ARTICLE **Communicated by Tim Gollisch**

Neural Quadratic Discriminant Analysis: Nonlinear Decoding with V1-Like Computation

Marino Pagan
marinopagan@gmail.com
Department of Psychology, University of Pennsylvania, Philadelphia, PA 19104, U.S.A.

Eero P. Simoncelli
eero.simoncelli@nyu.edu
Center for Neural Science and Courant Institute of Mathematical Sciences, New York University, New York, NY 10003, U.S.A. and Howard Hughes Medical Institute

Nicole C. Rust
nrust@sas.upenn.edu
Department of Psychology, University of Pennsylvania, Philadelphia, PA 19104, U.S.A.

Linear-nonlinear (LN) models and their extensions have proven successful in describing transformations from stimuli to spiking responses of neurons in early stages of sensory hierarchies. Neural responses at later stages are highly nonlinear and have generally been better characterized in terms of their decoding performance on prespecified tasks. Here we develop a biologically plausible decoding model for classification tasks, that we refer to as neural quadratic discriminant analysis (nQDA). Specifically, we reformulate an optimal quadratic classifier as an LN-LN computation, analogous to "subunit" encoding models that have been used to describe responses in retina and primary visual cortex. We propose a physiological mechanism by which the parameters of the nQDA classifier could be optimized, using a supervised variant of a Hebbian learning rule. As an example of its applicability, we show that nQDA provides a better account than many comparable alternatives for the transformation between neural representations in two high-level brain areas recorded as monkeys performed a visual delayed-match-to-sample task

1 Introduction

Sensory encoding models, which describe how the inputs to a neuron are converted into its responses, have proven effective in a broad array of sensory modalities, brain areas, and species (Eggermont, Aertsen, & Johannesma 1983; Jones & Palmer, 1987; DiCarlo, Johnson, & Hsiao, 1998). Within

Neural Computation **28, 2291–2319** (2016)
doi:10.1162/NECO_a_00890

vision, classic examples include the center-surround receptive field of a retinal ganglion cell (Enroth-Cugell & Robson 1966), the energy model of V1 complex cell (Adelson & Bergen, 1985), and the divisive normalization model of gain control (Heeger, 1992), as well as their more contemporary variants (e.g., Keat, Reinagel, Reid, & Meister, 2001; Rust, Schwartz, Movshon, & Simoncelli, 2005; Sharpee et al., 2006; Pillow et al., 2008; Carandini & Heeger, 2011). Considerable effort has been devoted to developing and refining techniques for fitting these models to data derived from a single experiment (reviewed by Ringach & Shapley, 2004; Schwartz, Pillow, Rust, and Simoncelli, 2006; Wu, David, & Gallant 2006; Sharpee, 2013). Successes were originally confined to brain areas positioned in early stages of the visual hierarchy but have since been extended to intermediate stages (David, Hayden, & Gallant, 2006; Rust, Mante, Simoncelli, & Movshon, 2006; Willmore, Prenger, & Gallant, 2010; Mineault, Khawaja, Butts, & Pack, 2012; Sharpee, Kouh, & Reynolds, 2013). Extending this approach to high-level brain areas has proven much more difficult.

One obstacle is that the techniques that have been developed to fit encoding models generally rely on a quantitative description of the inputs to the cells being fit. For example, encoding models of neurons within area MT have been fit based on a simulated population of inputs arriving from area V1 (Rust et al., 2006). Similar methods have been applied in V2 (Willmore et al., 2010), V4 (David et al., 2006; Sharpee et al., 2013), and MST (Mineault et al., 2012). But these approaches can extend our understanding only one stage beyond what is already relatively well understood.

An additional challenge arises from the fact that neural responses at higher stages become increasingly affected by behavioral task and context, and models for the "decoding" of task performance have generally been more successful than those for the "encoding" of sensory stimuli (Hung, Kreiman, Poggio, & DiCarlo, 2005; DiCarlo & Cox, 2007; Churchland et al., 2012; Mante, Sussillo, Shenoy, & Newsome, 2013; Pagan, Urban, Wohl, & Rust, 2013; Rigotti et al., 2013). These decoders have generally been assumed to be linear. And while the relative successes of handful of nonlinear (as compared to linear) decoders have been evaluated with neural data (Bialek, de Ruyter van Steveninck, Rieke, & Warland, 1996; Yu et al., 2007; Graf, Kohn, Jazayeri, & Movshon, 2011; Astrand et al., 2014), the neural mechanisms underlying nonlinear decoding remain unclear.

Arguably, what is needed to bridge this gap in describing high-level neural computations are techniques that (1) allow us to fit and evaluate biologically plausible descriptions of how the inputs to a brain area are transformed into its output responses in order to perform a specific task; (2) do not depend on a complete, quantitative description of how the inputs are derived from stimuli; and (3) can capture nonlinear transformations commonly found in neural responses. Here we develop such an approach. We reformulate a quadratic decoder, optimized for a prespecified classification task, as a linear-nonlinear-linear-nonlinear (LN-LN) cascade model, analogous

to models used for describing encoding in early visual areas. We show that this computation, which we call neural quadratic discriminant analysis (nQDA), can account for the increase in classification performance between high-level visual brain areas IT and perirhinal cortex during a target search task. We also introduce a biologically plausible supervised learning rule for optimizing the parameters of nQDA.

2 Results

2.1 nQDA, a Nonlinear Decoder Expressed as an LN-LN Model. Consider the problem of classifying a set of inputs (e.g. "Is observed input X a member of group A or group B?") based on a population of neural responses. The simplest solution is to assume a linear classifier, expressed as

$$f_{LIN}(r) = m^T r + k, \tag{2.1}$$

where r is the N-dimensional population response vector, $m^T r$ is the weighted sum (inner product) of the responses with N-dimensional vector of weights m, and k is a scalar constant. The class assignment (A versus B) is determined by the sign of the output (positive versus negative, respectively). The decision boundary corresponds to a hyperplane in the population response space, positioned such that it separates the two classes (Fisher, 1936; Cortes & Vapnik, 1995; see Figure 1a, top left). Many algorithms are available for fitting the parameters (m, k) (Duda, Hart, & Stork, 2000) and the "best" choice depends on the distribution of the data, noise properties, and the costs of making mistakes. When the population responses for each class are gaussian distributed, with equal covariance, the maximum likelihood solution is the Fisher linear discriminant (FLD; Fisher 1936). FLD parameters may be expressed directly as

$$m = \Sigma^{-1} \cdot (\mu_1 - \mu_2) \qquad k = \frac{1}{2} \cdot \left[(\mu_2^T \cdot \Sigma^{-1} \cdot \mu_2) - (\mu_1^T \cdot \Sigma^{-1} \cdot \mu_1) \right], \tag{2.2}$$

where μ_i indicates the mean population response vector for the ith class and Σ is the common response covariance matrix. If the two covariances differ, this is typically replaced with their average, $\Sigma = \frac{1}{2} \cdot (\Sigma_1 + \Sigma_2)$, although this is no longer the maximum likelihood solution.

Linear classifiers are considered biologically plausible—a weighted sum of inputs from the population in question, followed by a threshold nonlinearity. But linear classifiers are limited, relying on differences between the means of the class response distributions. Arguably, the simplest nonlinear extension is to include a quadratic term in the classifier:

$$f_{QUAD}(r) = r^T Q r + m^T r + k, \tag{2.3}$$

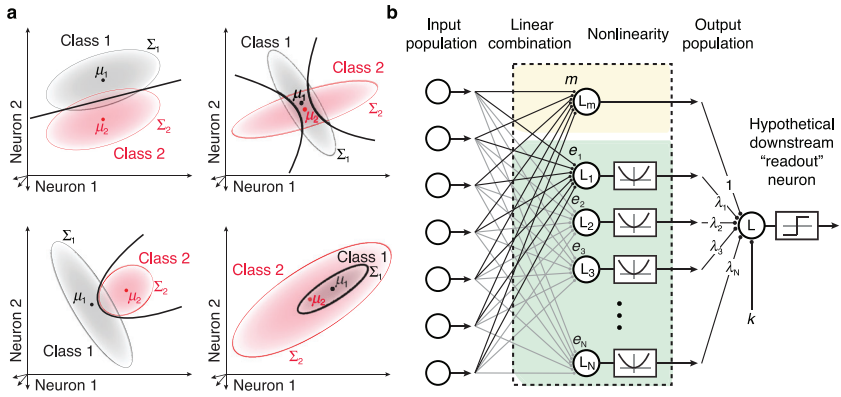


Figure 1: The nQDA framework. (a) Optimal quadratic discrimination boundaries (black lines) for four example pairs of population response distributions. Hypothetical class distributions are each multivariate gaussian (indicated by red and gray elliptical regions), with means μ_1 and μ_2 and covariances Σ_1 and Σ_2. Top left: A scenario in which the means of the two classes differ and the covariances are matched. In this special case, the optimal classifier is linear. Top right: A scenario in which the means of the two classes are similar and a linear classifier alone is an ineffective decision boundary. Instead, the optimal classifier uses a pair of parabolic boundaries. Bottom left: An example with differing mean and covariance, yielding a single parabolic boundary. Bottom right: An example yielding an elliptical boundary. (b) Depiction of the nQDA model, which implements the optimal quadratic classifier (equations 2.3 and 2.4) as an LN-LN model (see equation 2.6). The first LN transformation is achieved with a bank of linear filters, with all but the first followed by a squaring nonlinearity. The outputs of these individual LN units are combined via a weighted sum, followed by a threshold function that determines the class membership.

where Q is an N-by-N symmetric matrix. As with the linear classifier, the class assignment is determined by the sign of this function. Similar to linear classifiers, multiple methods exist for fitting the parameters of a quadratic classifier (Kendall, 1966; Hofmann, Schölkopf, & Smola, 2008). When the population responses are gaussian distributed the maximum likelihood solution (known as quadratic discriminant analysis, QDA; Kendall 1966) corresponds to

$$Q = \frac{1}{2} \cdot \left(\Sigma_2^{-1} - \Sigma_1^{-1}\right); \quad m = \Sigma_1^{-1}\mu_1 - \Sigma_2^{-1}\mu_2,$$
$$k = -\frac{1}{2}\left(\log\left|\Sigma_1\right| - \log\left|\Sigma_2\right| + \mu_1^T\Sigma_1^{-1}\mu_1 - \mu_2^T\Sigma_2^{-1}\mu_2\right). \tag{2.4}$$

The incorporation of the quadratic term creates a more powerful classifier, which exploits differences in covariance, generally

resulting in curved decision boundaries in the population response space (see Figure 1a).

Despite its potential for decoding information embedded in a population, it is not obvious how a brain area would implement QDA. Toward this end and inspired by the successes of quadratic encoding models, we have reformulated QDA into a more biologically plausible framework. We first expand the quadratic term $r^T Qr$ (see equation 2.3) into a set of linear-nonlinear operations, using the eigendecomposition of Q. Specifically, we write

$$Q = E\Lambda E^T, \text{ where } E = \begin{bmatrix} e_1 & e_2 & \cdots & e_N \end{bmatrix} \text{ and } \Lambda = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_N \end{bmatrix}.$$

With this expansion, the quadratic term $r^T Qr$ of equation 2.3 becomes

$$r^T Qr = r^T \left(E\Lambda E^T\right) r = \left(r^T E\right) \Lambda \left(E^T r\right) = \left(E^T r\right)^T \Lambda \left(E^T r\right) = \sum_{i=1}^{N} \lambda_i \cdot \left(e_i^T r\right)^2. \tag{2.5}$$

That is, the term $r^T Qr$ can be computed as a linear projection of the response vector r onto the eigenvectors of Q, followed by a squaring nonlinearity, and a final linear recombination weighted by the eigenvalues of Q. Substituting this back into the expression for QDA (see equation 2.3) yields

$$f_{QUAD} = \sum_{i=1}^{N} \lambda_i \left(e_i^T r\right)^2 + m^T r + k. \tag{2.6}$$

This equation, combined with the final thresholding (decision) nonlinearity, specifies an LN-LN computation, as illustrated in Figure 1b, which we refer to as neural QDA (nQDA).

2.2 Geometrical Intuition of nQDA. In this section, we provide an intuitive geometrical description of how the nQDA computation converts the nonlinearly separable population representation of two classes into a more linearly separable format. In general, the information available to separate two classes can be regarded in terms of discrepancies between the moments of the two class distributions (i.e., mean, covariance, skew, ...). Consequently if the two classes are identical in all of their moments, they cannot be separated. The second term of the nQDA solution (see equation 2.6)

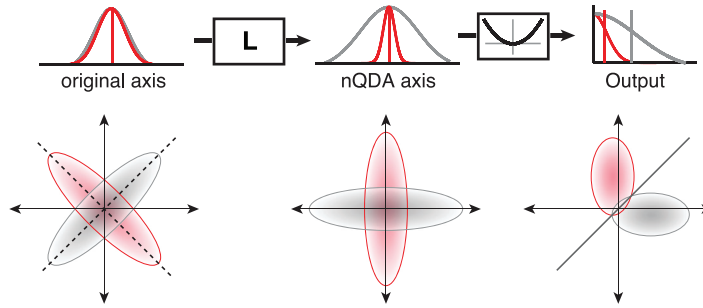


Figure 2: Geometrical intuition of nQDA. (Left) Response distributions of a hypothetical population to two classes, for which both the response mean and variance of each individual neuron are matched, but the class covariances differ. Dashed lines indicate nQDA axes (see equation 2.5). (Center) Population responses linearly transformed to nQDA axes, along which the responses exhibit large relative variance differences between the two classes. (Right), The squaring nonlinearity converts these variance differences into mean differences, and the resulting class distributions are more readily separated by a linear classifier (black line).

corresponds to a linear neuron that gathers the mean differences (i.e., the linearly separable information) present in the input population.

In contrast, the first term of equation 2.6 is a sum over a set of nonlinear neurons that act to transform nonlinearly separable information available in the input population into a linearly separable format. In particular, differences in variance are converted into differences in mean. To achieve this, the linear weights of this set of nonlinear neurons serve to rotate the input space to a coordinate system that captures the largest and smallest differences in inverse covariances (see Figure 2, left). Equivalently, since $\frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2} = \frac{\sigma_2^2 - \sigma_1^2}{\sigma_1^2 \cdot \sigma_2^2}$, this rotation maximizes the normalized variance differences between the projected responses for the two classes (see Figure 2, middle). Squaring these linearly transformed responses acts to convert the variance differences into mean differences (see Figure 2, right). These mean differences allow linear separation of the response clouds corresponding to the two classes in the output population.

2.3 nQDA Replicates the Transformation between IT and PRh. To determine the degree to which nQDA is useful for modeling neural data, we used it to characterize the transformation between responses of inferotemporal cortex (IT) and perirhinal cortex (PRh). Specifically, we recorded neural responses from IT and PRh as monkeys performed a delayed-match-to-sample sequential visual target search task that required them to indicate

when different target images appeared within sequences of distractors (see Figure 3a).

On each trial, monkeys sequentially viewed images while maintaining fixation and indicated when they saw a target image by shifting their gaze to a response dot on the screen. Our experimental design included four images presented in all possible combinations as a visual stimulus and as an intended target, resulting in 16 experimental conditions. We held the target image fixed for short blocks of trials and presented the same images as both targets and distractors in different blocks that were repeated several times. As monkeys performed this task, we recorded neural responses in IT or PRh using multichannel probes. To quantify the population response on any given trial, we counted spikes in a window starting 50 ms after stimulus onset (to allow time for signals to reach these brain areas) and ending at 220 ms, which always preceded the monkeys' eye movement responses. Here we present results based on population data concatenated across experimental sessions into larger "pseudopopulations" of 164 neurons for each brain area, following on our earlier report that factors specific to simultaneously recorded populations (i.e. noise correlations) do not affect population performance (Pagan et al., 2013; see Figure S2 in the online supplement).

This task can be envisioned as two-way classification of the same images presented as target matches (i.e., looking at and for the same image, for which the monkey is instructed to make an eye movement) versus distractors (i.e., looking at and for different images, for which the monkey is instructed to maintain fixation; see Figure 3b). To avoid the possibility that population performance could rely on factors other than the target match signal, we used an equal number of target matches and distractors (at any one time, we selected a subset of 4 of 12 possible distractors) and explored all possible subsets of distractors that spanned all visual stimuli and all targets (see Section 4).

As reported previously (Pagan et al., 2013), we found that total information for this classification was approximately matched in IT and PRh but that it was more linearly separable in PRh. This result is recapitulated in Figure 3c, where trial-level cross-validated performance of a linear decoder in PRh (PRh FLD) is seen to be similar to the performance of a nonlinear maximum likelihood decoder applied to IT (IT ML), suggesting that the linearly separable, task-relevant information contained within PRh is also largely present in its inputs arriving from IT. However, the performance of a linear classifier acting directly on IT responses (IT FLD) is significantly worse that the ML performance, suggesting that the information is nonlinearly embedded within IT. These results are consistent with a feedforward mechanism in which PRh performs computations to increase linear separability of its IT inputs (Pagan et al., 2013; Pagan & Rust, 2014a).

We wondered how well nQDA could mimic computations performed by PRh when applied to inputs arriving from IT. We found that a

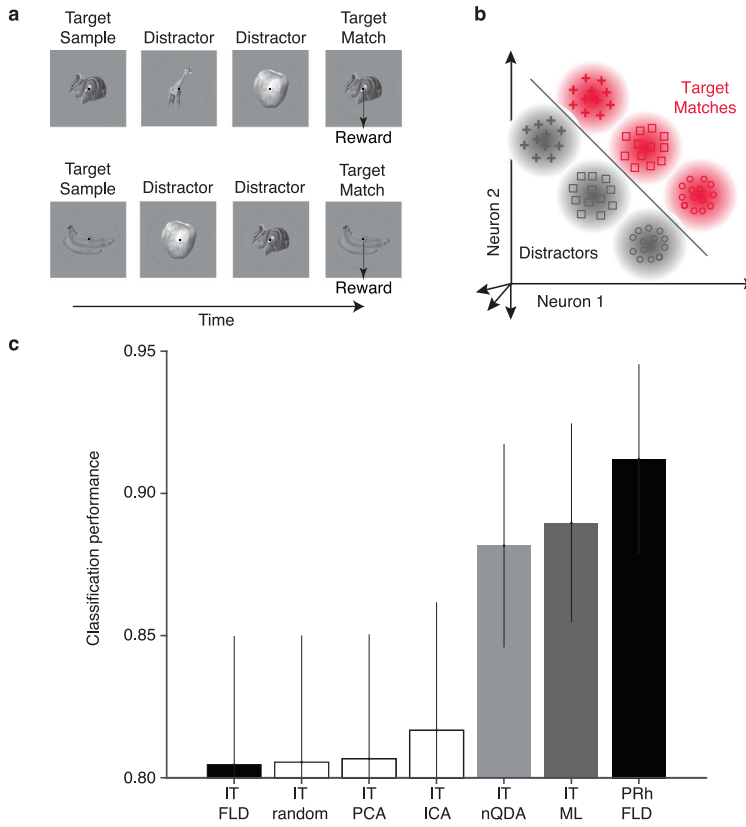


Figure 3: Application of nQDA to recorded neural responses. (a) We recorded neural responses in IT and PRh as monkeys performed a delayed-match-to-sample task. On each trial, monkeys were presented with a cue indicating the identity of the target ("target sample"), followed by a random number of distractors and a target match. Monkeys were required to indicate when the target match appeared. (b) Response clouds of two hypothetical neurons to target matches (red) and distractors (gray), where different shapes indicate different images. As described previously (Pagan et al., 2013), this task can be reformulated as a two-way classification between the set of responses to all the images presented as target matches versus the set of responses to the same images presented as distractors. (c) Trial-level cross-validated classification performance of different decoders applied to IT, along with a linear decoder applied to PRh. Plotted are the mean performance values, with error bars indicating bootstrapped estimates of standard error. Classifiers include an FLD decoder applied to IT and PRh (black); LN-LN decoders that include random, PCA, and ICA decompositions applied to IT, followed by optimized marginal nonlinearities and a final, FLD linear decoder (white); the nonlinear nQDA decoder applied to IT (light gray); and a nonlinear, maximum likelihood decoder applied to IT (dark gray).

164-dimensional nQDA-transformed IT population (see Figure 3c, IT nQDA) performed significantly better than a linear read-out of IT ($p = 0.022$), nearly as well as the upper bound imposed by the ML read-out of the same data, and comparable to a linear read-out of PRh. These results suggest that nQDA is largely successful at capturing the nonlinear transformation of target match information from IT to PRh.

We also considered whether a number of alternative LN-LN models could account for PRh computation. The structure of each of these alternative LN-LN models mirrored the structure of nQDA (see Figure 1b, and equation 2.6) insofar as each model began by projecting the same inputs onto a bank of linear filters, each followed by a scalar nonlinearity, and the resulting output values were combined using a final linear decoder. From a geometric perspective, these initial linear transformations can be interpreted as projections onto different sets of axes that span the original input space (see Figure 2), and our goal was to compare alternative methods for selecting these axes (random, PCA, and ICA) with nQDA. For each model the first axis of the linear transformation was chosen to be the same as the first nQDA axis (thus preserving the linearly separable information that already existed in IT), while the remaining axes were chosen using one of several methods (described below). Whereas nQDA used a fixed (squaring) nonlinearity, we allowed each of these alternative models to use a different nonlinearity for each axis, chosen as the log likelihood ratio between the projected responses to matches and distractors (see Figure 8).

For the first alternative model, we chose 164 random orthogonal axes (equal to the number of IT neurons). On average, this transformation did not exhibit better performance than the FLD applied to IT data (see Figure 3c, IT random). The performance of a random linear transformation can be improved by increasing the output dimensionality, but we found that using up to 1000 random axes led to only a small improvement (see Figure 4). It is worth noting, however, that a random linear transformation with output dimensionality equal to the number of degrees of freedom in the linear and quadratic terms in equation 2.3 ($N + N(N+1)/2$; in our case, approximately 14,000 axes) can exactly match the performance of nQDA.[1]

We also investigated whether an LN-LN model based on PCA or independent component analysis (ICA) linear transformations could match the performance of nQDA. We fit both methods to training subsets of IT population response to four target match and four distractor conditions (exploring

[1]Specifically, given a matrix P whose columns p_k are $N(N+1)/2$ randomly chosen N-vectors, with probability 1 there exists a diagonal matrix W (with diagonal elements w_k) such that the quadratic term in equation 2.3 can be expressed as $r^T Q r = r^T P W P^T r = \sum_k w_k (p_k^T r)^2$. This expression has the same LN-L form as the quadratic term of nQDA (see equation 2.5). The vector of diagonal elements w_k can be computed as $((P^T P)^{\cdot 2})^{-1} diag(P^T Q P)$, where $(.)^{\cdot 2}$ indicates element-wise squaring and *diag(.)* extracts a vector containing the diagonal elements of a matrix.

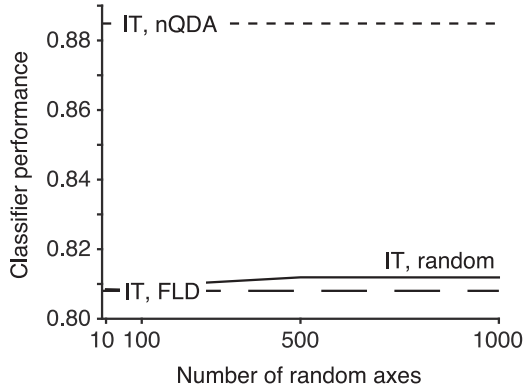


Figure 4: Performance of an LN-LN model with random axes improves marginally with increasing output dimensionality. Classifier performance is shown as a function of number of random axes. In contrast to the analysis depicted in Figure 3c IT random, which used a randomly chosen, orthogonal set of axes (with dimensionality $N = 164$, equal to that of the input space), here each random axis was selected independently. These randomly weighted inputs were then passed through a nonlinearity optimized for each axis, and the responses were combined using a final, linear decoder. For comparison, the FLD and nQDA decoders applied to IT (i.e., the same values in Figure 3c) are also indicated (by dashed lines).

all possible distractor subsets that spanned the four visual and four target dimensions, as described above), and we kept all the resulting dimensions. We found that a model using PCA as the initial linear transformation could not replicate the transformation from IT to PRh (see Figure 3c, IT PCA). Similarly, we found that despite a modest increase in performance over random projections and PCA, ICA also failed to replicate the transformation from IT to PRh (see Figure 3c, IT ICA). In summary, these results show that the optimal quadratic classifier implemented with nQDA provides an efficient (in terms of the number of cells) explanation of the transformation between high-level brain areas IT and PRh, surpassing other LN-LN models with optimized (nonquadratic) nonlinearities.

As an additional assessment of the degree to which nQDA replicated the transformation from IT to PRh, we examined the relative amounts of different task-relevant signals represented within the two brain areas. Specifically, we decomposed each neuron's responses into three component signals: (1) visual modulation (i.e., response modulation by changes in the identity of the visual stimulus), (2) target modulation (i.e., response modulation by changes in the identity of the target), and (3) target match modulation (i.e., modulation by changing whether a condition is a target match or a distractor), using the techniques described by Pagan and Rust (2014a; see section 4).

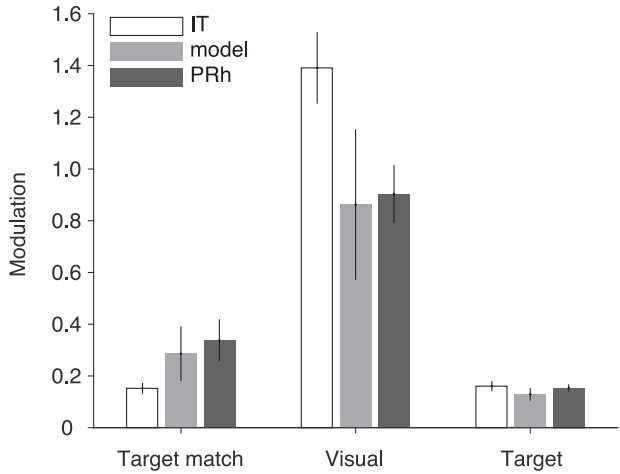


Figure 5: nQDA replicates magnitudes in PRh. The responses of each neuron were decomposed into different types of task-relevant signals by quantifying, for each neuron, the amount of spike count variance (around the grand mean spike count) that could be attributed to changes in experimental conditions: Target match (whether a condition was a target match or distractor), Visual (image identity), and Target (target identity) (see Section 4). These signal variances were then normalized by each neuron's trial-by-trial variance, averaged across conditions, to obtain unitless quantities that reflect the ratio of each type of signal relative to the noise. Shown are the means and standard errors of these signal measures, computed for each population.

We quantified the strength of each of these signals by estimating the proportion of spike count variance they accounted for, relative to the total variance across trials. As expected from the population performance results, the amount of target match signal increased from IT to PRh and the nQDA model replicated this increase (see Figure 5, Target match). Additionally, the amount of visual signal (which reflects visual image identity) decreased from IT to PRh, and this decrease was also replicated by the nQDA model (see Figure 5, Visual). Although the target match signal is similar to what was optimized by nQDA, visual signals were not directly fit by our procedure, and replication of their decrease from IT to PRh is thus nontrivial. We found no differences in the amount of target signal (which reflects target identity) between IT, PRh, and the model (see Figure 5, Target). These results provide additional support that nQDA captures transformations between IT and PRh.

2.4 Biologically Plausible Learning of nQDA Parameters. In the previous section, the parameters of an nQDA transformation were optimized

by estimating the mean and covariance of the IT population responses to each of the classes and inserting these into equations 2.4 and 2.5. We wondered how neural populations could learn these parameters without direct knowledge of the class covariances and without computing the eigendecomposition of the difference of the inverse covariances. While we lack a full solution to this problem, we have gained insight into the special case in which the two class distributions have matched eigenvectors but different eigenvalues (i.e., two input distributions with variance distributed along the same axes but in different amounts, including possible differences in rank order). For this case, a supervised extension to a Hebbian learning algorithm converges to nQDA weights, and we refer to this algorithm as Hebbian QDA (hQDA).

We first note that when the eigenvectors of two covariance matrices are matched, the eigendecomposition of difference of the inverse covariances (necessary for nQDA; see equation 4) can be computed via the eigendecomposition of the difference of the covariances. This emerges from the well-known fact that the eigenvectors of a matrix are the same as those of its inverse. Given this, we show that a simple local learning rule converges to the eigenvectors of the difference between class covariances. The hQDA learning rule is an extension of previous work by Oja (1982), who revealed that a Hebbian learning algorithm converges to the first PCA principal component. To review that work, we consider a simple scenario where a model neuron receives only two inputs, x_1 and x_2 which are weighted by synaptic weights w_1 and w_2 (see Figure 6a). The model neuron produces the output y as the linear combination of the two-dimensional vectors x and w:

$$y = x^T w \tag{2.7}$$

Under Hebbian learning, weights w_1 and w_2 are increased when the activity of inputs and output is correlated: when the input and the output "fire together," they "wire together." This is achieved by modifying the weights by a quantity Δw proportional to the vector of inputs x, scaled by the output value y, and by a constant coefficient η called the learning rate:

$$\Delta w = \eta \cdot x \cdot y. \tag{2.8}$$

Substituting y from equation 2.7 leads to

$$\Delta w = \eta \cdot x \cdot x^T w. \tag{2.9}$$

Finally, by taking the mean value of Δw, we obtain the following equation,

$$\langle \Delta w \rangle = \left\langle \eta \cdot x \cdot x^T w \right\rangle = \eta \cdot \left\langle x \cdot x^T \right\rangle w = \eta \cdot \Sigma \cdot w, \tag{2.10}$$

where we used angle brackets $\langle \cdot \rangle$ to denote averages and Σ to indicate the covariance of x. The solution of this equation (Oja, 1982) reveals that Hebbian learning converges to weights corresponding to the leading eigenvector of the covariance matrix, that is, to the first principal component (see Figure 6a).

The hQDA algorithm is inspired by the Hebbian rule (see equation 2.10), but it contains the modification that the model neuron receives an additional "top-down" supervision signal that specifies the class k of the current input (see Figure 6b; e.g., $k = 1$ for target matches and $k = 2$ for distractors). This top-down input acts to switch the sign of the learning rule between the two classes: it adopts a Hebbian rule for one class and an anti-Hebbian rule for the other, and as such, we refer to it as a "contrastive" Hebbian rule:

$$\begin{cases} \Delta w = \eta \cdot x \cdot y \quad if \quad k = 1 \\ \Delta w = -\eta \cdot x \cdot y \quad if \quad k = 2 \end{cases}. \tag{2.11}$$

If we now substitute equation 2.7 into equation 2.11 we obtain

$$\begin{cases} \Delta w = \eta \cdot x \cdot x \cdot w \quad if \quad k = 1 \\ \Delta w = -\eta \cdot x \cdot x \cdot w \quad if \quad k = 2 \end{cases}. \tag{2.12}$$

Taking the average value of we get:

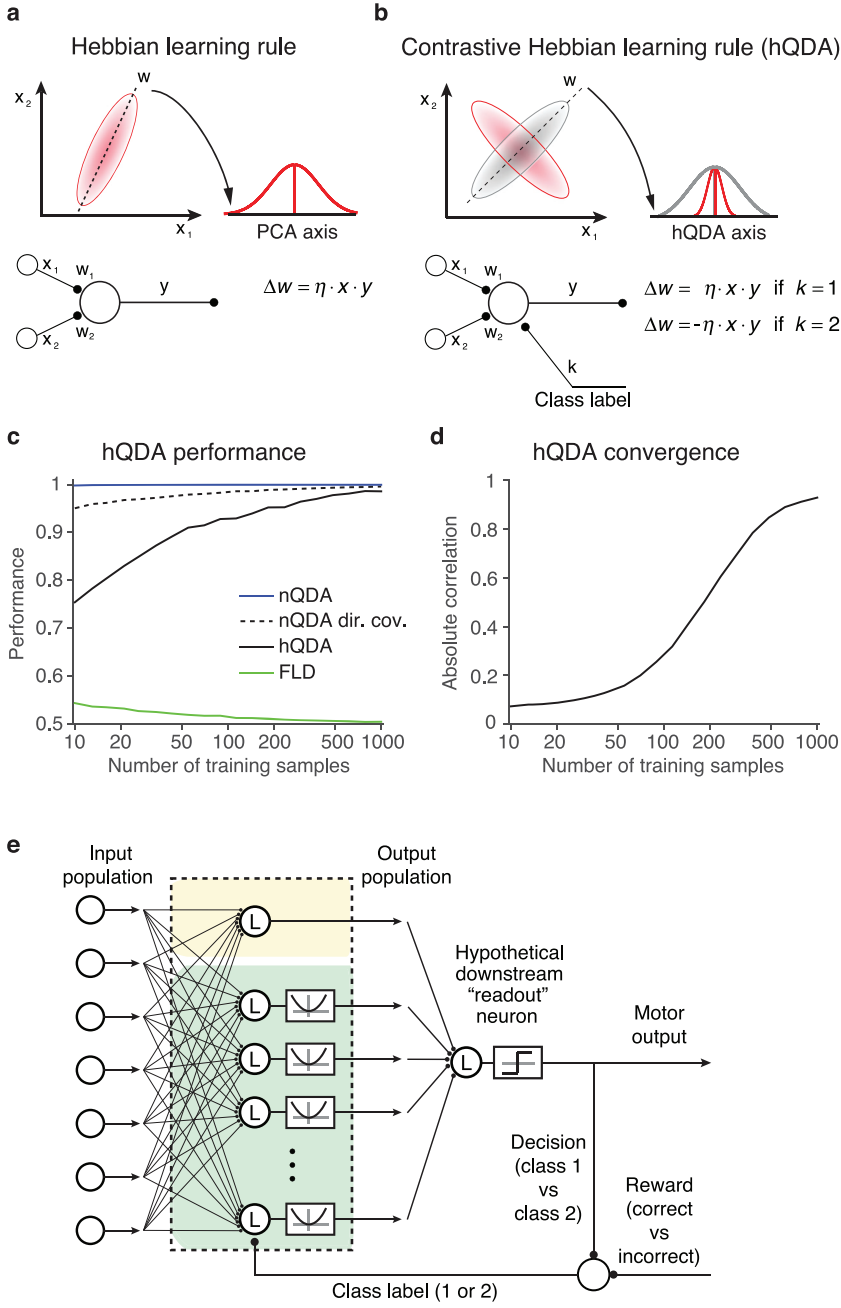
$$\begin{aligned} \langle \Delta w \rangle &= \langle \eta \cdot x_{k=1} \cdot x_{k=1} \cdot w \rangle - \langle \eta \cdot x_{k=2} \cdot x_{k=2} \cdot w \rangle = \dots \\ \dots &= \eta \cdot \langle x_{k=1} \cdot x_{k=1} \rangle \cdot w - \eta \cdot \langle x_{k=2} \cdot x_{k=2} \rangle \cdot w = \eta \cdot (\Sigma_1 - \Sigma_2) \cdot w, \end{aligned} \tag{2.13}$$

where $x_{k=1}$ indicates the inputs labeled for class 1, $x_{k=2}$ indicates those of class 2, Σ_1 is the covariance for the inputs labeled for class 1, and Σ_2 is the covariance for those of class 2. Note that this expression is equivalent to equation 2.10, with the exception that the covariance matrix is now replaced by the covariance difference $\Sigma_1 - \Sigma_2$. Consequently, the weights now converge to the leading eigenvector of the difference of the covariance matrices (i.e., to the axis with maximum variance difference), as determined by nQDA (see Figure 6b).

Two additional modifications are required to recover the hQDA weights. First, to avoid an unbounded exponential increase of the synaptic weights, we implement Oja's (1982) correction:

$$\begin{cases} \Delta w = \eta \cdot x \cdot y - \eta \cdot y^2 \cdot w \quad if \quad k = 1 \\ \Delta w = -\eta \cdot x \cdot y - \eta \cdot y^2 \cdot w \quad if \quad k = 2 \end{cases}. \tag{2.14}$$

Second, an elaboration is required to recover the weights of the neurons beyond the first. This can be achieved using Sanger's (1989) rule, in which



each successive neuron forces its weights to be orthogonal to all previously defined neurons:

$$\begin{cases} \Delta w = \eta \cdot x \cdot y - \eta \cdot y^2 \cdot w - \eta \cdot y \cdot \sum_i y^{(i)} \cdot w^{(i)} \quad if \quad k = 1 \\ \Delta w = -\eta \cdot x \cdot y - \eta \cdot y^2 \cdot w - \eta \cdot y \cdot \sum_i y^{(i)} \cdot w^{(i)} \quad if \quad k = 2 \end{cases}, \tag{2.15}$$

where $y^{(i)}$ is the output of the ith model neuron and $w^{(i)}$ are the weights of the ith model neuron.

hQDA is guaranteed to converge to the same axes as nQDA only when the eigenvalues of the two covariance matrices are identical, and we thus wanted to assess how well hQDA performs under realistic conditions. Because the IT data set described included too few conditions to evaluate hQDA, we performed a simulation in which we sampled a large number of artificial population responses from two multivariate gaussian distributions matched to the covariances matrices estimated from our data. To simplify the interpretation of this simulation, we subtracted the mean

Figure 6: Approximate nQDA parameters can be learned with a supervised, Hebbian rule, hQDA. (a) As described previously by Oja (1982), the Hebbian algorithm recovers the first PCA axis. Top: The population response distribution for two hypothetical neurons to two classes of stimuli. The first principal component axis (dotted line) captures the axis with maximum variance, summed across conditions. Bottom: Depiction of a neuron that receives inputs from the two neurons depicted at the top and whose weights are modified according to a classic Hebbian rule. Application of the Hebbian rule results in the convergence of the input weights to the first PCA axis. (b) Top: The population response distributions for two hypothetical neurons to two classes of stimuli. The first hQDA axis captures the axis with maximum variance difference (dotted). Bottom: depiction of the same neuron as shown in panel a but with an additional input that indicates the class label for each condition. The contrastive Hebbian rule is implemented in this framework by switching the sign of the classic Hebbian term according to this label. (c) Cross-validated classification performance of a number of decoders applied to a neurally inspired data set (see text), plotted as a function of the training set size. Classifiers are FLD (green), nQDA (blue), hQDA (black), and nQDA with the use of direct covariances instead of inverse covariances (black dashed line). hQDA was set to retrieve six axes, equal to the number of informative dimensions in the input space (see section 4). (d) Average absolute correlation between the axes obtained by applying hQDA to the same simulated data and the corresponding closed-form eigenvectors computed via the difference of covariances. Convergence to closed-form axes is shown as a function of the size of the training set. (e) Diagram for a model in which the contrastive Hebbian rule could be used to learn the hQDA parameters, where the class label is determined for each condition as a combination of the decision and whether the decision was rewarded.

of each class-conditioned distribution so that only differences in the covariances carried useful information. Figure 6d shows the classification performance of hQDA and other decoders as a function of the size of the training set. In this simulation, nQDA (blue) performed near 100%, whereas FLD (green) performed near chance, as expected due to of the absence of linearly separable information. To separately evaluate issues related to the approximation of the difference of the inverse covariances (with the difference of covariances) and other issues related to hQDA implementation, we began by determining performance when we applied a version of nQDA in which the axes were computed as the eigenvectors of the difference of the two covariance matrices (see section 4). Performance of this classifier converged to nQDA performance (see Figure 6c, black dashed line), suggesting that this is a reasonable approximation under the conditions imposed in this data-inspired simulation. Next, we recovered these axes using hQDA (see section 4) and found that it converged to nQDA performance in approxmately 1000 training trials (see Figure 6c, black solid line). As another measure of hQDA convergence, we computed the absolute value of the correlation between the axes retrieved by hQDA and the eigenvectors of the difference of the covariance matrices. As shown in Figure 6d, the average correlation between the hQDA axes and the closed-form solutions is more than 0.85 after about 500 training trials. Together these results suggest that hQDA does indeed converge to the axes predicted by equation 2.13, and it can do so with a reasonable amount of training data.

Unlike Hebbian learning, contrastive Hebbian learning as we have described it requires an input that signals the class of the current condition How might such an input be computed? Figure 6e shows a simple model demonstrating how this signal might be generated during training via a reinforcement learning algorithm. On each trial, a subject generates a behavioral decision (predicting whether each condition belongs to class 1 or 2) and receives a reward for correct responses. Class identity can uniquely be determined on each trial based on the combination of the predicted class and whether a reward was received: the presence of a reward confirms that the predicted label was correct, whereas the absence of a reward indicates that the predicted label was incorrect and should thus be switched. Our model proposes that information is used to compute the class label (i.e., $k = 1$ for target matches or $k = 2$ for distractors) and is then fed back as an input for contrastive Hebbian learning. Inputs that combine decision and reward information to mediate the modification of feedforward weights have been proposed by others (Law & Gold, 2009) and could be implemented biophysically via dopaminergic inputs, which are potent in PRh (Akil & Lewis, 1993; Richmond, 2006). Under this scenario, dopaminergic inputs could act as a gate for turning learning "on" and "off" and would thus prevent learning when it is inappropriate for it to occur (e.g., as a result of the cue period responses in our task; Soltani & Wang, 2006; Izhikevich, 2007).

3 Discussion

Quantitative descriptions of the computations implemented by higher brain areas have been difficult to develop. The responses of neurons in these areas are highly nonlinear functions of sensory inputs, requiring complex parameterization and large amounts of data. Alternatively, fitting simpler models to the computations performed only by the neurons in a given area generally relies on a precise description of the inputs to that area. Here we have used the structure of low-level encoding models as a basis for constructing decoding models to describe computation in higher stages. Our solution, neural quadratic discriminant analysis (nQDA), reformulates an optimal quadratic classifier as a linear-nonlinear, linear-nonlinear (LN-LN) cascade model, in which input arriving from a population of neurons is transformed at the first stage by a bank of linear filters, followed by squaring nonlinearities, and a final read-out with a second LN stage. The model provides both a means of fitting neural data with an optimal classifier, as well as a biologically plausible description of neural mechanism.

We arrived at nQDA via our previous attempts to account for the transformation between two high-level brain areas, IT and perirhinal cortex (Pagan et al., 2013). In that work, we fit a specific LN-LN classifier model (in which pairs of IT cells combined to form pairs of perirhinal neurons) with a brute force parameter search. Here we have used the insight gained from the pairwise model—that increased linear separability can be accomplished by converting class variance differences into mean differences by squaring—to develop a more general solution. Specifically, we have generalized this procedure to multiple dimensions by deriving it as a form of QDA (see equations 2.3–2.6); shown that nQDA provides a better account of a transformation between two high-level brain areas than a number of comparable alternatives (see Figures 2.3 and 2.4); and developed a biologically plausible learning rule that can be used to estimate nQDA parameters, hQDA (see Figure 6).

3.1 Similarities between Computation in PRh and V1. Although nQDA was designed from a decoding perspective, its structure (see Figure 1b) is qualitatively similar to functional models commonly used to describe neural computation in V1 (see Figure 7). In their simplest form, the response of a V1 simple cell, including selectivity for orientation, spatial frequency, and phase, is captured by an oriented linear filter followed by a threshold (Heeger, 1992), and the phase invariance of the V1 complex cell is captured by the two oriented linear filters of differing phase whose responses are squared and summed (Adelson & Bergen, 1985). In previous work, we and others have proposed a more elaborate LN-LN "subunit" model to capture the continuum of cells whose response properties lie between simple and complex (Rust et al., 2005; Touryan, Felsen, & Dan, 2005; Lochmann, Blanche, & Butts, 2013). In this model, one subunit consists of a linear filter

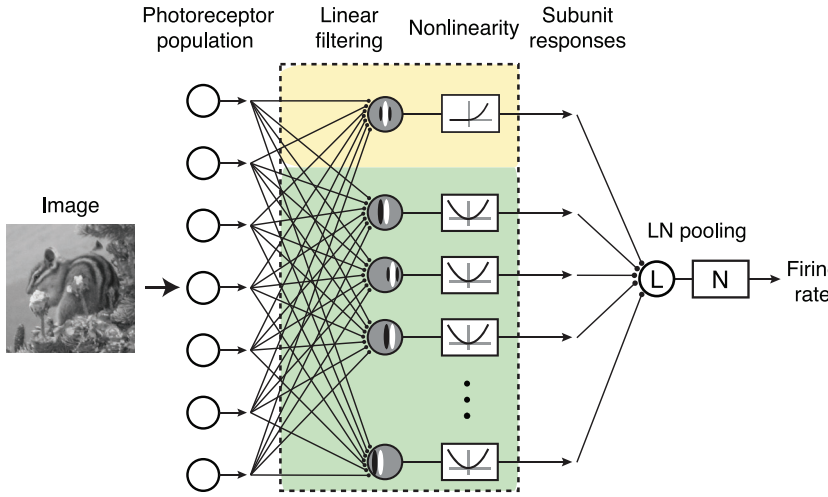


Figure 7: A generalized encoding description of V1 computation. Shown is the generalized LN-LN subunit model proposed to describe the conversion of visual images into the firing rate responses of individual V1 neurons. In the first LN stage, the stimulus is passed through a bank of linear filters followed by squaring, with the exception of the first subunit (which is half-squared). These responses are combined with a weighted sum, and the result is passed through a final nonlinearity. Note the similarity between the structure of this model and nQDA (see Figure 1b).

that is half-wave rectified and squared, the other subunits consist of linear filters followed by squaring, and all subunit responses are combined with a weighted sum, followed by a final response nonlinearity (see Figure 7). This model can be fit to neural data using spike-triggered covariance (Rust et al., 2005; Touryan et al., 2005; Lochmann et al., 2013), or more direct maximum likelihood methods (Vintch, Movshon, & Simoncelli, 2015). The structure of this generalized V1 model bears a remarkable resemblance to the nQDA model framework (compare Figure 1b and Figure 7): an LN-LN model in which one (linear) subunit combines with a bank of nonlinear subunits whose responses are squared.

Despite their structural similarity, the V1 subunit model and the nQDA computation aim to describe different phenomena: the V1 model is a single-neuron description of the transformation of a visual image into a firing rate response, whereas the nQDA framework is a population-level description of the conversion of an input population response into the solution for a predefined classification task. Nevertheless, a notable similarity between the models is that the parameters recovered by both procedures are not uniquely constrained. Consequently, care must be taken in interpreting both

the "subunits" of the V1 model and the parameters of the nQDA "model neurons" as biological elements. In the case of the V1 model, this can be observed empirically from the recovery of multilobed, physiologically implausible linear filters that span the same linear subspace as more plausible, localized, shifted subunits (Rust et al., 2005; Lochmann et al., 2013; Vintch et al., 2015). In the case of nQDA, the recovered "model neurons" are probably best regarded as "meta-neurons" that each represent something like a functional pool of individual neurons within the brain area of interest. In light of these issues, the advantage of both types of descriptions is that each provides an intuitive, quantitative account of an input-output transformation that can then be used to guide future experiments, as well as constrain more specific accounts of their biological implementation.

3.2 Links between nQDA and Other Proposed Transformations. While decoding has become a widely used tool for studying the representational content of neural populations, it has seldom been used as a substrate for explicit modeling of neural response. Linear classifiers are commonly used to describe decision tasks, and their physiological implementation is straightforward (i.e., a weighted sum followed by a threshold), but the degree to which the brain implements such decoders is not known, and even less is known about nonlinear population decoders. The quadratic nQDA decoder developed here has a complexity that is well matched to the computational capabilities of one stage of neural processing (e.g., a cortical area). Our results suggest that this type of decoder is sufficient to capture the transformation that perirhinal cortex performs on the inputs arriving from IT (see Figure 3c). The reformulation of QDA into an LN-LN framework also allowed us to arrive at an intuitive geometric description of the neural mechanisms used to create linear separability—that is, by finding the input dimensions with maximal variance differences and converting those into mean differences with squaring (see Figure 2). Finally, the LN-LN reformulation leads to a physiologically plausible contrastive Hebbian learning algorithm capable of approximating nQDA parameters (see Figure 6).

Our results demonstrate that nQDA provides a better account of the transformation between two high-level brain areas, IT and perirhinal cortex, than a number of comparable alternatives. Each transformation consisted of a linear transformation of the inputs, followed by a nonlinearity constrained to operate separately on each of the resulting responses. We found that a random linear transformation, using up to 1000 axes, failed to increase linear separability, suggesting that the randomly connected networks that have been proposed for other systems (Sussillo & Abbott, 2009; Caron, Ruta, Abbott, & Axel, 2013) are not a good description for these particular brain areas and this particular task. The inability of PCA to replicate the transformation (see Figure 3c) is largely explained by the fact that the response modulation in IT is primarily visual (Pagan et al., 2013), and thus PCA recovers dimensions along which the distributions to target matches

and distractors are highly overlapping, whereas increasing linear separability requires finding the dimensions along which the distributions of the two classes differ. The fact that ICA performs better than PCA can be understood in the context of the success of nQDA: directions along which the variance of two classes differs the most will also tend to have large kurtosis (since the distribution of inputs is a mixture of two distributions with very different variance), and thus are also likely to be found by ICA. However, nQDA did in fact outperform ICA, in part because of the additional information provided by the (supervised) class labels.

We evaluated the degree to which nQDA could replicate a two-way discrimination between target matches and distractors by assessing trial-level, cross-validated classification performance applied to data recorded from IT. While we found robust cross-validated performance across trials (see Figure 3c), nQDA applied to this data set does not generalize well across conditions (not shown). We suspect that this is because the data set is relatively small (16 total conditions), whereas generalization requires a considerable amount of data to accurately sample the covariance. While acquiring large data sets from the brains of awake, behaving animals is considerably more challenging than other situations in which generalization is typically applied (e.g., raw images or neural responses in animals that are passively viewing rapidly presented images, Cadieu et al., 2014; Yamins et al., 2014), we see expanding these data sets as an important future step of this modeling effort. We also note that our results do not guarantee that nQDA will provide an equally good account of other high-level transformations, such as the multiway classifications required for invariant object recognition (i.e., determining which of N possible objects is in view).

3.3 How might the Brain Learn the nQDA Transformation? We have demonstrated that the parameters of nQDA can be learned by neurons via a local, supervised rule, hQDA (see Figure 6). Our learning rule closely resembles classic Hebbian descriptions of synaptic plasticity (Oja, 1982; Sanger, 1989; Hebb, 2002), with the addition of an extra input that acts to switch the sign of the learning rule. Such an approach is reminiscent of the contrastive Hebbian (Hinton & Sejnowski, 1986) and contrastive divergence (Hinton, 2002) approaches that are generally adopted to train Boltzmann machines. Notably, this learning algorithm is supervised, and the best-performing unsupervised algorithm that we applied (ICA) did not perform well. However, unsupervised algorithms have been used successfully in contexts similar to the ones we describe here (Wiskott & Sejnowski, 2002; Serre, Wolf, & Poggio, 2005; Cadieu & Olshausen, 2008), and it remains unclear whether hQDA could be reformulated in an unsupervised context. We have also offered speculative suggestions of how hQDA might be implemented in the brain (see Figure 6e). One difficulty is the incorporation of Sanger's rule, which forces the nQDA axes to be orthogonal (see equation 2.15). In his original

proposal of the rule, Sanger (1989) speculated that this might be achieved by lateral or other competitive interactions between neurons.

3.4 The Value of Quantitative Descriptions at Higher Stages. Finally, the importance of techniques for fitting quantitative models to account for high-level computation is worth noting. While it is tempting to assume that the types of complex response properties exhibited by high-level brain areas result from neural computations that are themselves complex, this need not be the case. Rather, the responses of neurons in a high-level brain area reflect the net effect of computations up to that point of processing, and seemingly complex responses can arise from cascading many stages of simple computation. Consequently, determining the computations that a high-level brain area implements requires separating the response properties that a brain area inherits from its inputs from the ones that are computed de novo. The results we have presented here suggest that at least for the task of visual target identification, the computations implemented by perirhinal cortex bear a striking resemblance to the computations implemented at the earliest stage of visual cortical processing.

4 Methods

The experimental procedures involved in collecting our data are described in detail in Pagan et al. (2013) and briefly summarized here. All procedures were performed in accordance with the guidelines of the University of Pennsylvania Institutional Animal Care and Use Committee.

We recorded neural responses in IT and PRh as monkeys performed a delayed-match-to-sample task that required treating the same images as targets and as distractors on different trials (see Figure 3a). Monkeys initiated each trial by fixating a small dot. After a brief delay, a sample of the target image for that trial was presented, followed by a sequence of 0 to 3 distractors, and then by a target match. Monkeys were trained to maintain fixation during the presentation of the distractors and to make a saccade to a dot when the target match appeared. The same four images were used during all the experiments and were presented in all possible combinations as a visual stimulus and as a target, thus resulting in 16 conditions. Target matches that were presented after the maximal number of distractors ($n = 3$) occurred with 100% probability and were discarded from the analysis; all other conditions were included (e.g., distractors presented at the first, second, or third positions). For each condition, we collected at least 20 repeats on correct trials. Spikes were counted in a window 50 to 220 ms following stimulus onset.

4.1 Population Performance. To measure the amount and format of task-relevant information contained in each neural population, we performed a variety of cross-validated classification analyses of whether each

condition was a target match or a distractor (Pagan et al., 2013). For all analyses, we considered population response vectors of N neurons (where $N = 164$ in IT and PRh). The cross-validation procedure involved randomly assigning 18 trials from each condition to compute the representation (training set), setting aside one trial from each condition to optimize classifier parameters (parameter set), and finally using the remaining trial from each condition to test the performance of the classifier (test set). The number of target matches and distractors was matched on each iteration of the procedure (four from each class). Moreover, distractors were chosen to span all visual stimuli and all targets to avoid the possibility that classifiers could rely on visual or target information alone, leading to nine valid sets of four distractors. While we note that this changes the ratio of target matches and distractors for the classification analysis (4 target matches/4 distractors) as compared to the actual experiment (4 target matches/12 distractors), the monkeys' high performance on this task (averages of 94% and 92% for each of two monkeys) suggests that they were not simply relying on these suboptimal information sources (which would plateau at 66% correct) but rather computing the actual target match signal. The performance value for each iteration was computed as the mean of the eight test binary values (0 = wrong; 1 = correct), averaged across all nine valid choices of the distractor set. Mean and standard error were computed as the mean and standard deviation across 2000 iterations of the resampling procedure. To compare IT FLD and nQDA performance, we report a p-value as an evaluation of the probability that differences in the mean performance values that we observed were due to chance. We compute this probability as the fraction of 2000 iterations on which the difference was flipped in sign relative to the actual difference between the means of the full data set.

Before applying each classifier, the responses of each neuron were normalized to have zero mean and unit standard deviation across all training trials to ensure that the classifier parameters were assigned based on a combination of response magnitude and trial-by-trial variability rather than response magnitude alone. Responses could not be normalized before applying the maximum likelihood classifier because the classifier assumes inputs are Poisson distributed and nonnegative. We tested several types of classifiers (see Figure 3c).

4.1.1 FLD Linear Classifier. The expression for the FLD classifier is presented in equation 2.2. To minimize the impact of trial variability on our covariance estimates, we began by averaging the responses to each condition across the set of training trials; we then computed the means and covariances for the neural population using a regularized estimate equal to a linear combination of the sample covariance Σ and the identity matrix I:

$$\hat{\Sigma}_i = \gamma \cdot \Sigma_i + (1 - \gamma) \cdot I. \tag{4.1}$$

On each iteration of the resampling procedure, the regularizing parameter γ was optimized using the training set, and the γ that produced the best performance was chosen to compute the actual performance using the separately measured test set. The final FLD performance values were then computed by averaging the performances on the test set across all iterations.

4.1.2 Maximum Likelihood Classifier. The maximum likelihood classifier is described in detail in (Pagan et al., 2013). Briefly, we used the set of training trials to compute the average response r_{uc} of each neuron u to each condition c, and (consistent with our data) we modeled the likelihood that a test response k was generated from a particular condition as a Poisson-distributed variable:

$$L_{u,c}(k) = \frac{(r_{uc})^k \cdot e^{-r_{uc}}}{k!}. \tag{4.2}$$

The likelihood that a population response vector was generated in response to each condition was then computed as the product of the likelihoods of the individual neurons. Finally, we computed the likelihood that a test response vector arose from the category target match versus the category distractor as the mean of the likelihoods for target matches and distractors, respectively, and we assigned the classification label to the category with the higher likelihood.

4.1.3 nQDA. nQDA parameters were calculated as described in equations 2.4 to 2.6 using the regularized covariances described in equation 4.1.

4.1.4 Alternative Nonlinear Classifiers. We compared nQDA performance with three alternative nonlinear classifiers that (like nQDA) began by applying an initial linear transformation of the original IT space. We next describe the computation of (1) the linear weights (axes) for each type of classifier, (2) the nonlinearities applied to these linearly transformed responses, and (3) the final linear weighting used to combine the transformed responses into the classifier signal (which is then thresholded to obtain the classification response).

Linear weights. For all three alternative nonlinear classifiers, we computed the parameters for one linear model neuron in the same manner as nQDA to ensure that we preserved the linearly separable information that already existed in IT. The computation of random axes (see Figure 3c, IT random) involved selecting a 164-dimensional random rotation matrix (Mezzadri, 2006). When larger numbers of random axes were considered (see Figure 4), they were generated independently from a multivariate gaussian distribution with zero mean and unitary standard deviation. The PCA axes (see Figure 3c, IT PCA) were chosen as the eigenvectors of the covariance matrix

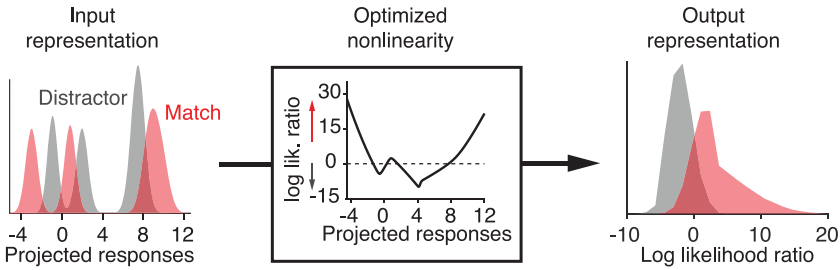


Figure 8: Computing the optimal nonlinearity for each alternative nonlinear classifier axis. The distributions of the responses to each condition, projected along each axis, were modeled as gaussian distributions and the responses to the set of target matches and set of distractors were each modeled by mixtures of these gaussian distributions. To compute an optimized nonlinearity that maximally separated the means of the target matches and distractors, we computed the log-likelihood ratio between the mixture of gaussians for matches and the mixture of gaussians for distractors. This optimized nonlinearity was computed using the training data and was applied to the test data following the application of different linear projections to obtain the results in Figure 3c.

computed from the combined responses to both target matches and distractors. The ICA axes (see Figure 3c, IT ICA) were chosen as those along which the kurtosis of the projected responses was maximal, and they were computed using the fastICA package (http://research.ics.aalto.fi/ica/fastica). Finally, the nQDA axes (see Figure 3c, IT nQDA) were computed as described in section 2.

Nonlinearities. Following the computation of the linear weights, an optimal nonlinearity was fit independently for each axis (with the exception of the linear axis) using the distribution of the projected responses for target matches and for distractors, obtained from the training data (see Figure 8). This nonlinearity is optimal in the sense that it is designed to maximally separate the output values for matches from the output values for distractors in a manner very similar to the ratio of gaussians method (Pillow & Simoncelli, 2006). To compute the optimal nonlinearity for a particular axis, we modeled the distribution of responses to each match condition and each distractor condition as a gaussian distribution, having the mean and the variance of the responses after the initial linear transformation. We then computed the log-likelihood ratio between the mixture of gaussians associated with matches and the mixture of gaussians associated with distractors. This procedure assigned positive values to responses that are likely to be matches and negative values to responses that are likely to be distractors.

The final linear read-out. For each alternative classifier, the weights of the final linear decoder were computed via the regularized FLD described by equation 4.1 (in contrast to pooling via the eigenvalues as described for

nQDA, equation 2.6). As described, computation of FLD weights included optimizing a single regularization parameter γ using the parameter set, whereas classifier performances were computed using the separately measured test set. Consequently, computation of both nQDA and each of these alternative nonlinear classifiers included the cross-validated optimization of a single regularization parameter.

4.2 Quantifying Neural Signals. To parse neural responses into different types of task-relevant signals, we applied the method described by Pagan and Rust (2014b). Briefly, because our experimental design included all possible combinations of each of 4 images presented as a target match and as a distractor, we can reexpress the 16-element vector reflecting the mean spike count of each unit to each condition as a weighted sum of 16 task-relevant signals by projecting it onto an orthonormal basis that we have designed specifically to capture different types of modulation (Pagan & Rust, 2014b; see Figure 1d) and then combining components that reflect the same type of modulation (e.g., changes in visual identity). Shown in Figure 5 are modulation magnitudes, each of which reflects spike count variance (around the grand mean) that result from changes in visual stimulus identity ("visual"), target identity ("target"), and whether each condition was a target match or a distractor ("target match"). These variances are normalized by each unit's trial variance, computed as the variance across the 20 repeated trials for each condition and then averaged across conditions, to obtain a unitless quantity that reflects the ratio of each type of signal relative to the noise.

4.3 Data-Inspired Simulation. To evaluate how well the hQDA contrastive Hebbian learning algorithm converged to nQDA parameters (see Figure 7c), we performed a data-inspired simulation that allowed us to generate the large number of trials necessary to evaluate convergence. Specifically, we computed the covariance matrices of the 167 IT neurons included in our data set across the four matches, and the covariance matrix across four distractors, and we then drew 167-dimensional vectors (up to 1000 for training and always 1000 for testing) from two multivariate gaussian distributions whose covariances matched those of the data and whose means were set to 0. This procedure was repeated 1000 times for each condition. On each iteration of the simulation, the four distractors were chosen randomly with the constraint of spanning all four visual stimuli and targets. We then used different numbers of training samples to fit the decoders, and we evaluated classification performances using the testing data (see Figure 7d).

With the exception of different numbers of training and testing trials, FLD and nQDA were implemented as previously described. To implement hQDA, the weights were randomly initialized (from a standard multivariate normal distribution) and were updated following the presentation of each training sample according to the learning rule described by equation 2.15.

The learning rate was decreased as the learning proceeded, with the learning rate η_i used for the nth training sample set to

$$\eta_i = \frac{\eta_0}{1 + \frac{n}{N}}, \tag{4.3}$$

where N is the total number of training samples and the initial learning rate η_0 was set to 0.001. Because our data contained four matches and four distractors and the mean of each distribution was set to zero, the dimensionality of the training data was equal to six. As a consequence, only six axes contained meaningful information to aid classification performance, and we thus restricted hQDA to learn six axes. More specifically, three axes were learned by adopting a positive Hebbian term for the match class and a negative Hebbian term for the distractor class (thus retrieving axes where the match variance was larger than the distractor variance), and three axes were learned using the opposite association (thus retrieving the axes where distractor variance was larger than match variance). Because the simulated data did not contain any differences between the means of the two distributions, hQDA was implemented here without the linear axis (see Figure 1b, yellow), thus focusing only on the squared axes (see Figure 1b, green). Following squaring of the hQDA responses, they were combined via a final FLD (trained as described above).

We also implemented a modified version of nQDA in which axes were computed as the eigenvectors of the difference of the two covariance matrices, as opposed the difference of the inverse covariances (see Figure 6c, black dashed). As with hQDA, we computed only the six axes associated with nonzero eigenvalues, we squared the resulting responses, and we combined the responses via a final FLD.

Acknowledgments

This work was supported by the National Eye Institute of the U.S. National Institutes of Health (award R01EY020851), the McKnight Endowment for Neuroscience, and the Howard Hughes Medical Institute. We thank Stefano Fusi and Josh Gold for helpful discussions.

References

Adelson, E. H., & Bergen, J. R. (1985). Spatiotemporal energy models for the perception of motion. *J. Opt. Soc. Am. A*, *2*(2), 284–299.

Akil, M., & Lewis, D. A. (1993). The dopaminergic innervation of monkey entorhinal cortex. *Cerebral Cortex*, *3*(6), 533–550.

Astrand, E., Enel, P., Ibos, G., Dominey, P. F., Baraduc, P., & Ben Hamed, S. (2014). Comparison of classifiers for decoding sensory and cognitive information from prefrontal neuronal populations. *PLoS One*, *9*(1), e86314.

Bialek, W., de Ruyter van Steveninck, R., Rieke, F., & Warland, D. (1996). *Spikes: Exploring the neural code*. Cambridge, MA: MIT Press.

Cadieu, C. F., Hong, H., Yamins, D. L., Pinto, N., Ardila, D., Solomon, E. A., ... DiCarlo, J. J. (2014). Deep neural networks rival the representation of primate IT cortex for core visual object recognition. *PLoS Computational Biology*, *10*(12), e1003963.

Cadieu, C., & Olshausen, B. A. (2008). Learning transformational invariants from natural movies. In D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou (Eds.), *Advances in neural information processing systems*, *21* (pp. 209–216). Cambridge, MA: MIT Press.

Carandini, M., & Heeger, D. J. (2011). Normalization as a canonical neural computation. *Nature Reviews Neuroscience*, *13*(1), 51–62.

Caron, S. J., Ruta, V., Abbott, L., & Axel, R. (2013). Random convergence of olfactory inputs in the Drosophila mushroom body. *Nature*, *497*(7447), 113–117.

Churchland, M. M., Cunningham, J. P., Kaufman, M. T., Foster, J. D., Nuyujukian, P., Ryu, S. I., & Shenoy, K. V. (2012). Neural population dynamics during reaching. *Nature*, *487*, 51–56. doi:10.10.1038/nature1129

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297.

David, S. V., Hayden, B. Y., & Gallant, J. L. (2006). Spectral receptive field properties explain shape selectivity in area V4. *J. Neurophysiol.*, *96*(6), 3492–3505.

DiCarlo, J. J., & Cox, D. D. (2007). Untangling invariant object recognition. *Trends Cogn. Sci.*, *11*(8), 333–341.

DiCarlo, J. J., Johnson, K. O., & Hsiao, S. S. (1998). Structure of receptive fields in area 3b of primary somatosensory cortex in the alert monkey. *Journal of Neuroscience*, *18*(7), 2626–2645.

Duda, R. O., Hart, P. E., & Stork, D. G. (2000). *Pattern classification* (2nd ed.). New York: Wiley.

Eggermont, J., Aertsen, A., & Johannesma, P. (1983). Quantitative characterisation procedure for auditory neurons based on the spectrotemporal receptive field. *Hearing Research*, *10*(2), 167–190.

Enroth-Cugell, C., & Robson, J. G. (1966). The contrast sensitivity of retinal ganglion cells of the cat. *Journal of Physiology*, *187*(3), 517–552.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, *7*(2), 179–188.

Graf, A. B., Kohn, A., Jazayeri, M., & Movshon, J. A. (2011). Decoding the activity of neuronal populations in macaque primary visual cortex. *Nat. Neurosci.*, *14*(2), 239–245.

Hebb, D. O. (2002). *The organization of behavior: A neuropsychological theory.* Florence, KY: Psychology Press.

Heeger, D. J. (1992). Normalization of cell responses in cat striate cortex. *Visual Neuroscience*, *9*(2), 181–197.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation*, *14*(8), 1771–1800.

Hinton, G. E., & Sejnowski, T. J. (1986). *Learning and relearning in Boltzmann machines.* Cambridge, MA: MIT Press.

Hofmann, T., Schölkopf, B., & Smola, A. J. (2008). Kernel methods in machine learning. *Annals of Statistics*, *36*, 1171–1220.

Hung, C. P., Kreiman, G., Poggio, T., & DiCarlo, J. J. (2005). Fast readout of object identity from macaque inferior temporal cortex. *Science*, *310*(5749), 863–866.

Izhikevich, E. M. (2007). Solving the distal reward problem through linkage of STDP and dopamine signaling. *Cereb. Cortex*, *17*(10), 2443–2452.

Jones, J. P., & Palmer, L. A. (1987). The two-dimensional spatial structure of simple receptive fields in cat striate cortex. *Journal of Neurophysiology*, *58*, 1187.

Keat, J., Reinagel, P., Reid, R. C., & Meister, M. (2001). Predicting every spike: A model for the responses of visual neurons. *Neuron*, *30*(3), 803–817.

Kendall, M. G. (1966). Discrimination and classification. In *Proc. Symp. Mult. Analysis*. New York: Academic Press.

Law, C.-T., & Gold, J. I. (2009). Reinforcement learning can account for associative and perceptual learning on a visual-decision task. *Nature Neuroscience*, *12*(5), 655–663.

Lochmann, T., Blanche, T. J., & Butts, D. A. (2013). Construction of direction selectivity through local energy computations in primary visual cortex. *PloS One*, *8*(3), e58666.

Mante, V., Sussillo, D., Shenoy, K. V., & Newsome, W. T. (2013). Context dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, *503*(7474), 78–84.

Mezzadri, F. (2006). *How to generate random matrices from the classical compact groups*. arXiv preprint math-ph/0609050

Mineault, P. J., Khawaja, F. A., Butts, D. A., & Pack, C. C. (2012). Hierarchical processing of complex motion along the primate dorsal visual pathway. *Proceedings of the National Academy of Sciences*, *109*(16), E972–E980.

Oja, E. (1982). Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, *15*(3), 267–273.

Pagan, M., & Rust, N. C. (2014a). Dynamic target match signals in perirhinal cortex can be explained by instantaneous computations that act on dynamic input from inferotemporal cortex. *Journal of Neuroscience*, *34*(33), 11067–11084.

Pagan, M., & Rust, N. C. (2014b). Quantifying the signals contained in heterogeneous neural responses and determining their relationships with task performance. *J. Neurophysiol.*, *112*(6), 1584–1598.

Pagan, M., Urban, L. S., Wohl, M. P., & Rust, N. C. (2013). Signals in inferotemporal and perirhinal cortex suggest an untangling of visual target information. *Nat. Neurosci.*, *16*(8), 1132–1139.

Pillow, J. W., Shlens, J., Paninski, L., Sher, A., Litke, A. M., Chichilnisky, E., & Simoncelli, E. P. (2008). Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature*, *454*(7207), 995–999.

Pillow, J. W., & Simoncelli, E. P. (2006). Dimensionality reduction in neural models, an information-theoretic generalization of spike-triggered average and covariance analysis. *Journal of Vision*, *6*(4), 9.

Richmond, B. J. (2006). Dopamine-dependent associative learning of workload-predicting cues in the temporal lobe of the monkey. In R. Pinaud, L. A. Tremere, & P. De Weed (Eds.), *Plasticity in the visual system* (pp. 307–320). New York: Springer.

Rigotti, M., Barak, O., Warden, M. R., Wang, X. J., Daw, N. D. . . . Fusi, S., (2013). The importance of mixed selectivity in complex cognitive tasks. *Nature*, *497*(7451), 585–590.

Ringach, D. L., & Shapley, R. (2004). Reverse correlation in neurophysiology. *Cognitive Science*, *28*, 147–166.

Rust, N. C., Mante, V., Simoncelli, E. P., & Movshon, J. A. (2006). How MT cells analyze the motion of visual patterns. *Nat. Neurosci.*, *9*(11), 1421–1431.

Rust, N. C., Schwartz, O., Movshon, J. A., & Simoncelli, E. P. (2005). Spatiotemporal elements of macaque V1 receptive fields. *Neuron*, *46*(6), 945–956.

Sanger, T. D. (1989). Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks*, *2*(6), 459–473.

Schwartz, O., Pillow, J. W., Rust, N. C., & Simoncelli, E. P. (2006). Spike-triggered neural characterization. *Journal of Vision*, *6*(4), 13.

Serre, T., Wolf, L., & Poggio, T. (2005). Object recognition with features inspired by visual cortex. In *Proceedings of the 2005 Computer Vision and Pattern Recognition*. Washington, DC: IEEE Computer Society.

Sharpee, T. O. (2013). Computational identification of receptive fields. *Annual Review of Neuroscience*, *36*, 103–120.

Sharpee, T. O., Kouh, M., & Reynolds, J. H. (2013). Trade-off between curvature tuning and position invariance in visual area V4. *Proc. Natl. Acad. Sci. U.S.A.*, *110*(28), 11618–11623.

Sharpee, T. O., Sugihara, H., Kurgansky, A. V., Rebrik, S. P., Stryker, & Miller, K. D. (2006). Adaptive filtering enhances information transmission in visual cortex. *Nature*, *439*(7079), 936–942.

Soltani, A., & Wang, X. J. (2006). A biophysically based neural model of matching law behavior: Melioration by stochastic synapses. *J. Neurosci.*, *26*(14), 3731–3744.

Sussillo, D., & Abbott, L. F. (2009). Generating coherent patterns of activity from chaotic neural networks. *Neuron*, *63*(4), 544–557.

Touryan, J., Felsen, G., & Dan, Y. (2005). Spatial structure of complex cell receptive fields measured with natural images. *Neuron*, *45*(5), 781–791.

Vintch, B., Movshon, J. A., & Simoncelli, E. P. (2015). A convolutional subunit model for neuronal responses in macaque V1. *J. Neurosci.*, *35*(44), 14829–14841.

Willmore, B. D., Prenger, R. J., & Gallant, J. L. (2010). Neural representation of natural images in visual area V2. *Journal of Neuroscience*, *30*(6), 2102–2114.

Wiskott, L., & Sejnowski, T. (2002). Slow feature analysis: Unsupervised learning of invariances. *Neural Computation*, *14*(4), 715–770.

Wu, M. C., David, S. V., & Gallant, J. L. (2006). Complete functional characterization of sensory neurons by system identification. *Annu. Rev. Neurosci.*, *29*, 477–505.

Yamins, D. L., Hong, H., Cadieu, C. F., Solomon, E. A., Seibert, D., & DiCarlo, J. J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, *111*, 8619–8624.

Yu, B. M., Kemere, C., Santhanam, G. A., Afshar, S. I. Ryu, T. H. Meng, . . . Shenoy, K. V., (2007). Mixture of trajectory models for neural decoding of goal-directed movements. *J. Neurophysiol.*, *97*(5), 3763–3780.

Received April 7, 2016; accepted June 29, 2016.