Motor cortex activity across movement speeds is predicted by network-level strategies for generating muscle activity

Shreya Saxena[1,3,4,5,6,], Abigail A. Russo[2,3,*], John P. Cunningham[3,4,5,6], Mark M. Churchland[2,3,4,7,8]*

[1]Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA

[2]Department of Neuroscience, Columbia University, New York, NY 10027, USA

[3]Zuckerman Mind Brain Behavior Institute, Columbia University, New York, NY 10027, USA

[4]Grossman Center for the Statistics of Mind, Columbia University, New York, NY 10027, USA

[5]Center for Theoretical Neuroscience, Columbia University, New York, NY 10027, USA

[6]Department of Statistics, Columbia University, New York, NY 10027, USA

[7]Kavli Institute for Brain Science, Columbia University, New York, NY 10027, USA

[8]Lead Contact

** These two authors contributed equally*

Correspondence: Mark M. Churchland, mc3502@columbia.edu

Abstract

Learned movements can be skillfully performed at different paces. What neural strategies produce this flexibility? Can they be predicted and understood by network modeling? We trained monkeys to perform a cycling task at different speeds, and trained artificial recurrent networks to generate the empirical muscle-activity patterns. Network solutions reflected the principle that smooth well-behaved dynamics require low trajectory tangling, and yielded quantitative and qualitative predictions. To evaluate predictions, we recorded motor cortex population activity during the same task. Responses supported the hypothesis that the dominant neural signals reflect not muscle activity, but network-level strategies for generating muscle activity. Single-neuron responses were better accounted for by network activity than by muscle activity. Similarly, neural population trajectories shared their organization not with muscle trajectories, but with network solutions. Thus, cortical activity could be understood based on the need to generate muscle activity via dynamics that allow smooth, robust control over movement speed.

Introduction

We can often perform the same action at different speeds. This flexibility requires adjustments: not simply scaling the pace of muscle activity, but also altering its magnitude and temporal pattern. Given that movement speed can typically be adjusted continuously, it seems unlikely that the brain finds a separate solution for each speed. What might a unified solution look like, and how might it be reflected in the structure of neural population activity?

How this question should be approached interacts with a more basic issue: our understanding of the relationship between cortical activity and movement. This understanding has recently been in flux. Hypotheses that regard neural responses as representations of controlled variables (e.g., hand velocity or muscle activity) have been challenged by hypotheses that view neural responses as reflecting the needs of movement-generating dynamical systems [1–9]. This 'network-dynamics' perspective seeks to explain activity in terms of computational mechanisms that generate outgoing motor commands. The network dynamics perspective explains multiple response features that are difficult to account for from a representational perspective [3,4,6,10]. Yet the older representational perspective did convey a practical advantage: it readily supplied predictions in new situations. For example, when movement speed increases, a neuron that represents velocity should simply reflect the increased velocity. A neuron that represents muscle activity should simply reflect the new pattern of muscle activity. Can the dynamical perspective make similarly clear (and presumably different) predictions?

Computations performed by artificial -- and presumably real -- networks can often be described by flow-fields in state space. Because these flow-fields shape population trajectories, and because empirical population trajectories are readily plotted and analyzed in state-space, this affords a means of comparing data with predictions. Goal-driven networks -- i.e., networks trained to perform a computation intended to be analogous to that performed by the empirical network[11–13] -- are increasingly used to model computations requiring internal or external feedback [3,5,14–19]. Although such models typically lack detailed anatomical realism, they yield solutions that can be understood through reverse engineering, and produce population trajectories that can be compared directly with data. The continuous control of movement speed presents a situation where this approach might hope to generate particularly clear predictions. The range of possible solutions is likely constrained by the need for network dynamics to be 'well-behaved' in key ways. Network solutions should vary smoothly with speed. The alternative -- a distinct solution for each speed -- is incompatible with the fact that speed can be

continuously adjusted. Similarly, network solutions should gracefully interpolate between the speeds used during training. Finally, noise should not cause large errors or the generation of the wrong speed. All these forms of robustness are expected to benefit from smoothly varying underlying dynamics, resulting in population activity with very low 'trajectory tangling' -- i.e., where similar states are never associated with dissimilar derivatives [4].

We sought to determine whether these constraints allow network solutions to predict empirical responses. We trained monkeys to perform a cycling task at different speeds, and trained recurrent neural networks to generate the observed muscle-activity patterns, with speed instructed by a continuously varying input. All networks adopted a strategy with two prominent characteristics. First, the network trajectory for every speed resembled an ellipse in the dominant two dimensions, and was thus strikingly invariant to the shape of the muscle-trajectory output. Second, individual-speed trajectories were separated by a translation in a third dimension. We confirmed that this structure reflected stable input-dependent limit cycles, maintained low trajectory tangling, and reflected the need for continuity of solutions. Thus, structure that was common across networks reflects a natural solution to the basic computational problem of controlling speed. We compared network solutions with empirical population activity recorded from motor cortex. At both single-neuron and population levels, neural activity matched network activity more closely than it did muscle activity. Neural trajectories had the same within-speed and across-speed organization as network trajectories, and displayed the same low trajectory tangling. Thus, while the dominant structure of motor cortex activity does not resemble that of muscle activity, it closely matches solutions adopted by networks that employ dynamics to generate muscle activity. These findings demonstrate that the features of good network-level solutions can provide strong and successful predictions regarding empirical neural activity.

Results

Task, Behavior, Muscle Activity, and Single Neuron Responses

We trained two rhesus macaque monkeys to grasp a pedal with their hand and cycle for juice reward [4,20]. Cycling produced movement through a virtual landscape (**Figure 1a**). Landscape color indicated whether forward virtual movement required 'forward' cycling or 'backward' cycling, performed in alternating blocks. Monkeys cycled continuously to track a moving target,

Figure 1

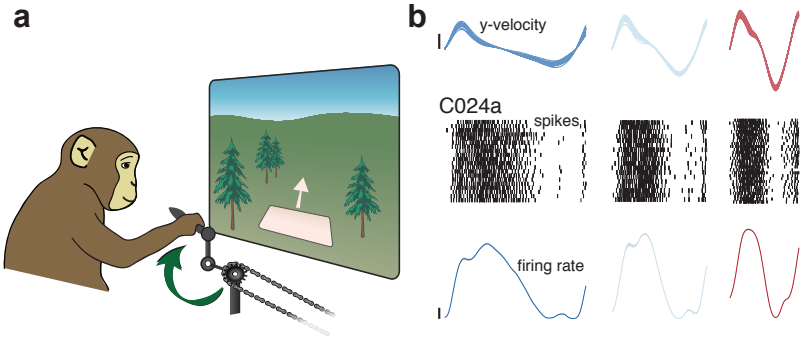

Figure 1. Illustration of behavior and neural activity. **a.** Schematic of the task, which required cycling at a continuum of different speeds for juice reward. Virtual environment color instructed cycling direction: green for forward (shown) or orange for backward. **b.** Behavior and neural activity for one example neuron (C024a). Data are for Monkey C, cycling forward. All data are plotted after temporal alignment within each bin. For visualization, plots show just over one cycle. Top: *blue / red* traces plot vertical hand velocity for every trial within that speed bin. These overlap heavily, forming an envelope spanning trial-to-trial variability. Rasters show spikes for ~25 trials per speed bin. Color denotes speed bin, with faster speeds in *red* and slower in *blue*. Bottom: *colored* traces plot the trial-averaged firing rate. Shaded envelopes for firing rates indicate the standard error of the mean (SEM). All vertical calibrations for neural activity are 10 spikes/s.

in contrast to [4] and [20] where targets were stationary. Target speed was constant within long (30 second) trials. There was also natural within-trial variation in cycling speed. Monkeys would often slow down modestly for a few cycles, fall behind the target, then speed up modestly for a few cycles. As a result, there was an essentially continuous range of cycling speeds within each session. To allow analysis, data were divided into individual cycles, beginning and ending with the pedal oriented straight down. Cycles were classified into eight speed bins, spaced in 0.2 Hz intervals. The eight bins were chosen to span the relevant range for each direction and monkey and are thus labeled by number (1-8, from ~0.8-2.1 Hz) rather than frequency. Cycles within a bin were scaled to have the same duration, but the values of key variables (e.g., velocity, firing rate) were computed before scaling and were never altered. Cycles where there was a large change in speed were not analyzed, ensuring that trials within a bin had nearly identical velocity profiles (**Figure 1b**).

Well-isolated single units (49 and 52 neurons for Monkeys C and D) were sequentially recorded from motor cortex, including sulcal and surface primary motor cortex and the immediately adjacent aspect of dorsal premotor cortex. Cycling evoked particularly strong responses. Nearly all neurons that could be isolated were modulating during the task. A clear, repeatable structure in spiking activity was apparent across single cycles (**Figure 1b**, raster plots). We computed the firing rate of each neuron by averaging across cycles within a speed bin (median of 84 and 80 trials, monkey C and D), which yielded a time-varying trial-averaged firing rate (**Figure 1b,** bottom) with a time-varying standard error of the mean (envelopes are narrow and thus barely visible).

Neural responses were heterogeneous (**Figure 2**). In some cases, response magnitude increased with cycling speed (**Figure 2b**, *top*) while in others it decreased (**Figure 2c**, *bottom*) or changed only modestly (**Figure 2b**, *bottom*). A given neuron could show different relationships with speed depending on cycling direction (**Figure 2a**). Response complexity was not due to noise; standard errors (flanking envelopes) were typically small. Across all neurons, 33 showed an increase in response magnitude with speed for both cycling directions, 24 showed a decrease for both, and 44 scaled oppositely. Responses could be mono-phasic or multiphasic depending on the neuron, cycling direction, and speed. Temporal response patterns were typically similar for neighboring speed bins, with changes mostly reflecting temporal scaling. Across the full range of speeds, it was common to observe more dramatic changes in temporal response pattern. For example, responses could be multiphasic at slow speeds but

Figure 2

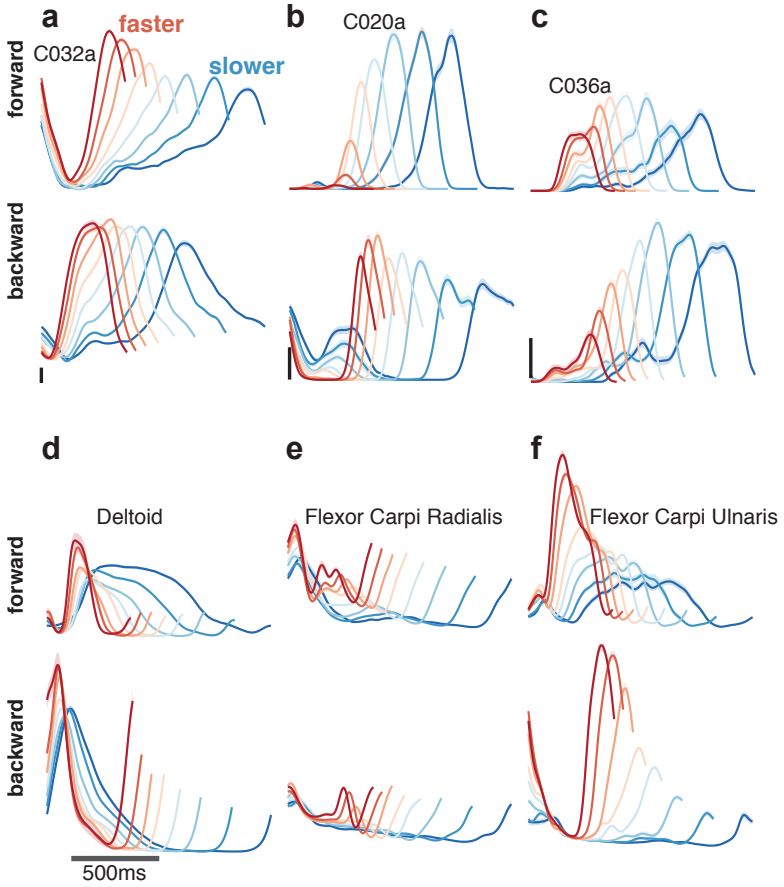

Figure 2. (a-c) Neural and **(d-f)** muscle activity during different speeds in the forwards and backwards conditions for monkey C. The activity in the fastest (slowest) speed bin is shown in *red* (*blue*), and the activity for the 8 speed bins are appropriately colored according to speed. Shaded envelopes indicate the SEM. All vertical calibrations for neural activity are 10 spikes/s.

monophasic at faster speeds (**Figure 2a**, *bottom*). The activity of major upper-arm muscles were also heterogeneous (15 and 33 recordings in monkeys C and D; **Figure 2d-f**). A key question is whether neural activity reflects structure beyond that present in the muscles -- structure that can be predicted by the character of network-level solutions.

Generating predictions using goal-driven modeling

We trained recurrent neural networks, via backpropagation through time, to produce the empirical patterns of muscle activity across speeds (**Figure 3a**). Target outputs were a projection of muscle activity onto the top 6 principal components (PCs). Because 6 PCs captured ~95% of the total variance in the muscle population (94.6 and 94.8% for monkey C and D), linear readouts of network activity yielded the activity of all recorded muscles with high fidelity. For simplicity, each network was trained to produce muscle activity for one cycling direction. We trained 25 networks for each monkey and direction (100 networks total). Of these, 96 successfully performed the task after 700,000 iterations (24 and 25 for Monkey C, forwards and backwards; 22 and 25 for Monkey D, forwards and backwards). Success was defined as <0.01 normalized mean-squared error between outputs and targets. Networks differed in their initial random weights, and thus showed non-identical final solutions.

No attempt was made at detailed realism: networks employed a single cell type, all-to-all connections, and no differentiation between feedback due to local versus long-range or re-afferent sources. This allowed us to focus on the form of the network-level solutions rather than their implementation. A natural first question regards the heterogenous single-neuron responses: does the network-based hypothesis provide improved explanatory power relative to the baseline hypothesis that neural activity is an upstream reflection of muscle activity? Regardless of whether this 'muscle-encoding' hypothesis is taken literally, it provides a stringent bar by which to judge success, because neural and muscle activity naturally share many features.

Under both hypotheses, single-neuron responses are predicted to reflect a set of underlying factors. E.g., $r_i(t) = \sum_j w_{ij}\, x_j(t)$, where $r_i(t)$ is the time-varying response of the ith neuron, w_{ij} are weights, and $x_j(t)$ are the factors. The two hypotheses can be compared by regressing empirical neural activity versus the relevant underlying factors: network factors or muscle

Figure 3

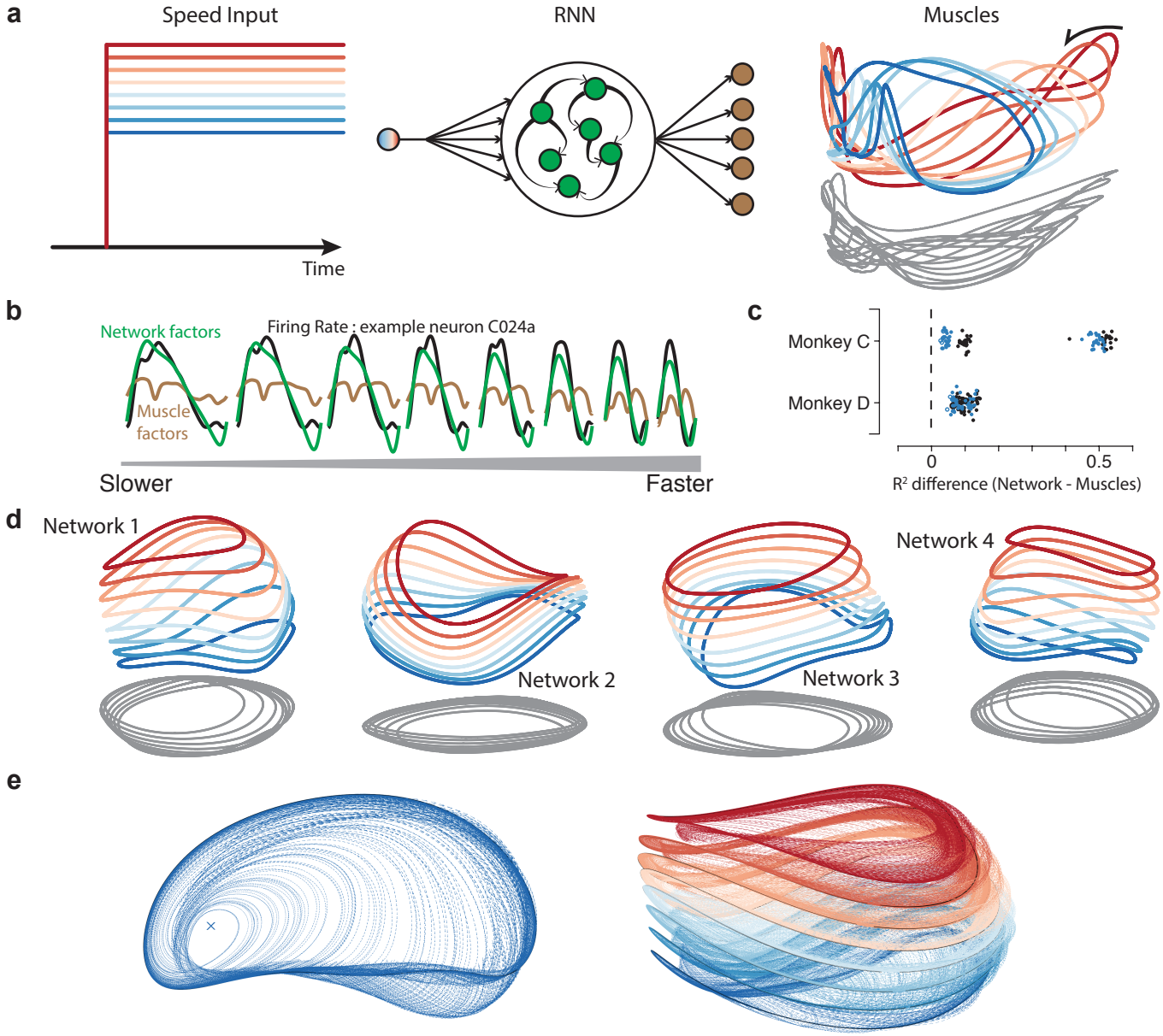

Figure 3. Goal-driven neural networks can accurately decode neural trajectories. **a.** Method for training neural networks. Recurrent neural networks (RNNs) were trained to receive a static speed input, and generate muscle data at the required speed. Different RNNs were trained for each cycling direction and each monkey. **b.** Firing rate (*black*) of an example neuron, as decoded with a linear decoder using the RNN trajectories (*green*, R^2=0.90), and the EMG trajectories (*brown*, R^2=0.02). The top 3 PCs of both the RNN and EMG trajectories were used for decoding. **c.** Difference in population R^2 (the mean R^2 across neurons, taking into account different response magnitudes) while decoding the firing rates of all neurons for both cycling directions using network factors versus muscle factors. Results are shown for decoding using two factors (*black circles*) and three factors (*blue circles*). Filled circles denote $p<0.05$ for a paired one-tailed t-test with n=8 speeds that the network factors provided better decoding than the muscle factors. **d.** Network trajectories for four example networks with different initializations projected onto global PCs 1, 2 and 3, i.e. the dominant dimensions for trajectories at all speeds. **e.** Exploring mechanisms of network function. (Left) The network was started at different initial conditions neighboring the network trajectory under a static input corresponding to the slow speed. All trajectories converge to the same limit cycle. (Right) The network was started at different initial conditions with varying inputs corresponding to the 8 different speeds. All trajectories converge to an input-dependent limit cycle.

factors. To identify those factors, we applied Principal Component Analysis (PCA) to the relevant population (muscle or network) and considered the top two or three PCs (three PCs explain >80% of the variance in both muscles and networks). Projections onto the muscle-derived PCs capture the dominant factors within the muscle population (i.e., the signals most strongly shared amongst muscles). Similarly, projections onto the network-derived PCs capture the dominant factors within the simulated network. One can then ask whether the empirical single-neuron responses are better accounted for by regressing against muscle or network factors. The coevolution of the top three muscle factors is shown in state-space in **Figure 3a** (*right*). Note that while muscle factors are target outputs of the networks, they need not be reflected in the top network-derived PCs. Indeed, network factors had a rather different structure (**Figure 3d** shows factors for four example networks). It is thus possible for a neuron's response to be fit poorly by the muscle factors but well by the network factors (**Figure 3b**).

Neural responses were overall better accounted for by network factors (**Figure 3c**). This improvement, relative to the muscle factors, was statistically significant for the vast majority of individual networks ($p<0.05$ for 90/96 networks, paired t-test with n=8 speeds, comparison using three PCs). This effect was clearest when considering the dominant signals (shown in **Figure 3c** for the first two PCs - *black circles* - and first three PCs - *blue circles*) but remained present when using more PCs. For example, when using six PCs, all networks provided an improvement over the muscles for monkey C, and all but one network provided an improvement for monkey D. These advantages were maintained if we assessed generalization R^2 (for left out conditions) to guard against overfitting.

Although network factors essentially always provided a better basis for explaining neural activity, the degree to which this was true varied across monkeys, cycling direction, and networks. For monkey C, the improvement in R^2 was highly dependent on cycling direction. Improvement ranged from 0.03-0.12 for backward cycling and from 0.41-0.55 for forward cycling. For monkey D, R^2 differences ranged from 0.05-0.14 across networks for both cycling directions. This range highlights that different networks found quantitatively different solutions. The same may be true of the biological networks in the two monkeys. The critical question is thus not whether factors match exactly, but whether the key features of network solutions are shared with the neural population response. To address this, we need to understand what network solutions look like and why they are successful.

Understanding network solutions

The major structural features shared by all solutions were visible in the top three PCs (**Figure 3d**). The network trajectory for each speed always resembled an ellipse (as revealed by the gray 'shadows'). An elliptical trajectory is not inevitable; the rhythmic task ensures a repeating trajectory but not an ellipse. Indeed, muscle factor trajectories (**Figure 3a**, *right*) were consistently less elliptical than network trajectories. For every network, ellipses provided a better fit ($p<0.05$) to the network trajectories (in the top two PCs) than to the corresponding muscle trajectories (paired, one-sided Wilcoxon signed rank test comparing R^2 values). Elliptical network trajectories formed stable limit cycles. We confirmed this by perturbing network activity, something only rarely possible in physiological circuits [1]. When perturbed off a cycle, the network trajectory returned to that cycle (**Figure 3e**, *left*). For all networks, limit cycles for different speeds were separated by a translation. The network's input determined which limit cycle was stable (**Figure 3e**, *right*).

Network trajectories were thus approximately elliptical for a given speed, and 'stacked' across speeds. The ability of elliptical network trajectories to generate non-elliptical muscle trajectories seems counter-intuitive but is expected [4]; 'muscle-encoding' structure is present in dimensions beyond the first three PCs. The observed stacking is a natural network strategy [15,21] for two reasons. First, unlike an arbitrary dynamical system, a recurrent network has no straightforward mechanism allowing an additive input to scale flow-field magnitude. Altering trajectory speed thus requires moving to a different region of state-space [22]. Second, the network's target output (muscle activity) does not simply speed up for faster speeds; it changes in magnitude and temporal pattern. It would thus be insufficient to traverse the same trajectory more rapidly, even if this were possible.

The stacked-elliptical network solution can also be understood at a more abstract level: it yields low trajectory tangling. High trajectory tangling is defined as two different moments with a similar location in state-space but very different derivatives. In networks where the present state determines the future state via internal recurrence or external feedback (i.e., that rely on strong dynamics) smooth noise-robust solutions require avoiding trajectory tangling[4]. Our networks had to rely on strong dynamics; their input lacked sufficient temporal structure to specify the output. Consistent with this, all networks adopted solutions where trajectory tangling was dramatically lower than muscle trajectory tangling (paired, one-tailed t-test; $p<10^{-10}$ for every network; for

example see **Supplementary Figure 1**). Elliptical limit cycles ensured low trajectory tangling within a given speed (a circle is the least-tangled rhythmic trajectory) and the separation between limit cycles ensured low tangling between speeds. This latter fact can be appreciated by considering the implications of traversing the same trajectory at different speeds: the same set of states would be associated with different derivatives.

A natural solution for maintaining low tangling is thus elliptical limit cycles separated by a translation. We saw this solution in all networks described above, and also when networks were training in the presence of noise and thus forced to adopt noise-robust solutions. Yet the consistency of the stacked-elliptical solution was due not only to low tangling, but also to the fact that it allowed continuity of solutions. For example, inputs 'between' the trained levels produced trajectories 'between' those shown in **Figure 3d**, which generated outputs at intermediate frequencies. Similarly, a ramping input produced trajectories with steadily increasing speed. Thus, networks could adjust their speed anywhere within the trained range. In principle, networks did not have to find this solution, but in practice training on eight speeds was sufficient to always produce it. In contrast, we did not observe the stacked structure in networks that received unique inputs for each speed. Those networks had distinct activity at each speed (and thus low tangling) but lacked a clear relationship between adjacent speeds (**Supplementary Figure 2**).

Thus, while there are in principle many possible network solutions, the combined need for low tangling and across-speed continuity yielded solutions that always had a stacked elliptical structure. That consistency allows clear predictions regarding the empirical neural activity. Qualitatively, neural trajectories should be dominated by elliptical structure within each speed, and separated by a translation between speeds. Quantitatively, this organization should yield trajectory tangling that is low within individual-speed trajectories, and remains low across speeds.

Neural trajectories are elliptical

We first consider the simplest prediction: the dominant structure of the neural population trajectory, for every speed, should be elliptical regardless of the structure of muscle activity. To identify the dominant shape of each trajectory, we applied PCA to the neural (or muscle) population response for that condition only (i.e. for only that speed and direction) and plot the

Figure 4

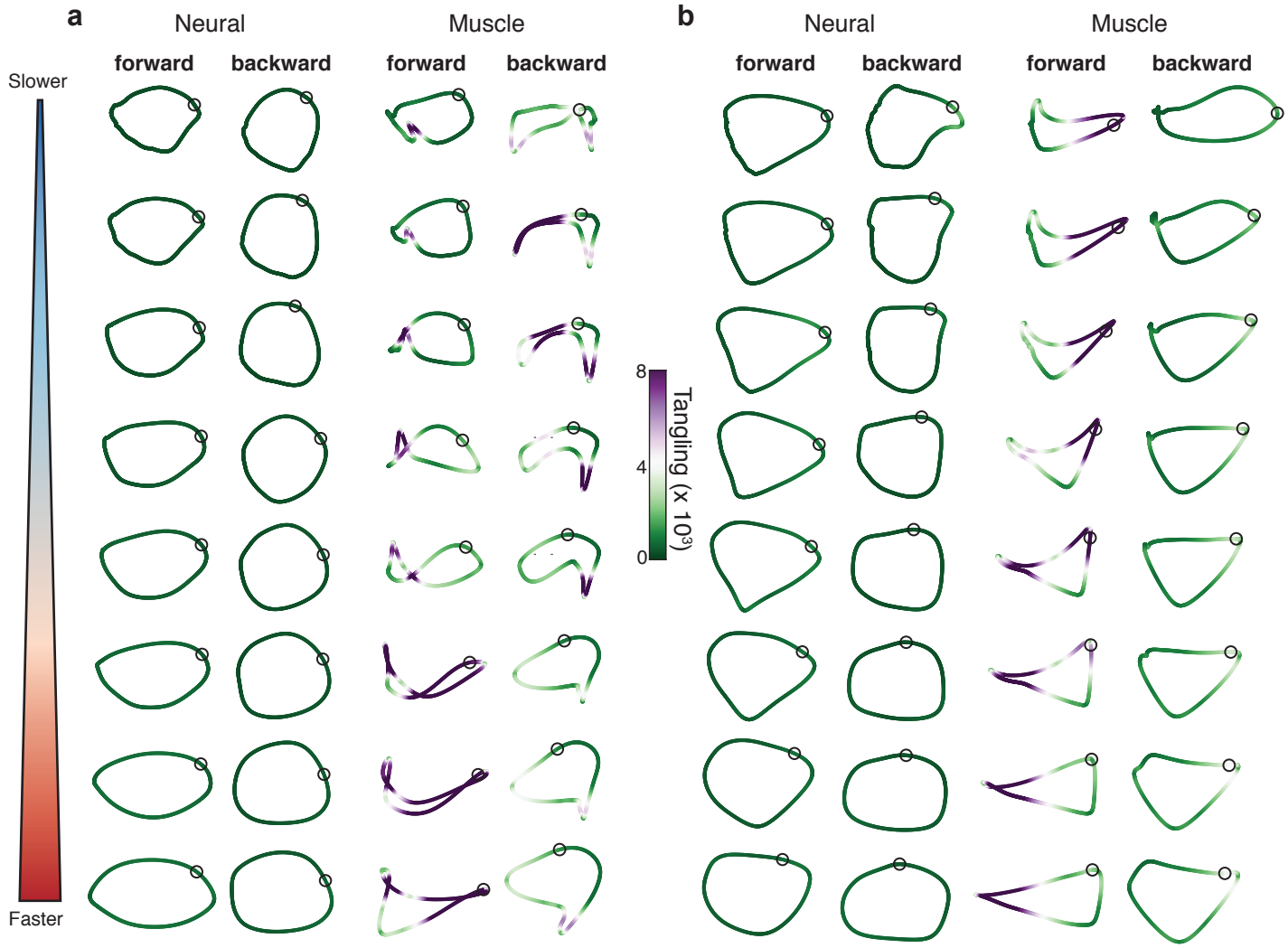

Figure 4. Visualization of population structure via principal component analysis (PCA) and tangling values. **a.** Neural and muscle responses of monkey C projected on the top two PCs, separately computed for each speed bin and each condition (forwards and backwards). Each data point represents the value of the neural responses at a different time point *t*, and all responses are shaded using the instantaneous tangling value *Q(t)*. The time at which the pedal is at the top-start position is indicated by a *black circle*. **b.** Same as **a** but for monkey D.

projection onto the first two PCs. This approach differs from the typical approach of applying PCA across multiple conditions (as was done in **Figure 3**) and allows us to focus on within-condition structure while ignoring (for now) across-condition structure. Neural trajectories were approximately elliptical in every case: all speeds, both cycling directions, and both monkeys (**Figure 4**). This was in sharp contrast with muscle trajectories, which exhibited a variety of shapes across speeds and directions, reflecting the different patterns of muscle activity necessary to generate the differing movements. Neural trajectories were more elliptical than muscle trajectories for every condition: all speeds and both directions. This difference was significant for both monkeys ($p<0.0001$; paired, one-sided Wilcoxon signed rank test comparing R^2 values).

The network perspective predicts that neural trajectory tangling should remain low even if muscle trajectory tangling becomes high. The trajectories in **Figure 4** are colored according to their tangling at every time point, computed as:

$$Q_s(t) = \max_{t'} \frac{\|\dot{x}_s(t) - \dot{x}_s(t')\|_2{}^2}{\|x_s(t) - x_s(t')\|_2{}^2 + \varepsilon}$$

(Equation 1)

where $Q_s(t)$ is the tangling at time t for speed s. $x_s(t)$ is the two-dimensional neural or muscle state. To focus on within-trajectory tangling, t' indexes across times within the same cycling direction and speed. Muscle tangling was often high (*purple*) due to tightly kinked regions, trajectory crossings, or segments travelling near one another in opposing directions. Such features were absent in the neural trajectories, due to their elliptical structure. Neural trajectory tangling was thus much lower than muscle trajectory tangling. This was true for every condition and both monkeys (paired, one-tailed t-test; $p<0.001$ for every comparison). This difference relates primarily to the dominant structure in the top two PCs, but was similarly present when more dimensions were considered (**Supplementary Figure 3**).

Neural trajectories are similar across speeds

The dominant structure of neural activity was surprisingly insensitive to the details of muscle activity; neural trajectories had a similar shape for every speed, even though muscle trajectories did not. We quantified trajectory-shape similarity among speeds within the same cycling direction (i.e., within a column in **Figure 4**). Because PCA was applied per speed, trajectories have different bases, yet one can still ask whether they share the same shape. To do so, we take two trajectories, $x_{ref}(t)$ and $x_i(t)$, apply the rigid rotation that maximizes their similarity, and measure that similarity. This procedure was performed for both neural trajectories (**Figure 5a**) and muscle trajectories (**Figure 5b**). Similarity was defined as $R^2(x_{ref}, x_i)$, the variance in $x_i(t)$ accounted for by $x_{ref}(t)$, with no scaling or offset allowed. We defined $R^2{}_k$ as the average $R^2(x_{ref}, x_i)$ over all situations where $x_{ref}(t)$ and $x_i(t)$ were separated by k speed bins.

$R^2{}_k$ is by definition unity for k = 0, and is expected to decline for larger values of k. That decline was modest for the neural trajectories and steeper for the muscle trajectories (**Figure 5c,d**). This confirms that neural trajectories had a consistent shape in the dominant two dimensions, while muscle-trajectory shapes were speed-specific. Yet if neural trajectories encode muscle-like commands (or other speed-specific features), there must be some dimensions where neural trajectories have different shapes for different speeds. This was indeed the case. We repeated the analysis using neural trajectories that were projections onto PCs three and four (top dashed gray curve) or five and six (bottom dashed gray curve). These aspects of the neural population response were less similar across speeds. For example, when considering neural PCs five and six, the decline in $R^2{}_k$ was comparable to that for the first two muscle PCs. This agrees with a fundamental feature of network solutions: signals that encode the output are 'pushed' into higher dimensions, allowing tangling to remain low in the dominant dimensions.

Neural population trajectories across speeds

Network solutions led to two predictions regarding the organization of neural activity across speeds. Qualitatively, trajectories should be separated by a translation across speeds. Quantitatively, that separation should be large and consistent enough to yield low overall tangling (a much stronger statement than the simple presence of average separation). We evaluated the first prediction by applying PCA jointly across all speeds for a given cycling direction. Unlike in the speed-specific projections above, a unified projection does not perfectly capture the elliptical trajectories, because they unfolded in somewhat speed-specific dimensions

Figure 5

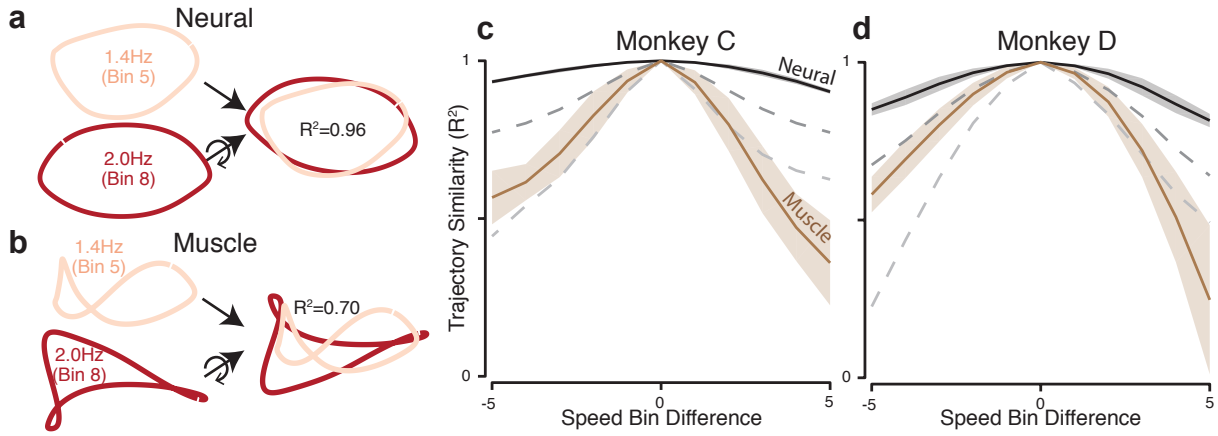

Figure 5. Path similarity between trajectories at different speeds in the forward condition. **a.** Method for calculating path similarity. The neural trajectory (projected onto PCs 1 and 2) for each speed bin was chosen as the reference trajectory, and the trajectory in neighboring speed bins was stretched to match the reference trajectory, then optimally rotated and flipped to maximize R^2. **b.** Same as **a** but for muscle data. **c.** Path similarity (R^2 values) as calculated in **a**, **b**, as a function of speed bin difference for neural data projected onto PCs 1 and 2 (*solid black line*) and muscle data projected onto PCs 1 and 2 (*solid brown line*). Shaded envelopes indicate the SEM across reference trajectories. *Dashed dark gray line* indicates the path similarity for neural data projected onto PCs 3 and 4, and *dashed light gray line* indicates the path similarity for neural data projected onto PCs 5 and 6. **d.** Same as **c** but for monkey D.

(as will be explored further below). Nevertheless, the first two PCs did a reasonable job of capturing the dominant elliptical structure, which was lacking in the muscle trajectories (**Figure 6a**). Employing three PCs revealed the predicted translation separating individual-speed neural trajectories (**Figure 6b**, *left*). Considered in three PCs, neural and muscle trajectories (**Figure 6b**, *right*) differed in much the same way as network and muscle trajectories. Neural trajectories were stacked ellipses, while muscle trajectories had a less clear organization.

There is no particular reason that the 'speed axis', the dimension that best captures stacking across speeds, should align with the third PC. For monkey C, the third PC naturally revealed stacking, but that feature was clearer (**Figure 6c**) when we identified the speed axis directly. For monkey D, stacking was not particularly prevalent in the third PC, but was quite apparent when the speed axis was found directly (**Supplementary Figure 4**). The speed axis was found by identifying the dimension where the mean neural state provided the best decode of mean angular velocity (Methods). In every plot, all dimensions share the same scale. Thus, the speed axis captured considerable variance: 4.94% and 4.37% (Monkey C and D). This was only modestly less than the variance captured by the third PC (5.34% and 8.57%) which is the maximum variance that could have been captured by the speed axis.

When considered in more than two dimensions, neural trajectories could depart from planar ellipses, sometimes tracing a pringle-like (hyperbolic paraboloid) shape. This was similarly observed for the networks (**Figure 3d**). In principle such departures could increase trajectory tangling between speeds -- for example they could cause neighboring trajectories to come close while moving in different directions. Yet for both neurons and networks, neighboring trajectories tended to depart from planar in similar ways. This could agree with a prediction derived from the networks: individual-trajectory shapes, combined with between-trajectory separation, should prevent across-speed tangling. To test this, we again employed **Equation 1** but had t' index across times in all other speeds (and not within the same speed). This complements the within-speed tangling analysis above. Tangling was computed in 12 dimensions as this captured most of the variance within and between conditions (results were extremely similar if more dimensions were employed).

Neural trajectories had consistently low across-speed tangling (**Figure 6c**, *left, green shading* indicates low tangling). In contrast, muscle trajectories often displayed high across-speed tangling (**Figure 6c,** *right, purple shading* indicates high tangling). This difference was

Figure 6

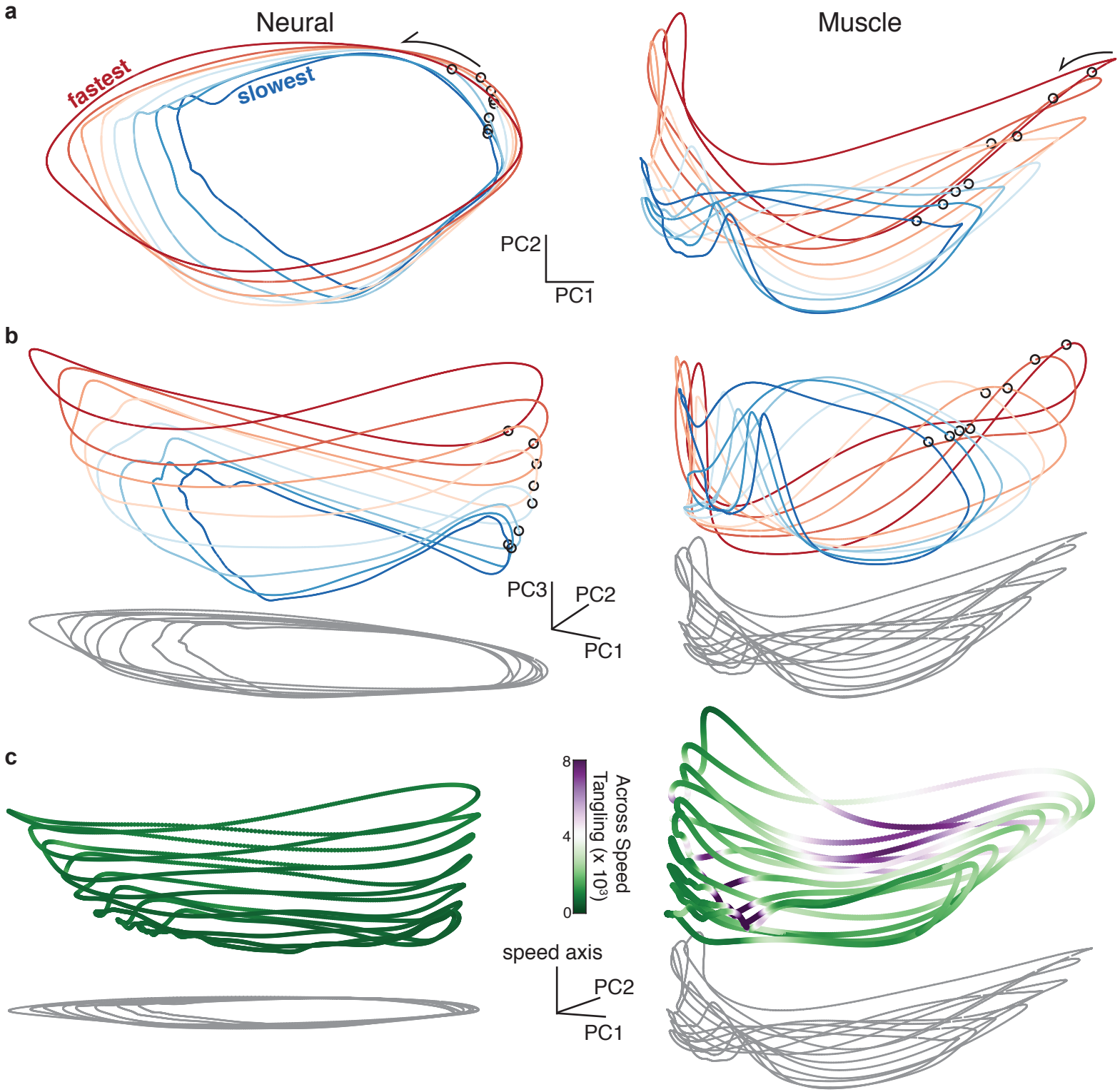

Figure 6. Visualization of neural and muscle population activity across all speeds for monkey C, forward condition. **a.** Neural and muscle activity projected onto global PCs 1 and 2, i.e. the dominant dimensions for trajectories at all speeds. **b.** Same as a but for PCs 1, 2, and 3. **c.** Same as **a** but for PCs 1, 2, and the 'speed' axis. The speed axis is defined as the direction in which the mean of each trajectory can best decode the speed bin. The trajectories in **c** are colored according to instantaneous tangling values in the global subspace, only considering the tangling across conditions.

pronounced for both monkey C (**Figure 6c**) and D (**Supplementary Figure 4c**). Note that muscle trajectories did display a 'speed axis' (more pronounced for monkey C and less so for monkey D) in the sense that the mean of each trajectory differed across speeds. This underscores the specificity of the network-derived prediction: trajectories be separated not just on average, but in a consistent manner that maintains low tangling.

Low global tangling depends upon inter-speed separation

As documented above, neural trajectory tangling was low when considering within-trajectory comparisons (**Figure 4**) and also when considering between-trajectory comparisons (**Figure 6c**). Muscle trajectory tangling was high in both cases. A reasonable inference is thus that global trajectory tangling -- employing all comparisons within and between trajectories -- should be much lower for neurons versus muscles. This was indeed the case (**Figure 7**, black dots). Each dot represents global trajectory tangling at one time for one speed, computed by allowing t' in **equation 1** to index across all times for all speeds and both directions. There were many moments when global trajectory tangling became high for the muscle populations. In contrast, global trajectory tangling never became high for the neural populations. The dramatic difference between neural and muscle trajectory tangling was not due to differences in intrinsic dimensionality; it was present regardless of whether we employed a matched number of dimensions for neural and muscle data (as in **Figure 7**), employed as many dimensions as needed to capture >95% of the variance (typically more for the neural data), or employed no dimensionality reduction and simply computed tangling in the full-dimensional space (which yielded nearly identical results but was time-consuming).

For the simulated networks, global tangling was only slightly higher than within-speed tangling, due to the separation of trajectories along the speed axis described above (**Supplementary Figure 5**). For the neural data, it was similarly true that tangling was low when computed purely within speed (*orange circles* and *bars*) and rose little when computed globally (*black circles* and *bars*). We used two methods to confirm that the lack of rise was due to the separation between speeds. First, we constructed an artificial population response by taking the empirical neural trajectory at a reference speed and constructing a trajectory for every other speed by appropriately scaling the derivative. We 'recorded' single-neuron responses (spikes generated via a Poisson process) and then analyzed the artificial population in the same way as the recorded neural data. This procedure yields trajectories that overlap but evolve at different

Figure 7

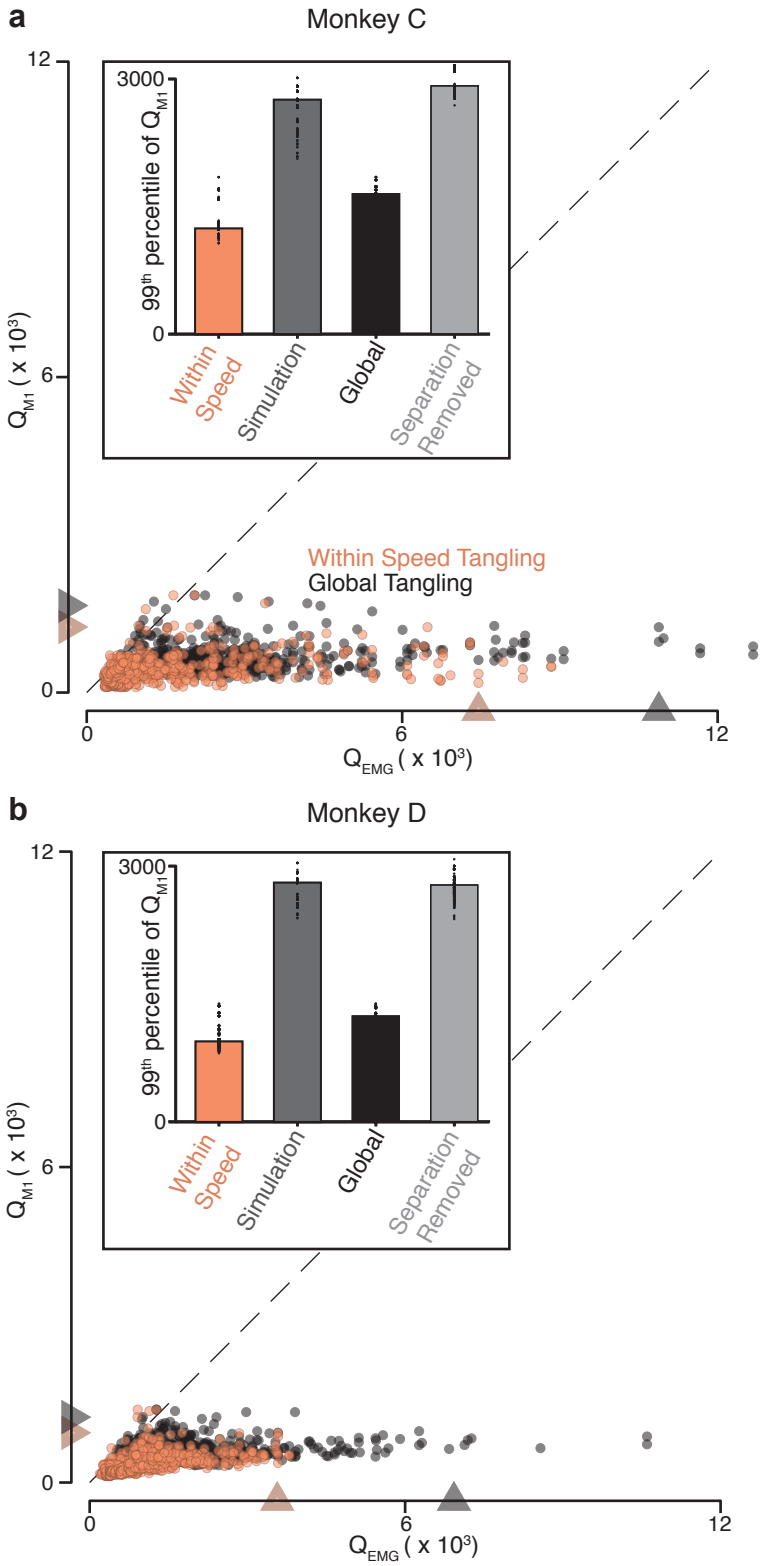

Figure 7. Trajectory tangling values in the global subspace. **a.** Scatterplot of neural-trajectory tangling versus muscle-trajectory tangling (monkey C) for 'within speed' tangling and 'global' tangling. Each point shows tangling for one moment (one time during one speed and one direction). Points are shown for all times during movement (sampled every 25 ms) for all sixteen conditions. *Black line* indicates unity slope. *Gray* / *orange triangles* indicate 99th percentile tangling. (inset) 99th percentile tangling values for (1) 'within speed' tangling, (2) simulated trajectories, generated by speeding up and slowing down a reference trajectory and adding biophysical noise, (3) 'global' tangling, and (4) 'separation removed' trajectories, which were optimally translated and rotated to maximize R^2 to a reference trajectory. *Black circles* show 99th percentile tangling values for bootstrapped data (Methods). **b.** Same as **a** but for Monkey D.

speeds. As expected, this organization led to an increase in tangling (*dark gray bars*). Second, we took the empirical population response and removed both the separation between trajectory means and any trajectory separation into different dimensions (a feature documented in more detail below). This similarly led to an increase in global tangling (*light gray bars*). Thus, for both the empirical and network trajectories, the dominant elliptical structure ensures low within-speed tangling, and the separation across trajectories maintains low global tangling.

Organization of trajectory separation

Low tangling requires not just overall separation (different trajectory means) but consistent separation at all points along neighboring trajectories. Can this separation be confirmed directly? To do so, we calculated the distance between trajectories as a function of θ, the phase of the physical movement. We chose a middle speed (speed bin 4) as a 'reference' trajectory, and computed the euclidean distance to all other trajectories at every phase (**Figure 8a**). To link with the analysis in **Figure 4c**, we also computed the distance between trajectory means. The distance between trajectory means behaved as anticipated: it grew monotonically as a function of the difference in speed from the reference trajectory (**Figure 8b**). This monotonicity was true for both cycling directions and both monkeys. A similar pattern was observed when distance was considered as a function of θ (**Figure 8c**). As speeds became more different from the reference speed, the corresponding trajectories became more distant. Most critically, there was never a value of θ where the distance became very small. Thus, no other trajectory, for any other speed, came 'close' to the reference trajectory. Moreover, distance generally grew monotonically with increasing differences in speed.

The above remained true regardless of the speed chosen for the reference trajectory. This is worth noting because, in **Figure 8c**, traces corresponding to speeds distant from the reference speed occasionally touch (e.g., the blue traces at bottom). This could imply either that those trajectories lack separation between them, or simply that they are equi-distant from the reference trajectory, but in somewhat different directions. These possibilities can be distinguished by repeating the analysis with different speeds as the reference trajectory. For example, if we chose speed-bin 2 (corresponding to the traces second from the bottom in **Figure 8c**) as the reference trajectory, it was well separated from the trajectory for speed-bin 1 and 3 at all phases (**Supplementary Figure 6a**). This organization resembles that observed for neural trajectories during a cognitive task where intervals must be internally timed [22], indicating

Figure 8

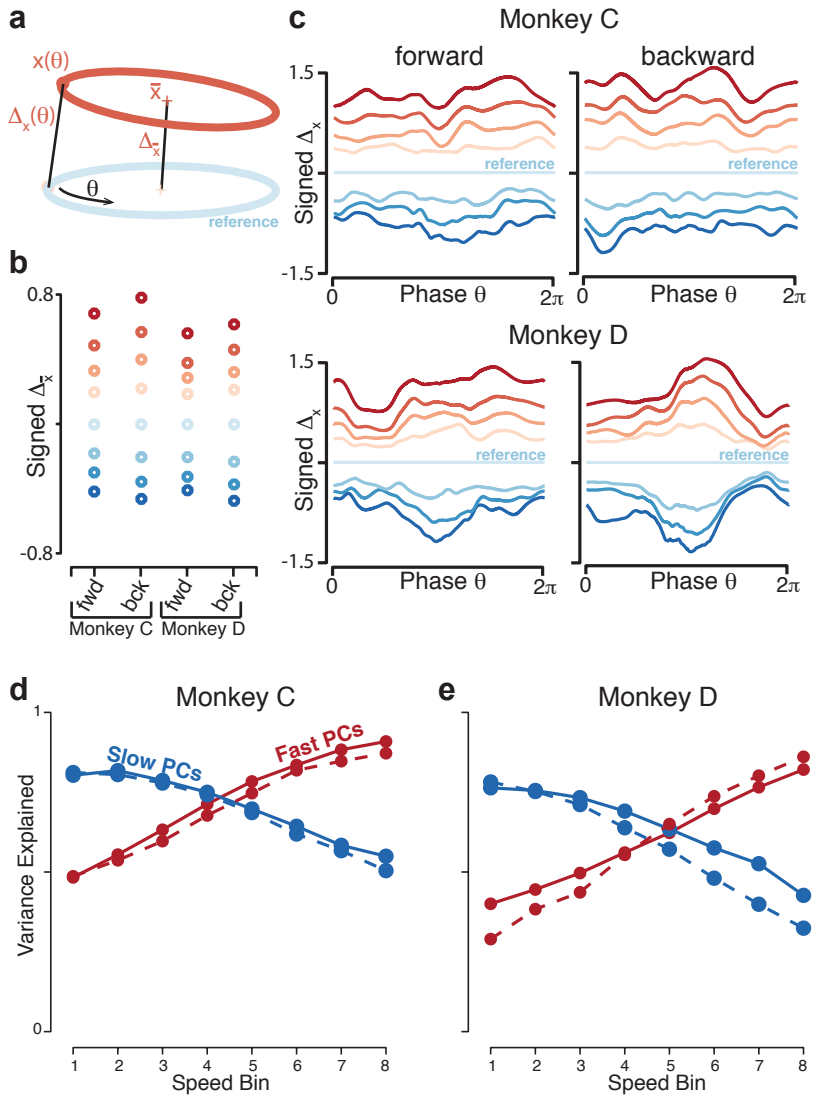

Figure 8. Distance and overlap between trajectories at different speeds. **a.** Method for calculating distance. The neural trajectory for speed bin 4 was chosen as the reference trajectory, and the trajectory in all other speed bins (*blue* indicates slow, *red* indicates fast) was stretched or compressed to match the reference trajectory. The Euclidean distance between the reference trajectory and each of the other trajectories was then calculated as a function of the phase of the reference trajectory. **b.** The distance between the mean of the trajectory in each speed and the mean of the reference trajectory, for each monkey and each cycling direction. **c.** The distance between the trajectories at each speed and the reference trajectory, at each phase in the cycle, for both monkeys and cycling directions. For results using different reference trajectories, see **Supplementary Figure 5**. Overlap between subspaces of neural trajectories at different conditions for **d.** Monkey C and **e.** Monkey D. Fraction of variance explained by using the principal components of the neural trajectories during the slow movement (*blue line*), to project the neural data during all speeds in the same direction. Similarly, the principal components during the fast movements (*red line*) were used to project the neural data in all conditions. The *solid line* indicates the forwards cycling direction, and the *dashed line* indicates the backward cycling direction. Here, the data is projected onto the top 2 PCs. For results with the data projected onto the top 10 PCs, see **Supplementary Figure 6**.

that the geometry of neural activity can reflect the essence of a computational problem (adjusting the rate at which something is accomplished) across different tasks (cognitive versus motor) and brain areas (medial prefrontal versus primary motor cortex).

Trajectories separate into different dimensions

All network solutions exhibited one additional feature not documented above: the elliptical individual-speed trajectories were not perfectly parallel to one another. The plane that best captured the elliptical trajectory for the slowest speed was moderately different from that for the fastest speed (**Supplementary Figure 7a,b**), providing an additional form of separation between trajectories. Do empirical neural trajectories exhibit this same feature? One observation above suggests so: elliptical trajectories were more cleanly observed when PCA was applied to each speed separately (**Figure 4**) than to all speeds together (**Figure 6a**). This suggests that the elliptical trajectories for different speeds are not perfectly parallel to one another (if they were, looking down on the stack should reveal each and every elliptical trajectory perfectly).

To test this prediction directly, we computed the top two PCs from the slowest-speed trajectory, which captured >75% of the variance for that trajectory. We fixed these PCs and asked how much variance they explained for the other trajectories. Variance explained declined monotonically as speed increased, reaching a minimum of ~40% for the fastest speed). The same effect was observed in reverse if we computed the two PCs from the fastest trajectory and then considered slower trajectories (**Figure 8d,e**). This effect held even if we considered the top ten PCs (**Supplementary Figure 7c,d**). Thus, the elliptical individual-speed trajectories unfolded in dimensions that overlapped but were not identical and became progressively more different for larger differences in cycling speed.

Discussion

Movement-speed control provides a test for the network-dynamics perspective: can it explain the dominant structure of neural responses? We found that it could, at both the single-neuron and population levels. The ability to account for single-neuron responses improved when regressing against network factors rather than muscle factors. This is notable because even

qualitatively accurate network solutions are not guaranteed to provide quantitatively useful factors, especially when competing against empirical muscle factors that naturally share features with the neural responses. Yet network factors essentially always provided some improvement, and frequently provided large improvements.

More deeply, network solutions anticipated the organization of neural trajectories. Networks and neurons employed a similar 'stacked-elliptical-trajectory' organization. That organization explained what would otherwise be a paradoxical finding: the neural trajectory for every speed was elliptical, independent of the muscle-trajectory shape. This disconnect would, from a traditional perspective, have argued against a tight relationship between neural and muscle activity. In fact the opposite is true: this property is expected of a network generating muscle-like commands.

Network solutions and neural activity were also similar at a more abstract level. An emerging approach is to consider geometric properties beyond the level of individual-task responses. Certain classes of computation require or imply certain geometric properties [4,16,20,23–25]. Of relevance to movement generation, low trajectory tangling is necessary for noise-robust dynamics. Consistent with this, network and neural activity displayed very low tangling relative to muscle trajectories. The level of geometry illuminates why neural activity likely employs stacked-elliptical structure. Such structure is a natural way of maintaining both low trajectory tangling and continuity of solutions across speeds. This perspective illuminates commonalities among seemingly disparate features. Trajectories were separated by both a translation and by occupying moderately different dimensions. Superficially, these features seem unrelated. The first produces firing-rate profiles with different offsets for different speeds, while the second yields speed-specific neuron-neuron correlations. Yet both serve to maintain low global tangling.

The desirability of low tangling holds across a broad range of situations [4]. Low tangling, and the stacked-elliptical trajectories that achieve it, are therefore expected across a broad range of models that use dynamics (created by internal and/or external feedback) to generate outputs across a continuous range of speeds. This feature allowed network simulations to make clear predictions; every network whose solution respected continuity employed the stacked-elliptical solution. Yet the generality of that solution also limits interpretation: the dominant features of the data are expected of good network solutions in general, and thus do not incisively discriminate amongst models within that class. This quandary is becoming common. For example,

quasi-rotational dynamics are a natural way of maintaining low tangling during reaching. Consistent with this, a range of dynamical models reproduce that empirical finding [5–7,15,17]. This highlights progress -- the dominant features of neural data are becoming easier to understand and even predict -- but also an emerging challenge: discriminating amongst models may require examining less-dominant features that are harder to visualize and quantify.

An implicit assumption of our analyses is that the observed trajectories are the product of the dynamics of a large network, rather than the handful of recorded neurons. Given this, it is impossible to say from recordings alone whether the hypothesized dynamics are primarily due to recurrent connections within cortex, or to a broader network spanning cortex, thalamus, basal ganglia, and cerebellum [26–28]. Such areas may provide key inputs and/or be critical recurrent partners in shaping dynamics. For example, in mouse, the cortical trajectory during reaching is interrupted by thalamic inhibition [29]. This is consistent with the fact that, in monkey, the largest feature of the cortical trajectory during reaching is a condition-invariant 'loop', hypothesized to reflect a movement-triggering input [30–32]. In addition to neural recurrence, the natural recurrence of sensory feedback will also shape dynamics.

Given the many influences on cortical activity, it is potentially surprising that motor cortex population trajectories exhibit low tangling, and agree with solutions found by generic network models that use only internal recurrence. Noise robustness requires the system as a whole to maintain low tangling, but this need not be true of motor cortex per se. It may be that, precisely because motor cortex activity is influenced by other key sources of recurrence and feedback, the overall state of the system is largely 'observable' (in the mathematical sense) cortically [3,33]. Put differently, as long as the impact of future feedback (local, long-range or sensory) is a predictable consequence of the present population state, population dynamics can be modeled via the dynamics of autonomous networks [34]. As an example, [3] trained networks where recurrence was purely local or also involved feedback loops with downstream sources. In all cases, the 'cortical' solution was similar: quasi-rotational dynamics with phase and amplitude set by preparatory activity. This underscores that good solutions are often fairly generic.

Conversely, when dynamics are not fully observable in a recorded population, trajectory tangling may be high and activity may be poorly accounted for by autonomous dynamics [33]. For example, this is true in primary somatosensory cortex during cycling [4] and (unsurprisingly) in primary visual cortex [4,33]. Tangling will also be high when unpredictable external inputs dominate the motor cortex response [34]. For example, autonomous dynamics provide a poorer fit to the

data during grasping [35]. The issue of observability may also explain why [36], studying cortical activity during locomotion at different speeds, observed stacked-elliptical structure that was qualitatively similar to that during cycling but with far less trajectory separation between trajectories. During locomotion, the ‘stacking’ axis captured <1% of the population variance, which is unlikely to provide enough separation to minimize tangling. If locomotion relies more on spinal pattern generation, and less on cortex, the full network solution might not be observable cortically.

Of course, even our simplified networks do not solve the problem of speed control entirely on their own; they rely on a speed-instructing input. The solution observed in motor cortex presumably also depends on instructive inputs, which could arise from any number of sources including frontal cortex and the basal ganglia [37] via motor thalamus. The assumption of inputs raises a subtle point. Tangling can become high when trajectories are primarily shaped by inputs. However, this is not expected for additive inputs that lack the temporal structure necessary to shape time-evolving trajectories. For example, in our networks, the static instructive input does not directly produce the elliptical trajectory, but rather alters where that trajectory is centered and thus the effective network dynamics.

The stacked-elliptical structure we observed is likely a common solution to the problem of adjusting the speed of a process or computation. In the present study, not only was the effect seen across both monkeys, it emerged universally in networks trained to produce the empirical patterns of muscle activity. A similar solution was seen across networks with a large variety of architectures trained to produce idealized periodic signals at different speeds [15]. A similar solution was even seen in a cognitive task, in both network and neural activity in the dorsomedial frontal cortex [22].

This underscores that network solutions often depend on the nature of a computational problem, rather than the particular domain. Conversely, seemingly similar problems within a domain may require different solutions. Even the simple goal of ‘speeding up movement’ may require very different computational solutions depending on movement type. For example, generating a faster reach does not involve a briefer pattern of muscle activity. It instead requires larger accelerations and decelerations. Consistent with this, motor cortex activity exhibits a higher-amplitude ~2Hz oscillatory pattern during fast reaches, not a higher-frequency pattern [6]. At the other end of the spectrum, sequences of multiple movements are generated similarly by motor cortex regardless of overall sequence speed, which is presumably controlled by upstream

areas [38]. Thus, the broader point is not that there is any single strategy for speed-control across all types of behaviors, but rather that the network-dynamics perspective can often predict and/or explain the strategies employed by biological networks.

References

1. Bruno, A. M., Frost, W. N. & Humphries, M. D. A spiral attractor network drives rhythmic locomotion. *Elife* **6**, (2017).
2. Pandarinath, C. *et al.* Inferring single-trial neural population dynamics using sequential auto-encoders. *Nat. Methods* **15**, 805–815 (2018).
3. Sussillo, D., Churchland, M. M., Kaufman, M. T. & Shenoy, K. V. A neural network that finds a naturalistic solution for the production of muscle activity. *Nat. Neurosci.* **18**, 1025–1033 (2015).
4. Russo, A. A. *et al.* Motor Cortex Embeds Muscle-like Commands in an Untangled Population Response. *Neuron* **97**, 953–966.e8 (2018).
5. Michaels, J. A., Schaffelhofer, S., Agudelo-Toro, A. & Scherberger, H. A neural network model of flexible grasp movement generation. *bioRxiv* 742189 (2019) doi:10.1101/742189.
6. Churchland, M. M. *et al.* Neural population dynamics during reaching. *Nature* **487**, 51–56 (2012).
7. Lillicrap, T. P. & Scott, S. H. Preference distributions of primary motor cortex neurons reflect control solutions optimized for limb biomechanics. *Neuron* **77**, 168–179 (2013).
8. Shenoy, K. V., Sahani, M. & Churchland, M. M. Cortical control of arm movements: a dynamical systems perspective. *Annu. Rev. Neurosci.* **36**, 337–359 (2013).
9. Hall, T. M., de Carvalho, F. & Jackson, A. A common structure underlies low-frequency cortical dynamics in movement, sleep, and sedation. *Neuron* **83**, 1185–1199 (2014).
10. Michaels, J. A., Dann, B. & Scherberger, H. Neural Population Dynamics during Reaching Are Better Explained by a Dynamical System than Representational Tuning. *PLoS Comput. Biol.* **12**, e1005175 (2016).
11. Zipser, D. & Andersen, R. A. A back-propagation programmed network that simulates

response properties of a subset of posterior parietal neurons. *Nature* **331**, 679–684 (1988).

12. Yamins, D. L. K. *et al.* Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proc. Natl. Acad. Sci. U. S. A.* **111**, 8619–8624 (2014).

13. Lindsay, G. W. & Miller, K. D. How biological attention mechanisms improve task performance in a large-scale visual system model. *Elife* **7**, (2018).

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

15. Maheswaranathan, N., Williams, A. H., Golub, M. D., Ganguli, S. & Sussillo, D. Universality and individuality in neural dynamics across large populations of recurrent networks. *arXiv [q-bio.NC]* (2019).

16. Sohn, H., Narain, D., Meirhaeghe, N. & Jazayeri, M. Bayesian Computation through Cortical Latent Dynamics. *Neuron* **103**, 934–947.e5 (2019).

17. Kao, T.-C., Sadabadi, M. S. & Hennequin, G. Anticipatory control of movement in a thalamo-cortical circuit model. *bioRxiv* 2020.02.02.931246 (2020) doi:10.1101/2020.02.02.931246.

18. Rajan, K., Harvey, C. D. & Tank, D. W. Recurrent Network Models of Sequence Generation and Memory. *Neuron* **90**, 128–142 (2016).

19. Perich, M. G. & Rajan, K. Rethinking brain-wide interactions through multi-region ‘network of networks’ models. *Curr. Opin. Neurobiol.* **65**, 146–151 (2020).

20. Russo, A. A. *et al.* Neural Trajectories in the Supplementary Motor Area and Motor Cortex Exhibit Distinct Geometries, Compatible with Different Classes of Computation. *Neuron* (2020) doi:10.1016/j.neuron.2020.05.020.

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

22. Remington, E. D., Narain, D., Hosseini, E. A. & Jazayeri, M. Flexible Sensorimotor Computations through Rapid Reconfiguration of Cortical Dynamics. *Neuron* **98**, 1005–1019.e5 (2018).

23. Bernardi, S., Benna, M. K., Rigotti, M., Munuera, J. & Fusi, S. The geometry of abstraction in hippocampus and prefrontal cortex. *bioRxiv* (2018).

24. DiCarlo, J. J., Zoccolan, D. & Rust, N. C. How does the brain solve visual object recognition? *Neuron* **73**, 415–434 (2012).

25. Hénaff, O. J., Goris, R. L. T. & Simoncelli, E. P. Perceptual straightening of natural videos. *Nat. Neurosci.* **22**, 984–991 (2019).

26. Middleton, F. A. & Strick, P. L. Basal ganglia and cerebellar loops: motor and cognitive circuits. *Brain Res. Brain Res. Rev.* **31**, 236–250 (2000).

27. Wang, J., Narain, D., Hosseini, E. A. & Jazayeri, M. Flexible timing by temporal scaling of cortical responses. *Nat. Neurosci.* **21**, 102–110 (2018).

28. Guo, Z. V. *et al.* Maintenance of persistent activity in a frontal thalamocortical loop. *Nature* **545**, 181–186 (2017).

29. Sauerbrei, B. A. *et al.* Cortical pattern generation during dexterous movement is input-driven. *Nature* **577**, 386–391 (2020).

30. Kaufman, M. T. *et al.* The Largest Response Component in the Motor Cortex Reflects Movement Timing but Not Movement Type. *eNeuro* **3**, (2016).

31. Zimnik, A. J., Lara, A. H. & Churchland, M. M. Perturbation of Macaque Supplementary Motor Area Produces Context-Independent Changes in the Probability of Movement Initiation. *J. Neurosci.* **39**, 3217–3233 (2019).

32. Lara, A. H., Elsayed, G. F., Zimnik, A. J., Cunningham, J. P. & Churchland, M. M. Conservation of preparatory neural events in monkey motor cortex regardless of how

movement is initiated. *Elife* **7**, (2018).

33. Seely, J. S. *et al.* Tensor Analysis Reveals Distinct Population Structure that Parallels the Different Computational Roles of Areas M1 and V1. *PLOS Computational Biology* vol. 12 e1005164 (2016).

34. Perich, M. G., Conti, S., Badi, M., Bogaard, A. & Barra, B. Motor cortical dynamics are shaped by multiple distinct subspaces during naturalistic behavior. *BioRxiv* (2020).

35. Suresh, A. K., Goodman, J. M., Okorokova, E. V. & Kaufman, M. T. Neural population dynamics in motor cortex are different for reach and grasp. *bioRxiv* (2019).

36. Foster, J. D. *et al.* A freely-moving monkey treadmill model. *J. Neural Eng.* **11**, 046020 (2014).

37. Yttri, E. A. & Dudman, J. T. Opponent and bidirectional control of movement velocity in the basal ganglia. *Nature* **533**, 402–406 (2016).

38. Zimnik, A. J. & Churchland, M. M. Independent generation of sequence elements by motor cortex. *Nature Neuroscience* (in press).

39. Gal, O. fit_ellipse. (2020).

40. Kabsch, W. A solution for the best rotation to relate two sets of vectors. *Acta Crystallogr. A* **32**, 922–923 (1976).

41. Ogata, Y. On Lewis' simulation method for point processes. *IEEE Trans. Inf. Theory* (1981).

Methods

Experimental Apparatus

Monkeys were trained to perform a cycling task [4]. The two primate subjects were adult male rhesus macaques (monkeys C and D). The animal protocols were approved by the Columbia University Institutional Animal Care and Use Committee. Experiments were controlled and data collected using the Speedgoat Real-time Target Machine. During experiments, monkeys sat in a customized chair with the head restrained via a surgical implant. Stimuli were displayed on a

monitor in front of the monkey. A tube dispensed juice rewards. The left arm was loosely restrained using a tube and a cloth sling. With their right arm, monkeys manipulated a pedal-like device. The device consisted of a cylindrical rotating grip (the pedal), attached to a crank-arm, which rotated upon a main axle. That axle was connected to a motor and a rotary encoder that reported angular position with 1/8000 cycle precision. In real time, information about angular position and its derivatives was used to provide virtual mass and viscosity, with the desired forces delivered by the motor. The delay between encoder measurement and force production was 1 ms.

Horizontal and vertical hand position were computed based on angular position and the length of the crank-arm (64 mm). To minimize extraneous movement of the wrist, the right wrist rested in a brace attached to the hand pedal. The motion of the pedal was thus almost entirely driven by the shoulder and elbow, with the wrist moving only slightly to maintain a comfortable posture. Wrist movements were monitored via two reflective spheres attached to the brace, which were tracked optically (Polaris system; Northern Digital, Waterloo, Ontario, Canada) and used to calculate wrist angle. The small wrist movements were highly stereotyped across cycles. Visual monitoring (via infrared camera) confirmed the same was true of the arm as a whole (e.g., the lateral position of the elbow was quite stereotyped across revolutions). Eye position and pupil dilation were monitored but are not analyzed here.

Task

The monitor displayed a virtual landscape, generated by the Unity engine (Unity Technologies, San Francisco). Surface texture and landmarks to each side provided visual cues regarding movement through the landscape. A salient visual cue (landscape color) indicated whether pedaling must be ‘forward’ (the hand moved away from the body at the top of the cycle) or ‘backward’ (the hand moved toward from the body at the top of the cycle) to produce forward progress in the virtual world. Trials were blocked into forward and backward pedaling. Movement was along a linear ‘track’. One rotation of the pedal produced one arbitrary unit of movement. The target on the landscape was continuously moving at a fixed speed, and reward was delivered every 550 ms so long as the monkey’s virtual position was close to the target. The recordings were then divided into cycles based on the stereotypy of the x and y positions.

The cycles were divided into eight speed bins for each of the two cycling directions, leading to a total of sixteen different conditions.

Neural recordings during task

After initial training, we performed a sterile surgery during which monkeys were implanted with a head restraint. Cylinders (Crist Instruments, Hagerstown, MD) were centered over the border between caudal PMd and primary motor cortex, located according to a previous magnetic resonance imaging scan. Cylinders were placed surface normal to the cortex. The skull within the cylinder was left intact and covered with a thin layer of dental acrylic. Neural recordings were made using conventional single electrodes (Frederick Haer Company, Bowdoinham, ME) driven by a hydraulic microdrive (David Kopf Instruments, Tujunga, CA). Electrodes were introduced through small (3.5 mm diameter) burr holes drilled by hand through the acrylic and skull, under ketamine / xylazine anesthesia. Sequential recording with conventional electrodes (as opposed to simultaneous recording with an array) allowed us to acquire recordings from a broader range of sites, including sulcal sites inaccessible to many array techniques. Recording locations were guided via microstimulation, light touch, and muscle palpation protocols to confirm the trademark properties of each region. Recordings were made from primary motor cortex (both surface and sulcal) and the adjacent (caudal) aspect of dorsal premotor cortex. For all analyses, these recordings are analyzed together as a single motor cortex population. Motor cortex recordings were restricted to regions where microstimulation elicited responses in shoulder, upper arm, chest and forearm.

Neural signals were amplified, filtered, and manually sorted using Blackrock Microsystems (Digital Hub and 128-channel Neural Signal Processor). A total of 126 isolations were made in monkeys C and D. Nearly all neurons that could be isolated in motor cortex were responsive during cycling. A number of isolations (25) were discarded due to low signal-to-noise ratios or insufficient trial counts. No further selection criteria were applied. On each trial, the spikes of the recorded neuron were filtered with a Gaussian (20ms SD) to produce an estimate of firing rate versus time. These were then averaged across trials as described below.

EMG recordings

Intra-muscular EMG was recorded from the major muscles of the arm, shoulder, and chest using percutaneous pairs of hook-wire electrodes (30mm x 27 gauge, Natus Neurology)

inserted ~1 cm into the belly of the muscle for the duration of single recording sessions. Electrode voltages were amplified, bandpass filtered (10-500 Hz) and digitized at 1000 Hz. To ensure that recordings were of high quality, signals were visualized on an oscilloscope throughout the duration of the recording session. Recordings were aborted if they contained significant movement artifacts or weak signal. That muscle was then re-recorded later. Offline, EMG records were high-pass filtered at 40 Hz and rectified. Finally, EMG records were smoothed with a Gaussian (20ms standard deviation, same as neural data) and trial averaged (see below). Recordings were made from the following muscles: the three heads of the *deltoid*, the two heads of the *biceps brachii*, the three heads of the *triceps brachii*, *trapezius*, *latissimus dorsi*, *pectoralis*, *brachioradialis*, *extensor carpi ulnaris*, *extensor carpi radialis*, *flexor carpi ulnaris*, *flexor carpi radialis*, and *pronator*. Recordings were made from 1-8 muscles at a time, on separate days from neural recordings. We often made multiple recordings for a given muscle, especially those that we have previously noted can display responses that vary with recording location (e.g., the *deltoid*).

Trial alignment and averaging

The average firing rate was computed across trials with nearly identical behavior. This was achieved by 1) training to a high level of stereotyped behavior, 2) discarding rare aberrant trials, and 3) 'adaptive alignment' of individual trials prior to averaging in each speed bin. First, trials were aligned so that vertical pedal orientation occurred at the same moment for every trial. The trials were then sorted into 1 of 8 total speed bins per cycling direction. Individual trials were then scaled so that all trials in a speed bin had the same duration (set to be the median duration across trials). Because monkeys usually cycled at a consistent speed (within a given condition) this brought trials largely into alignment: *e.g.,* the top of each cycle occurred at nearly the same time for each trial. The adaptive alignment procedure was used to correct any remaining slight misalignments. The time-base for each trial was scaled so that the position trace on that trial closely matched the average position of all trials. This involved a slight non-uniform stretching, and resulted in the timing of all key moments – such as when the hand passed the top of the cycle – being nearly identical across trials. This ensured that high-frequency temporal response features were not lost to averaging.

All variables of interest (firing rate, hand position, hand velocity, EMG, etc.) were computed on each trial before adaptive alignment. Thus, the above procedure never alters the magnitude of

these variables, but simply aligns when those values occur across trials. The adaptive procedure was used once to align trials within a condition on a given recording session, and again to align data across recording sessions. This allowed, for example, comparison of neural and muscle responses on a matched time-base.

The median hand velocity of the cycles was calculated for the average firing rate for each neuron, and the cycles were then binned into speed bins of size 0.2Hz, with the first and last speed bins containing the spillover.

To calculate the modulation of the firing rate across speeds, the dynamic range was first calculated for each neuron and each speed by subtracting the minimum firing rate from the maximum firing rate over the course of the trajectory. Next, for each neuron, a line of best fit was calculated for the vector containing the dynamic range values versus the vector of speeds. If the slope of this line was positive, the firing rate increased with speed, and vice versa.

Preprocessing and PCA

Because PCA seeks to capture variance, it can be disproportionately influenced by differences in firing rate range (*e.g.*, a neuron with a range of 100 spikes/s has 25 times the variance of a similar neuron with a range of 20 spikes/s). This concern is larger still for EMG, where the scale is arbitrary and can differ greatly between recordings. The response of each neuron / muscle was thus normalized prior to application of PCA. EMG data were fully normalized: $response = \frac{response}{range(response)}$, where the range is taken across all recorded times and conditions. Neural data were 'soft' normalized: $response = \frac{response}{range(response)+5}$. This is a standard technique [4] to balance the desire for PCA to explain the responses of all neurons with the desire that weak responses not contribute on an equal footing with robust responses. In practice, nearly all neurons had high firing rate ranges during cycling, making soft normalization nearly identical to full normalization.

PCA per speed

Neural data in each speed were formatted as a 'full-dimensional' matrix, X_s^{full}, of size $n \times T$, where n is the number of neurons and $t \in [1, T]$ indexes across all analyzed times in the trajectory taken during speed s. We similarly formatted muscle data as a matrix, Z_s^{full}, of size

$m \times T$, where m is the number of muscles. Because PCA operates on mean-centered data, we mean-centered X_s^{full} and Z_s^{full} so that every row had a mean value of zero. PCA was used to find X_s, a reduced-dimensional version of X_s^{full} with the property that $X_s^{full} \approx V_s X_s$, where V_s are the speed-specific principal components, one per speed. PCA was similarly used to find Z_s, the reduced-dimensional version of Z_s^{full}.

PCA across conditions

To calculate the 'global subspace', i.e. the subspace that explains a high variance across all speeds, neural data were first formatted as a 'full-dimensional' matrix, X^{full}, of size $n \times T$, where n is the number of neurons and $t \in [1, T]$ indexes across all analyzed times and conditions. We mean-centered X^{full} and then used PCA to find X, a reduced-dimensional version of X^{full} with the property that $X^{full} \approx VX$, where V are now the 'global' principal components. We employed analogous methods for the muscle data to obtain Z, a reduced-dimensional version of Z^{full}.

Recurrent Neural Networks

We trained recurrent neural networks (RNNs) to produce a target output consisting of the muscle data projected into its top 6 PCs, with a static speed input. Networks consisted of 50 nodes, since roughly the same number of neurons were recorded in each monkey. The networks had the following dynamics:

$$v(t+1, s) = v(t) + \frac{\Delta t}{\tau}(-v(t,s) + Af(v(t,s)) + Bu(t,s) + b)$$

Here, $v(t,s)$ can be thought of as the membrane voltage, and $f(v(t,s))$ is the state of the network at time t and speed bin s. The network was simulated for 2000 timesteps and trained on the data for 8 speed bins, thus $t \in [1, 2000]$ and $s \in [1, 8]$. We consider the state of a network node as representing the firing rate of that node. The function $f := tanh$ is an element-wise transfer function linking a unit's input to its firing rate. The matrix A captures the connection weights between the network units, and $Bu(t,s)$ captures the effect of the external

input. The vector b captures the bias for each network node. Network output is a linear readout of its firing rates:

$$y(t,s) = Cf(v(t,s)) + d$$

, with d representing the bias vector in the output. The parameters A, B, C, b, d were optimized to minimize the mean squared error between the network output y and the target output y_{targ}. The external input was of constant amplitude a that was a linear function of the speed bin s: $a = 0.5 + 0.5\frac{s-1}{8}$. To avoid overly specific solutions, the duration of the input was varied: the input was of amplitude a from timestep 800 to n, where n was sampled from a uniform distribution between 1500 to 1900, i.e. $n \sim U[1500, 1900]$. The input was of amplitude 0 for the rest of the timesteps. The target output was of amplitude 0 when the input was 0. When the input was of amplitude a, the output was the muscle activity in the corresponding speed bin s, projected into the top 6 PCs and downsampled by a factor of 4. The muscle activity was repeated in time for as many cycles as necessary to fill the time period where the input was 'on' (anywhere from 3 to 8 cycles). The different output cycles were very similar for the trained network, with the exception of the first cycle for some of the trajectories. Thus, one of the middle cycles was used to plot the network activity in **Figure 3d**. Networks were also trained in the presence of moderate additive noise, with noise variance 0.01, to verify that the results held in the presence of noise. RNNs were trained using TensorFlow's Adam optimizer.

Linear decoding of neural firing rates

The single neuron firing rates were regressed against the muscle factors and the network factors separately. First, PCA was performed to muscle signals to produce two- or three-dimensional factors. Then, ridge regression was performed using the muscle factors, with each neuron's firing rate as a separate predictor. The data was divided into ten folds and the final regularization parameter was taken as the mean of the regularization parameters fit to each fold. A population R^2 was computed to quantify the fit to the single neurons, which is the average R^2 across neurons, taking into account different response magnitudes. Next, the same was done using the network factors, and the R^2 were compared as in **Figure 3c**.

Fit to ellipses

The trajectories were first projected into their top 2 PCs, and the best-fit ellipse to the resulting 2D trajectory was found using least squares [39]. The R^2 to the ellipse was calculated by first uniformly sampling this ellipse with the same number of time points as the trajectory, and then calculating the population R^2 in time. Some muscle trajectories did not have an ellipse as a solution, and were instead better fit by a hyperbole. We discarded these in subsequent tests of the differences in R^2.

Understanding network structure

Network perturbations

To ascertain the presence of a limit cycle, the nominal network trajectories in PC space were perturbed randomly by ε, where $\varepsilon \sim U[-0.1, 0.1]$. This was first transformed into the full-dimensional global space of the network trajectories, and the network was started from this initial condition with the same static input as for the nominal trajectory. The resulting network trajectory after 2000 time points was then transformed back into PC space, and it was verified that these returned back to the nominal trajectory. This procedure was also performed for any one nominal trajectory while providing different speed inputs. Speed inputs that the network had not seen before were also tested, namely (a) static speed inputs with values between the eight input speeds, (b) dynamic inputs taking on values from the slowest speed to the fastest speed in a ramp manner.

Unique inputs per speed

To train networks that received unique inputs for each speed (**Supplementary Figure 2**), networks were trained with the same equations as above, with u now a one-hot encoding vector of dimension 8, i.e. if the input speed bin was 3, the static input would be $[0,0,1,0,0,0,0,0]$.

Tangling

Trajectory tangling *Q(t)* is described in detail in [4], and is defined as the following.

$$Q(t) = \max_{t'} \frac{\|\dot{x}(t) - \dot{x}(t')\|_2^{\,2}}{\|x(t) - x(t')\|_2^{\,2} + \varepsilon}$$

Here, $x(t)$ is the neural state at time t, i.e. a vector of firing rates at time t, $\dot{x}(t)$ is the temporal derivative at time t, $\|\cdot\|$ is the Euclidean norm, and ε is a small constant that prevents division by 0. Briefly, the neural state, $x(t)$ is a vector comprised of the tth column of X, where X is the neural activity projected into its principal components. Muscle tangling was computed analogously, based on Z. We computed the derivative of the state as $\dot{x}_t = \frac{x_t - x_{t-\Delta t}}{\Delta t}$, where Δt was 1 ms. The constant ε (set to 0.1 times the variance of x) determines how small the denominator can become, which prevents the denominator from shrinking below 0.1 times the average squared distance from zero.

Path similarity

Path similarity was calculated in a two dimensional subspace. The neural data at the reference speed $X_{s_{ref}}^{full}$ was first projected into its PCs 1 and 2 to form $X_{s_{ref}}$ of size $2 \times T_{s_{ref}}$. The trajectory at the speed bin $s = s_{ref} + s_{diff}$ was first projected into its own PCs 1 and 2, and then interpolated or compressed to be of the same length as the reference trajectory $T_{s_{ref}}$, i.e. trajectories were resampled to have the same number of sampled timepoints. We then wanted to apply a rigid rotation to X_s to minimize the root mean squared (RMS) distance from the reference data $X_{s_{ref}}$, i.e. find the rotation matrix R that minimizes the cost function $\min_R \|X_{s_{ref}} - RX_s\|_F \;\; s.t.\, R^T R = I,\; det(R) = 1$. The solution was implemented using singular value decomposition as in the Kabsch algorithm [40] for $s_{diff} = \{\pm 1,\ \pm 2,\ \ldots\ \pm 5\}$. The coefficient of determination $R^2(x_{s_{ref}}, Rx_s)$ between the reference trajectory and the rotated trajectory was calculated, and averaged across all values for s_{ref}. Only the values of s_{diff} that kept s between the admissible values of 1 and 8 were taken. Analogous methods were applied for the 2D subspace formed by projecting into PCs 3 and 4, and PCs 5 and 6, as well as for the muscle data.

Speed axis

The speed axis was calculated by finding the one-dimensional axis in the neural data that is orthogonal to the first two PCs, that best decodes the mean-centered mean angular velocity. The neural data was first projected into its top 12 PCs. To create **Fig. 6c**, we plot the data projected the data into the top 2 PCs, against the data projected into the speed axis. We used the data projected into PCs 3-12 to find the speed axis, where PCA was performed across speeds. We calculated the mean of the neural data in each speed s as $\bar{x}_s = \sum_{t=1}^{T_s} x_s(t)$, where $x_s(t)$ is the data for speed s at time t, projected into the across-speed PCs 3-12 and is thus of size 10×1. Finally, we found the best linear decoder d of size 10×1 that minimizes the following cost function.

$$\min_d \| [\bar{x}_1\ \bar{x}_2 \ldots \bar{x}_8]\ d\ -\ [\bar{v}_1\ \bar{v}_2 \ldots \bar{v}_8]\ \|_2$$

$$s.t.\ \|d\|_2\ =\ 1$$

Here, $\bar{v}_s$ is the mean-centered mean angular velocity of the hand during speed bin s. We projected all the neural data into this speed axis d to create **Fig. 6c**. Analogous methods were used on the muscle data to create **Fig. 6c**.

Simulated Trajectories

We wanted to test how high trajectory tangling could reach for neural data if the trajectories are constrained be following the same path at different speeds. In order to simulate these trajectories, we used the path taken during speed bin 6 as the reference path and interpolated or compressed this trajectory to generate the trajectories at all other speeds. In order to include the effect of biophysically-based noise to avoid the tangling becoming artificially high, we simulated spike trains for each neuron at each speed, using the trajectory as the conditional intensity function. The same number of trials were simulated as the original number of trials for that neuron. The thinning method was used to simulate spike trains; this simulates a point process given a bounded conditional intensity function [41]. The same preprocessing steps were applied to these simulated spike trains as the neural recordings, and then trajectory tangling was computed as before.

Separation Removed Trajectories

To test the effect of separation in terms of translation and rotation, we artificially removed these two forms of separation between the trajectories in high-dimensional space and calculated the trajectory tangling. Specifically, we used the speed bin 6 as the reference condition and interpolated or compressed the data in other speed bins to the number of time points in the reference condition. We then applied the optimal rotation to minimize RMS between the trajectories in all speed bins to the trajectory in the reference speed bin using the Kabsch algorithm, in the same way as described in the Path Similarity section of the Methods.

Distance between trajectories

We first interpolate all neural trajectories in each cycling direction to be the same length as the trajectory during the reference speed bin, here $s_{ref} = 4$. We calculate the distance from the trajectories at all speeds as $\Delta_{x_s}(\theta) = \|x_{s_{ref}}(\theta) - x_s(\theta)\|_2$. The phase θ is the angular displacement of the limb, with $\theta = 0$ corresponding to the limb being in the vertical position. When data was not available for the same angular displacement for the two trajectories under comparison, the closest values for angular displacement were taken. The distance between the means is similarly calculated as $\Delta_{\bar{x}} = \|\bar{x}_{s_{ref}} - \bar{x}_s\|_2$, where $\bar{x}_s = \sum_{t=1}^{T_s} x_s(t)$.