PNAS
www.pnas.org

Supplementary Information for

A goal-driven modular neural network predicts parietofrontal neural dynamics during grasping

Jonathan A. Michaels, Stefan Schaffelhofer, Andres Agudelo-Toro, Hansjörg Scherberger

Corresponding Author: Hansjörg Scherberger
Email: HScherberger@dpz.eu

This PDF file includes:

Supplementary Information Text
Figures S1 to S6

Supplementary Information Text

Experimental setup

We developed an experimental setup that allowed us to present many graspable objects to the monkeys while monitoring their behavior, neural activity, and hand kinematics. Monkeys grasped a total of 42 different objects of equal weight that were placed on interchangeable turntables (Figure 1C), presented in random order for each recording session. Objects were of different shapes and sizes including rings, cubes, spheres, horizontal cylinders, vertical cylinders, and bars. A special turntable held objects of abstract forms, which differed visually, but required almost identical hand configurations for grasping. The abstract turntable was not presented for every session of monkey Z. A mixed turntable held objects of different shapes of average size, but this turntable was excluded from the analyses in this study, since all objects were present on other turntables. Monkeys were also trained on a grasping box that cued one of two grasping types, power or precision grip, but for simplicity this data was also not included in the current study.

Monkeys were trained to grasp 42 objects in a delayed grasp, lift, and hold task (Figure 1C,D). While sitting in the dark the monkeys could initiate a trial (self-paced) by placing their grasping hand (left hand in monkey Z, right hand in monkey M) onto a rest sensor that enabled a fixation LED close to the object. Looking at (fixating) this spot for a variable time activated a spot light that illuminated the graspable object. After the light was turned off the monkeys had to withhold movement execution until the fixation LED blinked for 100 ms. After this, the monkeys released the rest sensor, reached for and grasped the object and briefly lifted it up (500 ms). The monkeys had to fixate the LED throughout the task (max. deviation: ~5 deg of visual angle). All correctly executed trials were rewarded with a liquid reward (juice) and monkeys could initiate the next trial after a short delay. Error trials were immediately aborted without reward and excluded from the analysis.

Finger, hand, and arm kinematics of the acting hand were tracked with an instrumented glove for small primates. Eight magnetic sensor coils (model WAVE, Northern Digital) were placed onto the fingernails, the hand's dorsum as well as the wrist to compute the centers of 18 individual joints in 3D space, including thumb, digits, wrist, elbow and shoulder. The method and its underlying computational model have been described previously (1). Recorded joint trajectories were then used to drive a 3D-musculoskeletal model (2, 3), which was adjusted to the specific anatomy of each monkey. The model was implemented in OpenSim (4) and allowed extracting a total of 27 DOF in joint angle space, and 50 DOF in muscle tendon length space. All extracted joint angles and muscle lengths were sampled at 100 Hz and low-pass filtered (2^{nd}-order Butterworth filter, 3 Hz low-pass).

Electrophysiological recordings

Single and multiunit activity was recorded simultaneously using floating microelectrode arrays (FMA, Microprobe Inc., Gaithersburg, MD, USA). In each monkey we recorded 192 channels from 6 individual arrays implanted into the cortical areas AIP, F5, and M1 (Figure 1B). In each array, the lengths of the electrodes increased towards the sulcus and ranged from 1.5 (1^{st} row) to 7.1 mm (4^{th} row). In area F5, one array was placed in the posterior bank of the inferior arcuate sulcus approximately targeting F5a (longer electrodes) and approaching the F5 convexity (F5c; shorter electrodes). The second and more dorsally located array was positioned to target F5p. In AIP, the arrays were implanted into the end of the posterior intraparietal sulcus at the level of area PF and more dorsally at the level of area PFG. In M1, both arrays were placed into the hand area of M1 into the anterior bank of the central sulcus at the level of the spur of the arcuate sulcus. Surgical

procedures have been described previously (5). Neural activity was recorded at full bandwidth with a sampling frequency of 24 kHz and a resolution of 16 bits (model: RZ2 BioAmp Processor; Tucker Davis Technologies, FL, USA). Neural data was synchronously stored to disk together with the behavioural and kinematic data. Raw recordings were filtered offline (bandpass cutoff: 0.3–7 kHz) before spikes were detected (threshold: 3.5x std) and extracted. Spike sorting was processed in two steps: First, we applied super-paramagnetic clustering (6) and then revised the results by visual inspection using Offline Sorter (Plexon, TX, USA) to detect and remove neuronal drift and artefacts. No other pre-selection was applied and single and multiunit activity were analyzed together.

Visual and muscle feature analysis

In order to model the visual features of the objects being presented in the grasping task, we generated simulated images from the monkey's perspective (Fig. S1). We preprocessed and fed these images into a convolutional neural network (CNN), VGG (7), that used spatial convolution over pixels to classify objects in an image into a set of predefined categories. VGG was pre-trained on ImageNet (8), a massive set of labeled images. We did not train VGG on our images.

To test how well features within the layers of VGG could explain neural activity averaged over the period where the objects were presented (cue period), we first transformed the responses of the CNN to the presentation of all of our objects into its first 20 principal components (separately for each layer), which explained 91-99% of the signal variance. Next, we regressed the average spike rate over the cue period of each unit (for all single trials) onto the features of each layer separately (Matlab function *fitrlinear*), using leave-one-out cross-validation. All regressions had a standard L2 (ridge) penalty of $\lambda = 1/n$, where n was the number of in-fold observations. We then took the median r-value over all units within a recording session, and plot the mean of those values across recording sessions in Figure S2A.

In order to make comparisons between regions, we must control for differences in recording quality. Therefore, we generated a conservative estimate of the noise ceiling for each recorded unit. Instead of regressing neural activity onto features, we simply correlated the firing rate of every single trial with the mean firing rate of that unit for each condition, over all trials simultaneously. In other words, we assume that the best a given model can do is to predict the true mean of the neural activity for each condition. Finally, the results of the regression in Figure S2a were normalized on a per-neuron basis by the noise ceiling, the median taken over all units in an area, and the mean across recording sessions plotted in Figure 2B.

For the analysis of muscle kinematics in Figure 2D we performed the same regression analysis using the condition averaged muscle velocity of all 50 muscles during movement initiation (average of the time window from 200 ms before to 200 ms after movement onset) to predict average neural firing rate during the same time period.

It should be noted that since we take the median prediction across units before taking the average across recording sessions for the analyses in Figure 2B,D and Figure S2A,B, the variability in the prediction of individual unit activity from the visual and muscle features is not captured by the standard error, which rather captures the variability across recording sessions.

Modular recurrent neural network

In order to model the planning and execution of a grasping task, we implemented the dynamical system, $\dot{x} = F(x, u)$, using a standard continuous RNN equation of the form

$$\tau \dot{\mathbf{x}}_i(t) = -x_i + \sum_{k=1}^{N} J_{ik} r_k(t) + \sum_{k=1}^{I} B_{ik} u_k(t) + b_i^x \quad (1)$$

where the network has N units and I inputs, x are the activations and r the firing rates in the network, which were related to the activations by either the rectified hyperbolic tangent function (ReTanh), such that $r = \{0, x < 0;\ tanh\,(x), x \geq 0\}$, or a rectified linear function (ReLU), such that $r = \{0, x < 0;\ x, x \geq 0\}$. The units in the network interact using the synaptic weight matrix, J. The inputs are described by u and enter the system by input weights, B. Each unit has an offset bias, $b_i{}^x$. The time integration constant of the network is τ.

For all simulations N was fixed at 300, where each module contained 100 units (N_m). The inputs were a condition-independent hold signal that was released 200 ms before movement onset and was sent to all modules, and a 20-dimensional signal representing the visual features of the current visual stimulus that was sent only to the input module. The elements of B were initialized to have zero mean (normally distributed values with $SD = 1/\sqrt{I}$). The elements of J were initialized to have zero mean (normally distributed values with $SD = g/\sqrt{N_m}$) within each module, normally distributed with $SD = 1/\sqrt{N_m}$ between each connected module, and zero for all other connections. The synaptic scaling factor, g, was set at 1.2 following previous work (9). We used a fixed time constant of 100 ms for τ, with Euler integration every 10 ms. In addition, the sparsity of the connectivity between (but not within) modules was manipulated for the results in Figure S5.

The network was required to generate average muscle velocities in 50 dimensions until 400 ms after movement onset, where movement onset was determined by a threshold crossing in elbow position that approximately corresponded to the hand lifting off the handrest. The output of the network was defined as a linear readout of the output module

$$z_i(t) = \sum_{k=1}^{N} W_{ik}\ r_k(t) + b_i^z \quad (2)$$

where z represents the 50-dimensional muscle velocity signal and is a linear combination of the internal firing rates using weight matrix W, which was initialized with near zero entries, and $b_i{}^z$, which is a bias term for each output dimension.

All non-zero values of the input weights, B, internal connectivity, J, output weights, W, and all biases, were trained using Hessian free optimization (10) (code: https://github.com/JonathanAMichaels/hfopt-matlab) also utilized in previous work (11, 12). The error function used to optimize the network considered the squared error between the output of the linear readout and the desired muscle velocity profiles, v,

$$E = \frac{1}{CMT} \sum_{c=1}^{C} \sum_{m=1}^{M} \int_0^T \left(z_m(c,t) - v_m(c,t)\right)^2 dt \quad (3)$$

across all time, T, all muscles, M, and all conditions (trials), C, the networks were trained on. We report normalized error, which is the sum of the squared error from Eq. 3 over all times, dimensions, and trials, divided by the total variance of the target signal. In addition to the above error signal, we also implemented two commonly used regularizations. The penalties were a

standard L2 cost on the firing rates, R_{FR} , to keep units from saturating, and a standard L2 cost on the input and output weights, R_{IO}. Therefore, the total error function minimized during training was

$$E^R = E + \alpha R_{FR} + \beta R_{IO} \quad (4)$$

The hyper-parameter values, α and β, were varied systematically to test the effect of each regularization. The two regularizations were defined as

$$R_{FR} = \frac{1}{CNT} \sum_{c=1}^{C} \sum_{i=1}^{N} \int_0^T r_i(c,t)^2 dt \quad (5)$$

and

$$R_{IO} = \sum_{i,j=1}^{N,I} B_{ij}^2 + \sum_{i,j=1}^{M,N} W_{ij}^2 \quad (6)$$

Similar to previous work, we opted not to model any feedback, since the goal of the study was to illustrate the main points parsimoniously and without relying on confronting the issue of what kind of feedback is most biologically plausible in such a network. All networks were trained until the change in objective from one iteration to the next fell below 1e-6.

Assessing similarity of model and neural data

Imagine we have two shapes in front of us, for example a square and a triangle, consisting of the set of two dimensional points that make up each shape ($X \in \Re^{P,2}$, $Y \in \Re^{P,2}$, P - points). If we would like to see if it's possible to overlap the triangle on the square *without distorting the overall shapes*, procrustes analysis provides a method for finding the optimal rotation that aligns them in arbitrary dimensionality. First, any differences in position and scale are removed by centering each object on a common coordinate system and by scaling all points by the Frobenius norm (yielding $\hat{X}$and $\hat{Y}$). Then, the optimal rotation matrix, R, to align the triangle with the square can be found as follows: $A = \hat{X}^T\hat{Y}$, where the singular value decomposition $A = U\Sigma V^T$ yields $R = UV^T$subject to $det(R) = 1$, making it a special orthogonal matrix. We can then quantify the success of the rotation by calculating $1 - \left|\left|\hat{X} - \hat{Y}R\right|\right|_F^2 / \left|\left|\hat{X}\right|\right|_F^2$, which would yield 1 for a perfect fit. This method is ideal for comparing model and neural data, where the square represents the activity of model data and the triangle of neural data ($X \in \Re^{CT,N}$, $Y \in \Re^{CT,N}$, C - conditions, T - time points, N - neurons). In every instance where this analysis was used, the time points cover the entire trial aligned to two events, cue and movement, following the time period in Figure 3. This procedure can only take place if the number of neurons in the model and neural data sets is equivalent. In the case that the number of neurons in the model data was less than the neural data, we padded those columns with zeros. In the case that the number of neurons in the model data was greater, we first performed PCA on the model data and truncated the number of principal components to be equal to the number of neurons in the neural data, which in every case explained >99% of the variance in the model data. Procrustes analysis was used in three different metrics in the results, Overall Fit, Area-wise Fit, and Inter-area Fit, and are explained in

detail starting with Figure 3.

Additionally, for the analysis of Inter-area Fit we needed to calculate an estimate of the expected similarity between each pair of brain regions in the neural data. To calculate this estimate for each recording session over the entire trial (time window and alignment as in Figure 3), we resampled trials within each condition (with replacement, equal to the number of recorded trials in each condition), condition-averaged the data, then performed pairwise procrustes between each pair of brain regions using this resampled data. In the case that the number of neurons did not match between regions, we did as described in the previous paragraph to equalize the number of neurons. This resampling procedure was repeated 100 times per recording session, and the procrustes fit between regions was averaged over these repetitions to produce the 3 by 3 similarity matrices used to evaluate how well the modules matched the brain regions they were predicted to explain (e.g. Fig 3F,H). The final Inter-area Fit value was calculated as the correlation between the similarity matrix generated from this resampling procedure and one generated when using the same procrustes procedure to fit the model data of each module to the neural data in each brain region.

Fixed point analysis

To identify the mechanisms guiding the computations within our models, we searched for fixed points using standard nonlinear dynamical systems methods combined with linear stability analysis, as has been described in detail previously (11, 13). We performed an optimization to find a set of points in the high-dimensional state space, $\{x^{1*}, x^{2*}, \ldots\}$, where the dynamics described in Eq. 1 are at an approximate equilibrium, $\dot{x}^* = F(x^*, u^{const}) = 0$, for a constant input to the system. For some volume around these points, Eq. 1 can be replaced by a linear dynamical system, $\dot{\delta}x = M\delta x$, with $\delta x = x - x^*$ and $M = F'(x^*)$, by definition. These points are considered fixed points if their speed is very slow relative to the speed of the network during normal operation (>1000 times slower for most of our results).

For each time epoch of interest, we repeated the optimization to find fixed points many times (100 repetitions) using optimization starting points sampled randomly from activity the networks visited during normal operation. In general, this process yielded a single unique fixed point during the memory period and a single fixed point during the movement period. In some cases, tight clusters of fixed points (2-3) with similar properties and locations in the neural state space were considered a single fixed point.

To test how well the dynamics of the linear system matched the network model (Fig. 4C), we simply calculated how much normalized variance in the network model could be explained by the linear dynamics around each fixed point. In order to test the individual contributions of each eigenvalue of each fixed point, we sequentially removed each eigenvector from the linear system using the following procedure: 1) Perform eigendecomposition on the Jacobian of the linear system at each fixed point, yielding $J = Q\Lambda Q^{-1}$, where the columns of Q correspond to the eigenvectors of J and Λ is a diagonal matrix of eigenvalues. 2) Replace the eigenvalue of interest in Λ with -1, causing contraction along the corresponding eigenvector at the timescale of the network. If the eigenvalue of interest was part of a complex conjugate pair, we set both eigenvalues in the pair to -1. 3) Reconstruct J and run the linear dynamics. 4) Calculate the amount of normalized variance explained in the network model by the linear dynamics (Fig. 4C).

The above procedure allowed us to identify the eigenvalues that were important matching the dynamics of the network. To further determine the relative contribution of each module to each eigenvalue of a given fixed point, we performed an iterative analysis where the synaptic weights between all units within a module were multiplied by a damping factor in the range of 0.5

to 1, the Jacobian recalculated, and linear stability analysis performed and visualized to see how the eigenvalues of the system changed after damping (Fig. 4E).

Supplemental Figures

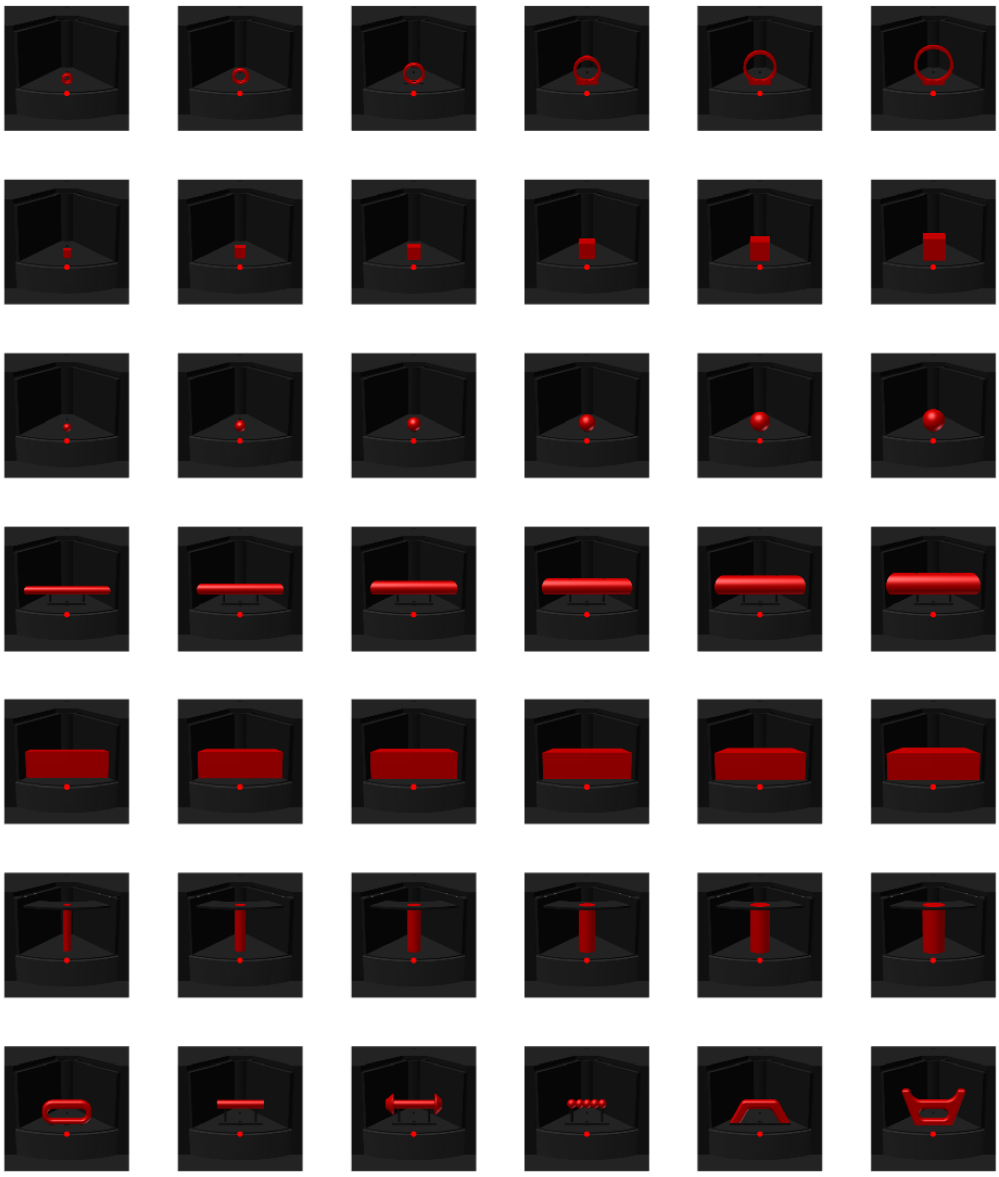

Fig. S1 | Simulated monkey view of objects used as input for VGG. Physical objects were CNC manufactured based on mesh models. Red fixation point was added in the approximate location that it was presented to the animals. All input images were RGB and 227x227 pixels in size.

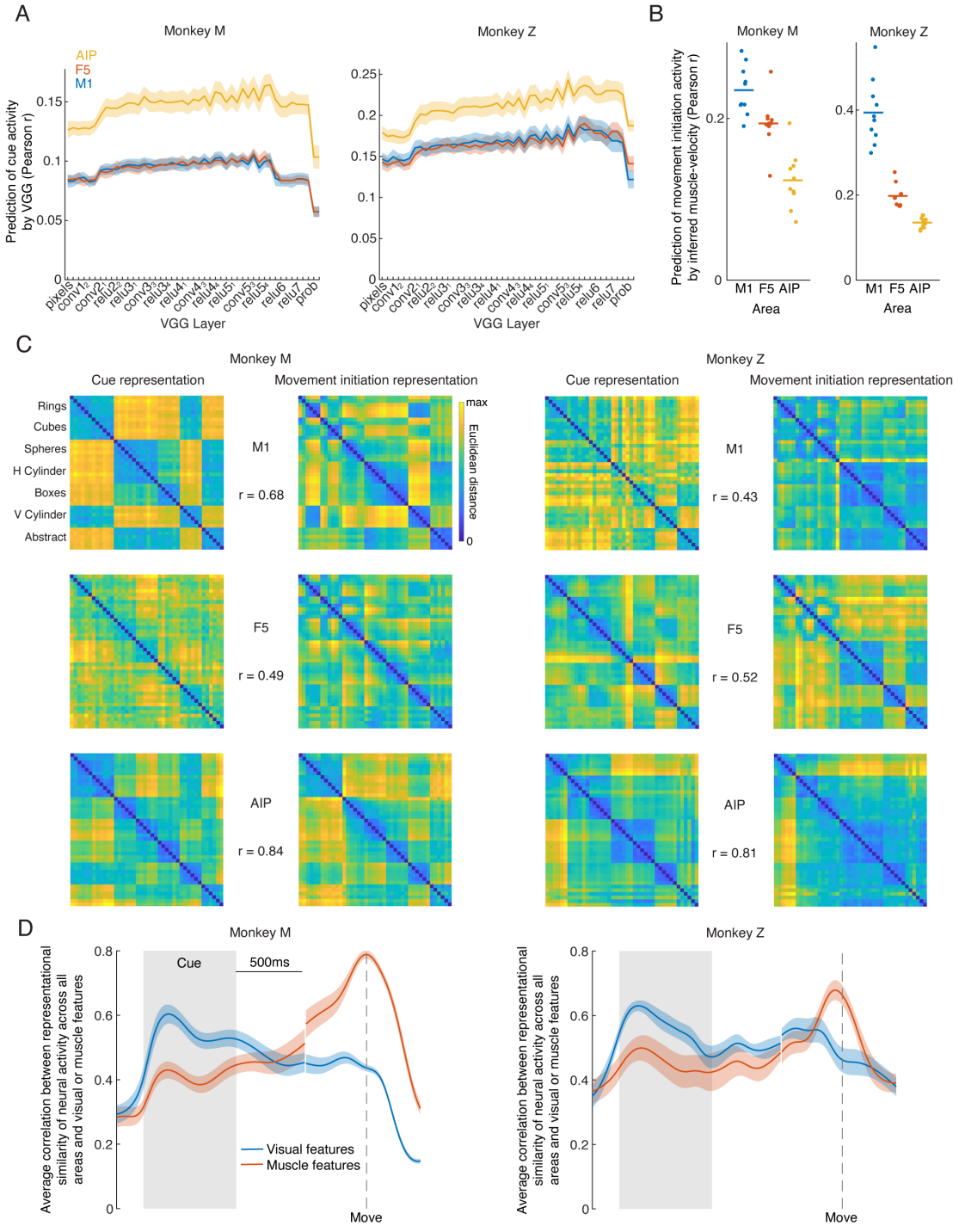

Fig. S2 | Visual and kinematic features explain neural activity across the fronto-parietal grasping circuit. (A) Single-trial neural activity of each unit averaged during the cue period was regressed (leave-one-out cross-validated) against the representation of all objects in each layer of the CNN (first 20 principal components), and the median fit was taken over all units within one recording session. The solid line and error surfaces represent the mean and s.e.m. over all recording sessions of each monkey. (B) Single-trial neural activity of each unit averaged during

the movement initiation period (200 ms before to 200 ms after movement onset) was regressed (leave-one-out cross-validated) against the muscle velocity of all grasping conditions averaged over the same time period. Each point represents one recording session of each monkey. (C) Representational similarity matrices were generated by calculating the Euclidean distance between each pair of conditions (example session M7 and Z9) in the full neural space of the mean activity of each condition in each area. r-values represent the correlation between the upper triangle of the representational similarity matrices between the cue and movement initiation periods. (D) Average correlation between the representational similarity matrix (Euclidean distance) of neural data across all brain regions with the representational similarity matrix of either visual features (VGG layer relu5_4) or mean inferred muscle velocity around movement onset (200 ms before to 200 ms after). The solid line and error surfaces represent the mean and s.e.m. over all recording sessions of each monkey.

Fig. S3 | Feature representation in VGG. Representation of the features of all conditions in the first two principal components of each layer in VGG with corresponding variance explained.

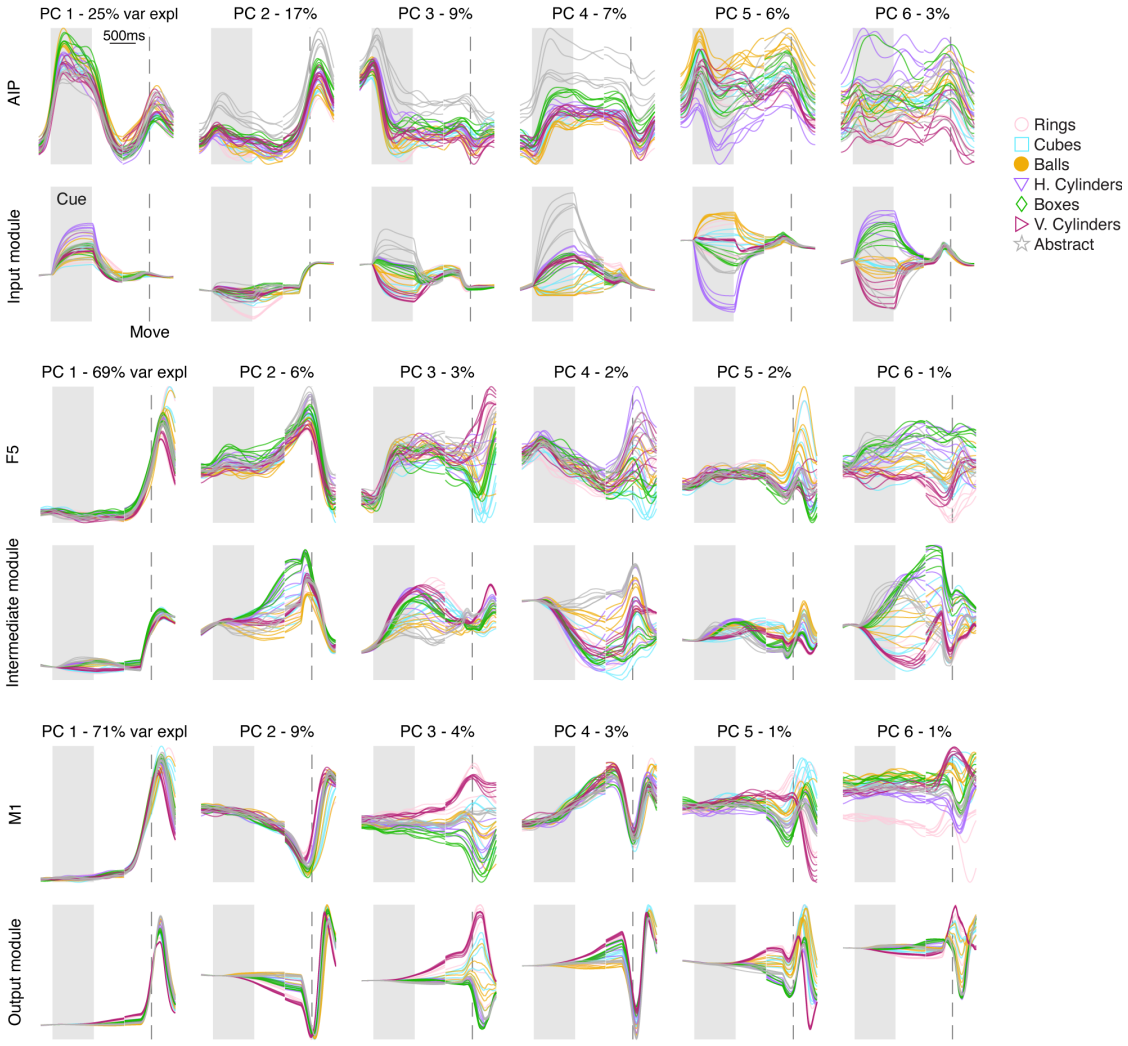

Fig. S4 | Area-wise ability of mRNN model to explain neural data in AIP, F5, and M1. Procrustes analysis comparing the dynamics of each module in an example model (Fig. 3) to the neural data of each brain region (session M2). For visualization purposes, after model data was fit to neural data it was projected onto the first 6 PCs defined on the neural data, and percentages show variance explained in the neural data per PC. The multiple traces for each type of object represent the different sizes within a turntable.

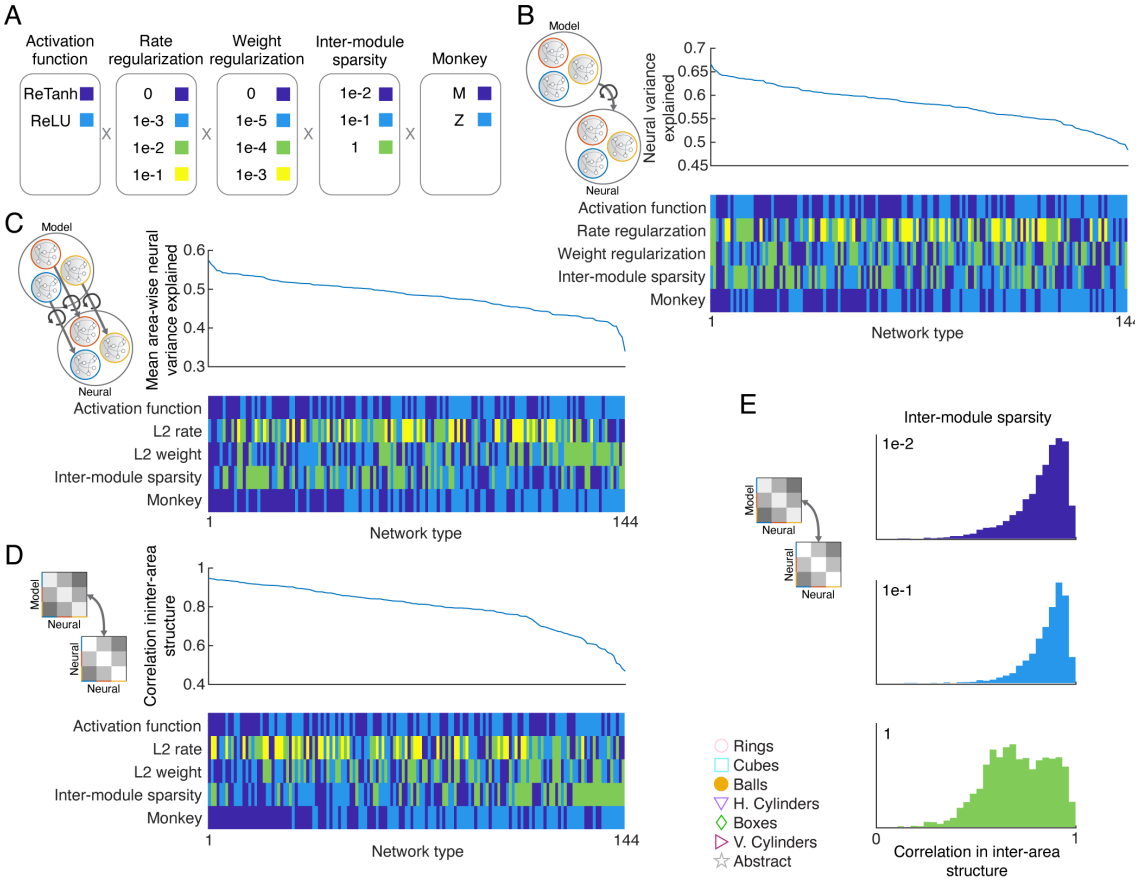

Fig. S5 | Effect of network parameters on the ability of the mRNN model to match the grasping circuit. (A) 720 networks were trained (5 repetitions of each network type), varying the activation function, rate and weight regularizations, inter-module sparsity, and the monkey being modeled. Note that the highest level of weight regularization was excluded, since task performance was severely affected. (B-D) Average performance across recording sessions for each of the three proposed metrics sorted from best to worst across all networks. (E) The effect of inter-module sparsity on the Inter-Area Fit. (F) Procrustes analysis (Overall Fit) comparing the dynamics of two exemplar mRNN models to neural data across all brain regions (session

M2). For visualization purposes, after model data was fit to neural data it was projected onto the first 6 PCs defined on the neural data, and percentages show variance explained in the neural data per PC. (G) Pairwise procrustes was performed between each brain region and a resampled version of its own activity, or between each module and brain region (I,K, Inter-Area Fit). Individual rows and columns specify from top to bottom and from left to right either the output, intermediate, and input module, or M1, F5, and AIP, respectively. (H) Exemplar model with the parameters (ReLU activation function, L2 rate regularization - 1e-1, L2 weight regularization - 1e-5, inter-module sparsity - 1). (J) Exemplar model with the parameters (ReTanh activation function, rate regularization - 0, weight regularization - 0, inter-module sparsity - 0.1). For F,H,J, the multiple traces for each type of object represent the different sizes within a turntable.

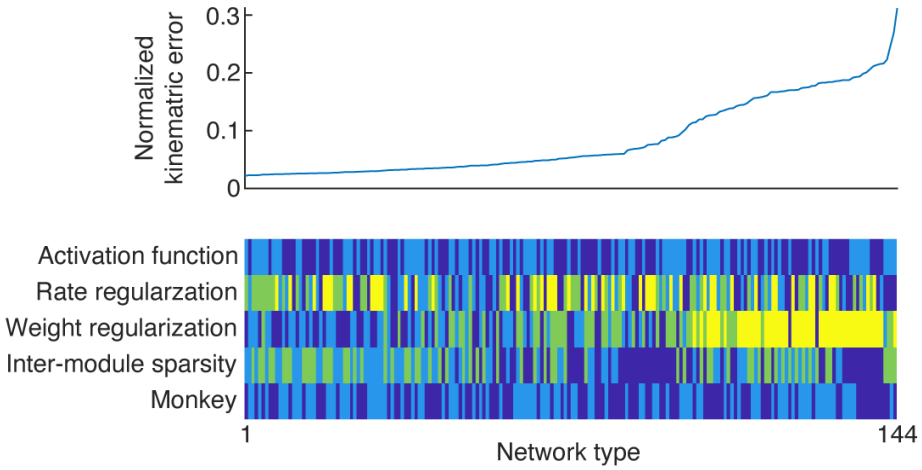

Fig. S6 | Normalized kinematic error across choice of network parameters. Average kinematic error for networks presented in Figure S5. For color code, see Fig. S5A.

References

1. S. Schaffelhofer, H. Scherberger, A new method of accurate hand- and arm-tracking for small primates. *J. Neural Eng.* **9**, 026025 (2012).

2. S. Schaffelhofer, M. Sartori, H. Scherberger, D. Farina, Musculoskeletal representation of a large repertoire of hand grasping actions in primates. *IEEE Trans. Neural Syst. Rehabil. Eng.* **23**, 210–220 (2015).

3. K. R. S. Holzbaur, W. M. Murray, S. L. Delp, A model of the upper extremity for simulating musculoskeletal surgery and analyzing neuromuscular control. *Ann. Biomed. Eng.* **33**, 829–840 (2005).

4. S. L. Delp, *et al.*, OpenSim: open-source software to create and analyze dynamic simulations of movement. *IEEE Trans. Biomed. Eng.* **54**, 1940–1950 (2007).

5. S. Schaffelhofer, A. Agudelo-Toro, H. Scherberger, Decoding a wide range of hand configurations from macaque motor, premotor, and parietal cortices. *J. Neurosci.* **35**, 1068–1081 (2015).

6. R. Q. Quiroga, Z. Nadasdy, Y. Ben-Shaul, Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. *Neural Comput.* **16**, 1661–1687 (2004).

7. K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv [cs.CV]* (2014).

8. J. Deng, *et al.*, ImageNet: A large-scale hierarchical image database in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, (2009), pp. 248–255.

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

10. J. Martens, I. Sutskever, Learning recurrent neural networks with hessian-free optimization in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, (2011), pp. 1033–1040.

11. D. Sussillo, M. M. Churchland, M. T. Kaufman, K. V. Shenoy, A neural network that finds a naturalistic solution for the production of muscle activity. *Nat. Neurosci.* **18**, 1025–1033 (2015).

12. J. A. Michaels, B. Dann, H. Scherberger, Neural Population Dynamics during Reaching Are Better Explained by a Dynamical System than Representational Tuning. *PLoS Comput. Biol.* **12**, e1005175 (2016).

13. D. Sussillo, O. Barak, Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks. *Neural Comput.* **25**, 626–649 (2013).