Title

A Low-Rank Method for Characterizing High-Level Neural Computations

Authors

Kaardal, Joel T

Theunissen, Frédéric E

Sharpee, Tatyana O

Publication Date

2017

DOI

10.3389/fncom.2017.00068

Peer reviewed

frontiers
in Computational Neuroscience

METHODS
published: 31 July 2017
doi: 10.3389/fncom.2017.00068

A Low-Rank Method for Characterizing High-Level Neural Computations

Joel T. Kaardal[1,2]*, Frédéric E. Theunissen[3] and Tatyana O. Sharpee[1,2]

[1] Computational Neurobiology Laboratory and Crick-Jacobs Center for Theoretical and Computational Biology, Salk Institute for Biological Studies, La Jolla, CA, United States, [2] Center for Theoretical Biological Physics, University of California, San Diego, La Jolla, CA, United States, [3] Department of Psychology, University of California, Berkeley, Berkeley, CA, United States

The signal transformations that take place in high-level sensory regions of the brain remain enigmatic because of the many nonlinear transformations that separate responses of these neurons from the input stimuli. One would like to have dimensionality reduction methods that can describe responses of such neurons in terms of operations on a large but still manageable set of relevant input features. A number of methods have been developed for this purpose, but often these methods rely on the expansion of the input space to capture as many relevant stimulus components as statistically possible. This expansion leads to a lower effective sampling thereby reducing the accuracy of the estimated components. Alternatively, so-called low-rank methods explicitly search for a small number of components in the hope of achieving higher estimation accuracy. Even with these methods, however, noise in the neural responses can force the models to estimate more components than necessary, again reducing the methods' accuracy. Here we describe how a flexible regularization procedure, together with an explicit rank constraint, can strongly improve the estimation accuracy compared to previous methods suitable for characterizing neural responses to natural stimuli. Applying the proposed low-rank method to responses of auditory neurons in the songbird brain, we find multiple relevant components making up the receptive field for each neuron and characterize their computations in terms of logical OR and AND computations. The results highlight potential differences in how invariances are constructed in visual and auditory systems.

Keywords: neural coding, auditory cortex, computational neuroscience, receptive fields, dimensionality reduction

OPEN ACCESS

Edited by:
Paul Miller,
Brandeis University, United States

Reviewed by:
Maoz Shamir,
Ben-Gurion University of the Negev, Israel
Tim Gollisch,
University of Göttingen, Germany

****Correspondence:***
Joel T. Kaardal
jkaardal@physics.ucsd.edu

Received: *01 May 2017*
Accepted: *07 July 2017*
Published: *31 July 2017*

Citation:
Kaardal JT, Theunissen FE and Sharpee TO (2017) A Low-Rank Method for Characterizing High-Level Neural Computations. Front. Comput. Neurosci. 11:68. doi: 10.3389/fncom.2017.00068

1. INTRODUCTION

Signal processing in neurobiological systems involves multiple nonlinear transformations applied to multidimensional inputs. Characterizing these transformations is difficult but essential to understanding the neural basis of perception. For example, neurons from successive stages of sensory systems represent inputs in terms of increasingly complex combinations of stimulus features (Felleman and Van Essen, 1991; King and Nelken, 2009). Although a number of statistical tools have been developed to analyze responses of sensory neurons, analysis of high-level sensory neurons remains a challenge because of two interrelated factors. First, to signal the presence of certain objects or events, high-level sensory neurons perform sophisticated computations that are based on multidimensional transformations of the inputs, which together form the *receptive field* of the neuron. Second, high-level neurons are unresponsive to noise stimuli and usually require structured stimuli reflective of the natural sensory environment. However, even when presented with natural stimuli, the specific combinations of inputs necessary to elicit responses of a given neuron do not occur frequently. As a result, current statistical methods fail to recover receptive fields for many high-level neurons due to a lack of sufficient sampling of the stimulus/response distribution relative to the number of model parameters. Therefore, to systematically probe high-level responses we need statistical methods that (i) estimate multidimensional transformations of the inputs, (ii) account for the biases in natural stimuli or other strongly correlated distributions, and (iii) are resistant to overfitting. Here we describe a practical method that satisfies these criteria and apply the method to gain new insights into the structure of receptive fields of high-level auditory neurons from the zebra finch auditory forebrain.

Present dimensionality reduction methods for recovering receptive fields of sensory neurons can be roughly divided into linear and quadratic methods. Linear methods attempt to reconstruct components of a neuron's receptive field by correlating the neural response to a set of features composed of stimulus components, s_i. Examples of these methods include the spike-triggered average (STA), maximally informative dimensions (MID), and first-order maximum noise entropy (MNE) methods (Sharpee et al., 2004; Bialek and de Ruyter van Steveninck, 2005; Schwartz et al., 2006; Fitzgerald et al., 2011b). With MID being a notable exception, many of these linear methods are only capable of recovering a single component of the receptive field. The necessity of characterizing multiple components of receptive fields has led to the development of quadratic methods where the feature space is expanded quadratically to include all pairwise products, s_is_j, between the components of a D-dimensional stimulus vector, $\mathbf{s}$ (Schwartz et al., 2006; Fitzgerald et al., 2011b; Park and Pillow, 2011; Rajan and Bialek, 2013). Generally speaking, such quadratic methods construct a weight matrix, $\mathbf{J}$, that captures correlations between a neuron's responses and the quadratic feature space. The relevant subspace of stimulus space that spans the receptive field is recovered by diagonalizing $\mathbf{J}$.

Methods designed to recover this relevant subspace can be susceptible to bias when the model is constructed based on incorrect assumptions. For instance, the spike-triggered covariance (STC) method (the quadratic analog of the STA method) assumes that the stimulus components are drawn from a Gaussian white noise distribution (Bialek and de Ruyter van Steveninck, 2005). When STC is applied to other stimulus distributions such as natural stimuli, the receptive field estimation is susceptible to bias and often leads to a poor reconstruction of the receptive field components. In response to this short-coming of the STC method, the MID and MNE methods were developed to minimize bias using principles from information theory. In the case of the MID method, components are found that maximize the mutual information between the response and stimuli independent of the nonlinear function relating stimuli to responses, also called the *nonlinearity* (Sharpee et al., 2004). The MNE method instead invokes the principle of maximum entropy to construct a nonlinearity that maximizes the noise entropy, $H_{\text{noise}}(y|\mathbf{s})$, between the response, y, and stimulus distribution subject to constraints on the response-weighted moments of the stimulus space; e.g., $\langle y\rangle$, $\langle y\mathbf{s}\rangle$, and $\langle y\mathbf{s}\mathbf{s}^{\mathrm{T}}\rangle$ (Jaynes, 2003; Fitzgerald et al., 2011a,b). In order to minimize bias in the receptive field estimate, H_{noise} is maximized subject to only these data-dependent constraints.

Our proposed method builds on the second-order MNE model for the probability of a binary response given a set of stimuli (Fitzgerald et al., 2011a). This model is a logistic function of a linear combination of inputs in the expanded feature space (truncated here to second-order):

$$P(y=1|\mathbf{s}) = \frac{1}{1+e^{-z(\mathbf{s})}}, \quad z(\mathbf{s}) = a + \mathbf{h}^{\mathrm{T}}\mathbf{s} + \mathbf{s}^{\mathrm{T}}\mathbf{J}\mathbf{s} \qquad (1)$$

where unknown weights a, $\mathbf{h}$, and $\mathbf{J}$ are determined by minimizing the negative log-likelihood. The order of the moments used in constructing the MNE model correspond to the order of the polynomial that appears in the argument, $z(\mathbf{s})$. Including the nth-order constraint in the MNE model leads to an additional D^n weights that must be estimated. The MNE model is truncated to second-order to facilitate the reconstruction of multi-component receptive fields while avoiding the curse of dimensionality that appears when including constraints on the model from higher-order moments. At the same time, the second-order model is sufficient to describe contributions from multiple components that excite and suppress the neurons' response (Schwartz et al., 2006); one can approximate selectivity for higher-than-second-order features through combinations of pairwise constraints (Perrinet and Bednar, 2015). Other advantages of this approach are that (i) it works with arbitrary, including natural, stimuli, and (ii) the optimization is convex, converging swiftly to a global optimum. The disadvantage of this model is that, for a D-dimensional stimulus vector $\mathbf{s}$, one needs to determine $1 + D + D(D+1)/2$ parameters of which only $1 + D + rD$ parameters will be ultimately used to specify r components obtained by diagonalizing the $D \times D$ matrix $\mathbf{J}$. Note that an arbitrary antisymmetric matrix may be added to $\mathbf{J}$ without changing the output of the nonlinearity, P, and it is therefore sufficient, but not necessary, to optimize an MNE model with $\mathbf{J}$ constrained to be symmetric where only $D(D+1)/2$ elements of $\mathbf{J}$ need to be optimized. However, this constraint was not part of the original optimization procedure (Fitzgerald et al., 2011a,b). Below we show that adding a constraint that ensures symmetric $\mathbf{J}$ improves the estimation accuracy in our proposed model.

For currently available datasets, the over-expansion of the stimulus space can be a severe limitation leading to overfitting of quadratic models. To resolve this issue, we designed low-rank MNE models with an explicit rank constraint where a rank r matrix $\mathbf{J}$ is modeled as a product of two low-rank $D \times r$ matrices, $\mathbf{J} = \mathbf{U}\mathbf{V}^{\mathrm{T}}$ (Burer and Monteiro, 2003; Bach et al., 2008; Rajan and Bialek, 2013; Haeffele et al., 2014). For models where $r \ll D$, this bilinear factorization leads to a substantial reduction in the number of parameters that are necessary to estimate. Furthermore, as is the case with many optimization methods, one can improve the robustness of estimation to noise from limited sampling through regularization that penalizes the magnitude of certain model parameters. Since we seek low-rank representations of $\mathbf{J}$, we choose to invoke *nuclear-norm* (or trace-norm) regularization to penalize $\mathbf{J}$ based on the sparsity of its eigenvalue spectrum which, in addition to improved estimation, has the advantage of allowing us the flexibility to set r as an upper bound on the rank of $\mathbf{J}$ while applying regularization to further reduce the rank of $\mathbf{J}$ (Fazel, 2002; Fazel et al., 2003; Recht et al., 2010). We apply the low-rank MNE method to recover receptive field components from recordings of neurons in regions field L and the caudal mesopallium (CM) of the zebra finch auditory forebrain subject to auditory stimulation. Our results provide novel insights into the structure of multicomponent auditory receptive fields and suggest important differences between object-level respresentations in the auditory and visual cortex.

2. RESULTS

2.1. Mathematical Approach for Low-Rank Characterization of Neural Feature Selectivity

2.1.1. Problem Set-up

An optimal low-rank MNE model is one that minimizes the negative log-likelihood function with respect to the weights a, $\mathbf{h}$, $\mathbf{U}$, and $\mathbf{V}$. The mean negative log-likelihood is:

$$L(a,\mathbf{h},\mathbf{U},\mathbf{V}) = -\frac{1}{N}\sum_t \left[y_t \log(P_t) + (1-y_t)\log(1-P_t)\right] \quad (2)$$

where P_t is introduced as a short-hand to represent the nonlinearity, $P(y = 1|\mathbf{s}_t)$, N is the number of samples, and $y_t \in [0,1]$ is the response to tth sample of the stimulus space, $\mathbf{s}_t$.

Additional structure can be imposed on the weights by adding a penalty function to the mean negative log-likelihood (Equation 2). Nuclear-norm regularization is defined as the sum of absolute values of the eigenvalue spectrum (i.e., $\sum_k |\sigma_k|$ where σ_k is the kth eigenvalue of $\mathbf{J}$). When applied as a penalty function, the nuclear-norm increases the sparsity of $\mathbf{J}$'s eigenvalue spectrum (Fazel, 2002; Fazel et al., 2003; Recht et al., 2010). Because matrix $\mathbf{J}$ can possess both positive and negative eigenvalues, the nuclear-norm does not simply equal its trace. While in principle one can compute the nuclear-norm of $\mathbf{J}$ by diagonalization it is in practice more efficient to embed $\mathbf{J}$ within a larger positive semidefinite matrix, (Fazel, 2002; Fazel et al., 2003):

$$\mathbf{Q}\mathbf{Q}^{\mathrm{T}} = \begin{bmatrix}\mathbf{U}\\ \mathbf{V}\end{bmatrix}\left[\,\mathbf{U}^{\mathrm{T}},\ \mathbf{V}^{\mathrm{T}}\,\right] = \begin{bmatrix}\mathbf{U}\mathbf{U}^{\mathrm{T}}, & \mathbf{J}\\ \mathbf{J}^{\mathrm{T}}, & \mathbf{V}\mathbf{V}^{\mathrm{T}}\end{bmatrix} \quad (3)$$

and instead take the trace over $\mathbf{Q}\mathbf{Q}^{\mathrm{T}}$:

$$\ell_*(\mathbf{Q}) = \frac{1}{2}\sum_{k=1}^{r}\epsilon_k \left\|\mathbf{Q}_{\bullet,k}\right\|_2^2 = \frac{1}{2}\sum_{k=1}^{r}\epsilon_k\left(\left\|\mathbf{U}_{\bullet,k}\right\|_2^2 + \left\|\mathbf{V}_{\bullet,k}\right\|_2^2\right) \quad (4)$$

where $\|\cdot\|_2$ is the ℓ_2-norm and $\mathbf{Q}_{\bullet,k}$, $\mathbf{U}_{\bullet,k}$, and $\mathbf{V}_{\bullet,k}$ refer to the kth column of each matrix. Here, $\epsilon_k \geq 0$ is a regularization parameter which is a hyperparameter that controls the strength of the nuclear-norm penalty. Regularizing over this semidefinite embedding penalizes the rank of $\mathbf{U}\mathbf{U}^{\mathrm{T}}$ and $\mathbf{V}\mathbf{V}^{\mathrm{T}}$. This leads to a penalization of the rank of $\mathbf{J}$ by proxy since $\mathrm{rank}(\mathbf{J}) \leq \min\left(\mathrm{rank}(\mathbf{U}), \mathrm{rank}(\mathbf{V})\right)$ (Fazel, 2002; Fazel et al., 2003; Cabral, 2013) shown in the following.

Proof. Since $\mathbf{U}$ spans the same range space as $\mathbf{U}\mathbf{U}^{\mathrm{T}}$ and $\mathbf{V}$ spans the same range space as $\mathbf{V}\mathbf{V}^{\mathrm{T}}$, it can be shown that regularizing over the trace of $\mathbf{Q}\mathbf{Q}^{\mathrm{T}}$ is an effective strategy for regularizing the rank of $\mathbf{J}$ by showing that $\mathbf{J}$ has zero projection into the null space of $\mathbf{U}$ and $\mathbf{V}$. The null space operators of $\mathbf{U}$ and $\mathbf{V}$ are defined as $\mathcal{P}_{\mathcal{N}}(\mathbf{U}) = \mathbf{I} - \mathbf{U}\mathbf{U}^{\dagger}$ and $\mathcal{P}_{\mathcal{N}}(\mathbf{V}) = \mathbf{I} - \mathbf{V}\mathbf{V}^{\dagger}$ where † indicates a generalized matrix inverse. Projecting $\mathbf{J}$ onto $\mathcal{P}_{\mathcal{N}}(\mathbf{U})$ yields $\mathcal{P}_{\mathcal{N}}(\mathbf{U})\mathbf{J} = \mathbf{J} - \mathbf{U}\mathbf{U}^{\dagger}\mathbf{U}\mathbf{V}^{\mathrm{T}} = \mathbf{J} - \mathbf{U}\mathbf{V}^{\mathrm{T}} = \mathbf{0}$. Similarly, $\mathbf{J}\mathcal{P}_{\mathcal{N}}(\mathbf{V}) = \mathbf{0}$. Therefore, the range space of $\mathbf{J}$ is a subset of the range spaces of $\mathbf{U}$ and $\mathbf{V}$ where $\mathrm{rank}(\mathbf{J}) \leq \min(\mathrm{rank}(\mathbf{U}), \mathrm{rank}(\mathbf{V}))$ and penalizing $\mathrm{Tr}(\mathbf{Q}\mathbf{Q}^{\mathrm{T}})$ (where $\mathrm{Tr}(\cdot)$ is the trace) is an effective surrogate to the nuclear-norm of $\mathbf{J}$ for penalizing the rank of $\mathbf{J}$.

This surrogate regularization readily works with gradient based methods for optimization, whereas diagonalization of $\mathbf{J}$ does not. Unlike typical implementations of the nuclear-norm that use only a single regularization parameter (i.e., $\epsilon_k = \epsilon$ for all k), we found that assigning a unique regularization parameter for each of the r columns of $\mathbf{Q}$ led to substantial improvement in the characterization of $\mathbf{J}$. A single regularization parameter has the tendency to eliminate insignificant components at the expense of degrading the quality of the significant high variance components. Using multiple regularization parameters allows us to eliminate the insignificant components while avoiding degradation of the significant components.

While the bilinear factorization of $\mathbf{J}$ into $\mathbf{U}$ and $\mathbf{V}$ sets an upper bound of r on the rank of $\mathbf{J}$, there is a subtle inconsistency in how the rank behaves caused by the symmetry of the problem. In the present formulation of the optimization problem, $\mathbf{J}$ can be nonsymmetric and therefore possess an undesirable complex eigenvalue spectrum. This issue cannot simply be solved by symmetrizing $\mathbf{J} \leftarrow \mathbf{J}_{\mathrm{sym}} = \frac{1}{2}\left(\mathbf{J} + \mathbf{J}^{\mathrm{T}}\right)$ post-optimization. While symmetrizing would provide a real eigenvalue spectrum, this symmetrization procedure can have the unintended consequence of increasing the rank of $\mathbf{J}$ up to $2r$. This can be a problem because there are generally not enough variables provided by the $D \times r$ matrices $\mathbf{U}$ and $\mathbf{V}$ to fit a rank $2r$ matrix. We resolved this inconsistency by requiring that $\mathbf{U}$ and $\mathbf{V}$ satisfy:

$$\mathbf{U}\mathbf{V}^{\mathrm{T}} = \mathbf{V}\mathbf{U}^{\mathrm{T}} \implies \mathbf{J} = \mathbf{J}^{\mathrm{T}}, \quad (5)$$

with proof that this guarantees the rank of $\mathbf{J}$ is invariant to symmetrization provided in the following.

Proof. The symmetry constraint (Equation 5) is a sufficient condition to guarantee $\mathrm{rank}(\mathbf{J}_{\mathrm{sym}}) \leq r$ since $\mathrm{rank}(\mathbf{J}_{\mathrm{sym}}) = \mathrm{rank}(\mathbf{J} + \mathbf{J}^{\mathrm{T}}) = \mathrm{rank}(2\mathbf{J}) \leq \min\left(\mathrm{rank}(\mathbf{U}), \mathrm{rank}(\mathbf{V})\right) \leq r$.

From a practical point of view, the bilinear formulation of the symmetry constraints (Equation 5) is potentially problematic since its Jacobian can be rank-deficient and it introduces $D(D-1)/2$ unique constraints which can lead to an overly large number of constraint equations to satisfy. The difficulty in applying constraints with rank-deficient Jacobian is that such constraints can fail to satisfy the Karush-Kuhn-Tucker (KKT) conditions (Prop 1 in the Appendix) when a local minimizer lies on the boundary of the feasible region. Since we require the application of equality constraints to enforce invariance of the rank of **J** to symmetrization, any feasible local minimum lies on the boundary of the feasible region. Consequently, we would like to formulate an optimization problem for low-rank MNE that will generally satisfy the KKT conditions at a local minimizer. A safe choice (see the discussion following Prop 1) is to replace the bilinear formulation with a set of rD linear equality constraints:

$$\mathbf{w}_k = \mathbf{U}_{\bullet,k} + \pi_k \mathbf{V}_{\bullet,k} = \mathbf{A}_{k,k}\mathbf{Q}_{\bullet,k} = \mathbf{0} \text{ for all } k, \tag{6}$$

where $\pi_k \in \{-1, 1\}$ for the kth column of **U** and **V** and

$$\mathbf{A}_{k,k} = \left[\, \mathbf{I},\ \pi_k \mathbf{I} \,\right] \tag{7}$$

is the $D \times 2D$ dimensional Jacobian matrix of $\mathbf{w}_k$ with respect to $\mathbf{Q}_{\bullet,k}$. For a brief summary of alternative constraints that satisfy $\mathrm{rank}(\mathbf{J}_{\mathrm{sym}}) \leq r$, see the Section 5.1 in the Appendix.

Putting this all together, the low-rank MNE method is a nonlinear program of the form:

$$\begin{aligned} \min_{a,\mathbf{h},\mathbf{Q}} f(a, \mathbf{h}, \mathbf{Q}) &= \min_{a,\mathbf{h},\mathbf{Q}} L(a,\ \mathbf{h}, \mathbf{Q}) + \ell_*(\mathbf{Q}) \\ \text{subject to } \mathbf{w}_k &= \mathbf{0} \text{ for all } k. \end{aligned} \tag{8}$$

This problem can be transformed from a constrained to "unconstrained" optimization via the Lagrangian method:

$$\mathcal{L}(a, \mathbf{h}, \mathbf{Q}, \mathbf{\Lambda}) = f(a, \mathbf{h}, \mathbf{Q}) - \sum_{k=1}^{r} \mathbf{\Lambda}_{\bullet,k}^{\mathrm{T}} \mathbf{w}_k \tag{9}$$

where $\mathbf{\Lambda}$ is a $D \times r$ matrix of unconstrained Lagrange multipliers and:

$$f(a, \mathbf{h}, \mathbf{Q}) = L(a, \mathbf{h}, \mathbf{Q}) + \ell_*(\mathbf{Q}). \tag{10}$$

is the objective function. Alternatively, one may directly substitute $\mathbf{V}_{\bullet,k} = -\pi_k \mathbf{U}_{\bullet,k}$ into f for an equivalent unconstrained problem. Once a solution is found to Equation (8), relevant quadratic components of **J** are identified by diagonalizing $\mathbf{J}_{\mathrm{sym}}$:

$$\mathbf{J}_{\mathrm{sym}} = \frac{1}{2}\left(\mathbf{J} + \mathbf{J}^{\mathrm{T}}\right) = \mathbf{\Omega}\mathbf{\Sigma}\mathbf{\Omega}^{\mathrm{T}}, \tag{11}$$

where $\mathbf{\Omega}$ is a $D \times D$ matrix with columns forming an orthonormal basis and $\mathbf{\Sigma}$ is a $D \times D$ diagonal matrix where the $\Sigma_{k,k}$ element corresponds to the variance of the $\mathbf{\Omega}_{\bullet,k}$ basis vector. Those columns of $\mathbf{\Omega}$ with nonzero variance span the subspace of stimulus space relevant to a response (Fitzgerald et al., 2011a). Note that, in theory, a solution to Equation (8) should yield $\mathbf{J}_{\mathrm{sym}} = \mathbf{J}$ with maximum rank r. In practice, this is dependent on the desired precision to which the constraints are satisfied and at what variance the eigenvalues are defined to be approximately zero. If an investigator employs an eigenvalue solver that is more precise than the constraint satisfaction, diagonalizing **J** may result in a complex eigenvalue spectrum with small imaginary parts. Diagonalizing $\mathbf{J}_{\mathrm{sym}}$ instead via Equation (11) eliminates these small imaginary components and will admit at most r eigenvalues with variance above the desired precision.

Unlike the full-rank MNE optimization, the low-rank MNE optimization is a nonconvex problem (see the discussion surrounding Prop 2 in the Appendix). This nonconvexity is caused by the bilinear factorization of **J** in the negative log-likelihood term of the cost function, f. The nuclear-norm penalty, on the other hand, is convex since $\epsilon_k \geq 0$ for all k. Due to this property of the nuclear-norm, it is possible to show that there is a regularization domain where any solution to the low-rank MNE problem is globally optimal (Burer and Monteiro, 2003; Bach et al., 2008; Haeffele et al., 2014). Specifically, if all ϵ_k are greater than or equal to the magnitude of the largest variance eigenvalue of the $D \times D$ gradient matrix $\nabla_{\mathbf{J}} L$ (where $\nabla_{\mathbf{J}}$ is the gradient operator with respect to **J**) evaluated at a solution, then the weights a, **h**, **U**, and **V** are globally optimal solutions of the low-rank MNE problem. Conversely, if any ϵ_k is less than the magnitude of the largest variance eigenvalue of $\nabla_{\mathbf{J}} L$, then a solution to the low-rank MNE problem is not guaranteed to be globally optimal and belongs to the locally optimal domain. For proof of this, see the Sections 5.2 and 5.3 in the Appendix.

When the rank of the ground truth of matrix **J** is low-rank, solutions of the low-rank MNE problem in the globally optimal domain can be a good approximation to the ground truth of **J**. This approximate solution can be attractive due to its certifiable global optimality and can be helpful when D is practically too large to fit with the full-rank MNE method or to find compressed solutions for **J** of rank less than the ground truth. In some cases, however, it is possible that solutions that lie in the locally optimal domain better reconstruct the ground truth of **J** compared to solutions in the globally optimal domain. In the following sections, we detail optimization algorithms that may be used to find solutions in either the locally or globally optimal domains of the low-rank MNE problem.

2.1.2. Optimizing the Weights

To find a feasible local minimizer of the low-rank MNE problem (Equation 8) for given set of nuclear-norm regularization parameters, a line search interior-point method designed to find local minima of nonlinear, nonconvex programming problems based on Ch. 19 of *Numerical Optimization* by Nocedal and Wright (2006) is used. The interior-point method iteratively searches for a local minimum of the low-rank MNE minimization problem (Equation 8) by recursively solving:

$$\underbrace{\begin{bmatrix} \nabla_{\mathbf{xx}}^2 \mathcal{L}, & \mathbf{A}^{\mathrm{T}} \\ \mathbf{A}, & \mathbf{0} \end{bmatrix}}_{\mathcal{H}} \begin{bmatrix} \mathbf{p}_{\mathbf{x}} \\ -\mathbf{p}_{\mathbf{\Lambda}} \end{bmatrix} = - \underbrace{\begin{bmatrix} \nabla_{\mathbf{x}} \mathcal{L} \\ \mathbf{w} \end{bmatrix}}_{\mathrm{KKT}} \tag{12}$$

for the weight and Lagrange multiplier update directions, $\mathbf{p}_{\mathbf{x}}$ and $\mathbf{p}_{\mathbf{\Lambda}}$, respectively, where a weight vector $\mathbf{x}^{\mathrm{T}} = \left[a,\ \mathbf{h}^{\mathrm{T}},\ \mathbf{Q}_{\bullet,1}^{\mathrm{T}}, \cdots, \mathbf{Q}_{\bullet,r}^{\mathrm{T}}\right]$ is defined. The matrix **A** is the full Jacobian matrix of the constraints and **w** is a concatenation of the equality constraints (i.e., $\mathbf{w}^{\mathrm{T}} = \left[\mathbf{w}_1^{\mathrm{T}}, \cdots, \mathbf{w}_r^{\mathrm{T}}\right]$). The matrix labeled $\boldsymbol{\mathcal{H}}$ will be referred to as the constrained Hessian and the vector on the right-hand-side contains the KKT conditions (Prop 1 in the Appendix). For nonconvex problems, it can be useful to employ an optimization method that reduces the chances of converging to a saddle point of f. The implementation of the interior-point algorithm from Nocedal and Wright (2006) has the advantage of circumventing saddle points by adding a $(1 + D + 2rD) \times (1 + D + 2rD)$ positive diagonal shift matrix, $\delta\mathbf{I}$ where $\delta > 0$, to the Hessian of the Lagrangian, $\nabla^2_{\mathbf{xx}}\mathcal{L} + \delta\mathbf{I}$, to maintain proper *matrix inertia* of the constrained Hessian. The matrix inertia is specified by the number of positive eigenvalues, m, the number of negative eigenvalues, n, and the number of eigenvalues equal to zero, l, of the constrained Hessian. To prevent convergence of the interior-point method to a saddle point of f, we maintain a matrix inertia of $m = 1 + D + 2rD$ (the number of rows/columns of $\nabla^2_{\mathbf{xx}}\mathcal{L}$), $n = rD$ (the number of constraints), and $l = 0$. If the constrained Hessian does not meet this condition, the inertia is enforced by adjusting δ until this condition is satisfied.

The trouble with using this interior-point method to solve Equation (8) is that the size of matrix $\boldsymbol{\mathcal{H}}$ is $(1 + D + 3rD) \times (1 + D + 3rD)$ which can be prohibitively large for typical memory constraints and lead to substantial time spent solving the linear system (Equation 12). Some alternative approaches are to use quasi-Newton methods such as the Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm (Nocedal and Wright, 2006) or gradient-only heuristics like stochastic gradient descent (Bottou, 2010). Another option is to divide the weights into blocks and perform block coordinate descent using constrained block Hessians (Wright, 2015). We chose the latter to better exploit the structure of the regularization function (Equation 4).

The block coordinate descent algorithm cyclically solves the subproblems:

$$\text{block } k \text{ subproblem: } \begin{cases} \displaystyle\min_{a,\mathbf{h},\mathbf{Q}_{\bullet,k}} f(a,\mathbf{h},\mathbf{Q}) \\ \qquad\qquad \text{subject to } \mathbf{w}_k = \mathbf{A}_{k,k}\mathbf{Q}_{\bullet,k} = \mathbf{0} \end{cases} \tag{13}$$

until the KKT conditions (Prop 1 in the Appendix) and second-order sufficient conditions (Prop 2 in the Appendix) are satisfied. The block coordinate descent is performed by cyclically minimizing the cost function with respect to the kth block of weights $\mathbf{x}_k^{\mathrm{T}} = \left[a, \mathbf{h}^{\mathrm{T}}, \mathbf{Q}_{\bullet,k}^{\mathrm{T}}\right]$ using the interior-point algorithm described above to recursively solve:

$$\underbrace{\begin{bmatrix} \nabla^2_{\mathbf{x}_k\mathbf{x}_k}\mathcal{L}, & \mathbf{A}^{(k)\mathrm{T}} \\ \mathbf{A}^{(k)}, & \mathbf{0} \end{bmatrix}}_{\boldsymbol{\mathcal{H}}_k} \begin{bmatrix} \mathbf{p}_{\mathbf{x}_k} \\ -\mathbf{p}_{\boldsymbol{\Lambda}_k} \end{bmatrix} = - \begin{bmatrix} \nabla_{\mathbf{x}_k}\boldsymbol{\mathcal{L}} \\ \mathbf{w}_k \end{bmatrix} \tag{14}$$

while holding the remaining $\mathbf{Q}_{\bullet,j}$ $(j \neq k)$ fixed. The new indexing on the Jacobian $\mathbf{A}^{(k)\mathrm{T}} = \nabla_{\mathbf{x}_k}\mathbf{w}_k^{\mathrm{T}}$ is the Jacobian of the kth block constraints and is a $D \times (1 + 3D)$ matrix. Proof that the block coordinate descent algorithm converges to a feasible local minimizer of the low-rank MNE problem (Equation 8) appears in Section 5.4 of the Appendix.

2.1.3. Hyperparameter Optimization

Now we turn to the procedure for setting the nuclear-norm regularization parameters. In the globally optimal domain (Prop 4 in the Appendix), the goal is to use nuclear-norm regularization to find a globally optimal solution that approximates a solution to the unregularized problem where all $\epsilon_k = 0$. Therefore, it makes sense to make the regularized and unregularized problems as similar as possible by using the minimal amount of regularization necessary to reach the globally optimal domain. To do so, one can optimize each block of the block coordinate descent such that ϵ_k is approximately equal to the magnitude of the largest variance eigenvalue of $\nabla_{\mathbf{J}}L$, which will be defined as λ_L. A simple algorithm for achieving this is: (i) optimize $\mathbf{x}_k$, then (ii) increase ϵ_k if $\epsilon_k < \lambda_L$ or decrease ϵ_k if $\epsilon_k > \lambda_L$, and then repeat steps i and ii until $\epsilon_k \approx \lambda_L$ for each block (see **Algorithm 1** for a pseudocode implementation). By contrast, in the locally optimal regularization domain we instead adjust ϵ_k to find the model that best generalizes to novel data in a cross-validation set. This approach to hyperparameter optimization is in common use in modern machine learning applications (Bergstra et al., 2011; Bergstra and Bengio, 2012).

Our approach to the hyperparameter optimization in the locally optimal domain exploits the structure of the block coordinate descent subproblems (Equation 13) where the gradient and Hessian of the block k subproblem only depends explicitly on ϵ_k. Holding the remaining $\mathbf{Q}_{\bullet,j}$ $(j \neq k)$ fixed, the kth block is optimized while varying ϵ_k via a grid search on the domain $\epsilon_k \in [0, \epsilon_{\max}]$ where $\epsilon_{\max}$ is chosen to be large enough such that $\mathbf{Q}_{\bullet,k} \approx \mathbf{0}$ when $\epsilon_k = \epsilon_{\max}$. We can estimate the generalization ability of the ith solution $\mathbf{x}^{*(i)}$ for a chosen value of the ϵ_k parameter, $\epsilon_{k_i} \in [0, \epsilon_{\max}]$, by evaluating the negative log-likelihood $L_{\mathrm{CV}}(\mathbf{x}^{*(i)})$ where y_t and $\mathbf{s}_t$ are now samples drawn from the cross-validation set. If $L_{\mathrm{CV}}(\mathbf{x}^{*(i)}) \leq L_{\mathrm{CV}}(\mathbf{x}^{*(j)})$ for all $\epsilon_{k_j} \in [0, \epsilon_{\max}]$ of the block k subproblem (Equation 13), then $\mathbf{x}^{*(i)}$ is taken to be the most generalizeable estimate of the weights for the block k subproblem. The optimization completes when several full cycles through all r blocks of the block coordinate descent algorithm fail to provide a further decrease in $L_{\mathrm{CV}}(\mathbf{x})$. For a pseudocode implementation, see **Algorithm 2**.

We tested both of these algorithms and found the locally optimal domain to be most appropriate for recovering receptive field components. In the applications of low-rank MNE to model neurons and avian auditory neurons (for details about the data, see the methods section), we found the minimum amount of regularization necessary to reach the globally optimal domain was unreasonably large ($\epsilon_k \sim 1$ or more). These large regularization parameters were found to severely attenuate the variance of the recovered components (i.e., the eigenvalues of $\mathbf{J}$). For instance, the variance of the components of $\mathbf{J}$ reconstructed from the model neuron data was two orders of magnitude lower than the ground truth. This attenuation was accompanied by substantial distortion of the components. Similarly, solutions that were found in the locally optimal domain had much better generalization ability across both the model neurons and the avian neurons as measured by evaluating the negative log-likelihood on the cross-validation sets.

Algorithm 1 Low-rank MNE block coordinate descent algorithm (globally optimal domain)

```
inputs: maximum rank $r$, paired data samples $(\mathbf{s}_t, y_t)$ for all $t$, initial guess for weights $a$, $\mathbf{h}$, $\mathbf{U}$, and $\mathbf{V}$, set $\pi_k$ for all $k = 1, \cdots, r$, maximum number of iterations $M_{\max}$, regularization parameter precision $\delta_\epsilon$, convergence precision $\delta_x$
initialization: $\mathbf{J} \leftarrow \mathbf{U}\mathbf{V}^{\mathrm{T}}$

for $m \leftarrow 1, \cdots, M_{\max}$ do
    for $k \leftarrow 1, \cdots, r$ do
        $a' \leftarrow a, \mathbf{h}' \leftarrow \mathbf{h}, \mathbf{U}' \leftarrow \mathbf{U}, \mathbf{V}' \leftarrow \mathbf{V}$
        $\mathbf{J} \leftarrow \mathbf{J} - \mathbf{U}'_{\bullet,k}\mathbf{V}'^{\mathrm{T}}_{\bullet,k}$        ▷ remove block $k$ from $\mathbf{J}$
        $\lambda_L \leftarrow \max\left(|\lambda_{\max}\left(\nabla_{\mathbf{J}}L\right)|, |\lambda_{\min}\left(\nabla_{\mathbf{J}}L\right)|\right)$ evaluated with primed variables and $\mathbf{J}$
        do
            $\epsilon_k \leftarrow \lambda_L$
            $a', \mathbf{h}', \mathbf{U}'_{\bullet,k}, \mathbf{V}'_{\bullet,k} \leftarrow$ Solve the block $k$ subproblem (Equation 13) using an
                interior-point method algorithm with inputs $a', \mathbf{h}', \mathbf{U}'_{\bullet,k}, \mathbf{V}'_{\bullet,k}, \mathbf{J},$
                $\epsilon_k, \pi_k, (\mathbf{s}_t, y_t) : \forall t$
            $\lambda_L \leftarrow \max\left(|\lambda_{\max}\left(\nabla_{\mathbf{J}}L\right)|, |\lambda_{\min}\left(\nabla_{\mathbf{J}}L\right)|\right)$        ▷ update eigenvalue threshold
        while $\epsilon_k \notin [\lambda_L, \lambda_L + \delta_\epsilon]$
        $\mathbf{J} \leftarrow \mathbf{J} + \mathbf{U}'_{\bullet,k}\mathbf{V}'^{\mathrm{T}}_{\bullet,k}$        ▷ include $k$th block solution in $\mathbf{J}$
    $a \leftarrow a', \mathbf{h} \leftarrow \mathbf{h}', \mathbf{U} \leftarrow \mathbf{U}', \mathbf{V} \leftarrow \mathbf{V}'$
    if $\left\|\mathbf{x} - \mathbf{x}'\right\|_2 \leq \delta_x$ and $\{\epsilon_k : \epsilon_k \notin [\lambda_L, \lambda_L + \delta_\epsilon], \forall k\} = \varnothing$ then
        (where $\mathbf{x} = \left[a, \mathbf{h}^{\mathrm{T}}, \mathbf{Q}^{\mathrm{T}}_{\bullet,1}, \cdots, \mathbf{Q}^{\mathrm{T}}_{\bullet,r}\right]$ and $\mathbf{x}'$ is the analogous vector for primed weights)
        break        ▷ optimization has finished

outputs: $a$, $\mathbf{h}$, $\mathbf{J}$
```

The global optimization procedure outlined in **Algorithm 1** runs very quickly, usually finding a solution within 1–4 hours for problems of size $D = 400$ to $1,200$ and $r = 1$ to $r = 20$ (see Section 4.8 for hardware/software details). The local optimization procedure in **Algorithm 2**, on the other hand, can range from on the order of less than an hour to a day for problems of size $D = 400$ to $1,200$ and $r = 1$ to $r = 20$. It should be said, however, that the goal of these algorithms are to find good solutions to Equation (8) but we made little attempt to optimize these algorithms for speed. There are two primary bottlenecks in the optimization: (i) the choice of subproblem solver and (ii) the choice of hyperparameters to use in the optimization. Since the optimization procedure is highly customizeable, the timing of these bottlenecks will be highly variable on the choices made by the investigator. For instance, solving the block subproblem may be sped-up by using L-BFGS instead of the exact Hessian on the larger D problems. Furthermore, there are other approaches that may be taken in place of **Algorithm 2** to choose hyperparameters. In particular, one can replace the blockwise grid search with a random search for the hyperparameter settings (Bergstra and Bengio, 2012) or use Bayesian optimization (Brochu et al., 2010; Snoek et al., 2012). We performed some preliminary analysis using Bayesian optimization and found it to be a competitive alternative to **Algorithm 2** that may speed up the optimization for large D.

2.1.4. Rank Optimization

Depending on the application, there are a several possible ways to choose the rank of $\mathbf{J}$ in the low-rank MNE model. For instance, one may intend to find the optimal rank of $\mathbf{J}$, r_{opt}, defined as the rank of $\mathbf{J}$ of the model that has the best generalization performance to novel data. In this instance, an unregularized model would be fit by trying different signs, π_k, for the constraints and maximum rank r and then choose the r_{opt} model as that which makes the best predictions on novel data. For a nuclear-norm regularized model, the fit is more flexible since r can instead be treated as an upper bound on the rank of $\mathbf{J}$ while the regularization can be used to lower the rank, if necessary. In this case, r_{opt} can instead be determined by finding some model of maximum rank r where $\mathbf{J}$ is rank-deficient with respect to at least one $\pi_k = 1$ and $\pi_k = -1$ constraint as determined by the number of negative and positive eigenvalues of $\mathbf{J}$. The justification for this approach is that if the regularization procedure leads to $\mathbf{U}_{\bullet,r_{\mathrm{opt}}+1} = \mathbf{V}_{\bullet,r_{\mathrm{opt}}+1} = \mathbf{0}$ for trials with both signs of $\pi_k = \pm 1$, the value of the cost function (f) is left unchanged from the r_{opt} model. Adding additional columns in $\mathbf{U}$ and $\mathbf{V}$ beyond $r_{\mathrm{opt}} + 1$ would be equivalent to optimizing the $r_{\mathrm{opt}} + 1$ model. Procedurally, one can guess r that is ostensibly an upper bound on r_{opt} and if there is at least one vector $\mathbf{Q}_{\bullet,i} = \mathbf{0}$ for $\pi_i = -1$ and at least one vector $\mathbf{Q}_{\bullet,j} = \mathbf{0}$ for $\pi_j = 1$ that is zero in $\mathbf{Q}$, then $r_{\mathrm{opt}} = \mathrm{rank}(\mathbf{Q})$. This is equivalent to splitting $\mathbf{J}$ into a sum of a positive semidefinite and negative semidefinite matrix, $\mathbf{J} = \mathbf{J}_{\mathrm{psd}} + \mathbf{J}_{\mathrm{nsd}}$, where the optimal rank would be $\mathrm{rank}(\mathbf{J})$ when both $\mathbf{J}_{\mathrm{psd}}$ and $\mathbf{J}_{\mathrm{nsd}}$ are composed of rank-deficient bilinear factorization matrices (i.e., $\mathrm{rank}(\mathbf{Q}_{\mathrm{psd}}) < r_{\mathrm{psd}}$ and $\mathrm{rank}(\mathbf{Q}_{\mathrm{nsd}}) < r_{\mathrm{nsd}}$). If this condition is not met, however, the maximum rank, r, must be increased and the optimization must continue with extra columns appended to $\mathbf{Q}$ and each π_k set appropriately until this condition is met. Since we are looking for the optimal rank in our applications, we use this procedure for determining the rank.

Algorithm 2 Low-rank MNE block coordinate descent algorithm (locally optimal domain)

```
inputs: maximum rank $r$, maximum range for regularization parameters $\epsilon_{\max}$, number of regularization parameter grid points $n_{\text{grid}}$, training set indices $T_{\text{train}} \subseteq \{1, \cdots, N\}$ and cross-validation set indices $T_{\text{CV}} \subset \{1, \cdots, N\}$ where $T_{\text{train}} \cap T_{\text{CV}} = \varnothing$, paired data samples $(\mathbf{s}_t, y_t)$ for all $t \in T_{\text{train}} \cup T_{\text{CV}}$, initial guess for weights $a$, $\mathbf{h}$, $\mathbf{U}$, and $\mathbf{V}$, set $\pi_k$ for all $k = 1, \cdots, r$, maximum iterations $M_{\max}$, convergence precision $\delta_p$, maximum failures to find a better solution $\sigma_{\max}$
initialization: $\mathbf{J} \leftarrow \mathbf{U}\mathbf{V}^{\text{T}}$, $L_{\text{best}} \leftarrow L(a, \mathbf{h}, \mathbf{U}, \mathbf{V})|_{T_{\text{CV}}}$ (evaluated over data indices $t \in T_{\text{CV}}$), regularization grid resolution $\delta_\epsilon \leftarrow \epsilon_{\max}/n_{\text{grid}}$, early completion switch $\sigma \leftarrow 1$

for $m \leftarrow 1, \cdots, M_{\max}$ do
    for $k \leftarrow 1, \cdots, r$ do
        $a' \leftarrow a, \mathbf{h}' \leftarrow \mathbf{h}, \mathbf{u}' \leftarrow \mathbf{U}_{\bullet,k}, \mathbf{v}' \leftarrow \mathbf{V}_{\bullet,k}$
        $\mathbf{J} \leftarrow \mathbf{J} - \mathbf{u}'\mathbf{v}'^{\text{T}}$                                  ▷ remove block $k$ from $\mathbf{J}$
        for $n \leftarrow 0, \cdots, n_{\text{grid}}$ do
            $\epsilon_k \leftarrow n\delta_\epsilon$
            $a', \mathbf{h}', \mathbf{u}', \mathbf{v}' \leftarrow$ Solve the block $k$ subproblem (Equation 13) using an
                interior-point method algorithm with inputs $a', \mathbf{h}', \mathbf{u}', \mathbf{v}', \mathbf{J}$,
                $\epsilon_k, \pi_k, (\mathbf{s}_t, y_t) : \forall t \in T_{\text{train}}$
            $L' = L(a', \mathbf{h}', \mathbf{J} + \mathbf{u}'\mathbf{v}'^{\text{T}})|_{T_{\text{CV}}}$
            if $L' < L_{\text{best}} - \delta_p$ then
                $L_{\text{best}} \leftarrow L'$
                $a \leftarrow a', \mathbf{h} \leftarrow \mathbf{h}', \mathbf{U}_{\bullet,k} \leftarrow \mathbf{u}', \mathbf{V}_{\bullet,k} \leftarrow \mathbf{v}'$
                $\sigma \leftarrow 0$
            else if $L' \leq L(a, \mathbf{h}, \mathbf{U}, \mathbf{V})|_{T_{\text{CV}}}$ then
                $a \leftarrow a', \mathbf{h} \leftarrow \mathbf{h}', \mathbf{U}_{\bullet,k} \leftarrow \mathbf{u}', \mathbf{V}_{\bullet,k} \leftarrow \mathbf{v}'$ (or skip for monotonic convergence)
        $\mathbf{J} \leftarrow \mathbf{J} + \mathbf{U}_{\bullet,k}\mathbf{V}_{\bullet,k}^{\text{T}}$                          ▷ include block $k$'s solution in $\mathbf{J}$
    if $\sigma = \sigma_{\max}$ then
        break                                                   ▷ optimization has finished
    $\sigma \leftarrow \sigma + 1$

outputs: $a$, $\mathbf{h}$, $\mathbf{J}$
```

We initialize the π_k parameters such that $\mathbf{J}_{\text{psd}}$ and $\mathbf{J}_{\text{nsd}}$ have equal maximum rank.

If instead one intends to find a compressed representation of $\mathbf{J}$ where $r < r_{\text{opt}}$, the only remaining unset parameters are π_k. This can be done by solving the problems with different choices of π_k and keeping the model that fits the best either to the training or cross-validation sets. Instead of solving models that enumerate all possible choices of π_k for all $k = 1 \cdots r$, one can instead take a shortcut by solving lower-rank models of rank r_n, incrementing the rank of the model (e.g., $r_{n+1} = r_n + 1$), enumerating solutions with π_k fixed for all $k \leq r_n$, and repeating until reaching a rank r model. Alternatively, one can also use other principled means for choosing π_k including the eigenvalues of $\mathbf{J}$ from the full-rank MNE model or from an unconstrainted low-rank MNE model. One may also attempt to do away with the π_k parameters entirely by using one of the alternative constraint formulations (Section 5.1 in the Appendix).

2.2. Testing the Algorithm on Model Neurons

We now illustrate the proposed method by analyzing responses of model neurons. Details about the model data may be found in Section 4.4 in methods. First, we tested the method on two model neurons with different signal-to-noise ratios (SNR) (cf. **Figure 1**). Both the low-rank and full-rank approaches yielded good reconstructions in the high SNR regime, finding all of the four relevant components of the model. The subspace overlap (Equation 22 in methods) between the set of model and reconstructed dimensions was 0.933 ± 0.007 and 0.909 ± 0.008 for the low and full-rank approaches, respectively. The STC method that is standard for noise-like stimuli (Schwartz et al., 2006) performs worse here, because it is not designed to work with stimuli drawn from correlated distributions, with subspace overlap of 0.32 ± 0.05. We note that although the low-rank and full-rank approaches recover the component subspace with reasonable accuracy, the low-rank models produce much more accurate predictions on the test sets (0.233 ± 0.009 vs. 0.32 ± 0.02 for the negative log-likelihood of low-rank and full-rank models, respectively). The main advantage of the low-rank approach becomes apparent in the ultra-low SNR regime. Here, the full-rank model failed to recover all of the relevant components finding only two out of four with a subspace overlap of 0.17 ± 0.05. In contrast, the low-rank model correctly determined the number of relevant components with a subspace overlap of 0.83 ± 0.02. This is much better than the STC method where the subspace overlap was 0.30 ± 0.06.

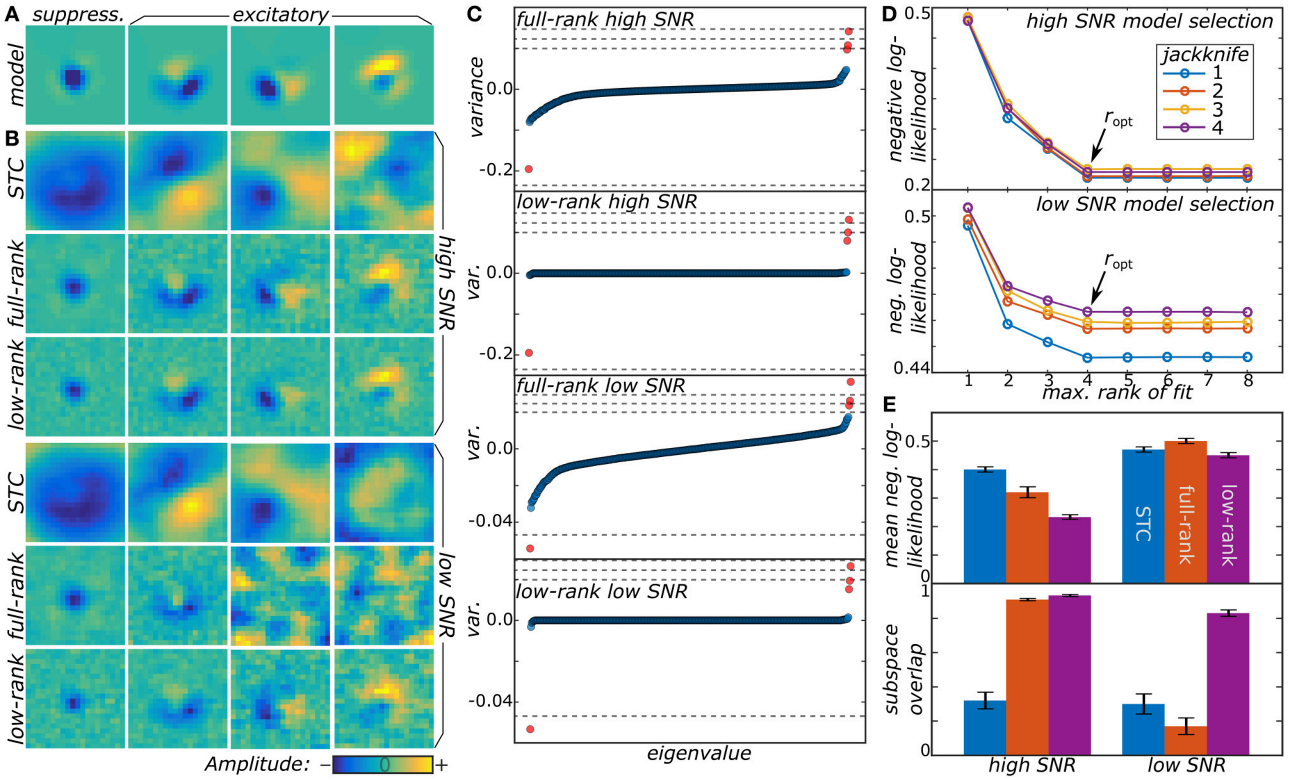

FIGURE 1 | (A) Two model neurons were generated from a synthetic receptive field, one with high SNR and another with low SNR. **(B)** Low-rank MNE, full-rank MNE, and STC models were optimized for both of the neurons and the mean top four largest magnitude components of **J** were plotted. **(C)** These mean components correspond to the largest variance eigenvalues in the eigenvalue spectra. The dashed lines in the eigenvalue spectra correspond to the eigenvalues of the ground truth, $\mathbf{J}_{\text{GT}}$. **(D)** Low-rank models with maximum rank ranging over $r = 1 \cdots 8$ were trained on four different jackknives of the data set where for each jackknife the data set was split into a training and cross-validation set. The predictive power of each jackknife's trained models evaluated on its cross-validation set is shown to saturate when $r \geq r_{\text{opt}} = 4$. **(E)** Quantitative comparisons of the receptive field reconstructions from the three methods is compared based on each model's predictive power on the test sets (top) and the subspace overlap onto the ground truth (bottom).

From a qualitative point of view, the low-rank model performs better than the full-rank model because the matrix **J** is less corrupted by noise present in the data. This can be seen by looking at the eigenvalue spectra of **J** in **Figure 1C** where the full-rank models recover a nearly full-rank **J** matrix dominated by fictitious components. The large number of fictitious components contribute substantially to the variance of **J** leading to an overall decrease in the predictive power of the model. The significance of these fictitious components becomes even more substantial in the low SNR regime where their eigenvalues nearly engulf the eigenvalues of the relevant components. By contrast, the low-rank models exhibit a sparse eigenvalue spectrum of rank consistent with the ground truth in **Figure 1A**.

We also found fits of the low-rank model to be resilient even when the rank of **U** and **V** was larger than the ground truth. For example, in **Figure 1D**, we show the results for fitting low-rank models with rank $r = 1, \cdots, 8$ (using signs of the eigenvalues of the mean **J** matrix from the full-rank models to initialize the π_k values). Here, the negative log-likelihood evaluated on the cross-validation set saturates as r becomes greater than or equal to the ground truth value of 4. Above $r = 4$, the regularization procedure eliminates the fictitious dimensions that infected the full-rank models leading to a rank-deficient solution equivalent to the $r = 4$ model. On the other hand, the models with $r < 4$ have a higher negative log-likelihood because, by design, they cannot recover all four components and represent a low-rank compression of **J**.

2.3. Application to Avian Auditory Data

We now show that the proposed low-rank MNE method offers substantial improvement in our ability to resolve multiple relevant components of sensory neurons' receptive fields by applying it to recordings from the avian auditory forebrain (Gill et al., 2006; Amin et al., 2010). For details about these recordings and data processing, see Section 4.5 in methods. First, the low-rank method produces much sharper components that are more localized in both frequency and time compared to components of the full-rank estimation (**Figures 2A,B**). The improvement over the STC components is even more dramatic (**Figures 2A,B**). This difference becomes more pronounced for components that account for lower variance in the neural response. For such components, the low-rank method can resolve localized regions of sensitivity under the broad bands

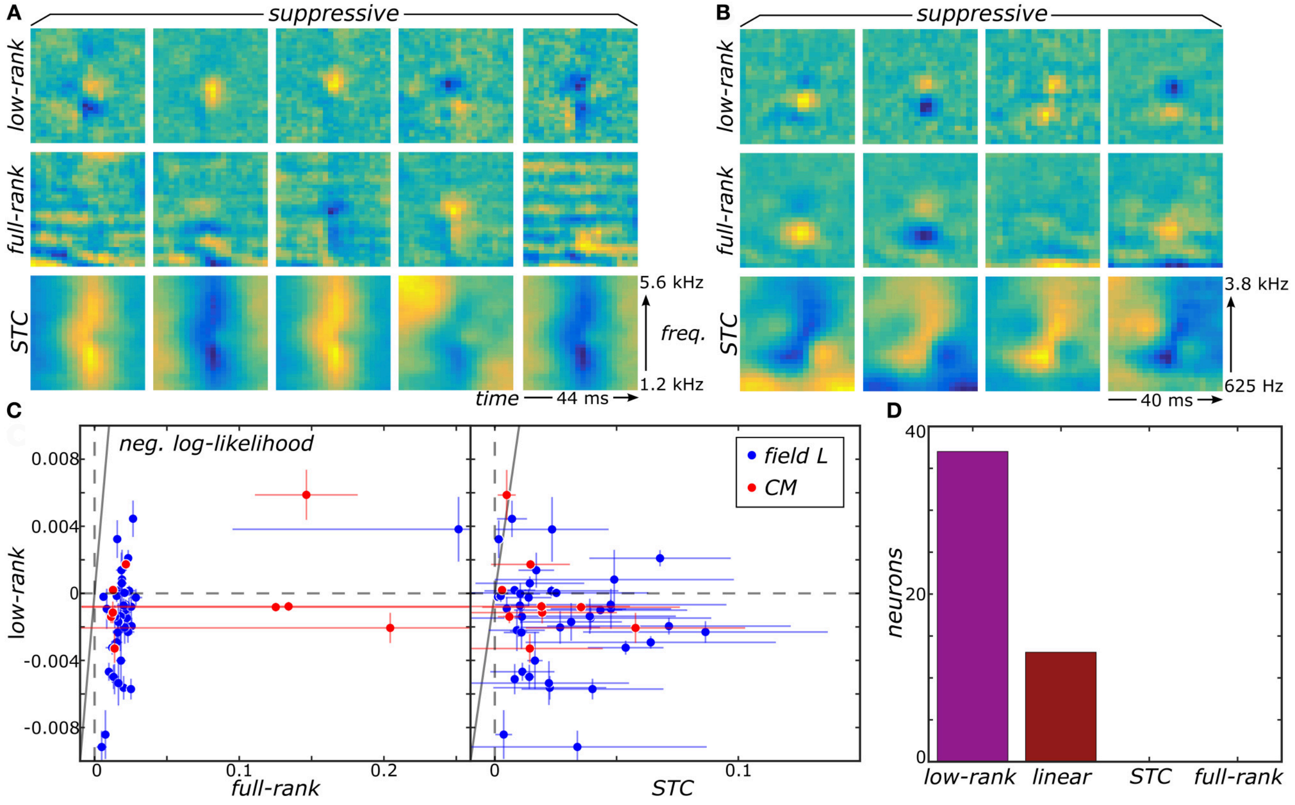

FIGURE 2 | Low-rank MNE, full-rank MNE, and STC models were optimized on a dataset of 50 avian auditory forebrain neurons. Logical OR and logical AND FB models were fit using linear combinations of the subspace components of each method. Logical AND FB models from two example neurons are shown **(A,B)**. The quality of each model is measured using the difference between the mean negative log-likelihood of the model and the linear MNE model evaluated on the test sets and plots summarize predictive ability across the population of neurons **(C)**. A bar plot quantifies the number of neurons in the population best fit by each model **(D)**. Note low-rank and linear MNE models outperform all STC and full-rank MNE models across the population.

that dominate in the full-rank method, e.g., for components in columns 2–4 in **Figure 2A**. Quantitatively, reconstructions of neural responses obtained with low-rank models yield universally higher predictive power on novel data subsets compared to the full-rank and STC models (**Figure 2C**). Importantly, the full-rank models did not yield better predictions over the linear one-component MNE models ($\mathbf{J} = \mathbf{0}$) for all neurons. Therefore, the additional variables in the full-rank model do not yield any statistically significant components because the full-rank model does not improve predictions on the test sets relative to the linear model. This is despite the leading components of the full-rank reconstructions bearing apparent similarity to the top components of the low-rank reconstruction. STC models made worse predictions than the linear MNE models across all neurons as well. By comparison, the low-rank optimization yielded better predictions on the test sets compared to the linear model for 37 of the 50 neurons (**Figure 2D**).

The ultimate utility of methods for receptive field reconstruction is to produce models that can inform our understanding of the transformations performed by high-level sensory neurons. Toward that goal, one can subject the components obtained from low-rank reconstructions to a functional basis (FB) transformation (Kaardal et al., 2013). This transformation aims to account for the observed neural responses in terms of logical operations, such as logical AND/OR, on the set of input components. By studying whether populations of neurons are best fit by logical AND/OR functions, we can learn whether populations of neurons in certain regions of the brain compute primarily conjunctive or integrative functions of their inputs. A conjunctive neuron would be selective toward coincidences of multiple relevant inputs and corresponds to a logical AND function. An integrative neuron is responsive toward any relevant input and corresponds to a logical OR function. Here we find that FB models based on logical AND combinations overwhelmingly outperformed models based on logical OR across the population where 40 of the 41 field L neurons and 8 of the 9 CM neurons were best fit by logical AND models (**Figure 3**). To gain intuitive understanding for these results, we note that a logical AND operation is equivalent to a logical OR followed by negation. These results therefore suggest that logical AND better represents cases where invariances are built into suppressive receptive field components. This is because logical OR combinations often work well to approximate invariance in neural responses that occur if any relevant stimulus features are present, corresponding to the logical OR operation

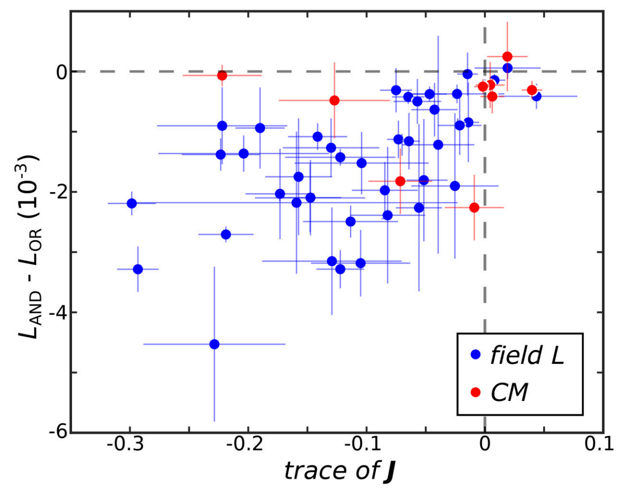

FIGURE 3 | The difference between the negative log-likelihood of the best logical AND (L_{AND}) and logical OR (L_{OR}) models averaged across test sets is plotted against $\mathrm{Tr}(\mathbf{J})$ where here **J** is averaged across jackknives. The horizontal dashed line demarcates neurons best fit by logical OR models on top from neurons best fit by logical AND models below. The vertical dashed line separates neurons where the eigenvalue spectra of mean **J** are dominated by negative variance on the left and positive variance on the right.

(i.e., if $\mathbf{v}_k \cdot \mathbf{s}_t$ for *any* k is greater than some threshold, where $\mathbf{v}_k$ is a receptive field component, the neuron spikes in response to sample t). When these components are all suppressive, this implies that the neural response occurs if none of the relevant features are present in the stimulus (i.e., if $\mathbf{v}_k \cdot \mathbf{s}_t$ for *any* k is greater than some threshold, the neuron is silent at sample t). This would correspond to the logical AND model. Supporting these arguments, we found neurons with stronger suppressive components were better described by logical AND models over logical OR models (**Figure 3**) with a t-test p-value of 0.1%.

3. DISCUSSION

By using low-rank MNE models that are resistant to both overfitting and the biases of naturalistic stimuli, we have estimated multiple components relevant to responses of neurons from the avian auditory forebrain with much greater accuracy than by prior methods. Interestingly, we found that receptive fields of neurons from field L and CM, relatively high-level regions of the avian auditory forebrain, were composed of few components ($r \leq 20$). This number is small enough for the resultant models to provide interpretable representations of the underlying receptive fields.

We demonstrated that the low-rank MNE models produced better predictive models than full-rank MNE, STC, and linear MNE models and did so with a fewer components than the full-rank MNE models across the population of neurons. There are several reasons why this improvement is observed. As mentioned before, MNE models in principle produce better reconstructions of the relevant components than STC for stimuli drawn from distributions other than Gaussian white noise. This was demonstrated in practice where we saw the STC models performing worse than the low-rank and linear MNE models on both model neurons and recordings from auditory neurons subject to correlated stimuli. With regard to the full-rank MNE models, the low-rank MNE models have three major advantages: (i) the number of components necessary to optimize is explicitly reduced, (ii) the optimization procedure uses nuclear-norm regularization to eliminate fictitious components, and (iii) significant components are recovered via a nonlinear matrix factorization. The first reason is simply a matter of reducing overfitting since fewer weights were estimated in the low-rank MNE models than the full-rank MNE models. To the second point, both the full-rank and low-rank MNE methods used a form of regularization as an attempt to reduce this overfitting but the early exiting procedure used by the full-rank MNE method (Fitzgerald et al., 2011a) did not impose defined structure on **J** while the low-rank MNE regularization procedure directly eliminated components from **J**. Lastly, while it may be approximately true in many cases, one cannot generally assume that the size of the contribution of each component of **J** toward the predictive power of an MNE model will correspond to the variance of the component. In fact, we observed from randomly generated low-rank MNE problems with suboptimal local minima that an optimal component corresponding to highest variance was not necessarily the component that led to the best fit. This seems likely to be an important consideration for other nonlinear matrix factorization methods as well. In contrast, linear matrix factorizations like in the STC method where each component is a local minimum of the matrix factorization has a direct correspondence between the variance of the component and quality of the low-rank fit.

Ultimately, since the optimization of full-rank MNE models is convex, the weights that minimize the negative log-likelihood represent the ground truth as captured by the training data. Thus, if the full-rank MNE model's representation of **J** is high or even full-rank, the global solution of the low-rank MNE optimization problem in the absence of regularization would also be high or full-rank. In such cases, including our applications, it is unsurprising that the locally optimal domain would produce solutions that generalize significantly better than the globally optimal domain for low-rank representations of **J**. This is because the solutions in the globally optimal domain must have large enough regularization parameters to eliminate all possible higher-rank solutions; a requirement that is relaxed in the locally optimal domain.

Overall, the efficiency of the low-rank optimization for the extraction of multiple input components makes it possible to begin to resolve a long-standing puzzle of how high-level auditory responses can combine selectivity to sharp transients with integration over broad temporal components. We find that it is possible to reconstruct many more components more accurately than was possible before. The results highlight an interesting potential difference between high-level visual and auditory responses. In vision, current studies (Serre et al., 2007; Kaardal et al., 2013) show that logical OR models are better at describing the neural computations while this initial analysis suggests that logical AND operations are better at explaining responses in the avian auditory forebrain. It is worth noting that both logical OR and logical AND models could indicate the presence of invariance to certain stimulus transformations. The difference is that logical OR models would capture invariance constructed by max pooling responses among excitatory dimensions whereas logical AND would capture invariance constructed by max pooling among suppressive dimensions. Here, pooling is used as an approximation to logical OR; when applied to suppressive dimensions it converts to a logical AND operation because of negation (that is, in the logical AND model the response is observed if the stimulus has no features that simultaneously strongly project onto any of the receptive field components). Thus, one arrives at a potential important difference between visual and auditory processing. In the visual system (Serre et al., 2007), invariance is achieved by max pooling across primary excitatory dimensions whereas in the auditory system invariance is achieved by suppression.

4. METHODS

4.1. The First-Order and Full-Rank MNE Methods

The full-rank MNE method (Fitzgerald et al., 2011a) solves the convex, nonlinear program:

$$\min_{a,\mathbf{h},\mathbf{J}} L(a, \mathbf{h}, \mathbf{J}) \tag{15}$$

where L is the negative log-likelihood from before (Equation 2) but with $\mathbf{J}$ directly optimized and without explicit regularization. Since the full-rank MNE problem is convex, we find a global optimum of L via conjugate gradient descent. As a mild form of regularization, early stopping is used where the performance of the model after each conjugate gradient descent step is measured on a cross-validation set and the algorithm returns the weights a, $\mathbf{h}$, and $\mathbf{J}$ with minimal negative log-likelihood evaluated on the cross-validation set. The early stopping criterion is to halt optimization after 40 consecutive iterations of the conjugate gradient descent algorithm fail to decrease the negative log-likelihood evaluated on the cross-validation set. The quadratic weights that form a subspace relevant to the neural response are extracted via eigendecomposition in the same way as the low-rank method (Equation 11).

First-order MNE models solve (Equation 15) with $\mathbf{J}$ fixed to zero. The receptive field is approximated entirely by the linear weights, $\mathbf{h}$. Since the first-order MNE method is also convex, it is fit using conjugate gradient descent with early stopping using the exact same procedure as the full-rank MNE method.

4.2. Spike-Triggered Average (STA) and Spike-Triggered Covariance (STC) Methods

The STA and STC methods are standard methods for analyzing receptive fields of neurons stimulated by Gaussian white noise stimuli (Schwartz et al., 2006). To calculate the STA, the difference between a spike-weighted average of zero-centered stimuli is computed:

$$\mathbf{h}_{\mathrm{STA}} = \frac{1}{N_{\mathrm{spk}}} \sum_{t=1}^{N} y_t \mathbf{s}_t, \tag{16}$$

where $\mathbf{h}_{\mathrm{STA}}$ is a single component estimate of the receptive field. Note that the STA can only compute one component. STC, on the other hand, can estimate multiple components of the receptive field. For STC, the difference of two covariance matrices is calculated:

$$\mathbf{C} = \frac{1}{N_{\mathrm{spk}}} \sum_{t=1}^{N} y_t \mathbf{s}_t \mathbf{s}_t^{\mathrm{T}} - \frac{1}{N} \sum_{t=1}^{N} \mathbf{s}_t \mathbf{s}_t^{\mathrm{T}}, \tag{17}$$

one the spike-weighted mean stimulus covariance and the other the mean stimulus covariance. As with the STA, the stimuli are zero-centered. The matrix $\mathbf{C}$ is diagonalized:

$$\mathbf{C} = \mathbf{\Omega}\mathbf{\Sigma}\mathbf{\Omega}^{\mathrm{T}} \tag{18}$$

and the relevant components are spanned by the eigenvectors corresponding to the largest variance eigenvalues. Determining where to cut-off the eigenvalue spectrum is done by randomly shuffling the responses in the training set to break correlations between the stimuli and responses (Bialek and de Ruyter van Steveninck, 2005; Rust et al., 2005; Schwartz et al., 2006; Oliver and Gallant, 2010) and then generating randomized STC matrices, $\mathbf{C}_{\mathrm{rand}}$, from Equation (17). All eigenvalues of $\mathbf{C}$ with larger variance than the largest variance eigenvalue of $\mathbf{C}_{\mathrm{rand}}$ are considered significant and form the estimate of the relevant subspace.

Since STC is not equipped with a nonlinearity, we optimize a full-rank MNE model with all $\mathbf{s}_t$ projected into the relevant components from above and use the resulting weights to estimate the predictive power of the model on the test set. If $\mathbf{\Omega}_r$ is the rank r STC basis, then the stimulus space is transformed into the reduced stimulus space, $\mathbf{s}_t^{(\mathrm{red})} = \mathbf{\Omega}_r^{\mathrm{T}} \mathbf{s}_t$, and then the full-rank MNE problem (Equation 15) is minimized on the training set projected into this reduced stimulus space.

To contend with the strong correlations present in the data sets, we repeated the above STC analysis with stimulus correlations removed through data whitening. Data whitening removes the mean correlations between elements of the stimulus space such that the mean covariance of the stimulus samples is the identity matrix. This can be a beneficial pre-processing step for STC models since STC models are biased when the stimulus space is not Gaussian white noise distributed. There are standard ways of whitening data known as principal component analysis (PCA) and zero-phase (ZCA) whitening (Bell and Sejnowski, 1997), both of which we implemented. In both cases, the first step is to take the singular-value decomposition of the mean centered (mean stimulus vector subtracted) stimulus covariance matrix:

$$\frac{1}{N} \sum_{t=1}^{N} (\mathbf{s}_t - \langle \mathbf{s}_t \rangle)(\mathbf{s}_t - \langle \mathbf{s}_t \rangle)^{\mathrm{T}} = \mathbf{L}\mathbf{E}\mathbf{L}^{\mathrm{T}}. \tag{19}$$

Then, for the PCA whitening transform the stimuli are transformed as $\mathbf{s}_t^{(\mathrm{PCA})} = \mathbf{E}^{-\frac{1}{2}}\mathbf{L}^{\mathrm{T}}\mathbf{s}_t$ while the ZCA whitening transforms the stimuli as $\mathbf{s}_t^{(\mathrm{ZCA})} = \mathbf{L}\mathbf{E}^{-\frac{1}{2}}\mathbf{L}^{\mathrm{T}}\mathbf{s}_t$. From here, the procedure is the same as before with the whitened stimuli substituted for $\mathbf{s}_t$. We observed, however, that these decorrelation methods performed poorly, producing receptive fields that lacked discernible structure and with worse predictive power compared to standard STC.

4.3. Functional Basis Method

The FB method (Kaardal et al., 2013) was used to recover biologically interpretable characterizations of the receptive field. This is done by modeling the nonlinearity as logical circuit elements where a linear combination of the inputs determine the probability of a spike. The two most basic descriptions are logical AND:

$$P_{\mathrm{AND}}(y = 1|\mathbf{s}_t) = \prod_k \frac{1}{1 + e^{-b_k - \zeta_1 \mathbf{c}_k^{\mathrm{T}}\mathbf{s}_t - \zeta_2(\mathbf{c}_k^{\mathrm{T}}\mathbf{s}_t)^2}}, \tag{20}$$

and logical OR:

$$P_{\mathrm{OR}}(y = 1|\mathbf{s}_t) = 1 - \prod_k \left[1 - \frac{1}{1 + e^{-b_k - \zeta_1 \mathbf{c}_k^{\mathrm{T}}\mathbf{s}_t - \zeta_2(\mathbf{c}_k^{\mathrm{T}}\mathbf{s}_t)^2}}\right], \tag{21}$$

where b_k are thresholds, ζ_1 is a linear weighting, ζ_2 is a quadratic weighting, and $\mathbf{c}_k$ are FB vectors that are formed by taking linear combinations of the relevant components from $\boldsymbol{\Omega}$ (Equation 11). The functional basis is fit by minimizing the negative log-likelihood using an L-BFGS algorithm. However, since it is a nonconvex optimization and therefore not guaranteed to converge to a global minimum, the optimization is repeated with multiple random initializations until 50 consecutive optimizations fail to produce a model that better fits the training set data. The FB model that best fits the training data is returned.

In prior applications, the FB method has produced basis vector spaces that are equal to r_{opt}. However, this need not be the case and the FB method may yield a basis with more or less dimensions than the underlying subspace. The optimal basis size can be determined by varying the number of components until the negative log-likelihood evaluated on the cross-validation set saturates to desired accuracy. The minimal number of components necessary to reach saturation is the desired number of basis vectors.

4.4. Synthetic Data

The low-rank MNE method was tested on synthetic data generated from model receptive fields: (i) a neuron with a high SNR and (ii) a neuron with a low SNR. In this case, high and low SNR correspond to the relative decisiveness of each model neuron's response to a stimulus. For instance, a high SNR model neuron is more likely to have a nonlinearity where $P(y = 1|\mathbf{s})$ is close to 0 or 1 compared to a low SNR model neuron which is more likely to take on intermediate probabilities. Using Equation (1), high and low SNR model neurons can be generated from a given set of weights by adjusting the gain of z. Concretely, the model neurons were generated by taking a sum of the weighted outer-products of the orthonormal vectors stored in the 400 × 4 ($D \times r_{\mathrm{opt}}$) matrix $\mathbf{F}$ (**Figure 1A**) yielding the ground truth matrix, $\mathbf{J}_{\mathrm{GT}} = \mathbf{F}\mathbf{W}\mathbf{F}^{\mathrm{T}}$. The linear weights, $\mathbf{h}_{\mathrm{GT}}$, were set to zero while the threshold, a_{GT}, and the 4 × 4 diagonal weight matrix, $\mathbf{W}$, were independently rescaled to produce a mean firing rate $\langle y\rangle \approx 0.2$ by averaging $P(y = 1|\mathbf{s}_t)$ over t where $\mathbf{s}_t$ is the tth 20 × 20 pixel stimulus sample drawn from a correlated Gaussian distribution unrolled into a $D = 400$ vector. The diagonal elements of the rescaled weight matrix, $\mathbf{W}$, appear as the dashed lines in **Figure 1C** and correspond to the eigenvalues of $\mathbf{J}_{\mathrm{GT}}$. Note that the eigenvalues of $\mathbf{J}_{\mathrm{GT}}$ have larger variance for the high SNR model neuron compared to the low SNR model neuron which corresponds to the high SNR model having a higher gain.

The correlated Gaussian stimuli were generated by first drawing sample vectors, $\hat{\mathbf{s}}_t$, from a normal distribution with zero mean and unit variance. A correlated Gaussian stimulus vector is then obtained via $\mathbf{s}_t = \mathbf{C}_{\mathrm{cov}}^{\frac{1}{2}}\hat{\mathbf{s}}_t$ where $\mathbf{C}_{\mathrm{cov}}$ is a covariance matrix constructed from natural images. Explicitly, $\mathbf{C}_{\mathrm{cov}} = \frac{1}{n_{\mathrm{samp}}}\sum_{t=1}^{n_{\mathrm{samp}}} \boldsymbol{\kappa}_t\boldsymbol{\kappa}_t^{\mathrm{T}}$ where $\boldsymbol{\kappa}_t$ is the tth sample of a total of n_{samp} samples drawn from a set of 20 × 20 images unrolled into vectors. The responses were binarized into spikes by generating a list of uniformly distributed random numbers, $\xi_t \in [0, 1]$. If $\xi_t < P(y = 1|\mathbf{s}_t)$, then $y_t = 1$; otherwise, $y_t = 0$. The total number of spikes was 11,031 for the high SNR model and 10,434 for the low SNR model. The same 48,510 stimulus samples were used as stimulus input to the model neurons. All models (first-order MNE, low-rank MNE, full-rank MNE, and STC models) were trained, cross-validated, and tested on 70%/20%/10% nonintersecting subsets of the data samples, respectively. In these proportions, the results were validated via jackknife analysis where four training, cross-validation, and test sets were defined by circularly shifting the sample indices in each set upward by 25% intervals of the total number of samples in the set ($t \leftarrow t + N/4$; see **Figure 4**).

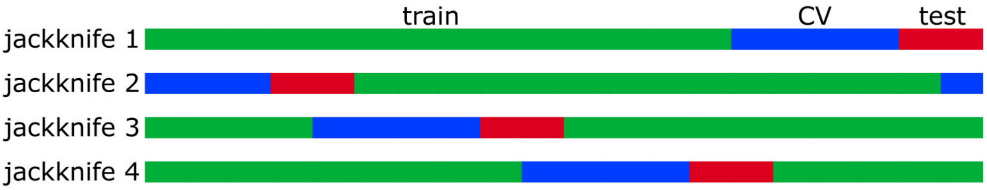

FIGURE 4 | The data samples for each neuron are divided into 70% training (green), 20% cross-validation (blue), and 10% test (red) sets. The samples that appear in each set are varied between 4 jackknives.

4.5. Avian Data

Data from the CRCNS database provided by the Theunissen laboratory was composed of *in vivo* electrophysiological recordings from anesthetized adult male zebra finches subjected to auditory stimuli (Gill et al., 2006; Amin et al., 2010). The recordings captured single-neuron action potentials from the auditory mid-brain, specifically 143 neurons from the mesencephalicus lateral dorsalis (MLd), 59 neurons from the ovoidalis (Ov), 37 neurons from the caudal mesopallium (CM), and 189 neurons from field L (L); the latter two of which are the focus of this paper. The temporal sampling resolution of the response was 1 ms. Two types of auditory recordings were used to stimulate action potentials: (1) 2 s samples of conspecific birdsong from 20 male zebra finches and (2) 10 synthetic recordings composed of sums of spectro-temporal ripples. Both stimulus types were bandpass filtered between 250 Hz and 8 kHz. These stimuli were presented to the zebra finches through speakers in a sound-attenuation chamber and each stimulus was repeated up to 10 times.

We processed the stimuli using MATLAB's spectrogram function on each sound clip and adjusted the frequency resolution of the spectrograms to find a reasonably well balanced compromise between the frequency and temporal resolution. Each Hamming window of the spectrogram had a 50% overlap with its neighbors. We found that a 250 Hz frequency resolution, which is coupled with a 2 ms temporal resolution, demonstrated a reasonable spectro-temporal resolution in the linear weights ($\mathbf{h}$) of first-order MNE models where structure of the receptive field could be resolved and there were plenty of stimulus/response pairs for model fitting. The spike times, each corresponding to one spike, were accumulated in 2 ms bins across all trials of an auditory recording. Any 2 ms period without any spikes was set to zero. Since the number of spikes in a bin could be greater than one, the spike count was divided by the maximum number of spikes across temporal bins $y_{\max} = \max(y_1, \cdots, y_N)$, ($y_t \leftarrow y_t/y_{\max}$). This ensured that the binned response was within the required range $y_t \in [0, 1]$ and effectively corresponds to reducing the bin width by $y_{\max}$. The stimulus samples were assigned by extracting 40–60 ms windows from the spectrogram preceding the neural response at y_t, excluding frequency bins well above and below the receptive field structures observed in $\mathbf{h}$, and unrolling each spectro-temporal window into a stimulus feature vector, $\mathbf{s}_t$. The response/stimulus pairs were then randomly shuffled to ensure that the training sets each provided a wide sampling of the response/stimulus distribution.

We selected 41 of the 189 field L neurons and 9 of the 37 CM neurons for analysis. These neurons were chosen based on whether a spectro-temporal window of the stimuli could be identified that produced an estimate of a single-component receptive field with an observable structure. For each neuron, the STA (Schwartz et al., 2006) and first-order MNE methods were used to extract this component and the spectro-temporal window for each neuron was manually adjusted such that the estimated receptive field amplitude was confined to the spectro-temporal window. We did not use STC or second-order MNE methods for this pre-processing step because the computation of these second-order methods are relatively slow compared to the aforementioned first-order methods and the first-order methods appear to produce a single component that is a weighted average of the second-order components. Since the stimulus samples were spectro-temporally correlated, the STA suffered from bias leading to single component receptive field estimates covering excessively large spectral and temporal ranges when compared to the smaller spectro-temporal extent of the more appropriate MNE models. Furthermore, the STA was prone to yielding structure even when the temporal window was set far (e.g., >100 ms) from the spike-onset time where the MNE methods found no observable structure. By contrast, we found the first-order MNE method to be more reliable and less misleading, so first-order MNE was ultimately chosen to determine the spectro-temporal windowing of the neurons. The chosen neurons were then those that exhibited structure in the first-order MNE receptive field estimate determined by visual inspection. This procedure if anything biases the results toward greater performance of linear models. Despite this potential bias, we found that low-rank models outperformed the models based on one component. It is possible the relative improvement would be greater for other neurons not considered here.

The number of stimulus samples ranges from 9,800 to 58,169 with a median sample size of 42,474 and the spike counts are between 276 and 29,121 with a median count of 6,120. First-order MNE, low-rank MNE, full-rank MNE, STC, and FB models were trained, cross-validated, and tested on 70%/20%/10% of the data, respectively, over four jackknives incremented in the same way as was done with regard to the model neuron data (i.e., **Figure 4**).

4.6. Data Analysis

Since we knew already that $r_{\text{opt}} = 4$ for the model neurons, we fit all $r = 1, \cdots, 8$ low-rank MNE models demonstrating saturation of cross-validation performance at a $r = r_{\text{opt}} = 4$ model. For the avian data, r_{opt} was not known *a priori* so we instead fit low-rank MNE models with a maximum rank of $r = 20$ which satisfied the conditions set by the rank optimization section (above) where less than 10 of each positive and negative eigenvalues exceeded a magnitude greater than $1 \cdot 10^{-4}$ on the majority of all jackknives for each neuron. A summary of the specific parameters used in **Algorithm 2** to solve the low-rank MNE problems may be found in **Table 1**. We fit low-rank MNE models using both the stricter monotonic convergence approach and the looser nonmonotonic convergence approach in **Algorithm 2** and found the difference in predictive power on the test sets between the two models to be insignificant. However, the nonmonotonic convergence approach had a tendency to produce sparser eigenvalue spectra of $\mathbf{J}$ so we opted to present the results from this version of the algorithm instead.

Two measurements were used to evaluate the quality of the our models. The overlap metric (Fitzgerald et al., 2011a):

$$\mathcal{O}(\mathbf{X}, \mathbf{Y}) = \frac{\sqrt[r]{|\operatorname{Det}\left(\mathbf{X}\mathbf{Y}^{\mathrm{T}}\right)|}}{\sqrt[2r]{|\operatorname{Det}\left(\mathbf{X}\mathbf{X}^{\mathrm{T}}\right)|}\,\sqrt[2r]{|\operatorname{Det}\left(\mathbf{Y}\mathbf{Y}^{\mathrm{T}}\right)|}} \qquad (22)$$

measures how well the receptive field is recovered as measured on an interval $\mathcal{O} \in [0, 1]$ where 0 means the two subspaces, $\mathbf{X}, \mathbf{Y}$ ($\mathbf{X}$ and $\mathbf{Y}$ are generic matrices and unrelated to any other variables defined in the paper), are complementary while 1 means the subspaces span the same range space. The overlap metric allowed us to compare the quality of the r_{opt} recovered vectors of highest variance in $\boldsymbol{\Omega}$ (Equation 11) to the model neuron subspace defined in the matrix $\mathbf{F}$. Of course, since $\mathbf{F}$ was not available for the avian neurons, this measure was not used to evaluate solutions on the avian data. A second measure of the quality of the fit was the predictive power of the models in the reserved test sets. This was done by calculating the negative log-likelihood $L_{\text{test}}(a, \mathbf{h}, \mathbf{J})$ evaluated over the test sets composed of the remaining data samples that were not used to train or cross-validate models (see **Figure 4**). In this latter assessment, models with minimal L_{test} were the best at predicting neural responses and assumed to recover better approximations of the underlying receptive field. Our application of these assessments to the model neurons were consistent with this assumption (**Figure 1**).

Once the mean components were recovered from $\mathbf{J}_{\text{sym}}$ averaged across jackknives, the FB method was applied to the avian data using the same data divisions as before (4 jackknives with 70%/20%/10% samples reserved for training, cross-validation, and testing). The FB basis set size was determined by finding the number of vectors, $\mathbf{c}_k$, necessary to saturate the negative log-likelihood up to a precision of $\sim 10^{-4}$. Both logical AND and logical OR functions were fit for each neuron.

4.7. Resolving Inconsistent Optimal Rank

In cases where model parameters are determined from multiple training and cross-validation sets, there is a risk that, due to the unique biases of each data set's sampling of stimulus and response space, the datasets may produce weights that disagree on the value of r_{opt}, the optimal rank of $\mathbf{J}$. For such cases, we use a statistical approach based in random matrix theory as a standard for deciding which eigenvalues of the mean $\langle \mathbf{J}_{\text{sym}} \rangle$ (Equation 11) across jackknives are significantly distinguishable from eigenvalues dominated by noise. The logic behind this statistical approach is as follows. Suppose that $\langle \mathbf{J}_{\text{sym}} \rangle$ is a large ($D \gg 1$) matrix that comes from a distribution of random symmetric matrices $\hat{\mathbf{J}}$ with elementwise mean $\langle \hat{J}_{i,j} \rangle = 0$ and variance $\langle \hat{J}_{i,j}^2 \rangle = \hat{\delta}^2$. What is the probability that the kth eigenvalue of $\langle \mathbf{J}_{\text{sym}} \rangle$ comes from this distribution of random matrices?

According to the Wigner semi-circle law, in the limit $D \to \infty$ the eigenvalues of random symmetric matrices of this type follow the probability distribution $P(\beta) = \frac{1}{2\pi\hat{\delta}^2}\sqrt{4\hat{\delta}^2 - \beta^2}$ for $|\beta| \leq 4\hat{\delta}^2$ and 0 otherwise. In other words, the probability distribution is bounded at $-2|\hat{\delta}| \leq \beta \leq 2|\hat{\delta}|$ and β outside of these bounds is asymptotically improbable. Thus, if we can generate this probability distribution, we can define a principled method to find the mean optimal rank $\langle r_{\text{opt}} \rangle$ using the bounds of the eigenvalue distribution, $P(\beta)$, as a null hypothesis. Unfortunately, we do not know the probability distribution from which $\langle \mathbf{J}_{\text{sym}} \rangle$ is drawn so the analytic probability distribution is out of reach; but we can assume a conservative estimate of the bounds of the null hypothesis through an empirical estimate of a broad-$\hat{\delta}^2$ variance distribution. We make this empirical estimate by generating random symmetric matrices $\hat{\mathbf{J}}$ where $\hat{J}_{i,j} = \pm\langle J_{\text{sym}}^{(m,n)} \rangle$ and m, n are random integers on the interval $[1, D]$. The sign on $\hat{J}_{i,j}$ is chosen with equal probability to ensure that $\langle \hat{J}_{i,j} \rangle = 0$ across the distribution while randomly drawing elements $\langle J_{\text{sym}}^{(m,n)} \rangle$ ensures a constant $\langle \hat{J}_{i,j}^2 \rangle$ across i, j. By aggregating the magnitude of the minimum and maximum eigenvalues from each of the random matrices, an estimate can be made on the bounds of the null hypothesis. With regard to this estimate of the bounds, p_k is defined as the probability that the magnitude of the kth eigenvalue of $\langle \mathbf{J}_{\text{sym}} \rangle$ is less than or equal to the magnitude of the bounds. If $p_k < p_{\text{thres}}$ where $p_{\text{thres}} \in [0, 1]$ is a significance threshold, then the kth eigenvalue of $\langle \mathbf{J}_{\text{sym}} \rangle$ is considered a statistically significant outlier from the null distribution with probability $1 - p$. This estimate of the underlying probability distribution is conservative because it is designed to have a large variance and thus a large width for the semi-circle distribution such that an eigenvalue β of $\langle \mathbf{J}_{\text{sym}} \rangle$ is more likely to fall within the bounds of the null distribution. For a pseudocode outline of this algorithm, (see **Algorithm 3**).

TABLE 1 | Summary of parameter values used in our application of **Algorithm 2**.

Parameter	Value	Definition
r	Varies	Maximum rank of $\mathbf{J}$
$\pi_1, \cdots, \pi_r$	Varies	Constraint signs
$\epsilon_{\max}$	0.5	Maximum value of the regularization parameters
n_{grid}	501	Number of different values the regularization parameter can assume forming a uniform grid from 0 to $\epsilon_{\max}$
T_{train}	70% of samples	Indices of data samples that form the training set
T_{CV}	20% of samples	Indices of data samples that form the cross-validation set
$M_{\max}$	20	Maximum number of iterations of the block coordinate descent algorithm
δ_p	0 (machine precision)	Convergence precision
$\sigma_{\max}$	3	Number of allowed failures to improve cross-validation performance

Since the convergence precision, δ_p, is set to zero, the algorithm converges when the negative log-likelihood evaluated on the cross-validation set does not decrease above machine precision. The constraint signs are assigned to equal numbers of positive and negative components.

4.8. Resources

The interior-point method, block coordinate descent algorithm, and FB method were written in Python 2.7 using standard numerical packages numpy (version 1.11.1) and scipy (version 0.18.1) and the machine learning package Theano (version 0.8.2) (Al-Rfou et al., 2016). These packages were installed through Anaconda (version 1.5.1) and were linked against Intel MKL (version 1.1.2) for CPU parallelization of linear

Algorithm 3 Statistical approach for choosing $\langle r_{\text{opt}} \rangle$

```
inputs: $\mathbf{J}_{\text{sym}} \leftarrow \langle \mathbf{J}_{\text{sym}} \rangle$ averaged across jackknives, $p_{\text{thres}}$, the number of random matrices to generate $M$
initialization: calculate the vector of eigenvalues $\boldsymbol{\beta} \leftarrow \text{eig}(\mathbf{J}_{\text{sym}})$ arranged in descending order of magnitude, initialize empty
  vector $\boldsymbol{\zeta} \leftarrow \emptyset$, $\langle r_{\text{opt}} \rangle \leftarrow 0$

for $m = 1$ to $M$ do
    $\hat{\mathbf{J}} \leftarrow$ randomly sample $D(D+1)/2$ elements of $\pm\mathbf{J}_{\text{sym}}$ with uniform
        probability and generate a $D \times D$ symmetric matrix.
    $\boldsymbol{\zeta} \leftarrow \left[\boldsymbol{\zeta},\ |\min(\text{eig}(\hat{\mathbf{J}}))|,\ \max(\text{eig}(\hat{\mathbf{J}}))\right]$        ▷ Append magnitude of maximum and minimum
                                                                  eigenvalues of $\hat{\mathbf{J}}$
for $k = 1$ to $D$ do
    $p \leftarrow \frac{1}{2M} \sum_{m=1}^{2M} H(\zeta_m - |\beta_k|)$        ▷ $H(\cdot)$ is the Heaviside step function
    if $p \geq p_{\text{thres}}$ then
        break
    else
        $\langle r_{\text{opt}} \rangle \leftarrow k$

output: $\langle r_{\text{opt}} \rangle$
```

algebra operations. Theano was chosen because it conveniently allows investigators to flexibly choose between using graphics processing units (GPUs) or central processing units (CPUs) as a backend to the optimization code without requiring modification to the code itself. We initially experimented with using GPUs to optimize the low-rank MNE models but found that the limitation to 32-bit floating-point precision on the available GPUs was inadequate without much hands-on tuning of the optimization parameters of the interior-point method which was not ideal for applications involving large datasets. In particular, using 32-bit floating-point precision in the algorithm often led to ill-conditioning of the Hessian matrix. These issues with 32-bit floating-point precision were replicated on CPUs as well. On the other hand, using 64-bit floating-point precision on CPUs did not present any issues with convergence or ill-conditioning. Consequently, we performed our low-rank MNE optimizations on a cluster of CPUs using 64-bit floating-point precision. The FB method, on the other hand, did not have any issues with precision so we ran these optimizations on GPUs using 32-bit precision. The optimization of full-rank and first-order MNE problems was done in C using OpenMP (version 4.0) and OpenBlas (version 0.2.14) for CPU parallelization. Figures were generated using MATLAB (version R2016b) and Inkscape (version 0.92.1).

ETHICS STATEMENT

This study was carried out with the approval of the Animal and Use Committee at University of California, Berkeley.

AUTHOR CONTRIBUTIONS

The author contributions are itemized below: Model development-JK; Data acquisition-FT; Data analysis and interpretation-JK, TS; Drafting of the manuscript/revising for critically important intellectual content: JK, FT, TS.

FUNDING

This research was supported by the National Science Foundation (NSF) CAREER award number IIS-1254123 and award number IOS-1556388, the National Eye Institute of the National Institutes of Health under Award Numbers R01EY019493, P30EY019005, and T32EY020503, McKnight Scholarship, Ray Thomas Edwards Career Award, and the H.A. and Mary K. Chapman Charitable Trust.

ACKNOWLEDGMENTS

We would like to thank the members of the Sharpee group at the Salk Institute for Biological Studies for helpful discussions and insight.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: http://journal.frontiersin.org/article/10.3389/fncom.2017.00068/full#supplementary-material

REFERENCES

Al-Rfou, R., Alain, G., Almahairi, A., Angermueller, C., Bahdanau, D., Ballas, N., et al. (2016). Theano: a python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688.

Amin, N., Gill, P., and Theunissen, F. E. (2010). Role of zebra finch auditory thalamus in generating complex representations for natural sounds. *J. Neurophysiol.* 104, 784–798. doi: 10.1152/jn.00128.2010

Bach, F., Mairal, J., and Ponce, J. (2008). Convex sparse matrix factorizations. *arXiv preprint*.

Bell, A. J., and Sejnowski, T. J. (1997). The "independent components" of natural scenes are edge filters. *Vis. Res.* 23, 3327–3338. doi: 10.1016/S0042-6989(97)00121-1

Bergstra, J., and Bengio, Y. (2012). Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* 13, 281–305.

Bergstra, J. S., Bardenet, R., Bengio, Y., and Kégl, B. (2011). "Algorithms for hyper-parameter optimization," in *Advances in Neural Information Processing Systems*, eds J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger (Granada: MIT Press in Cambridge), 2546–2554.

Bialek, W., and de Ruyter van Steveninck, R. R. (2005). Features and dimensions: motion estimation in fly vision. q-bio/0505003.

Bottou, L. (2010). "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, eds Y. Lechevallier and G. Saporta (Paris), 177–186.

Brochu, E., Cora, V. M., and De Freitas, N. (2010). A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*.

Burer, S., and Monteiro, D. C. (2003). A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Math. Prog.* 95, 329–357. doi: 10.1007/s10107-002-0352-8

Cabral, R. (2013). "Unifying nuclear norm and bilinear factorization approaches for low-rank matrix decomposition," in *IEEE International Conference on Computer Vision* (Sydney, NSW).

Fazel, M. (2002). *Matrix Rank Minimization with Applications*. Ph.D. thesis, Stanford, CA: Stanford University.

Fazel, M., Hindi, H., and Boyd, S. P. (2003). "Log-det heuristic for matrix rank minimization with applications to hankel and euclidean distance matrices," in *American Control Conference, 2003* (Denver, CO).

Felleman, D. J., and Van Essen, D. C. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cereb. Cortex* 1, 1–47. doi: 10.1093/cercor/1.1.1

Fitzgerald, J. D., Rowekamp, R. J., Sincich, L. C., and Sharpee, T. O. (2011a). Second-order dimensionality reduction using minimum and maximum mutual information models. *PLoS Comput. Biol.* 7:e1002249. doi: 10.1371/journal.pcbi.1002249

Fitzgerald, J. D., Sincich, L. C., and Sharpee, T. O. (2011b). Minimal models of multidimensional computations. *PLoS Comput. Biol.* 7:e1001111. doi: 10.1371/journal.pcbi.1001111

Gill, P., Zhang, J., Woolley, S. M., Fremouw, T., and Theunissen, F. E. (2006). Sound representation methods for spectrotemporal receptive field estimation. *J. Comput. Neurosci.* 21, 5–20. doi: 10.1007/s10827-006-7059-4

Haeffele, B. D., Young, E. D., and Vidal, R. (2014). "Structured low-rank matrix factorization: optimality, algorithm, and applications to image processing," in *31st International Conference on Machine Learning, ICML 2014, Vol. 5*, (Beijing), 4108–4117.

Jaynes, E. T. (2003). *Probability Theory: The Logic of Science*. Cambridge: Cambridge University Press.

Kaardal, J., Fitzgerald, J. D., Berry, M. J. II, and Sharpee, T. O. (2013). Identifying functional bases for multidimensional neural computations. *Neural Comput.* 25, 1870–1890. doi: 10.1162/NECO_a_00465

King, A. J., and Nelken, I. (2009). Unraveling the principles of auditory cortical processing: can we learn from the visual system? *Nat. Neurosci.* 12, 698–701. doi: 10.1038/nn.2308

Nocedal, J., and Wright, S. J. (2006). *Numerical Optimization*. New York, NY: Springer.

Oliver, M. D., and Gallant, J. L. (2010). "Recovering nonlinear spatio-temporal receptive fields of v1 neurons via three-dimensional spike triggered covariance analysis," in *Program No. 73.1. 2010 Neuroscience Meeting Planner* (San Diego, CA: Society for Neuroscience).

Park, I. M., and Pillow, J. W. (2011). "Bayesian spike-triggered covariance analysis," in *Advances in Neural Information Processing Systems 24*, eds J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger (Cambridge, MA: MIT Press), 1692–1700.

Perrinet, L. U., and Bednar, J. (2015). Edge co-occurances can account for rapid categorization of natural versus animal images. *Sci. Rep.* 5:11400. doi: 10.1038/srep11400

Rajan, K., and Bialek, W. (2013). Maximally informative "stimulus energies" in the analysis of neural responses to natural signals. *PLoS ONE* 8:e71959. doi: 10.1371/journal.pone.0071959

Recht, B., Fazel, M., and Parillo, P. A. (2010). Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Rev.* 52, 471–501. doi: 10.1137/070697835

Rust, N. C., Schwartz, O., Movshon, J. A., and Simoncelli, E. P. (2005). Spatiotemporal elements of macaque v1 receptive fields. *Neuron* 46, 945–956. doi: 10.1016/j.neuron.2005.05.021

Schwartz, O., Pillow, J., Rust, N., and Simoncelli, E. P. (2006). Spike-triggered neural characterization. *J. Vis.* 6, 484–507. doi: 10.1167/6.4.13

Serre, T., Oliva, A., and Poggio, T. (2007). A feedforward architecture accounts for rapid categorization. *Proc. Natl. Acad. Sci. U.S.A.* 104, 6424–6429. doi: 10.1073/pnas.0700622104

Sharpee, T., Rust, N., and Bialek, W. (2004). Analyzing neural responses to natural signals: maximally informative dimensions. *Neural Comput.* 16, 223–250. doi: 10.1162/089976604322742010

Snoek, J., Larochelle, H., and Adams, R. P. (2012). "Practical bayesian optimization of machine learning algorithms," in *Advances in Neural Information Processing Systems*, eds F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Stateline, NV: MIT Press in Cambridge), 2951–2959.

Wächter, A., and Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Prog.* 106, 25–57. doi: 10.1007/s10107-004-0559-y

Wright, S. J. (2015). Coordinate descent algorithms. *Math. Prog.* 151, 3–34. doi: 10.1007/s10107-015-0892-3

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.