RESEARCH ARTICLE

Probabilistic neural transfer function estimation with Bayesian system identification

Nan Wu¹, Isabel Valera¹, Fabian Sinz², Alexander Ecker², Thomas Euler³, Yongrong Qiu²,3,4,5,6

¹Department of Computer Science, Saarland University, Saarbrücken, Germany, ²Department of Computer Science and Campus Institute Data Science (CIDAS), Göttingen University, Göttingen, Germany, ³Institute for Ophthalmic Research and Centre for Integrative Neuroscience (CIN), Tübingen University, Tübingen, Germany, ⁴Department of Ophthalmology, Byers Eye Institute, Stanford University School of Medicine, Stanford, California, United State of America, ⁵Stanford Bio-X, Stanford University, Stanford, California, United State of America, ⁶Wu Tsai Neurosciences Institute, Stanford University, Stanford, California, United State of America

*yongrong.qiu@uni-goettingen.de

Abstract

Neural population responses in sensory systems are driven by external physical stimuli. This stimulus-response relationship is typically characterized by receptive fields, which have been estimated by neural system identification approaches. Such models usually require a large amount of training data, yet, the recording time for animal experiments is limited, giving rise to epistemic uncertainty for the learned neural transfer functions. While deep neural network models have demonstrated excellent power on neural prediction, they usually do not provide the uncertainty of the resulting neural representations and derived statistics, such as most exciting inputs (MEIs), from in silico experiments. Here, we present a Bayesian system identification approach to predict neural responses to visual stimuli, and explore whether explicitly modeling network weight variability can be beneficial for identifying neural response properties. To this end, we use variational inference to estimate the posterior distribution of each model weight given the training data. Tests with different neural datasets demonstrate that this method can achieve higher or comparable performance on neural prediction, with a much higher data efficiency compared to Monte Carlo dropout methods and traditional models using point estimates of the model parameters. At the same time, our variational method provides us with an effectively infinite ensemble, avoiding the idiosyncrasy of any single model, to generate MEIs. It allows to estimate the uncertainty of stimulus-response function, which we have found to be negatively correlated with the predictive performance at model level and may serve to evaluate models. Furthermore, our approach enables us to identify response properties with credible intervals and to determine whether the inferred features are meaningful by performing statistical tests on MEIs. Finally, in silico experiments show that our model generates stimuli driving neuronal activity significantly better than traditional models in the limited-data regime.
Author summary

Neural system identification methods learn stimulus-response functions using experimental data to predict responses. These neuronal prediction models demand large amounts of training data, however, the recording time for each experiment is restricted, introducing the uncertainty about the neural features derived from trained models. Here, we present a Bayesian approach incorporating weight uncertainty to identify response functions and show that our method has higher or comparable predictive performance with a higher data efficiency compared to traditional methods using point estimates of model parameters. Additionally, our model provides an effective infinite ensemble to derive neural features, which avoid the idiosyncrasy of a single model. In this way, our method also allows us to estimate the uncertainty of the derived features and to conduct statistical tests on them. Generally, our Bayesian approach enables us to generate many similar stimuli to investigate biological information processing.

Introduction

Current neural interfaces allow to simultaneously record large populations of neural activity. In sensory neuroscience, such ensemble responses are driven by external physical stimuli (e.g., natural images), and their relation has been characterized by tuning curves or receptive fields (RFs; [1]). Such stimulus-response functions have been estimated by neural system identification methods (reviewed in [2]). Classically, they used a linear-nonlinear-Poisson (LNP) model or variants of it [3–6] to predict responses to unseen stimuli such as white noise and natural images [7, 8]. More recently, deep neural networks (DNNs) with multiple layers of non-linear processing have shown great success for learning neural transfer functions along the ventral visual stages from retina [9–11] and primary visual cortex [12–15] to higher visual areas [16, 17]. Moreover, through in silico experiments, these models are able to generate specific stimuli to control neural activity and identify novel neuronal properties from a high-dimensional space [18–22]. For example, closed-loop paradigms show that performing gradient ascent on a deep model can yield most exciting inputs (MEIs) to drive a neuron’s activity optimally [20, 23].

Yet, these system identification approaches demand significant amounts of stimulus-response pair data for the model training, given the high dimensional stimulus space and the non-linear neural transformations [9, 15, 24, 25]. Due to limited recording time for each experiment, the amount of data for fitting these models is restricted introducing epistemic uncertainty about the learned stimulus-response function. To estimate this uncertainty, traditional LNP methods obtain full posterior distribution of model parameters by leveraging a Bayesian framework to provide confidence intervals for the estimated RFs [5, 24, 26–28]. However, DNN models rarely consider the uncertainty of the neuronal properties that are recovered from in silico experiments. Typically, MEIs are computed on an ensemble of models because we want to avoid the idiosyncrasy of a single model. Utilizing a full posterior provides us with an effectively infinite ensemble through sampling. Additionally, it is important to quantify the uncertainty of derived features as we are interested in whether they are biologically meaningful. Consider a scenario: we observe elements in the surround, but they are faint. This prompts the question: Are they real? With a single MEI, we cannot answer that; however, using a model equipped with uncertainty, we can assess whether the posterior significantly deviates from zero.
Here, we propose a Bayesian system identification approach to estimate response features of neurons with uncertainties (Fig 1). We test whether incorporating uncertainties by learning the full distribution of model parameters is beneficial for learning neural representations. To this end, we build a DNN model to predict responses to unseen visual stimuli by using variational inference to estimate the distribution of network weights, i.e., Bayes by Backprop [29–32].

Our contributions are: (1) We incorporate weight variability in deep neural networks for identifying neural response functions with uncertainty and extend the Bayes by Backprop with a hyperparameter which effectively adjusts the sparsity of model parameters. (2) We apply our Bayesian models on different experimental datasets and find that our method can achieve higher or comparable performance on neural prediction, with a much better data efficiency, compared to Monte Carlo dropout methods and traditional models using point estimates of the model parameters. (3) Our approach with full posterior allows to estimate neural features with credible intervals and run statistical test for the derived MEIs, bypassing the idiosyncrasy of a single model. (4) Finally, simulation experiments demonstrate that the variational model yields stimuli that drive neuronal activation better than the traditional models in the condition of limited training data. This supports that weight uncertainty, as implemented in our model, may contribute to a more efficient identification of non-linear neuronal response functions.

**Materials and methods**

**Dataset**

We tested our method on two publicly available datasets.

The first dataset contains calcium signals driven by static natural gray-scale images for neurons in primary visual cortex (V1) of mice [12]. We used 103 neurons from the first scan field, whose single-trial responses to 1,600 images for training models and 200 for tuning hyperparameters. Then we used the mean of response repeats to 50 test images for evaluating models.

The second dataset comprises Ca^{2+} responses to natural green/UV images (36x64 pixels) for neurons in mouse V1 [21]. We selected the natural stimuli that were presented in both UV and green channels and used the neurons whose quality index ($QI = \text{Var}[E[C]_r]/E[\text{Var}[C]_r]$),
time samples \( t \) and repetitions \( r \), a response matrix \( C \) with a shape of \( t \times r, \text{E}[X] \), and \( \text{Var}[X] \), denoting the mean and variance along the dimension \( d \) of \( X \), respectively) of 10-repeat test responses were larger than 0.3. In this way, we had 161 neurons from one scan field, whose single-trial responses to 4,000 images for training and 400 for validation. Then we used mean of response repeats to 79 test images for evaluation.

**Models**

**Variational model.** DNN for system identification can be seen as a probabilistic model: given the training data \( \mathcal{D} = (\mathbf{x}, \mathbf{y}) \), where \( \mathbf{x} \) is an input (such as natural images) and \( \mathbf{y} \) is the output (such as neural responses), we aim to learn the weights \( \mathbf{w} \) of a network which can predict the output for the unseen stimuli (Fig 1). Compared to a traditional method using point estimates of the weights, Bayesian approaches learn full distributions of these \( \mathbf{w} \). Estimating the full posterior distribution of the weights \( P(\mathbf{w}|\mathcal{D}) \) given the training data is usually not feasible. An alternative is to approximate \( P(\mathbf{w}|\mathcal{D}) \) by a new distribution \( q(\mathbf{w}|\theta) \) whose parameters \( \theta \) are trained to minimize the distance between the proxy and the true posterior, which is called variational inference [29–32]. Usually we use Kullback-Leibler (KL) divergence as a measure of distance between two distributions:

\[
\theta^* = \arg \min_{\theta} \text{KL}[q(\mathbf{w}|\theta)||P(\mathbf{w}|\mathcal{D})]
\]

\[
= \arg \min_{\theta} \text{KL}[q(\mathbf{w}|\theta)||P(\mathbf{w})] - \mathbb{E}_{q(\mathbf{w}|\theta)}[\log P(\mathcal{D}|\mathbf{w})]
\]

(1)

(2)

The optimization function can be viewed as a trade-off between the distance between the variational posterior and the selected prior and the likelihood cost. We can view it as a constrained optimization problem as [33]:

\[
\arg \min_{\theta} \mathbb{E}_{q(\mathbf{w}|\theta)}[\log P(\mathcal{D}|\mathbf{w})] \quad \text{subject to} \quad \text{KL}[q(\mathbf{w}|\theta)||P(\mathbf{w})] < \epsilon
\]

(3)

Here \( \epsilon \) represents the specific distance between the variational posterior and the prior. According to KKT conditions [34] and non-negative properties of KL divergence, we get:

\[
\mathcal{F} = \mathbb{E}_{q(\mathbf{w}|\theta)}[\log P(\mathcal{D}|\mathbf{w})] - \beta_{s}(\text{KL}[q(\mathbf{w}|\theta)||P(\mathbf{w})] - \epsilon)
\]

(4)

\[
\geq \mathbb{E}_{q(\mathbf{w}|\theta)}[\log P(\mathcal{D}|\mathbf{w})] - \beta_{s}\text{KL}[q(\mathbf{w}|\theta)||P(\mathbf{w})]
\]

(5)

where \( \beta_{s} \) is non-negative and represents a Lagrangian multiplier. So the final loss function for the model is:

\[
\mathcal{L} = \beta_{s}\text{KL}[q(\mathbf{w}|\theta)||P(\mathbf{w})] - \mathbb{E}_{q(\mathbf{w}|\theta)}[\log P(\mathcal{D}|\mathbf{w})]
\]

(6)

\[
\approx \sum_{i=1}^{n} \beta_{s}(\log q(\mathbf{w}^{(i)}|\theta) - \log P(\mathbf{w}^{(i)})) - \log P(\mathcal{D}|\mathbf{w}^{(i)})
\]

(7)

Eq (7) is a result of Monte Carlo sampling \( n \) instances \( \mathbf{w}^{(i)} \) from \( q(\mathbf{w}|\theta) \) because we can not calculate (6) directly. Note that the use of \( \beta_{s} \) is equivalent to a common operation of tempering likelihood in KL-weighted evidence lower bound [35–37].

Here, we implemented convolutional neural networks (CNNs) for all experiments. For a CNN using variational inference on model weights (variational model), we picked independent Gaussian distributions for the variational posterior and a scale mixture of two Gaussians.
for the prior [32]. The log posterior was defined as

$$\log q(\mathbf{w}|\mathbf{\theta}) = \sum_{k=1}^{K} \log \mathcal{N}(w_k | \mu^k, (\sigma^k)^2)$$

where $w_k$ denotes $k$th weight of the neural network and $(\mu^k, \sigma^k)$ are the posterior parameters $\mathbf{\theta}^k$. We omitted the superscript $k$ for $\mu$ and $\sigma$ in other formulas to maintain simplicity when there is no ambiguity. To keep $\sigma$ non-negative, we parameterised it using $\sigma = \log(1 + \exp(\rho))$.

We selected the log prior

$$\log P(\mathbf{w}) = \sum_{k=1}^{K} \log \left( \pi \mathcal{N}(w_k | 0, \sigma_1^2) + (1 - \pi) \mathcal{N}(w_k | 0, \sigma_2^2) \right)$$

where $\pi$ is a mixture component weight ($0 \leq \pi \leq 1$) [32, 38]. This prior, compared to a single Gaussian distribution, encourages sparseness in learned kernels, reminiscent of neural representations in visual systems [39–42]. The likelihood loss depends on the specific task of the network. For neural system identification, we use Poisson loss $-\log P(D|\mathbf{w}) = \sum_i r_i - \mathbf{r} \log \mathbf{r}_\mu$, where $l$, $r_i$, and $\mathbf{r}$ denote neuronal index, prediction responses and true responses, respectively.

**Baseline and control models.** We used a CNN without any regularization as a baseline model and used a CNN with L2 regularization in each convolutional layer and L1 regularization in fully connected layer (L2+L1) as a control model. We adopted an ensemble of L2+L1 models with different initialization seeds as a second control model, whose predicted responses are the average of five model outputs. To examine the contribution from weight uncertainties, we built a maximum a posteriori (MAP) model which contains prior and likelihood terms in Eq (7) as loss functions. Additionally, as a fourth control, we adopted a CNN with Monte Carlo dropout for probabilistic prediction; it used the same dropout rate for each model layer and in both training and test stages [43, 44].

**Model details.** The CNN model for the first dataset shared by Antolik and colleagues consisted of a convolutional layer (24x1x9x9, output channels x input channels x image width x image height), a rectified linear unit (ReLU) function, another convolutional layer (48x24x7x7, output channels x input channels x image width x image height), another ReLU function, and —after flattening all dimensions—one fully connected (FC) layer (103x13872, output channels x input channels), followed by an exponential function (S1 Appendix). We used stride = 1 and no padding for both convolutional layers. We trained the six models and tuned their respective hyperparameters. For the variational one, we tested different parameters for prior distribution on validation data, such as $\pi = 0$ or $\pi = 0.5$, $\sigma_1 = 1$ or $\sigma_1 = 100$, $\sigma_2 = \exp(-3)$ or $\sigma_2 = \exp(-6)$, and found that a scale mixture of two Gaussians had similar predictive performance, higher than one Gaussian distribution. As the predictive performance was similar for distinct priors on model layers, we used the same prior distribution with parameters $\pi = 0.5$, $\sigma_1 = 1$, $\sigma_2 = \exp(-6)$ for all layers. We also examined the number of Monte Carlo sampling times for model training and found that the predictive performance was similar for different numbers. Therefore, we used 1 or 2 sampling times for all model training.

The CNN model for the second dataset shared by Franke and colleagues contained a convolutional layer (48x2x9x9), a ReLU function, another convolutional layer (48x48x7x7), another ReLU function, and one FC layer (161x52800), followed by an exponential function. We used stride = 1 and no padding for both convolutional layers.

**Training and evaluation**

We trained all models with a learning rate of 0.0003 for a maximum of 200 epochs using the Adam optimizer [45]. We computed linear correlation (correlation coefficient, CC) between predicted and recorded responses, which was used to evaluate models on validation or test
data. We tuned model hyperparameters and selected the ones as well as the respective epoch number with the best predictive performance on validation data. We also evaluated models on test data using root mean square error (RMSE) and log likelihood, however, similar to other studies on neural prediction [9, 10, 12, 15], we primarily used CC for analysis. To keep the comparison fair, the test models shared similar network architecture for each dataset, except that the dropout model featured dropout layers.

For each trained model, we estimated MEIs of all neurons by running gradient ascent on a random input image for 100 steps with a learning rate of 10 and we picked the stimulus with the highest activity [20, 46]. All generated MEIs had the same mean and standard deviations as the training images. For the two probabilistic (variational and dropout) models, we ran the estimation for 100 times with Monte Carlo sampling, hence, we got 100 MEIs (matrix the training images. Note that we fixed the random seed/state for each sampling, in this way, model weights did not change stochastically during the iterative generation of each MEI. We defined MEI variance of one neuron as MEI variance = \(E[\text{Var}[C]]\) where \(C\) is the output of the last layer for each recorded neuron. All generated MEIs had the same mean and standard deviations as the training images. For the two probabilistic (variational and dropout) models, we ran the estimation for 100 times with Monte Carlo sampling, hence, we got 100 MEIs (matrix \(C\)) for each recorded neuron. Note that we fixed the random seed/state for each sampling, in this way, model weights did not change stochastically during the iterative generation of each MEI. We defined MEI variance of one neuron as MEI variance = \(E[\text{Var}[C]]\) where \(C\) is the output of the last layer.

Results

**\(\beta\), balances model capacity and data likelihood**

Compared to a conventional evidence lower bound in Eq (2), we used a Lagrangian multiplier \(\beta\) in (7) by borrowing the idea of constrained optimization from [33]. In this way, Blundell and colleague’s work can be seen as a special case of \(\beta = 1.0\) [32]. We first analyzed the possible roles of \(\beta\). We investigated it from the perspective of information theory, given that Eq (7) has a similar form to the objective functions in deep variational information bottleneck [47, 48] and \(\beta\)-VAE [33, 49].

The training objective jointly minimizes the KL divergence between the posterior \(q(w|\theta)\) and the prior \(P(w)\) and maximizes the data likelihood under the distribution \(q(w|\theta)\). The distribution distance becomes zero when \(q(w|\theta) = P(w)\). The prior comprises a mixture of two Gaussian distributions centered at zero, with one exhibiting a relatively small standard deviation (e.g., a distribution with \(\sigma_2 = 1, \sigma_3 = \exp(-6)\)). Large \(\beta\), downweights the log likelihood \(\log P(D|w)\). In the extreme case of very large \(\beta\), the variational posterior \(q(w|\theta)\) converges towards the prior distribution \(P(w)\). However, given that the posterior is restricted to a single Gaussian and that minimizing the reverse KL results in mode-seeking and zero-forcing, the weights \(w\) of the posterior will largely be forced to zeros. In such a case, the model has very sparse parameters but very limited expressive power. Therefore, \(\beta\) can be interpreted as a coefficient to adjust the sparsity of model parameters for fitting the data.

Empirically, we measured the sparsity of the model weights means (\(\mu\)) for different \(\beta\) values on the dataset 1 shared by by Antolik and colleagues [12]. We computed the proportion of mass volume near zero, e.g., within a certain threshold, for the weight means. We used thresholds of \(\pm 2\times \sigma_2, \pm 3\times \sigma_2\) and \(\pm 4\times \sigma_2\) and observed an increase of the ratio with the rise of \(\beta\) for the three conditions, indicating an increase of sparsity of model weights (Fig 2). Therefore, the hyperparameter \(\beta\) served to tune the model capacity via weight sparseness for data prediction.
System identification incorporates model uncertainty to predict neural responses

We trained the six models on the dataset 1 (Fig 3a) and tuned their respective hyperparameters using validation data. For the variational model, we found the one with $\beta_v = 0.1$ had best predictive performance with a sharp decrease when increasing $\beta_v$ till 1.0 or 3.0 (S1 Fig). We also observed that at training stage, the variational model presented a more stable performance on validation data compared to the baseline CNN, confirming the regularization effect of prior to prevent overfitting.

Next, we selected the hyperparameters achieving the best performance on validation data for each model. To examine the feature properties learned by these models, we estimated the MEIs of recorded neurons and found that these models yielded antagonistic center-surround and Gabor filters in a local region, reminiscent of neural representations in early visual processing ([1, 3]; Fig 3a and S2). To compare the performance of neural prediction, we then evaluated all models using test data. For a probabilistic model, we ran model predictions for 100 sampling times and computed the mean and the standard deviation of neuronal responses. We used a significance level of 0.01 after applying a Bonferroni correction. When using RMSE as a metric and 50% of the training data (Fig 3b), we observed that the variational model had similar prediction accuracy with the MAP and MC dropout, and outperformed other methods. When using RMSE and full data, our Bayesian one had higher predictive performance than others. When using log likelihood (Fig 3c), our Bayesian method had significantly higher prediction accuracy than others for both training data sizes. When using linear correlation and half of the data (Fig 3d), the variational one had similar prediction accuracy with MC dropout, the baseline, and MAP, but outperformed the L2+L1 and the ensemble. When using CC as a metric and full training data, our method had similar prediction performance with MC dropout and the ensemble, but outperformed other test models and Klindt’s model. In summary, our results suggested that the variational method had better/comparable predictive performance compared to other models.

We also reanalyzed the influence of $\beta_v$ on prediction using linear correlation with test data (Fig 3e). Similar to the case with validation data, we noticed a rather steady predictive performance with increasing $\beta_v$ until a sudden drop at $\beta_v = 1.0$ or 3.0, implying that a large Lagrangian multiplier imposing excessive sparsity on weights yields model underfitting. Note that we
got best predictive performance when using $\beta_v < 1.0$, which is referred to as the cold posterior effect and is consistent with previous results [35–37, 50].

Together, the superior/equivalent performance of our variational approach suggests that incorporating weight uncertainty is beneficial for predicting neural responses.

**Probabilistic models learn variance of neural transfer functions**

Probabilistic models predict neuronal responses to test stimuli with uncertainty. We first assessed such uncertainty by conducting standard calibration analysis (Fig 4a). In this way, we can address questions such as whether the predicted 90% credible interval contains the recorded responses 90% of the time. We did calibration analysis on the test data for credible intervals from 0% to 100%, and found that the predicted confidence was higher than the observed confidence for both the variational and MC dropout models. This suggested that both models were overconfident on their predictions, which might be caused by inappropriate prior, bad likelihood function, etc.

Next, we investigated the relationship between predictive performance and response uncertainty using test data (S3 Fig). We did not observe a significant correlation for the variational model ($\text{CC} = 0.24, p = 0.05$), the MC dropout model ($\text{CC} = 0.18, p = 0.07$) and the ensemble
model \( (CC = 0.19, p = 0.06) \), suggesting no relation between predictive performance and response uncertainty at neuronal level. This might be related to the differential response variability driven by distinct stimuli [51, 52]. Therefore, our results suggest no exact linear correlation between the response uncertainty and predictive performance.

Furthermore, we tested whether the variability of the learned transfer function was related to the predictive performance. At model level, we found a sudden increase of overall MEI variance at \( \beta_v = 1.0 \) or 3.0 (Fig 4b), where an abrupt drop of model performance was present (cf. Fig 3e). This opposite change between MEI variability and predictive performance was confirmed by the negative correlations between overall MEI variance and overall response \( CC \) \( (CC = -0.95, p < 0.0001; \text{Fig 4c}) \), suggesting a model with higher predictive performance have higher confidence on the learned transfer function. At neuronal level (S3 Fig), we computed the partial correlation between response \( CC \) and MEI variance by removing the effect from the mean firing rate. We observed significant correlation for the ensemble model \( (CC = -0.63, p < 0.0001) \), but not for the variational model \( (CC = -0.17, p = 0.09) \) and the MC dropout model \( (CC = -0.07, p = 0.50) \). The inconsistency between these three models suggest no exact linear relationship between prediction accuracy and MEI uncertainty at neuronal level.

In summary, these results demonstrate that, at model level, a probabilistic model with smaller uncertainty on the learned stimulus-response function yields higher predictive performance.

**Variational model features high data efficiency on neural prediction**

Here we applied our method on the second dataset shared by Franke and colleagues ([21]). After hyperparameter tuning, we selected \( \beta_v = 0.3 \) for the variational network and evaluated the five models on test data.

We observed that our Bayesian method had better/comparable prediction accuracy compared to other models when using linear correlation, RMSE and log likelihood as evaluation metrics (Fig 5a and S4). We then examined the relationship between the uncertainty of the learned stimulus-response function and the predictive performance measured with \( CC \) at model level. We expect that, with more data used for training, the model yields better prediction along with smaller variance for the learned MEIs. We focused on the variational method. Indeed, when more training data was used, the predictive model performance increased (Fig 5a) while the overall MEI variance decreased (Fig 5b), with a negative correlation between
them at model level ($CC = -0.73$, $p < 0.0001$; Fig 5c). Note that we did not observe a steady decrease of the overall response variance (S5 Fig).

Next, we investigated whether the performance difference between the variational and the L2+L1 model was sensitive to the training data size (Fig 5d). We observed that the variational method had higher correlations except for the case of extremely little data (20%). The difference peaked at 40% with an increase of 9% ($p < 0.0001$, two-sided permutation test with $n = 10,000$ repeats) and gradually decreased with more training data, indicating the benefit of variational inference for system identification. We note that such improvement in model performance might vanish in the scenario of unlimited data. We also compared the predictive performance on individual neurons at one random seed and observed that, the Bayesian model outperformed the L2+L1 one for the condition of 40% of training data ($p < 0.0001$; Fig 5e) and two methods had comparable performance for the condition of full data ($p = 0.0927$; Fig 5f).

Together, compared to a traditional method, our Bayesian approach with weight uncertainty yielded higher predictive performance with a higher data efficiency.

**Variational model yields stimuli driving high neuronal activation**

Bayesian methods with full posterior provide an infinite ensemble of models for computing MEIs and allow to perform statistical tests for the derived features. Before using the variational
model to yield MEIs, we evaluated the effectiveness of our method. We fed the verified MEIs from [21] into the model trained by full data and estimated the neuronal activation driven by distinct stimuli. We assumed that, if a model was well trained, each verified MEI would drive higher activity for the corresponding neuron compared to other stimuli. Firstly, we selected randomly 6 neurons with their MEIs and got 6 activation distributions by sampling the model for 100 times for each neuron (Fig 6a; S7 Fig). We observed that each neuron was activated most by its respective MEI. Furthermore, we used the neurons (n = 40) with available verified MEIs and prediction correlation values exceeding 0.5, and fed these preferred stimuli into our

**Fig 6. In silico experiments of neuronal activity with derived MEIs.** (a) Activation distributions of 6 exemplary neurons driven by the 6 verified MEIs (green and gray representing the respective MEI and remaining MEIs, respectively). (b) Response matrix of each neuron activated by the verified MEIs of all neurons. Scaling was applied to each row to ensure that the maximum of the responses to all stimuli is equal to one. (c) Estimated MEIs for L2+L1 (first row) and variational (second row) models, MEI_std (third row), as well as significance map (fourth row; white, \( p < 0.01 \), one-sample two-sided permutation test against zero for 10,000 repeats), for three exemplary neurons when using 40% of training data. MEI and MEI_std in the UV channel, with different color scales. Note that MEI has much larger absolute values than MEI_std. (d) 1D histogram of neuronal activity driven by the generated MEIs from the variational model for Neuron 1 when using 40% of training data. Insets: example MEIs with corresponding activation indicated by dotted lines (red, maximum of L2+L1; green, variational). (e) Scatter plot of activation driven by MEIs yielded from variational (using the weight mean \( \mu \) and L2+L1 models at one random seed when using 40% of training data. Each dot representing one cell. (f,g,h) Same with (c), (d) and (e), but using 100% of training data.
trained model. We observed that 92.5% of neurons were elicited by the respective MEI with more than 80% of highest activity (Fig 6b), suggesting that our Bayesian method works effectively well.

To further assess which method generates the more exciting stimuli for each cell, we conducted in silico experiments using a held-out L2+L1 model trained by full data as a digital testbed. We used a CNN model with regularization instead of other models as previous studies have demonstrated its feasibility on yielding cells’ preferred stimuli [20–23]. We fed the MEIs generated by other models to the testbed and compared the neural activity. Specifically, we tested the model using variational inference and the one using L2 and L1 regularization with 40% and 100% of training data.

We found that these learned filters resembled neural features in the early visual system [1, 3] and localized more in the visual field with more training data (Fig 6c and 6f; S8 Fig). Like for the first dataset (Fig 3a), MEI_std was not uniform across visual space, e.g., some presented Gaussian or bar shapes. Additionally, we examined whether the posterior of each pixel differs significantly from zero for the 100 sampled MEIs and found that the significance map may indicate zero-crossings in visual representations. For an example neuron, we measured the responses for all the 506 MEIs yielded from five variational models, and observed that these stimuli drove this neuron with quite different activity, with the maximum response larger than the maximum one yielded (from 6 MEIs) by the traditional models (Fig 6d and 6g). With more training data, the activation distribution shifted towards higher mean with smaller variance. Additionally, we compared the activation on individual neurons for two methods (Fig 6e and 6h), and observed that the Bayesian approach yielded significantly higher responses for the condition using 40% ($p = 0.0473$, two-sided permutation test with $n = 10,000$ repeats) of training data and comparable responses for the condition using full data ($p = 0.2114$).

In summary, our variational model allowed statistical test for the derived response functions and yielded the stimuli driving neurons better than traditional methods in the limited data regime, suggesting that weight uncertainty benefits the learning of neural representations.

Discussion

We presented a Bayesian approach for identification of neural properties by incorporating model uncertainty through learning the distribution of model weights, aiming to estimate neural features with credible intervals. Our empirical results on different datasets show that the variational method had higher or comparable predictive performance, especially in the limited data regime, compared to methods using dropout or traditional methods learning point estimates of model parameters. Moreover, by sampling from posterior distribution of model weights, our approach enabled to provide credible intervals and test statistics for the learned MEIs, avoiding the idiosyncrasy of a single model. Finally, in silico experiments show that the variational model yielded the MEIs driving neurons with higher activity compared to the traditional model when limited data were used for training. This suggests that model uncertainty contributes to learning neural transfer functions with a high data efficiency.

Variational models with cold posterior effects

Our Bayesian method had best prediction accuracy when using log likelihood as an evaluation metric. The results with three metrics demonstrated that the variational model had higher/comparable predictive performance compared to others. We noticed that the improvement was marginal for many conditions, primarily occurring in a limited data regime. Such improvement was sensitive to the data size and might disappear with the increase of training data. However, high data efficiency is of great importance given the high cost of data collection and...
the limited recording time in neuroscience. A common scenario is that non-linear response functions, with a high-dimensional stimulus space, are learned from limited amounts of recorded data.

Compared to a standard objective function, we used $\beta_v$ to adjust the weighting between the prior and the likelihood and achieved the best predictive performance with $\beta_v < 1.0$. [35] did many hypothesis tests to study the origin of this cold posterior effect, including minibatch noise, bias-variance tradeoff, dirty likelihood, prior variance, training set size, model capacity and initialization. They showed that the inappropriate prior was related to the predictive performance improvement for $\beta_v < 1.0$ and claimed that this was probably not the only cause. Therefore, we should be more careful when interpreting the inferred posterior. In fact, we did observe that the variational method and the MC dropout method underestimated the uncertainty of model parameters, suggesting future work on calibrating these models [53, 54]. Additionally, such calibration analysis could be used in future studies to evaluate model reliability, compare model assumptions and guide model selections. Furthermore, it might be interesting to use proper scoring rules, e.g., Continuous Ranked Probability Score, to evaluate the accuracy of probabilistic predictions for the variational method [55, 56].

Relation to trial-to-trial variability

Neural information process is probabilistic, i.e., neurons respond with trial-to-trial fluctuations to a repeated presentation of a stimulus [57, 58]. Response variability is found across neural systems, originating from diverse factors, such as synapse variation, channel noise, brain state, and attention [52, 59–63]. Additionally, the variability between populations of neurons are correlated. In a simplified case, a pair of neurons may present correlations for the single-trial responses, i.e., pairwise noise correlation, which also contributes to neural coding ([64]; reviewed in [65–68]). Such response variability is inherent in neural data itself and is a kind of aleatoric, but not the epistemic uncertainty. We note that the standard deviations of the estimated MEIs from our models decreased with the increasing amounts of training data, suggesting that the variability of the sampled predicted responses may not be related to the response uncertainty in biological neurons or our models may predict a mix of both uncertainties (S6 Fig). Additionally, the uncertainty in $q(w|\theta)$ should be primarily related to the epistemic uncertainty, it could converge to zero in the unlimited-data regime. Still, it may be related to aleatoric uncertainty too, as we may use suboptimal loss functions and model architectures, both of which may also reflect a model’s inability to capture the inherent noise in data.

Necessity of uncertainty quantification for yielded preferred stimuli

Though DNN approaches have demonstrated remarkable power in predicting neural responses to diverse stimuli and generating novel hypothesis about neuronal features, they require significant amounts of stimulus-response pair data for the training. Besides the epistemic uncertainty introduced by limited data, such hypothesis also entails further closed-loop animal experiments to verify the derived properties, which consumes much experimental time [20, 23]. Still, it is impossible to confirm the yielded preferred stimuli for all neurons across the high-dimensional stimulus space with experiments. Practically, only a subset of cells are selected for verification. Probabilistic MEIs allow us to pinpoint those neurons with interesting but subtle/uncertain derived features. Therefore, it is critical to quantify the uncertainty of the yielded representations for all recorded neurons [69, 70]. Additionally, the credible interval of the derived features offers an opportunity to generate an ensemble of infinite preferred stimuli. An interesting feature would be to compare the neuronal activity driven by these similar MEIs in animal experiments, which may allow to test the robustness of the biological system.
We note that, even for a model using point estimate of parameters (such as L2+L1), it may yield different preferred stimuli by initializing the MEI generation randomly. Yet, this uncertainty depends on the starting points of non-convex optimizations, rather than the training data. Empirically, we found that such variance was quite stable when using L2+L1 models with different amounts of training data and was also much smaller than the MEI variance we computed (cf. Fig 5b). Therefore, the measure of epistemic uncertainty calls for a Bayesian framework or an ensemble of many models.

Bayesian models, unlike traditional ‘point-estimation’ approaches, offer the advantage of quantifying the uncertainty of inferred neural features. Nevertheless, in seeking to capture the full distribution of weights, they entail increased complexity including model parameters and hyperparameters, which in turn necessitate higher computational demands and may pose challenges for interpretation.

**Future work & general impact**

Incorporating uncertainty to DNNs have flourished in recent years [53, 54, reviewed in], including Bayesian methods which specify a prior distribution for network weights and approximate the full posterior given the training data using different tricks such as variational inference [32, 71], Laplace approximation [72, 73] and expectation propagation [74]. Non-Bayesian methods include applying MC dropout in the network [44] or training an ensemble of models that are initialized by different seeds [75]. In the future, our variational approach could be extended by more advanced methods such as incorporating correlated model parameters [71] or jointly considering model and parameter uncertainty [76]. Additionally, other uncertainty representation methods such as Stochastic Weight Averaging-Gaussian [77] are interesting and promising directions, especially for large-scale model architectures and datasets. While these methods are powerful to predict uncertainty, it would be interesting to investigate biologically inspired methods such as adding noise to network parameters/activation in the future.

We used a Poisson loss function to train our model, which may not be optimal considering that neuronal responses may present non-exact Poisson behavior [51]. In this case, our model may capture only a part of trial-to-trial variability in neural data. Such response fluctuation depends on many conditions, including biochemical process, internal brain states and engaged behavioral tasks [51, 52, 59, 60]. These factors have been described by a low-dimensional latent state models [52, 78, 79]. Therefore, a potential extension of our method could be a variational network incorporated with latent state variables.

Bayesian frameworks could potentially advance neural prediction in multiple ways. One promising direction is Bayesian model comparison. It enables us to select model hyperparameters and network architectures with promising out-of-distribution detection performance by computing marginal likelihood. However, it is challenging in deep neural networks and requires additional estimation methods such as Gauss-Newton approximations to the Hessian [80–82]. A starting point could be to try stochastic variational inference which estimates both model and weight uncertainty [76]. Another interesting research is to estimate the amount of data for achieving performance saturation for specific animal-electrode noise-recording method combinations. This may involve leveraging techniques such as Bayesian optimization [83, 84] and Bayesian experimental design [85, 86]. However, it may pose a considerable challenge. Notably, neurons at different brain regions may feature different levels of nonlinear processing. As a result, cells at higher visual areas probably demand more training data compared to at the retina. Additionally, it is important to calibrate a probabilistic model, which could overestimate/underestimate the uncertainty of the posterior. Furthermore, trial-to-trial variability inherent in neural data varies across brain regions [51], potentially influencing...
predictive performance as well. Considering our model may not capture such aleatoric uncertainty well, alternative probabilistic methods may be demanded. A promising start may be to use Bayesian experimental design to actively select the informative stimuli during closed-loop experiments to fit response functions and distinguish between models [24].

Our in silico experiments indicate that the stimuli generated by the variational model driving higher neuronal activation than the CNN with regularization, which requires future animal experiments to test. Additionally, we noticed that the MEI_std was not uniform in the visual field for each neuron and its location was not overlaid with the central MEI, for example, it seems to sit on the surround of the corresponding MEI. It would be interesting to examine and quantify the MEI uncertainty in regard of visual space, which might be related to contextual sensory processing [87–89].

More generally, why do we care about the uncertainty of the estimated neural representations? Even with closed-loop experiments, it is impossible for us to test all potential (preferred) inputs for the recorded neurons [20, 21, 23]. Therefore, we always expect to have a confidence interval for the test statistics. Besides, a Bayesian model offers a manner to generate many stimulus candidates by sampling for stimulating neural systems, which may offer new insights for understanding the biological computation.

Supporting information

S1 Appendix. Additional model details.
(PDF)

S1 Fig. Neural prediction for first dataset. (a,b) Predictive performance (correlation coefficient, CC) based on validation data during training for variational models ($\beta_v = 0.1$) with different prior distributions. All layers adopted the same $\sigma_2 = \exp(-6)$ with different $\pi$ and $\sigma_1$ values (a), or with the same parameters of prior distribution (b). We picked $\pi = 0.5$, $\sigma_1 = 1$, $\sigma_2 = \exp(-6)$ for subsequent model training. (c) Predictive performance based on validation data during model training for different numbers of Monte Carlo sampling. We picked number = 1 or 2 to save training time. (d) Model performance based on validation data during training for the baseline and variational models with different $\beta_v$ values. (e) Overall variance of predicted responses to test stimuli for different $\beta_v$ values. (f) Histogram of response variance (top) and MEI (RF) variance (bottom) for the variational and the MC dropout models. Dotted line represents the mean of histogram. (g) Model performance (left) based on test data and RF overall variance (right) for two probabilistic models with different amounts of training data. Error bars in (e) and (g) represent standard deviation of $n = 10$ random seeds for each model.
(TIF)

S2 Fig. Estimated MEIs for first dataset. (a,b) MEIs of 30 exemplary neurons for the first dataset generated by the L2+L1 model (a) and the variational one (b).
(TIF)

S3 Fig. Uncertainty analysis for first dataset. (a) Scatter plot of response CC and response variance for variational model at one random seed (each dot representing one neuron; $CC = 0.24, p = 0.05$). (b) Scatter plot of response CC and MEI variance for variational model at one random seed (each dot representing one neuron; $CC = −0.37, p = 0.0001$). (c) Scatter plot of mean firing rate and MEI variance for variational model at one random seed (each dot representing one neuron; $CC = −0.47, p < 0.0001$). (d) Same with (a), but for MC dropout model ($CC = 0.18, p = 0.07$). (e) Same with (b), but for MC dropout model ($CC = −0.23, p = 0.02$). (f) Same with (c), but for MC dropout model ($CC = −0.33, p = 0.006$). (g) Same with (a), but for ensemble model ($CC = 0.19, p = 0.06$). (h) Same with (b), but for ensemble model ($CC = −0.34$, $p = 0.05$).
(i) Same with (c), but for ensemble model (CC = −0.35, p = 0.0003).

S4 Fig. Model evaluation using RMSE and log likelihood for second dataset. (a,b) Like Fig 5a but using RMSE (a) and log likelihood (b) to compare models. When using RMSE, we found our variational method had equivalent prediction accuracy to the MC dropout model in the condition of full data (p = 0.1709), and the Bayesian one outperformed the MC dropout one in conditions of less data (p = 0.0001 at 20% of data, p < 0.0001 at 40%, p = 0.0001 at 60%, p = 0.0021 at 80%). When using log likelihood, the variational model had significantly higher predictive performance than the MC dropout method (p = 0.0002 at 20%, p < 0.0001 at 40%, p < 0.0001 at 60%, p < 0.0001 at 80%, p < 0.0001 at 100%).

S5 Fig. Neural prediction for second dataset. (a) Model performance based on validation data during training for the baseline and the variational models with different $\beta_v$ values. (b,c) Scatter plot of response CC and MEI (RF) variance for MC dropout (b) and variational (c) models for 10 seeds (CC = −0.25, p = 0.001 and CC = −0.34, p < 0.0001 for dropout and variational one, each dot representing one neuron at one random seed). (d) Predictive performance, overall RF variance and overall response variance for variational models with different $\beta_v$ values. (c) Predictive performance based on validation data during model training for different numbers of Monte Carlo sampling. We picked number = 1 or 2 to save training time. (d) Model performance based on validation data during training for the baseline and the variational ones with different $\beta_v$ values. (e) Scatter plot for overall response CC and overall RF variance for the variational methods with different $\beta_v$ values (d) and at 10 seeds (CC = −0.82, p < 0.0001). Each dot represents one model. (f) Overall response variance for different amounts of training data for the variational models (10 seeds per model). (g) Scatter plot for overall response CC and overall RF variance for the dropout model with different amounts of training data and at 10 seeds (CC = −0.17, p = 0.24). Each dot represents one model. Error bars in (d) and (f) represent standard deviation of n = 10 random seeds for each model.

S6 Fig. Variance of predicted vs. recorded responses for second dataset. Using the trained models, we tested whether the variance of predicted responses was related to the variance of recorded responses for each neuron. We first estimated the predicted response variance to a stimulus. For the L2+L1 model, as the mean of neural responses is proportional to the variance, we used the model output (a single predicted value) as a substitute. For the variational one, we either used the mean of predicted responses (multiple sampling times) as a substitute or calculated the response variance explicitly. (a) Scatter plot (axes in log scale) of predicted response variance (using response mean as a substitute) and recorded response variance for one neuron for a variational model. Each dot representing one stimulus. (b) Distribution of correlations between recorded and predicted response variance for all neurons for the L2+L1, variational-mean (using response mean as a substitute) and variational-variance (calculating response variance), at one random seed. Horizontal lines representing distribution means. (c) Mean correlations between two response variances (10 seeds per model). Note that variational-variance had lower correlation than the L2+L1. Error bars represent standard deviation of n = 10 random seeds for each model. We computed the correlation using the predicted and recorded response variances of the test stimuli for each neuron (CC = 0.34, p = 0.002, Spearman correlation for an exemplary neuron; a). We found that the variational one using response mean as a substitute of variance had a slightly higher mean correlation across neurons compared to the L2+L1 (p = 0.0368, two-sided permutation tests on 10 random seeds for
Instead of performing closed-loop experiments to examine the effectiveness of our variational model, we used the verified MEIs from previous study to compare the neuronal activities driven by different stimuli [21]. We plotted 1D histograms of activation of 6 exemplary cells (from left to right) driven by the verified MEIs (from top to bottom). We used green color instead of gray to highlight the highest neuronal activation on the diagonal (driven by the respective verified MEI).

S8 Fig. Estimated MEIs for second dataset when using full training data. (a,b) MEIs in the UV channel of 30 exemplary neurons generated by L2+L1 model (a) and variational one (b).

Acknowledgments
We thank Philipp Berens, Katrin Franke, Suhas Shrinivasan and Ziwei Huang for helpful discussions.

Author Contributions
Conceptualization: Yongrong Qiu.
Data curation: Nan Wu, Yongrong Qiu.
Formal analysis: Nan Wu, Yongrong Qiu.
Funding acquisition: Fabian Sinz, Alexander Ecker, Thomas Euler.
Investigation: Nan Wu, Yongrong Qiu.
Methodology: Nan Wu, Isabel Valera, Fabian Sinz, Alexander Ecker, Thomas Euler, Yongrong Qiu.
Project administration: Yongrong Qiu.
Resources: Thomas Euler.
Software: Nan Wu, Yongrong Qiu.
Supervision: Isabel Valera, Fabian Sinz, Alexander Ecker, Thomas Euler, Yongrong Qiu.
Validation: Yongrong Qiu.
Visualization: Nan Wu, Yongrong Qiu.
Writing – original draft: Yongrong Qiu.
Writing – review & editing: Nan Wu, Isabel Valera, Fabian Sinz, Alexander Ecker, Thomas Euler, Yongrong Qiu.

References


