# Neural Likelihood

**Christoph Blessing**[*,1], **Edgar Y. Walker**[*,1−3], **Katrina R. Quinn**[4], **R. James Cotton**[5]
**Wei Ji Ma**[6], **Andreas S. Tolias**[2,3,7], **Hendrikje Nienborg**[4], **Fabian H. Sinz**[1−3,8] **(fabian.sinz@uni-tuebingen.de)**
[1]Institute for Bioinformatics and Medical Informatics, University Tübingen, Germany
[2]Center for Neuroscience and Artificial Intelligence, Baylor College of Medicine, Houston, Texas
[3]Department of Neuroscience, Baylor College of Medicine, Houston, Texas
[4]Centre for Integrative Neuroscience, University Tübingen, Germany
[5]Shirley Ryan Ability Lab, IL, USA
[6]Center for Neural Science and Department of Psychology, New York University, NY, USA
[7]Department of Electrical and Computer Engineering, Rice University, TX, USA.
[8]Bernstein Center for Computational Neuroscience, University Tübingen, Germany
[*] equal contribution

## Abstract

**A large body of evidence shows that perceptual decision making in humans and animals accounts for uncertainty in the relevant stimulus variable. This suggests that the decision is based on a distribution over stimuli given the neuronal activity rather than single point estimates. The likelihood over the stimuli captures this uncertainty for a fixed neuronal response. Because the neuronal population response can be high dimensional, estimating a per-trial likelihood can be challenging. Previous work has thus focused on parametric models, which can introduce a bias by ignoring noise correlations. Here, we present a simple yet general method to decode a per-trial likelihood based on neural networks. Our method applies to discrete and continuous, as well as static and time-series data. We demonstrate it on recordings from two experimental visual paradigms in macaque V1 and V2.**

**Keywords:** likelihood decoding; visual cortex; neuronal networks; flow models; non-human primates

## Introduction

Bayesian models of behavior have been widely successful in explaining decision-making under various tasks in both human and monkeys, frequently demonstrating that animals make nearly Bayes optimal decisions (Ma & Jazayeri, 2014). This suggests that the underlying computations are based on representations of probability distributions rather than their individual moments, such as the mean or the maximum (Pouget, Dayan, & Zemel, 2003). Representations of probability arise naturally in noisy systems, such as the brain. If one stimulus $s$ can yield several possible neural responses $\mathbf{r}$, then each $\mathbf{r}$ is naturally associated with a posterior $p(s|\mathbf{r})$ or a likelihood $p(\mathbf{r}|s) \equiv \mathcal{L}_{\mathbf{r}}(s)$. While the posterior also accounts for prior expectations about $s$, the likelihood captures the information the population response carries about the stimulus. It simultaneously encodes the best estimate of the stimulus (the peak of $\mathcal{L}_{\mathbf{r}}(s)$) as well as the associated uncertainty (e.g. the width of $\mathcal{L}_{\mathbf{r}}(s)$) (Ma, Beck, Latham, & Pouget, 2006).

Because $\mathbf{r}$ is often high dimensional, estimating a trial-by-trial likelihood function from recordings of cortical population responses is challenging. Existing methods typically make strong parametric assumptions about the stimulus conditioned distribution of the population response $p(\mathbf{r}|s)$, such as an independent Poisson (Graf, Kohn, Jazayeri, & Movshon, 2011) or Poisson-like distribution (Ma et al., 2006). While these assumptions considerably simplify computing the trial-by-trial likelihood function, they can also introduce biases by ignoring potential noise correlations between different neurons (see Walker, Cotton, Ma, and Tolias (2018) Supplementary Figure 3) or subsequent time points, or internal brain state fluctuations (see Denfield, Ecker, Shinn, Bethge, and Tolias (2018); Ecker et al. (2014) for examples of conditional dependencies).

Here we present a simple yet general approach to estimate the trial-by-trial likelihood function that makes no parametric assumption about the stimulus conditioned distribution $p(\mathbf{r}|s)$ and only requires knowledge of the stimulus prior $p(s)$, which is known to the experimenter in most experimental paradigms. Our approach is completely generic and can be applied to static or dynamic responses, as well as discrete or continuous data. It relies on two key steps:

1. Instead of estimating the likelihood $p(\mathbf{r}|s)$ explicitly, which is infeasible in most cases, we estimate it implicitly by recovering an unnormalized likelihood from a model $q(s|\mathbf{r},\theta)$ of the posterior using the known prior. $\theta$ denotes the parameters of the posterior model.

2. We estimate the posterior $q(s|\mathbf{r},\theta)$ using a flexible probabilistic model such as a neuronal network.

Because our method relies on estimating the posterior directly, it will work particularly well in situations when $\dim(s) \ll \dim(\mathbf{r})$. Estimating a conditional distribution over the lower dimensional $s$ instead of a the high dimensional $\mathbf{r}$ substantially simplifies the problem and allows us to use more flexible models such as neural networks. While algorithm yields a per-trial likelihood $\mathcal{L}_{\mathbf{r}}(s)$ for each $\mathbf{r}$, it trades the ability to provide a distribution over $\mathbf{r}$ for a given $s$ for a simplified estimation problem.

Here, we present the mathematical foundation as well as a continuous and recurrent version of the algorithm. We recently applied a discrete static version of this method to decode likelihood function from a population of V1 neurons responding to orientation stimulus in monkeys (Walker et al., 2018).

## Derivation of the Algorithm

**Implicit estimation of the likelihood** Given data $\{s_i, \mathbf{r}_i\}_{i=1}^m$, we train a flexible model $q(s|\mathbf{r},\theta)$ of the posterior by minimizing the objective

$$\sum_{i=1}^{m} -\log q(s_i|\mathbf{r}_i,\theta) \tag{1}$$

with respect to $\theta$. Equation (1) is the Monte-Carlo estimate of the expected Kullback-Leibler (KL) divergence

$$\ell(\theta) = \langle D\left[p\left(s|\mathbf{R}\right)\|q\left(s|\mathbf{R},\theta\right)\right]\rangle_{\mathbf{R}}$$

between the true posterior and the model. Because the KL-divergence is a strictly non-negative quantity, $\ell(\theta) = 0$ if and only if $D\left[p\left(s|\mathbf{r}\right)\|q\left(s|\mathbf{r},\theta\right)\right] = 0$ for each $\mathbf{r}$ except for sets of measure zero. The minimum is attained at $p\left(s|\mathbf{r}\right) = q\left(s|\mathbf{r},\theta\right)$ which yields

$$p\left(\mathbf{r}|s\right) \propto \frac{q\left(s|\mathbf{r},\theta\right)}{p\left(s\right)} \tag{2}$$

with a proportionality constant that depends on $\mathbf{r}$, but not on $s$. Again, note that the simplified estimation comes at the expense of the normalization constant, i.e. the method does provide an estimate of likelihood $\mathcal{L}_{\mathbf{r}}(s)$ not the conditional distribution of $\mathbf{r}$ given $s$.

**Flexible models for finite discrete posteriors** We use neural networks to get flexible models for $q(s|\mathbf{r},\theta)$. In the following, we describe the finite discrete and the continuous case.

In the discrete case, the likelihood function, the posterior, and the prior for each response vector $\mathbf{r}$ are represented by vectors $\mathbf{L},\mathbf{p}_s,\mathbf{q} \in \mathbb{R}^n$ where $\mathbf{L} = \mathcal{L}_{\mathbf{r}}(s)$ and $n$ is the number of different states $s$ can be in. For instance, $n = 2$ in a task where trials can have two possible outcomes. We use a deep neural network $f$ (DNN) (Goodfellow, Ian, Bengio, Yoshua, Courville, 2016) to directly predict $\mathbf{L} = f(\mathbf{r})$ from the population response vector $\mathbf{r}$. To get the objective function, we combine $\mathbf{L}$ and $\mathbf{p}_s$ to compute the log posterior over the stimulus $s$ up to some scalar value $b(\mathbf{r})$,

$$\mathbf{z} \equiv \log \mathbf{p}_s + \log \mathbf{L} = \log q(s|\mathbf{r},s) + b(\mathbf{r}). \tag{3}$$

and take a softmax $\mathbf{q} = \exp(\mathbf{z})/\sum_{j=1}^{n}\exp(z_j)$ to normalize it. The network $f$ is then trained to maximize the log posterior for the available data

$$\text{maximize}_{\theta}\ \frac{1}{m}\sum_{i=1}^{m}\delta(s_i)^{\top}\log\mathbf{q}\left(\mathbf{r}_i\right), \tag{4}$$

where $\delta(s_i)$ is a one-hot encoding of the stimulus $s_i$.

**Flexible models for continuous posteriors** To estimate a flexible posterior model for continuous stimulus variable, we use the idea of projection pursuit density estimation (Friedman, Stuetzle, & Schroeder, 1984) recently repopularized as *flow models* (Dinh, Sohl-Dickstein, & Bengio, 2017). A flow model transforms random variables $\xi$ from some easy to evaluate source density $p_{\xi}(\xi)$ into random variables $x = f(\xi)$ using an invertible and differentiable function $f$. The density of $x$ is then given by

$$\log p_x\left(x\right) = \log p_{\xi}\left(f^{-1}\left(x\right)\right) + \log\left|\det\frac{\partial f^{-1}}{\partial x}\right|. \tag{5}$$

The function $f$ is usually implemented by a deep neuronal network with special layers that are invertible and have an easy to compute log-determinant of the Jacobian. The full log-determinant is then simply the sum of the single log-determinants.

In our case we use a conditional flow model $s = f(\xi,\mathbf{r})$ that generates a density $q(s|\mathbf{r},\theta)$ for each population response vector $\mathbf{r}$. Note that $f$ needs to be invertible and differentiable in $s$ only. Because we can explicitly compute density values in a flow model, we can use equation (1) to find the parameters $\theta$. After $q(s|\mathbf{r},\theta)$ has been trained, we use equation (2) to get the likelihood function.

## Experiments

We tested the discrete and continuous stimulus likelihood decoders on two distinct sets of macaque population recording, each collected by different groups. All experimental protocols were approved by the local authorities (Regierungspräsidium Tübingen and Baylor College of Medicine Institutional Animal Care and Use Committee).

### Discrete stimulus task

Two male macaques performed a variant (Kawaguchi et al., 2018) of the disparity discrimination task described in Nienborg and Cumming (2009) while neuronal single and multi-unit activity in visual area V2 was recorded using linear multichannel recording electrodes (Plexon, Inc. V-probes, 24 channels). Monkeys had to classify trials into near and far, based on the predominantly occurring disparity in a random sequence of disparities. The signal strength on each trial was defined to be the proportion of stimulus frames that show the disparity from the class. The remaining frames of the trial were randomly picked from a set of predefined disparity values including the disparity value associated with the current class. Here, data from a single session (939 trials) in one animal was analyzed.

### Continuous stimulus task

We obtained V1 population responses from two macaques performing on orientation classification task as was presented in Walker et al. (2018). Up to 96 channels of multi-unit activities were recorded from each recording session, and the contrast of the orientation stimuli were varied on a trial by trial basis. There were total of 132 recording sessions, and for each session, trials with same contrast were grouped into a contrast-session. There were total of 546 contrast-sessions in the whole dataset, with a total of 303,326 trials.

| Symbol | Description | Possible Values |
|---|---|---|
| $N_h$ | size of hidden state | $\{1,2,3,4,5,6\}$ |
| $\lambda$ | learning rate | $\{0.0025, 0.005, 0.01, 0.02, 0.04\}$ |

Table 1: Possible values of hyperparameters during RNN model selection.

## Models

**Discrete case** We modeled the mapping $f(\mathbf{r})$ as a recurrent neural network (RNN) consisting of a single-layer gated recurrent unit (GRU) (Cho et al., 2014), which had a hidden state size of $N_h$, a learned initial hidden state and the same linear readout at each time step. The training to validation set split was set to 70%:30%. We trained the network on the training set with a fixed learning rate of $\lambda$ while monitoring its performance on the validation set for early stopping, which was carried out once the validation set loss failed to improve over 400 epochs. Upon training completion, the parameter set that produced the lowest loss on the validation set was restored. Using a random grid search over candidate hyperparameter values (Table 1), we found $N_h = 2$ and $\lambda = 0.02$ to be the combination that yielded the lowest loss on the validation set.

**Continuous case** For the continuous task, we modeled the mapping $\xi = f^{-1}(s, \mathbf{r})$ by a monotonously increasing function in $s$ using linear interpolation between 10 base points $g_i$ and predefined stimulus locations $s_i$. This choice is motivated by the fact that any mapping between two one-dimensional continuous distributions can be written as $(\mathcal{F}_2^{-1} \circ \mathcal{F}_1)(s)$ where $\mathcal{F}_k$ denote the cumulative distribution functions (cdfs). As both cdfs are monotonously increase, so must be their inverse and their composition. The base points $g_i$ were predicted from each $\mathbf{r}$ using a 3-layer fully-connected ResNet (He, Zhang, Ren, & Sun, 2016) with a final softmax layer followed by a cumulative sum and appropriate scaling.

As in Walker et al. (2018), separate instance of the model were fit for each contrast-session. Trials from a contrast-session were divided into training and validation sets based on a 80%:20% split. We trained the network on the training set with a fixed learning rate of $1e-4$ and monitored its loss on the validation set. Upon training completion, the parameter set that produced the lowest loss on the validation set over the course of training was restored.

## Results

Here we demonstrate that our approach on two real world datasets. Our intention is not to provide biological insight but to showcase that our method behaves reasonably given our current knowledge about the neural system. Because of limited number of trials, we report the behavior on the validation set used for early stopping. In a real experiment, the decoder would be used on all data (including training), just like a fitted tuning curve.

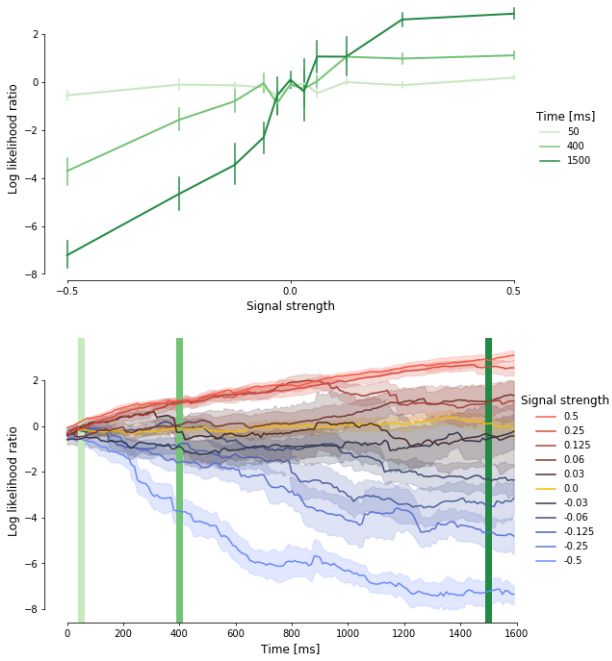


Figure 1: Mean log likelihood ratio on the validation set as a function of signal strength for different readout times (top) or as a function of readout time for different signal strengths (bottom). Positive signal strength corresponds to near stimulus class, and vice versa. Vertical lines in the bottom plot mark the time points where the sigmoidal curves in the top plot were extracted. The error bars and bands represent the standard error of the mean.

**Discrete case** We evaluate the resulting model by the log-likelihood ratio $\log \frac{p(\mathbf{r}|s=\text{near})}{p(\mathbf{r}|s=\text{far})}$ at different time points during the trial (Fig. 1, bottom). Since the prior distribution over the stimulus classes was uniform, this also reflects the log-posterior ratio. As expected, the evidence for a particular class increases over time (Fig. 1, top) and becomes more pronounced for higher signal strengths.

**Continuous case** We visually evaluate the model by plotting the likelihood over different stimulus orientations for different values of the contrast. For low contrast the uncertainty about the stimulus is higher so the likelihood curves become wider, as expected (Fig. 2).

## Summary

We developed a simple yet general deep-learning based method for decoding per-trial likelihood functions from population responses which makes substantially less assumptions about the form of the likelihood function. Our method thus deviates from the traditional approach where likelihood functions are computed by using a conditional distribution over the neural responses $\mathbf{r}$ given $s$. Estimating this distribution is

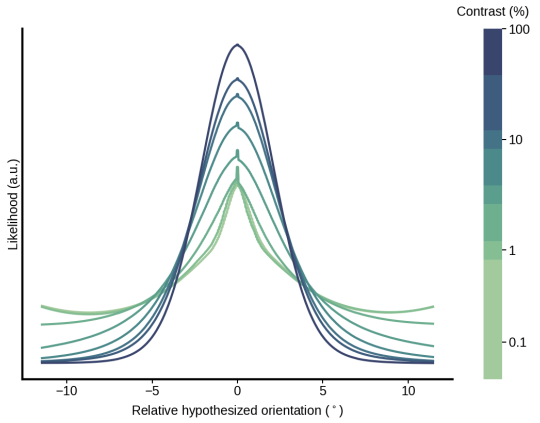


Figure 2: Average likelihood function decoded by the continuous decoder for each contrast. On each trial, the decoded likelihood function over the continuous stimulus orientation was shifted such that the peak of the normalized likelihood function occurred at $0^\circ$. The centered likelihood functions were then averaged across all trials within the same contrast bin.

very hard due to the *curse of dimensionality* (Nagler & Czado, 2016) requiring infeasibly large number of samples. While the use of strong parametric assumption on the distribution, such as conditional independence, can substantially decrease the sample size needed for the parameter estimation, this places heavy constraint on the distribution which is often only justified by computational feasibility. Our method avoids these issues by directly learning the likelihood function from the data and avoids modeling in the generative direction $s \to \mathbf{r}$. This means that our resulting approximation of the likelihood function $\hat{\mathcal{L}}_{\mathbf{r}}(s)$ does not yield a generative model of $\mathbf{r}$ given $s$. However, as long as our interest lies on the likelihood function $\mathcal{L}_{\mathbf{r}}(s)$ as a function of $s$, our method correctly approximates the true likelihood function $\mathcal{L}_{\mathbf{r}}(s)$ up to a multiplicative constant (or constant offset in log domain), and typical applications of likelihood functions are insensitive to such constant multiplication.

## Acknowledgments

Supported by the Institutional Strategy of the University of Tübingen (Deutsche Forschungsgemeinschaft, ZUK 63); the Carl-Zeiss-Stiftung; the DFG Cluster of Excellence Machine Learning New Perspectives for Science, EXC 2064/1, project number 390727645; CRC 1233 "Robust Vision" project number 276693517, FOR 1847 project NI1718/1-1; by National Science Foundation Grant IIS-1132009; NIH DP1 EY023176 Pioneer Grant; NIH R01 EY026927.

## References

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. Retrieved from http://arxiv.org/abs/1406.1078 doi: 10.3115/v1/D14-1179

Denfield, G. H., Ecker, A. S., Shinn, T. J., Bethge, M., & Tolias, A. S. (2018, dec). Attentional fluctuations induce shared variability in macaque primary visual cortex. *Nature Communications*, *9*(1), 2654. doi: 10.1038/s41467-018-05123-6

Dinh, L., Sohl-Dickstein, J., & Bengio, S. (2017). *Density Estimation using Real NVP* (Vol. 2; Tech. Rep.).

Ecker, A., Berens, P., Cotton, R., Subramaniyan, M., Denfield, G., Cadwell, C., ... Tolias, A. (2014, apr). State Dependence of Noise Correlations in Macaque Primary Visual Cortex. *Neuron*, *82*(1), 235–248. doi: 10.1016/J.NEURON.2014.02.006

Friedman, J. H., Stuetzle, W., & Schroeder, A. (1984, sep). Projection Pursuit Density Estimation. *Journal of the American Statistical Association*, *79*(387), 599–608.

Goodfellow, Ian, Bengio, Yoshua, Courville, A. (2016). Deep Learning. *MIT Press*. doi: 10.1142/S1793351X16500045

Graf, A. B. A., Kohn, A., Jazayeri, M., & Movshon, J. A. (2011). Decoding the activity of neuronal populations in macaque primary visual cortex. *Nature Neuroscience*, *14*(2), 239–245. doi: 10.1038/nn.2733

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 770–778).

Kawaguchi, K., Clery, S., Pourriahi, P., Seillier, L., Haefner, R. M., & Nienborg, H. (2018). Differentiating between Models of Perceptual Decision Making Using Pupil Size Inferred Confidence. *The Journal of Neuroscience*. doi: 10.1523/JNEUROSCI.0735-18.2018

Ma, W. J., Beck, J. M., Latham, P. E., & Pouget, A. (2006, nov). Bayesian inference with probabilistic population codes. *Nature Neuroscience*, *9*(11), 1432–1438. Retrieved from http://www.nature.com/articles/nn1790 doi: 10.1038/nn1790

Ma, W. J., & Jazayeri, M. (2014). Neural Coding of Uncertainty and Probability. *Annual Review of Neuroscience*, *37*(1), 205–220. doi: 10.1146/annurev-neuro-071013-014017

Nagler, T., & Czado, C. (2016). *Evading the curse of dimensionality in nonparametric density estimation with simplified vine copulas* (Tech. Rep.).

Nienborg, H., & Cumming, B. G. (2009). Decision-related activity in sensory neurons reflects more than a neurons causal effect. *Nature*. doi: 10.1038/nature07821

Pouget, A., Dayan, P., & Zemel, R. S. (2003). Inference and Computation with Population Codes. *Annual Review of Neuroscience*, *26*(1), 381–410. doi: 10.1146/annurev.neuro.26.041002.131112

Walker, E. Y., Cotton, R. J., Ma, W. J., & Tolias, A. S. (2018, apr). A neural basis of probabilistic computation in visual cortex. *bioRxiv*, 365973. doi: 10.1101/365973