Supplementary Figures

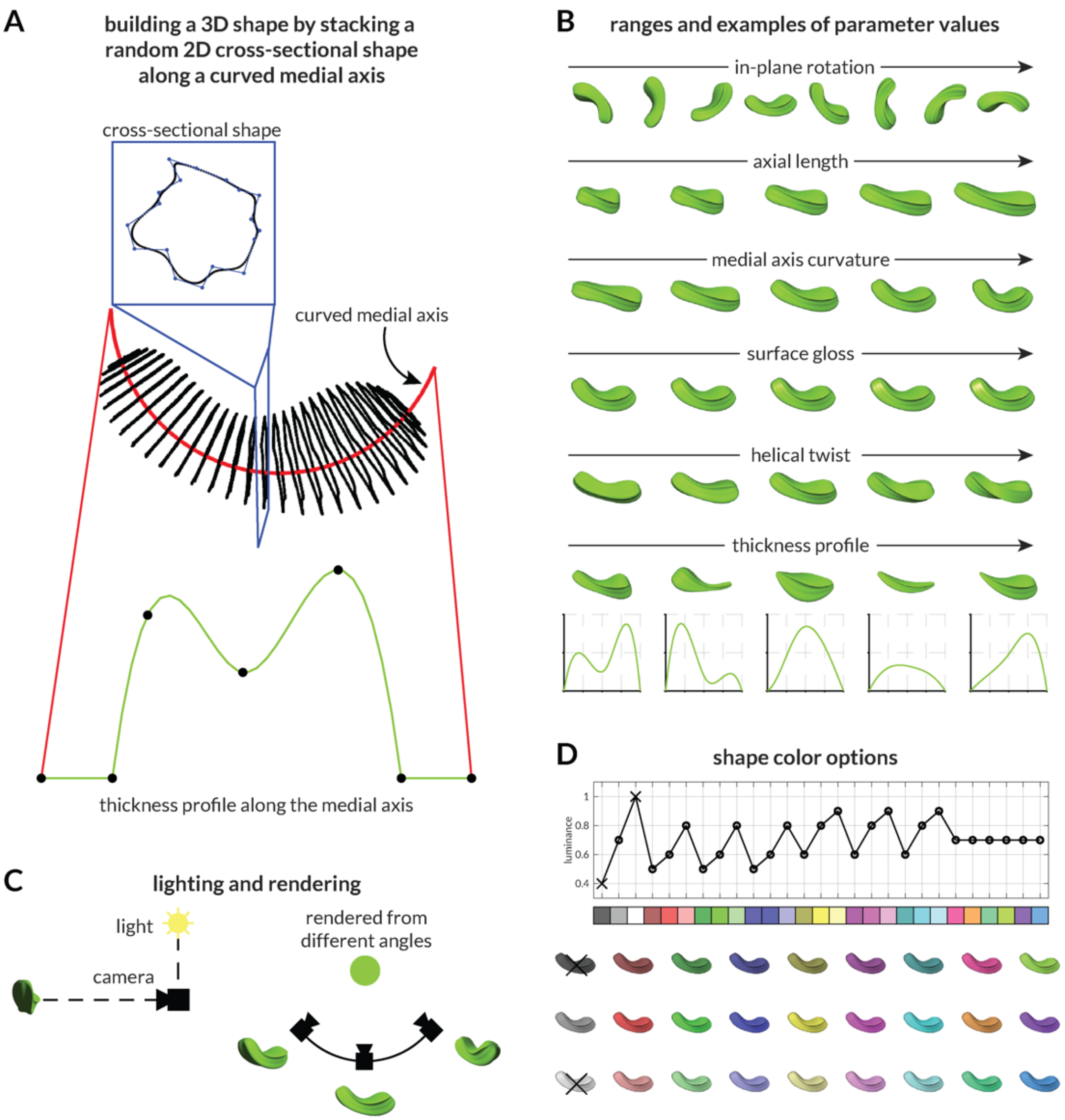


Figure S1: Stimulus generation details and parameters

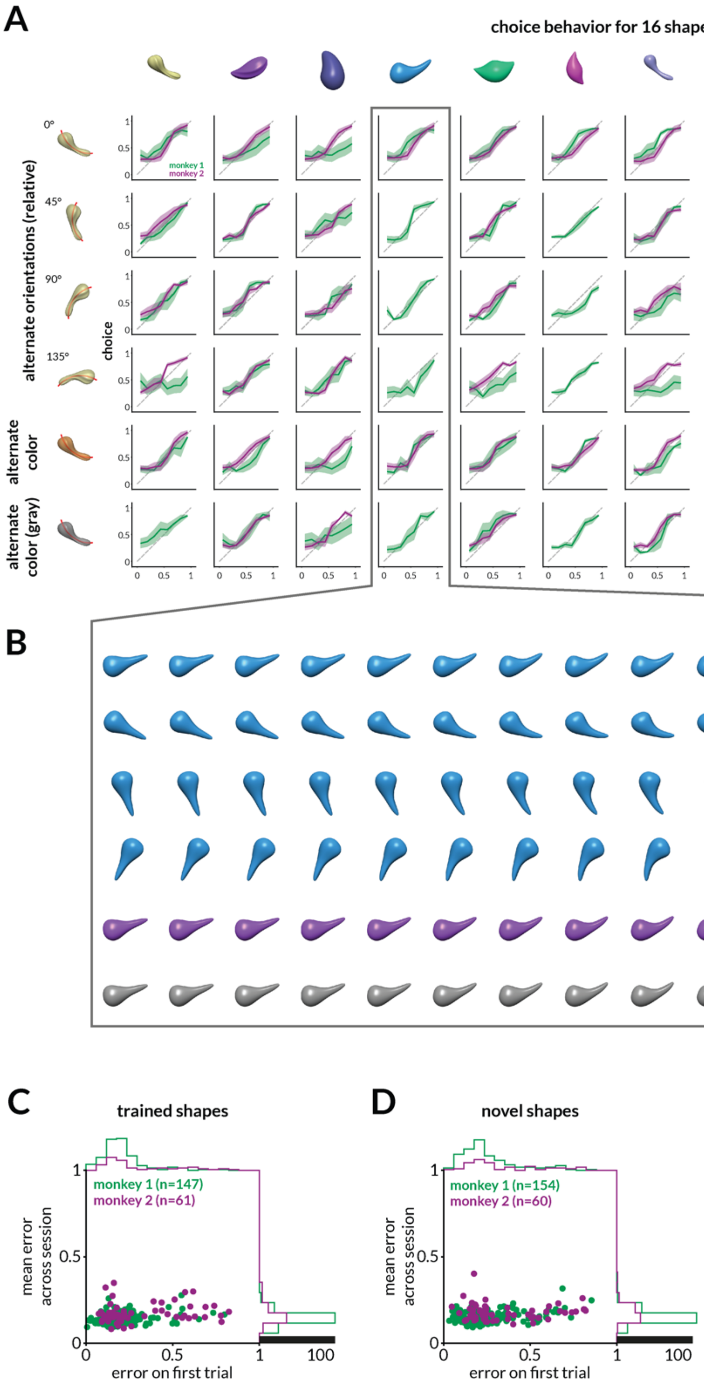
A: To build a random 3D stimulus shape, we first generate a cross-sectional 2D shape by randomizing a closed-loop cubic b-spline with arbitrary complexity (number of control points and randomization). We also generate a curved axis with a randomized length and curvature, a thickness profile along the axis (by randomizing a 5-point Bezier curve), and a helical twist profile along the axis. We then scale the cross-sectional shape by the thickness profile and stack it along the axis. Once the vertices are in place, we join them with edges to create a closed 3D object. We render the object with a randomized color, position, in-plane orientation, and surface specularity/gloss.

B: Example renders of the same object with changes in different features. Wherever possible, the ranges spanned the parameter space (orientation, color, curvature, twist, etc.) and were otherwise chosen manually (length, gloss, thickness profile).

C: The lighting and camera were held in the same positions across all images – the camera was directly in front of the object and the light was an omnidirectional source above the camera. On the right, for demonstration, the object is shown being rendered along three camera positions.

D: The shape colors were chosen from 25 random values which were generated by permuting three absolute R, G, and B values – 0.4, 0.7, 0.9. The swatches and average luminance values thus generated are shown at the top and the example stimulus rendered in those values are shown below.

Figure S2: Curvature estimation behavior across all tested shapes



A. Curvature estimation behavior shown as psychometric curves for the two monkeys (blue and orange lines, respectively) for all unique shapes tested (the medial axis of each shape, shown in red in the icons, was not presented to the monkeys and is shown for illustration). Each panel shows the comparison between the actual curvature and reported curvature in solid lines (in nine bins across the curvature range). Shading indicates SEM within each bin. For each shape tested, 10 or 20 curvature values are presented as shown in the icons in the exploded view (in B) for shape 4 as an example. Each column represents a unique shape. The first four rows represent orientation variations as shown in the icons on the left and in the exploded view. The last two rows similarly represent color variations. Both monkeys tended to underestimate high curvature values and overestimate low curvature values. This might be a result of the bounded nature of the curvature report and/or anisotropies in curvature representations across the range of curvatures tested.
B. An exploded view showing all stimuli tested for a single shape. In the text, we use the term ‘shape' to refer to a full set of curvatures with no other parameter changes. Across shapes, the orientation only, the color only, or the entire shape (including many other parameters) could change.

C. Comparison of absolute behavioral error on the first presentation of a shape (one of 10 shapes used during behavioral training) during a recording session and the mean absolute error across the whole session. For both monkeys, the distribution of errors (marginal histograms on the right and top) tightens across the session and settles at a median value of ~ 0.12.
D. Same as C but for shapes never seen before during recording or training. The distribution of errors for both monkeys is similar to the one seen in C for the first trial and across the session.

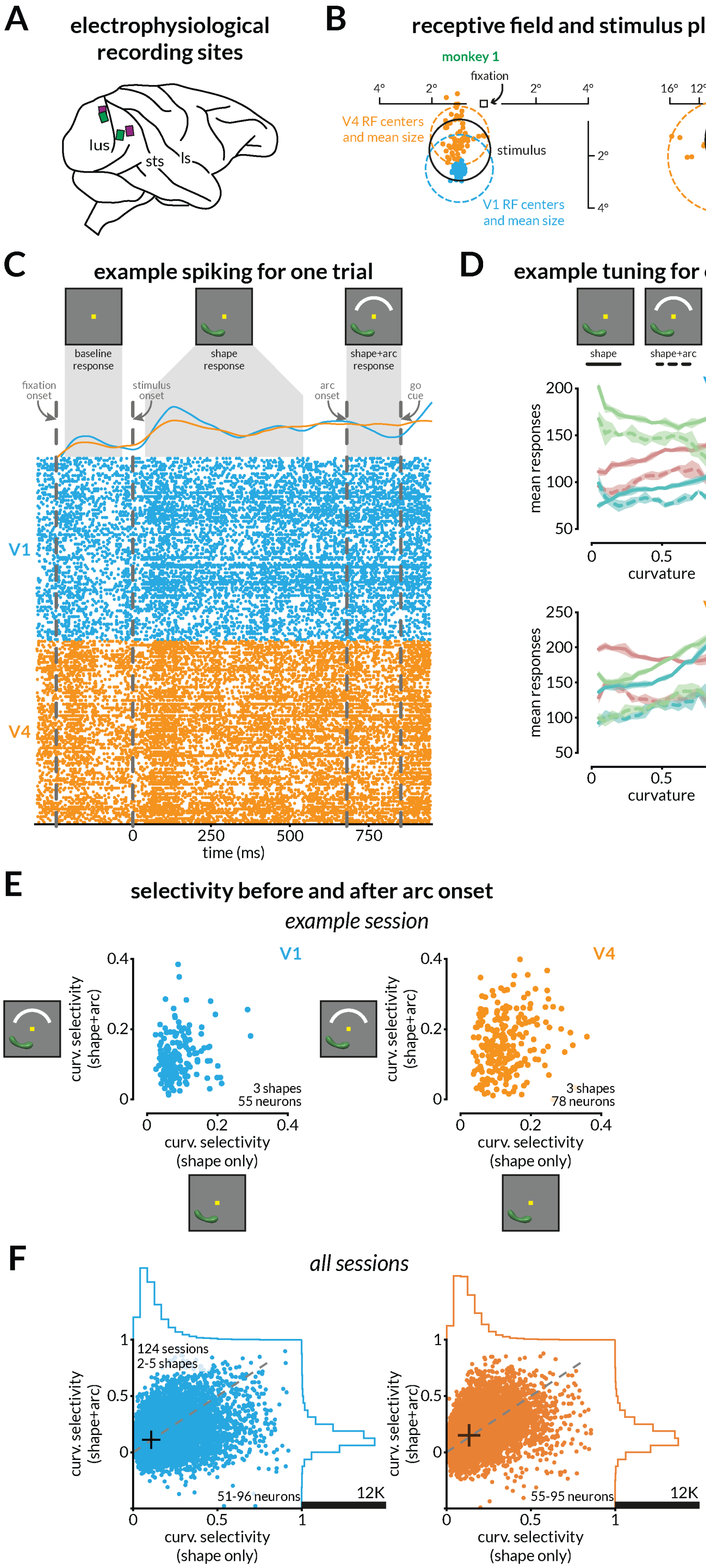


Figure S3: Recording locations, receptive fields, example responses to curved shapes, and neuron-wise curvature selectivity across shapes.

A. Locations of the multielectrode array implantations for the two monkeys.
B. Locations (dots) and mean size (dotted circle) of V1 and V4 receptive fields. The black circle indicates the stimulus location for an example session.
C. Raster plot and peri-stimulus time histogram for a single example trial. Each row corresponds to a V1 or V4 multiunit site. The baseline, stimulus, and arc periods (during which spikes are counted toward the response) are labeled in gray.
D. Trial-averaged responses for an example site in V1 and in V4 for three shapes as a function of curvature. Solid/dotted lines indicate responses during the stimulus/arc period, and shading indicates SEM.
E. Comparison of selectivity (defined as how much curvature modulates responses of a single shape for one unit; see Methods) during stimulus and arc periods for all shapes and each V1 and V4 site recorded in an example session.
F. The curvature selectivity shown for all shapes tested for all neurons across all sessions. The selectivity for curvature does not change systematically after the onset of the arc in either visual area. This was calculated for each shape (2-5; mean 3.55), for each recording site (51-96 for V1, 55-95 for V4), and for each of 124 sessions, totaling 34827 points for V1 and 35537 points for V4. Curvatures that elicited the maximum and minimum firing during the stimulus epoch were kept consistent while calculating selectivity during the arc period. Mean selectivity is shown in black + symbols; V1 selectivity changes from 0.13 to 0.142, and V4 selectivity changes from 0.134 to 0.156 after arc onset on average. Marginal distributions are plotted on the top and right of both panels. Gray line indicates the unity line.

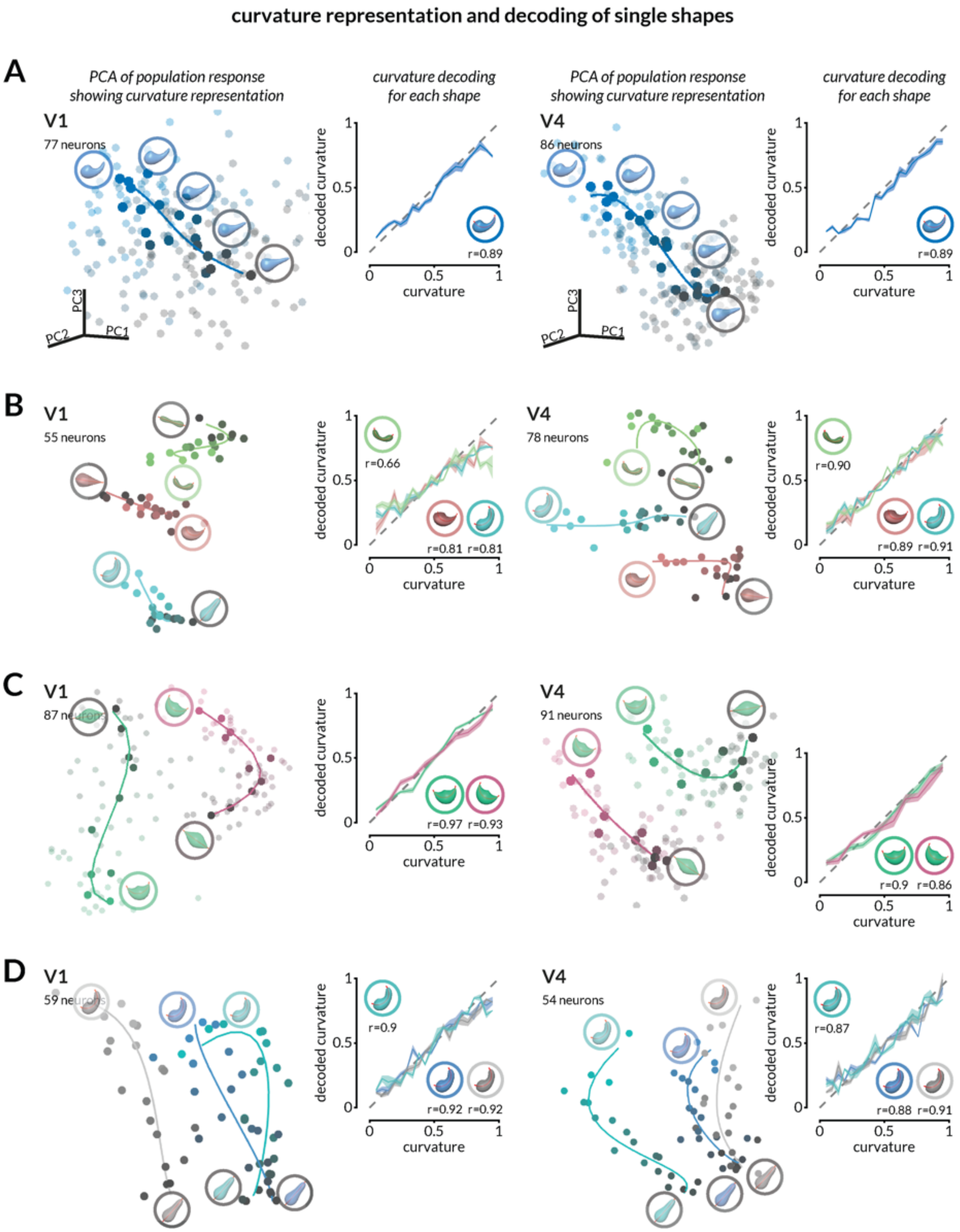


Figure S4: Curvature is robustly encoded in V1 and V4 population activity. (see Figure 1 for choice behavior during the same sessions)

A. (left) V1 stimulus-evoked population response from 77 neurons recorded during an experimental session projected onto the first three principal components of the population space where each dimension represents the response of one neuron. The principal component analysis (PCA) is for visualization rather than formal analysis. Each dot represents population responses during a single stimulus presentation, and the dot luminance (black-to-blue gradient) represents the curvature of the stimulus (also shown as superimposed icons for a subset of the presented curvatures). The dimmer dots represent all presentations and brighter dots are trial-averaged responses of stimuli with unique curvature. The solid curve represents a polynomial fit along the curvature representation. After training a linear decoder to predict curvature using neural responses, we binned trials by curvature and plotted their average decoded curvature and SEM (middle left); decoding performance was calculated as the correlation between the actual and decoded curvature (here, r=0.89). The same analysis is shown for a simultaneously recorded V4 population of 86 neurons on the right.

B. Same as A, for three unique shapes shown on randomly interleaved trials in the same session. The decoding performance for each shape is shown under the shape icons in the relevant plots.

C. Same as A, for the same shape presented in two different orientations.

D. Same as A, for the same shape presented in three different colors.

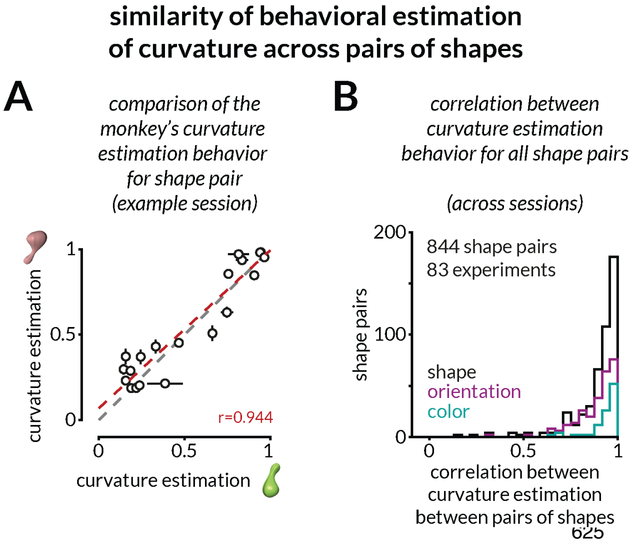


Figure S5: Quantifying the similarity in curvature estimation behavior across pairs of shapes.

A. While the representations of curvature in visual cortex responses are not typically aligned, the monkey's curvature estimation behavior is similar across shapes in the same session. Each dot is a unique curvature value, and horizontal and vertical error bars indicate SEM for the two shapes shown as icons. All behavioral biases are consistent across these shapes (correlation coefficient, r=0.944). Compare with curvature decoding in V1 and V4 for the same session in Figure 2.

B. Monkeys' curvature estimation behavior for any pair of shapes (unique, orientation-change, and color-change) is consistently correlated across sessions (mean 0.89, SEM 0.01, 844 shape pairs).

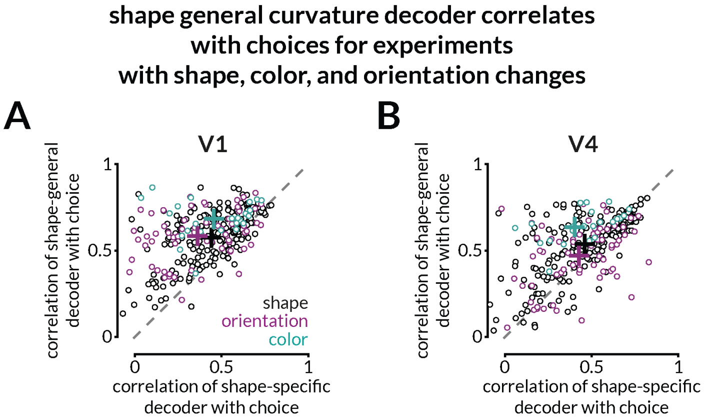


Figure S6: Shape-general decoders trained on responses to shapes differing in color only, orientation only, or overall shape are correlated with choices.

Curvature decoded using shape-general or shape-specific decoders (calculated in Figure 2D) split by sessions where only color, only orientation, or overall shape were varied across trials (same splits as in Figure 2F). Curvature decoded with shape-general decoders is more correlated with the monkey's choices than curvature decoded with shape-specific decoders for all training set variations (Wilcoxon signed rank test; p<0.001).

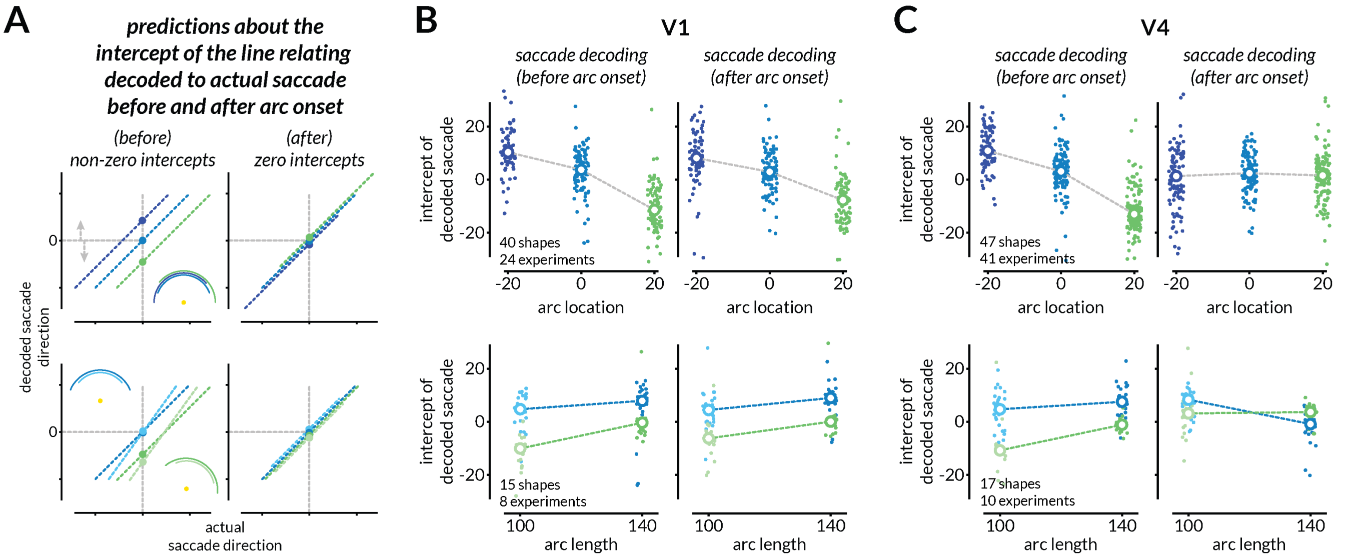


Figure S7: Analyzing intercepts of the linear relationship between the actual and decoded saccade reveals that V4, but not V1 responses reformat to encode the upcoming saccade.

A. We quantify the extent to which each population reflects the direction of the upcoming eye movement by evaluating the intercept of the line relating the decoded to the actual saccade. If the population does not reflect the direction of the upcoming eye movement, the intercepts will erroneously depend on the arc location (top left). In this scenario, rightward arc shifts will produce a negative intercept (green) and leftward shifts will produce a positive intercept (dark blue). When the arc length is varied, the intercepts will be zeros for centrally presented arcs (blues) but will be different negative values for rightwards shifted arcs (greens) (bottom left). If the population responses reflect the direction of the upcoming eye movement, all intercepts will be zero (top right and bottom right.).

B. Across experiments, V1 responses reflect the curvature judgment, not the direction of the upcoming eye movement. The intercept of the line relating the decoded and actual saccade direction depends on the arc location both before (left) and after the onset of the arc (right). Each point represents one combination of shape and arc condition.

C. Across experiments, V4 responses reflect the direction of the upcoming eye movement. Conventions as in B. Before the arc onset, the results are qualitatively similar to V1. Unlike in V1, the saccade direction can be decoded from V4 responses after the arc onset (as indicated by zero intercepts across arc conditions; right panels).

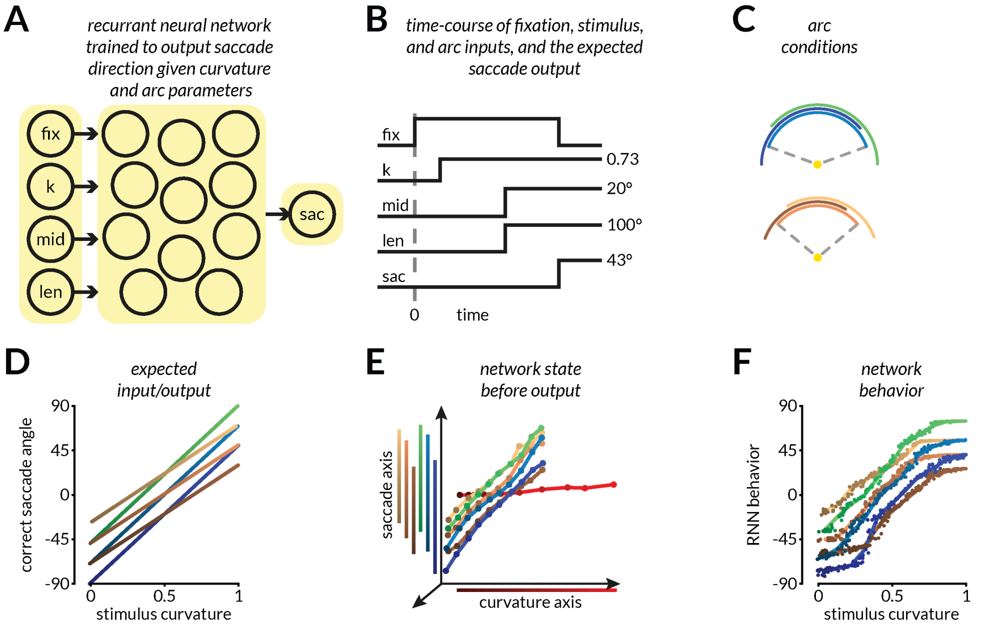


Figure S8: Recurrent neural networks (RNNs) can be trained to represent continuous variables and reformat those representations based on stimulus-extrinsic inputs in ways that could produce saccade-like outputs.

A. Schematic of RNN layout. Four time-varying signals that indicate the fixation spot ("fix"), stimulus curvature ("k"), and the two arc conditions – arc location ("mid") and arc length ("len") – are input to the network. The network is trained to produce saccade outputs ("sac") (shown in D) appropriate for each arc condition (shown in C).

B. The time course and values of input and expected output signals. Notably, the fixation, stimulus, and arc conditions are staggered to mimic the timing in the monkey task. The relative onsets are also randomized using the same timing distributions.

C. Legend for the various arc conditions tested.

D. The expected outputs for each arc condition as trained using backpropagation.

E. The network states (as visualized by a low dimensional embedding of the hidden layer activations using PCA and QR decomposition) before (red line) and after (colored lines) arc onset. We found the curvature and saccade axes by finding the linear decoding axes for curvature (before arc onset) and for saccades (after arc onset). Black-to-red dots each correspond to increasing curvatures of the stimulus and the projection of this curvature representation on the curvature axis is shown in the black-to-red line along the x-axis. The network activations after arc onset are projected onto the same dimensions and then used to decode the output/saccade. The projection of network states after arc onset (but before fixation offset) for each arc condition is shown in black-to-color lines along the y-axis. The layout of these projections recapitulates the expected output of the network (also shown in F). Importantly, the network does not lose the ability to read out stimulus curvature while encoding the output before the saccade.

F. The output of the trained network compared with the stimulus curvature for each arc condition. Compare with D and note intersecting points across arc conditions.

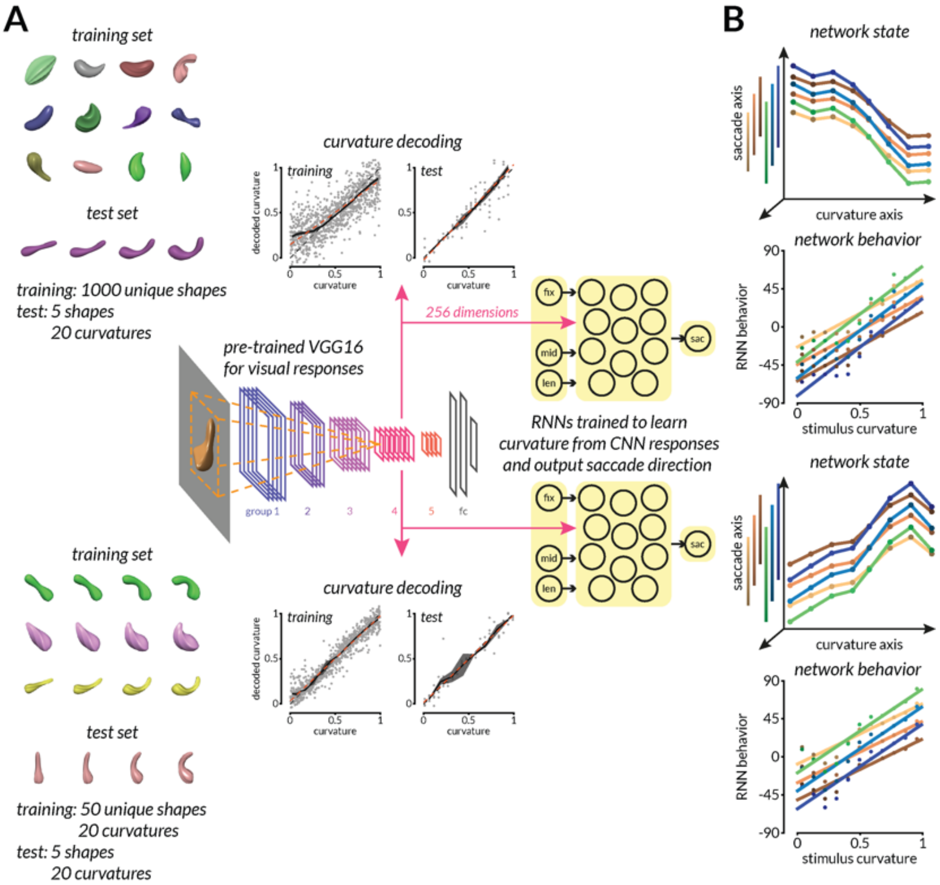


Figure S9: Convolutional Neural Network (CNN) activations can be used to extract visual features and train RNNs to flexibly align representations to behaviorally relevant readout axes.

A. We extract visual responses from an intermediate layer of a pre-trained convolutional neural network (VGG-16). We use these responses (picked because that layer has been shown to be representationally similar to V4) to train RNNs as in Figure S3. Visual stimuli are created using the same methods as described in Figure 1 and rendered in the central 91 pixels (RF of layer 4 units in VGG16) of the image, and visual responses are extracted. The dimensionality of activations is reduced to 256 to aid in RNN training and linear decoding. Two types of training image sets were created – 1000 unique shapes with randomly varying curvatures (top), or 50 unique shapes each with 20 curvature values (bottom). The testing stimuli in both conditions comprised of five unique shapes with 20 curvature values each. To show that curvature is represented in VGG16 layer 4, we trained a linear decoder using those activations in a leave-one-out cross-validated fashion using training images and tested it on the held-out test images. The same activations, reduced to 256-dimensions, were used to train RNNs instead of the one-dimensional stimulus input in Figure S8.

B. The trained RNN's hidden layer activations and output behavior are shown in the two pairs of panels. The top two are for the training set with 1000 unique shapes, bottom two are for the training set with 50 unique shapes. In both cases, the hidden layer represents the stimulus before arc onset, and the representation reformats to align with the saccade-like axis after arc onset. After this realignment, the curvature information is not lost but is still decodable, as in V4 (compare with Figure 3 and Figure S8).

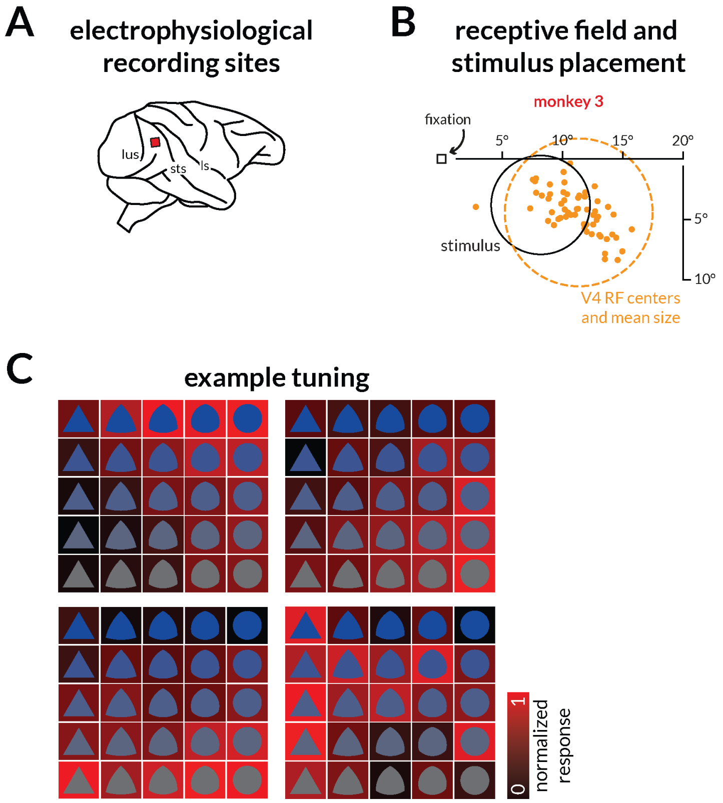


Figure S10: Recording locations, receptive fields, example responses to images used in two-feature 2AFC task

A. Locations of the multielectrode array implantations for the third monkey.
B. Locations (dots) and mean size (dotted circle) of V4 receptive fields. The array location yielded relatively eccentric RFs and the stimulus (black circle) was chosen to overlap with the receptive fields of a majority of the recorded units.
G. Normalized responses of four example multiunits recorded simultaneously. The black-red saturation of the background of the shape image indicates the normalized firing rate of that neuron. Across the array, the shape and color selectivity of the neurons varied considerably.

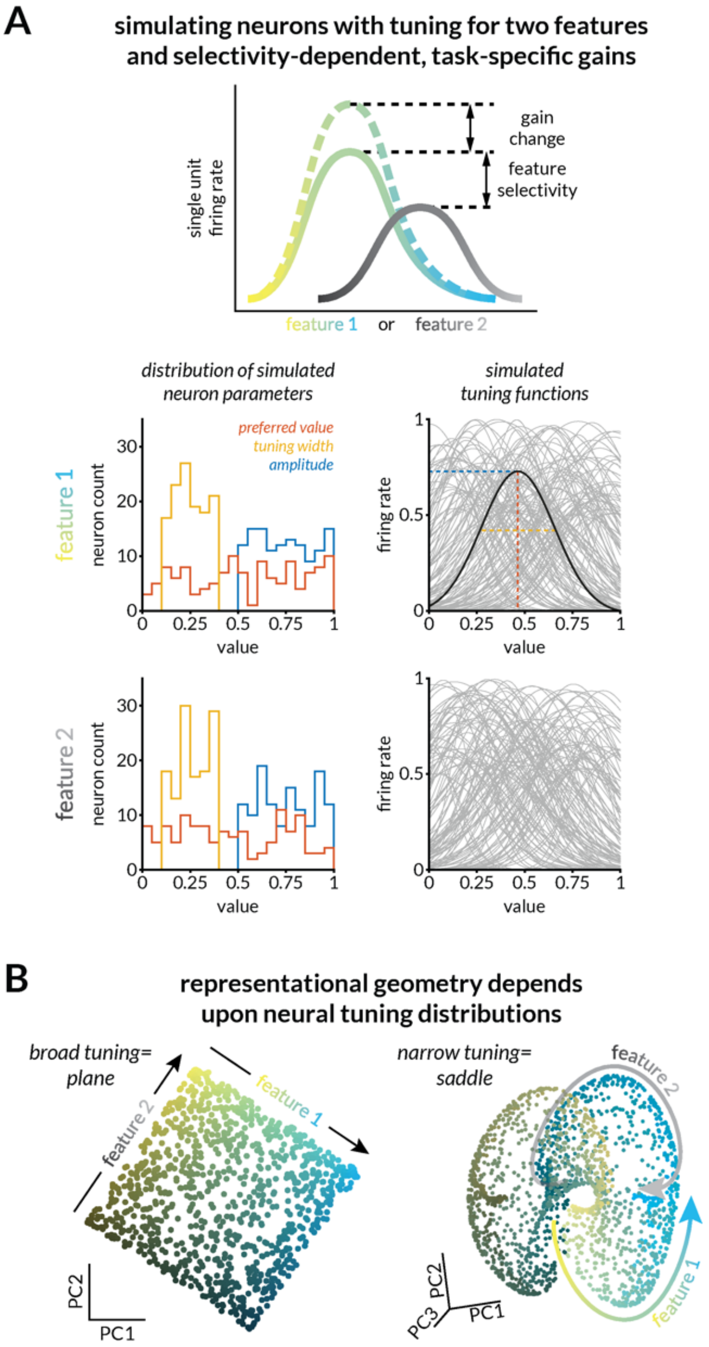


Figure S11: Simulations of tuning functions for two features can result in very different representational geometries.

A. Gaussian functions for two arbitrary parameters had identical distributions of Gaussian amplitude (0.5-1sp/s), tuning width/standard deviation (0.1-0.4), and preferred value/gaussian mean (0-1). The parameters were drawn from uniform distributions.

B. Two example population geometries created using different instantiations of the simulation. Here, varying the distribution of tuning widths gives rise to a planar representation or a saddle-shaped representation. In both cases, the bounded nature of the two features causes anisotropies at the edges of the representations causing linear decoders to underestimate higher feature values and overestimate lower values.

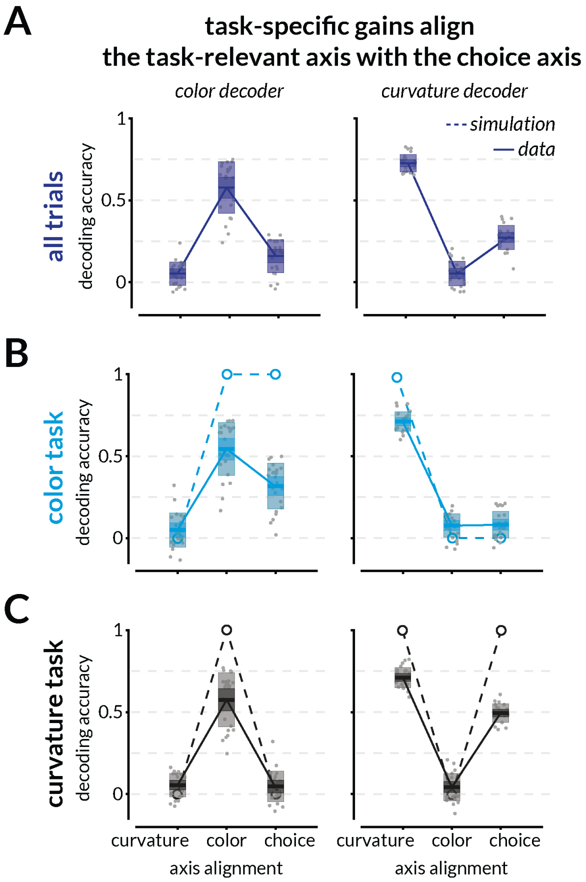


Figure S12: Breakdown of accuracies for the color and curvature decoders trained on either color task or shape task or all trials while decoding color, curvature, or choices

Decoding accuracies (correlation between actual and decoded values) for decoders trained on color (left) and curvature (right) while decoding curvature, color, and choice trained and tested across subsets of (**A**) all trials, (**B**) color task trials, and (**C**) curvature task trials. Points in B and C are identical to those plotted in the scatter diagram in Figure 4H. To illustrate that the shape and color axes are consistent across the two tasks, we decoded curvature, color, and choice across trials from both tasks together (shown in A). Curvature and color decoding accuracies were comparable to the accuracies of those trained using the individual task trials and choice decoding accuracy was approximately halfway between. The open circles/dashed lines depict model predictions.