Bayesian Oracle for bounding information gain in neural encoding models

Konstantin-Klemens Lurz,[1,*] Mohammad Bashiri,[1,*] Fabian H. Sinz[2,1,†]
[1] Institute for Bioinformatics and Medical Informatics, University of Tübingen, Germany
[2] Department of Computer Science, University Göttingen, Germany

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

Abstract

Many normative theories that link neural population activity to cognitive tasks, such as neural sampling and the Bayesian brain hypothesis, make predictions for single trial fluctuations. Linking information theoretic principles of cognition to neural activity thus requires models that accurately capture all moments of the response distribution. 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 *Information Gain* (IG) which evaluates the likelihood of a model relative to a lower and upper bound. However, while a lower bound is usually easy to obtain and evaluate, 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 jackknife oracle estimator for the mean – commonly used for correlation metrics – to a flexible Bayesian oracle estimator for IG 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. Finally, we provide the derivation of the upper bound estimator for a variety of common distributions including the state-of-the-art zero-inflated mixture models.

1 Introduction

Information theoretic approaches to perception and cognition have a long and successful history linking first principles to neural response properties [1]. For instance, neural sampling [2; 3] links perceptual decision making to posterior inference and makes quantitative predictions about noise correlations, i.e. the joint fluctuations around the mean activity in a trial [4; 5; 6]. By their nature, mutual information and entropy, the building blocks of information theory, are sensitive to the entire distribution of the participating variables, capturing all moments. Linking information theoretic principles for cognition to neural activity thus requires models that accurately capture the entire response distribution. However, despite great success in modeling neural population activity in the past decades, most encoding models currently focus on estimating the trial-averaged mean activity [7; 8; 9; 10; 11; 12; 13; 14]. If we want to use neural encoding models as a quantitative underpinning for our theories, we need them to predict and be evaluated on complete response distributions.

To evaluate mean-predicting models, correlation-based metrics are often used [12; 15]. Correlation is an interpretable measure since it is naturally bounded between -1 and 1. However, for vanilla

Information-Theoretic Principles in Cognitive Systems Workshop at the 36th Conference on Neural Information Processing Systems (NeurIPS 2022).

correlation, it is impossible for any model to achieve a correlation of 1 in the presence of trial-to-trial fluctuations. Therefore, model correlation is often normalized by an upper bound oracle estimator [16; 12], 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. To this end, one can use Information Gain (IG) [17], which uses an estimate of both upper and lower bound, to put the likelihood between two meaningful values.

In this work, we show how to estimate such lower and upper bounds for IG 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 to generalize the point estimate approach to the rigorous Bayesian framework of posterior predictive distributions. Our approach yields lower and upper bounds which are robust to all the above-mentioned complexities in neural data. Finally, we provide the derivation of the upper bound estimator for a variety of common distributions including the state-of-the-art zero-inflated mixture models [18; 19].

2 Information Gain and Bayesian Gold Standard Models

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 Information Gain (IG) [17; 20] which sets the model likelihood on an interpretable scale between an estimated lower and upper bound (Eq. 1): a Null distribution $p_0(y)$ and a Gold Standard distribution $p_*(y|x)$. This method of computing IG can be interpreted as a normalized comparison of lower bounds of mutual information.

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

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: $p_0(y|x) := \hat{p}(y)$. The *Gold Standard (GS) model* $p_*(y|x)$, on the other hand, should be the best possible approximation of the true conditional distribution $p(y|x)$. We estimate its parameters from responses 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 $\backslash i$. Given such a setting, the challenge is how to obtain a robust GS model which we address in the following sections.

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:

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

Often, moment matching $\theta_i = f(\mathbf{y}_{\backslash i}) = f(E[\mathbf{y}_{\backslash i}|x], Var[\mathbf{y}_{\backslash i}|x])$ is used to obtain the point estimate of θ. $E[y_i|x] = \frac{1}{n-1}\sum_{y_j \in y_{\backslash i}} y_j$ is typically used as an oracle predictor for the conditional mean in correlation-based measures to obtain an upper bound on the achievable performance in the presence of noise [16; 12].

Problems with the PE approach To demonstrate the problems with point estimate GS models, we modeled neural responses with a Zero-Inflated LogNormal likelihood and estimated the upper bound using the PE approach (see Appendix A.1 for details on data, [18; 19] for details on zero-inflated distributions, and Appendix C 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).

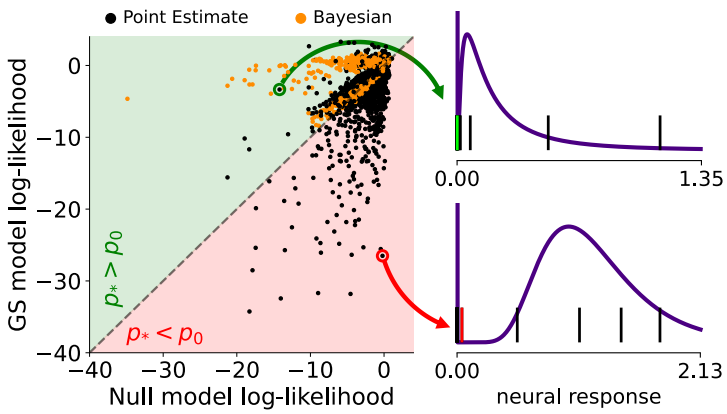

Figure 1: Comparison of lower and upper bound likelihood estimates (Null vs GS) per neuron. **Left:** For many neurons, the PE approach yields worse GS than the Null score. The Bayesian method results in the expected outcome of upper bound scores being higher than lower bound scores. **Right:** Two example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

Bayesian oracle to the rescue To avoid an overconfident GS model, we add uncertainty to the parameter estimation of the point estimate. We achieve this by a full Bayesian treatment of the GS parameters and inference via the full posterior predictive distribution:

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

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. In general, this integral is intractable and the posterior predictive 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. In Appendix D, we analytically derive the posterior predictive for any choice of zero-inflated likelihood where the posterior predictive of the non-zero part is known. Zero-inflated likelihoods are the basis of current state-of-the-art neural encoding models [18; 19] and, to the best of our knowledge, an analytical solution for their posterior predictive has been missing. Our derivation is general and can be used for distributions such as the Zero-Inflated Gamma [18], Zero-Inflated LogNormal or even Zero-Inflated Flow models [19]. Our final solution only involves a single one-dimensional integral on the range $[0, 1]$ which can be solved efficiently using numerical integration.

We choose the parameters of the prior $p(\theta)$ using the responses of all neurons across all stimuli. Applying the Bayesian approach to neural data, we observe that the GS model is more robust against outliers and yields higher likelihoods than the null model, as expected (Fig. 1, orange points).

3 Analysis

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 thousands of neurons to hundreds of stimuli, each repeated 20 times (see Appendix A.1 for details on neural data) as well as simulated data (see Appendix A.2 for details on simulated data). The model we used is the Zero-Inflated LogNormal distribution (Appendix D).

Bayesian GS estimates higher order moments better. In order to determine for which parameters the likelihood profits the most from the probabilistic estimation we compared GS models where the individual parameters are either estimated via the PE or the Bayesian approach (Fig. 2**a**). For this we used the largest number of left-out repeats possible $n - 1 = 19$. 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, IG ought to yield consistent estimates for

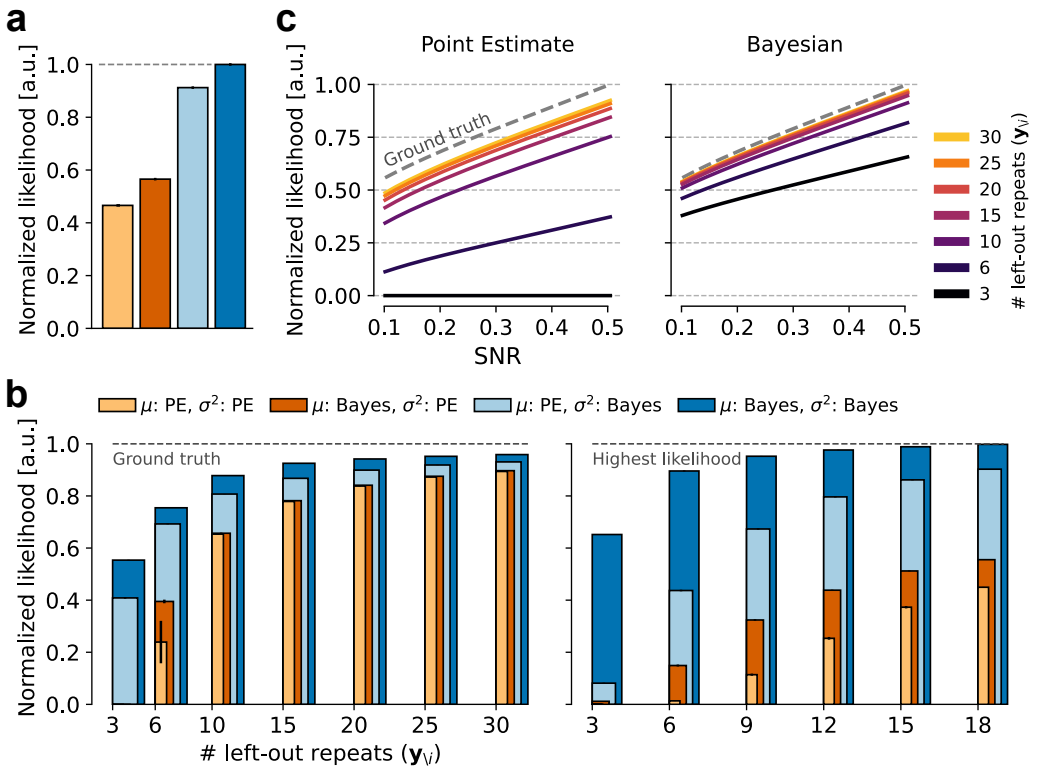

Figure 2: Comparison of the point estimate and Bayesian GS model. **a:** Comparison of different GS models where the individual parameters are either estimated via the PE or the Bayesian approach. The number of left-out repeats $\mathbf{y}_{\setminus i}$ is 19. Colors are the same as in **b**. Normalized wrt. the max. likelihood value, i.e. the dark blue bar. **b:** Similar to **a** but for different numbers of left-out repeats $\mathbf{y}_{\setminus i}$. **Left:** Simulated data. Normalized wrt. the ground truth likelihood. **Right:** Neural responses. Normalized wrt. the max. likelihood value, i.e. the dark blue bar at 18 left-out repeats. **c:** Upper bound likelihood scores for different signal-to-noise ratios and different number of left-out repeats $\mathbf{y}_{\setminus i}$. **Left:** Point Estimate. **Right:** Bayesian. Normalized wrt. the ground truth likelihood. In all panels the likelihood values are averaged over stimuli and neurons, and the error bars and shaded areas show SEM over 5 random selections of the left-out repeats.

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

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

References

[1] Eero P Simoncelli and Bruno A Olshausen. Natural image statistics and neural representation. *Annu. Rev. Neurosci.*, 24(1):1193–1216, November 2003.

[2] 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.

[3] 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.

[4] 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.

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

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

A Data

A.1 Neural data

Data from real neural activity is used in Fig. 1 and in Fig. 2 panel **a** and panel **b** on the right side. 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 [12]. The data 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.

A.2 Simulated neural data

Simulated neural data is used in Fig. 2 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 LogNormal distribution where the parameters μ and σ^2 of the LogNormal part are normal-gamma distributed. The complete description of the simulation is as follows:

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

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:

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

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

B Zero-inflated likelihood

We assume that the neural responses $\mathbf{y}$ come from a zero-inflated distribution [18] corresponding to the graphical model in Fig. 3. Such a distribution is a mixture of two distributions which do not overlap and which are separated at the zero-threshold τ.
In the experiments of this work we choose $\tau = \exp(-10)$ and assume a uniform distribution $U(0, \tau)$ for the zero part with a LogNormal distribution for the non-zero part. However, the following derivations are kept general and are valid for any zero-inflated distribution.

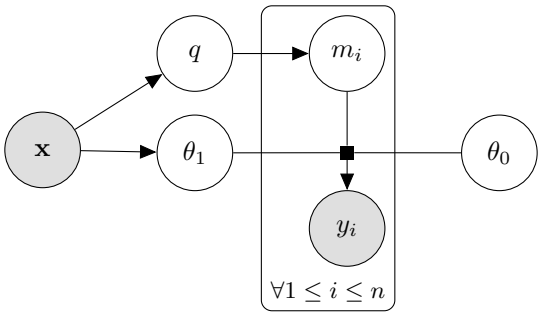

Figure 3: Graphical model for a zero-inflated distribution. A stimulus $\mathbf{x}$ determines the probability q whether a neurons fires or not and the parameters θ_1 of the response distribution of the non-zero response distribution. A Bernoulli random variable m_i determines whether a neuron fires on a particular trial i. If $m_i = 1$ a response is drawn from $p(y_i|\theta_1)$, otherwise from $p(y_i|\theta_0)$.

The probability density for the graphical model in Fig. 3 is defined as:

$$p(y) = (1-q) \cdot \underbrace{p(y|\theta_0)}_{\text{uniform}} + q \cdot \underbrace{p(y|\theta_1,\ loc = \tau)}_{\text{positive distribution}}$$

Note that the distribution of the non-zero part is shifted by the zero-threshold τ. For simplicity of notation, we will omit the loc parameter in the following derivations.

C Moment matching for zero-inflated likelihood

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

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

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

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

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

and

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

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

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

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

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

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

C.1 Zero-inflated Log-Normal likelihood

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

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

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

D 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 $p(y_i|\mathbf{y}_{\setminus i})$. Following the graphical model in Fig. 3, let's define some of the density functions that will be used later on:

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

Marginalizing over m:

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

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

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

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

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

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

The posterior can be derived as follows:

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

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

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

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

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

$$
\begin{aligned}
p(y_i|\mathbf{y}_{\setminus i}) &= \int_{\theta,q} p(y_i|\theta, q)p(\theta, q|\mathbf{y}_{\setminus i})\, d\theta dq \\
&= \int_{\theta,q} p(y_i|\theta, q)p(\theta_0|\mathbf{y}^0_{\setminus i})p(\theta_1|\mathbf{y}^1_{\setminus i})p(q|\mathbf{y}_{\setminus i})\, d\theta dq \\
&= \int_q \underbrace{\left(\int_\theta p(y_i|\theta, q)p(\theta_0|\mathbf{y}^0_{\setminus i})p(\theta_1|\mathbf{y}^1_{\setminus i})\, d\theta\right)}_{=p(y_i|q,\mathbf{y}_{\setminus i})} p(q|\mathbf{y}_{\setminus i})\, dq \\
&= \int_q p(y_i|q, \mathbf{y}_{\setminus i})p(q|\mathbf{y}_{\setminus i})\, dq
\end{aligned} \tag{2}
$$

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

If $y_i < \tau$:

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

And if $y_i \geq \tau$:

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

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

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

D.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:

$$
\begin{aligned}
p(y_i|\mathbf{y}_{\setminus i}) &= p(\log(y_i)|\log(\mathbf{y}_{\setminus i})) \cdot |\det \nabla_{y_i} \exp(y_i)| \\
&= p(\log(y_i)|\log(\mathbf{y}_{\setminus i})) \cdot \frac{1}{y_i}
\end{aligned} \tag{3}
$$

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

$$
p(z_i|z_{\setminus i}) = t_{2\alpha'} \left(z_i|\mu', \frac{\beta'(\nu'+1)}{\nu'+\alpha'} \right), \tag{4}
$$

where

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

with n being the number of left-out repeats $z_{\setminus i}$ and $\bar{z}_{\setminus i}$ being the mean of the left-out repeats. As the final step, to compute the posterior predictive in the original $\log$ space, we plug Eq. 4 back into Eq. 3:

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