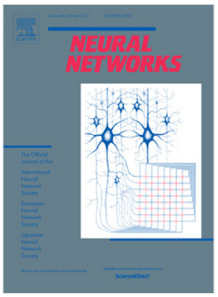
Accepted Manuscript



A video-driven model of response statistics in the primate middle temporal area

Omid Rezai, Pinar Boyraz Jentsch, Bryan Tripp

PII: S0893-6080(18)30266-1
DOI: https://doi.org/10.1016/j.neunet.2018.09.004
Reference: NN 4033

To appear in: *Neural Networks*

Received date : 24 April 2018
Revised date : 20 July 2018
Accepted date : 6 September 2018

Please cite this article as: Rezai, O., Jentsch, P.B., Tripp, B., A video-driven model of response statistics in the primate middle temporal area. *Neural Networks* (2018), https://doi.org/10.1016/j.neunet.2018.09.004

***Title Page (With all author details listed)**

A Video-Driven Model of Response Statistics in Primate Middle Temporal Area

Omid Rezai[1,2], Pinar Boyraz Jentsch[3,4], Bryan Tripp[1,2]

[1]Department of Systems Design Engineering, University of Waterloo, Canada

[2]Centre for Theoretical Neuroscience, University of Waterloo, Canada

[3]BAST GmbH, Heidelberg, Germany

[4]Cognitive Neuroscience Laboratory, German Primate Center, Leibniz Institute for Primate Research, Goettingen, Germany

Corresponding Author:

Omid Rezai

Department of Systems Design Engineering

Engineering 5, 6th Floor

University of Waterloo

200 University Avenue West

Waterloo, Ontario, Canada N2L 3G1

Email: omid.srezai@uwaterloo.ca

A Video-Driven Model of Response Statistics in the Primate Middle Temporal Area

Omid Rezai[a,b], Pinar Boyraz Jentsch[c,d], Bryan Tripp[a,b]

[a] *Department of Systems Design Engineering, University of Waterloo, Canada*
[b] *Centre for Theoretical Neuroscience, University of Waterloo, Canada*
[c] *BAST GmbH, Heidelberg, Germany*
[d] *Cognitive Neuroscience Laboratory, German Primate Center, Leibniz Institute for Primate Research, Goettingen, Germany*

Abstract

Neurons in the primate middle temporal area (MT) encode information about visual motion and binocular disparity. MT has been studied intensively for decades, so there is a great deal of information in the literature about MT neuron tuning. In this study, our goal is to consolidate some of this information into a statistical model of the MT population response. The model accepts arbitrary stereo video as input. It uses computer-vision methods to calculate known correlates of the responses (such as motion velocity), and then predicts activity using a combination of tuning functions that have previously been used to describe data in various experiments. To construct the population response, we also estimate the distributions of many model parameters from data in the electrophysiology literature. We show that the model accounts well for a separate dataset of MT speed tuning that was not used in developing the model. The model may be useful for studying relationships between MT activity and behavior in ethologically relevant tasks. As an example, we show that the model can provide regression targets for internal activity in a deep convolutional network that performs a visual odometry task, so that its representations become more physiologically realistic.

Keywords: area MT, dorsal visual stream, visual representations, statistical model, deep convolutional networks, visual odometery

1. Introduction

The middle temporal cortex (MT) receives strong feedforward input from early visual areas V1, V2, and V3 (Maunsell and Van Essen, 1983; Markov et al., 2014), as well as direct sub-cortical input (Sincich et al., 2004; Born and Bradley, 2005). It projects to the higher-level middle superior temporal and ventral intraparietal areas, and also receives strong feedback connections from these. Electrical stimulation of MT affects perception of visual motion (Nichols and Newsome, 2002). Inactivation or damage of MT impairs motion perception (Newsome and Pare, 1988; Rudolph and Pasternak, 1999) and the ability to smoothly follow a moving object with the eyes (Newsome et al., 1985). Illusions in speed perception have also been linked with subtle properties of MT neuron responses (Boyraz and Treue, 2011).

Consistent with these effects, many neurons in MT respond strongly to visual motion. The spike rates of individual MT neurons vary with a number of stimulus features, including direction and speed of visual motion, and binocular disparity. Many MT neurons are sensitive to motion in depth, i.e. toward or away from the eyes (Czuba et al., 2014). MT is the earliest visual region in which a substantial number of neurons solve the motion "aperture problem", responding to the actual direction of motion of a stimulus, rather than the component of motion that is orthogonal to local edges, which requires only local computations (Pack and Born, 2001; Smith et al., 2005). In summary, MT exhibits a particular representation of visual motion, which is similar in scope to scene flow (Mayer et al., 2015).

Although much is known about this representation, and its causal role in visual motion perception, some aspects of the relationship between the representation and ethologically relevant functions are less clear. For example, the accuracy of smooth-pursuit eye movement, self-motion perception, and motion-based segmentation may be sensitive to particular tuning properties or population statistics, in addition to artificial disruptions of MT activity. Computational models can be used to study such relationships, and sophisticated computational models of MT responses have been developed (Nishimoto and Gallant, 2011; Baker and Bair, 2016). However, we wondered if a new model could be developed that spans a more comprehensive range of MT response phenomena, and captures MT response statistics in more detail. Rather than building on existing mechanistic models of MT, we instead pursued an empirical model, in which we directly specify the neurons' tuning curves. This approach allows us to approximate the response statistics in

almost arbitrary detail, without requiring a complete understanding of how these responses arise in the brain.

1.1. Deep representations and realistic neural function

A potential application of this model is to make the internal representations of deep networks more physiologically realistic. In general, deep learning may facilitate development of visual cortex models with more ethologically realistic functions. For example, various deep networks excel in scene segmentation (Chen et al., 2016, 2017; He et al., 2017), depth estimation from stereo disparity (Žbontar and LeCun, 2016), and scene flow (Mayer et al., 2015). The internal visual representations of deep networks that have been trained for object recognition have striking relationships with representations in the ventral stream (Yamins et al., 2014; Cadieu et al., 2014; Khaligh-Razavi and Kriegeskorte, 2014; Hong et al., 2016; Yamins and DiCarlo, 2016) (relatedly, Güçlü and van Gerven (2017) found that action-recognition CNNs were predictive of function magnetic resonance imaging data from the dorsal stream), but there are also striking differences (Tripp, 2017).

Our empirical model of MT may provide regression targets for intermediate network layers, helping to impose a physiologically realistic representation. A related approach was taken by Arai et al. (1994), who optimized a two-layer network model of superior colliculus with two cost terms, one (applied to the output) related to the task, and the other (applied to the hidden layer) derived from neural activity. More recently, Yamins et al. (2014) trained deep networks to approximate recordings of neurons in the inferotemporal cortex. The resulting networks accounted for much of the variance in held-out inferotemporal neural data. Interestingly, IT predictions of similar quality were obtained simply by training the networks for object recognition, although the neural dataset was small enough (5760 images) that overfitting was possible in this case. Other groups have reported good results training deep networks to emulate neural recordings in the retina (McIntosh et al., 2016), V1 (Kindel et al., 2017; Klindt et al., 2017; Cadena et al., 2017), and V4 (Oliver and Gallant, 2016). McIntosh et al. (2016) found that a deep network generalized better across stimulus types than other models, and also reproduced sub-Poisson noise scaling found in the retina. We previously trained an intermediate layer of a deep network, which had motion-energy components in a lower layer, to emulate a simplified empirical model of MT activity, and then trained the full network to estimate self-motion speed and direction from video (Tripp, 2016). We extend this approach with the present

(more detailed) empirical model in section 2.8. This approach has advantages and disadvantages compared to using MT data directly for regression targets. The main disadvantage is reduced physiological validity. Advantages are the possibility of unlimited training data, and the ability to directly manipulate tuning statistics, to allow detailed exploration of the relationships between representations and and behavior.

2. Methods

2.1. Structure of the empirical model

Our model produces approximations of MT spike rates directly from input video. We focus on producing spike rates, rather than spike sequences. As an aside, given these rates, it is straightforward to produce Poisson spike sequences (Dayan and Abbott., 2001), including those with noise correlations that are realistic for MT (Tripp, 2012).

The model structure is sketched in Figure 1. The model requires five fields as input. The field values are defined at each image pixel x, y. The five fields are $u(x, y)$ (horizontal flow velocity), $v(x, y)$ (vertical flow velocity), $d(x, y)$ (disparity), $c(x, y)$ (contrast), and $a(x, y)$ (attention). Section 2.2 below discusses calculation of these fields.

The response of each neuron is approximated as a nonlinear-linear-nonlinear (NLN) function of these fields. The first nonlinear step requires calculation of four additional fields for each neuron, each of which is a point-wise nonlinear function of the five input fields. We refer to these functions as tuning functions (see details in Section 2.3). Each of these tuning functions is used to scale the neuron's response to a different stimulus feature. Specifically, we calculate $g_s(u, v, c)$ (a function of flow speed and contrast), $g_\theta(u, v)$ (a function of flow direction), $g_d(d)$ (a function of disparity), and $g_g(a, c)$ (a function of attention and contrast). Whereas the first five fields are correlates of MT responses (e.g. velocity), these additional fields represent nonlinear tuning functions of these correlates. In the excitatory part of a unit's receptive field, each of these fields has a monotonic relationship with spike rates when other fields are held constant.

The full model therefore requires calculation of four times as many of these tuning-function fields as there are neurons with distinct sets of parameters. The model has uniform response statistics across the visual field (similar to convolutional networks), so there is one such set of parameters per distinct response channel in the MT layer. This number can be specified at run time,

but we would expect it to normally be on the order of 100-1000, therefore 400-4000 of these fields must be calculated by the full model. One additional field per neuron is then calculated as the point-wise product of these fields (consistent with data from Rodman and Albright, 1987; Treue and Martínez Trujillo, 1999). We refer to this as the neuron's tuning field,

$$t(x, y) = g_s g_\theta g_d g_g. \tag{1}$$

This completes the first nonlinear stage of the NLN model. Similar to convolutional networks, only one tuning field is needed per channel (feature map), corresponding to a set of model parameters, regardless of the pixel dimensions of the channel. Henceforward, when we talk about a "neuron model", it should be understood that this "neuron model" is ultimately tiled across the visual field to simulate many neurons with different receptive field centers.

The remaining linear and nonlinear steps consist of a conventional convolutional layer, with one channel per MT neuron (we specify the number of MT neurons at instantiation time, and choose parameters for each one as discussed below in section 2.4.2). Kernels combine tuning-field values $t(x, y)$ over a receptive field. However (in contrast with typical convolutional layers with learned kernels), kernels are parameterized to resemble MT receptive fields. The kernels include excitatory, direction-selective suppressive, and non-selective suppressive components. Such components have been found to account well for MT responses to complex motion stimuli (Cui et al., 2013). The excitatory component of the kernel models the neuron's classical receptive field. This component has positive weights and a Gaussian structure, which is elongated so that the axis of elongation is orthogonal to the neuron's preferred direction (Raiguel et al., 1995). It spans a single channel of the tuning-field layer, and therefore has a speed and direction selectivity that match that channel. The direction-selective suppressive component also spans a single tuning-function channel. It has negative weights, and is also modeled as a Gaussian function. Relative to the excitatory kernel, it can be symmetrically larger, or elongated, or offset. For each neuron, we draw at random from these spatial relationships with the proportions reported by Xiao et al. (1997). The preferred direction of this suppressive component is generally different from that of the excitatory component. We draw this difference from the distribution in Cui et al. (2013) (their Figure 5). Finally, the non-direction-selective suppressive component receives the same tuning-function channel with g_θ removed. It has negative weights and an annular structure that we model as a rectified difference of Gaussians. The full kernel

is the sum of these components. When we fit tuning curves for speed, disparity, and direction tuning in response to stimuli that are spatially uniform in these properties, we simplify the kernels as broad Gaussian functions.

The final nonlinearity is,

$$f(x) = [Ax + B]_+^n \,, \tag{2}$$

composed of a half-wave rectification ($[]_+$) followed by a power function ($[]^n$). A and B are a scaling factor and a background spike rate, respectively.

We have chosen this form for our model (versus other possible forms with different orders of the linear and nonlinear parts), because the linear kernel must follow at least some of the tuning curves for consistency with data from Majaj et al. (2007) (see our Figure 7). Also to avoid negative spike rates due to inhibitory surrounds, the final rectifying nonlinearity must come after the linear kernel.

2.1.1. Eccentricity and Receptive Field Size

The visual cortex differs from convolutional networks in that the receptive fields of neurons in many visual areas scale almost linearly with eccentricity (visual angle from the fovea). This difference could be reduced by remapping the input images. However, to simplify use of the model with standard uniform-resolution videos, we instead model the whole visual field uniformly, as is typical in convolutional networks. There is also variation in receptive field sizes at any given eccentricity. We modelled the spread of receptive field sizes on parafoveal receptive fields (2-10 degree eccentricity) from Figure 2 of Maunsell and van Essen (1987).

2.2. Input Fields

The model requires contrast, attention, optic flow, and binocular disparity fields.

2.2.1. Contrast

The contrast field is calculated using the definition of Peli (1990). This is is a local, band-limited measure, in contrast with other notions of contrast (e.g. root-mean-squared luminance) that are global and frequency-independent. A local definition is needed to modulate neuron responses according to contrast within their receptive fields (as opposed to remote parts of the image). Frequency dependence allows us to match the contrast definition to primate contrast sensitivity (Robson, 1966; De Valois et al., 1982a).

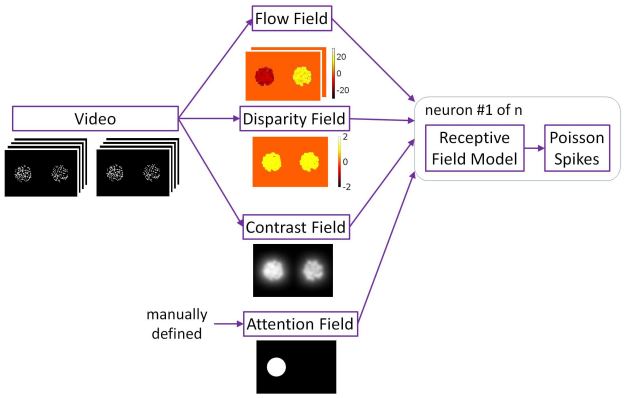


Figure 1: Model structure. The model uses nonlinear-linear-nonlinear models to approximate neuron responses as functions of optic flow, contrast, disparity, and attention fields. Optic flow, contrast, and disparity are calculated from input images, as described in the text. An example of these fields can be seen for a video input with two patches of random dots moving in opposite directions (i.e., up and down; with far disparity) where the left patch was attended. Units for flow and disparity maps are deg/sec and deg. Poisson spikes can optionally be generated at the estimated spike rates to emulate neural activity more closely, but they are not used in this paper.

In Peli's definition, contrast at each spatial frequency band (i) is defined as a ratio of two functions,

$$c_i(x,y) = \frac{\alpha_i(x,y)}{l_i(x,y)}. \tag{3}$$

The numerator function is,

$$\alpha_i(x,y) = I(x,y) * g_i(x,y), \tag{4}$$

where I is the image, g_i is a spatial frequency dependent filter, and $*$ denotes convolution. The denominator function is,

$$l_i(x,y) = \bar{I} + \sum_{j=1}^{i-1} \alpha_j(x,y), \tag{5}$$

where $\bar{I}$ is the image mean. Peli suggested cosine log filters as the choice for g_is since an image filtered by a bank of these filters can be reconstructed by a simple addition process without distortion. However, to relate the contrast definition more directly to V1, we instead used a bank of Gabor filters with four different frequencies and four different orientations for a total of 16 contrast channels. We combined these channels in a weighted sum:

$$c'(x,y) = \sum_{k=1}^{16} A_k c_k(x,y), \tag{6}$$

where $A_k s$ were chosen to approximate macaque contrast sensitivity (De Valois et al., 1982a,b).

We then smoothed the resulting contrast field with a 2D Gaussian kernel, which was meant to approximate integration over V1 cells, and scaled it so that its mean over the image was equal to the root-mean-squared contrast measure:

$$c(x,y) = A_{scale} gaussfilt(c'(x,y)). \tag{7}$$

2.2.2. Attention

Attention is typically driven by task demands, so in general it can not be derived from images alone (in contrast with saliency). Recent models approximate top-down influences (Borji and Itti, 2013). However, in the context of training neural networks that have attention mechanisms (e.g. Xu

et al., 2015), the attention field should ideally be defined by the network itself, to align attention modulation of activity with the network's focus of attention. Therefore we treated the attention field as an input to the model. To test the model, and to compare its output with electrophysiology data, we manually defined attended stimulus regions by drawing polygons around them in a custom user interface.

2.2.3. Flow and disparity fields

Flow and disparity fields were calculated using computer-vision algorithms. Specifically, we used the Lucas-Kanade method (Lucas and Kanade, 1981) to estimate both optic flow and disparity from images. This generally produced good fits to MT data (see Results).

The classical Lucas-Kanade algorithm does not capture large displacements, but this limitation is addressed by a multi-scale version of the algorithm (Marzat et al., 2009). In this version, the Gaussian pyramids method is used to repeatedly halve the image resolution. Flow or disparity is then estimated at the lowest resolution first. Then at each finer resolution, the immediate lower-resolution estimate is used to warp the earlier image, and the Lucas-Kanade algorithm is used to find residual differences between the warped earlier image and the later image. The multi-scale version of the algorithm also helps to solve the aperture problem, since it finds estimates that are consistent with global motion apparent in downsampled images. We typically used the multiscale algorithm in our simulations, with 3-5 scales. To simulate combined local and pattern motion selectivity (Pack and Born, 2001), we mixed the outputs of single-scale and multi-scale versions of the algorithm.

We also explored a variety of other algorithms for flow and disparity estimation, including semi-global matching (Hirschmuller, 2005), Classic++ (Sun et al., 2010), loopy belief propagation on a Markov random field (Felzenszwalb and Huttenlocher, 2006), and a convolutional neural network Žbontar and LeCun (2016). Several of these methods extrapolated far beyond well-textured regions, e.g. reporting motion over the whole image in response to a small stimulus. We interpreted this as being physiologically unrealistic, because it involves lateral communication over the whole visual field. However it does not actually expand the units' classical receptive fields unrealistically, because there is no response at zero contrast (see Equations 14). For our experiments, we used the Lucas-Kanade with pyramids, because it is simple and well established, and we did not find other methods to provide substan-

tial advantages within the scope of this paper. However, future work may reveal such advantages.

2.3. Tuning Functions

Given these fields, the next step in approximating a neuron's activity was calculation of a new four-channel image that consisted of pixel-wise nonlinear functions of the fields. Specifically, we calculated $g_s(u, v, c)$ (a function of flow speed and contrast), $g_\theta(u, v)$ (a function of flow direction), $g_d(d)$ (a function of disparity), and $g_g(a, c)$ (a function of attention and contrast). These functions were adopted from previous studies, as described below.

2.3.1. Speed Tuning

We used a contrast-dependent speed tuning function, (Nover et al., 2005),

$$g_s = \exp\left(-\frac{[\log\big(q(s,c)\big)]^2}{2\sigma_s^2}\right), \tag{8}$$

where,

$$q(s,c) = \frac{s + s_0}{s_p(c) + s_0}, \tag{9}$$

$s = \sqrt{u^2 + v^2}$ is motion speed, s_p is the preferred speed. The tuning curve has parameters s_0 (offset) and σ_s (width). Preferred speed is a function of contrast,

$$s_p(c) = \frac{A_p c}{c + B_p}, \tag{10}$$

where c is contrast at each pixel (Equation 7) and A_p and B_p are additional parameters that define a saturating dependence of preferred speed on contrast.

When stimulated with sinusoidal gratings, about a quarter of MT neurons show selectivity for certain spatial and temporal frequencies, rather than speed (defined as the ratio between spatial and temporal frequencies) (Priebe et al., 2003). Another quarter of MT neurons are selective to grating speed, regardless of its spatiotemporal components, and the remaining neurons form a continuum between these two behaviors. A similar distribution is also observed in V1 (Priebe et al., 2006). However, more complex stimuli containing a broader spectrum of frequencies, e.g. random dot fields, elicit in MT selective responses to speed. Since our goal was to apply this model

on naturalistic stimuli, which have broad frequency contents, we included speed tuning and ignored selectivity for spatial and temporal frequencies in the model.

2.3.2. Direction Tuning

Direction tuning was modeled as (Wang and Movshon, 2016),

$$g_\theta = \exp\left(\frac{cos\left(\theta - \theta_p\right) - 1}{\sigma_\theta}\right) + a_n \exp\left(\frac{cos\left(\theta - \theta_p - \pi\right) - 1}{\sigma_\theta}\right), \tag{11}$$

where $\theta = atan2(v, u)$ is motion direction, θ_p, σ_θ, and a_n are the preferred direction, direction width, and relative amplitude in null direction (i.e. 180 degrees away from preferred direction), respectively.

2.3.3. Disparity Tuning

Similarly, disparity tuning was modeled using Gabor functions (DeAngelis and Uka, 2003),

$$g_d = \exp\left(\frac{-\left(d - d_p\right)^2}{2\sigma_d^2}\right) \cos\left(2\pi f_d(d - d_p) + \phi_d\right), \tag{12}$$

where d_p and σ_d set the center and width of the Gaussian component and f_d and ϕ_d are the frequency and phase of the oscillatory component.

2.3.4. Attention and Contrast

Lastly, the gain function was (Treue and Martínez Trujillo, 1999; Martınez-Trujillo and Treue, 2002),

$$g_g(a, c) = \begin{cases} A_g g_c(c), & \text{if } a = 1 \\ \\ g_c(c), & \text{if } a = 0 \end{cases} \tag{13}$$

where A_g is the attentional gain and g_c, is the contrast response function defined as:

$$g_c(c) = \frac{A_c c^n}{c^n + B_c}, \tag{14}$$

where A_c and B_c are the contrast gain, contrast offset, contrast exponent, and c is contrast at each pixel (Equation 7).

2.3.5. Binocular Interactions

In many of the electrophysiology experiments that inform the model, monkeys were free to converge their eyes on a single, flat computer display, with constant (near zero) binocular disparity. However in a more complex environment, some MT neurons are tuned for motion-in-depth (Czuba et al., 2014). To account for such 3D motion encoding of MT neurons, we extended our model by modifying Equation 2 as,

$$f(x) = [A_L x_L + A_R x_R + B]_+^n, \tag{15}$$

where A_L and A_R are left and right eye gains, and x_L and x_R are weighted sums of tuning functions in left and right eye respectively.

A limitation is that the model of motion-in-depth is not realistically integrated with the model of binocular disparity. To retain realistic disparity tuning, we simply used our disparity tuning field, and identical disparity tuning curves in each eye, so that disparity and motion-in-depth tuning are orthogonal.

2.4. Model Fitting

2.4.1. Tuning Curve Fits

To test the model, we fit various tuning curves from the electrophysiology literature using Matlab's nonlinear least-squares curve fitting function, *lsqcurvefit* (trust-region-reflective algorithm). The fitting procedure for a given tuning curve selected the parameters of the relevant tuning functions (e.g. $g_s(u, v, c)$), along with parameters A and B of Equation 2. As the optimization was non-convex, we initiated it from at least 100 different starting points for each neuron, and took the most optimal answer.

This approach was designed to have a high success rate, in order to reliably support development of a rich statistical model of MT activity. Aside from failures of the optimization procedure (which we minimized by restarting from many initial parameter values), the approach has two potential failure modes. The first would arise from a poor choice of nonlinear function, however we chose functions that are well supported by previous work. The second would be a failure of the computer vision algorithms to estimate the relevant parameters from the images. We generally had good results with the Lucas-Kanade algorithm (see Results).

2.4.2. Parameter Distributions

We drew the neurons' tuning parameters from statistical distributions that were based on histograms and scatterplots in various MT electrophysiology papers. The model required distributions of preferred disparity, preferred speed, speed tuning width, attentional index (Treue and Martínez Trujillo, 1999), and a number of other tuning properties. As a first step in approximating these distributions, we extracted histograms and scatterplots of various tuning properties from the literature using Web Plot Digitizer (http://arohatgi.info/WebPlotDigitizer/). We then modelled each histogram using either a standard distribution (one of Gaussian, log-Gaussian, Gaussian mixture, gamma, t location-scale, exponential, and uniform), or the Parzen-window method (Parzen, 1962). For Parzen-window method, we selected the bandwidths using Silverman's rule of thumb (Silverman, 1986). In each case, we chose the distribution model that minimized the Akaike Information Criterion (Akaike, 1974). The parameter distributions are summarized in Table 1.

2.4.3. Correlation between Model Parameters

To make our model more realistic, we looked for studies that examined the correlation between the tuning parameters in area MT. Bradley and Andersen (1998) found that the center-surround effects of disparity and direction are mainly independent of each other, supporting the way we combine them over the MT receptive field. In another study, DeAngelis and Uka (2003) did not find a correlation between direction and disparity tuning parameters. They reported a non-zero correlation between speed and disparity tuning (neurons with higher speed preference tend to have weak and broad disparity tuning). However, this correlation was weak (see their Figure 11.A) and therefore we ignored it in our model. They also found a correlation between the preferred disparity and the disparity phase of the neurons whose preferred disparity is close to zero. We included this correlation by modelling the conditional distribution of disparity phase given the preferred disparity.

2.5. Dynamics of Component and Pattern Selectivity

The neurophysiology of the aperture problem in optic flow has been studied with overlapping pairs of drifting sinusoidal gratings at different angles, which together form a percept of a plaid pattern moving in an intermediate direction. MT is the earliest visual area to solve the aperture problem, in the sense that many MT neurons respond to the direction of the plaid pattern

Table 1: Distribution families used for various tuning parameters, and sources in the literature from which distributions were estimated. The number in the bracket specifies the dimension of a parameter, for those that have more than one.

Parameter	Distribution	Source
Preferred direction	Uniform	DeAngelis and Uka (2003)
Direction bandwidth	Gamma	Wang and Movshon (2016)
Null-direction amplitude	t location-scale	Maunsell and Van Essen (1983)
Preferred speed	Log uniform	Nover et al. (2005)
Speed width	Gamma	Nover et al. (2005)
Speed offset	Gamma	Nover et al. (2005)
Attentional index	t location-scale	Treue and Martínez Trujillo (1999)
Contrast influence on preferred speed (2)	2D Gaussian mixture	Pack et al. (2005)
Contrast influence on gain (3)	Conditional on attentional index	Martınez-Trujillo and Treue (2002)
Preferred disparity	t location-scale	DeAngelis and Uka (2003)
Disparity frequency	Log normal	DeAngelis and Uka (2003)
Disparity phase	Gaussian mixture (two components)	DeAngelis and Uka (2003)
Ocular dominance	t location-scale	DeAngelis and Uka (2003)
CRF size	t location-scale	Maunsell and van Essen (1987)

rather than the sinusoidal components (Movshon J, 1985; Tsui et al., 2010). Pattern selectivity in MT evolves over time (Pack and Born, 2001; Smith et al., 2005). A rather complete study of MT neural response dynamics has been conducted by Smith et al. (2005). They examined the responses of 143 MT neurons over cumulative time windows, and reported the Z-transformed pattern and component-response correlations (Z-scores). They classified each of the cells, based on their Z-scores in the last time window, as pattern direction selective, component direction selective, or "unclassified".

Our model approximates the distributions of pattern and component selectivity in each time window, and also realistic trajectories of the mean selectivities of each category of cells. To reproduce this behavior, we first fit 2D Gaussian distributions to scatterplots of pattern and component selectivity (Figures 3 and 5 of Smith et al. (2005)). To create a model of an n-neuron population, we drew n samples from the distribution for each time window. Then, to model each cell, we grouped together one pattern/component selectivity sample from each time window, as follows. Starting from the final time window, we classified the pairs to one of the three classes (pattern, component, or unclassified, as in (Smith et al., 2005)). Then we used the Hungarian algorithm (Kuhn, 1955) to match each sample in the second-last time window with a sample in the last time window. The match minimized the total of Euclidean distances between matched pairs of samples, except that we perturbed these distances with Gaussian noise, 0 +/- 2.5SD, to reproduce overlap between groups in the second-last time window. We continued this assignment process backwards in time until the pairs of the first time window were assigned to those of the second.

We produced responses with specified pattern and component correlations by combining pure pattern and component responses. To do this, we began by drawing a direction-tuning width sample. We then calculated the correlation between the pattern and component responses, r_{pc} (which depends on the direction-tuning width), and we calculated the partial pattern and component correlations R_p and R_c from the corresponding Z-scores. We then constructed a new signal $S_t = F(S_c, S_p, \boldsymbol{p})$ where F is a function of the component-direction-selective response (S_c), pattern-direction-selective response (S_p), and a vector of parameters $\boldsymbol{p}$. We found the parameters $\boldsymbol{p}$ in an optimization process whose objective was to fit the partial pattern and component correlations (R_p and R_c).

We tried the simple additive form for F:

$$S_t = F(S_c, S_p, \boldsymbol{p}) = p_1 S_c + p_2 S_p, \tag{16}$$

but this gave poor results. We therefore considered three other forms,

1. Multiplicative, $S_t = F(S_c, S_p, \boldsymbol{p}) = p_1 S_c + p_2 S_p + p_3 S_c S_p$,
2. Expansive $S_t = F(S_c, S_p, \boldsymbol{p}) = p_1 S_c + p_2 S_p + p_3 (S_c + S_p)^2$,
3. Compressive $S_t = F(S_c, S_p, \boldsymbol{p}) = p_1 S_c + p_2 S_p + p_3 (S_c + S_p)^{.5}$.

(see Results for comparison).

2.6. Comparison With Previous Models

We compared tuning curves of our model to the models of Nishimoto and Gallant (2011) and Baker and Bair (2016), with some modifications. We chose these models because they are recent and video-driven. Both build on a previous influential MT model (Rust et al., 2006). Below we describe our adaptations of these models. Note that we only use these models to provide points of comparison with our empirical model, which is otherwise unrelated.

In the model of Nishimoto and Gallant (2011), a video sequence first passes through a large bank of V1-like spatiotemporal filters with rectifying nonlinearities. The filter outputs are combined over local neighborhoods through divisive normalization. Finally, the normalized outputs are weighted optimally to approximate neural data.

As in Nishimoto and Gallant (2011), we used a bank of $N = 1296$ filters, including those with spatial frequencies up to two cycles per receptive field. In a departure from Nishimoto and Gallant (2011), we used multivariate linear regression to optimize the weights, as in Rust et al. (2006). More specifically, to optimize the weights, we generated training and testing movies for each tuning curve. Each movie was $2000 \times M$ frames in length, where M was the number of data points in the tuning curve. We used the training movie as input to the model and found the weights that minimized the error function,

$$E(\mathbf{w}) = \|\mathbf{X}_{train}\mathbf{w} - \mathbf{R}\|^2 + \lambda \|\mathbf{w}\|^2, \tag{17}$$

where $\mathbf{w} \in \mathbb{R}^{10\mathrm{N}}$ is the weight vector, $\mathbf{X}_{train} \in \mathbb{R}^{2000\mathrm{M} \times 10\mathrm{N}}$ is a matrix containing normalized V1 responses (from the spatiotemporal filters) when the training movie was used as input, $\mathbf{R} \in \mathbb{R}^{2000\mathrm{M}}$ is a vector containing the

MT responses, and λ is a regularization constant. The optimal weights that minimize this error function can be computed from,

$$\mathbf{w} = \left(\mathbf{X}_{train}^{T}\mathbf{X}_{train} + \lambda I\right)^{-1}\mathbf{X}_{train}^{T}\mathbf{R}, \tag{18}$$

where T denotes matrix transpose, -1 denotes matrix inverse, and I denotes the identity matrix.

The model of Baker and Bair (2016) is composed of two cascaded circuits. The first circuit calculates the motion response while the second calculates disparity. However, they used only the first circuit to approximate the motion tuning of MT neurons. We implemented their motion circuity, which is similar to that of Nishimoto & Gallant, but includes an additional V1 opponency stage.

The motion circuity described by Baker and Bair (2016) included a population of units tuned to different motion directions. However, their population did not span multiple motion speeds or texture frequencies. To make the model respond realistically to a wider range of stimuli, we replaced their groups of twelve direction-selective units with the same filter bank that we used for the Nishimoto and Gallant (2011) model (1296 filters). A separate filter bank was used for each eye (2592 filters in total). We used the same procedure to find the optimal weights as we did for Nishimoto and Gallant (2011) model. More specifically, given the normalized responses of spatiotemporal filters corresponding to the left and right eye $\mathbf{X}_{train}^{l}$ and $\mathbf{X}_{train}^{r}$ shown the same training movie (zero disparity), we first calculated the motion-opponent suppressed responses in each eye $\mathbf{O}_{train}^{l}$ and $\mathbf{O}_{train}^{r}$. For example for the left eye,

$$\mathbf{O}_{train}^{l} = \left[\mathbf{X}_{train}^{l} - c_{opp}\mathbf{Y}_{train}^{r}\right]_{+}, \tag{19}$$

where $\mathbf{Y}_{train}^{r} \in \mathbb{R}^{2000\mathrm{M}\times\mathrm{N}}$ is the result of reordering $\mathbf{X}_{train}^{r}$ such that each column corresponding to a filter's response with direction θ was replaced by the column corresponding to the opponent filter (i.e., a filter with $\theta - 180^{\circ}$ direction) and c_{opp} is the motion-opponency parameter (e.g., $c_{opp} = 0.5$ means the normalized V1 responses from the opponent motion filters are scaled by 0.5 before being subtracted). Finally, $[]_{+}$ denotes half-wave rectification.

We then calculated the binocular-integrated response in the left and right eye, $\mathbf{M}_{train}^{l}$ and $\mathbf{M}_{train}^{r}$. For example for the left eye,

$$\mathbf{M}_{train}^{l} = b\,\mathbf{O}_{train}^{l} + (1-b)\,\mathbf{O}_{train}^{r}, \tag{20}$$

where b is the binocular-integration parameter. We set $b = 0.5$.

We defined the error function,

$$E(\mathbf{w}) = \|\mathbf{P}_{train}\mathbf{w} - \mathbf{R}\|^2 + \lambda\|\mathbf{w}\|^2, \tag{21}$$

where $\mathbf{w} \in \mathbb{R}^{2\mathrm{N}}$ is the weight vector, $\mathbf{P}_{train} \in \mathbb{R}^{2000\mathrm{M}\times 2\mathrm{N}}$ is a matrix containing the concatenated binocular-integrated responses $\mathbf{O}^l_{train}$ and $\mathbf{O}^r_{train}$ when the training movie was used as input, $\mathbf{R} \in \mathbb{R}^{2000\mathrm{M}}$ is a vector containing the target MT responses after transforming by the inverse of nonlinearity $a\exp(bx)$, and λ is a regularization constant. We finally found the weights using regularized linear regression, as

$$\mathbf{w} = \left(\mathbf{P}^T_{train}\mathbf{P}_{train} + \lambda I\right)^{-1}\mathbf{P}^T_{train}\mathbf{R}, \tag{22}$$

where T denotes matrix transpose, -1 denotes matrix inverse, and I denotes the identity matrix.

After finding the weights, the predicted MT responses to the test movie was calculated as,

$$\mathbf{mt} = a\exp(b\,\mathbf{P}_{test}\mathbf{w}), \tag{23}$$

where exp() denotes the exponential function, and a and b are the parameters of this nonlinear function.

2.7. Prediction of Unseen MT Data

We validated the empirical model by predicting a neural dataset that had not been used to develop or parameterize the model. Specifically, we used 73 speed-tuning curves from a previous study where MT cells were shown patches of random-dot stimuli moving in eight different motion speeds (Boyraz and Treue, 2011). We created model neural populations of different sizes, and found how well the single most-similar model neuron accounted for the response of each MT cell. The inputs to the model were random-dot stimuli that were based on the description in (Boyraz and Treue, 2011).

We also used this dataset to validate and test sensitivity to a related response distribution parameter (see 3.3), specifically the scale parameter of the gamma distribution from which we drew the speed tuning widths (see Table 1). We compared how well our model predicted the speed-tuning dataset with our original scale parameter versus a range of alternative scales.

2.8. A Deep CNN for Visual Odometry

A potential application of the empirical model is to shape intermediate representations of deep networks so that they more closely resemble those of the primate dorsal stream. To experiment with this approach, we developed a deep CNN to solve a visual odometry task, and used our empirical model to train one of the middle layers of the network. Visual odometry is the process of using visual information to estimate self-motion, a function that probably involves the dorsal stream. Neurons in the middle superior temporal area (MST), which receives strong input from MT, respond to large optic flow patterns such expansion and spiral motion, which are highly relevant to self motion.

2.8.1. Architecture

We created a deep convolutional network that was based loosely on the the macaque dorsal visual stream. The network architecture is shown in Figure 3, and Table 2 lists the network parameters. The left and right input layers (stereo frames) were each followed by a convolutional layer. We then merged these two layers together and connected the result to the third convolutional layer. This convolutional layer was followed by a max-pooling layer. These layers correspond roughly to the primary visual cortex (V1), which includes binocular neurons and complex cells. We used two convolutional layers to model each of V2, V3/V3a, MT, and MST, to model a separation between input and output cortical layers in each area. We added skip-connections consistent with (Markov et al., 2014). Specifically, the first convolutional layer corresponding to V3/V3a received input from the last layers of both areas V1 and V2, and the first convolutional layer of area MT received input from the last layer of all earlier areas. After the MST layers, we added a dense layer with 1024 hidden units and an output layer with three units to estimate medio-lateral, antero-posterior and angular velocities from input frames.

2.8.2. Dataset

We needed a stereo odometry dataset, with high frame rate, but did not find suitable existing datasets. We therefore created a new synthetic dataset in Unreal Engine 4. We used the Modular Neighborhood Pack, which contains a residential neighborhood that looks fairly realistic.

We used UnrealCV plugin to move a stereo camera system along curvilinear paths inside this virtual world. The baseline of the camera system

Figure 2: Two example stereo frames from our odometery dataset. Both left and right frames were 76x76 pixels.

was 60mm, which is within the range of human interpupillary-distance. The dataset consisted of "moves" of six frames each, starting at different locations and moving along different trajectories. For each move, we drew random numbers for medio-lateral, antero-posterior and angular velocities. For each move, we collected six grayscale 76x76 stereo frames (at 60 FPS). Figure 2 depicts two example stereo frames from the dataset.

The dataset had 75000 moves for training and 9000 moves for validation and testing. For each move, the deep CNN took the stereo sequence as input and medio-lateral, antero-posterior and angular velocities as regression targets for the output layer. To train the middle layer of the deep CNN, corresponding to area MT, we calculated dense direction, speed, and disparity fields (pyramidal Lucas-Kanade method) as well as contrast fields (Peli method) for every frame of the sequence. For each of these four fields, we fed the sequence-average field to our empirical model, to produce regression targets for the MT layer. The target therefore reflects average stimulus features over several frames, roughly consistent with the low-pass properties of MT neurons Bair and Koch (1996).

Layer	# Kernels	Kernel Size	Shape	Pool	Nonlinearity
V1-1	256	7 x 7	70 x 70	None	ReLU
V1-2	256	7 x 7	70 x 70	None	ReLU
V1-binocular	256	7 x 7	64 x 64	3 x 3	ReLU
V2-1	256	7 x 7	21 x 21	None	ReLU
V2-2	256	7 x 7	21 x 21	None	ReLU
V3-1	128	7 x 7	21 x 21	None	ReLU
V3-2	128	7 x 7	21 x 21	None	ReLU
MT-1	64	5 x 5	17 x 17	None	ReLU
MT-2	64	5 x 5	13 x 13	2 x 2	ReLU
MST-1	128	9 x 9	6 x 6	None	ReLU
MST-2	128	9 x 9	6 x 6	2 x 2	ReLU
Dense			1024		ReLU
Output			3		None

Table 2: Structure of the example CNN that we used in the visual odometry task.

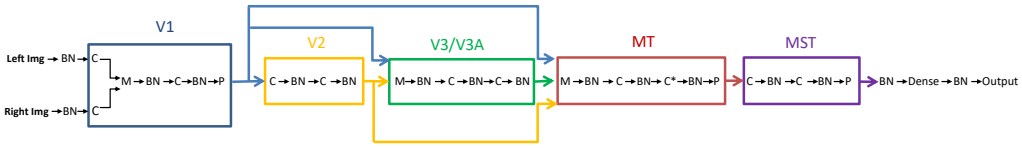


Figure 3: Structure of the CNN. BN: batch normalization, M:merge layer, C: Convolutional layer, P: pool layer. *The MT cost is applied at the output of the second convolutional layer in MT.

2.8.3. Training

To train the deep network to both approximate odometery and emulate the empirical model we pursued two approaches. In the first approach, we first trained the network up to MT-2 layer (see Table 2) to only approximate MT activity, for forty epochs. We used the root-mean-square error of MT-2 layer outputs and the MT activity targets as the training loss. These target activities were calculated with a simplified version of our empirical model where the dynamics of pattern and component selectivity and motion-in-depth tuning were omitted, and we chose difference of Gaussian kernels as receptive fields. Each of these kernels was elongated orthogonal to the preferred direction of its respective unit. After training for MT activity, we "froze" these layers and trained the rest of the network (i.e. from MST-1 layer to the end) for odometery task, for forty epochs. Here we used the root-mean-square error of the network outputs and velocity labels in our odometery dataset as the training loss. Finally to achieve a better performance on odometery task, we unfroze all network layers and trained it only on odometery for another five epochs.

In the second approach, we trained the network on both odometery and MT activity simultaneously. In this case, our training loss was a linear combination of both MT activity and odometery losses. This combined loss function can be written as,

$$E = A_1 \sum_v (y_v - t_v)^2 + A_2 \sum_i (y_i - r_i)^2, \tag{24}$$

where t_vs are target velocities, y_vs are network outputs, r_is are normalized neural responses, computed using our empirical model on input frames, and y_is are unit activities of MT-2 feature maps. Finally, A_1 and A_2 are linear weights.

We implemented the convolutional network in Keras (Chollet et al., 2015) using TensorFlow (Abadi et al., 2016) as a backend, and trained it on an NVIDIA GeForce GTX Titan Xp GPU.

We used the Adam algorithm (Kingma and Ba, 2014) as the optimizer with the default parameters. We also used batch normalization (Ioffe and Szegedy, 2015) in some layers (see Figure 3). Like dropout (Srivastava et al., 2014), batch normalization has regularization benefits, which reduces overfitting, while it also speeds up training.

2.9. Sensitivity Analysis of Response Features on Odometery Performance

Our model provides the possibility of investigating the influence of individual MT response features on task performance. To illustrate this, we chose two response features: (1) direction tuning bandwidth and (2) speed tuning width. We only used the part of the deep CNN after the layers corresponding to MT (i.e. from MST-1 layer to output layer). Hence, the network received empirical model responses as input. This simplification allowed us not only to train faster but to make sure that the change in performance was directly due to the modification of the response feature under study not the failure of the deep CNN in emulating the modified response because of that feature.

3. Results

3.1. Tuning Curve Approximation Examples

We tested how accurately our model could reproduce tuning curves of real MT neurons from the electrophysiology literature. For each tuning curve, we generated the same kinds of visual stimuli (e.g. drifting gratings, plaids, and fields of moving random dots) that were shown to the monkeys. We used these stimuli as input to the model, and optimized the model parameters to best fit the neural data.

Table 3 summarizes the results of the tuning curve fits for our model, which we call Lucas-Kanade Nonlinear-Linear-Nonlinear (LKNLN), and our adaptations of the previous models by Nishimoto and Gallant (2011) (NG) and (Baker and Bair, 2016) (BB). Note that Baker and Bair (2016) provide a software implementation of their model, but it has a small filter bank (see Methods) that is inadequate for processing many stimuli. We optimized relevant model parameters individually for each tuning curve. In our LKNLN model, there are relatively few such parameters, because the tuning curves are independent, and we did not change the calculation of the input fields. So only the parameters of the relevant tuning function and final nonlinearity were optimized. For the NG and BB models we optimized all the models' variable parameters, including weights of the spatiotemporal filters, for each tuning curve. Examples of tuning curve fits are shown in the following figures. Sources of error in our empirical model include non-ideal behavior of the computer-vision methods operating on input images, and the data falling outside the tuning curve function family.

	#Tuning Curves	LKNLN	NG	BB
Speed	11 (8)	0.0531	0.1075	0.1654
Speed/Contrast	2 (8)	0.0543	0.2959	0.3650
Attention/Direction	2 (12)	0.0384	0.0848	0.1100
3D Motion	8 (12)	0.2144	NA	0.2450
Stimulus size	2 (7)	0.0667	0.0841	0.0599

Table 3: Summary of RMSE comparison between our model (LKNLN), Nishimoto and Gallant (2011) (NG), and Baker and Bair (2016) (BB) to the neural data for different tuning parameters. The second column provides the number of tuning curves (along with the number of points in each tuning curve). Note that the NG model is monocular, so it does not reproduce binocular phenomena.

Figure 4 shows the speed tuning curves of four neurons (with different preferred speeds) where the monkeys were shown fields of random dots moving with different speeds. Our model approximates the neural data more closely than our adaptations of the models of Nishimoto and Gallant (2011) and Baker and Bair (2016).

Figure 5 illustrates the speed tuning of a neuron for moving random dots in two cases: when dot luminance was high, resulting a high contrast stimulus (Figure 5A), and when dot luminance was low, resulting a low contrast stimulus (Figure 5B). As shown in the figure, increasing the contrast not only modulated the response gain (peak spike rate) but it also shifted the preferred speed (position of the peak on the speed axis). Our model reproduces both these phenomena, whereas the previous models reproduce only the first. Note however that our empirical model does not provide a mechanistic explanation of the MT data, but only a fit.

Figure 6 shows models' fits to data on the effect of attending to stimuli in a neuron's receptive field. Attending to stimuli modulates responses of different MT neurons to varying degrees. While our model received attention masks, there was no mechanism for attention modulation in the other models. In our adaptations of these models, we modulated their responses with a scalar gain for attended stimuli. This gain was found such that the mean-squared error of data and model responses were minimized.

Majaj et al. (2007) showed that motion integration by MT neurons oc-

curs locally within small sub-regions of their receptive fields, rather than globally across the full receptive fields. They identified two regions within the receptive fields of a neuron where presenting the stimulus evoked similar neural responses. Then, they studied motion integration by comparing the direction selectivity of MT neurons to overlapping and non-overlapping gratings presented within the receptive field. Since motion integration was local, the ability of the neurons to integrate the motions of the two gratings was compromised when gratings were separated. Our model approximates this neural behavior well (see Figure 7). According to Nishimoto and Gallant (2011), their model does not account for this phenomenon, and extending it to do so would require including nonlinear interactions between the V1 filters of the model, which would drastically increase the number of parameters, making estimation more difficult. Other previous models that treat overlapping and non-overlapping features identically (Simoncelli and Heeger, 1998; Rust et al., 2006; Baker and Bair, 2016) would also not reproduce this phenomenon.

MT neurons also encode binocular disparity, with a variety of responses across the MT population, including preferences for near and far disparities, and various selectivities and depths of modulation. Our model closely approximates a wide variety of MT neuron disparity-tuning curves (Figure 8).

Recent studies (Czuba et al., 2014) have revealed that some MT neurons respond to 3D motion, confirming area MT's role in encoding information about motion in depth. Figure 9 shows the neural responses of two different neurons to monocular and binocular stimuli. One neuron (Figure 9A-D) is tuned for fronto-parallel motion while the other neuron is tuned for motion toward the observer (Figure 9E-H). Our model approximates both types of neuron.

Size tuning is a result of antagonistic surrounds. Increasing the size of the stimulus to a certain point (optimal size) will increase an MT neuron's response, while larger-than-optimal stimuli evoke smaller responses. Figure 10 shows an approximation of two size-tuning curves using a symmetric difference-of-Gaussians kernel, one of three types that we adapt from (Xiao et al., 1997).

3.2. Dynamics of Pattern and Component Selectivity

Figure 11 shows the distribution and dynamics of pattern and motion selectivity in the empirical model. The model closely approximates the data

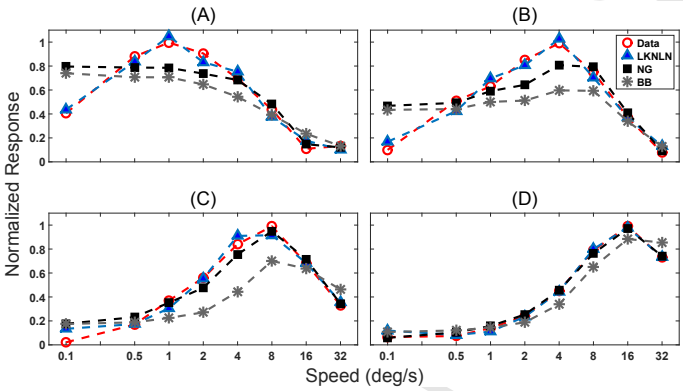


Figure 4: Speed tuning curves of four MT neurons, plotted on a logarithmic speed axis. Responses have been normalized so that the peak response of each neuron is equal to 1 . Mean ± SD error for (A): 0.00 ± 0.06 spike/s (LKNLN), 0.00 ± 0.18 spike/s (NG), and 0.06 ± 0.21 spike/s (BB); for (B): -0.01 ± 0.06 spike/s (LKNLN), -0.00 ± 0.18 spike/s (NG), and 0.09 ± 0.23 spike/s (BB); for (C): -0.01 ± 0.06 spike/s (LKNLN), -0.00 ± 0.08 spike/s (NG), and 0.11 ± 0.21 spike/s (BB); for (D): -0.00 ± 0.02 spike/s (LKNLN), -0.00 ± 0.02 spike/s (NG), and 0.02 ± 0.09 spike/s (BB). Data replotted from Nover et al. (2005).

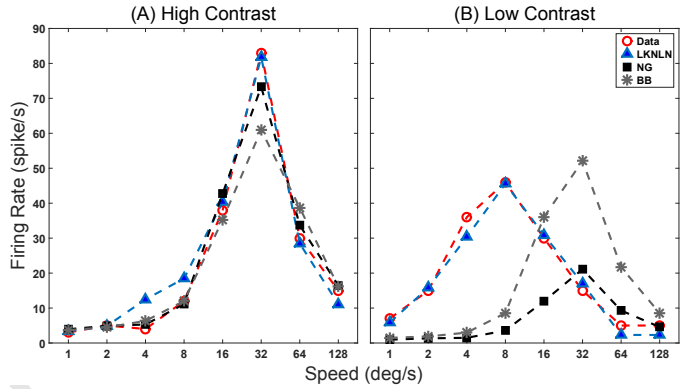


Figure 5: Effect of contrast on speed tuning curves. A, Speed tuning in high contrast. Mean ± SD error: -1.19 ± 3.35 spike/s (LKNLN), -0.18 ± 4.40 spike/s (NG), and 1.55 ± 8.90 spike/s (BB). B, Speed tuning in low contrast. Mean ± SD error: 1.19 ± 2.97 spike/s (LKNLN), 13.09 ± 17.83 spike/s (NG), and 3.22 ± 24.84 spike/s (BB). Contrast modulates the response and also shifts the peak (i.e., the preferred speed). While contrast modulates the response amplitude in all three models, only our model (LKNLN) accurately shifts the peak. Data replotted from Pack et al. (2005).

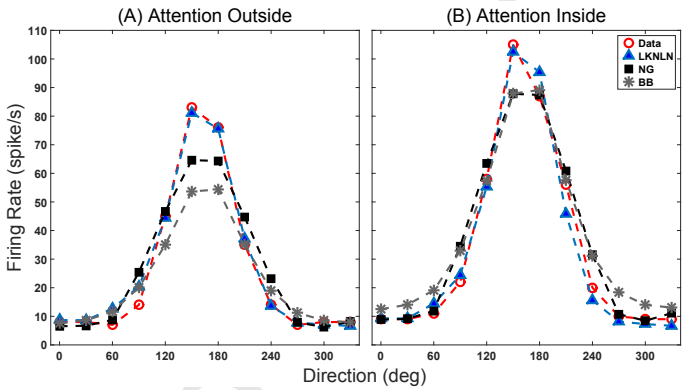


Figure 6: Attentional modulation of direction tuning. A, When the stimulus inside the RF was not attended. Mean ± SD error: -0.90 ± 2.73 spike/s (LKNLN), -0.02 ± 8.51 spike/s (NG), and 3.34 ± 11.26 spike/s (BB). B, When the stimulus inside the RF was attended. Mean ± SD error: 0.90 ± 4.63 spike/s (LKNLN), -1.73 ± 7.44 spike/s (NG), and -3.52 ± 7.43 spike/s (BB). Neural data for both cases replotted from Treue and Martínez Trujillo (1999). Our model (i.e., LKNLN) receives attention masks as input, so we defined the masks so that they did not cover the stimulus for the unattended case and covered for the attended case. For the other two models, we first found the best fit for the unattended case by multivariate regression. Given the unattended solution we then found the gain that minimized the error difference between the attended tuning curve and the modulated unattended solution.

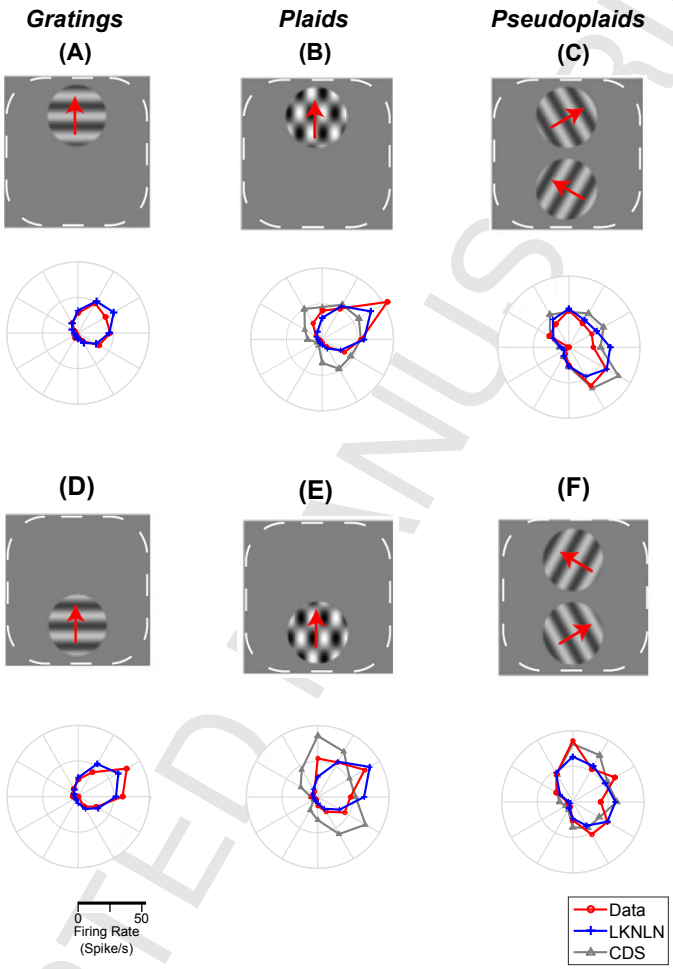


Figure 7: Response of an MT cell to gratings and plaids placed within different regions of the cell's receptive field (RF). The response magnitude is plotted on the radial axis, and the angular axis is the direction of motion. A,D, The neuron's response to grating stimuli at two different patches within RF. B, E, The neuron's response to plaids placed at two different regions over RF. The plaid stimuli are made by overlapping two gratings oriented 120° apart. Since this cell is selective for the motion of plaids independent of the orientation of their components (gratings), it is classified as a pattern direction selective (PDS) neuron. D, F, The two grating components of the plaids in (B,E) separated to different parts of the receptive field. If motion integration in MT cells were global (i.e., if these cells simply pooled all of their inputs from V1 cells), these plots would be similar plots as (B,E). Instead, the response in this case is close to the component direction selective (CDS) prediction, indicating that motion integration in MT cells are local rather than global. Our model produces realistic responses. Neural data (red) and CDS prediction (gray) replotted from Majaj et al. (2007); blue is our model.

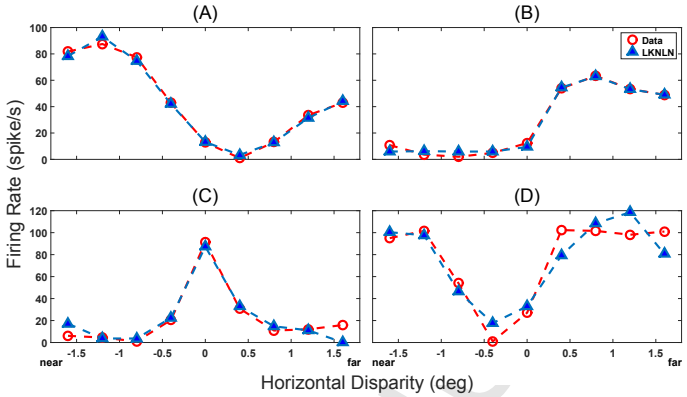


Figure 8: Disparity tuning curves of four neurons. Data replotted from (DeAngelis and Uka, 2003). A, Near (0.00 ± 1.96 spikes/s; mean error $\pm$ SD). B, Far (0.50 ± 1.58 spikes/s; mean error $\pm$ SD). C, Tuned-zero (-0.43 ± 2.93 spikes/s; mean error $\pm$ SD). D, Tuned inhibitory (1.38 ± 3.77 spikes/s; mean error $\pm$ SD).

from Smith et al. (2005).

As described in the Methods, we experimented with four different ways of combining pattern and component responses. To compare performance between these different forms, we used the population Pearson correlation coefficient between Z-scores that we randomly drew from the distributions, which were approximated for each time window, and the Z-scores that we calculated after building the response S_t based on S_c, S_p, and $\boldsymbol{p}$, which we found in the optimization process. Table 4 summarizes the results for a population of 500 neurons. The best results were obtained by the compressive form where we linearly combined pattern response, component response, and a third term, which was constructed by passing the sum of these two responses through a compressive nonlinearity.

3.3. Parameter Distributions

The empirical model is meant to closely approximate population activity in MT, so statistical distributions of parameters are also an important part of the model. Such distributions have frequently been estimated in the literature. However, past computational models of MT have typically not attempted to produce realistic population responses, except along a small number of tuning dimensions (e.g. Nover et al., 2005).

Figure 12 shows nine examples of fits of parameter distributions. In each case we chose the best of seven different distributions according to the Akaike

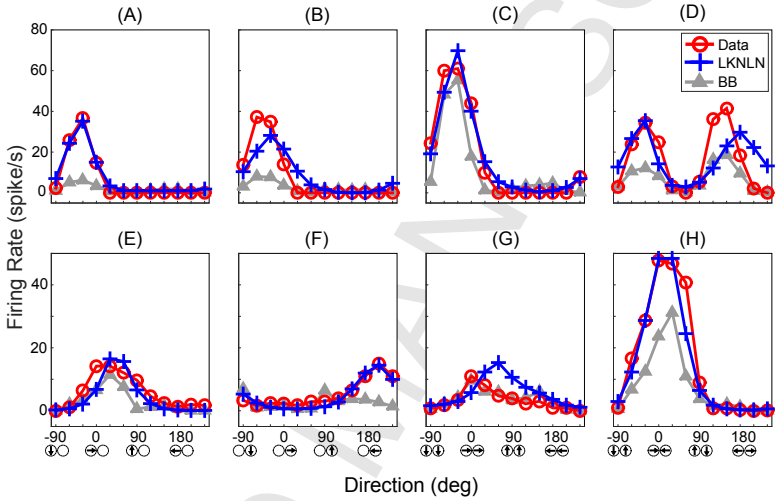


Figure 9: Examples of direction tuning of two MT neurons to monocular and binocular stimuli. A–D, An MT neuron tuned for frontoparallel motion. A-B, Direction tuning for gratings presented monocularly to the left (A) and right eye (B). C, Direction tuning for binocular presentation of identical gratings. D, Direction tuning for gratings drifting in opposite directions in the two eyes. E–H, Responses of an MT neuron tuned for motion toward the observer. Direction tuning curves for monocular gratings (E, F), binocularly matched (G), and binocularly opposite motion (H). Neural data replotted from Czuba et al. (2014) in red, prediction of our model (LKNLN) in blue, and prediction of Baker and Bair (2016) model (BB) in gray. Mean ± SD error, A: -1.11 ± 1.77 spikes/s (LKNLN) and 4.62 ± 10.63 spikes/s (BB), B: -0.25 ± 7.04 spikes/s (LKNLN) and 5.70 ± 11.39 spikes/s (BB), C: -0.61 ± 5.25 spikes/s (LKNLN) and 5.27 ± 9.85 spikes/s (BB), D: -0.71 ± 12.91 spikes/s (LKNLN) and 8.95 ± 9.40 spikes/s (BB), E: 1.48 ± 2.88 spikes/s (LKNLN) and 2.84 ± 3.02 spikes/s (BB), F: 0.52 ± 1.30 spikes/s (LKNLN) and 2.58 ± 4.84 spikes/s (BB), G: -2.47± 3.98 spikes/s (LKNLN) and -0.45 ± 1.42 spikes/s (BB), H: 1.42 ± 4.96 spikes/s, (LKNLN) and 8.15 ± 10.85 spikes/s (BB).

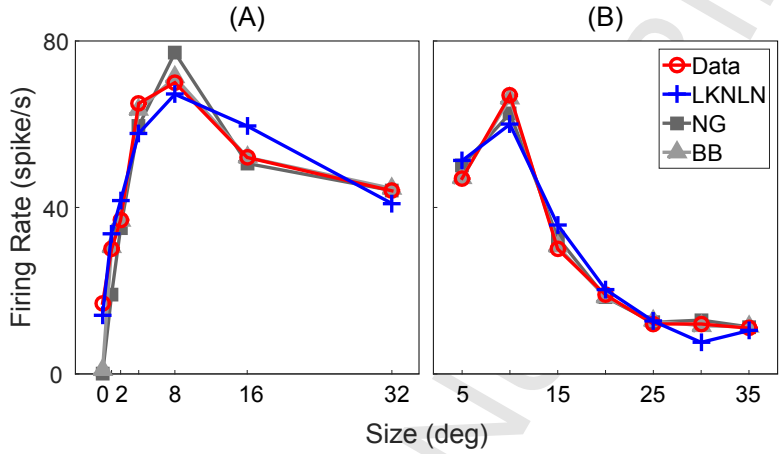


Figure 10: Two examples of size tuning curves. The kernels, which gave rise to the size tuning in our model (LKNLN), were radially symmetric difference of Gaussians centered at the centre of video frames (the same as neuron's receptive field centre). A, Neural data replotted from DeAngelis and Uka (2003). Mean ± SD error: -0.00 ± 5.32 spikes/s (LKNLN), 4.21 ± 7.86 spikes/s (NG), and 2.18 ± 6.16 spikes/s (BB). B, Neural data replotted from Pack et al. (2005). Mean ± SD error: 0.00 ± 4.54 spikes/s (LKNLN), -0.35 ± 2.50 spikes/s (NG), and -0.01 ± 0.59 spikes/s (BB).

Form	30-50ms	30-70ms	30-90ms	30-110ms	30-320ms
Additive	0.31	0.52	0.06	0.57	0.53
Multiplicative	0.65	0.98	1.00	0.993	0.80
Expansive	0.99	0.99	0.99	1.00	0.91
Compressive	1.00	1.00	1.00	1.00	0.95

Table 4: Summary of comparison between four different forms of combining component and pattern direction selective responses. A population of 500 neurons was modelled. Numbers indicate the population Pearson correlation coefficients between the sampled and calculated Z-scores based on a specific form for the corresponding time window. For example, 0.53 in the last column of the second row indicates that $\rho_{sampled,calculated} = 0.53$ where *sampled* refers to the population of 1000 sampled Z-scores (500 Z_cs and 500 Z_ps) drawn from the modelled distribution of 30-320ms time window, and *calculated* means the Z-scores calculated for S_c, S_p, and S_t where S_t was calculated by combining S_c and S_p signals in the additive form (see Equation 16). The compressive form had the best performance (highest Pearson correlation) in all time windows.

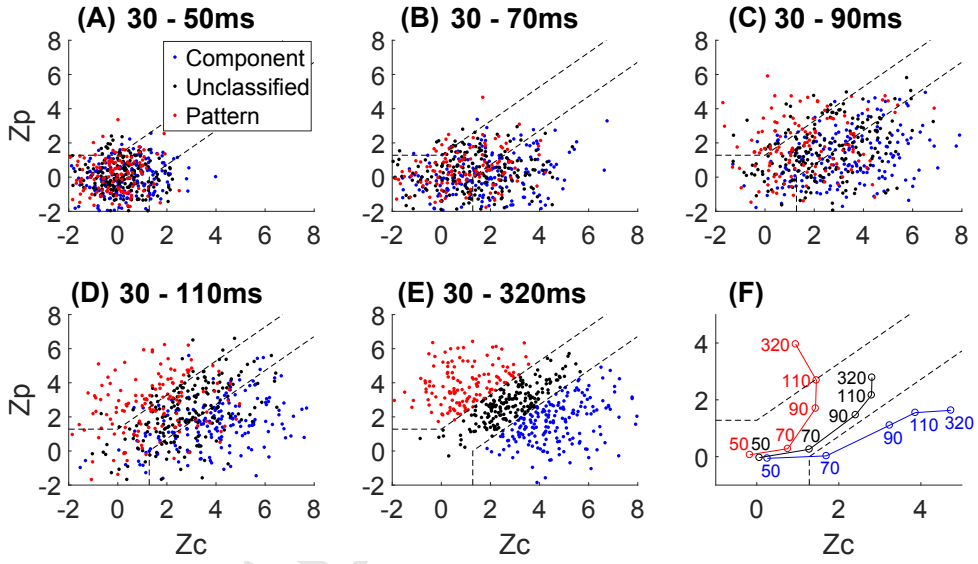


Figure 11: Pattern selectivity of empirical model. A-E, Scatterplots of Z-transformed pattern and component correlations (Z_p and Z_c) for 500 modelled neurons over time. The red and blue dots represent the pattern and component neurons, respectively. The black dots represent neurons which are not classified. For the final time window (E), we classified each neuron based on its location on the Z-transformed-correlations plane as in Smith et al. (2005). For other time windows (A-D), we used Hungarian algorithm to match each sample in a time window (e.g. D) to its latter time window (e.g. E) so that the total Euclidean distance between matched samples, perturbed with Gaussian noise (0 +/- 2.5SD), was minimized. F, the time evolution of each class. Each data point represents the average Z_p and Z_c values, of a particular class, in a time window whose ending time has been written next to it (see Figures 5-6 of Smith et al. (2005) for comparison with actual neural data; we do not replot the data here because some of the dots are too dense to be extracted accurately).

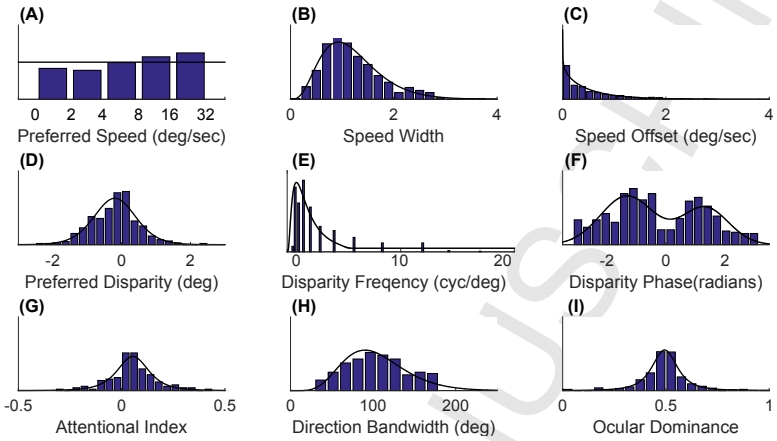


Figure 12: Examples of parameter distributions. In each case we replot the data (histograms) along with the selected distribution. A-C, speed parameters including preferred speed (log uniform) in logarithmic space, speed width (gamma), and speed offset (gamma) (Nover et al., 2005). D-F, disparity parameters including preferred disparity (t location-scale), disparity frequency (log normal), and disparity phase (Gaussian mixture) (DeAngelis and Uka, 2003). G, Attentional index (t location-scale) (Treue and Martínez Trujillo, 1999). H, Direction bandwidth (gamma) (Wang and Movshon, 2016). I, Ocular dominance (t location-scale) (DeAngelis and Uka, 2003).

Information Criterion (Akaike, 1974), as described in the Methods.

3.4. Neural Response Predictions

Beyond examining fits of published tuning curves and distributions of response properties, we further validated the model using a more detailed dataset of speed tuning in 73 MT cells, from a previous study (Boyraz and Treue, 2011). This experiment involved random dot stimuli moving coherently at one of eight different speeds (0.5, 1, 2, 4, 8, 16, 32, 64deg/sec).

We approximated the responses of these MT cells by creating a population of N synthetic neurons of our empirical model. We chose N to be 8, 16, 32, 64, 128, 265, or 1048. At each speed of motion, we recreated ten sequences of moving random dot stimuli (with the same dot size, density, contrast, and replotting scheme as the original study) and fed them to the synthetic neural population. The final response of each synthetic neuron in the population at each speed was calculated as the average of the ten sequences at that speed. Next, for each MT cell, we selected the synthetic neuron from the population that had the highest correlation with that MT cell. We calculated the coefficient of determination (r^2) as the proportion of the variance in the

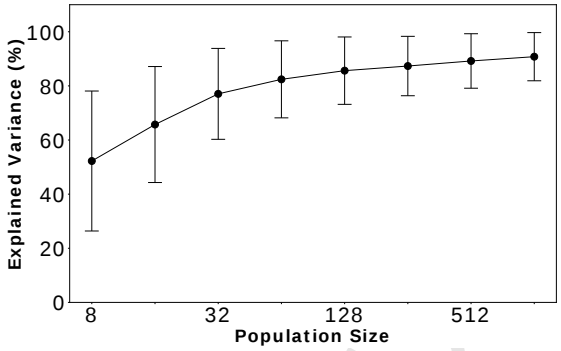


Figure 13: Explained variance vs. population size of empirical model. As the population of empirical model grows, the probability of having a synthetic neuron with more similar response increases. Each point and error bar, respectively, represents the average and standard deviation of 365 (73x5) r^2 values (73 MT cells times five different model populations for any given population size).

MT cell, which was predictable from that synthetic neuron. In summary, we used a nearest-neighbor approximation of each cell rather than linear-regression from the full model population.

Because of the stochastic population parameters of the empirical model, two N-neuron populations sampled from these distributions will not have identical responses. Therefore, instead of a single N-neuron population, we created five populations, repeating the above process for each population.

Figure 13 illustrates how the average explained variance for 73 MT cells increases as the empirical model grows in size. Each point of the curve is the average of 365 (73x5) r^2 values, because there were 73 MT cells and five different populations for any given population size.

We repeated this process with various scale parameters of the gamma distribution (see Table 1) from which the speed tuning widths (see Equation 8) were drawn, to validate this parameter and test sensitivity to it. We chose 64 as the population size N and again created five different populations. As can be seen in Figure 14, there is a modest dependence on this parameter, and the averaged explained variance is indeed highest when the speed tuning widths were drawn from our original estimate.

3.5. CNN for Visual Odometery

We trained convolutional neural networks (CNNs) to estimate self motion from visual input, as described in the Methods. The dataset included natural-

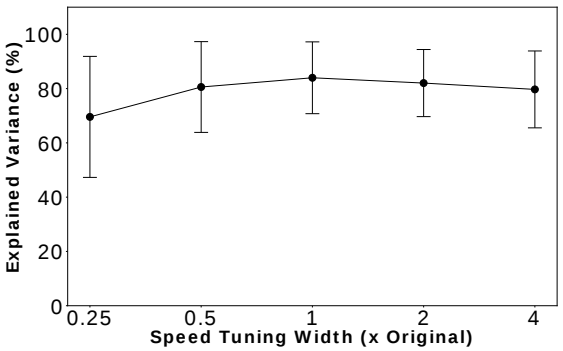


Figure 14: Explained variance vs. the scale parameter of the speed-tuning-width gamma distribution of empirical model (see text for details). We changed the scale parameter by multiplying it with one of [0.25, 0.5, 1, 2, 4] values. The model population size was 64. Each point and error bar, respectively, represents the average and standard deviation of 365 (73x5) r^2 values. Our original scale parameter produced the best predictions on average.

istic visual stimuli, but since the dataset was synthetic, we had ground-truth velocity labels. We used the empirical model for MT labels, but we omitted the dynamics of pattern and component selectivity, as emulating these dynamics might require a more complex recurrent network. Figure 15 shows the validation loss curves of two different networks. CNN-O network was trained only on odometery task (no emulation of MT responses). CNN-OMT was trained with a linear combination of both costs (Equation 24). We chose $A_1 = 1$ and tested different values for A_2. We found $A_2 = 4$ to be the best choice, as larger values prevented the combined validation loss from going down, and smaller values made the second cost negligible compared to the first. We also trained a third network, CNN-3Phases, in three phases (as described in section 2.8.3): we first trained the part of the network up to MT, with the MT cost (CNN-MT); then the rest of the network with the odometry cost (with the weights up to the MT layer frozen); and finally the full network with the odometry cost. Figure 16 shows a scatter plot of actual velocities (of the validation set) against the velocities predicted by these three networks. As the correlations between the network-output and target velocities suggest, all three networks perform quite well.

In Figure 17, we show the MT validation loss curves for CNN-OMT and CNN-MT (i.e. the first phase of CNN-3Phases). The loss is higher for CNN-OMT since the network had to learn not only to emulate MT activity targets

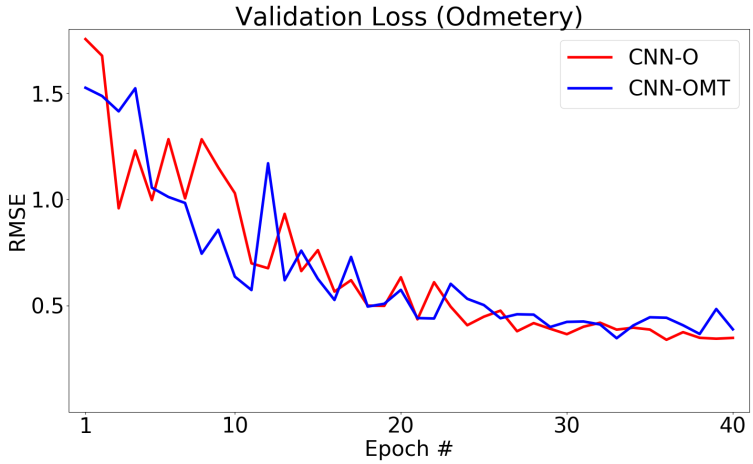


Figure 15: Validation loss curves for odometry task in two networks. CNN-O: the network was trained only with odometry cost, CNN-OMT: the network was trained simultaneously with MT and odometery costs.

but also to estimate velocity targets. Although we tried both larger and smaller values for A_2, we could not reduce MT loss any further for CNN-OMT (data not shown). To confirm that this was in fact due to the odometry cost, rather than details of the training approach, we continued training of CNN-OMT with the odometry cost removed. The MT cost then declined rapidly (dashed line).

3.6. Speed and Direction Tuning of CNN Units

In this section we examine speed and direction tuning of units in the MT layers of three CNNs: one trained for the odometry task alone (CNN-O), one trained for MT response approximation alone (CNN-MT), and one trained with both these cost terms simultaneously (CNN-OMT).

Figures 18 and 19 show tuning curves of CNN-O (trained for the odometry task alone). Previous work (e.g. Yamins et al., 2014) has shown that task-trained CNNs often have physiologically relevant representations. Indeed, our CNN-O units have tuning for both direction and speed of visual motion. This is unsurprising, because the task depends entirely on the pattern of direction and speed across the visual field. However, the tuning curves are somewhat different than physiological tuning curves. Many of the direction-tuning curves are narrow (Figure 20), and most of the speed tuning curves are monotonic and high-pass. Also, the tuning curves of many units are quite

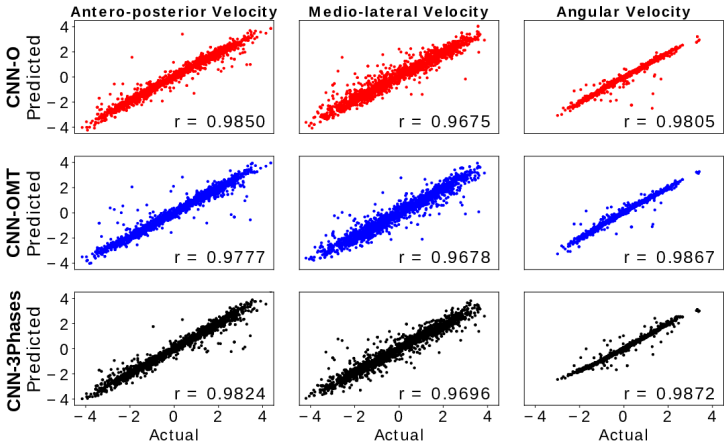


Figure 16: Scatter plots of actual vs. predicted self-motion velocities of the validation set. Top: CNN-O, the network only trained with odometry loss. Middle: CNN-OMT, the network trained simultaneously with both MT and odometery losses. Bottom: CNN-3Phase, the network trained in three phases: (1) up to the MT-2 layer with MT loss, (2) after the MT-2 layer with odometery loss, (3) the complete network with odometery loss.

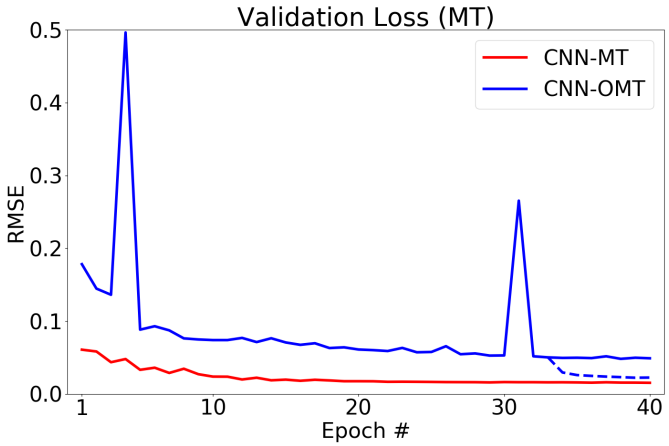


Figure 17: Validation loss curves for MT regression. CNN-MT: the network was trained only with MT cost, CNN-OMT: the network was trained with MT and odometery costs. The dashed line shows the training curve of CNN-OMT for seven epochs when we trained only with MT cost (no odometry cost), initialized with 33th-epoch weights that gave us the lowest MT loss.

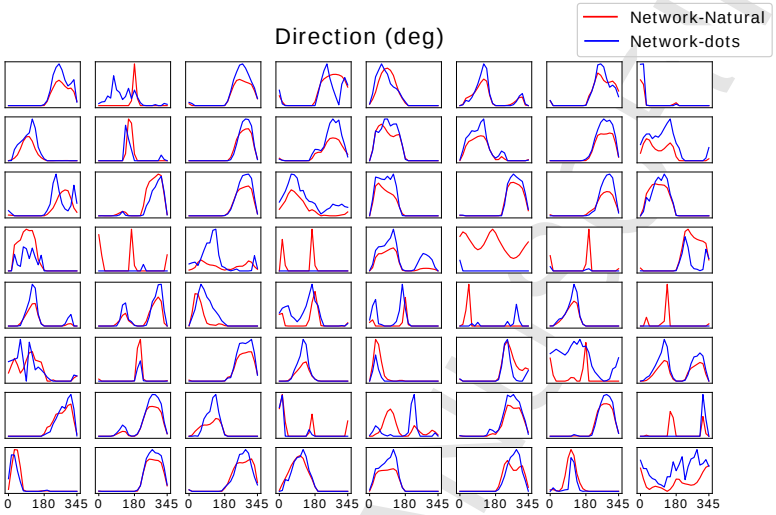


Figure 18: Direction-tuning curves of example units in CNN-O. The red and blue curves are responses to natural-scene stimuli and random-dot stimuli, respectively. The networks were trained only on natural-scene stimuli. The direction-tuning curve of each unit is measured at the maximum of 0.5°/s and speed at which the unit responded most strongly. We used minimum of 0.5°/s because the computer-vision results were less reliable at lower speeds, resulting in noisier tuning curves. The curves are normalized to their peak responses.

sensitive to the stimulus used to calculate the curves. In particular, they are quite different for dot stimuli vs. scene stimuli.

Figures 21 and 22 show example tuning curves for CNN-MT, along with the target curves for each unit, from the empirical model. Despite fairly low regression error on the validation stimuli, some substantial differences are evident in the tuning curves. This is likely because the distribution of stimuli in the odometry task is different than the distribution of stimuli used to make the tuning curves. For example, in the task stimuli, horizontal motion is represented more strongly than motion in other directions, due to horizontally curvilinear self-motion paths.

The sensitivity of the tuning curves to the stimulus (i.e. random dots vs. natural scenes) is much lower in CNN-MT than CNN-O (Figure 25, top vs. middle row), despite the fact that both networks were trained only on natural scenes, and in fact with the same set of stimuli.

Figures 23 and 24 show example tuning curves of CNN-OMT. This network's tuning was weakly related to the targets from the empirical model.

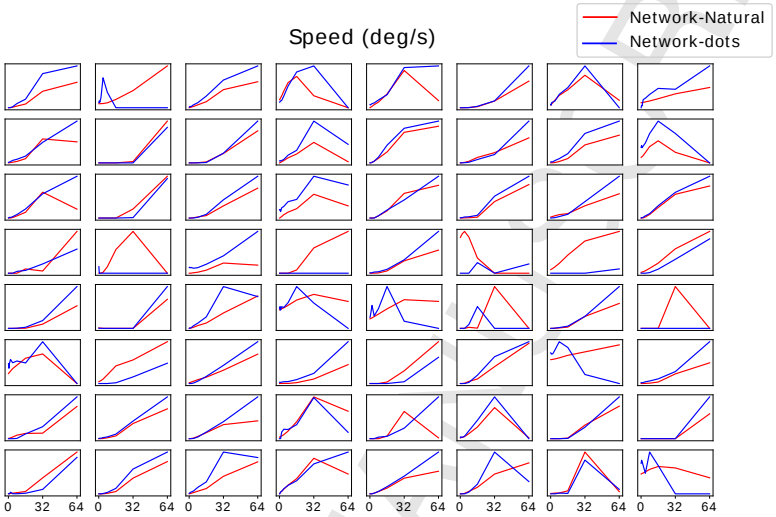


Figure 19: Speed-tuning curves of example units in CNN-O (the same units as in Figure 18). These were measured at the motion direction that evoked the strongest response. Conventions as in Figure 18.

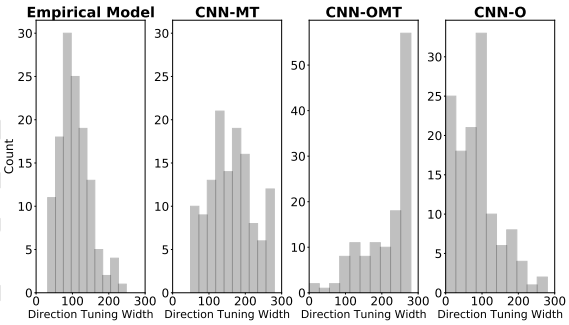


Figure 20: Left, half-height widths of direction-tuning curves in the empirical model units that make up the target population for CNN-MT and CNN-OMT. Second left, half-height widths of direction-tuning curves in the MT layer of CNN-MT. Second right, half-height widths of direction-tuning curves in the MT layer of CNN-OMT. Right, half-height widths of direction-tuning curves in the MT layer of CNN-O (trained only for the odometry task). Many of these direction-tuning curves are narrow.

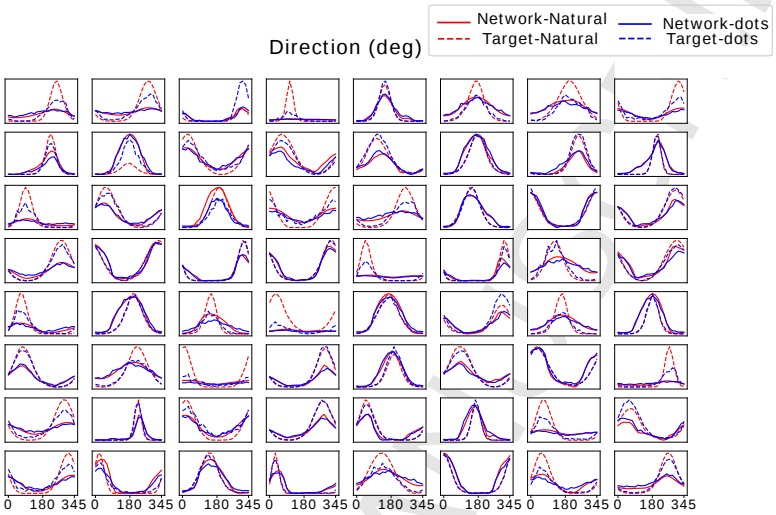


Figure 21: Direction-tuning curves of example units in CNN-MT. The red and blue traces are tuning with natural-scene and dot stimuli, respectively. The dashed lines indicate target values from the empirical model. These are slightly different for natural-scene and dot stimuli, due to differences in interpretation by the computer-vision algorithms, and differences in contrast between the stimuli. Similar to Figure 18, the direction-tuning curves were measured at the preferred speeds of the empirical model, or 0.5°/s, whichever was greater. Preferred speeds were calculated separately for dot and natural-scene stimuli, based on their mean contrasts.

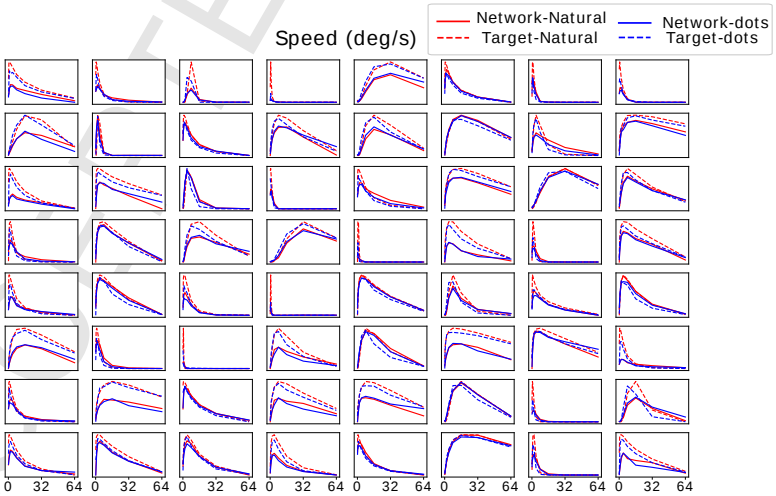


Figure 22: Speed-tuning curves of example units in CNN-MT, calculated at the preferred directions of the empirical model units. Conventions as in Figure 21.

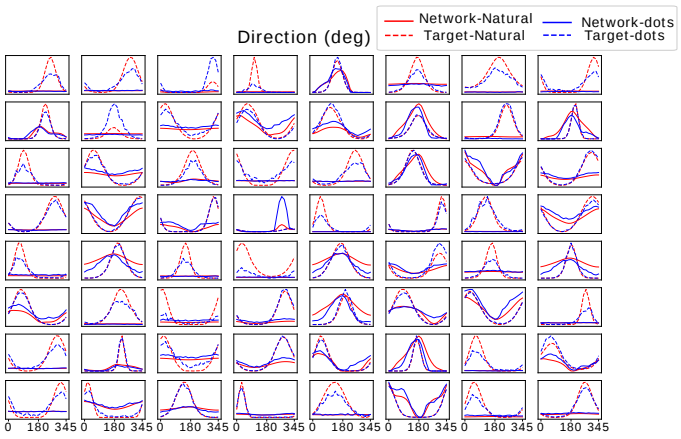


Figure 23: Direction-tuning curves of example units in CNN-OMT. Conventions as in Figure 21.

For example, the direction-tuning curves are quite broad (see also Figure 20). This is somewhat surprising, because the MT regression error was only moderately higher in this network than in CNN-MT (root mean-squared error .048 vs. 0.015). Low regression cost may be possible, despite poor tuning curves, due to good prediction of low activities, and good prediction for speeds and directions that are most commonly seen in training and testing. This outcome suggests that changes to the regression cost may be needed to produce realistic tuning in this context. Possible changes include training with a different distribution of stimuli, or weighing the cost of rare cases more heavily. This network CNN-OMT was also sensitive to the stimulus (Figure 25, bottom row).

Figure 26 compares correlations between target and actual tuning curves for CNN-MT (top row), CNN-3Phases (second and third rows), and CNN-OMT (bottom row). These plots show that many tuning curves of CNN-OMT are poorly related to those of the empirical model, particularly for dot stimuli. The same can be said for CNN-3Phases especially as the third training phase (i.e. training only on odometery task) advanced (first epoch vs. fifth epoch). Also, pursuing different training approaches (see 2.8.3) affected the tuning similarities, hence CNN-OMT had higher speed-tuning correlations vs. CNN-3Phases with moderately higher direction-tuning correlations.

One effect on tuning of the additional task cost in CNN-OMT (compared

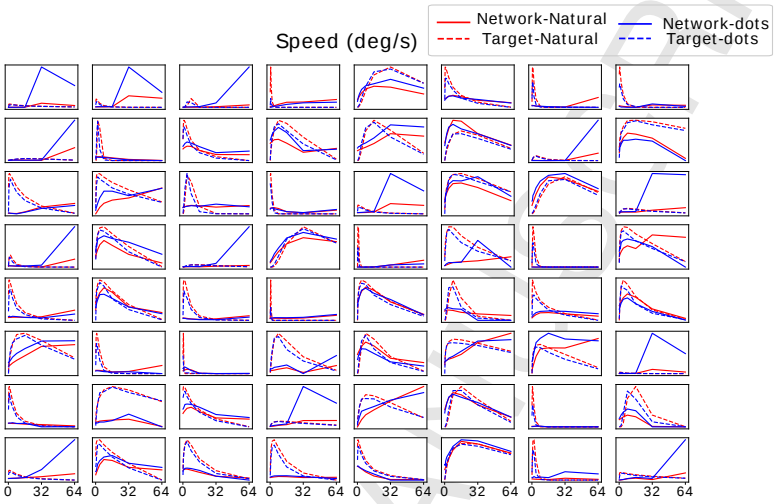


Figure 24: Speed-tuning curves of example units in CNN-OMT. Conventions as in Figure 21.

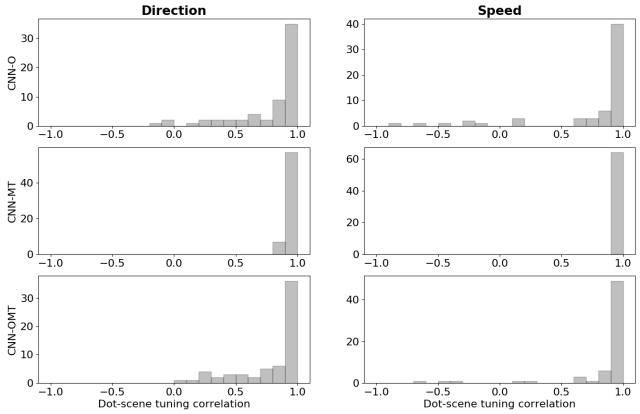


Figure 25: Correlations between tuning curves with dot stimuli and scene stimuli. Correlations for direction-tuning curves are shown on the left and those for speed-tuning curves are shown on the right. The correlations are frequently high in the CNN-O network (top), and very high in the CNN-MT network (middle). However, those in the CNN-OMT network are nearly uncorrelated on average, indicating that tuning in this network is highly sensitive to details of the stimuli.

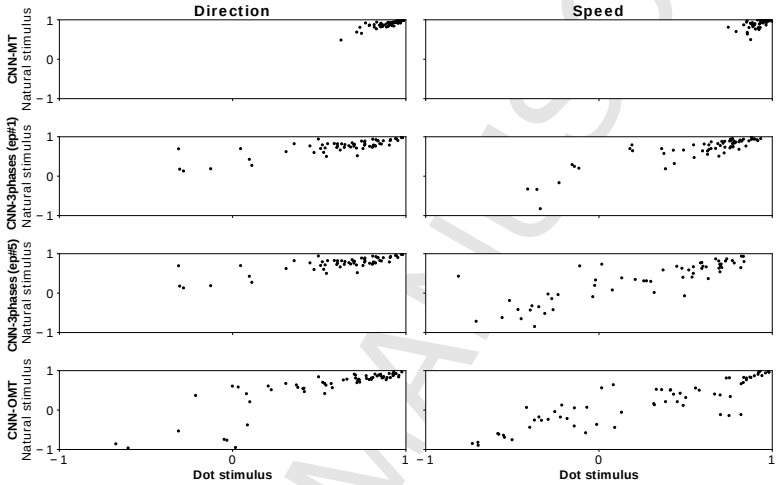


Figure 26: Correlations between empirical-model tuning curves and CNN tuning curves. Each point indicates these correlations for dot stimuli (horizontal axis) and natural-scene stimuli (vertical axis). Higher correlations mean that the tuning curves more closely reflect the empirical model. This is related to the regression error, but it differs due to the very different distributions of tuning-curve stimuli vs. training stimuli (for example, most training stimuli are not at the units' preferred speed or direction). The top row shows correlations for CNN-MT. Individual units' correlations are similar for dot and natural-scene stimuli. The second and third rows respectively show correlations between empirical-model and network tuning curves in CNN-3Phases after the first and fifth training epochs of the third training phase (i.e. training only for odometery task). As the third training phase progresses, the correlations for many units become weaker. The bottom row shows correlations between empirical-model and network tuning curves in CNN-OMT. Correlations in direction tuning, especially in response to natural scenes, include many high values. However speed-tuning correlations are more spread. Comparing the two bottom rows demonstrates how our two approaches of training on both MT activity and odometery affects the correlations. The CNN-OMT direction-tuning curves are less similar to those of the empirical model, especially in response to natural scenes, whereas the CNN-3Phases speed-tuning curves are more different from the empirical model, especially in response to dot stimuli.

to CNN-MT) was to reduce the depths of the direction and speed tuning curves. When tuning curves were normalized to the peak of their targets, the standard deviation of CNN-OMT direction-tuning curves averaged 0.32 (vs. 0.63 for CNN-MT). Similarly, the standard deviation of the normalized CNN-OMT speed-tuning curves averaged 0.71 (vs. 0.67 for CNN-MT). The MT cost affected tuning. For example the speed tuning curves are less uniformly high-pass in CNN-OMT than in CNN-O. However, the MT cost did not make tuning realistic in either CNN-OMT or CNN-3Phases. The two cost terms may have exerted conflicting influences on tuning during training, suggesting either a limitation of the model or the training algorithm, or low specialization of MT for visual odometry.

3.7. Sensitivity Analysis

We studied the sensitivity of visual odometry performance to changes in the speed-tuning-width and direction-tuning-width distributions, using networks that had the empirical model as input. Both of these were drawn from gamma distributions (see Table 1). A gamma distribution has two parameters, shape and scale. We altered each of these two gamma distributions by changing their scale parameters. Hence, the distribution of speed-tuning widths and direction-tuning bandwidths changed and we could examine how these changes influenced odometery performance.

Figure 27 illustrates the RMSE of the odometery task vs. different scale parameters for gamma distribution of direction-tuning widths. We changed the bandwidth by multiplying the original scale parameter, which we had estimated from literature, by each of $[0.25, 0.5, 1, 2, 4]$. The ∞ symbol refers to not having any direction selectivity in the model (i.e. direction bandwidth is infinite). We found the best performance at four times the original scale parameter. To verify this, we created another three populations, all using four times the original scale parameter. The average RMSE of these four populations was lower than other cases, although the 0.5x, 1x, and 2x means differed by less than five percent. We tested statistical significance of differences in mean absolute errors with each of these scale factors, compared to the 4x scale factor, using multiple t-tests. Only the 0.5x errors were significantly higher ($\alpha < .05$) with a Bonferroni correction for multiple comparisons.

Figure 28 shows the RMSE of the odometery task vs. different scale parameters for gamma distribution of the speed-tuning widths, where we

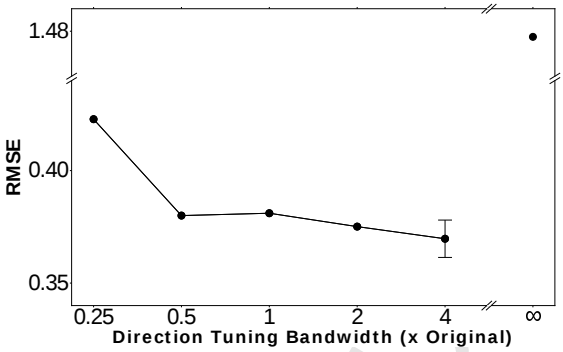


Figure 27: Task performance comparison with respect to changing direction-tuning-bandwidth distribution of the empirical model. To change the distribution we multiplied the original scale parameter of the modelled gamma distribution with $[0.25, 0.5, 1, 2, 4]$. The ∞ symbol refers to the case where we omitted direction selectivity of the response. For four times the original scale parameter case, we created four different populations (hence the error bar).

applied the same idea for speed-tuning widths as we did for the direction-tuning bandwidths. In this case, two times the original scale parameter outperformed the other cases in all four different populations that we tested. Mean absolute errors in the 2x case were significantly lower than all other cases ($\alpha < .05$), accounting for multiple comparisons. This suggests that odometry task performance is more sensitive to moderate modulations of speed-tuning widths than to similar modulations of direction-tuning widths. However, comparing the RMSEs of ∞-width cases of Figures 27 and 28, elimination of direction tuning had a noticeably larger impact than elimination of speed tuning.

4. Discussion

We developed a video-driven, empirical model of activity in the primate middle temporal area (MT) that emulates many tuning properties and statistics from the literature. The model uses well-supported tuning curves, and well-established computer-vision methods of generating represented signals such as speed and disparity.

As far as we know, this is the most thorough video-driven model of MT population activity developed so far. We expect that it will be useful in the future for examining relationships between features of MT population activity and performance of tasks that make use of visual motion information.

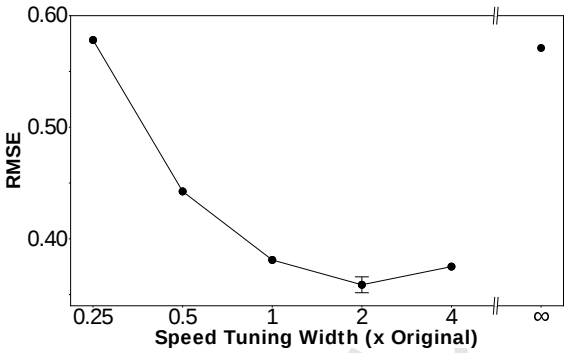


Figure 28: Task performance comparison with respect to changing speed-tuning-width distribution of the empirical model. To change the distribution we multiplied the original scale parameter of the modelled gamma distribution with $[0.25, 0.5, 1, 2, 4]$. The ∞ symbol refers to the case where we omitted speed selectivity of the response.

As a preliminary example, we showed in Figures 27 and 28 that estimation of ego-motion in our model is sensitive to speed tuning width, but fairly insensitive to direction tuning width over an order of magnitude. In general, embedding our model in deep networks that perform visually sophisticated tasks may help to clarify the functional significance of MT tuning properties.

Compared with other MT models (Perrone and Thiele, 2002; Tsui et al., 2010), a limitation of our approach is that its responses are not produced by biologically plausible mechanisms. That is, the model is empirical rather than mechanistic. This may impair the model's ability to generalize beyond the source data.

When we used the empirical model to train convolutional networks, the interaction between the task cost and the MT-regression cost was complex. We trained odometry networks to use the empirical MT model as input (Figures 27 and 28), and these performed as well as odometry networks with video input (Figure 15), indicating that our model of the MT representation is compatible with the odometry task. We also trained an odometry network with video input, and included another cost term that encouraged an intermediate layer to approximate the empirical model. In this case the task performance was barely affected, but the network failed to learn an MT-like intermediate representation. We believe this is a robust negative result. When we first trained networks with the MT cost alone, the representation degenerated with further training on the odometry cost (Figure 26). When we trained with both costs together, and then continued training without the

odometry cost, the MT approximation rapidly improved (Figure 17), while the odometry cost went up substantially. Combining the costs did not have a linear effect on unit tuning. For example, compared to CNN-MT, direction tuning was narrower in CNN-O, but wider in CNN-OMT (Figure 20). In summary, while we found that an MT-like representation supports the task, we were unable to produce a convolutional network that had both an MT-like internal representation and good task performance. It may be that a different training strategy is needed, or that more physiologically realistic mechanisms are needed earlier in the network, such as those in Baker and Bair (2016). A deeper network may be needed to reduce the coupling between the task and the MT representation. We also suspect that it is important for the network to perform a realistic range of tasks, rather than just visual odometry. Optimizing the MT representation for any single task may bias the representation toward properties that are useful for that task, rather than making it more realistic.

It would also be useful in future work to explore variations of our CNN-MT network, aiming to maximize similarity between target and actual tuning curves. In addition to standard hyperparameter tuning approaches, other potential avenues include balancing or weighting training data differently (corresponding more closely to tuning curves), using architectures that conform more closely to anatomy (Markov et al., 2014), and inclusion of more realistic mechanisms such as those in Baker and Bair (2016).

4.1. Alternative Regression Targets for Deep Representations

Training data for intermediate layers of deep networks could also be obtained directly from neural recordings (Arai et al., 1994; Yamins et al., 2014; McIntosh et al., 2016; Oliver and Gallant, 2016; Kindel et al., 2017) or from functional magnetic resonance imaging. An ideal neuron-level dataset would include hundreds of neurons, recorded chronically over at least tens of thousands of trials, with a variety of rich visual stimuli. It is not practical to collect such data in the macaque brain, as MT is located deep in a sulcus, preventing use of standard multielectrode arrays without damage to nearby visual areas. So far, recordings with up to 24-electrode arrays have been possible in the macaque (Cui et al., 2016). In marmosets, MT is located on the cortical surface, allowing the use of larger electrode arrays (Solomon et al., 2014; Chen et al., 2015; Chaplin et al., 2017). This may allow rich MT activity datasets in the future.

However, our approach has several advantages over potential large-scale MT recordings. One advantage is that the model properties can be modified, allowing investigation of the influence of individual response features on task performance. Also, empirical models allow specification of an attention field at run-time. This should allow generation of attention-modulated activity labels that are consistent with the attention focus of a network, rather than the (perhaps different and/or unknown) attention focus of the animal. Finally, the model provides infinite labelled data at low cost.

Acknowledgements

This work was supported by Mitacs and CrossWing Inc.

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Akaike, H. (1974). A New Look at the Statistical Model Identification. *Automatic Control, IEEE Transactions on*, 19(6):716–723.

Arai, K., Keller, E. L., and Edelman, J. A. (1994). Two-dimensional neural network model of the primate saccadic system. *Neural Networks*, 7(6-7):1115–1135.

Bair, W. and Koch, C. (1996). Temporal precision of spike trains in extrastriate cortex of the behaving macaque monkey. *Neural Computation*, 8(6):1185–1202.

Baker, P. M. and Bair, W. (2016). A Model of Binocular Motion Integration in MT Neurons. *The Journal of Neuroscience*, 36(24):6563–6582.

Borji, A. and Itti, L. (2013). State-of-the-art in visual attention modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):185–207.

Born, R. T. and Bradley, D. C. (2005). Structure and function of visual area MT. *Annual Review of Neuroscience*, 28:157–89.

Boyraz, P. and Treue, S. (2011). Misperceptions of speed are accounted for by the responses of neurons in macaque cortical area MT. *Journal of Neurophysiology*, 105(3):1199–211.

Bradley, D. C. and Andersen, R. A. (1998). Center–surround antagonism based on disparity in primate area mt. *The Journal of Neuroscience*, 18(18):7552–7565.

Cadena, S. A., Denfield, G. H., Walker, E. Y., Gatys, L. A., Tolias, A. S., Bethge, M., and Ecker, A. S. (2017). Deep convolutional models improve predictions of macaque v1 responses to natural images. *bioRxiv*, page 201764.

Cadieu, C. F., Hong, H., Yamins, D. L. K., Pinto, N., Ardila, D., Solomon, E. A., Majaj, N. J., and DiCarlo, J. J. (2014). Deep Neural Networks Rival the Representation of Primate IT Cortex for Core Visual Object Recognition. *PLoS Computational Biology*, 10(12).

Chaplin, T. A., Allitt, B. J., Hagan, M. A., Price, N. S. C., Rajan, R., Rosa, M. G. P., and Lui, L. L. (2017). Sensitivity of neurons in the middle temporal area of marmoset monkeys to random dot motion. *Journal of Neurophysiology*, 118(3):1567–1580.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2016). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*.

Chen, L.-C., Papandreou, G., Schroff, F., and Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*.

Chen, S. C., Morley, J. W., and Solomon, S. G. (2015). Spatial precision of population activity in primate area mt. *Journal of neurophysiology*, 114(2):869–878.

Chollet, F. et al. (2015). Keras. `https://github.com/fchollet/keras`.

Cui, Y., Liu, L. D., Khawaja, F. a., Pack, C. C., and Butts, D. a. (2013). Diverse suppressive influences in area MT and selectivity to complex motion features. *Journal of Neuroscience*, 33(42):16715–16728.

Cui, Y., Liu, L. D., McFarland, J. M., Pack, C. C., and Butts, D. A. (2016). Inferring cortical variability from local field potentials. *Journal of Neuroscience*, 36(14):4121–4135.

Czuba, T. B., Huk, A. C., Cormack, L. K., and Kohn, A. (2014). Area MT Encodes Three-Dimensional Motion. *The Journal of Neuroscience*, 34(47):15522–33.

Dayan, P. and Abbott., L. F. (2001). *Theoretical Neuroscience*. MIT Press.

De Valois, R. L., Albrecht, D. G., and Thorell, L. G. (1982a). Spatial frequency selectivity of cells in macaque visual cortex. *Vision research*, 22(5):545–559.

De Valois, R. L., Yund, E. W., and Hepler, N. (1982b). The orientation and direction selectivity of cells in macaque visual cortex. *Vision research*, 22(5):531–544.

DeAngelis, G. C. and Uka, T. (2003). Coding of horizontal disparity and velocity by MT neurons in the alert macaque. *Journal of Neurophysiology*, (2):1094–111.

Felzenszwalb, P. F. and Huttenlocher, D. P. (2006). Efficient belief propagation for early vision. *International journal of computer vision*, 70(1):41–54.

Güçlü, U. and van Gerven, M. A. J. (2017). Increasingly complex representations of natural movies across the dorsal stream are shared between subjects. *NeuroImage*, 145 Part B:6–13.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE.

Hirschmuller, H. (2005). Accurate and efficient stereo processing by semi-global matching and mutual information. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 807–814. IEEE.

Hong, H., Yamins, D. L. K., Majaj, N. J., and DiCarlo, J. J. (2016). Explicit information for category-orthogonal object properties increases along the ventral stream. *Nature Neuroscience*, 19(4):613–622.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456.

Khaligh-Razavi, S. M. and Kriegeskorte, N. (2014). Deep Supervised, but Not Unsupervised, Models May Explain IT Cortical Representation. *PLoS Computational Biology*, 10(11).

Kindel, W. F., Christensen, E. D., and Zylberberg, J. (2017). Using deep learning to reveal the neural code for images in primary visual cortex. pages 1–9.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Klindt, D., Ecker, A. S., Euler, T., and Bethge, M. (2017). Neural system identification for large populations separating "what" and "where". In *Advances in Neural Information Processing Systems*, pages 3509–3519.

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, 2(1-2):83–97.

Lucas, B. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proc 7th Intl Joint Conf on Artificial Intelligence*, pages 121–130.

Majaj, N. J., Carandini, M., and Movshon, J. A. (2007). Motion integration by neurons in macaque MT is local, not global. *The Journal of Neuroscience*, 27(2):366–70.

Markov, N. T., Ercsey-Ravasz, M. M., Ribeiro Gomes, a. R., Lamy, C., Magrou, L., Vezoli, J., Misery, P., Falchier, A., Quilodran, R., Gariel, M. a., Sallet, J., Gamanut, R., Huissoud, C., Clavagnier, S., Giroud, P., Sappey-Marinier, D., Barone, P., Dehay, C., Toroczkai, Z., Knoblauch, K., Van Essen, D. C., and Kennedy, H. (2014). A weighted and directed interareal connectivity matrix for macaque cerebral cortex. *Cerebral Cortex*, 24(1):17–36.

Martınez-Trujillo, J. C. and Treue, S. (2002). Attentional modulation strength in cortical area mt depends on stimulus contrast. *Neuron*, 35(2):365–370.

Marzat, J., Dumortier, Y., and Ducrot, A. (2009). Real-time dense and accurate parallel optical flow using CUDA. In *WSCG*.

Maunsell, J. H. and Van Essen, D. C. (1983). Functional properties of neurons in middle temporal visual area of the macaque monkey. I. Selectivity for stimulus direction, speed, and orientation. *Journal of Neurophysiology*, 49(5):1127–47.

Maunsell, J. H. and van Essen, D. C. (1987). Topographic organization of the middle temporal visual area in the macaque monkey: representational biases and the relationship to callosal connections and myeloarchitectonic boundaries. *Journal of Comparative Neurology*, 266(4):535–555.

Mayer, N., Ilg, E., Häusser, P., Fischer, P., Cremers, D., Dosovitskiy, A., and Brox, T. (2015). A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation.

McIntosh, L. T., Maheswaranathan, N., Nayebi, A., Ganguli, S., and Baccus, S. A. (2016). Deep Learning Models of the Retinal Response to Natural Scenes. *Advances in Neural Information Processing Systems 29 (NIPS)*, (Nips):1–9.

Movshon J, Adelson E, G. M. N. W. (1985). The analysis of moving visual patterns. In *Pattern Recognition Mechanisms. Eds. Chagas C, Gattass R, Gross C*, volume 54, pages 117–151. Rome:Vatican Press.

Newsome, W. T. and Pare, E. B. (1988). A selective impairment of motion perception following lesions of the middle temporal visual area (mt). *The Journal of Neuroscience*, 8(6):2201–2211.

Newsome, W. T., Wurtz, R. H., Dursteler, M., and Mikami, A. (1985). Deficits in visual motion processing following ibotenic acid lesions of the middle temporal visual area of the macaque monkey. *The Journal of Neuroscience*, 5(3):825–840.

Nichols, M. J. and Newsome, W. T. (2002). Middle Temporal Visual Area Microstimulation Influences Veridical Judgments of Motion Direction. *The Journal of Neuroscience*, 22(21):9530–9540.

Nishimoto, S. and Gallant, J. L. (2011). A three-dimensional spatiotemporal receptive field model explains responses of area MT neurons to naturalistic movies. *The Journal of Neuroscience*, 31(41):14551–64.

Nover, H., Anderson, C. H., and DeAngelis, G. C. (2005). A logarithmic, scale-invariant representation of speed in macaque middle temporal area accounts for speed discrimination performance. *The Journal of Neuroscience*, 25(43):10049–60.

Oliver, M. and Gallant, J. (2016). A deep convolutional energy model of v4 responses to natural movies. *Journal of Vision*, 16(12):876–876.

Pack, C. C. and Born, R. T. (2001). Temporal dynamics of a neural solution to the aperture problem in visual area MT of macaque brain. *Nature*, 409(6823):1040–2.

Pack, C. C., Hunter, J. N., and Born, R. T. (2005). Contrast dependence of suppressive influences in cortical area mt of alert macaque. *Journal of Neurophysiology*, 93(3):1809–1815.

Parzen, E. (1962). On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076.

Peli, E. (1990). Contrast in complex images. *Journal of the Optical Society of America. A, Optics and Image Science*, 7(10):2032–2040.

Perrone, J. a. and Thiele, A. (2002). A model of speed tuning in MT neurons. *Vision research*, 42(8):1035–51.

Priebe, N. J., Cassanello, C. R., and Lisberger, S. G. (2003). The neural representation of speed in macaque area MT/V5. *Journal of Neuroscience*, 23(13):5650–5661.

Priebe, N. J., Lisberger, S. G., and , J. A. (2006). Tuning for spatiotemporal frequency and speed in directionally selective neurons of macaque striate cortex. *The Journal of Neuroscience*, 26(11):2941–2950.

Raiguel, S., Hulle, M., Xiao, D.-K., Marcar, V., and Orban, G. A. (1995). Shape and spatial distribution of receptive fields and antagonistic motion surrounds in the middle temporal area (v5) of the macaque. *European journal of neuroscience*, 7(10):2064–2082.

Robson, J. (1966). Spatial and temporal contrast-sensitivity functions of the visual system. *Josa*, 56(8):1141–1142.

Rodman, H. R. and Albright, T. D. (1987). Coding of visual stimulus velocity in area mt of the macaque. *Vision research*, 27(12):2035–2048.

Rudolph, K. and Pasternak, T. (1999). Transient and permanent deficits in motion perception after lesions of cortical areas mt and mst in the macaque monkey. *Cerebral Cortex*, 9(1):90–100.

Rust, N. C., Mante, V., Simoncelli, E. P., and Movshon, J. A. (2006). How MT cells analyze the motion of visual patterns. *Nature Neuroscience*, 9(11):1421–31.

Silverman, B. W. (1986). *Density estimation for statistics and data analysis*, volume 26. CRC press.

Simoncelli, E. P. and Heeger, D. J. (1998). A model of neuronal responses in visual area MT. *Vision Research*, 38(5):743–761.

Sincich, L. C., Park, K. F., Wohlgemuth, M. J., and Horton, J. C. (2004). Bypassing v1: a direct geniculate input to area mt. *Nature neuroscience*, 7(10):1123–1128.

Smith, M. A., Majaj, N. J., and Movshon, J. A. (2005). Dynamics of motion signaling by neurons in macaque area MT. *Nature Neuroscience*, 8(2):220–8.

Solomon, S. S., Chen, S. C., Morley, J. W., and Solomon, S. G. (2014). Local and global correlations between neurons in the middle temporal area of primate visual cortex. *Cerebral Cortex*, 25(9):3182–3196.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research (JMLR)*, 15:1929–1958.

Sun, D., Roth, S., and Black, M. J. (2010). Secrets of optical flow estimation and their principles. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2432–2439. IEEE.

Treue, S. and Martínez Trujillo, J. (1999). Feature-based attention influences motion processing gain in macaque visual cortex. *Nature*, 399(575-579).

Tripp, B. (2016). A convolutional model of the primate middle temporal area. In *ICANN*.

Tripp, B. P. (2012). Decorrelation of Spiking Variability and Improved Information Transfer through Feedforward Divisive Normalization. *Neural Computation*, pages 1–27.

Tripp, B. P. (2017). Similarities and differences between stimulus tuning in the inferotemporal visual cortex and convolutional networks. In *Neural Networks (IJCNN), 2017 International Joint Conference on*, pages 3551–3560. IEEE.

Tsui, J. M. G., Hunter, J. N., Born, R. T., and Pack, C. C. (2010). The role of V1 surround suppression in MT motion integration. *Journal of neurophysiology*, 103(6):3123–38.

Wang, H. X. and Movshon, J. A. (2016). Properties of pattern and component direction-selective cells in area MT of the macaque. *Journal of Neurophysiology*, page 74.2/OO9.

Xiao, D., Raiguel, S., Marcar, V., and Orban, G. (1997). The Spatial Distribution of the Antagonistic Surround of MT / V5 Neurons. *Cerebral Cortex*, 7:662–677.

Xu, K., Ba, J. L., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R. S., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *ICML-2015*.

Yamins, D. L. and DiCarlo, J. J. (2016). Using goal-driven deep learning models to understand sensory cortex. *Nature neuroscience*, 19(3):356.

Yamins, D. L. K., Hong, H., Cadieu, C. F., Solomon, E. a., Seibert, D., and Dicarlo, J. J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex. *PNAS*.

Žbontar, J. and LeCun, Y. (2016). Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 17:1–32.