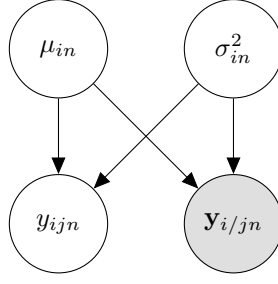


Figure 2.5: Graphical model of neural responses for a Gaussian likelihood.

$$\begin{aligned}
 p(y_{ijn} | y_{i/jn}) &= \int_{-\infty}^{\infty} \int_0^{\infty} \underbrace{p(y_{ijn} | \mu, \sigma^2)}_{Normal} \underbrace{p(\mu, \sigma^2 | \mathbf{y}_{i/jn})}_{NInvG} d\mu d\sigma^2 \\
 &= \int_{-\infty}^{\infty} \int_0^{\infty} N(y_{ijn}; \mu, \sigma^2) NInvG(\mu, \sigma^2; \mu'(\mathbf{y}_{i/jn}), \lambda'(\mathbf{y}_{i/jn}), \alpha'(\mathbf{y}_{i/jn}), \beta'(\mathbf{y}_{i/jn})) d\mu d\sigma^2.
 \end{aligned}$$

Here, the posterior hyperparameters μ', λ', α' and β' can be obtained from the prior hyperparameters $\mu_0, \lambda_0, \alpha_0$ and β_0 by

$$\begin{aligned}
 \mu'(\mathbf{y}_{i/jn}) &= \frac{\lambda_0 \mu_0 + n \bar{y}_{in}}{\lambda_0 + n} \\
 \lambda'(\mathbf{y}_{i/jn}) &= \lambda_0 + n \\
 \alpha'(\mathbf{y}_{i/jn}) &= \alpha_0 + n/2 \\
 \beta'(\mathbf{y}_{i/jn}) &= \beta_0 + \frac{1}{2} \sum_{j=1}^n (y_{i/jn} - \bar{y}_{in})^2 + \frac{n \lambda_0}{\lambda_0 + n} \frac{(\bar{y}_{in} - \mu_0)^2}{2}
 \end{aligned}$$

with $\bar{y}_{in}$ being the mean taken over the repeats dimension. The dependence of the posterior hyperparameters on $\mathbf{y}_{i/jn}$ and the indices of y_{ijn} will be omitted from now on for readability. Plugging in the expressions for the Gaussian and NInvG distributions, the posterior predictive distribution can then be evaluated as follows:

$$\begin{aligned}
 p(y | \mathbf{y}) &= \int_{-\infty}^{\infty} \int_0^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-\mu)^2}{2\sigma^2}} \frac{\sqrt{\lambda'}}{\sqrt{2\pi\sigma^2}} \frac{\beta'^{\alpha'}}{\Gamma(\alpha')} \left(\frac{1}{\sigma^2}\right)^{\alpha'+1} e^{-\frac{2\beta' + \lambda'(\mu - \mu')^2}{2\sigma^2}} d\mu d\sigma^2 \\
 &= \frac{\sqrt{\lambda'} \beta'^{\alpha'}}{2\pi \Gamma(\alpha')} \int_0^{\infty} \left(\frac{1}{\sigma^2}\right)^{\alpha'+2} \underbrace{\left(\int_{-\infty}^{\infty} \exp\left(-\frac{(y-\mu)^2 + 2\beta' + \lambda'(\mu - \mu')^2}{2\sigma^2}\right) d\mu \right)}_I d\sigma^2
 \end{aligned}$$

The integral over the mean μ gives

$$I = \int_{-\infty}^{\infty} \exp\left(-\frac{(y-\mu)^2 + 2\beta' + \lambda'(\mu - \mu')^2}{2\sigma^2}\right) d\mu$$

$$\begin{aligned}
&= \int_{-\infty}^{\infty} \exp \left(-\frac{y^2 + \mu^2 - 2y\mu + 2\beta' + \lambda'\mu^2 + \lambda'\mu'^2 - 2\lambda'\mu\mu'}{2\sigma^2} \right) d\mu \\
&= \int_{-\infty}^{\infty} \exp \left(-\frac{\mu^2 (\lambda' + 1) - 2\mu (y + \lambda'\mu') + \overbrace{y^2 + 2\beta' + \lambda'\mu'^2}^{C_1}}{2\sigma^2} \right) d\mu \\
&= \int_{-\infty}^{\infty} \exp \left(-\frac{\mu^2 (\lambda' + 1) - 2\mu (y + \lambda'\mu') + \frac{(y + \lambda'\mu')^2}{\lambda' + 1} - \overbrace{\frac{(y + \lambda'\mu')^2}{\lambda' + 1} + C_1}^{C_2}}{2\sigma^2} \right) d\mu \\
&= \int_{-\infty}^{\infty} \exp \left(-\frac{\left(\mu \sqrt{\lambda' + 1} - \frac{y + \lambda'\mu'}{\sqrt{\lambda' + 1}} \right)^2 + C_2}{2\sigma^2} \right) d\mu \\
&= \exp \left(-\frac{C_2}{2\sigma^2} \right) \int_{-\infty}^{\infty} \exp \left(-\frac{\left(\mu - \frac{y + \lambda'\mu'}{\lambda' + 1} \right)^2}{\frac{2\sigma^2}{\lambda' + 1}} \right) d\mu \\
&= \exp \left(-\frac{C_2}{2\sigma^2} \right) \sqrt{2\pi \frac{\sigma^2}{\lambda' + 1}} \\
&= \sqrt{2\pi \frac{\sigma^2}{\lambda' + 1}} \exp \left(-\frac{y^2 + 2\beta' + \lambda'\mu'^2 - \frac{y^2 + \lambda'^2\mu'^2 + 2\lambda'\mu'y}{\lambda' + 1}}{2\sigma^2} \right) \\
&= \sqrt{2\pi \frac{\sigma^2}{\lambda' + 1}} \exp \left(-\frac{y^2 \left(1 - \frac{1}{\lambda' + 1} \right) - 2y \frac{\lambda'\mu'}{\lambda' + 1} + \lambda'\mu'^2 - \frac{\lambda'^2\mu'^2}{\lambda' + 1} + 2\beta'}{2\sigma^2} \right) \\
&= \sqrt{2\pi \frac{\sigma^2}{\lambda' + 1}} \exp \left(-\frac{y^2 \frac{\lambda'}{\lambda' + 1} - 2y \frac{\lambda'\mu'}{\lambda' + 1} + \lambda'\mu'^2 - \frac{\lambda'^2\mu'^2}{\lambda' + 1} + 2\beta'}{2\sigma^2} \right) \\
&= \sqrt{2\pi \frac{\sigma^2}{\lambda' + 1}} \exp \left(-\frac{(y - \mu')^2 \frac{\lambda'}{\lambda' + 1} + 2\beta'}{2\sigma^2} \right)
\end{aligned}$$

The evaluation of the integral follows by realizing that this integral is over an unnormalized Gaussian with mean $\frac{y + \lambda'\mu'}{\lambda' + 1}$ and variance $\frac{\sigma^2}{\lambda' + 1}$. Substituting this result back into the posterior predictive, the integral over the variance can be evaluated by using the definition of the Gamma function $\int_0^\infty t^b e^{-at} dt = \frac{\Gamma(b+1)}{a^{b+1}}$ and by applying the change of variable $u = 1/\sigma^2$ with $d\sigma^2 = -du/u^2$, finally resulting in a Student's t-distribution.

$$\begin{aligned}
p(y \mid \mathbf{y}) &= \underbrace{\frac{\sqrt{\lambda'} \beta'^{\alpha'}}{\sqrt{2\pi (\lambda' + 1)} \Gamma(\alpha')}}_{C_3} \int_0^\infty \left(\frac{1}{\sigma^2} \right)^{\alpha' + 3/2} \exp \left(-\frac{1}{\sigma^2} \underbrace{\frac{(y - \mu')^2 \frac{\lambda'}{\lambda' + 1} + 2\beta'}{2}}_{C_4} \right) d\sigma^2 \\
&= -C_3 \int_\infty^0 \frac{u^{\alpha' + 3/2}}{u^2} \exp(-u C_4) du
\end{aligned}$$

$$\begin{aligned}
&= C_3 \int_0^\infty u^{\alpha'-1/2} \exp(-u C_4) du \\
&= C_3 \Gamma(\alpha' + 1/2) C_4^{-(\alpha'+1/2)} \\
&= \frac{\sqrt{\lambda' \beta' \alpha'}}{\sqrt{2\pi}(\lambda' + 1)} \frac{\Gamma(\alpha' + 1/2)}{\Gamma(\alpha')} \left((y - \mu')^2 \frac{\lambda'}{2(\lambda' + 1)} + \beta' \right)^{-(\alpha'+1/2)} \\
&= \frac{\sqrt{\lambda' \alpha'}}{\sqrt{2\pi \alpha' \beta' (\lambda' + 1)}} \frac{\Gamma(\alpha' + 1/2)}{\Gamma(\alpha')} \left(\frac{(y - \mu')^2}{2\alpha' \frac{\beta'(\lambda'+1)}{\alpha' \lambda'}} + 1 \right)^{-(\alpha'+1/2)} \\
&= \frac{1}{\sqrt{2\pi \alpha'}} \frac{1}{\sqrt{\frac{\beta'(1+1/\lambda')}{\alpha'}}} \frac{\Gamma(\alpha' + 1/2)}{\Gamma(\alpha')} \left(\frac{(y - \mu')^2}{2\alpha' \frac{\beta'(1+1/\lambda')}{\alpha'}} + 1 \right)^{-(\alpha'+1/2)} \\
&= t_{2\alpha'} \left(y; \mu', \frac{\beta'(1+1/\lambda')}{\alpha'} \right)
\end{aligned}$$

3 Generalization in data-driven models of primary visual cortex

The content and figures of this chapter are based on the following publication:

- **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.

3.1 Motivation

A common criticism of ANN models in the field of neural system identification is their high (over)parameterization and the resulting ability to fit the noise in the data on top of the signal. Since they are relatively unrestricted and can exploit a large function space compared to other model classes (refer to Chapter 1.2), they can, in theory, pick up on undesired peculiarities in the dataset which are predictive for the data at hand but do not give insights in the general process of biological vision. For example, an individual animal could be in an aroused state during the recording or it could have developed specific adaptations to certain visual stimuli due to its individual life experience. A normative theory of Vision Neuroscience, however, would aim to understand the general process of vision as it is performed by a whole species. One way to examine whether a model generalizes to a whole species, is to obtain the representations in the model from a source that is independent of the neural data of the animal that the model is tested on. Yamins and DiCarlo [67] achieved such generalizing representations for the monkey visual cortex using a *task-driven* approach by a method called *transfer-learning*: They trained the core (the part of the network that contains the representations, see Chapter 2.2) on the task of object classification on ImageNet, transferred it to their neural system identification model and, using their neural data, trained a readout on top of these representations. Apart from acquiring a generalizing model of certain areas in the monkey visual cortex, this also offered the normative perspective that machine vision and biological vision could be aligned by common computational goals. These computational goals, however, may not be shared with all species, as became apparent when Cadena et al. [75] showed that, as opposed to the monkey experiments, the task-driven approach did not lead to predictive models for the mouse visual cortex - at least for the tasks available in the field of computer vision at the time. It was thus the goal of this project to close this gap in knowledge and obtain generalizing representations also for the mouse visual cortex.

3.2 Methods

The approach used in this project to obtain generalizing representations was *data-driven*. Contrarily to the task-driven approach, the representations were trained on neural data, as was the readout. However, in order to insure the independence of the neural data of the test animal which the readout was trained on, the representations in the core were obtained from neural data of different animals of the same species. Since neural data is much more scarce than task data like ImageNet, the following challenges had to be addressed in order to train the data-driven representations: The datasets needed to be enlarged and augmented in order to obtain the biggest possible data size. And the model architecture had to be as efficient as possible, to make optimal use of the available data.

Data-stitching Due to experimental restrictions, datasets could only be recorded to contain up to a certain number of image-response datapoints, see Chapter 2.4. In order to obtain bigger datasets, data from several experimental sessions of the same animal had to be stitched together. This was done by recording the anatomical positions of the neurons in the cortex along with their functional data, in order to later match identical neurons from different experimental sessions. This way, up to four times more datapoints could be recorded per neuron than would otherwise have been possible.

Gaussian readout architecture

In order to make the best use of the available data and push all non-linear computations into the generalizing representation of the core, the readout needed to be as lightweight and low-parameterized as possible. Klindt et al. [60] had already decreased the number of parameters from 10^9 of a naive fully connected readout ($n_n * n_f * w * h$) to 10^7 of their *factorized* readout ($n_n * (n_f + w * h)$). Here, n_n is the number of neurons, n_f the number of features, and w and h the width and height of the tensor block in the core.

Here, a novel readout architecture was introduced, called the Gaussian readout, which drastically reduces

the number of parameters again. It does so by learning only a single location in the feature block via bilinear interpolation [61]. The spatial position is modeled as a 2D Gaussian distribution from which, during training, the position is sampled. During inference, the mean of the distribution is taken. This method then allowed to cut down the number of parameters ($n_n * (n_f + 7)$) to 10^5 , because it only needs two parameters for the the mean, four for the variance, and one for the bias. Since the area V1 preserves the topology of space, a phenomenon called retinotopy, the readout can additionally be informed by a *readout position network*. This network maps the

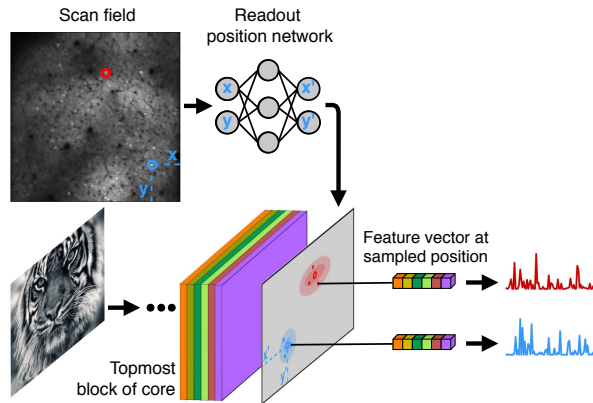


Figure 3.1: **Gaussian readout architecture.**

The feature vector is read out from a single position of the core. This location is obtained by learning a Gaussian distribution over likely positions which is additionally informed via retinotopy by the readout positional network.

recorded anatomical positions of the neurons (refer to Chapter 2.4) to positions in the feature block of the core. The mapping can be learned shared across neurons making it very efficient and cheap in terms of number of parameters.

Experimental paradigm As opposed to the transfer-learning in the task-driven approach, the representations in the data-driven setup were obtained via multi-task learning, where each neuron is one task and the core network is shared across all neurons. However, in order to make sure that the representations truly generalize, the neural data for training the core needed to be independent of the neural data that the full network was finally tested on. This was insured by the following experimental setup: The full network (core + readout) was trained on datasets from several animals and sessions. Subsequently, the weights of the core were frozen and the readout was dropped. On top of the frozen core, a new readout was then finally trained on the data of a different test animal and its generalization performance was reported as fraction oracle [26], see Figure 3.4. The experiments in Figure 3.3 have similar paradigms.

3.3 Results

With the Gaussian readout for increasing data efficiency at hand and using the data-stitching method for expanding the size of the datasets, the data-driven approach for obtaining generalizing representations could be tested. As a first step, the regular predictive performance (no generalization) of the new readout was investigated by comparing it against the state-of-the-art factorized readout. Then, the Gaussian readouts ability to push the non-linear computations to the core and produce features that generalize between neurons of the same animal was tested. Finally, it was verified that the learned representations even generalize to other animals of the same species.

Standard predictive performance The Gaussian readout outperforms the factorized readout for various settings and setups, making it the new state-of-the-art, see Figure 3.2. The training was done in an end-to-end fashion, meaning that the full network was trained on the same set of neurons and no generalization ability towards new neurons was measured. Instead, only the predictive performance to new stimuli was tested. It surpasses the performance of the factorized readout in the regimes of both low and high number of neurons. Also when sharing the features of neurons that were matched between different sessions of the same animal (see "data-stitching" in Chapter 3.2) the Gaussian readout performed best.

Generalization within one animal In order to test whether the Gaussian readout not only performs better on standard neural prediction but also allows the generation of generalizing representations, the following experiments were conducted: Models were trained on large stitched datasets of a mouse but tested on different neurons of the same mouse, see Figure 3.3. For some models the core was trained on variable amounts of data as indicated by the x-axis but the readout was trained on the full data (purple lines). To other models (yellow lines) the full data was given for training the core but the readout was fitted using only subsets of the data. Comparing the two experiments shows that the fastest increase of performance for the Gaussian readout

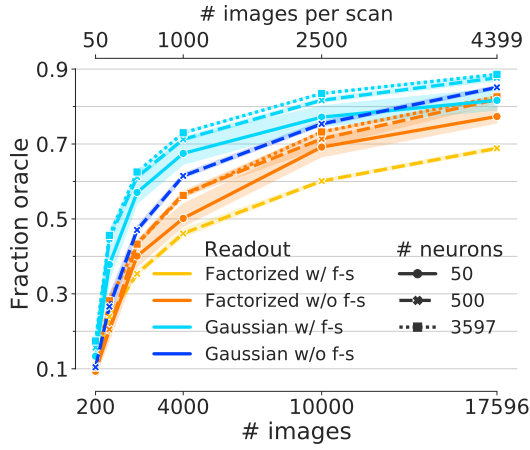


Figure 3.2: **Performance of end-to-end trained networks.** Models using the Gaussian readout outperform the former state-of-the-art factorized readout in all settings: with and without feature sharing between matched neurons of different scans and for low and high number of neurons. Shaded areas depict 95% confidence intervals across random picks of the neuron subsets.

can be observed in the *best-core/diff-readout* setup. This shows that 1) the Gaussian readout needs little data to be trained, leaving the data to be used in the core and 2) given a well trained core, the models using a Gaussian readout can generalize to unseen neurons in the same animal. This gap between the yellow and purple line is not nearly as prominent in the factorized readout.

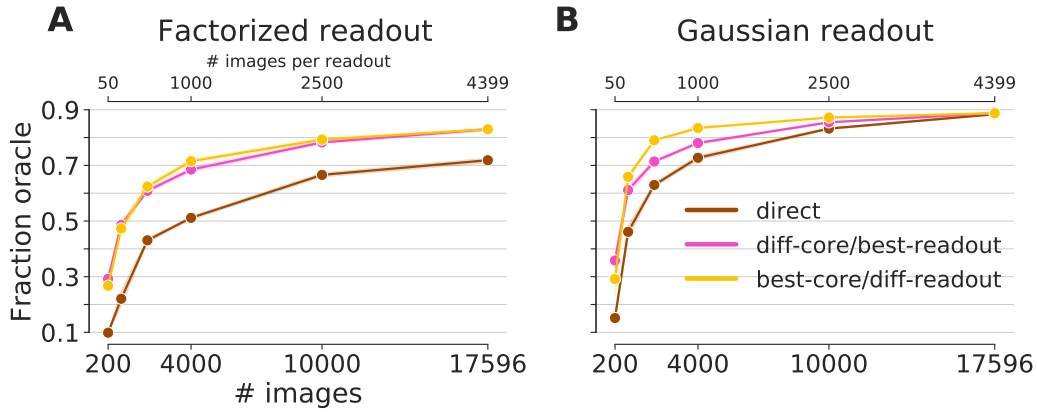


Figure 3.3: **Generalization to other neurons in the same animal.** Purple lines: Models whose core was trained on various amounts of data (indicated on the x-axis) and whose readout was trained on the full data (17596 datapoints). Performance measured on new held out neurons (generalization). Yellow lines: Vice versa. Brown lines: End-to-end trained models (compare Figure 3.2) as baseline, no generalization. The gap between the yellow and purple curve for the Gaussian readout indicates that it is data efficient, pushing computations to the core. The gap is not prominent in the factorized readout.

Generalization across animals Finally, the generalization ability of these models to unseen animals of the same species was tested, see Figure 3.4. To this end, models were trained on many datasets of various animals and sessions using the Gaussian readout. The frozen representations were then transferred to a model of an unseen test animal whose data was only used to fit the new readout (*best-core/diff-readout* condition from Figure 3.3). For comparison, a model was trained directly end-to-end on the training data samples of the test animal (blue line). The models trained on many animals not only matched the overall performance of the direct model but greatly exceeded it by up to 11% (yellow line, 11 datasets), showing that truly general

representations of mouse V1 have been achieved. Moreover, the performance of the direct model was achieved with only 40% of the data. Additionally, the findings from Cadena et al. [75] that task-driven models like VGG16 do not yield good representations for mouse V1 could be confirmed (black dotted line). Finally, a comparison between data consisting of many neurons but few samples (red line) and few neurons with many datapoints (green line) showed only a slight advantage of the latter.

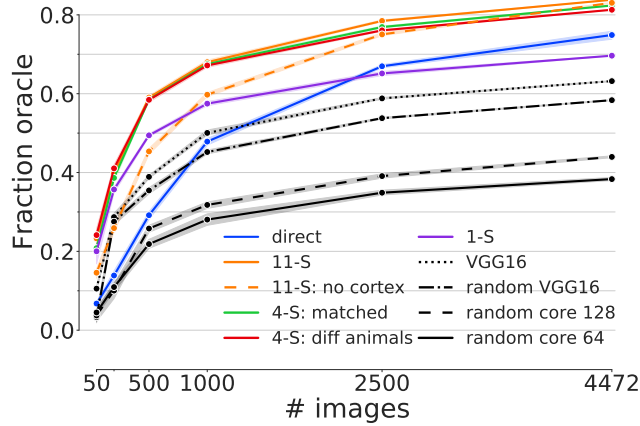


Figure 3.4: **Generalization to neurons in other animals of the same species.** Direct: End-to-end trained on test animal. 11-S: Contains 11 datasets from various animals. 4-S: 4 datasets from the same vs different animals. 1-S: Dataset comparable in size to the test animal (direct). VGG16: Task-driven. Random: Models with random features.

3.4 Discussion

An important aspect of scientific statistical modeling is the extend to which previously unseen data can be explained. Success in doing so is a strong indicator that the model has truly captured the underlying characteristics of the modeled process. Especially when using Deep Artificial Neural Networks for modeling, this aspect becomes essential due to their high expressiveness, relatively unrestricted function space and ability to overfit on the data.

In Vision Neuroscience, deep ANNs have been shown to have such generalizing properties for the monkey visual cortex when they were trained on a task like object classification [67, 69]. Since current task-driven representations do not generalize to the mouse visual cortex [75], a data-driven approach was suggested here. To this end, a new readout architecture was introduced with the aim of making the model more data-efficient. Additionally, a method of creating bigger datasets was developed by which datasets of different sessions were stitched together matching the respective neurons with anatomical and functional data.

The resulting model sets a new state-of-the-art for the prediction of neural responses to natural stimuli in mouse V1, outperforming both task-driven networks as well as older data-driven models. The so obtained representations not only generalize to other neurons in the same animal but also to completely new individuals of the same species. This suggests that these representations are truly characteristic for mouse primary cortex. Given that the model is both high performing as well as data-efficient, it offers great future possibilities for applications in closed loop experiments such as [26, 27]. Also, further analysis of the generalizing representation in the future in order to investigate the operations of visual processing could lead to helpful insights for understanding the underlying mechanism of biological vision. Last but not least, the potential benefit of the trained representation in computer vision tasks could be investigated. To this end, the model code and weights were shared publicly together

with the data of the test animal.

A limiting factor of this study is that all representations were trained solely on neural data consisting of deconvolved calcium traces integrated over 500 ms. Whether the results hold for other types of neural recordings, different time scales or dynamic responses is not clear. Also, the current lack of well performing task-driven representations is most likely not due to the approach itself but rather the choice of tasks as pointed out by [75]. Future transfer models trained on new types of tasks might better align with the computational goals of mouse vision.

4 Bayesian oracle for bounding information gain in neural encoding models

The content and figures of this chapter are 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.

4.1 Motivation

Many recent system identification models in Vision Neuroscience such as [26, 43, 48, 57, 58, 63, 69] and also Chapter 3 [23] focus on modeling the conditional mean of the neural response distribution given a stimulus. While it is reasonable that the mean as the first moment of the distribution should carry a major part of the information processed by the system, several normative theories hypothesize that higher moments play an important role as well. For instance, the Bayesian brain hypothesis [108], neural sampling [109, 110] or probabilistic population codes [108] all include the conditional variance of neural responses given a stimulus into their framework. They all rely on this variability or even make testable predictions about it [111–113].

From the experimental side, there is strong evidence that the conditional variance is not simply equal to the conditional mean, as would be assumed by a Poisson modeling approach, but that there is a more complex relationship, see Wei et al. [114]. The authors suggest to model neural responses from deconvolved calcium trails with higher parameterized distributions like the gamma distribution and propose to add a zero-inflation in order to account for the large number of zero responses. The result is then a mixture model such as the Zero-Inflated Gamma distribution (ZIG).

With both theory as well as data pointing towards investigating stimulus conditioned variance, there is a need for models that account for more complex response distributions such as the ZIG distribution. However, while progress towards such models has been made [62], it is not clear how to accurately measure their predictive performance and how to evaluate them. The choice of metric for mean-focused models is correlation which, coherently, is based on the mean. In order to get more accurate estimates of performance in the presence of noisy data, this correlation measure is then usually corrected by normalizing it with respect to an estimated upper bound, the oracle estimator [23, 61]. Often, this estimator is obtained in the form of a point estimate from the responses to repeated presentations of the same stimulus.

^{*}Equal contribution

However, when using models of distributions like ZIG, mean-based metrics are no longer meaningful and a metric which is based on the full likelihood is required. Such a metric is Normalized Information Gain (NInGa) [115] which, just like the oracle estimator for correlation, involves estimating an upper bound of performance. Obtaining such upper bound estimates, however, is challenging due to the noisiness and sparsity of neural responses and limited dataset sizes. This project thus aims at finding a solution to obtaining upper bound estimators of the full likelihood which are robust despite the challenges posed by the complexity of neural data. It also focuses on providing a derivation of the general mathematical framework of such estimators for a variety of distributions that are relevant for neural data.

4.2 Methods

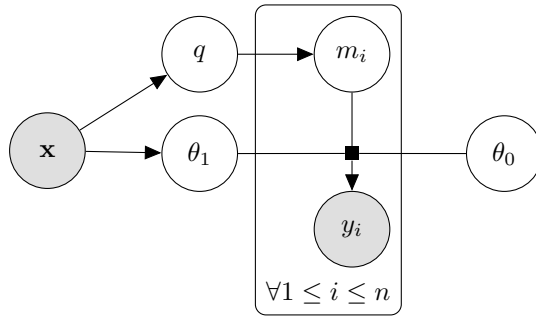
Zero-inflated distributions These distributions are currently the state-of-the-art for likelihood-based encoding models [62, 114]. They are mixture distributions consisting of a uniform distribution to model zero responses and another density for the positive part, e.g. Gamma or LogNormal. The uniform distribution is then defined in the interval $[0, \tau)$ and the positive distribution is shifted to $[\tau, \infty)$ with τ being a small positive real number. With mixing proportion $q(x)$ and parameters of the positive distribution $\theta_1(x)$, the model density $\hat{p}(y|x)$ can then be defined as:

$$\hat{p}(y|x) = (1 - q(x)) \cdot \underbrace{p_u(y)}_{\text{uniform}} + q(x) \cdot \underbrace{p(y|\theta_1(x))}_{\text{positive distribution on } [\tau, \infty)} \quad (4.1)$$

Both q and θ depend on the stimulus x and can be predicted from the stimulus by an encoding model, see also the graphical model in Figure 4.1.

Figure 4.1: **Graphical model for a zero-inflated distribution.**

The firing probability q and parameters θ_1 of the positive distribution are determined by the stimulus x . Whether a neuron actually fires on a particular trial i is then induced by a Bernoulli random variable m_i . If $m_i = 1$ a response is drawn from $p(y_i|\theta_1)$, otherwise from $p(y_i|\theta_0)$.



Normalized Information Gain With $p(y|x)$ being the distribution of a neuron's response y to a stimulus x , Normalized Information Gain (NInGa) [115] measures a model's performance in terms of likelihood $\hat{p}(y|x)$ by setting it on an interpretable scale between a lower bound $p_0(y)$ and an upper bound $p_*(y|x)$. The upper bound is also referred to as the *Gold Standard (GS)* distribution and the lower bound as the *Null* distribution.

$$\text{NInGa} = \frac{\langle \log \hat{p}(y | x) \rangle_{y,x} - \langle \log p_0(y) \rangle_{y,x}}{\langle \log p_*(y | x) \rangle_{y,x} - \langle \log p_0(y) \rangle_{y,x}} \quad (4.2)$$

The Null distribution should only reflect basic aspects of the responses and can easily be estimated. Here, a marginal distribution $p_0(y)$ without any information about the stimulus is chosen. The Gold Standard distribution, however, should be as close as possible to the true response distribution $p(y|x)$, making it very challenging to be estimated. Here, it is obtained comparably to the oracle estimator for correlation by giving it access to repeated presentations of the same stimulus, information that the model under evaluation does not have. Importantly, this is done in a leave-one-out manner, meaning that for n repeated trials the GS model for target repeat i has access to $n - 1$ trials for the estimation of its parameters.

Point Estimate (PE) for GS model In line with the computation of the oracle estimator for correlation, the parameters of the GS model can be obtained from the $n - 1$ left-out repeats by computing point estimates via a function f ,

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

which is equivalent to moment matching for the distribution class investigated here (zero-inflated exponential family). For simplicity, the conditioning of the GS model on the stimulus x will be dropped from here on.

Posterior predictive for GS model The full Bayesian expression for obtaining the GS model adds uncertainty over the parameters via the posterior predictive distribution:

$$p_*(y_i | \mathbf{y}_{\setminus i}) = \int_{-\infty}^{\infty} \underbrace{p(y_i | \theta)}_{\text{likelihood}} \underbrace{p(\theta | \mathbf{y}_{\setminus i})}_{\text{posterior}} d\theta \quad (4.3)$$

This is a more general description of $p_*(y_i | \mathbf{y}_{\setminus i})$ which collapses to the point estimate if $p(\theta | \mathbf{y}_{\setminus i}) = \delta(\theta - f(\mathbf{y}_{\setminus i}))$.

The integral in Equation 4.3 is generally intractable. It can, however, be solved analytically for distributions from the exponential family, as is shown for the example of the (Log)Normal distribution in Chapter 2.5. Using a distribution form this family for the positive part of a zero-inflated distribution, the full posterior predictive then boils down to numerically solving a one-dimensional integral over q in the range $[0, 1]$:

Lemma 1. *The posterior predictive distribution of a zero-inflated distribution as defined in Equation 4.1 is given by*

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

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

Proof. See the Appendix of the paper [24] in the Manuscripts. □

4.3 Results

This section first demonstrates experimentally that the point estimate ("PE") approach is not sufficient to estimate the upper bounds of a Zero-Inflated-LogNormal (ZIL) likelihood in the NInGa metric and that instead, the full posterior predictive ("Bayesian") is necessary. To investigate the reasons for this, PE and Bayesian are then compared in multiple ways in simulations to show in what respects they differ and how they compare with respect to complexities in the data such as signal-to-noise-ratio (SNR) or dataset size. Finally, the influence of the prior in the Bayesian case is highlighted and a system identification model of real neural responses is evaluated using the NInGa metric with the Bayesian upper bound estimator.

Point estimates are overconfident The PE approach is very sensitive to the sparsity of the data, see also Figure 4.2. Combined with the small dataset size containing only few repeats per stimulus, this results in the PE model underestimating the probability of a neural response being, for example, close to but greater than τ (bottom right of Figure 4.2). Since the PE approach does not attribute any uncertainty to its estimation of the GS parameters, it is overconfident given that the data can be complex and sparse. The result is that the upper bound estimation by the PE can be lower than the lower bound estimation of the Null model (black dots in the red triangle of Figure 4.2).

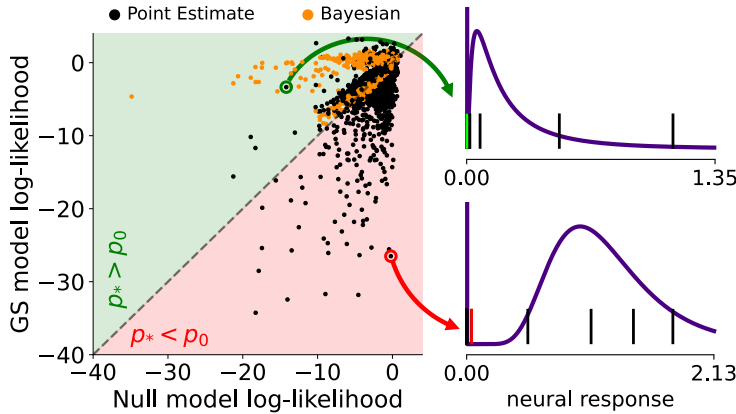


Figure 4.2: **Comparison of lower and upper bound estimates per neuron.** For many neurons, the PE is overconfident, resulting in the Null model outperforming the GS model. Two example neurons (right) demonstrate where the PE method fails (red) or succeeds (green).

Analysis of PE vs. Bayesian GS model The two methods for obtaining a GS model are compared in Figure 4.3 in more detail. Panel **a)** shows that the GS estimates of both ZIL moments μ and σ^2 profit from the Bayesian treatment. However, higher moments seem to be even less robust as a PE than lower moments, which is reflected in the higher increase in performance when estimating the respective parameter with the posterior predictive (orange to red vs. orange to light blue).

The Bayesian GS model is also much more data efficient than the PE approach, which can be seen in panel **b)**. The right side shows how decreasing the number of left-out repeats for the GS model results in a drop of GS likelihood. However, when estimating both parameters via the Bayesian method, this drop is far less pronounced, making it thus more robust to small dataset sizes, one of the main challenges of neural data besides sparsity. The left side extends this analysis with the use of simulated data to even higher numbers of left-out-repeats. The Bayesian method outperforms the PE method in all dataset sizes, while being consistent with the results of panel **a)**,

namely that the estimation of μ relies less on it than σ^2 . This offers an explanation for the question why the PE method works well in practice for the oracle estimator of correlation where only the mean is estimated. Finally, in panel **c**) simulated data is used to investigate the effect of the signal-to-noise-ratio (SNR) on both GS estimation methods. Again, the Bayesian method outperforms the PE method for all regimes of SNR, making it robust to differing levels of noise in the data.

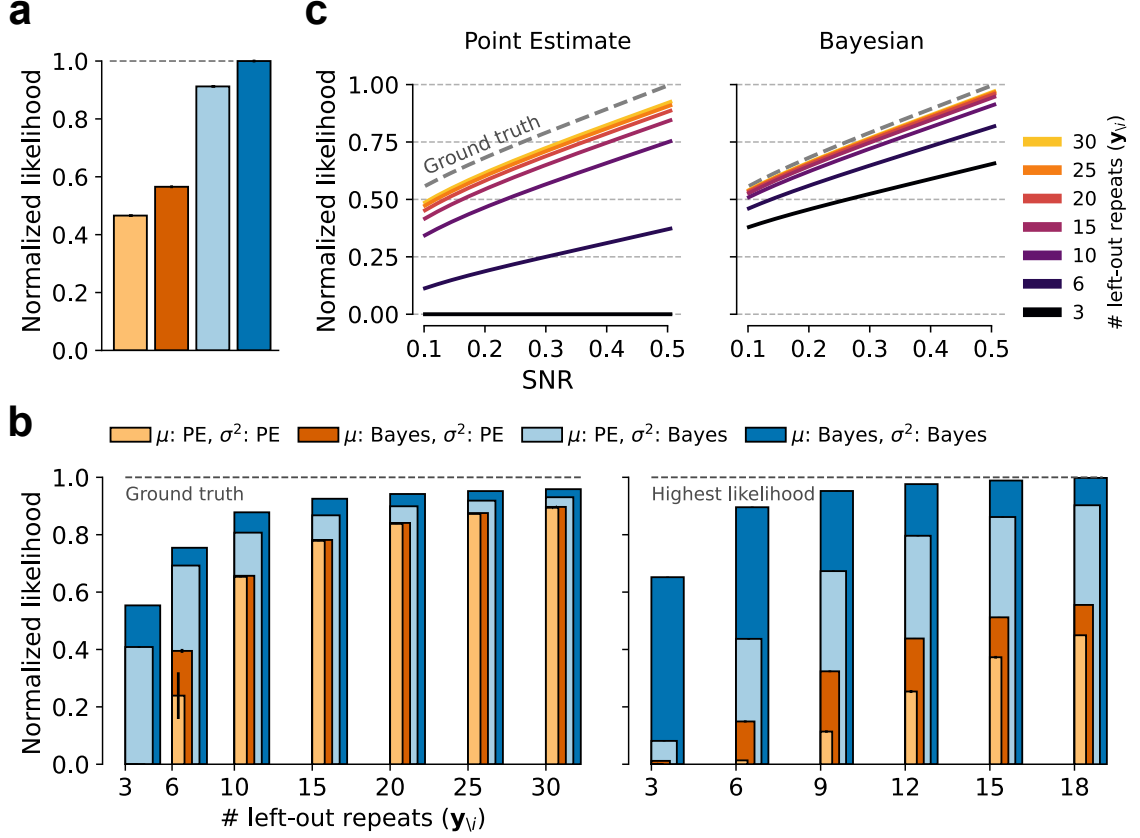


Figure 4.3: **Comparison of the PE and Bayesian GS model.** Panel **a**: Comparison of different GS models where the individual parameters are either estimated via the PE or the Bayesian approach. The number of left-out repeats $y_{\setminus i}$ is 19. Colors are the same as in **b**. Normalized w.r.t. the max. likelihood value, i.e. the dark blue bar. Panel **b**: Similar to **a** but for different numbers of left-out repeats $y_{\setminus i}$. **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 $y_{\setminus i}$. **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 $y_{\setminus i}$.

Evaluation of encoding model After determining that the Bayesian method yields robust estimates of the likelihood upper bounds, an encoding model could finally be evaluated using NInGa. The model was built according to Chapter 2.2 without modulator or shifter. The readout of Chapter 3 was used and slightly changed such that it predicted the three parameters of the LogNormal likelihood (q , μ and σ^2)

instead of the one parameter (λ) of the Poisson likelihood. The results can be seen in Figure 4.4. Panel **a**) shows that the performance of the encoding model exceeds the GS model, which is due to a poor choice of prior: Up to this point, the prior hyperparameters had been chosen independently of the neurons based on the statistics of the data. It was evident that the prior had to be chosen per neuron instead and was subsequently fitted on the sum of leave-one-out GS likelihoods, resulting in a more conservative upper bound estimate which can be seen in panel **b**). The same type of experiment as in Figure 4.3 was then performed on the encoding model, showing that the encoding model predicts all parameters of the distribution similarly well. Finally, the performance of the encoding model could be evaluated to be 90% NInGa in panel **c**). Additionally, by comparing the gray bars, it could be identified that the largest performance gain for future models can be achieved by improving the prediction of the parameter q .

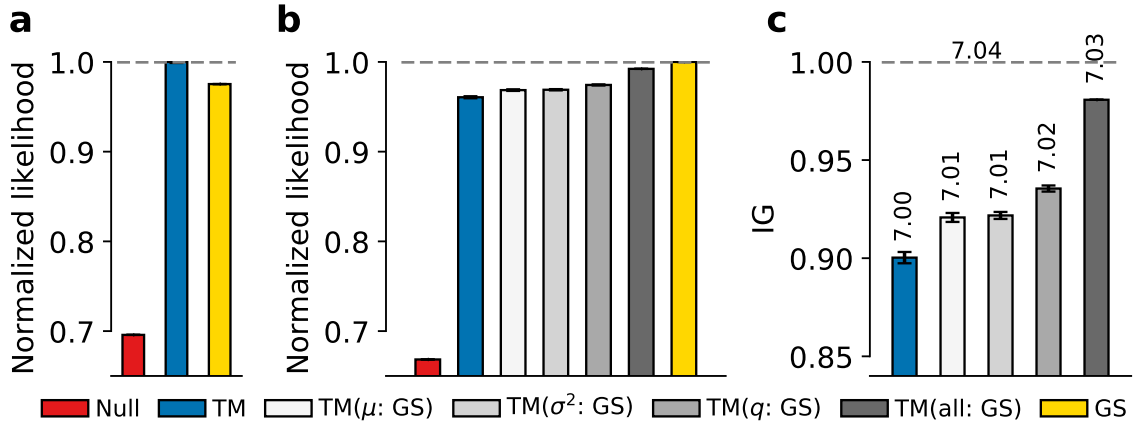


Figure 4.4: **Evaluation of ZIL encoding model on real neural data using the Bayesian GS model.** Panel **a**: A sub-optimal prior for the GS model results in a lower performance than the trained model (TM). Normalized w.r.t. the max. likelihood value, i.e. the blue bar. Panel **b**: The GS model outperforms the TM when its prior was optimized. The TM estimates all parameters similarly well. Gray scale colors indicate models for which all parameters are estimated by the TM except for the parameters in parentheses. Normalized w.r.t. the max. likelihood value, i.e. the yellow bar. Panel **c**: Normalized Information Gain (NInGa) for the TM and the gray 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.

4.4 Discussion

Modeling neural response properties beyond the simple mean of the firing rate can potentially provide valuable insights in the underlying processes of neural computation. Several normative theories of these processes like the Bayesian brain hypothesis [108], neural sampling [109, 110] or probabilistic population codes [108] rely on or make predictions of not just the mean neural activity conditioned on a stimulus but also the variability around it. While progress has been made towards building models that aim at capturing the full complex neural response distributions [62, 114], it was not clear how to accurately evaluate the performance of these models: Normalized Information Gain (NInGa), the full-likelihood alternative metric to traditional correlation-based metrics, involves the estimation of upper performance bounds. In their paper, Kümmerer et al. [115] estimate NInGa as a non-parametric model for the prediction of human fixation locations on images. In the context of such data, upper bounds can be obtained relatively easily as the estimation of the Gold Standard model converges already with relatively few data points.

This project, however, highlighted the challenges that come with estimating such upper bounds for complex data such as neural activity: Using conventional point estimates, the limited data size combined with the sparsity of neural responses led to an overconfident point estimate of the parameters of the GS model. A solution was provided which involves obtaining the parameters by computing the full posterior predictive distribution, thereby lowering overconfidence and adding an informative prior. The mathematical framework to compute this posterior predictive was derived for a variety of distributions, among them the state-of-the-art zero-inflated exponential family distributions [62, 114]. Finally, the upper bound estimates were tested in simulations as well as using real data and were found to be data-efficient and robust against different levels of noise. Using the so obtained upper bounds, an encoding model of real neural responses could then be evaluated to perform at 90% NInGa (Normalized Information Gain).

While the GS models obtained here are robust and perform well, they are only estimates of the true upper bounds of likelihood. The metric of NInGa is relative by design and depends on these estimates. Future better upper bound estimates will therefore be able to further increase the accuracy with which the performance of models is measured. Another limiting factor of this work is its treatment of neurons as independent, neglecting neural variability that is shared on a population level. Such correlated response patterns can be induced by different brain states and can contribute heavily to the overall variability [62]. However, the proposed framework in this project is general and flexible enough to be extended to full populations in future work.

5 Variably exciting stimuli

This chapter is based on the following project which currently has not been published yet:

- **Konstantin-Klemens Lurz** and Fabian Sinz. Variably exciting stimuli for mouse primary visual cortex. *Project is not published (yet)*, 2024.

5.1 Motivation

In recent years, ANN system identification models have become powerful enough to serve as digital twins of the brain, see Chapter 2.3. As a consequence, novel neuroscientific experiments could be conducted on these models which would not have been possible on their biological counterpart. One prominent example is the synthesis of artificial stimuli that elicit a certain response pattern in the neurons [26–30]. Importantly, the stimuli which were obtained by the digital twin in-silico can then be shown back to the animal and tested in-vivo. On the one hand, this gives additional credibility to the digital twin as an accurate model of the brain. On the other hand, the insights obtained from the experiments on the digital twin can help advance the understanding of the neural processing in the real brain.

Several studies have already been conducted using this method to gain such neuroscientific insights: Walker et al. [26] synthesized stimuli which maximally drive a target neuron’s activity in mouse V1, so called maximally exciting inputs (MEIs). They found various unknown complex receptive fields which differed from the traditional textbook Gabor-like filters. Burg et al. [116] synthesized maximally discriminative stimuli in order to identify functional cell types. Willeke et al. [117] found evidence of cortical columns in monkey V4 and Franke et al. [118] investigated state-dependent feature selectivity in the mouse cortex, both using the method of optimizing MEIs. Most relevant for this project is the work by Ding et al. [65] who synthesized so called diverse exciting inputs (DEIs): A set of stimuli which is optimized to elicit the same average firing rate in a neuron while each DEI is maximally different from every other DEI in the set. This is achieved by balancing a hinge loss with a diversity regularization during the optimization of the DEI. The so obtained stimuli are then used to study invariance manifolds in the primary visual cortex of mice.

This project uses a similar approach as Ding et al. [65] and aims at synthesizing stimuli which elicit the same mean activity. However, instead of enforcing the stimuli to be diverse in pixel space, they are optimized to evoke different variability around the mean, so called variably exciting inputs (VEIs). The project thus builds directly on Chapter 4 [24] by using full likelihood encoding models which can predict the conditional variance. The question attempted to be answered here is therefore: Is

*Equal contribution

information about the stimulus encoded in the variability of neural responses and, if yes, what kind of information?

5.2 Methods

The general framework of synthesizing stimuli is adapted from Walker et al. [26] and described in Chapter 2.3. Note, however, that the second part of the inception loop, the testing of the obtained stimuli back in the animal, could not be done here. This is due to a lack of feasibility given current experimental limitations as is explained in more detail in Chapter 5.3.

Encoding model The digital twins for synthesizing images were ensemble models. Each ensemble consisted of the 5 best performing out of 20 Zero-Inflated-Gamma (ZIG, see Chapter 4) encoding models which had been initialized with different seeds before training. The general architecture of the encoding models was as described in Chapter 2.2, meaning that they consisted of a core, readout, modulator and shifter network. As a readout, however, an adapted version of the Gaussian readout (Chapter 3 [23]) was chosen which learned three feature vectors per neuron to predict the three parameters of the ZIG distribution (θ, k, q) . The separation between zero-inflation and positive distribution was set to $\tau = e^{-10}$. The core was first initialized with the generalizing representation from Chapter 3 and then fine-tuned. All other parts of the network were trained starting from their random seed initialization.

Synthesis of MEIs In order to optimize VEIs, the firing rate of a neuron had to be kept constant. Since different neurons can exhibit very different firing rates, this constant mean firing rate was not defined as an absolute value but relative to the respective neuron’s maximum firing rate: its response to its MEI. Therefore, MEIs served as a reference level of firing rate and were optimized the following way, similar to [26]: The image was initialized with a constant value and then optimized using SGD and a learning rate of $lr = 0.2$. The total norm of the image was restricted to have a maximum value of 10 and a Gaussian blur ($\sigma = 1$) was applied to ensure smoothness in the result. The optimization was then stopped via early stopping when the mean firing rate had not increased for 20 iterations (minimum 200, maximum 2000 iterations).

Synthesis of standard VEIs The process of generating VEIs started by optimizing an image which activated the target neuron’s mean activity to a certain reference level c of its MEI activation. In practice, either $c = 0.8$ or $c = 1$. was used. With μ being the mean firing rate of the neuron, the reference loss L_{ref} was then defined as

$$L_{total} = L_{ref} = - \left(c - \frac{\mu}{\mu_{MEI}} \right)^2. \quad (5.1)$$

Differently from the MEI generation, no norm budget was applied because the loss avoids the optimization of extreme pixel values by design. Once the specified activation level was reached, optimization was switched to VEI synthesis, consisting of a loss term L_{var} for increasing/decreasing the target neuron’s variance and a more strict L_{ref}

term for keeping the mean strictly at the reference level. The design of L_{ref} proved to be important and delicate because the restriction on the mean activation needed to be enforced strictly while at the same time not drowning the gradient signal of L_{var} . Since the method of synthesis of VEIs described in this section was rather robust, a very steep exponential potential well function could be used, see also Figure 5.1:

$$L_{total} = L_{var} + L_{ref} \quad (5.2)$$

$$L_{var} = \pm \frac{\sigma^2}{\sigma_{MEI}^2} \quad (5.3)$$

$$L_{ref} = \exp\left(-\alpha\left(\frac{\mu}{\mu_{MEI}} + dx - c\right)\right) + \exp\left(\alpha\left(\frac{\mu}{\mu_{MEI}} - dx - c\right)\right) \quad (5.4)$$

with α (set to 100) being the scale of steepness of the potential well's borders and dx (set to 0.02) setting the size of the plateau inside the well.

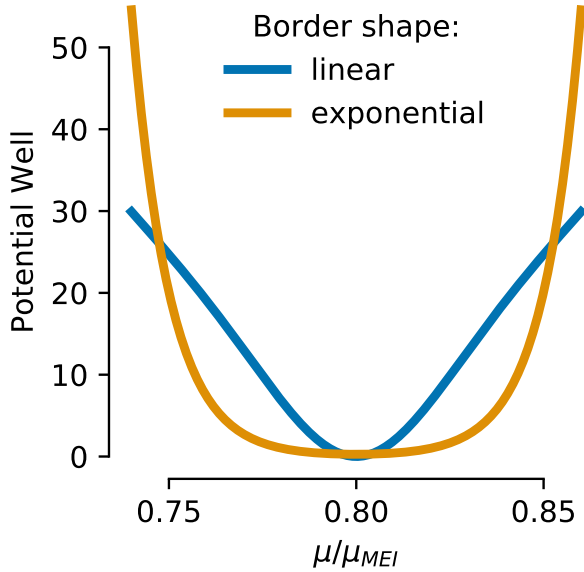


Figure 5.1: **Potential well functions.** The exponential potential well (Equation 5.4) allows to define the plateau of acceptable values of neural activity very cleanly. Small deviations from it result in an explosion of loss and, consequently, gradients. A more robust yet less well confining potential well has linear borders far from the plateau (Equation 5.9) making it easier for the algorithm to re-converge to the minimum. Parameters: $c = 0.8$, $dx = 0.02$, $\alpha = 100$ (exponential), $\alpha = 1000$ (linear).

Synthesis of orthogonal VEIs In this method, VEIs are generated such that they consist of the reference stimulus (outcome of training with Equation 5.1) plus another stimulus which is orthogonal to the reference stimulus. The sum of the two stimuli results in the VEI which by itself must fulfill the requirements of matching the mean activation of the reference stimulus and increasing/decreasing the variance around the mean. Using this method it is possible to determine more clearly which image features are responsible for the change in variance. To this end, the reference image resulting from optimizing Equation 5.1 is flattened to a vector v_{ref} and a second flattened orthogonal image Δv is added. Orthogonality is enforced by expressing Δv with a third flattened image vector $\tilde{v}$:

$$v_{VEI} = v_{ref} + \Delta v, \quad v_{ref} \perp \Delta v \quad (5.5)$$

$$\Delta v = \left(\mathbb{1} - \frac{1}{\|v_{ref}\|^2} v_{ref} v_{ref}^T \right) \cdot \tilde{v} \quad (5.6)$$

Here, only $\tilde{v}$ has learnable parameters, the reference image vector v_{ref} is kept fixed. Finally, the VEI vector v_{VEI} is reshaped to an image again. This method of synthesizing VEIs is more unstable because the orthogonality constraint can cause the VEI to change its mean activation in the beginning of optimization. Therefore, a more forgiving reference loss is needed which does not explode exponentially as in Equation 5.4. An alternative potential well is hence used which provides linearly increasing borders further outside the plateau, see also Figure 5.1:

$$L_{total} = L_{var} + L_{ref} \quad (5.7)$$

$$L_{var} = \pm \frac{\sigma^2}{\sigma_{MEI}^2} \quad (5.8)$$

$$L_{ref} = \frac{1}{2} \alpha \left(\frac{\mu}{\mu_{MEI}} - c \right) \operatorname{erf} \left(\frac{\frac{\mu}{\mu_{MEI}} - c}{dx \sqrt{2}} \right) \quad (5.9)$$

with $\operatorname{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt$ being the Error function. Here, the values $\alpha = 1000$ and $dx = 0.02$ were chosen.

5.3 Results

In this section, the information in the stimulus that potentially gives rise to variability in neural activity conditioned on a stimulus was analyzed. To this end, it was first tested, whether the models predicted a trivial relationship between mean and variance which would rule out the encoding of parts of the information in the stimulus via the variance alone. Since this was not the case the methods described above were then used to synthesize VEIs of mouse visual cortex for a deeper analysis of neural conditional variability.

To this end, first standard VEIs without constraint were generated and found to be partially linked to the surround of the stimulus. Then, an orthogonality constraint was applied on the VEIs to highlight features of increasing/decreasing conditional neural variability. Finally, a proxy experiment to inception loops was conducted in-silico where a new ensemble of encoding models instead of the original animal served as a test for the synthesized VEIs. It showed that while the results of VEIs are consistent over testing models, a real inception loop experiment with an animal is currently not feasible due to experimental limitations with respect to the dataset size.

No trivial relationship between mean and variance. If it is the case that neural variance around the conditional mean encodes information about the stimulus independently of the mean, it must not be possible to trivially predict the variance from the mean alone. To test this, full ZIG encoding models which learned the mean and variance independently were compared to ZIG models which learned the mean

from the stimulus x and parameterized the variance as a polynomial of the mean: $\sigma^2(\mu) = \beta_0 + \beta_1\mu(x) + \beta_2\mu^2(x)$. Indeed, the full ZIG models significantly outperformed the polynomial models, see Figure 5.2. Additionally, the fitted coefficients of the polynomials further emphasized the mismatch between the Poisson distribution ($\beta_0 = 0, \beta_1 = 1, \beta_2 = 0$) and the best fitting polynomial model ($\beta_0 \approx 0, \beta_1 \approx 1.5, \beta_2 \approx 0.25$). The Poisson distribution thus again proved to be a suboptimal choice for modeling neural responses, compare Chapter 4.

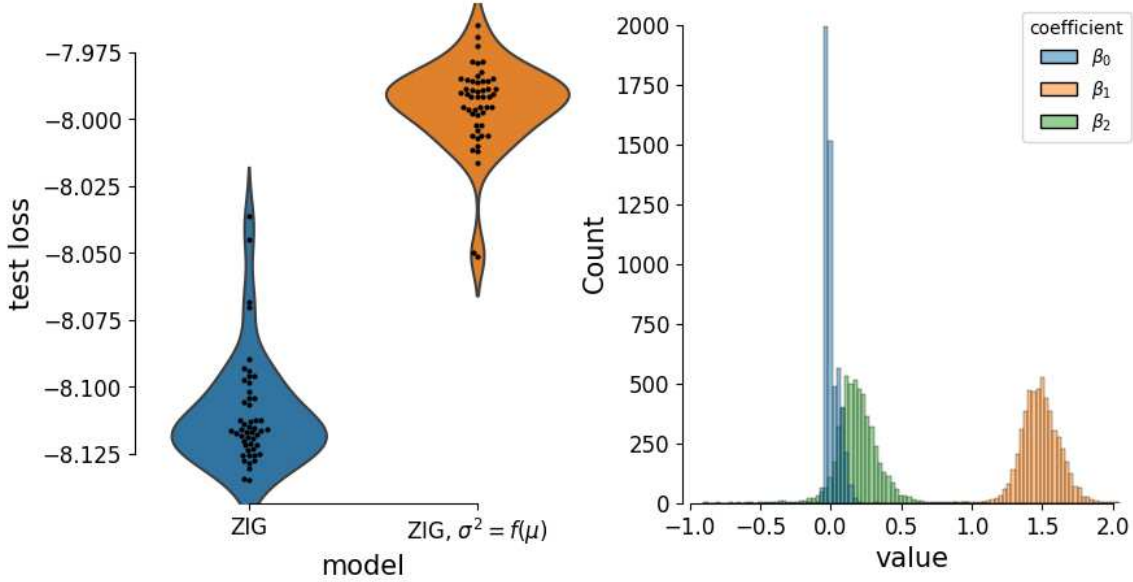


Figure 5.2: **Variance can not be trivially predicted by the mean.** **Left panel:** The full ZIG encoding models outperform the ZIG models where the variance is computed as a function (polynomial) of the mean, indicating that there is no trivial relationship between mean and variance. **Right panel:** The fitted coefficients for all neurons of one exemplary polynomial ZIG model.

Synthesis of standard VEIs Since the models predicted no trivial relationship between mean and variance, the information content of the variance about the input stimulus was then investigated using VEIs. First, vanilla VEIs without constraint were optimized as described in Equation 5.4 and examples are displayed in Figure 5.3, together with the respective neuron’s MEI. In accordance with [26], the obtained MEIs of mouse visual cortex often differed significantly from textbook Gabor-like shapes. While the encoding models predicted a constant mean and a clear increase/decrease in variance, the perceptual difference between MEIs and VEIs was only modest and hard to interpret. One visual feature that was standing out was the different emphasis that VEIs with high (VEI+) and low variance (VEI-) gave to the surround of the receptive field. High variance VEIs seemed to result in a larger stimulus with more contrast allocated in the surround of the neuron’s receptive field.

Variance is linked to stimulus size The observation that VEIs that were optimized to elicit higher variance resulted in a larger stimulus was so far only observed qualitatively. In order to test the effect quantitatively, a reverse experiment was conducted: No VEI was explicitly synthesized but instead the reference stimulus from Equation 5.1 was further optimized with an additional L1 regularization term. The

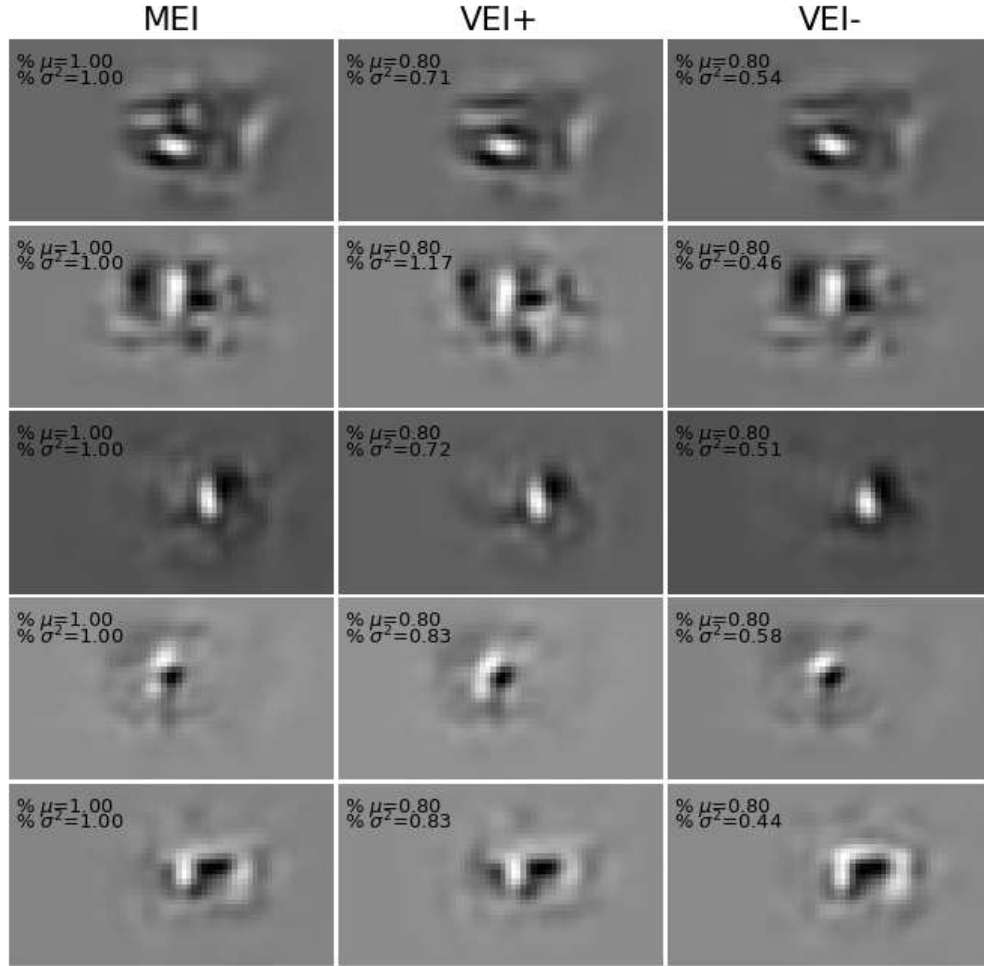


Figure 5.3: **Vanilla VEIs of mouse visual cortex.** The VEIs of five example neurons (rows) were compared to their neuron’s respective MEI. The relative mean firing rate of the VEIs was successfully kept at the reference level ($\% \mu = \mu / \mu_{MEI} = c = 0.8$). The variance relative to the MEI variance ($\% \sigma^2 = \sigma^2 / \sigma_{MEI}^2$) differed substantially between the VEIs. Note that due to the positive correlation between mean and variance, it is impossible for the VEIs to achieve the variance of the MEI while $c = 0.8$.

regularization restricts the norm budget that the algorithm can allocate to pixel values of the stimulus. Since contrast in the center of the receptive field drives neurons the most, a strong restriction on the norm budget meant that most contrast had to be spent in the center in order to match the reference firing rate. The weight of L1 regularization thus acted as a proxy for stimulus size and effectively resulted in smaller stimuli with less contrast in the surround. The variance in the firing rate predicted by the model as a response to these stimuli could then be compared to the size of the stimulus or, as a proxy, to its L1 weight, see the left panel of Figure 5.4. Many neurons (colored lines) indeed showed a tendency to have lower variance when shown a stimulus with smaller size. Note again that the mean firing rate was kept constant during the experiment. In order to measure the effect to a high precision, the experiment was then run for a large population of 250 neurons and the slope between L1 weight and variance was fitted with a simple Linear Regression model for each neuron. A significant portion of the population (74%) had a negative slope, meaning that a smaller stimulus elicited a smaller variance while the mean activity was constant, see the right panel of Figure 5.4.

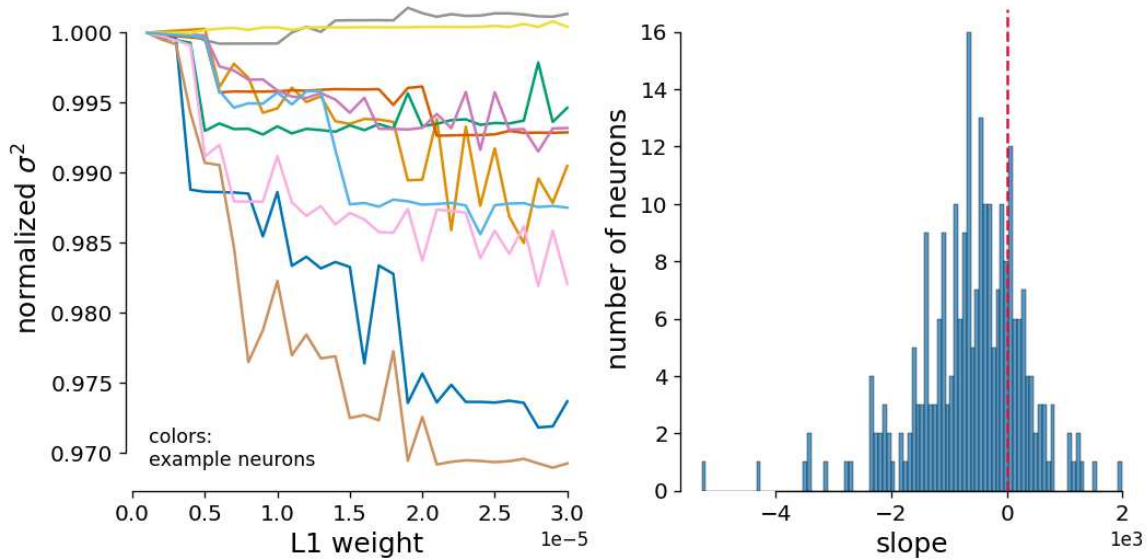


Figure 5.4: **Variance is linked to stimulus size.** Stimuli were optimized only to match the reference mean firing rate and there was no explicit optimization towards the variance. **Left panel:** Applying L1 regularization during the optimization acts as a proxy for restricting stimulus size. Ten random example neurons (colors) show decreasing variance for decreasing stimulus size while the mean was kept constant. Variance was normalized w.r.t. the value corresponding to no L1 regularization. **Right panel:** The slopes of the lines in the left panel were fitted with Linear Regression models for a population of 250 randomly selected neurons. A significant portion of 74% of the neurons show decreasing variance with decreasing stimulus size.

Synthesis of orthogonal VEIs In order to gain further insights into the image features that drive or suppress conditional variance in neural activity, orthogonal VEIs were synthesized. The VEIs are the sum of a reference stimulus v_{ref} and an orthogonal stimulus Δv and were obtained as described in Equation 5.6. Note that here, as opposed to Figure 5.3, the stimuli were tested on different model ensembles than they were trained or validated (early stopping) on. For more details about train, validation and test ensembles, refer to the next paragraph on in-silico inception loops.

The resulting orthogonal stimuli can be seen in Figure 5.5. As a first observation, they do clearly not correspond to noise that the model might have picked up, since they show distinct and complex patterns. Also, while producing similar types of patterns, the features can be quite different from those of their respective reference stimulus. The intensity of the features is in correspondence with the amount of change in variance that the respective $\Delta v_{\pm}$ can achieve. However, note that due to the use of a different model ensemble for testing than for optimizing, the results are much weaker than in Figure 5.3 where only one ensemble was used: While the mean was successfully kept constant in the test model, the increase/decrease in variance was only by a very small margin. This will be investigated further in the next paragraph.

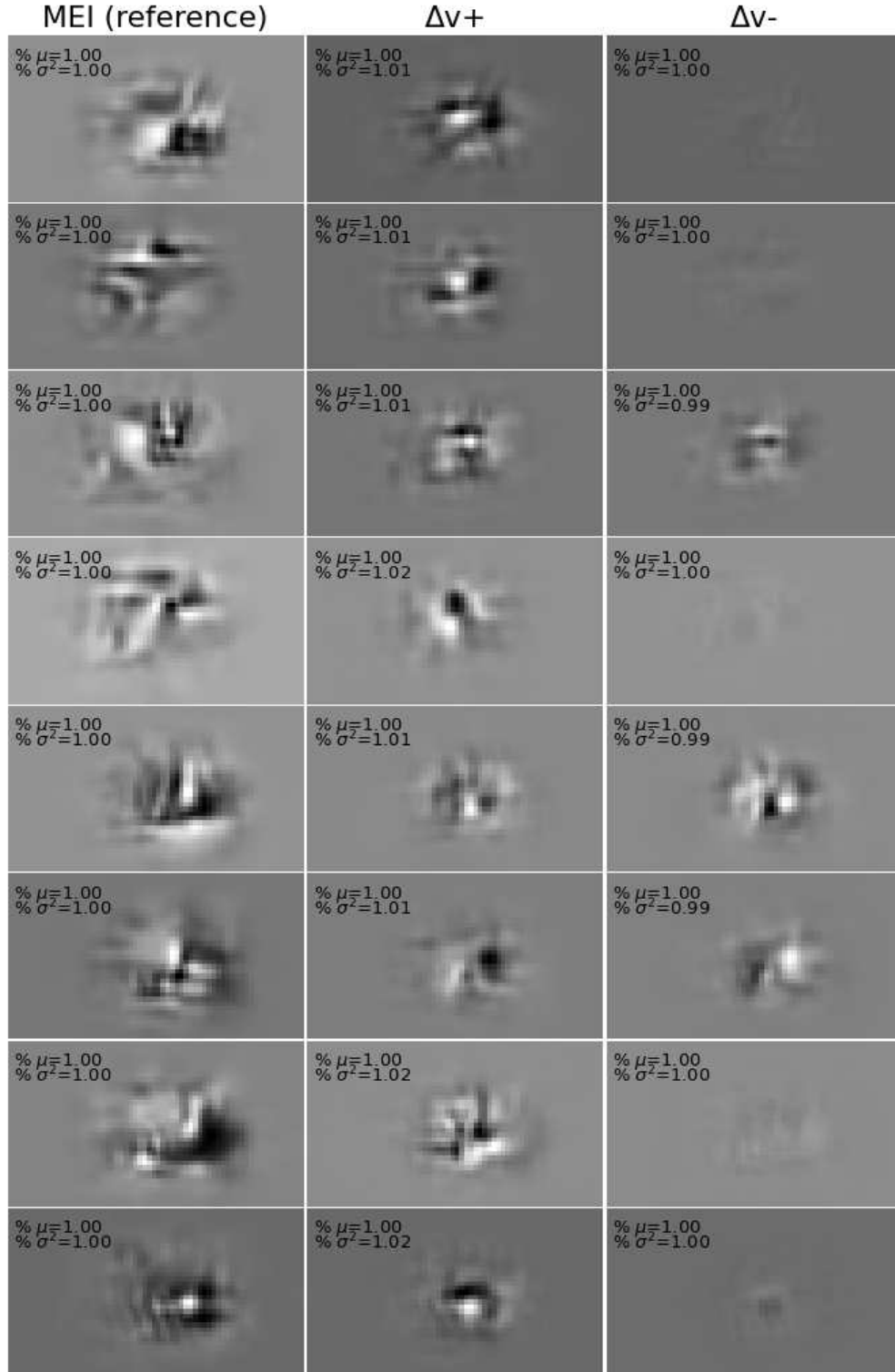


Figure 5.5: **Orthogonal stimuli for VEIs.** For eight example neurons, the MEIs and corresponding orthogonal images $\Delta v_{\pm}$ are plotted. The MEI served as a reference stimulus v_{ref} meaning that $c = 1$. Note that the $\Delta v_{\pm}$ have much smaller pixel values than the MEIs and are normalized only w.r.t. each other for better visibility. The resulting VEIs are the sum of the reference stimulus v_{ref} (here the MEI) and $\Delta v_{\pm}$ and are not displayed due to small perceptual difference to the MEIs. The mean and variance were predicted by a *test ensemble* of encoding models as a response to the *full* stimulus (not only $\Delta v_{\pm}$).

In-silico inception loop An important part of using digital twins in Neuroscience is to test the results that were obtained in-silico back in the animal (in-vivo). Walker et al. [26], for instance, tested their MEIs in the same mouse that they had obtained the training data from the previous day, in a so called inception loop. Before conducting such an intricate experiment on alive animals, a feasibility study is usually performed. Here, the VEIs obtained from one ensemble of encoding models were tested on a different ensemble of models which took the role of the animal: an in-silico inception loop. Additionally, the early stopping for synthesizing the VEIs on the train ensemble was done on a third ensemble of models, the validation ensemble. Figure 5.5 shows stimuli obtained in this manner. The small change in variance depicted on these example stimuli already indicates a challenge which is investigated in more detail in Figure 5.6: While the effect of the synthesized VEI stimuli is still present and significant in the test ensemble, it is much less prevalent than in the train ensemble itself (compare Figure 5.3). The mean is successfully kept at the reference level (top left panel) but the effect in variance is only in the order of 10^{-2} . For some of the neurons with small difference in variance, the predicted sign of variance shift is even switched (bottom left panel). In order to estimate how many repeated presentations of a VEI would be necessary in a real experiment in order to confidently tell the neural responses apart from the opposite VEI, a simulation was run: For the neuron with highest difference in variance between VEI+ and VEI-, different amounts of samples were drawn from the predicted distribution of VEI+. Then, the conditional probability $p(\text{VEI}_+ \mid \text{samples})$ of the stimulus being VEI+ given the samples was computed. The resulting plot (right panel) shows that about 10000 samples would be needed to identify the respective VEI with a confidence of 80%. Since the size of full datasets is currently about 6000 datapoints (refer to Chapter 2.4), this is out of the scope of today’s experimental possibilities. The results in this project can therefore currently not be validated back in the animal in an inception loop.

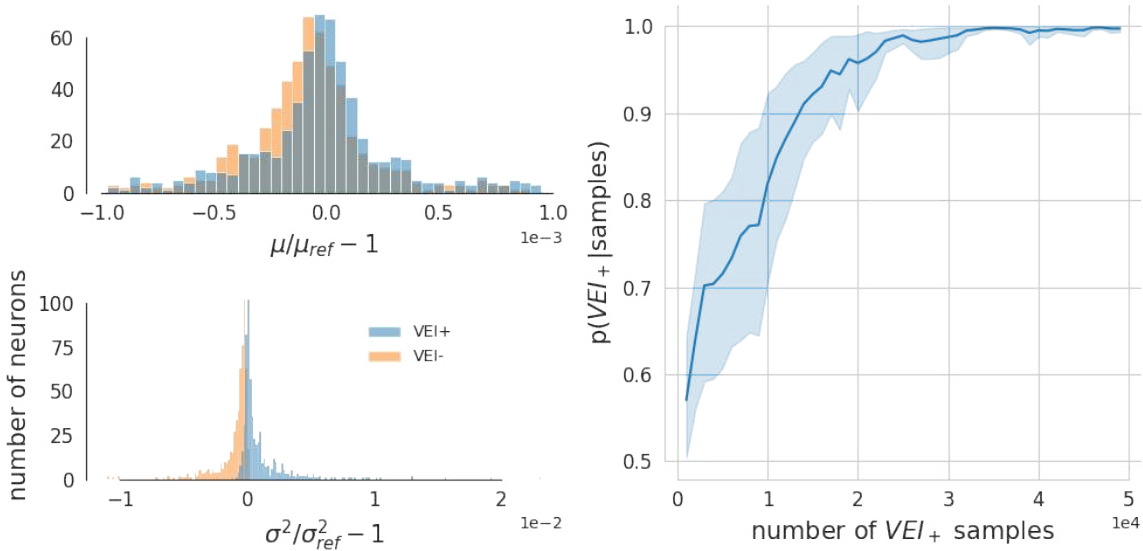


Figure 5.6: **In-silico inception loop.** The responses to VEIs were predicted by a separate test ensemble of encoding models. **Left panel:** While the mean is successfully kept at the reference level, the change in variance is only at the order of 10^{-2} . **Right panel:** Simulation of how many repeated presentations of the same VEI would be needed to identify with a certain confidence that the neural responses came from the respective VEI.

5.4 Discussion

New highly performing state-of-the-art models of neural representations [23] combined with novel network architectures and frameworks to model full response distributions [24, 62, 114] have made it possible to investigate neural response patterns with ANNs on a novel level: variability in neural firing conditioned on a stimulus. Insights into these patterns are important since they allow to test several normative theories which rely on or make predictions about the conditional variance of neural firing [111–113]. While work such as the mixture modeling approach by [119] and Wei et al. [114] investigates higher moments of neural response distributions, they lack the usage of stimuli that are especially suited to analyze these higher moments. A successful method to inspect neural response properties is to use ANN models to synthesize stimuli which drive a neuron in a certain desired way [26–30]. These stimuli can then be shown back to the animal in inception loops [26] in order to test the insights gained from the model, or "digital twin". However, such image synthesis has, to the best of our knowledge, only been done to investigate the mean firing rate as in [26, 65].

Here, the state-of-the-art ANN models were used to synthesize variably exciting stimuli (VEI) with the goal to investigate whether the variability of neural responses around the mean firing rate encodes information about the stimulus and what this information might contain. First, it was shown that the digital twin models predict a non-trivial relationship between the mean and variance, indicating that the variance does indeed encode information which is not already present in the mean. Then, using these models, VEIs were generated and found to be very similar perceptually to each other and to the most exciting stimuli (MEIs). However, it could be shown that there is a significant correlation between the size of the VEI and the variance that it elicits. This means that center-surround effects might play a role not only in the rate of neural firing but also in its variability for a fixed rate. Using a different method of generating VEIs, it was then possible to obtain the *orthogonal* changes in the stimulus that were responsible for increasing/decreasing variance. The extracted features were different from the respective neuron's MEI but had similar general patterns as receptive fields in mouse V1 [26]. In order to test these findings, however, an experiment in-vivo in the style of the inception loops was needed. As a feasibility study, such an experiment was conducted in-silico first, using a different test ensemble instead of the alive animal. The simulations showed that the investigated effects are consistent across models, adding to the credibility of the results. However, the size of the effect of VEIs was shown to be too small to be measured in an animal with the currently available technology. It therefore remains to be seen if the results can be tested in an experiment in the future. To this end, better models which can potentially synthesize stronger VEIs or better experimental tools for recording larger datasets will be needed.

6 Discussion and conclusion

In recent years, Vision Neuroscience has seen great advancements due to the mutually beneficial symbiosis with the field of Machine Learning. Especially the adaptation of Artificial Neural Networks as system identification models has proven to be of great value for modeling the complex computations that are performed by cortical neurons for visual processing. Such system identification models using ANNs are now the state-of-the-art for predicting neural activity as a response to natural stimuli. Additionally, these models have proven very useful for traditional biological Neuroscience as well. Experiments whose search space would be unfeasible to investigate in-vivo can in a first step be run on a system identification model instead. These digital twins can then help to shrink the search space in-silico to a size which is more manageable for an experiment in-vivo. This thesis built on these latest advancements of ANN system identification models of mouse visual cortex for neural prediction and in-silico experiments and expanded the knowledge of the field in the following way:

First, the problem of generalization was investigated: The question of how to obtain representations which generalize well between individuals and are truly characteristic for mouse visual cortex. To this end, a data-driven approach was chosen and shown to successfully yield generalizing representations by adapting two novel contributions: A data-efficient architecture for the model readout which can make use of anatomical data to learn the receptive field location. And a method to enlarge the size of the datasets by matching neurons from different sessions based on their cortical positions and functional data.

The second project of this thesis laid the mathematical and theoretical foundations for the successful adaptation of a promising new type of system identification model, the full likelihood model. Since such models predict all moments of the neural response distribution, they can be used to test several normative theories that make predictions about, for instance, neural variability around the mean. The right way to estimate the predictive performance of such models was, however, not clear. In this thesis, the mathematical framework for estimating the lower and upper bound of the Normalized Information Gain measure was derived for a variety of relevant distributions. The obtained estimators were then tested and found to be robust against many complexities often encountered in neural data.

Having paved the ground for full likelihood models which can predict neural variability and having obtained characteristic and generalizing representations for these models, the third project aimed to use these models for neuroscientific insights. To this end, the models were used as digital twins of the brain in order to investigate the role of neural variability for encoding information about the stimulus. So called variably exciting inputs (VEIs) were synthesized using the models in order to detect features in the input image that would increase or decrease neural variance. It was found that the surround of the stimulus has a significant effect on the variance, whereby more

surround tended to increase the variance in neural firing. Additionally, the features in the stimulus which are relevant for driving or suppressing variance could be extracted orthogonal to the main stimulus, giving insights into the type of patterns giving rise to the observed variability. Unfortunately, an in-silico proxy experiment for the in-vivo testing in inception loops showed that while the obtained results were consistent, they would not be detectable and testable with current experimental setups in-vivo.

The results in this thesis can hopefully prove as a fruitful basis for future research to build on. There is much potential in the pursuit of better system identification models that make use of potential advancements in the field of Machine Learning. Also, having prepared the foundational framework for full likelihood models, this thesis could only start to scratch on the possible insightful research that can be done by investigating full neural response distributions. Being able to test and therefore verify or reject competing theories of neural computation will advance the field as a whole and tailor future research efforts towards the most promising proposals of explaining the science behind neural visual processing. One start in this direction was attempted in this thesis by analyzing the information encoded by neural variance. While the results look promising, it remains to be seen whether future improvements in the field such as better system identification models or more sophisticated recording tools for biological data can make it possible to also put these results to the test in-vivo.

References

- [1] Stanley Finger. *Origins of neuroscience: a history of explorations into brain function*. Oxford University Press, 2001.
- [2] David B Baker. *The Oxford handbook of the history of psychology: Global perspectives*. Oxford University Press, 2012.
- [3] Kenneth David Keele. Leonardo da vinci on vision, 1955.
- [4] Isaac Newton. *Opticks: Or, A Treatise of the Relexions, Refractions, Inflexions and Colours of Light: Also Two Treatises of the Species and Magnitude of Curvilinear Figures*. Culture et civilisation, 1966 (originally published in 1704).
- [5] Johann Wolfgang Goethe. *Zur Farbenlehre*. BoD–Books on Demand, 2016 (originally published in 1810).
- [6] Neil Ribe and Friedrich Steinle. Exploratory experimentation: Goethe, land, and color theory. *Physics today*, 55(7):43–49, 2002.
- [7] Ragnar Granit and Gunnar Svaetichin. *Principles and Technique of the Electrophysiological Analysis of Colour Reception with the Aid of Microelectrodes*. Almqvist & Wiksells boktryckeri A.-B., 1939.
- [8] Haldan Keffer Hartline. The response of single optic nerve fibers of the vertebrate eye to illumination of the retina. *American Journal of Physiology-Legacy Content*, 121(2):400–415, 1938.
- [9] Haldan K Hartline. The receptive fields of optic nerve fibers. *American Journal of Physiology-Legacy Content*, 130(4):690–699, 1940.
- [10] Edward H Adelson, James R Bergen, et al. The plenoptic function and the elements of early vision. *Computational models of visual processing*, 1(2):3–20, 1991.
- [11] Horace B Barlow et al. Possible principles underlying the transformation of sensory messages. *Sensory communication*, 1(01):217–233, 1961.
- [12] 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.
- [13] 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.

- [14] 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.
- [15] Christina Enroth-Cugell and John G Robson. The contrast sensitivity of retinal ganglion cells of the cat. *The Journal of physiology*, 187(3):517–552, 1966.
- [16] Robert W Rodieck. Quantitative analysis of cat retinal ganglion cell response to visual stimuli. *Vision research*, 5(12):583–601, 1965.
- [17] Christina Enroth-Cugell and John G Robson. Functional characteristics and diversity of cat retinal ganglion cells. basic characteristics and quantitative description. *Investigative ophthalmology & visual science*, 25(3):250–267, 1984.
- [18] Christina Enroth-Cugell and Lawrence Pinto. Algebraic summation of centre and surround inputs to retinal ganglion cells of the cat. *Nature*, 226(5244):458–459, 1970.
- [19] Edward H Adelson and James R Bergen. Spatiotemporal energy models for the perception of motion. *Josa a*, 2(2):284–299, 1985.
- [20] Russell L De Valois, Duane G Albrecht, and Lisa G Thorell. Spatial frequency selectivity of cells in macaque visual cortex. *Vision research*, 22(5):545–559, 1982.
- [21] HB Barlow, Raghavan Narasimhan, and A Rosenfeld. Visual pattern analysis in machines and animals: The same principles may underlie the operation of sensory neurons, computer programs, and perception. *Science*, 177(4049):567–575, 1972.
- [22] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.
- [23] **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.
- [24] **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.
- [25] T. Söderström and P. Stoica. *System Identification*. Prentice-Hall Software Series. Prentice Hall, 1989. ISBN 9780138812362. URL https://books.google.de/books?id=X_xQAAAAMAAJ.
- [26] 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.
- [27] Pouya Bashivan, Kohitij Kar, and James J DiCarlo. Neural population control via deep image synthesis. *Science*, 364(6439), 2019.

- [28] 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.
- [29] William F Kindel, Elijah D Christensen, and Joel Zylberberg. Using deep learning to probe the neural code for images in primary visual cortex. *Journal of vision*, 19(4):29–29, 2019.
- [30] Jumpei Ukita, Takashi Yoshida, and Kenichi Ohki. Characterisation of nonlinear receptive fields of visual neurons by convolutional neural network. *Scientific reports*, 9(1):3791, 2019.
- [31] Robert C Emerson, MARK C Citron, WILLIAM J Vaughn, and STANLEY A Klein. Nonlinear directionally selective subunits in complex cells of cat striate cortex. *Journal of Neurophysiology*, 58(1):33–65, 1987.
- [32] Judson P Jones and Larry A Palmer. The two-dimensional spatial structure of simple receptive fields in cat striate cortex. *Journal of neurophysiology*, 58(6):1187–1211, 1987.
- [33] Panos Z Marmarelis and Ken-Ichi Naka. White-noise analysis of a neuron chain: an application of the wiener theory. *Science*, 175(4027):1276–1278, 1972.
- [34] Hiroko M Sakai, Naka Ken-Ichi, and Michael J Korenberg. White-noise analysis in visual neuroscience. *Visual neuroscience*, 1(3):287–296, 1988.
- [35] 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.
- [36] John Ashworth Nelder and Robert WM Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, 135(3):370–384, 1972.
- [37] 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.
- [38] 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.
- [39] Alexander Heitman, Nora Brackbill, Martin Greschner, Alexander Sher, Alan M Litke, and EJ Chichilnisky. Testing pseudo-linear models of responses to natural scenes in primate retina. *BioRxiv*, page 045336, 2016.
- [40] Urs Köster and Bruno Olshausen. Testing our conceptual understanding of v1 function. *arXiv preprint arXiv:1311.0778*, 2013.

- [41] Valerio Mante, Vincent Bonin, and Matteo Carandini. Functional mechanisms shaping lateral geniculate responses to artificial and natural stimuli. *Neuron*, 58(4):625–638, 2008.
- [42] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.
- [43] 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.
- [44] 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*, 46:33–54, 2019.
- [45] Lane McIntosh, Niru Maheswaranathan, Aran Nayebi, Surya Ganguli, and Stephen Baccus. Deep learning models of the retinal response to natural scenes. *Advances in neural information processing systems*, 29, 2016.
- [46] Charles F Cadieu, Ha Hong, Daniel LK Yamins, Nicolas Pinto, Diego Ardila, Ethan A Solomon, Najib J Majaj, and James J DiCarlo. Deep neural networks rival the representation of primate it cortex for core visual object recognition. *PLoS computational biology*, 10(12):e1003963, 2014.
- [47] William F Kindel, Elijah D Christensen, and Joel Zylberberg. Using deep learning to reveal the neural code for images in primary visual cortex. *arXiv preprint arXiv:1706.06208*, 2017.
- [48] 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.
- [49] Ryan Prenger, Michael C-K Wu, Stephen V David, and Jack L Gallant. Nonlinear v1 responses to natural scenes revealed by neural network analysis. *Neural Networks*, 17(5-6):663–679, 2004.
- [50] Brian Lau, Garrett B Stanley, and Yang Dan. Computational subunits of visual cortical neurons revealed by artificial neural networks. *Proceedings of the National Academy of Sciences*, 99(13):8974–8979, 2002.
- [51] Sidney R Leaky, Terrence J Sejnowski, and Robert Desimone. Predicting responses of nonlinear neurons in monkey striate cortex to complex patterns. *Journal of Neuroscience*, 12(9):3568–3581, 1992.
- [52] Umut Güçlü and Marcel AJ Van Gerven. Modeling the dynamics of human brain activity with recurrent neural networks. *Frontiers in computational neuroscience*, 11:7, 2017.
- [53] Pulkit Agrawal, Dustin Stansbury, Jitendra Malik, and Jack L Gallant. Pixels to voxels: modeling visual representation in the human brain. *arXiv preprint arXiv:1407.5104*, 2014.

- [54] Michael Eickenberg, Alexandre Gramfort, Gaël Varoquaux, and Bertrand Thirion. Seeing it all: Convolutional network layers map the function of the human visual system. *NeuroImage*, 152:184–194, 2017.
- [55] Umut Güçlü and Marcel AJ van Gerven. Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *Journal of Neuroscience*, 35(27):10005–10014, 2015.
- [56] Jonas Kubilius, Stefania Bracci, and Hans P Op de Beeck. Deep neural networks as a computational model for human shape sensitivity. *PLoS computational biology*, 12(4):e1004896, 2016.
- [57] 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.
- [58] 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.
- [59] Brett Vintch, J Anthony Movshon, and Eero P Simoncelli. A convolutional subunit model for neuronal responses in macaque v1. *Journal of Neuroscience*, 35(44):14829–14841, 2015.
- [60] 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.
- [61] 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.
- [62] 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.
- [63] 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.
- [64] 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>.
- [65] 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.

- [66] 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.
- [67] Daniel LK Yamins and James J DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nature neuroscience*, 19(3):356–365, 2016.
- [68] Seyed-Mahdi Khaligh-Razavi and Nikolaus Kriegeskorte. Deep supervised, but not unsupervised, models may explain it cortical representation. *PLoS computational biology*, 10(11):e1003915, 2014.
- [69] 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.
- [70] Jonas Kubilius, Martin Schrimpf, Aran Nayebi, Daniel Bear, Daniel LK Yamins, and James J DiCarlo. Cornet: Modeling the neural mechanisms of core object recognition. *BioRxiv*, page 408385, 2018.
- [71] Alexander JE Kell, Daniel LK Yamins, Erica N Shook, Sam V Norman-Haignere, and Josh H McDermott. A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy. *Neuron*, 98(3):630–644, 2018.
- [72] Aran Nayebi, Daniel Bear, Jonas Kubilius, Kohitij Kar, Surya Ganguli, David Sussillo, James J DiCarlo, and Daniel L Yamins. Task-driven convolutional recurrent models of the visual system. *Advances in neural information processing systems*, 31, 2018.
- [73] Umut Güçlü and Marcel AJ van Gerven. Unsupervised feature learning improves prediction of human brain activity in response to natural images. *PLoS computational biology*, 10(8):e1003724, 2014.
- [74] Martin Schrimpf, Jonas Kubilius, Ha Hong, Najib J Majaj, Rishi Rajalingham, Elias B Issa, Kohitij Kar, Pouya Bashivan, Jonathan Prescott-Roy, Franziska Geiger, et al. Brain-score: Which artificial neural network for object recognition is most brain-like? *BioRxiv*, page 407007, 2018.
- [75] Santiago A Cadena, Fabian H Sinz, Taliah Muhammad, Emmanouil Froudarakis, Erick Cobos, Edgar Y Walker, Jake Reimer, Matthias Bethge, Andreas Tolias, and Alexander S Ecker. How well do deep neural networks trained on object recognition characterize the mouse visual system? In *Real Neurons {→ Hidden Units: Future directions at the intersection of neuroscience and artificial intelligence@ NeurIPS 2019*, 2019.
- [76] Nikolaus Kriegeskorte. Deep neural networks: a new framework for modeling biological vision and brain information processing. *Annual review of vision science*, 1:417–446, 2015.
- [77] Tim C Kietzmann, Patrick McClure, and Nikolaus Kriegeskorte. Deep neural networks in computational neuroscience. *BioRxiv*, page 133504, 2017.

- [78] Fabian H Sinz, Xaq Pitkow, Jacob Reimer, Matthias Bethge, and Andreas S Tolias. Engineering a less artificial intelligence. *Neuron*, 103(6):967–979, 2019.
- [79] Kendrick N Kay. Principles for models of neural information processing. *NeuroImage*, 180:101–109, 2018.
- [80] Liz Allen, Jo Scott, Amy Brand, Marjorie Hlava, and Micah Altman. Publishing: Credit where credit is due. *Nature*, 508(7496):312–313, 2014.
- [81] 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.
- [82] Alex O Holcombe. Contributorship, not authorship: Use credit to indicate who did what. *Publications*, 7(3):48, 2019.
- [83] **Konstantin-Klemens Lurz** and Fabian Sinz. Variably exciting stimuli for mouse primary visual cortex. *Project is not published (yet)*, 2024.
- [84] Suhas Shrinivasan, **Konstantin-Klemens Lurz**, Kelli Restivo, George Denfield, Andreas S Tolias, Edgar Y Walker, and Fabian H Sinz. Taking the neural sampling code very seriously: A data-driven approach for evaluating generative models of the visual system. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- [85] **Konstantin-Klemens Lurz***, Mohammad Bashiri*, and Fabian H Sinz. Bayesian oracle for bounding information gain in neural encoding models. In *NeurIPS Workshop on Information-Theoretic Principles in Cognitive Systems*, 2022.
- [86] 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.
- [87] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5:115–133, 1943.
- [88] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.
- [89] 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.
- [90] 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.
- [91] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

- [92] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [93] Facundo Bre, Juan M Gimenez, and Víctor D Fachinotti. Prediction of wind pressure coefficients on building surfaces using artificial neural networks. *Energy and Buildings*, 158:1429–1441, 2018.
- [94] 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.
- [95] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- [96] M Rama Nopan Ariyadi, Muhammad Rizky Pribadi, and Eka Puji Widiyanto. Unmanned aerial vehicle for remote sensing detection of oil palm trees using you only look once and convolutional neural network. In *2023 10th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, pages 226–230. IEEE, 2023.
- [97] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.
- [98] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.
- [99] 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.
- [100] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.
- [101] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.
- [102] Carsen Stringer, Marius Pachitariu, Nicholas Steinmetz, Charu Reddy, Matteo Carandini, and Kenneth D Harris. Spontaneous behaviors drive multidimensional, brain-wide neural activity. *BioRxiv*, page 306019, 2018.
- [103] Cristopher M Niell and Michael P Stryker. Modulation of visual responses by behavioral state in mouse visual cortex. *Neuron*, 65(4):472–479, 2010.
- [104] Mikhail Ya Marov. *The Fundamentals of Modern Astrophysics: A Survey of the Cosmos from the Home Planet to Space Frontiers*. Springer, 2014.
- [105] 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.

- [106] 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.
- [107] Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 1995.
- [108] 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.
- [109] 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.
- [110] 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.
- [111] 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.
- [112] Ralf M Haefner, Pietro Berkes, and József Fiser. Perceptual decision-making as probabilistic inference by neural sampling. *Neuron*, 90(3):649–660, 2016.
- [113] 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.
- [114] 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.
- [115] 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.
- [116] Max F Burg, Thomas Zenkel, Michaela Vystrčilová, Jonathan Oesterle, Larissa Höfling, Konstantin Friedrich Willeke, Jan Lause, Sarah Müller, Paul G Fahey, Zhiwei Ding, et al. Maximally discriminative stimuli for functional cell type identification. In *The Twelfth International Conference on Learning Representations*, 2023.
- [117] Konstantin F Willeke, Kelli Restivo, Katrin Franke, Arne F Nix, Santiago A Cadena, Tori Shinn, Cate Nealley, Gabby Rodriguez, Saumil Patel, Alexander S Ecker, et al. Deep learning-driven characterization of single cell tuning in primate visual area v4 unveils topological organization. *bioRxiv*, pages 2023–05, 2023.
- [118] Katrin Franke, Konstantin F Willeke, Kayla Ponder, Mario Galdamez, Na Zhou, Taliah Muhammad, Saumil Patel, Emmanouil Froudarakis, Jacob Reimer, Fabian H Sinz, et al. State-dependent pupil dilation rapidly shifts visual feature selectivity. *Nature*, 610(7930):128–134, 2022.

- [119] Lucas Theis, André 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.

Manuscripts

The following pages include the manuscripts from Chapter 3 [23] and Chapter 4 [24].

GENERALIZATION IN DATA-DRIVEN MODELS OF PRIMARY VISUAL CORTEX

Konstantin-Klemens Lurz,^{1,2,*} **Mohammad Bashiri**,^{1,2} **Konstantin Willeke**,^{1,2}
Akshay K. Jagadish,¹ **Eric Wang**,^{4,5} **Edgar Y. Walker**,^{1,2,4,5}
Santiago A. Cadena,^{2,3} **Taliah Muhammad**,^{4,5} **Erick Cobos**,^{4,5}
Andreas S. Tolias,^{4,5} **Alexander S. Ecker**,⁶ **Fabian H. Sinz**^{1-5, **}

¹ Institute for Bioinformatics and Medical Informatics, University of Tübingen, Germany

² International Max Planck Research School for Intelligent Systems, Tübingen, Germany

³ Bernstein Center for Computational Neuroscience, University of Tübingen, Germany

⁴ Department for Neuroscience, Baylor College of Medicine, Houston, TX, USA

⁵ Center for Neuroscience and Artificial Intelligence, Baylor College of Medicine, Houston, TX, USA

⁶ Department of Computer Science / Campus Institute Data Science, University of Göttingen, Germany

*konstantin.lurz@uni-tuebingen.de, **fabian.sinz@uni-tuebingen.de

ABSTRACT

Deep neural networks (DNN) have set new standards at predicting responses of neural populations to visual input. Most such DNNs consist of a convolutional network (core) shared across all neurons which learns a representation of neural computation in visual cortex and a neuron-specific readout that linearly combines the relevant features in this representation. The goal of this paper is to test whether such a representation is indeed generally characteristic for visual cortex, i.e. generalizes between animals of a species, and what factors contribute to obtaining such a generalizing core. To push all non-linear computations into the core where the generalizing cortical features should be learned, we devise a novel readout that reduces the number of parameters per neuron in the readout by up to two orders of magnitude compared to the previous state-of-the-art. It does so by taking advantage of retinotopy and learns a Gaussian distribution over the neuron’s receptive field position. With this new readout we train our network on neural responses from mouse primary visual cortex (V1) and obtain a gain in performance of 7% compared to the previous state-of-the-art network. We then investigate whether the convolutional core indeed captures *general* cortical features by using the core in transfer learning to a different animal. When transferring a core trained on thousands of neurons from various animals and scans we exceed the performance of training directly on that animal by 12%, and outperform a commonly used VGG16 core pre-trained on imagenet by 33%. In addition, transfer learning with our data-driven core is more data-efficient than direct training, achieving the same performance with only 40% of the data. Our model with its novel readout thus sets a new state-of-the-art for neural response prediction in mouse visual cortex from natural images, generalizes between animals, and captures better characteristic cortical features than current task-driven pre-training approaches such as VGG16.

1 INTRODUCTION

A long lasting challenge in sensory neuroscience is to understand the computations of neurons in the visual system stimulated by natural images (Carandini et al., 2005). Important milestones towards this goal are general system identification models that can predict the response of large populations of neurons to arbitrary visual inputs. In recent years, deep neural networks have set new standards in predicting responses in the visual system (Yamins et al., 2014; Vintch et al., 2015; Antolík et al., 2016; Cadena et al., 2019a; Batty et al., 2016; Kindel et al., 2017; Klindt et al., 2017; Zhang et al., 2018; Ecker et al., 2018; Sinz et al., 2018) and the ability to yield novel response characterizations (Walker et al., 2019; Bashivan et al., 2019; Ponce et al., 2019; Kindel et al., 2019; Ukita et al., 2019).

Such a general system identification model is one way for neuroscientists to investigate the computations of the respective brain areas *in silico*. Such *in silico* experiments exhibit the possibility to study the system at a scale and level of detail that is impossible in real experiments which have to cope with limited experimental time and adaptation effects in neurons. Moreover, all parameters, connections and weights in an *in silico* model can be accessed directly, opening up the opportunity to manipulate the model or determine its detailed tuning properties using numerical optimization methods. In order for the results of such analyses performed on an *in silico* model to be reliable, however, one needs to make sure that the model does indeed replicate the responses of its biological counterpart faithfully. This work provides an important step towards obtaining such a generalizing model of mouse V1.

High performing predictive models need to account for the increasingly nonlinear response properties of neurons along the visual hierarchy. As many of the nonlinearities are currently unknown, one of the key challenges in neural system identification is to find a good set of characteristic nonlinear basis functions—so called *representations*. However, learning these complex nonlinearities from single neuron responses is difficult given limited experimental data. Two approaches have proven to be promising in the past: *Task-driven* system identification networks rely on transfer learning and use nonlinear representations pre-trained on large datasets for standard vision tasks, such as object recognition (Yamins & DiCarlo, 2016). Single neuron responses are predicted from a particular layer of a pre-trained network using a simple readout mechanism, usually an affine function followed by a static nonlinearity. *Data-driven* models share a common nonlinear representation among hundreds or thousands of neurons, and train the entire network end-to-end on stimulus response pairs from the experiment. Because the nonlinear representation is shared, it is trained via massive multi-task learning (one neuron—one task) and can be learned even from limited experimental data.

Task-driven networks are appealing because they only need to fit the readout mechanisms on top of a given representation and thus are data-efficient in terms of the number of stimulus-response pairs needed to achieve good predictive performance (Cadena et al., 2019a). Moreover, as their representations are obtained independently of the neural data, a good predictive performance suggests that the nonlinear features are characteristic for a particular brain area. This additionally offers the interesting normative perspective that the functional representations in deep networks and biological vision could be aligned by common computational goals (Yamins & DiCarlo, 2016; Kell et al., 2018; Kubilius et al., 2018; Nayebi et al., 2018; Sinz et al., 2019; Güçlü & van Gerven, 2014; Kriegeskorte, 2015; Khaligh-Razavi & Kriegeskorte, 2014; Kietzmann et al., 2019). In order to quantify the fit of the normative hypothesis, it is important to compare a given representation to other alternatives (Schrimpf et al., 2018; Cadena et al., 2019a). However, while representations pre-trained on ImageNet are the state-of-the-art for predicting visual cortex in primates (Cadena et al., 2019a; Yamins & DiCarlo, 2016), recent work has demonstrated that pre-training on object categorization (VGG16) yields no benefits over random initialization for mouse visual cortex (Cadena et al., 2019b). Since random representation should not be characteristic for a particular brain area and other tasks that might yield more meaningful representations have not been found yet, this raises the questions whether there are better ways to obtain a generalizing nonlinear representation for mouse visual cortex.

Here, we investigate whether such a generalizing representation can instead be obtained from *data-driven* networks. For this purpose, we develop a new data efficient readout which is designed to push non-linear computations into the core and test whether this core has learned general characteristic features of mouse visual cortex by applying the same criteria as for the task-driven approach: The ability to predict a population of unseen neurons in a new animal (transfer learning). Specifically, we make the following contributions: ① We introduce a novel readout mechanism that keeps the number of per-neuron parameters at a minimum and learns a bivariate Gaussian distribution for the readout position from anatomical data using retinotopy. With this readout alone, we surpass the previous state-of-the-art performance in direct training by 7%. ② We demonstrate that a representation pre-trained on thousands of neurons from various animals generalizes to neurons from an *unseen animal* (transfer learning). It exceeds the direct training condition by another 11%, setting the new state-of-the-art and outperforms a *task-driven* representation—trained on object recognition—by about 33%. ③ We then show that this generalization can be attributed to the *representation* and not the readout mechanism, indicating that the data-driven core indeed captures generalizing features of cortex: A representation trained on a single experiment (4.5k examples) in combination with a readout trained on anatomically matched neurons from four experiments (17.5k examples) did not achieve this performance. ④ Lastly, we find that transfer learning with our data-driven core is more data-efficient than direct training, achieving the same performance with only 40% of the data.

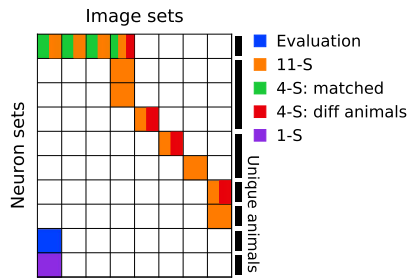


Figure 1: **Scans and training sets.** Overview of the datasets and how they are combined into different training sets. Each scan was performed on a specific set of neurons (rows) using a specific set of unique images (columns). Repeatedly presented test images were the same for all scans. Some scans were performed on the same neuron but with different image sets (first row). Colors indicate grouping of scans into training sets and match line colors in Fig. 5 to indicate which dataset a representation/core (not the readout) was trained on.

2 METHODS

2.1 DATA

Functional data The data used in our experiments consists of pairs of neural population responses and grayscale visual stimuli sampled and cropped from ImageNet, isotropically downsampled to 64×36 px, with a resolution of 0.53 ppd (pixels per degree of visual angle). The neural responses were recorded from layer L2/3 of the primary visual cortex (area V1) of the mouse, using a wide field two photon microscope (Sofroniew et al., 2016). Activity was measured using the genetically encoded calcium indicator GCaMP6s. V1 was targeted based on anatomical location as verified by numerous previous experiments performing retinotopic mapping using intrinsic imaging. We selected cells based on a classifier for somata on the segmented cell masks and deconvolved their fluorescence traces (Pnevmatikakis et al., 2016). We did not filter cells according to visual responsiveness. The stimulation paradigm and data pre-processing followed the procedures described by Walker et al. (2019). A single scan contained the responses of approximately 5000–9000 neurons to up to 6000 images, of which 1000 images consist of 100 unique images which were presented 10 times each to allow for an estimate of the reliability of the neuron (see Appendix for a detailed description of the datasets). We used the repeated images for testing, and split the rest into 4500 training and 500 validation images. The neural data was preprocessed by normalizing the responses of the neurons by their standard deviation on the training set. To put the number of recorded neurons per scan into perspective, assuming that V1 has an area of about 4mm^2 , that L2/3 is about $150\text{--}250\mu\text{m}$ thick and has a cell density of 80k excitatory cells per mm^3 , entire V1 L2/3 should contain about $48\text{k} - 80\text{k}$ neurons (Garrett et al., 2014; Jurjut et al., 2017; Schüz & Palm, 1989), similar to the maximum number of neurons that we train a model on (72k neuron, 11-S, Fig. 1, orange). Note, however, that this does not mean that these 72k neurons sample V1 or the visual field of a mouse evenly because of possible experimental biases in the choice of the recording location.

All together, we used 13 scans from a total of 7 animals (Fig. 1). Each scan is defined by the set of neurons it was performed on (rows/neuron sets in Fig. 1) and the set of images that were shown (columns/image sets in Fig. 1). Different image sets had non-overlapping training/validation images, but the same test images. Some of the scans were performed on the same neurons, but with different sets of natural images (first row in Fig. 1). These neurons were matched across scans by cross-correlating the structural scan planes against functionally recorded stacks (Walker et al., 2019). Stitching data from several scans in this way allowed us to increase the number of image presentations per neuron beyond what would be possible in a single scan. We combined these scans into different training sets (one color—one training set in Fig. 1) and named each one of them—e.g. 11-S for a set with 11 Scans. The different sets are further explained in the respective experiments they are used in. All data from the seven mice used in this work has been recorded by trained personnel under a strict protocol according to the regulations of the local authorities at Balor College of Medicine.

2.2 NETWORKS AND TRAINING

The networks are split conceptually into two parts: a *core* and a *readout*. The core captures the nonlinear image representation and is shared among all neurons. The readout maps the features of the core into neural responses and contains all neuron specific parameters.

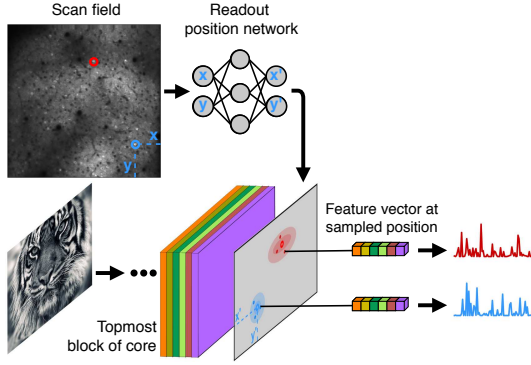


Figure 2: **Using retinotopy to learn the readout position from anatomical data.**

The Gaussian readout for each neuron uses features from a single location on the final tensor of the core CNN (bottom). The position is drawn from a 2D Gaussian for every image during training. The parameters of the Gaussian for each neuron are learned during training. The means of the Gaussians are predicted from each neuron’s coordinates on cortex by a *Readout Position Network* whose weights are shared across neurons and learned during training (top). During testing, the mean of the Gaussian is used as the neuron’s position.

Representation/Core We model the core with a four-layer convolutional neural network (CNN), with 64 feature channels per layer. In each layer, the 2d-convolutional layer is followed by a batch normalization layer and an ELU nonlinearity (Ioffe & Szegedy, 2015; Clevert et al., 2015). All convolutional layers after the first one are depth-separable convolutions (Chollet, 2017) which we found to yield better results than standard convolutional layers in a search among different architecture choices.

Readouts We compared two different types of readouts to map the nonlinear features of the core to the response of each neuron. For each neuron, a tensor of $\mathbf{x} \in \mathbb{R}^{w \times h \times c}$ (width, height, channels) needs to be mapped to a single scalar, corresponding to the target neuron’s response. All of our readouts assume that this function is affine with a linear weight tensor $\mathbf{w} \in \mathbb{R}^{w \times h \times c}$, followed by an ELU offset by one (ELU+1), to keep the response positive. Furthermore, both readouts assume that in feature space the receptive field of each neuron does not change its position across features, but they differ in how this receptive field location is constrained and learned.

The *factorized readout* (Klindt et al., 2017) factorizes the 3d readout tensor into a lower-dimensional representation by using a spatial mask matrix u_{ij} and a vector of feature weights v_k , i.e. $w_{ijk} = u_{ij}v_k$. The spatial mask u_{ij} is restricted to be positive and encouraged to be sparse through an L1 regularizer.

Our novel *Gaussian readout* reduces the number of per-neuron parameters. It computes a linear combination of the feature activations at a single spatial position—parametrized as (x, y) coordinates—via bilinear interpolation (Sinz et al., 2018). To facilitate gradient flow during training, we replace the spatial downsampling used in (Sinz et al., 2018) by a sampling step, which during training draws the readout position of each n^{th} neuron from a bivariate Gaussian distribution $\mathcal{N}(\mu_n, \Sigma_n)$ for each image in a batch separately. This is the sampling version of (St-Yves & Naselaris, 2017) where the readout location is weighted spatially with a Gaussian profile. In our case, μ_n and Σ_n are learned via the reparametrization trick (Kingma & Welling, 2014). Initializing Σ_n large enough ensures that there is gradient information available to learn μ_n reliably. During training, Σ_n shrinks as the estimate of the neuron position improves. During evaluation we always use the position defined by μ_n , making the readout deterministic. This version of the Gaussian readout has $c + 7$ parameters per neuron (2 for μ , 4 for Σ because the linear mapping in the reparametrization trick is 2×2 , and 1 for the scalar bias).

The second innovation of our Gaussian readout is to couple the location estimation of single neurons by exploiting the retinotopic organization of primary visual cortex (V1) and other areas. Since V1 preserves the topology of visual space, we estimate a neuron’s receptive field location from its position $\mathbf{p}_n \in \mathbb{R}^2$ along the cortical surface available from the experiments. To that end, we learn a common function $\mu_n = f(\mathbf{p}_n)$ represented by a neural network that is shared across all neurons (Fig. 2). Since we work with neurons from local patches of V1, we model f as a linear fully connected network. This approach turns the problem of estimating each neuron’s receptive field location from limited data into estimating a single linear transformation shared by all neurons, and reduces the number of per-neuron parameters to $c + 5$. We initialized the *Readout Position Network* to a random orthonormal 2-2 matrix scaled by a factor which was optimized in hyper-parameter selection.

Finally, when training on several scans of anatomically matched neurons from the same mouse (see [Data](#)), we share the feature weights v_k across scans. To account for differences in spike inference between scans, we introduced a scan-specific scale and bias for each neuron after the linear readout. We mention in the respective sections whether features are shared or not. The bias of each readout is initialized with the average response on the training set. The effects of both feature sharing and learning from cortical anatomy on the performance of the readout are shown in the Appendix.

Training The networks were trained to minimize Poisson loss $\frac{1}{m} \sum_{i=1}^m (\hat{r}^{(i)} - r^{(i)} \log \hat{r}^{(i)})$ where m denotes the number of neurons, $\hat{r}$ the predicted neuronal response and r the observed response. We used early stopping on the correlation between predicted and measured neuronal responses on the validation set ([Prechelt, 1998](#)): if the correlation failed to increase during any 5 consecutive passes through the entire training set (epochs), we stopped the training and restored the model to the best performing model over the course of training. We found that this combination of Poisson objective and early stopping on correlation yielded the best results. After the first stop, we decreased the learning rate from 5×10^{-3} twice by a decay factor of 0.3, and resumed training until it was stopped again. Network parameters were iteratively optimized via stochastic gradient descent using the Adam optimizer ([Kingma & Ba, 2015](#)) with a batch size of 64. Once training completed, the trained network was evaluated on the validation set to yield the score used for hyper-parameter selection. The hyper-parameters were then selected with a Bayesian search ([Snoek et al., 2012](#)) of 100 trials and subsequently kept fixed throughout all experiments. Only the scale of the readout regularization was fine-tuned with additional Bayesian searches for the cases of different amounts of data independently. In transfer experiments, we froze all parameters of the core and trained a new readout only.

Evaluation We report performance as *fraction oracle* (see [Walker et al., 2019](#)), which is defined as the correlation of the predicted response and the observed single-trial test responses relative to the maximally achievable correlation measured from repeated presentations. We estimated the oracle correlation using a jackknife estimator (correlation of leave-one-out mean against single trial). Per data point, we trained 25 networks for all combinations of five different model initializations and five random partitions of the neurons into core and transfer sets. The image subsets were drawn randomly once and kept fixed across all experiments except in [Fig. 5](#) where the full neuron set was used and 5 random partitions of image subsets were drawn instead. We selected the best performing models across initializations and calculated 95% confidence intervals over neuron- or image seeds.

3 RESULTS

We investigated the conditions under which a *data-driven* core generalizes to new neurons in the same or different animals. We did this by pre-training a core on differently composed datasets (core sets, see [3.1](#)) and testing that core in transfer learning to a new set of neurons (transfer set, see [3.1](#)). Our main finding is that transferring a core trained on multiple scans (up to 35k unique images and 70k unique neurons) to a new set of 5335 unique neurons from a single scan (4.5k unique images) yields an improvement in performance of about 12% compared to a network directly trained on the single scan. This result was independent of whether the transferred core was trained on the same animal or not. By carefully choosing the [Transfer learning conditions](#), we can attribute this boost in performance to the generalization of the core.

3.1 TRANSFER LEARNING CONDITIONS

To test the generalization performance of the core, we used a separate set of 1000 neurons which we call *transfer set*. These neurons were not used to train the transferred core but only to fine-tune a new readout (note that [Fig. 4](#) and [5](#) use different transfer sets, for intra- and inter- animal transfer respectively). A transfer set is a subset of neurons, not images. Thus, each transfer set also had a train, validation, and test split of its images. We compared the performance in transfer learning to that of a network directly trained end-to-end on the transfer set or subsets thereof (*direct* condition). Because the core of the *direct* condition was not transferred, it could adapt to the neurons at hand giving it a fair chance to outperform the transferred cores.

In the transfer learning conditions (all except *direct*), the core was always trained on a separate set of neurons (*core set*) and subsequently frozen, while the readout was always trained on top of

the frozen core using the transfer set. Again, a core set is a subset of the neurons, not images. In order to quantify the generalization of the core, we needed to decouple the amount of data it takes to train the core from the amount of data it takes to train the readout. We thus considered two transfer conditions: *diff-core/best-readout* and *best-core/diff-readout*. In the *best-core/diff-readout* condition, the core was trained on the core set using all images, while the readout was trained on the transfer set using different numbers of images. This condition tests the data efficiency of the readout. We expect that better cores lead to a higher data efficiency in the readout, *i.e.* require less data to achieve good performance. In the *diff-core/best-readout* condition, we trained the core on the core set using different numbers of images while the readout was trained on the transfer set using all images. Thereby we tested how the generalization of the core is affected by the amount of data used to train it.

3.2 DIRECT TRAINING AND WITHIN ANIMAL/ACROSS NEURON GENERALIZATION

The following transfer experiments (Fig. 3 and 4) aim at investigating how the number of images and neurons provided to the core and the readout affects generalization to new neurons. To this end, we used a dataset that includes as many images as possible while still providing a reasonably large number of neurons. The 4-S:matched dataset (first row in Fig. 1, green) provides 17596 images and 4597 neurons anatomically matched across 4 scans. Each scan was performed on the same neurons, but showing different sets of images (test set was identical). The dataset was concatenated along the image dimension and split into core and transfer sets of 3597 and 1000 neurons.

Direct training We used the 4-S:matched core set to compare the performance of the Gaussian and factorized readout in the *direct* condition on the original core set before transfer. We tested both readouts with and without feature sharing (see [Networks and training](#)). The performance of the networks increased with the number of images (Fig. 3). It also increased with the number of neurons, but the number of images had a far stronger effect. While the performance saturated quickly with the number of neurons, saturation w.r.t. the number of images did not seem to be reached, even when using all 17.5k images. The Gaussian readout outperformed the factorized readout in predictive performance by 7% fraction oracle for the full 4-S:matched dataset, reaching 0.886 ± 0.005 and 0.826 ± 0.005 fraction oracle respectively (mean $\pm$ std). While the Gaussian readout profits from feature sharing, the factorized readout is hurt by it (Fig. 3, light vs. dark colors). This might be because the spatial masks in the factorized readout are less constrained in contrast to the Gaussian readout where the position network and the usage of only a single readout point exerts a stronger inductive bias. In all future experiments, we thus use feature sharing only for the Gaussian readout.

Within animal/across neuron generalization For both readouts, the generalization performance of the learned core, tested in the *diff-core/best-readout* condition, increased with the number of images used to train the core (Fig. 4, pink). The cores and readouts were trained on the core and transfer set of the 4-S:matched dataset. As before, the Gaussian readout outperformed the factorized readout, exhibiting a stronger increase in performance with the number of images and a better final performance when the entire dataset was used to train the core. Even for a core trained on few data, a readout can yield good performance if it has access to enough images (pink line). Importantly,

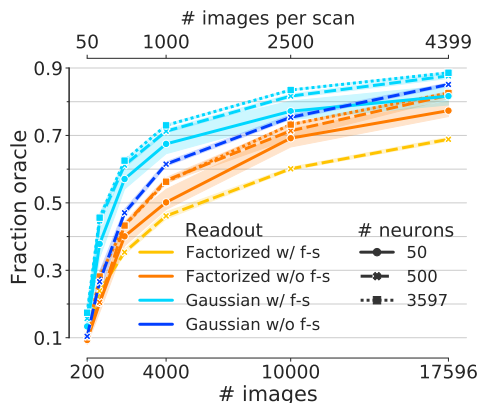


Figure 3: **Performance of end-to-end trained networks.** Performance for different subsets of neurons (linestyle) and number of training examples (x-axis). The same core architecture was trained for two different readouts with and without feature sharing (color) on the matched neurons of the 4-S:matched core set (Fig. 1, green). Both networks show increasing performance with number of images. However, the network with the Gaussian readout achieves a higher final performance (light blue vs. orange). While the Gaussian readout profits from feature sharing (light vs. dark blue), the factorized readout is hurt by it (yellow vs. orange). Shaded areas depict 95% confidence intervals across random picks of the neuron subsets.

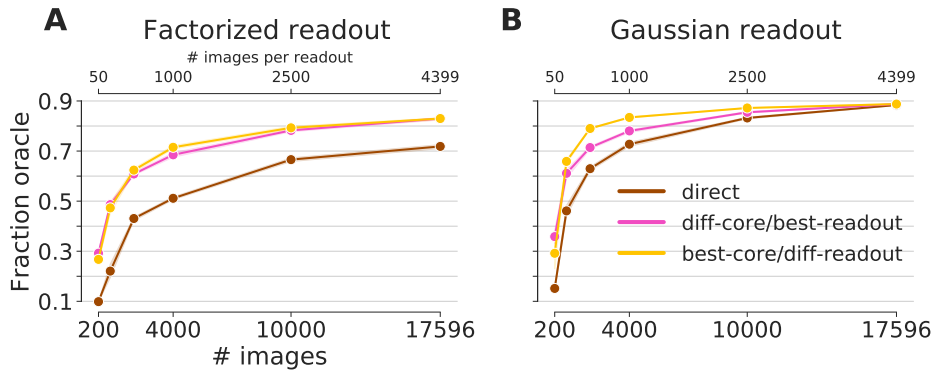


Figure 4: **Generalization to other neurons in the same animal.** A core trained on 3597 neurons and up to 17.5k images generalizes to new neurons (pink and yellow line). A core trained on the full data yields very good predictive performance even when the readout is trained on far less data (yellow). If the readout is trained with all data, even a core trained on few data can yield a good performance (pink). Both transfer conditions outperform a network directly trained end-to-end on the transfer dataset (brown). For the full dataset, all training conditions converge to the same performance. Except in the *best-core/diff-readout* condition for very few training data, the Gaussian readout (B) outperforms the factorized readout (A). The data for both the training and transfer comes from the 4-S:matched dataset (Fig 1, green). Not that the different number of images can be from the core or transfer set, depending on the transfer condition.

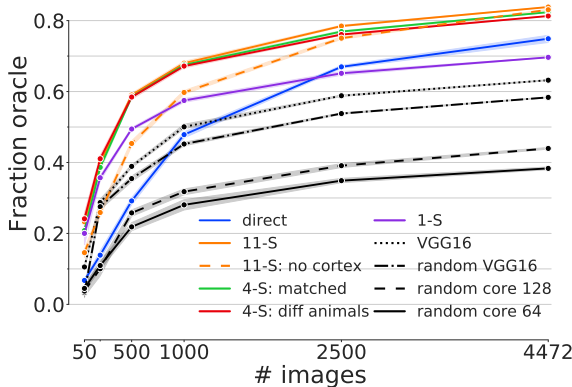


Figure 5: **Generalization across animals.** Prediction performance in fraction oracle correlation as a function of training examples in the transfer set for a Gaussian readout (x-axis) and different ways to obtain the core (colors). The transfer training was performed on the evaluation dataset (blue, Fig 1). Cores trained on several scans used in transfer learning outperform direct training on the transfer dataset (blue line; direct condition).

except for very low numbers of images, the fastest increase in performance occurred in the *best-core/diff-readout* condition, where a core trained on all images (but different neurons) was frozen and a new readout was trained on the transfer set for varying numbers of images (yellow lines). This result shows that a data-driven core provides general characteristic features for mouse V1, and these features generalize to new neurons. Importantly, for 4k images (about the size of a typical experiment), the performance of the *best-core/diff-readout* was approximately 7% better than the performance of the *diff-core/best-readout* condition (0.834 ± 0.003 and 0.780 ± 0.007 fraction oracle respectively). This observation that a readout on all 17.5k images on a core from 4k images could not reach the performance of a readout on 4k images on a core from 17.5k images suggests that the better performance is due to the generalizing core and not the readout.

3.3 GENERALIZATION ACROSS ANIMALS

So far, we tested generalization performance of data-driven cores to different neurons in the same animal using the 4-S:matched dataset. A stronger test for generalization is transfer learning across animals, where we may have to deal with inter-subject variability. To this end, we compared several cores derived from different core sets, random initialization, or ImageNet pre-training in terms of their generalization to neurons from a mouse which was not used to train the core. The transfer set consisted of 5335 neurons from a different mouse presented with images that were also in the

core set (Fig. 1, blue). Note that performance was still evaluated on a set of test images that were neither used to train core nor readout. Apart from the previously used core set with anatomically matched neurons (4-S:matched, Fig. 1, green), we also trained a core on a single scan from another animal (1-S, Fig. 1, purple), and on a set of four scans from different animals with different images (4-S: diff animals, Fig. 1, red). Finally we also trained a core on all datasets together, without any information on neuron matching or image sets (11-S, Fig. 1, orange). For completeness, we also compared to a task-driven VGG core (Simonyan & Zisserman, 2015), a randomly initialized VGG core (reading out from conv2-2 for both) (Cadena et al., 2019b), our core (64 features) with randomly initialized weights, and a scaled up version of our core with the same number of features as the VGG (128). As before, we compared the generalization performance to a model trained under the *direct* condition trained on the transfer set. Due to its better performance, we used the Gaussian readout for these experiments but a version with the factorized readout can be found in the Appendix.

We tested the generalization in the *best-core/diff-readout* condition on a single transfer set (Fig. 5), *i.e.* all performances are reported on the same transfer set (5335 neurons) (blue, Fig. 1 and 5), but differ in where the core was trained (colors, Fig. 5). The most striking finding is that representations trained on several scans not only reached better performances for fewer training images, but they also reached a better overall performance for the available data from a single scan compared to *direct* training (Fig. 5 orange, green, and red vs. blue). Interestingly, this improvement did not require matched neurons in the core set (Fig. 5) and any potential negative effects of inter-subject variability were outweighed by the benefits of using multiple scans to train the core. The two cores trained on a single scan reached about the same final performance (Fig. 5, blue vs. purple), with a slightly better performance for the directly trained model, as expected. However, the final performance of both single scan models was about 10% smaller than the transfer performance from the models pre-trained on four and eleven scans, respectively (Fig. 5, blue and purple vs. red, green, and orange). We provide visualizations of the receptive fields – produced via response maximization (Walker et al., 2019; Bashivan et al., 2019) – of some example neurons obtained from our best model (Fig. 5, orange at 4472 images) in the Appendix, Fig. 4.

Consistent with previous work (Cadena et al., 2019b), pre-trained and random VGG16 cores performed similarly (Fig. 5 gray, dotted vs. dash-dotted). Both VGG cores performed worse than a directly trained *data-driven* core (Fig. 5 blue vs. gray dotted and dash-dotted). Our core with random weights (64 features) performs worst, demonstrating that training on neural data extracts characteristic features. Scaling up this random core to VGG size (128) does not match its performance which could be due to the interaction of the initialization with architectural differences.

Lastly, we investigated the effect of constraining all neurons to share the transformation from cortical location to receptive field location by temporarily deactivating this feature (Fig. 5, 11-S: no cortex dashed orange), and found that this constraint was particularly useful for small numbers of images. An equivalent figure for Fig. 5 for the factorized readout, a table with a detailed overview over the most important results in numerical form, as well as a comparison with other performance metrics can be found in the Appendix.

4 RELATED WORK

The idea of a common cortical feature representation is wide-spread in sensory and systems neuroscience, going back to the idea of V1 as a bank of Gabor filters or edge detectors (Jones et al., 1987; Olshausen & Field, 1996). A substantial body of recent work focuses on feature representations learned by training deep networks on vision tasks such as object recognition (Cadena et al., 2019a; Yamins & DiCarlo, 2016; Güçlü & van Gerven, 2014; Kriegeskorte, 2015; Khaligh-Razavi & Kriegeskorte, 2014). The brain-score¹ initiative compares different representations, resulting from pre-training on different tasks or different network architectures, with regards to performance of multiple neural prediction tasks (Schrimpf et al., 2018). In contrast to that, we focused on whether multi-task learning between thousands of neurons leads to a generalizing representation.

While *task-driven* representations perform comparably to *data-driven* representations in primates (Cadena et al., 2019a), Cadena et al. (2019b) recently demonstrated that they show almost no difference in predictive performance for mice. Our results corroborate this finding (Fig. 5) and show that a

¹<https://www.brain-score.org/>

data-driven representation outperforms a *task-driven* representation by a substantial margin, even when tested on equal grounds with transfer learning.

Sharing a representation between neurons is commonly used to learn *data-driven* system identification networks (Cadena et al., 2019b;a; Batty et al., 2016; Sinz et al., 2018; Antolík et al., 2016; Klindt et al., 2017). Klindt et al. (2017) investigated the effect of the number of neurons and training examples onto the predictive performance of a data-driven network. However, this experiment was done on simulated data only and did not explore the generalization (transfer) performance of the learned representation. To the best of our knowledge, we are the first to systematically investigate the ability of data-driven representations to capture general characteristic features of visual cortex.

5 DISCUSSION

Machine learning applications in biology are often faced with limited amount of data. Especially, for recent deep learning approaches this poses a challenging problem. One promising way to approach it is multi-task learning by training a shared nonlinear representation on multiple tasks or subjects. This increases the data volume and can help to extract inductive biases to achieve better generalization. Here, we investigated a particular instance of this problem: Modeling the responses of thousands of cortical neurons as a function of natural visual stimuli. We demonstrated that nonlinear *data-driven* representations, trained via massive multi-task learning through parameter sharing among thousands of neurons from mouse primary visual cortex, generalize to other neurons and mice, and significantly outperform common *task-driven* alternatives that are predictive for monkey V1 (Cadena et al., 2019b; Yamins & DiCarlo, 2016). As noted by Cadena et al. (2019b) already, this does not imply that all task-trained representations are necessarily suboptimal but rather indicates that we have not found the right task yet. But so far, our network sets a new state-of-the-art for neural response prediction in direct training *as well as* transfer learning.

Our transfer results strongly suggests that data trained cores can capture features that are characteristic of mouse primary visual cortex, and the fact that the Gaussian readout can predict novel neurons from this core with relatively few training example corroborates this idea. In addition, the good transfer learning performance indicates that inter-subject variability, which could affect the success of multi-task learning, does not seem to be a major problem for this application. For that reason, we believe that our *data-driven* core is the most characteristic representation to date to predict mouse primary visual cortex and that it could be a great tool in new experiments where data is scarce and/or training time is limited, such as the *inception loops* introduced by Walker et al. (2019) and Bashivan et al. (2019). To facilitate this, we share the weights of the trained representation together with its code online² to allow others to predict neural responses with it. Other possible applications include for example the analysis of the learned feature representations to investigate the operations in visual processing, or using the core to support vision tasks like image categorization (Li et al., 2019). Additionally, we also share the dataset that we evaluate our core on (Fig. 1, blue) so that other representations can be tested and compared with ours on the same data³.

Our results are based on “deconvolved” calcium traces of neural activity integrated over 500 ms. Whether data driven cores can also provide generalizing features for shorter time scales, recordings with electrodes, to dynamic responses, or of higher areas remains to be seen in future studies.

ACKNOWLEDGMENTS

We thank A.K. Schalkamp, A. Nix, C. Blessing, S. Safarani, E. Froudarakis, and N. Patel for comments and discussions. 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. SAC and ASE are supported by the German Research Foundation (DFG grant EC-479/1-1). This work was supported by an AWS Machine Learning research award to FHS. MB was supported by the International Max Planck Research School for Intelligent Systems. Supported by the Intelligence Advanced Research Projects Activity via Department of Interior/Interior Business Center contract number D16PC00003.

²https://github.com/sinzlab/Lurz_2020_code

³<https://gin.g-node.org/cajal/Lurz2020>

REFERENCES

- J. Antolík, S. B. Hofer, J. A. Bednar, and T. D. Mrsic-flogel. Model Constrained by Visual Hierarchy Improves Prediction of Neural Responses to Natural Scenes. *PLoS Comput Biol*, pp. 1–22, 2016. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1004927.
- P. Bashivan, K. Kar, and J. DiCarlo. Neural Population Control via Deep ANN Image Synthesis. pp. 1–33, 2019. doi: 10.32470/ccn.2018.1222-0.
- E. Batty, J. Merel, N. Brackbill, A. Heitman, A. Sher, A. Litke, E. J. Chichilnisky, and L. Paninski. Multilayer network models of primate retinal ganglion cells. Number Nips, 2016.
- S. A. Cadena, G. H. Denfield, E. Y. Walker, L. A. Gatys, A. S. Tolias, M. Bethge, and A. S. Ecker. Deep convolutional models improve predictions of macaque V1 responses to natural images. *PLoS Comput Biol*, pp. 201764, 2019a. doi: 10.1101/201764. URL <https://www.biorxiv.org/content/early/2017/10/11/201764.article-info>.
- S. A. Cadena, F.H. Sinz, T. Muhammad, E. Froudarakis, E. Cobos, E. Y. Walker, J. Reimer, M. Bethge, and A. S. Ecker. How well do deep neural networks trained on object recognition characterize the mouse visual system? In *NeurIPS 2019 Workshop Neuro AI*, 2019b.
- M. Carandini, J. B. Demb, V. Mante, D. J. Tolhurst, Y. Dan, B. A. Olshausen, J. L. Gallant, and N. C. Rust. Do We Know What the Early Visual System Does? *J. Neurosci.*, 25(46):10577–10597, 11 2005. doi: 10.1523/JNEUROSCI.3726-05.2005.
- François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017. ISBN 9781538604571. doi: 10.1109/CVPR.2017.195.
- Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). pp. 1–14, 2015. ISSN 09226389. doi: 10.3233/978-1-61499-672-9-1760. URL <http://arxiv.org/abs/1511.07289>.
- A. S. Ecker, F. H. Sinz, E. Froudarakis, P. G. Fahey, S. A. Cadena, E. Y. Walker, E. Cobos, J. Reimer, A. S. Tolias, and M. Bethge. A rotation-equivariant convolutional neural network model of primary visual cortex. pp. 5–7, 2018. doi: arXiv:1809.10504v1. URL <http://arxiv.org/abs/1809.10504>.
- Marina E. Garrett, Ian Nauhaus, James H. Marshel, Edward M. Callaway, Marina E. Garrett, James H. Marshel, Ian Nauhaus, and Marina E. Garrett. Topography and areal organization of mouse visual cortex. *Journal of Neuroscience*, 34(37):12587–12600, sep 2014. ISSN 15292401. doi: 10.1523/JNEUROSCI.1124-14.2014. URL <https://www.jneurosci.org/content/34/37/12587><https://www.jneurosci.org/content/34/37/12587.abstract>.
- Umut Güçlü and Marcel A. J. van Gerven. Deep Neural Networks Reveal a Gradient in the Complexity of Neural Representations across the Brain’s Ventral Visual Pathway. Technical Report 27, 2014. URL <http://arxiv.org/abs/1411.6422><http://dx.doi.org/10.1523/JNEUROSCI.5023-14.2015><http://arxiv.org/abs/1411.6422v1>.
- Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. Technical report, 2015.
- J. P. Jones, Palmer L A, and L. A. Palmer. The two-dimensional spatial structure of simple receptive fields in cat striate cortex. *Journal of Neurophysiology*, 1987. ISSN 0022-3077.
- Ovidiu Jurjut, Petya Georgieva, Laura Busse, and Steffen Katzner. Learning enhances sensory processing in mouse V1 before improving behavior. *Journal of Neuroscience*, 37(27):6460–6474, jul 2017. ISSN 15292401. doi: 10.1523/JNEUROSCI.3485-16.2017. URL <https://sites.google.com/a/nyu>.
- Alexander J.E. Kell, Daniel L.K. Yamins, Erica N. Shook, Sam V. Norman-Haignere, and Josh H. McDermott. A Task-Optimized Neural Network Replicates Human Auditory Behavior, Predicts Brain Responses, and Reveals a Cortical Processing Hierarchy. *Neuron*, 2018. ISSN 10974199. doi: 10.1016/j.neuron.2018.03.044.

- S.-M. Khaligh-Razavi and N. Kriegeskorte. Deep Supervised , but Not Unsupervised , Models May Explain IT Cortical Representation. *PLoS computational biology*, 10(11), 2014. doi: 10.1371/journal.pcbi.1003915.
- Tim C. Kietzmann, Courtney J. Spoerer, Lynn K.A. Sørensen, Radosław M. Cichy, Olaf Hauk, and Nikolaus Kriegeskorte. Recurrence is required to capture the representational dynamics of the human visual system. *Proceedings of the National Academy of Sciences of the United States of America*, 116(43):21854–21863, 10 2019. ISSN 10916490. doi: 10.1073/pnas.1905544116.
- W. F. Kindel, Elijah D. Christensen, and Joel Zylberberg. Using deep learning to reveal the neural code for images in primary visual cortex. Technical report, 2017. URL <http://arxiv.org/abs/1706.06208>.
- William F Kindel, Elijah D Christensen, and Joel Zylberberg. Using deep learning to probe the neural code for images in primary visual cortex. *Journal of Vision*, 19(4):29, 10 2019. doi: 10.1167/19.4.29. URL https://app.dimensions.ai/details/publication/pub.1113752643https://jov.arvojournals.org/arvo/content_public/journal/jov/937940/i1534-7362-19-4-29.pdf.
- Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.
- Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 2014.
- D. A. Klindt, A. S. Ecker, T. Euler, and M. Bethge. Neural system identification for large populations separating "what" and "where". In *Advances in Neural Information Processing Systems*, number Nips, pp. 4–6, 2017. ISBN 978-1-5108-6096-4. doi: 10.12751/nncn.bc2017.0132. URL <http://arxiv.org/abs/1711.02653>.
- N. Kriegeskorte. Deep neural networks : a new framework for modelling biological vision and brain information processing. *Annual Reviews of Vision Science*, 2015. doi: 10.1101/029876.
- Jonas Kubilius, Martin Schrimpf, Aran Nayebi, Daniel Bear, Daniel L. K. Yamins, and James J. DiCarlo. CORnet: Modeling the Neural Mechanisms of Core Object Recognition. *bioRxiv*, pp. 408385, 2018. doi: 10.1101/408385. URL <https://www.biorxiv.org/content/10.1101/408385v1>.
- Zhe Li, Wieland Brendel, Edgar Y Walker, Erick Cobos, Taliah Muhammad, Jacob Reimer, Matthias Bethge, Fabian H Sinz, Xaq Pitkow, and Andreas S Tolias. Learning From Brains How to Regularize Machines. 2019. URL <http://arxiv.org/abs/1911.05072>.
- A. Nayebi, D. Bear, J. Kubilius, K. Kar, S. Ganguli, D. Sussillo, J. J. DiCarlo, and D. L. K. Yamins. Task-Driven Convolutional Recurrent Models of the Visual System. (NeurIPS):1–14, 2018. URL <https://arxiv.org/pdf/1807.00053.pdf><http://arxiv.org/abs/1807.00053>.
- B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 6 1996. doi: 10.1038/381607a0.
- Eftychios A. Pnevmatikakis, Daniel Soudry, Yuanjun Gao, Timothy A. Machado, Josh Merel, David Pfau, Thomas Reardon, Yu Mu, Clay Lacefield, Weijian Yang, Misha Ahrens, Randy Bruno, Thomas M. Jessell, Darcy S. Peterka, Rafael Yuste, and Liam Paninski. Simultaneous Denoising, Deconvolution, and Demixing of Calcium Imaging Data. *Neuron*, 2016. ISSN 10974199. doi: 10.1016/j.neuron.2015.11.037.
- 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. ISSN 10974172. doi: 10.1016/j.cell.2019.04.005. URL <http://dx.doi.org/10.1016/j.cell.2019.04.005>.

- Lutz Prechelt. Automatic early stopping using cross validation: Quantifying the criteria. *Neural Networks*, 11(4):761–767, 6 1998. ISSN 08936080. doi: 10.1016/S0893-6080(98)00010-0.
- M. Schrimpf, J. Kubilius, H. Hong, N. J. Majaj, R. Rajalingham, E. B. Issa, K. Kar, P. Bashivan, J. Prescott-Roy, K. Schmidt, D. L. K. Yamins, and J. J. DiCarlo. Brain-Score: Which Artificial Neural Network for Object Recognition is most Brain-Like? *bioRxiv*, pp. 407007, 2018. doi: 10.1101/407007. URL <https://www.biorxiv.org/content/early/2018/09/05/407007>.
- Almut Schüz and Günther Palm. Density of neurons and synapses in the cerebral cortex of the mouse. *Journal of Comparative Neurology*, 286(4):442–455, 1989. ISSN 10969861. doi: 10.1002/cne.902860404. URL <https://pubmed.ncbi.nlm.nih.gov/2778101/>.
- Karen Simonyan and Andrew Zisserman. VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION. Technical report, 2015. URL <http://www.robots.ox.ac.uk/>.
- F. Sinz, A. S. Ecker, P. Fahey, E. Walker, E. Cobos, E. Froudarakis, D. Yatsenko, X. Pitkow, J. Reimer, and A. Tolias. Stimulus domain transfer in recurrent models for large scale cortical population prediction on video. In *Advances in Neural Information Processing Systems 31*. 2018. doi: 10.1101/452672.
- Fabian H. Sinz, Xaq Pitkow, Jacob Reimer, Matthias Bethge, and Andreas S. Tolias. Engineering a Less Artificial Intelligence, 2019. ISSN 10974199.
- Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical Bayesian Optimization of Machine Learning Algorithms. Technical report, 2012.
- 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*, 2016. ISSN 2050084X. doi: 10.7554/eLife.14472.
- Ghislain St-Yves and Thomas Naselaris. The feature-weighted receptive field: an interpretable encoding model for complex feature spaces. *The feature-weighted receptive field: an interpretable encoding model for complex feature spaces*, 2017. ISSN 1095-9572. doi: 10.1101/126318.
- Jumpei Ukita, Takashi Yoshida, and Kenichi Ohki. Characterisation of nonlinear receptive fields of visual neurons by convolutional neural network. *Scientific Reports*, 9(1):3791, 2019. doi: 10.1038/s41598-019-40535-4.
- B. Vintch, J. A. Movshon, and E. P. Simoncelli. A Convolutional Subunit Model for Neuronal Responses in Macaque V1. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 35(44):14829–41, 2015. ISSN 1529-2401. doi: 10.1523/JNEUROSCI.2815-13.2015. URL <http://www.jneurosci.org/cgi/doi/10.1523/JNEUROSCI.2815-13.2015%5Cnhttp://www.ncbi.nlm.nih.gov/pubmed/26538653%5Cnhttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4635132>.
- E. Y. Walker, F. H. Sinz, E. Cobos, T. Muhammad, E. Froudarakis, P. G. Fahey, A. S. Ecker, J. Reimer, X. Pitkow, and A. S. Tolias. Inception loops discover what excites neurons most using deep predictive models. *Nature Neuroscience*, 2019. ISSN 15461726. doi: 10.1038/s41593-019-0517-x. URL <http://dx.doi.org/10.1038/s41593-019-0517-x>.
- D. L. K. Yamins and J. J. DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nature Neuroscience*, 19(3):356–365, 2016. ISSN 1097-6256. doi: 10.1038/nn.4244. URL <http://www.nature.com/doifinder/10.1038/nn.4244>.
- D. L. K. Yamins, H. Hong, C. F. Cadieu, E. A. Solomon, D. Seibert, and J. J. DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences of the United States of America*, 111(23):8619–24, 2014. ISSN 1091-6490. doi: 10.1073/pnas.1403112111. URL <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4060707&tool=pmcentrez&rendertype=abstract>.
- Yimeng Zhang, Tai Sing, Lee Ming, Li Fang, Liu Shiming, T.-S. Lee, M. Li, F. Liu, and S. Tang. Convolutional neural network models of V1 responses to complex patterns. Technical report, 2018.

GENERALIZATION IN DATA-DRIVEN MODELS OF PRIMARY VISUAL CORTEX (APPENDIX)

1 TWO PHOTON SCANS

The following table lists details about the datasets used. A session marks a continuous experimental session that can comprise several scans and in which the mouse does not leave the scanner. A scan is a single continuous recording of neural activity. Spike inference from the two photon fluorescence signal is performed on the scan level.

The column **matched** indicates whether neurons were anatomically matched between scans. The four scans from mouse 22564 had 4625 matched neurons.

animal_id	session	scan_idx	neurons	images	matched	in sets
20457	5	9	5335	5993	no	Evaluation
20505	6	1	8367	5996	no	1-S
22564	2	12	8115	5933	yes	4-S, 11-S
22564	2	13	8199	5955	yes	4-S, 11-S
22564	3	8	7916	5986	yes	4-S, 11-S
22564	3	12	8182	5967	yes	4-S, 11-S
22846	2	19	7700	5998	no	11-S
22846	2	21	8044	5947	no	11-S
22846	10	16	7344	5993	no	11-S
23343	5	17	7334	5927	no	11-S
23555	4	20	6848	5957	no	11-S
23555	5	12	6559	5994	no	11-S
23656	14	22	8107	5950	no	11-S

2 GENERALIZATION ACROSS ANIMALS (EXTENSION)

We showed in Fig 4 in the main paper that the Gaussian readout outperforms the factorized readout in transfer-learning, especially in the low data regime. Consequently we conducted the main transfer experiment, the generalization across animals (Fig 5 in the main paper), with the Gaussian readout. For completeness, we here show the same experiment with the factorized readout for the relevant transfer cores *11-S*, *1-S* and *VGG16* (Fig. 1, left). The exact numeric values for this experiment with full data (5335 neurons, 4472 images) for both readouts can be found in Fig. 1 on the right. Consistent with the previous experiments, the Gaussian readout outperforms the factorized readout for direct training as well as transfer learning with data-driven cores. Interestingly however, the factorized readout scores higher than the Gaussian readout when compared on the task driven transfer core (*VGG16*), levelling its performance with the transfer core from one dataset (*1-S*). We hypothesize that this is caused by the factorized readout’s less constrained spatial mask which can pool over more than one pixel in the final tensor and might thus enable it to compensate for the potentially suboptimal features in the *VGG16* core.

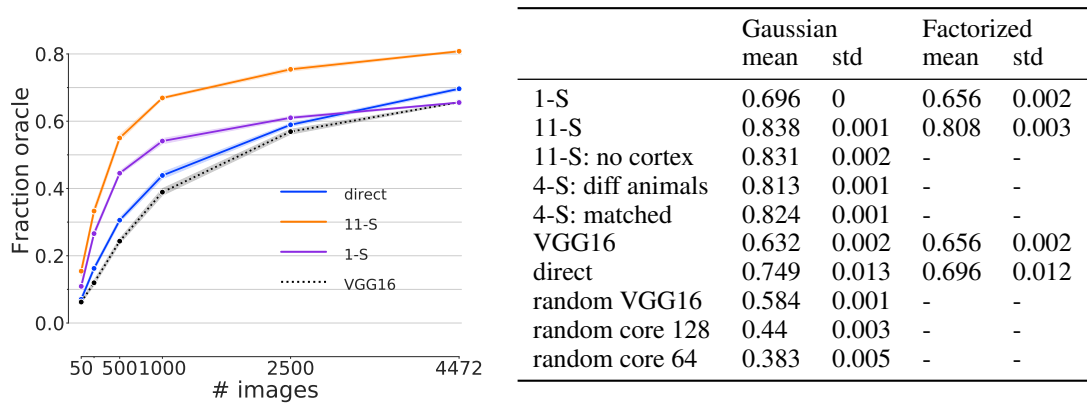


Figure 1: **Generalization across animals** (compare Fig 5 in the main paper). **Left:** Key experiments of Fig 5, conducted with the factorized readout. **Right:** Overview over the performances of the Gaussian and factorized readout models in the transfer-task across animals for full data (5335 neurons, 4472 images).

3 CONSISTENCY ACROSS OTHER PERFORMANCE MEASURES

Neural responses to (visual) stimuli suffer from trial-to-trial variability, even when keeping the input stimulus fixed. In order to get an unbiased estimate of the performance of a model that predicts such responses, the measure of performance needs to account for this statistical noise. Here we use the *fraction oracle* (Walker et al., 2019), see *Evaluation* in Section 2.2 *Networks and Training* in the main paper. However, there exists a variety of measures that attempt to tackle this issue and no standard measure has been established yet. Ideally, new findings should hold independently of the measure of performance and should be comparable across such measures. For this purpose we show the consistency of our main results (Fig. 5 in the paper), by comparing the *fraction oracle* to another measure, the *fraction of variance of the expected response* (r_{ER}^2) (Pospisil & Bair, 2020). The calculation of the r_{ER}^2 assumes that the variance over image repeats across unique images is constant. Note that this is not strictly true for our data, but to be able to compare the same model on equal ground, we chose to ignore the assumption for the sake of this comparison. Furthermore, the authors recommend that the signal-to-noise ratio of the data must be above a certain threshold (0.1 for data with 100 images and 10 repeats each, as in our case; see Fig. 14 in Pospisil & Bair (2020)). Our data meets this criterion (see Fig. 3). Figure 2 shows that both measures qualitatively yield the same results (same order of the curves).

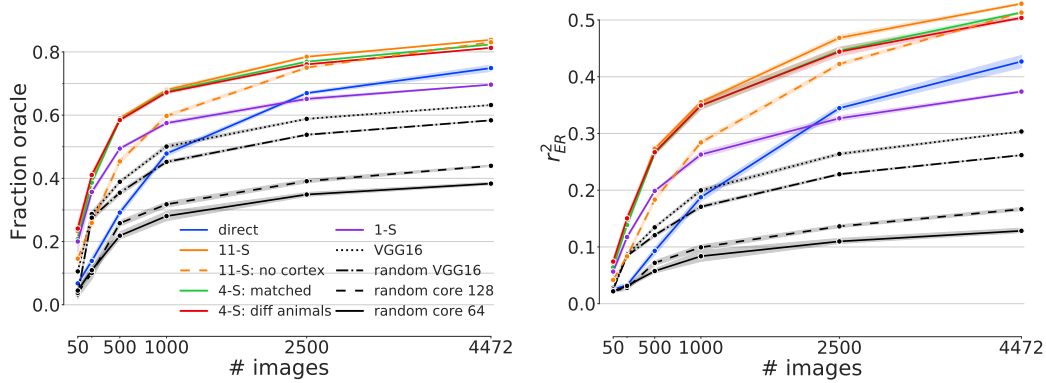


Figure 2: **Consistency across performance measures** (compare Fig 5 in the main paper). **Left:** *fraction oracle*. **Right:** *fraction of variance of the expected response*. Both measures qualitative show the same results.

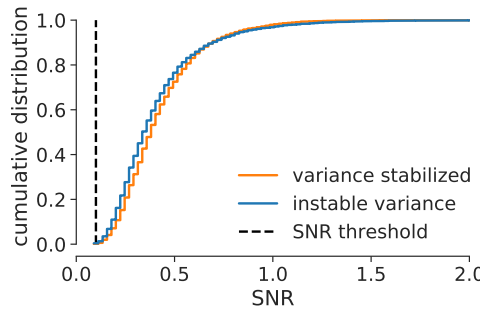


Figure 3: **Signal to noise ratio (SNR)**. The SNR distribution across neurons of the evaluation dataset (blue dataset in Fig. 1 in the paper) both with (orange) and without (blue) variance stabilizing transform (Anscombe). The neurons do not fall below the threshold of 0.1 (black), justifying the use of the performance statistic r_{ER}^2 for our dataset (see Fig. 14 in Pospisil & Bair (2020)).

4 INFLUENCE OF SEEDS

The performance scores reported in our study are subject to three different sources of statistical uncertainty: The random initialization of the model weights, the specific set of images used to train the model and the specific set of neurons that we wanted to predict. In order to get an estimate of how much each of these factors contribute to the variance in the performance of our models, we trained a total of 90 models, 30 for each source of uncertainty, and varied the respective seeds. While the seed of one source was altered, the seeds of the remaining two sources were kept fixed. Since the impact of the neuron and image seed naturally increases with decreasing amounts of data, we conducted this experiment on a medium range data regime of 1000 images and 1000 neurons. The results can be seen in Fig. 4. While the main contributions to the variance in model performance seem to stem from the model initialization and the random subset of neurons, the image seed did not seem to have a major influence. We thus only used a single value for it in most experiments in the paper. Since we do not consider the variance caused by the random initialization of the model weights as relevant for the underlying scientific problem, we decided to pick the models which performed best on the validation set across 5 model initialization seeds. Finally, we computed 95% confidence intervals across 5 seeds of random neuron subsets. In the cases where all available neurons were used in an experiment, the statistics were computed across 5 image seeds instead (see section Data in the paper). The total number of trained models per data point was thus always 25.

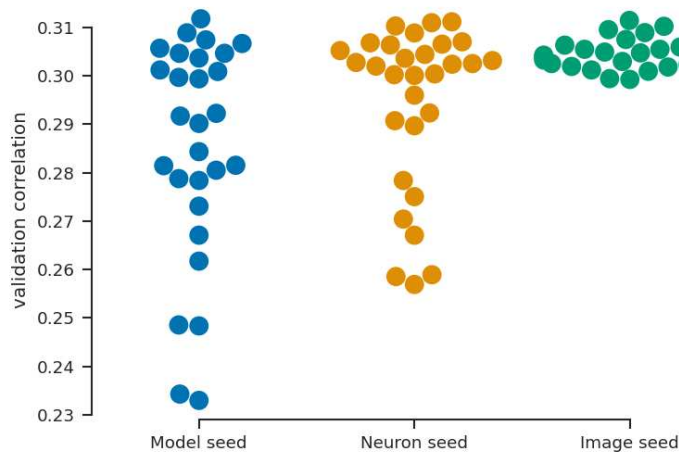


Figure 4: **Variation of model performance across seeds.** Several models (with Gaussian readout) were trained on 1000 neurons and 1000 images each, while varying the initialization of the model (blue), the specific set of 1000 neurons (yellow) and the specific set of 1000 images (green). Since the image seed did not have a major influence on model performance, we decided to only use a single seed value, select the best performing models across 5 model seeds and compute statistics across 5 neuron seeds throughout most of our experiments. In the cases where all available neurons were used in an experiment, the statistics were computed across 5 image seeds instead (see Chapter Data in the paper).

5 INFLUENCE OF CORTICAL DATA AND FEATURE SHARING ON THE GAUSSIAN READOUT

The models with Gaussian readout outperformed the ones with factorized readout, both in direct and transfer learning (see Fig. 3 and 4 in the paper). Here, we investigate which of its components this can be attributed to. To this end we trained models with Gaussian readout directly on the four matched datasets with 3625 neurons and varying number of images (Fig. 5, compare also Fig. 3 in the paper). We did this with and without using the components *feature sharing* and *cortex-data*: In the *feature sharing* condition, each neuron shared the same feature weight vector with its anatomical matches across the four datasets. The models with the *cortex-data* condition predicted the receptive field positions from anatomical cortical data via an affine transform. Both, *feature sharing* and *cortex-data* were switched on throughout the paper, and contributed to the good performance of the Gaussian readout. The better performance of the Gaussian readout compared to the factorized readout in Fig. 3 in the paper seems to be mainly due to *feature sharing*. The usage of cortical data to learn the receptive field positions is primarily advantageous for mid-range number of images.

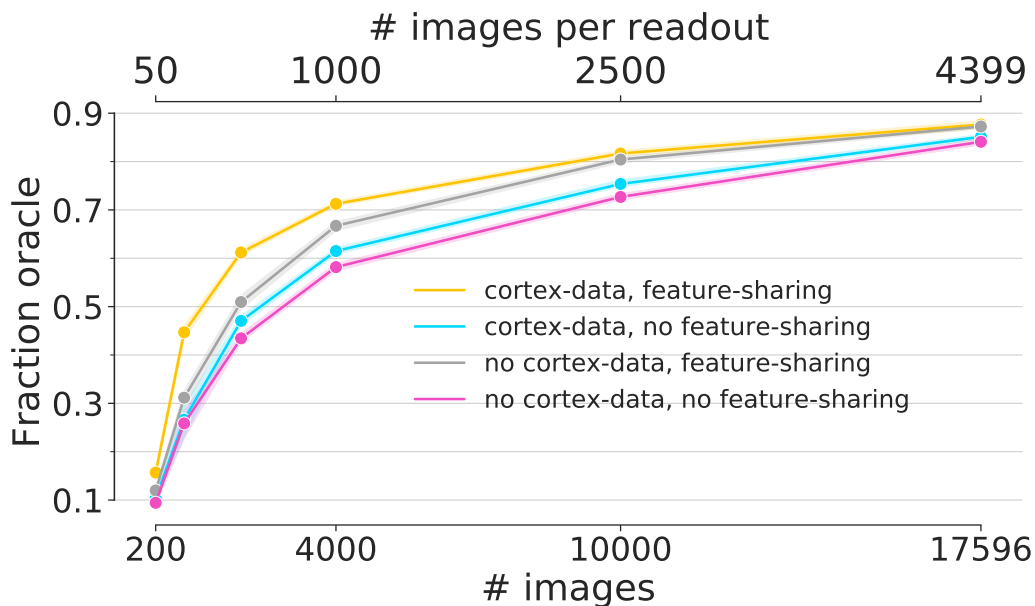


Figure 5: **Gaussian readout with and without feature sharing and position learning from anatomical data.** The training procedure is the same as in Fig. 3 in the paper. The *feature sharing* seems to be the main contributor to the better performance of the Gaussian readout compared to the factorized readout in Fig. 3 in the paper. The usage of cortical data to learn the receptive field positions seems to be primarily advantageous for low and mid range number of images.

6 MOST EXCITING INPUTS FOR MODELED NEURONS

One important application of general system identification models is the analysis of neural tuning, the relation that connects a neuron’s response to the stimulus. Describing neural response properties by the stimuli that drive them best has a long tradition in neuroscience (such as Gabor filters and gratings in early visual cortex, or face-selective cells in higher layers). Walker et al. (2019) and Bashivan et al. (2019) introduced a method to obtain such most exciting inputs (MEIs) which we analogously generated for our model with the 11-S transfer core (Fig. 5 in the paper, orange line). Like Walker et al. (2019) we use an ensemble of networks to generate the MEI. In our case, we used an ensemble of five 11-S transfer cores from five seed initializations for which we each trained a readout with the evaluation dataset on top. In Fig. 6 we show these MEIs for the 50 neurons with the best test performance. Walker et al. (2019) have shown that many MEIs differ quite strongly from Gabor-like stimuli, which would be expected to be the best drivers for V1 neurons based on previous work in monkeys and cats. Our MEIs exhibit very similar characteristics to those presented by Walker et al. (2019), which were obtained from a directly trained network and experimentally verified, highlighting the generality of our transfer core.

REFERENCES

- P. Bashivan, K. Kar, and J. DiCarlo. Neural Population Control via Deep ANN Image Synthesis. pp. 1–33, 2019. doi: 10.32470/ccn.2018.1222-0.
- Dean A Pospisil and Wyeth Bair. The unbiased estimation of the fraction of variance explained by a model. *bioRxiv*, pp. 2020.10.30.361253, nov 2020. doi: 10.1101/2020.10.30.361253. URL <https://doi.org/10.1101/2020.10.30.361253>.
- E. Y. Walker, F. H. Sinz, E. Cobos, T. Muhammad, E. Froudarakis, P. G. Fahey, A. S. Ecker, J. Reimer, X. Pitkow, and A. S. Tolias. Inception loops discover what excites neurons most using deep predictive models. *Nature Neuroscience*, 2019. ISSN 15461726. doi: 10.1038/s41593-019-0517-x. URL <http://dx.doi.org/10.1038/s41593-019-0517-x>.

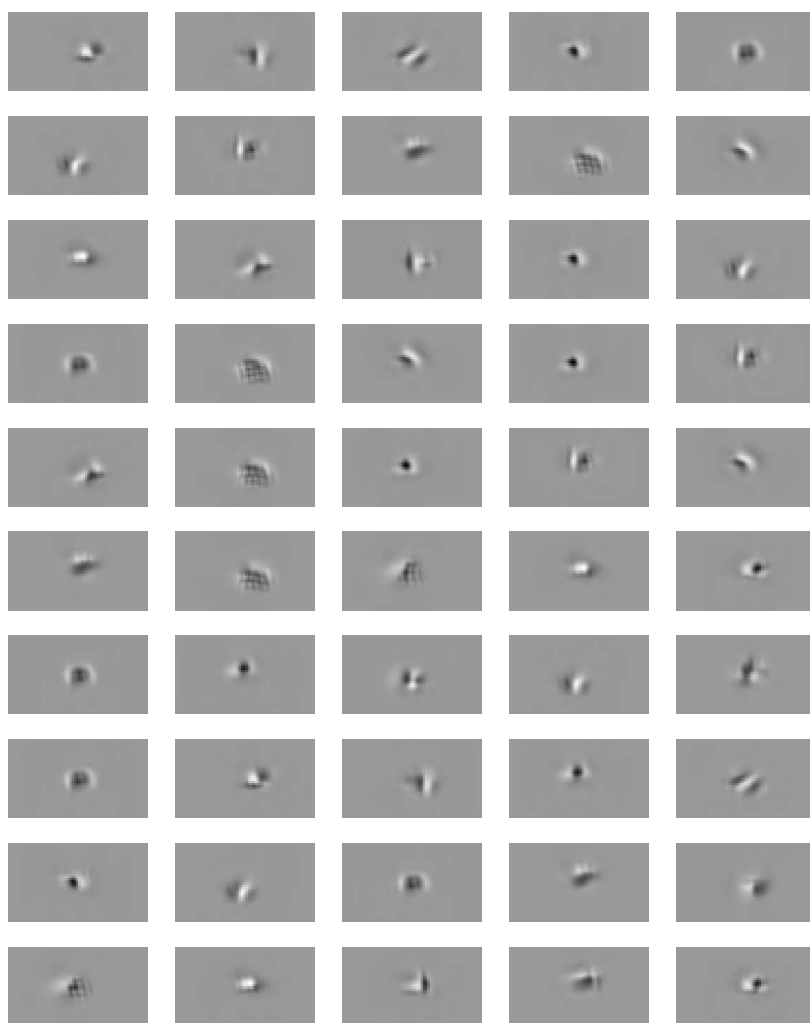


Figure 6: **Most exciting inputs (MEIs).** The image inputs that best drive the 50 best predicted neurons from the evaluation dataset, predicted with the best transfer core (Fig. 5 in the paper, orange line).

BAYESIAN ORACLE FOR BOUNDING INFORMATION GAIN IN NEURAL ENCODING MODELS

Konstantin-Klemens Lurz^{1,*}, Mohammad Bashiri^{1,*}, Edgar Y. Walker², 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

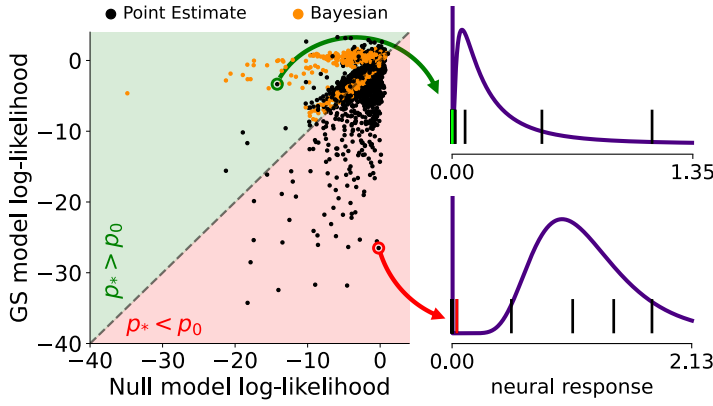


Figure 1: Comparison of lower and upper bound likelihood estimates (Null vs GS) per neuron. **Left:** For many neurons, the PE approach yields worse GS than the Null score. The Bayesian method results in the expected outcome of upper bound scores being higher than lower bound scores. **Right:** Two example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

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

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 $E[y_i|x] = \frac{1}{n-1} \sum_{y_j \in \mathbf{y}_{\setminus i}} y_j$ 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:

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

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.

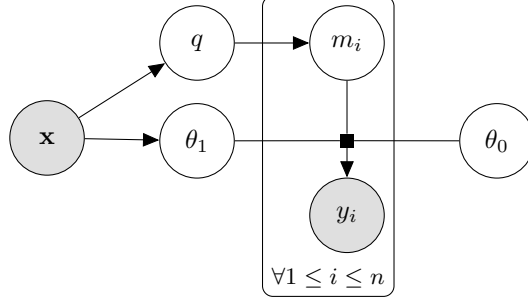
Efficient estimation for zero-inflated distributions In general, the integral in equation 2 is intractable and the posterior predictive distribution can only be evaluated using numerical approximations. Only for certain choices of likelihood the integral can be solved analytically if the right conjugate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

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

Figure 2: Graphical model for a zero-inflated distribution. A stimulus x determines the probability q whether a neuron 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*

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

where $\mathbf{y}_{\setminus i}^0$ and $\mathbf{y}_{\setminus i}^1$ 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. 3a). 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

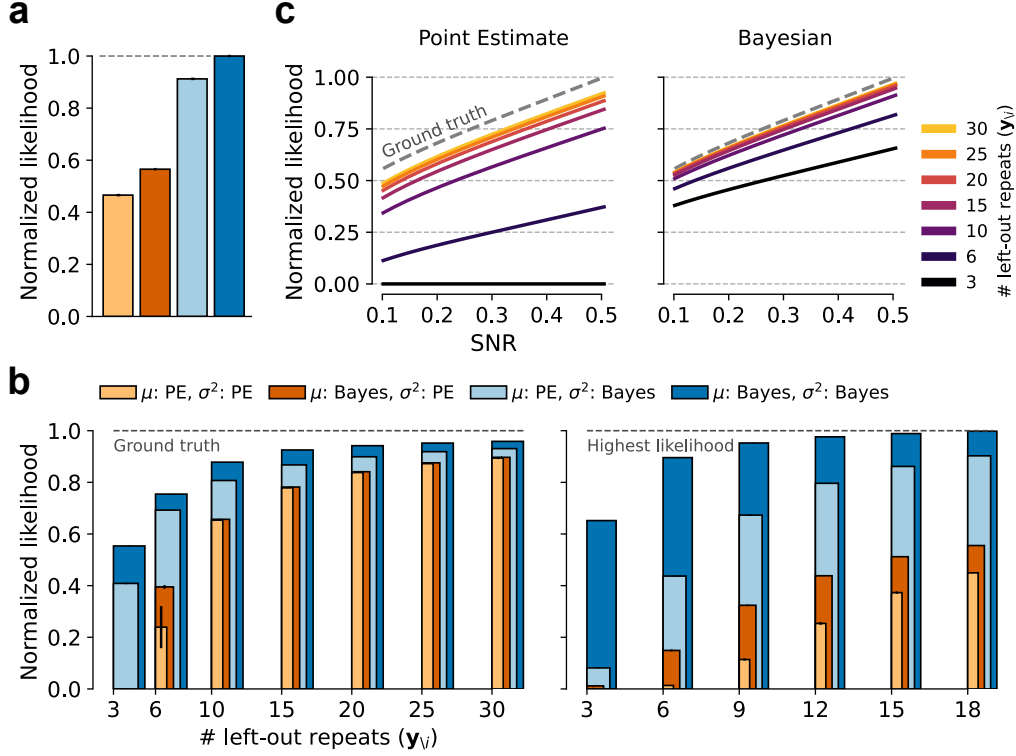


Figure 3: Comparison of the point estimate and Bayesian GS model. **a:** Comparison of different GS models where the individual parameters are either estimated via the PE or the Bayesian approach. The number of left-out repeats $y_{\setminus i}$ is 19. Colors are the same as in **b**. Normalized wrt. the max. likelihood value, i.e. the dark blue bar. **b:** Similar to **a** but for different numbers of left-out repeats $y_{\setminus i}$. **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 $y_{\setminus i}$. **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 $y_{\setminus i}$, in particular in the regime of low $n - 1$. The right panel of Fig. 3b shows this on neural data where we first observe that the results of Fig. 3a 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. 3b 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. 3c) and observed that the Bayesian approach consistently outperforms the PE approach across all SNRs (data for Fig. 3b left panel had an SNR of 0.42).

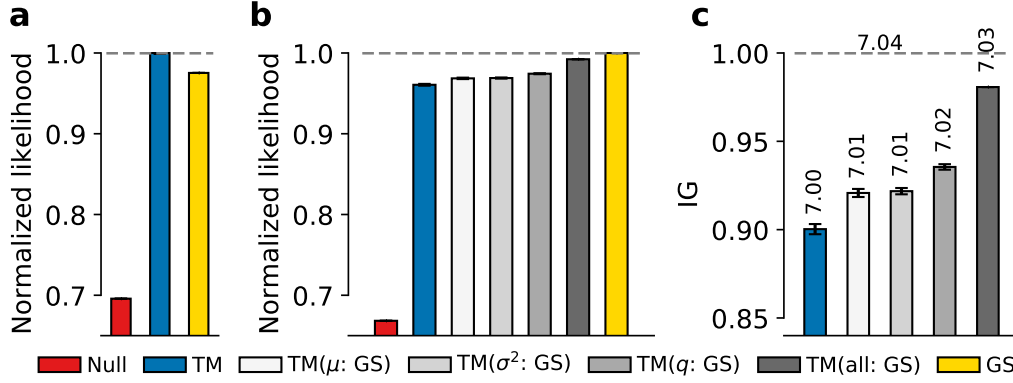


Figure 4: Real neural data evaluation of a neural encoding model trained on a zero-inflated Log-Normal likelihood (ZIL) using the Bayesian upper bound estimate (GS). **a:** A sub-optimal prior for the GS model results in a lower performance than the trained model (TM). Normalized wrt. the max. likelihood 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 $E_x[E_y[y|x]]$ and average conditional variance $E_x[\text{Var}_y[y|x]]$ 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):

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

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:

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

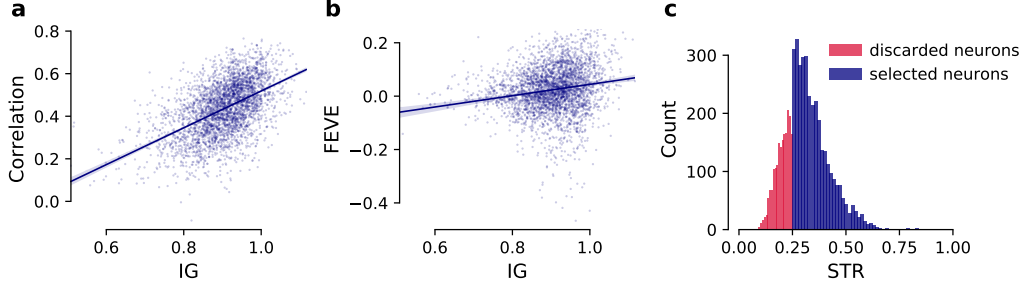


Figure 5: Per neuron comparison of NInGa with correlation and FEVE. The data comes from the TM in Fig. 4 (one of the 5 models from different initializations). **a:** Correlation vs. NInGa. **b:** FEVE vs. NInGa. **c:** The neurons displayed in panel **a** and **b** were selected based on whether their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

where $f(\hat{\sigma}_\epsilon) = \log\left(\frac{\hat{\sigma}_\epsilon}{\sigma_\epsilon}\right) - \frac{1}{2} + \frac{\sigma_s^2}{2\hat{\sigma}_\epsilon^2}$ 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. 5b, 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. 5c).

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 $\hat{\mu}_x$:

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

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. 5a) 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

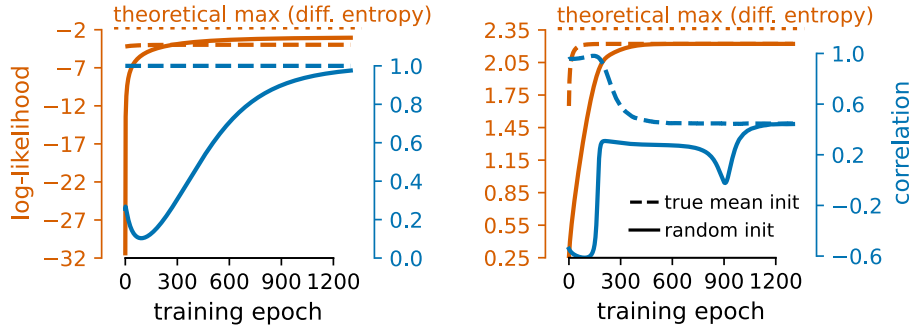


Figure 6: Comparison of log-likelihood and correlation of a model which is optimized via maximum likelihood estimation. **Left:** The model distribution is from the exponential family and has the mean as a sufficient statistic. The optima of the two metrics coincide but an improvement in likelihood does 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 $\text{NInGa} > 1$ (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 Kümmeler, 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, Andr e 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:

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

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:

$$y = \exp((z - \bar{z}) * c + \bar{z}),$$

where $\bar{z}$ 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):

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

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

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

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_{total}^2 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 $\mathbb{E}[\{a, b, \dots\}]$ is used to denote that the expectation involves the terms in the set $\{a, b, \dots\}$. The total variance can then be computed as

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

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

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

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_{LogN}^2 evaluate to

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

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_{LogN}^2 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:

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

Marginalizing over m :

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

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:

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

The posterior can be derived as follows:

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

where $\mathbf{y}_{\setminus i}^0$ are the zero responses, $\mathbf{y}_{\setminus i}^1$ 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:

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

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 $p(y_i|\mathbf{y}_{\setminus i})$:

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

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 $y_i < \tau$:

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

And if $y_i \geq \tau$:

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

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 from 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

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

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.

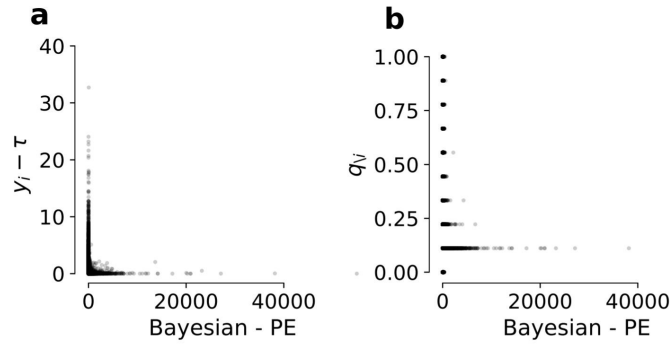


Fig. S1: Comparison between Bayesian and Point Estimate (PE) Gold Standard models. Since the two GS models share the same zero distribution, this analysis was only performed on the responses that fall into the positive distribution ($y \geq \tau$). **a:** Distance of the positive response from the zero-threshold τ as a function of the difference between Bayesian and the PE GS models. **b:** Fraction q_i of positive leave-one-out responses y_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:

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

F RELATION BETWEEN NORMALIZED INFORMATION GAIN AND FEVE

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

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

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. 5b):

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

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:

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

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:

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

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:

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

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:

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

Putting this back into Eq. 13:

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

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

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

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 $p(q) = \text{Beta}(q; \alpha'', \beta'')$:

$$\begin{aligned}\hat{q}_{MAP} &= \arg \max_q \log (q^{n_1} (1 - q)^{n_0} p(q)) \\ &= \arg \max_q \log \left(q^{n_1} (1 - q)^{n_0} \frac{q^{\alpha''-1} (1 - q)^{\beta''-1}}{B(\alpha'', \beta'')} \right) \\ &= \arg \max_q \underbrace{\log \left(q^{n_1 + \alpha'' - 1} (1 - q)^{n_0 + \beta'' - 1} \right)}_{:= f(q)} \\ \frac{\partial f(q)}{\partial q} &= \frac{\partial}{\partial q} \left[\log \left(q^{n_1 + \alpha'' - 1} (1 - q)^{n_0 + \beta'' - 1} \right) \right] \\ &= \frac{\partial}{\partial q} \left[(n_1 + \alpha'' - 1) \log(q) + (n_0 + \beta'' - 1) \log(1 - q) \right] \\ &= (n_1 + \alpha'' - 1) \frac{1}{q} - (n_0 + \beta'' - 1) \frac{1}{1 - q} \\ &\stackrel{!}{=} 0 \\ \hat{q}_{MAP} &= \frac{n_1 + \alpha'' - 1}{n_0 + n_1 + \alpha'' + \beta'' - 2}\end{aligned}$$

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

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

with

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

where $\bar{y} := 1/n_1 \sum_{y_j \in y_{\setminus i}^1} y_j$

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

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

And for σ^2 :

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

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.

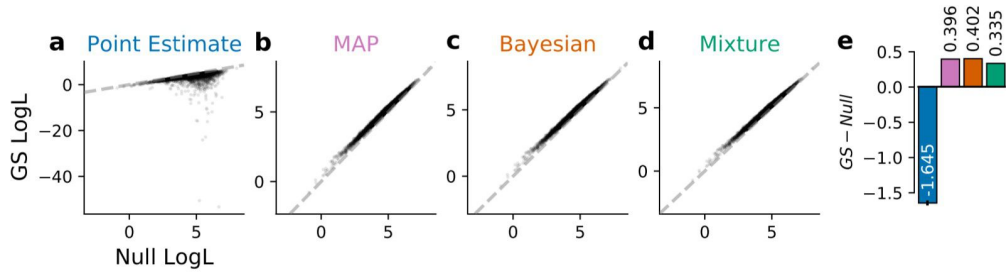


Fig. S2: Comparison of different methods to obtain an upper bound (GS): **a–d**: Various GS model log-likelihood vs. the Null model log-likelihood. Data is per neuron (averaged over repeats and stimuli). **e**: The full Bayesian Posterior Predictive outperforms the Point Estimate, the Maximum a posteriori (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.

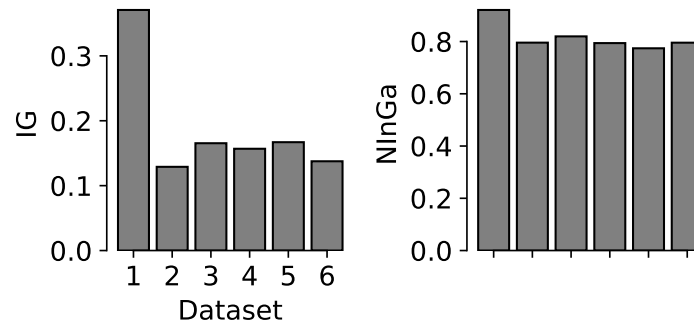


Fig. S3: Comparison of Trained Model performance on different datasets. **Left:** Models evaluated on simple Information Gain (IG), i.e. the numerator of Eq. 1. **Right:** Models evaluated on Normalized Information Gain (NInGa), i.e. the full Eq. 1.