Supplementary Figure S1. Mid-movement force during training and transfer. Note that this figure is identical in format to figure 4 in the main text, except that mid-movement force is used to measure adaptation and generalization rather than the raw adaptation coefficient. (A) The unfilled gray circles represent the mid-movement force (determined over a 70 ms window around the peak velocity) for the final five movements during the adaptation period for all subjects. The filled black circles indicate the average mid-movement force and movement velocity for each subject. The blue circles are the average mid-movement force binned by movement velocity. Ten velocity bins are displayed ranging from 0.25 and 0.35 m/s. (B) The blue circles display the same data shown in panel A. The three different lines represent different generalization functions: Linear Extrapolation Transfer (light gray), Level Transfer (black) and Gaussian Decay Transfer (dark gray, dashed). The red and pink circles display the mean mid-movement force and velocity for fast movements made during the post-adaptation period. The vertical lines represent standard error. (C) Comparison of the mid-movement force for the first fast movement post-adaptation with predictions for three different mechanisms for generalization. The error bars show standard error.
Supplementary Figure S2. Velocity-normalized adaptation coefficients during training and transfer. Note that this figure is identical in format to figure 4 in the main text, except that velocity-normalized adaptation coefficient is used to measure adaptation and generalization rather than the raw adaptation coefficient. Also note that because the velocity-normalized adaptation coefficient is a measure of motor output gain, rather than raw motor output, the transfer prediction change in shape. (A) The unfilled gray circles represent the velocity-normalized adaptation coefficients for the final five movements during the adaptation period for all subjects. The filled black circles indicate the average velocity-normalized adaptation coefficient and movement velocity for each subject. The blue circles are the average velocity-normalized adaptation coefficients binned by movement velocity. Ten velocity bins are displayed ranging from 0.25 and 0.35 m/s. (B) The blue circles display the same data shown in panel A. The three different lines represent different generalization functions: Linear Extrapolation Transfer (light gray), Level Transfer (black) and Gaussian Decay Transfer (dark gray, dashed). The red and pink circles display the mean velocity-normalized adaptation coefficient and velocity for fast movements made during the post-adaptation period. The vertical lines represent standard error. (C) Comparison of the velocity-normalized adaptation coefficient for the first fast movement post-adaptation with predictions for three different mechanisms for generalization. The error bars show standard error.
Supplementary Figure S3. Spatial generalization for the three primitive configurations. Displayed are the patterns of adaptation (amount of mid-movement force) generalization to different movement directions for the three models when the velocity vector is represented in terms of magnitude and direction: (A) isotopic force learning, (B) anisotropic force learning and (C) anisotropic gain learning primitives. The white lines represent the region of velocity space over which the force-field is trained for each model (v_x = 0 m/s, v_y = 0 to 0.3 m/s). The colored curves represent the range of movement directions (-180 to 180 degrees, speed of 0.3 m/s) that the spatial generalization functions in panel D are derived from. That is, the functions in panel D represent the amount of mid-movement force transferred to different movement directions (0 degrees is the training direction) for the three models. Note the similarity between the spatial generalization functions for the isotopic force learning (red trace) and anisotropic gain learning models (black trace).
**Wide force-encoding primitives cannot rapidly learn a linear force-velocity relationship**

Supplementary Figures 4 and 5 show that, compared to gain-encoding primitives, wide force-encoding primitives (Supplementary Figure 5) have trouble learning the linear velocity-dependent force-field that we studied \( f = Bv \) although narrow force-encoding primitives (Supplementary Figure 4) can readily learn this input output relationship. The first row of plots in these two figures illustrates the trial-to-trial evolution of motor output during the training period. Note that on the first few trials (the 1st three trials after exposure to the FF are colored green, red, and cyan, respectively) the narrow gain-encoding primitives (NGPs), the narrow force-encoding primitives (NFPs), and the wide gain-encoding primitives (WGP) all produce patterns of motor output that are shaped similarly to the desired output (thick blue lines) although, initially, the magnitude of the motor output is smaller than desired. However the wide force-encoding primitives (WFPs) initially produce a pattern of motor output that is shaped very differently from the desired motor output pattern as shown in the upper right panel of Supplementary Figure 5 (green, red and cyan lines). As a result, the initial learning of the desired input-output relationship (which is accomplished in the first 3-5 trials) consists of learning the mean output while the desired output shape (the ramp) is not well approximated. This problem occurs for ramp-learning, but not constant-value-learning for wide primitives, but (as noted above) does not occur at all for narrow primitives. As training progresses, the ramp-shape is “slowly” learned, but this learning takes orders of magnitude longer than for the constant-value-shape. There are several ways to think about why this occurs:

1. Wide Gaussians (compared to narrow ones) are mostly flat (their slopes are smaller if they have the same height). Therefore they are good at approximating flat (constant-value) shapes but not rapidly-changing shapes, meaning that large weighting factors (which would take a long time to learn) would be needed to approximate shapes with large slopes.

2. In order to represent a ramp, each of several narrow Gaussians can learn to represent part of the ramp function, without interfering much with other parts of this function. However, when wide Gaussians try to represent part of a ramp function their breadth (adversely) influences other parts of the function.

3. In order to learn a ramp with wide Gaussians, some Gaussian elements must learn increases in weights, whereas other nearby (and overlapping) elements must learn decreases (see the middle right panel of Supplementary Figure 5). Note that this is closely related to the idea that weights must be large in magnitude to represent rapidly changing shapes, because large weights would lead to large output values as well as large output slopes if not accompanied by output from nearby primitives that could effectively cancel the output values (but not the slopes). This is illustrated in the middle right panel of Supplementary Figure 5. This opposite learning for overlapping basis elements is, in a gross sense, wasteful and inefficient because much of what is “learned” by one learning element is cancelled out by another. Overall, this process is slow and likely to go unstable if learning rates are too high.
In summary, Supplementary Figures 4 and 5 illustrate the fact that WFPs generally have trouble approximating ramp-like input-output relationships when the WFP width is not much smaller than the ramp width. This does not mean that wide WFPs cannot be trained to represent ramp-like input-output relationships, but instead that a prolonged training period is required for such representations to be formed (see the lower right panel of Supplementary Figure 5), because opposing contributions from overlapping basis elements are required. Note, as mentioned above, that force-encoding primitives would need to learn a ramp-like input-output relationship on our task (or any task with linear dynamics) whereas gain-encoding primitives need to learn a constant-value input-output relationship, so the difficulty that WFPs have with ramp-like functions would translate to poor learning for WFPs but not WGPs.
Supplementary Figure S6: The relationships between tuning width, stability, learning rate ($\alpha$), adaptation levels, and generalization for gain-encoding versus force-encoding motor primitives.

The effects of encoding modality, primitive width and learning rate on stability, adaptability, and extrapolation.

While it’s useful to understand that wide basis functions generally require longer training periods to learn ramp-like input-output relationships compared to constant-value input-output relationships, the practical (and quantitative) question that must be addressed is: To what extent the learning curves observed in the current study (and similar studies of force-field adaptation) limit the widths of the force-encoding primitives (FPs) that would be capable of producing similar amounts of learning in a similar number of training trials?

Supplementary Figure 6 shows the results of an analysis that attempts to answer this question. Since the FPs learn slowly, they perform best when the learning rates are high. However, if the learning rate gets too large, instability and oscillatory behavior can result. Thus, we took the approach of first computing the maximum stable learning rates (Supplementary Figure 6, column 1) and then characterizing the ability to learn (columns 2 and 3) and extrapolate (column 4).

Instability can be defined in several different ways for our purposes. The most extreme type of instability (frank instability) is when the learning rate is so high that overlearning results in an unbounded expansion of motor output from one trial to the next. A less extreme type of instability is when the resulting overlearning results in an oscillatory pattern of motor output from one trial to the next that may converge with repeated practice – this is sometimes called overshoot in engineering terminology. Note that, as far as we are aware, neither of these types of instability (frank instability or overshoot) has ever been observed during motor adaptation, and so it is reasonable to believe that learning rates that are large enough to produce these types of instabilities are not biologically plausible. The green and navy plots show the maximum
learning rates that avoid frank instability and overshoot, respectively, for movements with a peak speed of 0.3 m/s, whereas the light-blue and red plots show the maximum learning rates that would avoid frank instability and overshoot, respectively, for movements with a peak speed of 1.0 m/s.

We ran a battery of simulations to determine the values for alpha that would lead to each of the four levels of instability discussed above over a range of values for sigma extending from 0.1 m/s to 1.2 m/s for gain-encoding and force-encoding primitives. The first column of plots in Supplementary Figure 6 show the values of these learning rates. Since none of these levels of instability has ever been observed experimentally, we focus our analysis on the behavior of models that avoid all of them (indicated by the thick red line in each of the panels.

The second and third columns of Supplementary Figure 6 characterize the levels of adaptation achieved during the training period by showing the adaptation errors. The second column of plots displays the normalized mean squared error (MSE), whereas the third column displays the normalized error in the average slope of the learned force-velocity relationship. In both cases, a value of one reflects performance that is no better than naïve and a value of zero would indicate ideal performance. Note that the gain-encoding models perform well for both measures with a maximum slope error of less than 7% (0.07) and a maximum MSE of less than 2%. However the force-encoding models perform much more poorly at large values of sigma. This trend is also reflected in the fourth column, which characterizes the absolute amount of extrapolation to a velocity 70% faster that what was trained: Larger primitive widths (sigmas) lead to monotonic improvements in extrapolation for the gain-encoding primitives, whereas force-encoding primitives actually show a reduction in extrapolation levels as sigma increases because progressive worsening in the ability to learn the trained adaptation more than outweighs the ability to extrapolate what is learned as sigma increases.
Supplementary Figure S7. The effect of primitive density on anisotropic force-encoding primitives. Displayed are the results of increasing the density of the anisotropic force encoding primitives when the velocity vector is represented in terms of magnitude and direction ($r_v$ and $\theta_v$, see Methods). The radial centers of the primitives were located at [0.05 and 0.55] (column 1), [0.05, 0.21, 0.38, 0.55] (column 2), and [0.05, 0.15, 0.25, 0.35, 0.45, 0.55] (column 3). For each primitive and configuration, $\sigma_r=0.5$ m/s and $\sigma_\theta=0.4$ radians. The primitives are tiled across velocity space in the circumferential direction with a spacing of about $\sigma/2$ (0.2 radians), but only a fourth are displayed for clarity. The layout and information displayed in each column is the same as each column of Figure 7. The resulting velocity profiles (red traces in panels M, N and O) for the three different extent configurations at the initial transfer speed ($v_y = 0.49$ m/s) fail to match the shape for the anisotropic gain-encoding primitives (black traces).
Supplementary Figure S8. The effect of radial extent on anisotropic force-encoding primitives. Displayed are the results of increasing the radial extent of the anisotropic force encoding primitives when the velocity vector is represented in terms of magnitude and direction (\( r_v \) and \( \theta_v \), see Methods). The radial extent, \( \sigma_r \), of the primitives is 0.5 (column 1), 0.75 (column 2), and 1.0 m/s (column 3). For each primitive and configuration \( \sigma_\theta = 0.4 \) radians. The primitives are tiled across velocity space in the radial direction with a spacing of \( \sigma_r / 2 \) (0.25, 0.375 and 0.5 m/s respectively) and about \( \sigma_\theta / 2 \) (0.2 radians) in the circumferential direction, but only a fourth are displayed for clarity. The layout and information displayed in each column is the same as each column of Figure 7. The resulting velocity profiles (red traces in panels M, N and O) for the three different extent configurations at the initial transfer speed (\( v_y = 0.49 \) m/s) fail to match the shape for the anisotropic gain encoding primitives (black traces).
Anisotropic primitives when represented in Cartesian coordinates

In the manuscript the anisotropic primitives are represented in velocity space in terms of magnitude and direction and the gain encoding model makes clear predictions (Figure 7) that are observed in the experimental data. These results are clearly not predicted by the force encoding models. Though somewhat more complicated, the same results also hold when these anisotropic primitives are represented in Cartesian coordinates in velocity space. Below we present these simulations in Cartesian coordinates (Supplementary Figure 9).

The first complication with the Cartesian representation is with the Gaussian-shaped anisotropic gain encoding motor primitives. In this case there is a non-biological negative force generalization for movements made in the opposite direction of the training (Supplementary Figure 9F). This occurs because of the wide, symmetric radial tuning of the basis elements which, because of their width, extend even beyond the origin. There is neurophysiological evidence that suggests that although basis elements are widely tuned in the radial direction, this tuning is asymmetrically smaller at lower magnitudes and larger at higher magnitudes in such a way that the breadth of tuning is consistent with logarithmic scaling (Bruce and Goldberg 1985; Bruce et al. 1985; Sparks et al. 1976). In Supplementary Figure 10 we show that introducing this type of asymmetry eliminates these negative responses.

The second and more challenging complication is that for the Cartesian representation it is possible to chose model parameters that result in linear extrapolation of the trained adaptation for the anisotropic force encoding primitive model. However, when confined to the modeling parameters that are experimentally derived (the generalization of adaptation across different movement directions) and those that are determined by the current experiment (duration of training) and other factors (learning rate values that maintain stability), we show that neither increasing the density of the anisotropic force encoding primitives (Supplementary Figure 11) nor extending the radial extent (Supplementary Figure 12) reproduces the hypergeneralization demonstrated in our experimental results and simulation with gain anisotropic encoding primitives.


Modeling of generalization to novel movement speeds in Cartesian Coordinates

The main difference for these simulations was the representation of the anisotropic primitives in velocity space. Here $v_{centerX}$ and $v_{centerY}$ represent the center of the primitives in velocity space and the velocity vector was expressed as $v_x$ and $v_y$. The Cartesian representation utilized $\sigma_{direction}$ (0.12 m/s) - the estimated circumferential tuning from previous studies (Donchin et al 2003; Shadmehr 2004; Thoroughman and Shadmehr 2000). The primitives were oval in shape and tiled across velocity space with a radial center of 0.6 m/s. These basis elements are illustrated in the top row of Supplementary Figure 9.

For both anisotropic models, the basis element, $g_k(v)$, is a function of velocity:

$$g_k(v) = e^{-(v^TR^T\Sigma^{-1}RV)}$$

where

$$V = \begin{bmatrix} v_x - v_{Xcenter,k} \\ v_y - v_{Ycenter,k} \end{bmatrix}$$

10
\[ R = \begin{bmatrix} \cos(\theta_k) & \sin(\theta_k) \\ -\sin(\theta_k) & \cos(\theta_k) \end{bmatrix} \] is the rotation matrix,

\[ \theta_k = \arctan\left(\frac{\nu_{Ycenter,k}}{\nu_{Xcenter,k}}\right) \] is the circumfrential location of the primitive center,

\[ \Sigma = \begin{bmatrix} \sigma_{\text{magnitude}}^2 & 0 \\ 0 & \sigma_{\text{direction}}^2 \end{bmatrix} \] is the unrotated covariance matrix,

and \( V^T \) and \( R^T \) represent the transpose of the above matrices.
**Supplementary Figure S9. Comparison of adaptation generalization for anisotropic motor primitives when represented in Cartesian coordinates.** Displayed is the comparison of the two different hypothetical anisotropic configurations of the neural bases (each column) used to simulate the generalization of adaptation to novel movement speeds. The configurations are represented in Cartesian coordinates and different in the relationship learned between movement speed and force during adaptation (either an absolute force-velocity (A) or force gain-velocity relationship (B)). (Note that the basis element illustrations in A and B show contours at the one-sigma point and only half of the elements are displayed in order to reduce visual overcrowding. Select bases are highlighted for clarity of their shape in velocity space.) In all cases the relationship learned by the bases during adaptation decayed in a Gaussian manner as the movement velocity moved away from the center of the tuning curve. (C and D) The pattern of adaptation (amount of mid-movement force) generalization to different movement velocities for the two configurations. The heat map represents the generalization of adaptation in terms of lateral force (N). Note that the color bar alongside panel D applies to the entire row. The white lines represent the region of velocity space over which the force-field is trained for
each model \( v_x = 0 \text{ m/s}, \ v_y = 0 \text{ to } 0.3 \text{ m/s} \). The next two rows display the learned force-velocity (\textbf{E and F}) and force gain-velocity relationships (\textbf{G and H}) along the slice of velocity space corresponding to the movement direction experienced during the training \( v_x = 0, \ v_y = -0.7 \text{ to } 0.7 \text{ m/s}, \) the range enclosed by the black dashed-line rectangle). The red and green lines in these panels highlight the relationships between force and velocity, or force gain and velocity that were trained, corresponding to the white line regions in panels \textbf{C} and \textbf{D}. Note that the red lines extend from the origin with a slope of 15 Ns/m = B, whereas the green lines have a constant value of 15 Ns/m over the range of 0 to 0.3 m/s for \( v_y \). The red and green circles display the extent to which the learning predicted by each model would generalize to the untrained, high-speed movements we observed during the initial high-speed testing trial \( v_y = 0.49 \text{ m/s} \). The spikes in the black trace in panel \textbf{G} is the result of division by zero in determining the force gain \( F / v_y \) at movement speeds near zero. The bottom row (\textbf{I and J}) displays the resulting lateral force patterns for the two different primitive configurations at the initial transfer speed \( v_y = 0.49 \text{ m/s} \).
Supplementary Figure S10. The effect of clamping negative responses for anisotropic gain-encoding primitives in Cartesian coordinates. Displayed are the results of introducing an asymmetry in the radial extent of the anisotropic gain encoding primitives when represented in Cartesian coordinates. The information displayed in each panel is the same as each column of Supplementary Figure 9. The asymmetry in the radial extent of the primitives (clamping any response beyond the origin in the radial direction of the primitive to zero) results in less overlap of the primitives near the origin and eliminates the negative force generalization (compare panels B and C with panels D and F of Supplementary Figure 9).
Supplementary Figure S11. The effect of primitive density on anisotropic force-encoding primitives in Cartesian coordinates. Displayed are the results of increasing the density of the anisotropic force encoding primitives when represented in Cartesian coordinates. The radial centers of the primitives were located at [0.1 and 0.6] (column 1), [0.1, 0.27 0.44, 0.6] (column 2), and [0.1, 0.2, 0.3, 0.4, 0.5, 0.6] (column 3). For each primitive and configuration \( \sigma_{\text{direction}} = 0.12 \) and \( \sigma_{\text{magnitude}} = 0.3 \). The primitives are tiled across velocity space in direction with a spacing of 0.2 radians, but only half are displayed for clarity. The layout and information displayed in each column is the same as each column of Supplementary Figure 9. The spikes in the black traces in the fourth row are the result of division by zero in determining the force gain \( (F / v_y) \) at movement speeds near zero. The resulting velocity profiles (red traces in the bottom row) for the three different density configurations at the initial transfer speed \( (v_y = 0.49 \text{ m/s}) \) do not match the shape for the anisotropic gain encoding primitives (black traces).
Supplementary Figure S12. The effect of radial extent on anisotropic force-encoding primitives in Cartesian coordinates. Displayed are the results of increasing the radial extent of the anisotropic force encoding primitives when represented in Cartesian coordinates. The radial extent of the primitives were 0.3 (column 1), 0.45 (column 2), and 0.6 m/s (column 3). For each primitive $\sigma_{\text{direction}}=0.12$. The primitives are tiled across velocity space in direction with a spacing of 0.2 radians, but only half are displayed for clarity. The layout and information displayed in each column is the same as each column of Supplementary Figure 9. The spikes in the black traces in the fourth row are the result of division by zero in determining the force gain ($F/Y_v$) at movement speeds near zero. The resulting velocity profiles (red traces in the bottom row) for the three different extent configurations at the initial transfer speed ($v_y = 0.49$ m/s) fail to match the shape for the anisotropic gain encoding primitives (black traces).