nature research

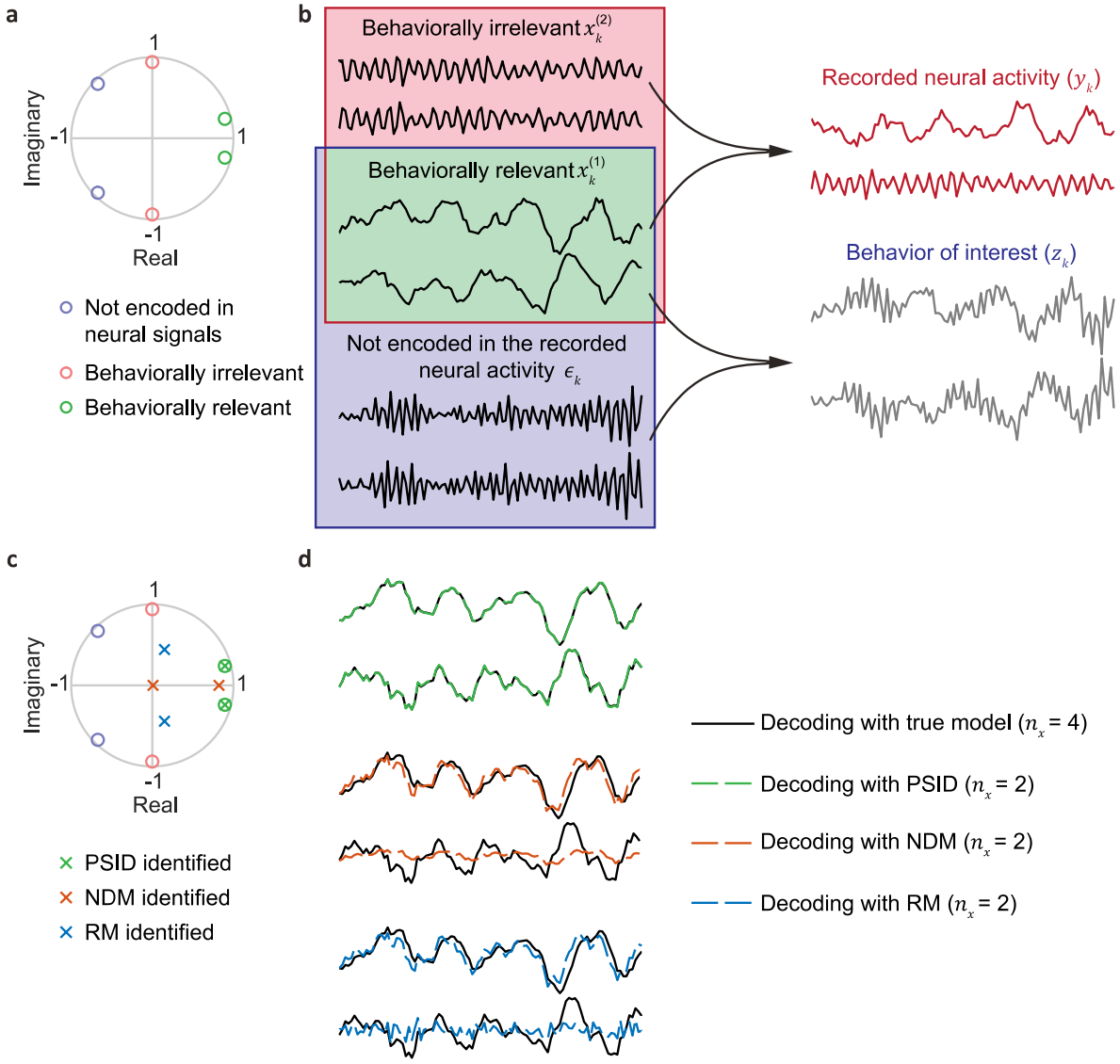
https://doi.org/10.1038/s41593-020-00733-0

Supplementary information

Modeling behaviorally relevant neural dynamics enabled by preferential subspace identification

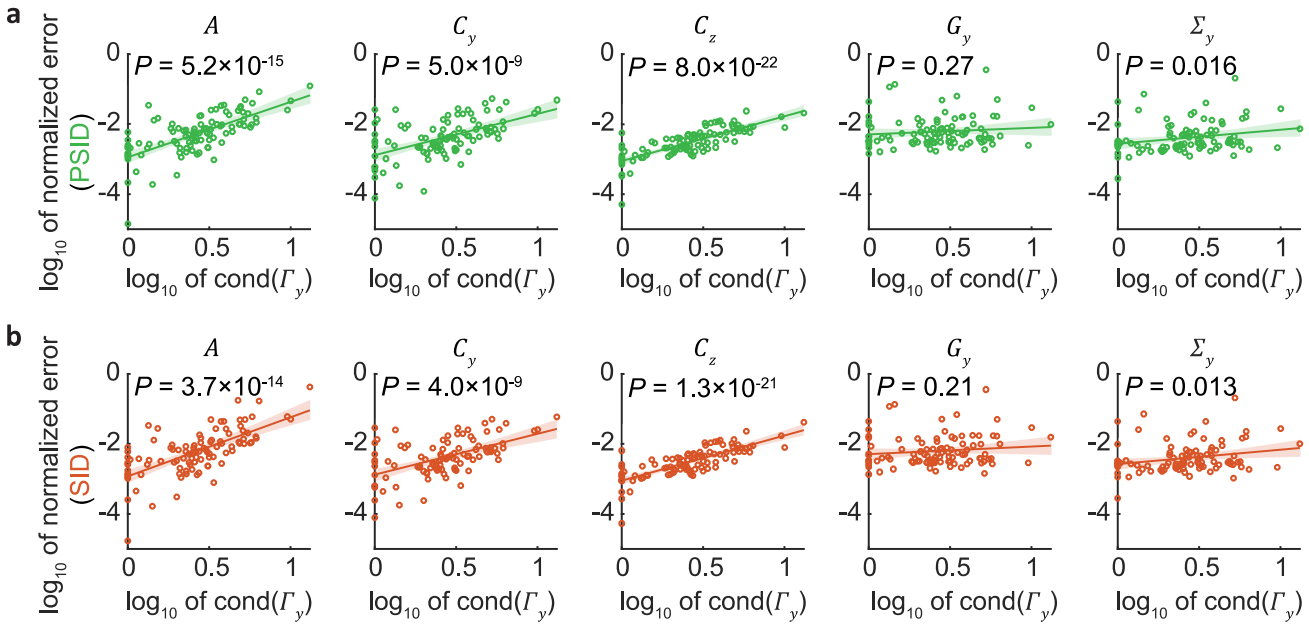
In the format provided by the authors and unedited

Supplementary Figures



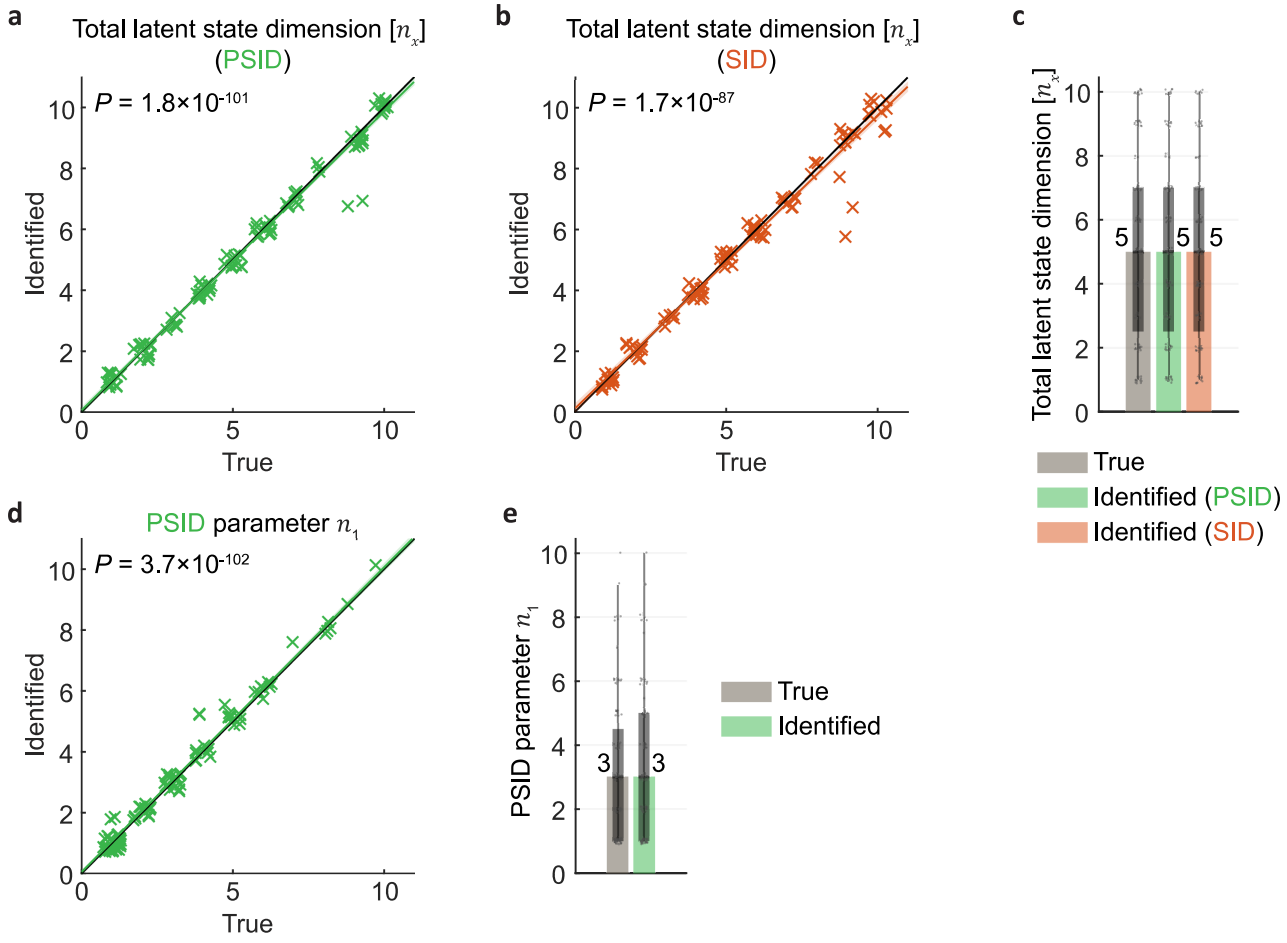
Supplementary Fig. 1 | An example numerical simulation shows how brain states could be related to behavior, neural activity or both and which states are identified by PSID, NDM and RM.

(**a**) The eigenvalues of a simulated brain model that are color coded to show whether they are behaviorally relevant, behaviorally irrelevant, or not encoded in neural activity. (**b**) Sample data for each latent state and the generated neural activity and behavior. Two latent states ($x_k^{(2)}$) only drive the recorded neural activity (y_k), two latent states (ϵ_k) only drive behavior (z_k) and two latent states ($x_k^{(1)}$) drive both. In other words, neural activity (y_k) is a linear combination of behaviorally relevant ($x_k^{(1)}$) and behaviorally irrelevant ($x_k^{(2)}$) states, and behavior is a linear combination of behaviorally relevant states ($x_k^{(1)}$) and states not encoded in neural activity (ϵ_k). (**c**) Eigenvalues identified using PSID, NDM and RM when a model with a 2-dimensional latent state is fitted to the simulated neural activity and behavior data from (b). To identify the eigenvalues using the training data, NDM only considers neural activity, RM only considers behavior, and PSID considers both. (**d**) The neural decoding of behavior with models learned using PSID (green), NDM (red) and RM (blue), as well as the best possible decoding when the true model is used for decoding (black). Decoding for all methods is done in test data by applying a Kalman filter associated with the learned model to the test neural activity alone to extract the latent states, and then computing a linear combination of these states (Fig. 1c).



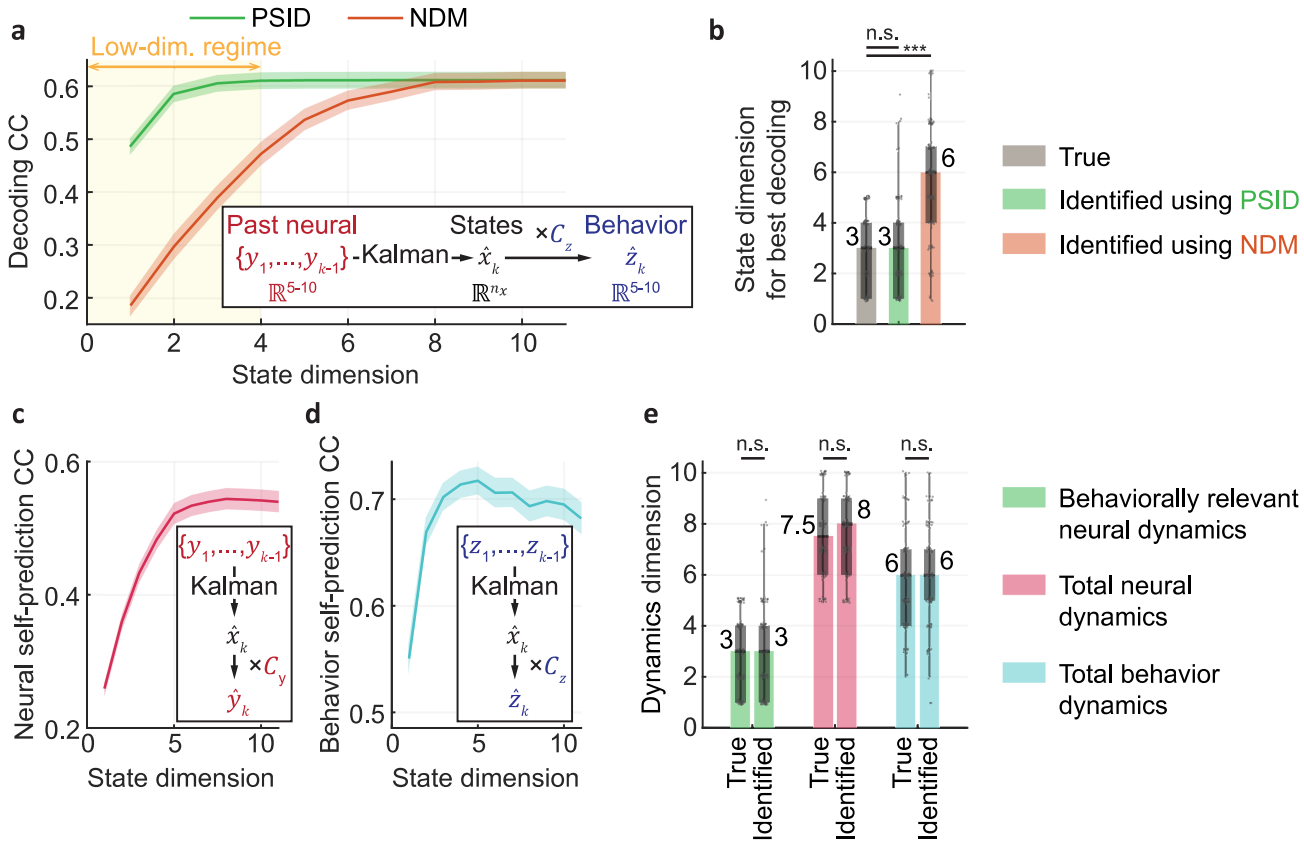
Supplementary Fig. 2 | Model identification error is correlated with a measure of the inherent difficulty of identification in random models.

(**a**) Normalized error of each parameter is shown as a function of the condition number of the neural observability matrix Γ_y (equation (40) in Supplementary Note 4) for the model, denoted by $\mathrm{cond}(\Gamma_y)$, which is defined as the ratio of its largest to its smallest singular values. Models are the same 100 random models as in Extended Data Fig. 2. The *P*-value for Pearson's correlation coefficient between $\log_{10}$ of $\mathrm{cond}(\Gamma_y)$ and $\log_{10}$ of normalized error is shown on each plot (*n* = 100 random models). The solid line shows the least squares solution to fitting a line to the data points and the shaded area shows the associated 95% confidence interval. The condition number of the neural observability matrix for each model is significantly correlated with the identification error for the three model parameters (i.e. A, C_y, and C_z) that have the widest range of identification errors (Extended Data Fig. 2a). As a model gets closer to being unobservable and more difficult to identify, the condition number for the observability matrix increases. Thus this result indicates that the models for which these three parameters were less accurately identified were closer to being unobservable and thus were inherently more difficult to identify given the same number of training samples. (**b**) Same as (a) for SID, which similarly shows relatively larger error for models that are inherently less observable.



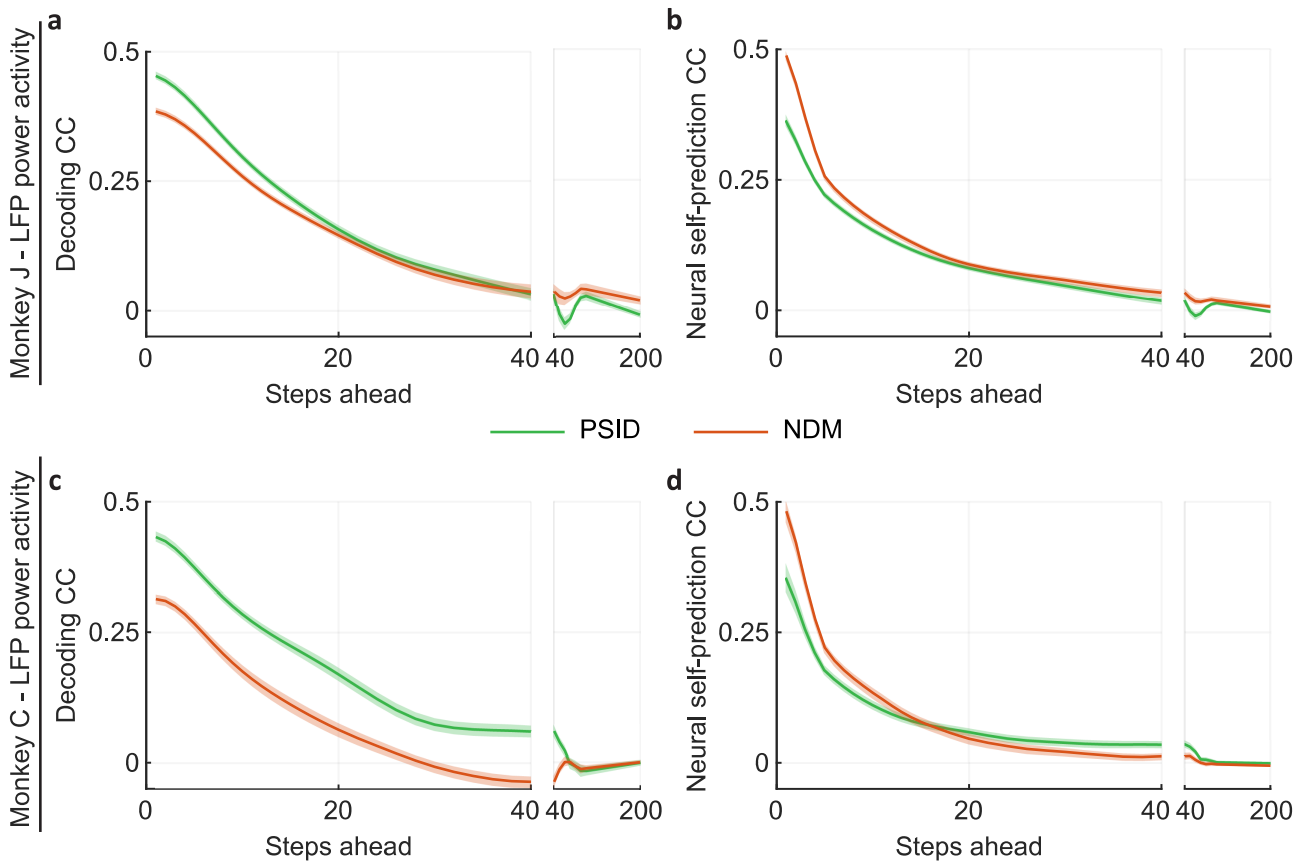
Supplementary Fig. 3 | Model structure parameters can be accurately identified using cross-validation.

(**a**) Detection of the total latent state dimension (n_x) using cross-validation is shown for the same 100 random simulated models as in Extended Data Fig. 2. We identify n_x by considering candidate values of n_x and selecting the value whose associated model reaches (within 1 s.e.m. of) the best neural *self-prediction* (predicting y_k using its past values) among all candidate values (Methods). The Pearson's correlation *P*-value between the true and identified values is shown on the plot. The colored line and shaded area are defined as in Supplementary Fig. 2a. n_x, which ranged from 1 to 10 across random models, was identified with no error in 98% of the models and with an average error of 0.040 $\pm$ 0.028 (mean $\pm$ s.e.m.). (**b**) Same as (a), for detection of n_x using cross-validation in standard SID. Using SID, n_x was identified with an average error of 0.08 $\pm$ 0.039. (**c**) The distribution of true and identified values of n_x from (a)-(b) is shown as a box plot. Bars and boxes are defined as in Fig. 3b. All data points are shown. (**d**) Same as (a), for detection of the PSID parameter n_1 (Methods). n_1, which ranged from 1 to 10 across random models, was identified with no error in 94% of the models and with an average error of 0.050 $\pm$ 0.021. (**e**) The distribution of true and identified values of n_1 from (d) is shown as a box plot. The true and identified n_x and n_1 are always integer values, so for better visualization and to avoid having multiple points at the exact same location on the plots a random small displacement has been added to each point. For all panels, *n* = 100 random models.



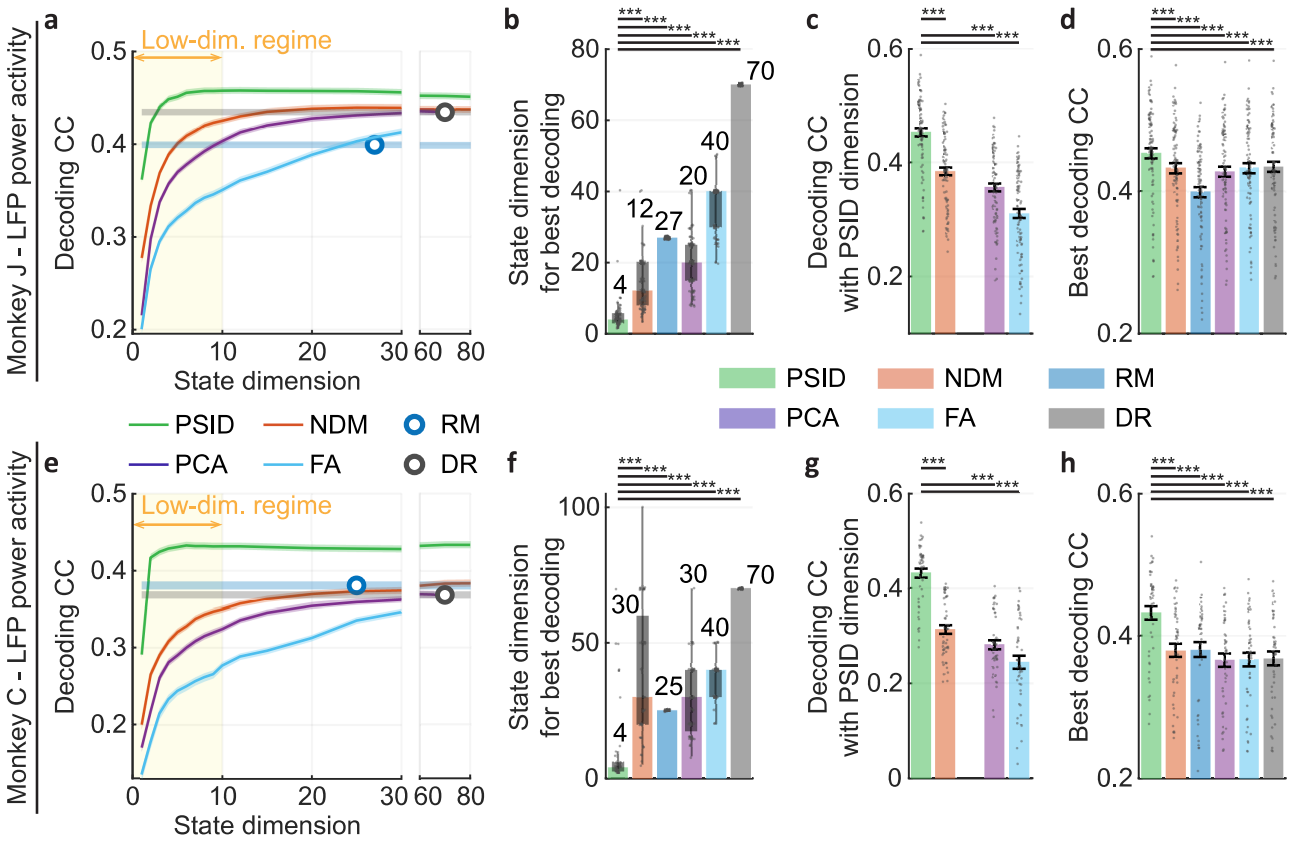
Supplementary Fig. 4 | PSID can accurately estimate the behaviorally relevant neural dynamics dimension, as well as the total neural dynamics dimension and the total behavior dynamics dimension in simulations.

(**a**) Cross-validated behavior decoding correlation coefficient (CC) as a function of latent state dimension using PSID and NDM within numerical simulations. Decoding CC is averaged across 100 random simulated models and the shaded area indicates the s.e.m. In each model, a random number of neural states were behaviorally irrelevant (Methods). (**b**) The behaviorally relevant neural dynamics dimension identified using PSID and NDM. This number is identified for each model as the smallest state dimension for which the CC reaches the best decoding performance. Bars, boxes and asterisks are defined as in Fig. 3b. While PSID accurately identifies the behaviorally relevant dynamics dimension (P = 0.19), NDM overestimates it ($P = 4 \times 10^{-15}$). Statistical tests are two-sided signed-rank tests. (**c**) One-step-ahead self-prediction of neural activity (cross-validated CC) as a function of latent state dimension. To compute the self-prediction, SID (i.e., PSID with $n_1 = 0$) is always used for modeling since dissociation of behaviorally relevant states is not needed. Figure convention is the same as in (a). (**d**) Same as (c) for one-step-ahead self-prediction of behavior. (**e**) True and identified values for behaviorally relevant neural dynamics dimension (PSID results from (b)), the total neural dynamics dimension (identified as the state dimension for best neural self-prediction from (c)) and the total behavior dynamics dimension (identified as the state dimension for best behavior self-prediction from (d)). Figure convention and statistical tests are the same as in (b). For all panels, n = 100 random models. These results confirm with numerical simulations that our approach for identifying the total neural and behavior dynamics dimensions correctly estimates these numbers, and that PSID accurately identifies the behaviorally relevant neural dynamics dimension from data. Consequently, the same cross-validation approach is used in Figs. 3 and 6, Extended Data Figs. 5-6 and 9-10, and Supplementary Figs. 5-9 and 11 for the real neural data to compute the dimensions.



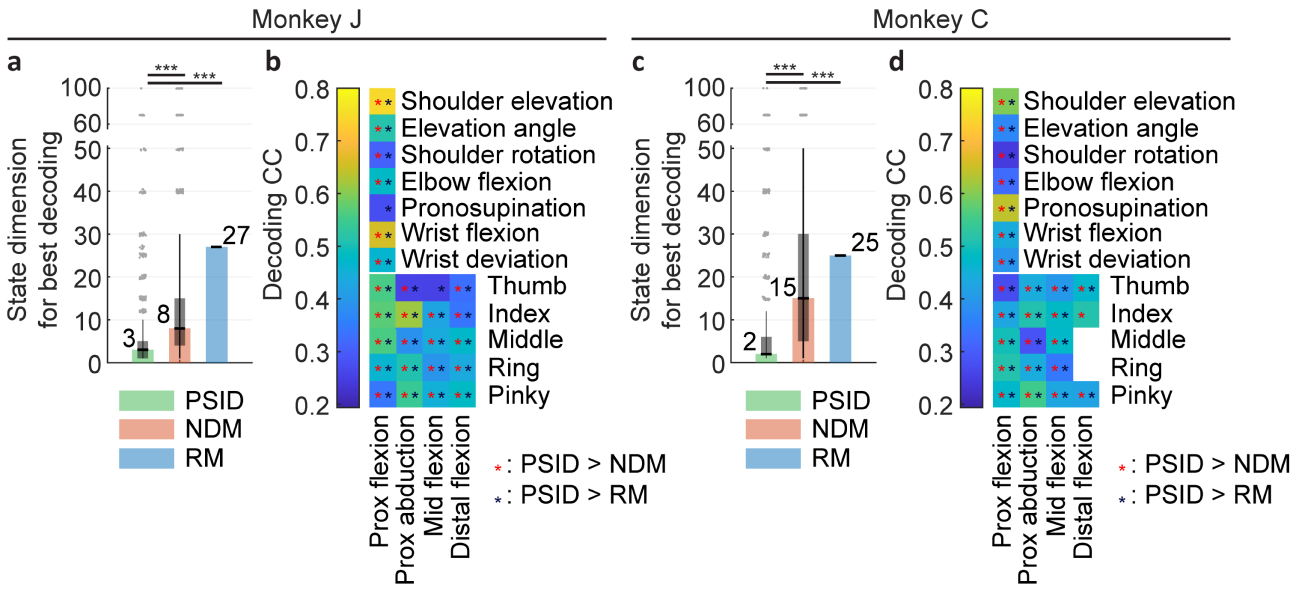
Supplementary Fig. 5 | PSID is more predictive of behavior than NDM for multiple-step-ahead prediction of behavior from neural activity.

(**a**) Decoding CC is shown for PSID and NDM when predicting the behavior multiple steps into the future from neural activity. Here n-step-ahead prediction of the latent state at time step k is obtained via a Kalman filter but with neural activity provided only up to time step $k-n$ (Methods). In running the Kalman filter, for time steps after $k-n$ that have no neural observation, the Kalman gain is equal to zero and the estimation of the latent state continues purely based on the previous latent state and the state transition matrix A in the model (Methods, Supplementary Note 5). Behavior at time step k is decoded via a regression from the predicted latent state at time step k with the parameter C_z. NDM used the same latent state dimension as PSID (as in Fig. 3c). Solid lines show the mean and the shaded areas show the s.e.m. (*n* = 91 datasets). (**b**) Same as (a), for neural self-prediction. Neural activity at time step k is predicted via a regression from the predicted latent state at time step k with the parameter C_y. As expected, at the same latent state dimension, PSID does better in behavior decoding and NDM does better in neural self-prediction, since NDM's only objective is to explain variance in neural activity regardless of relevance to behavior. However, the second stage of PSID allows it to also explain additional neural variance in higher latent state dimensions if desired (see Extended Data Fig. 5). (**c**)-(**d**) Same as (a)-(b), for monkey C (*n* = 48 datasets).



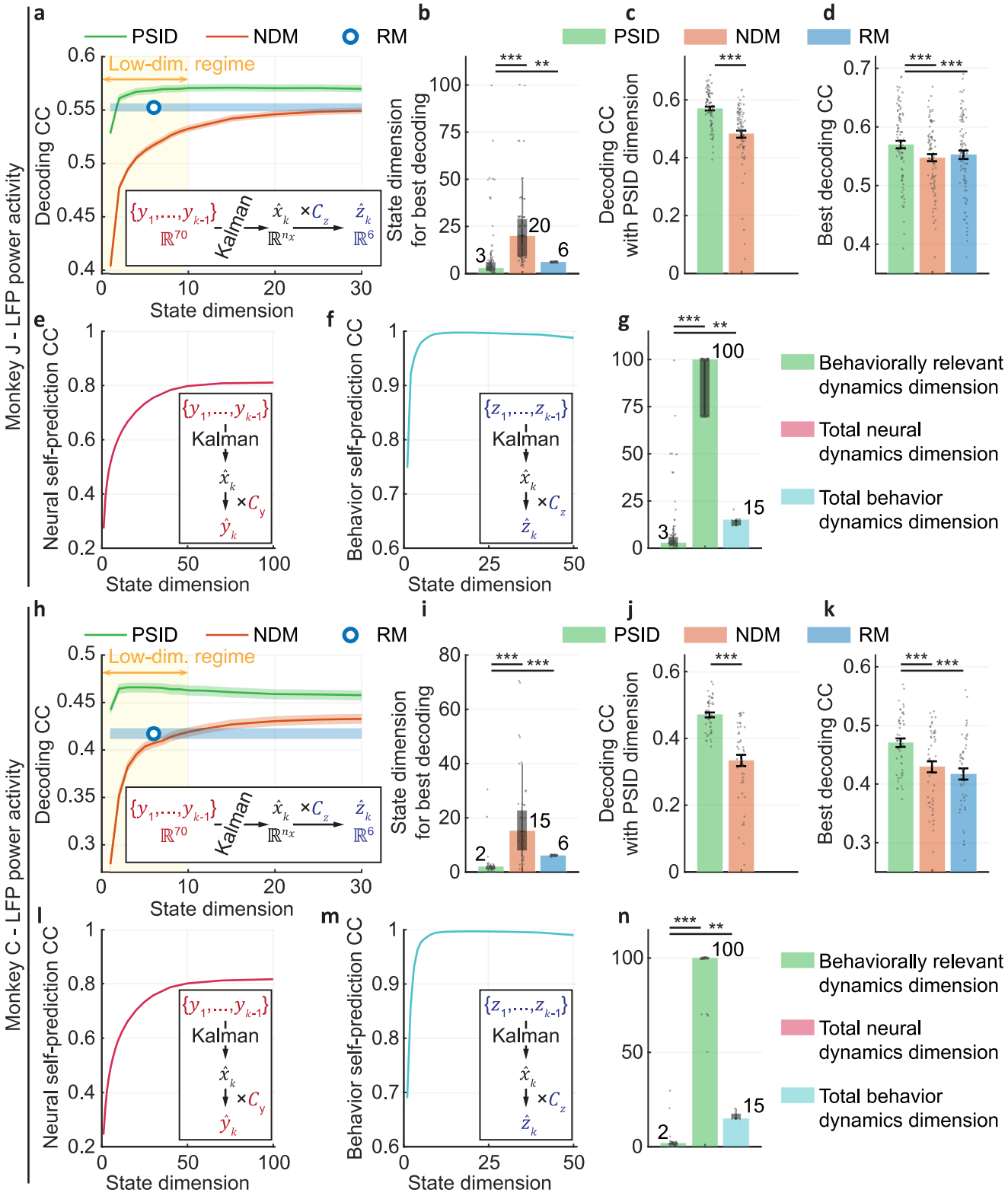
Supplementary Fig. 6 | PSID also reveals a markedly lower behaviorally relevant neural dimension and does better decoding compared with non-dynamic dimension reduction using principal component analysis (PCA), factor analysis (FA) and direct regression (DR) (i.e. PSID extracts the behaviorally relevant dynamics more accurately).

Figure convention is the same as in Fig. 3a-d, with non-dynamic PCA and FA dimension reduction, as well as DR without any dimension reduction added to the comparison. In PSID, NDM, and RM, temporal dynamics are modeled (equation (1)) and the decoder consists of a Kalman filter aggregating information from past neural activity to extract the state (allowing for denoising in time). In contrast, PCA, FA, and DR are non-dynamic methods that do not model temporal dynamics, thus their extraction of the state (i.e. principal components for PCA or factors for FA) involves a linear projection from neural activity only at the current time step. Note that in DR neural activity is used as the state itself, without dimensionality reduction. In all methods (dynamic or non-dynamic), the next step for decoding behavior after the extraction of states is the same and involves a linear combination of the extracted states to get the decoding of behavior. Compared with other methods, PSID revealed a significantly lower dimension for behaviorally relevant dynamics (panels b,f; $P < 10^{-5}$; one-sided signed-rank) and achieved significantly better decoding both at that dimension (panels c,g; $P < 10^{-8}$; one-sided signed-rank) and when other methods used much higher dimensions (panels d,h; $P < 10^{-8}$; one-sided signed-rank). n = 91 datasets in monkey J and n = 48 datasets in monkey C.



Supplementary Fig. 7 | The PSID-extracted latent states with markedly lower dimension achieve significantly better decoding of almost all joints in the upper-extremity.

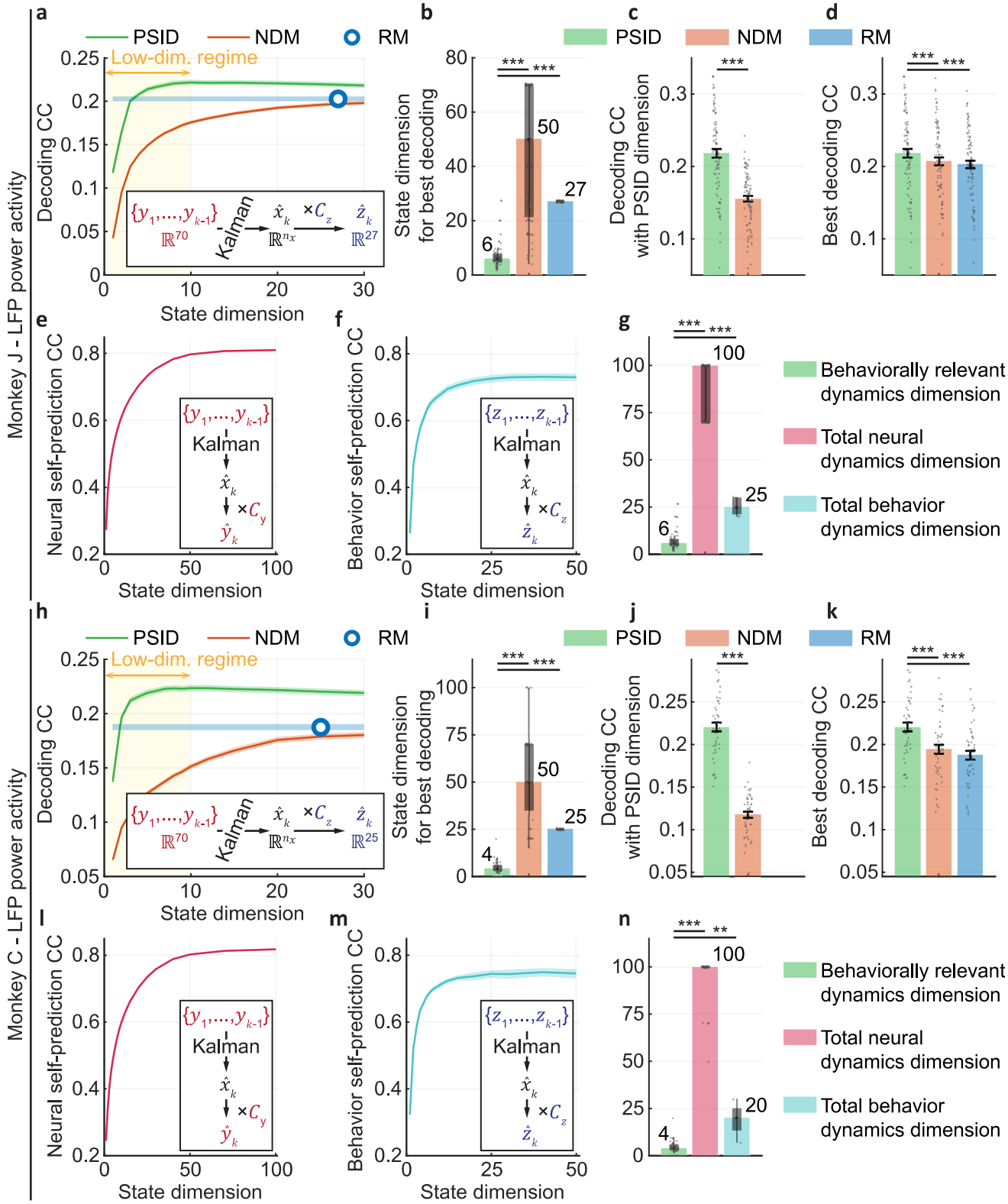
(**a**) The state dimension used by each method to achieve the best decoding for individual joints. For all methods, models are fitted to all joints as in Fig. 3. For PSID and NDM, models are fitted using various state dimensions; then for each joint, the latent state dimension is chosen to be the smallest value for which the decoding CC reaches within 1 s.e.m. of the best decoding CC possible for that joint among all latent state dimensions. Bars, boxes and asterisks are defined as in Fig. 3b. For better visualization of outliers, the vertical axis is broken. (**b**) Cross-validated correlation coefficient (CC) between the decoded and true joint angles is shown for PSID. Asterisks mark joints for which PSID results in significantly ($P < 0.05$) better decoding compared with NDM (red asterisk) or RM (dark blue asterisk). The latent state for each method is chosen as in (a). (**c**)-(**d**) Same as (a)-(b), for monkey C. In both monkeys, compared with NDM, PSID achieved better decoding for all individual joints with the difference being statistically significant in almost all joints (panels b,d; $P < 10^{-4}$ for all joints in monkey C and $P < 10^{-12}$ for 25 of 27 joints in monkey J; one-sided signed-rank test; $n = 240$ and $n = 455$ dataset test folds for monkeys C and J, respectively). Compared with RM, PSID achieved significantly better decoding for all 27 joints in monkey J (panel b; $P < 0.04$ for each joint; one-sided signed-rank; $n = 455$ dataset test folds) and for 24 of the 25 joints in monkey C (panel d; $P < 0.004$ for each joint; one-sided signed-rank; $n = 240$ dataset test folds), and similar decoding for 1 joint in monkey C ($P = 0.27$; two-sided signed-rank; $n = 240$ dataset test folds). The significantly better decoding in PSID was achieved using significantly lower-dimensional states compared with NDM and RM (panels a,c; $P < 10^{-90}$; one-sided signed-rank; $n = 2457$ and $n = 1200$ joint datasets in monkeys J and C, respectively). In degree units, the average root mean-squared error in decoding joint angles was 9.8 and 7.2 for monkeys J and C, respectively. Joints with larger activity ratio (quantified as std of motion divided by its range) during a given experiment session had better decoding ($P < 10^{-6}$; Spearman's rank correlation; $n \geq 100$ active joints across datasets) with active joints with at least 0.25 activity ratio (covering 12 and 11 joints in monkeys J and C, respectively) reaching an average CC of 0.63 and 0.56 and an explained variance (EV) of 0.41 and 0.32 for monkeys J and C, respectively.



Supplementary Fig. 8 | PSID again reveals a markedly lower dimension for behaviorally relevant neural dynamics in the motor cortex LFP activity when behavior is taken as the 3D end-point position (of hand and elbow) instead of the joint angles.

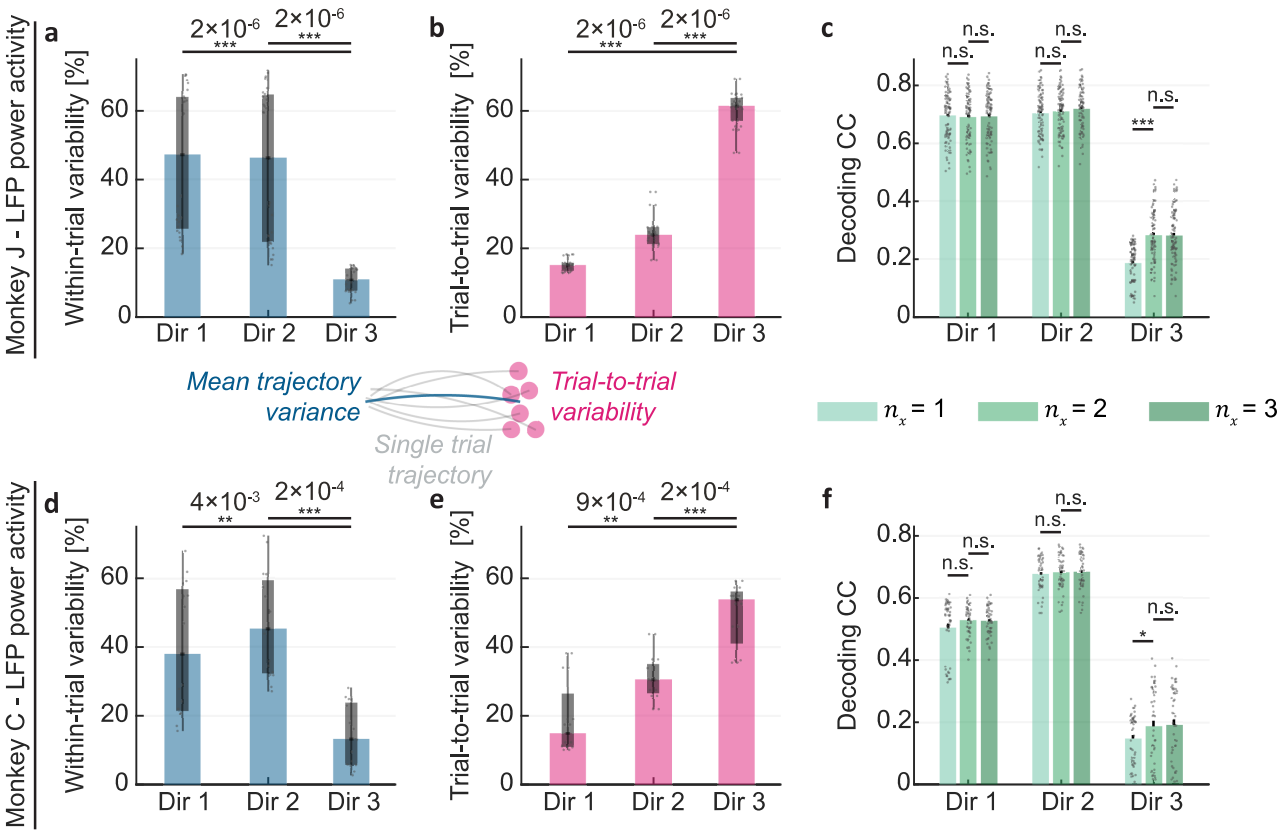
Figure convention and number of datasets in all panels is the same as in Fig. 3, but this time for behavior taken as the 3D position of hand and elbow (n_z = 6). Similar to the results for joint angles (Fig. 3), here, PSID again revealed a significantly lower dimension for behaviorally relevant neural dynamics compared with NDM for both monkeys (panels b,i; $P < 10^{-6}$; one-

sided signed-rank; $n \geq 48$) and achieved a significantly better decoding compared with NDM and RM even when they used much higher-dimensional states (panels c,d,j,k; $P < 10^{-8}$; one-sided signed-rank; $n \geq 48$). Moreover, in both monkeys, the dimension of behaviorally relevant neural dynamics revealed by PSID was again significantly smaller than the dimension of dynamics in the recorded neural activity ($P < 10^{-18}$; one-sided rank-sum) and in behavior ($P < 0.004$; one-sided rank-sum) as estimated based on their self-prediction (panels g,n). Statistical test details and number of samples are the same as those from Fig. 3 in Supplementary Table 1.



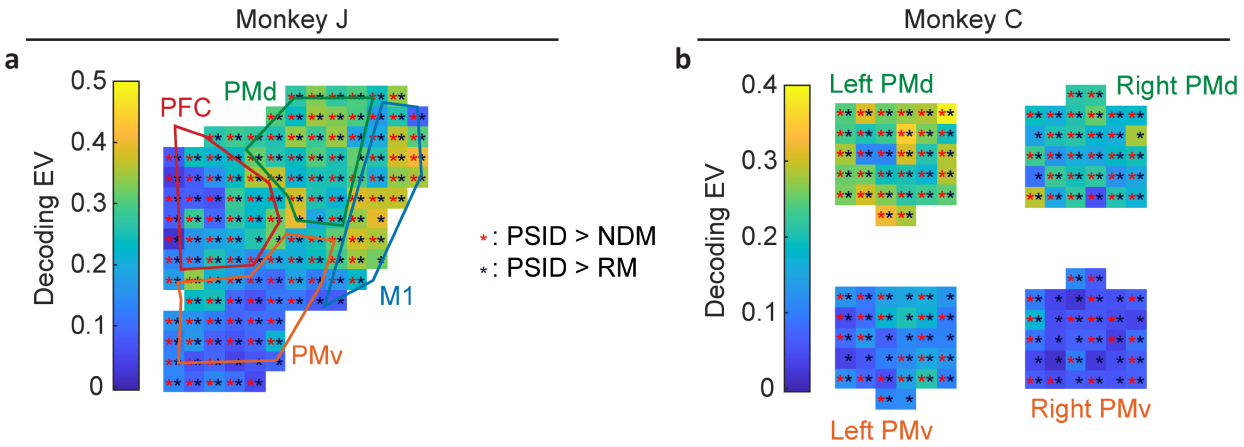
Supplementary Fig. 9 | PSID again reveals a markedly lower dimension for behaviorally relevant neural dynamics in the motor cortex LFP activity when behavior is taken as the angular velocity of joint angles instead of the joint angles.

Figure convention and number of datasets in all panels is the same as in Fig. 3, but this time for behavior taken as the first order derivative of joint angles (i.e. joint velocity). Statistics and conclusions are as in Supplementary Fig. 8.



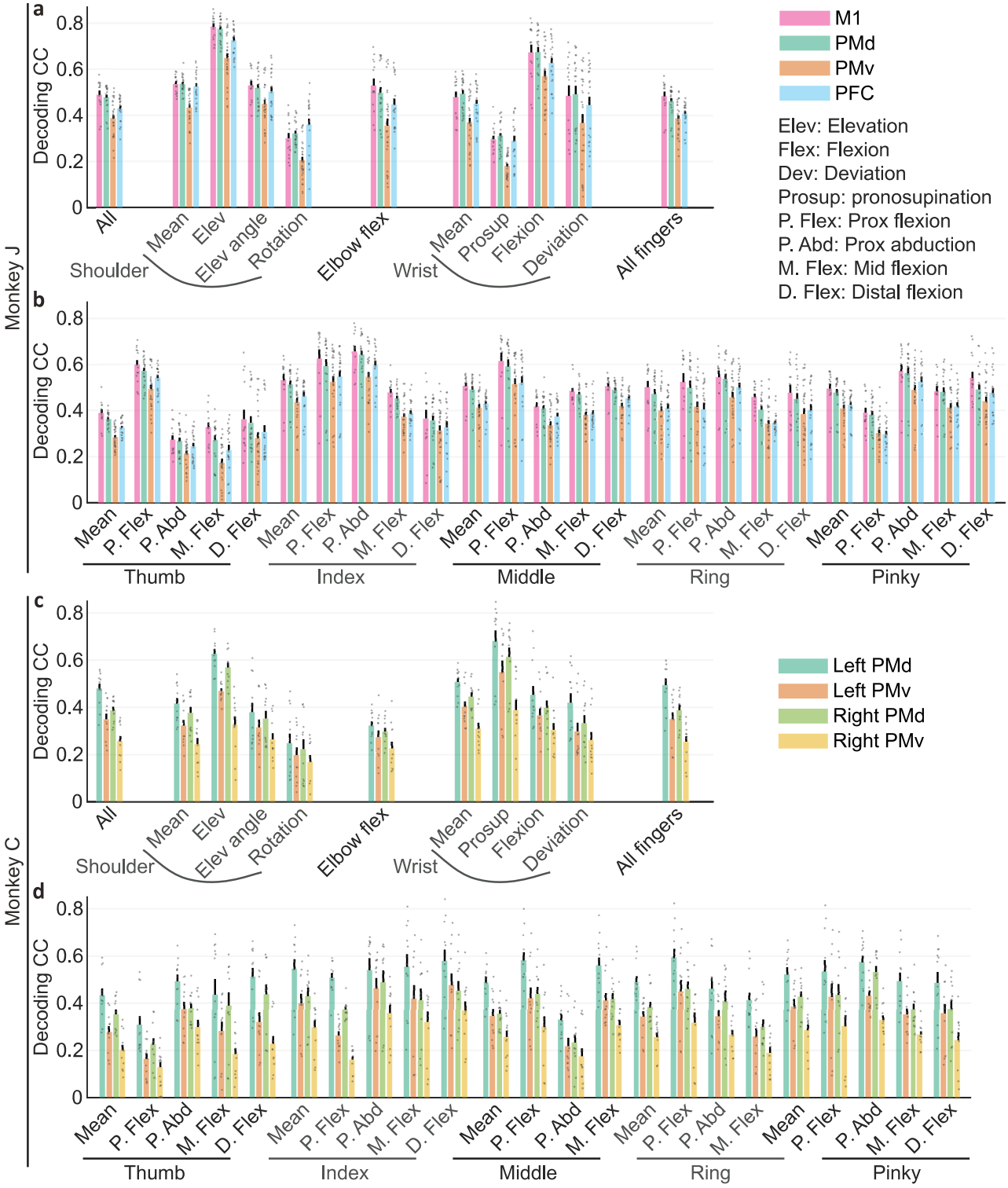
Supplementary Fig. 10 | PSID-extracted latent states capture both within-trial and trial-to-trial variability in behavior.

(**a**) Within-trial variability, quantified as the variance of behavior trajectory averaged separately across all reach and return epochs (square root of average variance in each movement direction (Dir)). Dir 1, 2 and 3 were depth, height and breadth from the perspective of the subject, respectively. Behavior is taken as the 3D end-point position of hand and elbow as in Supplementary Fig. 8. Bars and asterisks are defined in as in Fig. 3b. % indicates what percentage of total within-trial variability is due to a given direction. (**b**) Trial-to-trial variability, quantified as the s.d. of target position in each direction in the 3D physical space. % indicates what percentage of total trial-to-trial variability is due to a given direction. Figure convention is as in (a). (**c**) The PSID behavior decoding accuracy is shown averaged in each of the 3 movement directions for different neural latent state dimensions of n_x = 1, 2, 3. Bars and whiskers are defined as in Fig. 3d. (**d**)-(**f**) Same as (a)-(c) for monkey C. In both monkeys, out of the 3 directions in 3D movement space, in some directions the behavior showed relatively more within-trial variability and relatively less trial-to-trial variability (i.e. Dir 1 and 2, panels a,b,d,e; $P < 0.004$ with the exact values noted above asterisks in the plot; one-sided signed-rank; $n = 28$ and $n = 16$ end point datasets in monkeys J and C, respectively). The two movement directions that had relatively larger within-trial variability in behavior (i.e. Dir 1 and Dir 2) reached almost peak decoding accuracy with only a 1D latent state (panels c,f). Learning a model with a 2D latent state only significantly improved the decoding for the third movement direction that had relatively larger trial-to-trial variability in behavior (i.e. Dir 3, panels c,f; $P = 4\times10^{-12}$ and $P = 0.04$; one-sided rank-sum; $n = 91$ and $n = 48$ datasets in monkeys J and C, respectively). This suggests that compared with a 1D latent state, a 2D latent state encoded additional dynamics that were largely related to the trial-to-trial variability in behavior.



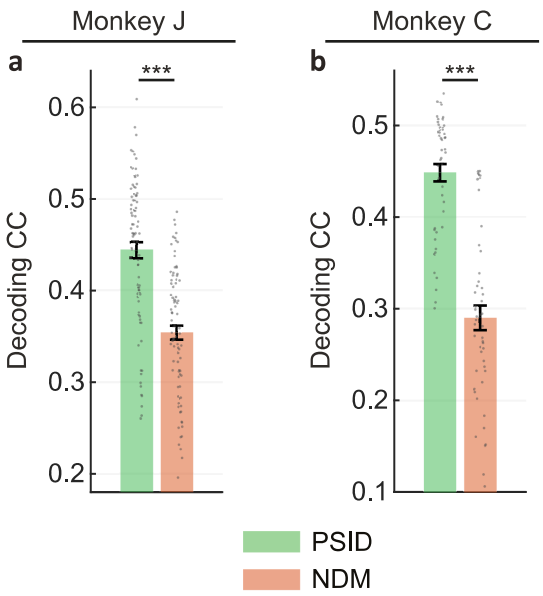
Supplementary Fig. 11 | PSID more accurately identified the behaviorally relevant neural dynamics in each recording channel across premotor, primary motor, and prefrontal areas.

(**a**) Same as Fig. 4, but instead showing EV as the performance measure in decoding of active joints using each recording channel separately. (**b**) Similar to (a) for channels in monkey C, which had bilateral coverage (Methods). For both monkeys, PSID resulted in significantly better decoding than NDM and RM for at least 80% and 100% of recordings channels, respectively ($P < 0.05$ for each channel; one-sided signed-rank; $n = 35$ test folds in monkey J and $n = 20$ test folds in monkey C) while also achieving similar decoding to NDM in the remaining at most 20% of channels ($P > 0.05$ for the same test). PSID achieved this decoding while using significantly lower state dimensions than NDM ($P = 10^{-108}$ for monkey J and $P = 10^{-68}$ for monkey C) and RM ($P = 10^{-156}$ for monkey J and $P = 10^{-83}$ for monkey C), where statistical tests are one-sided signed-rank ($n = 959$ and $n = 512$ channel datasets in monkeys J and C, respectively).



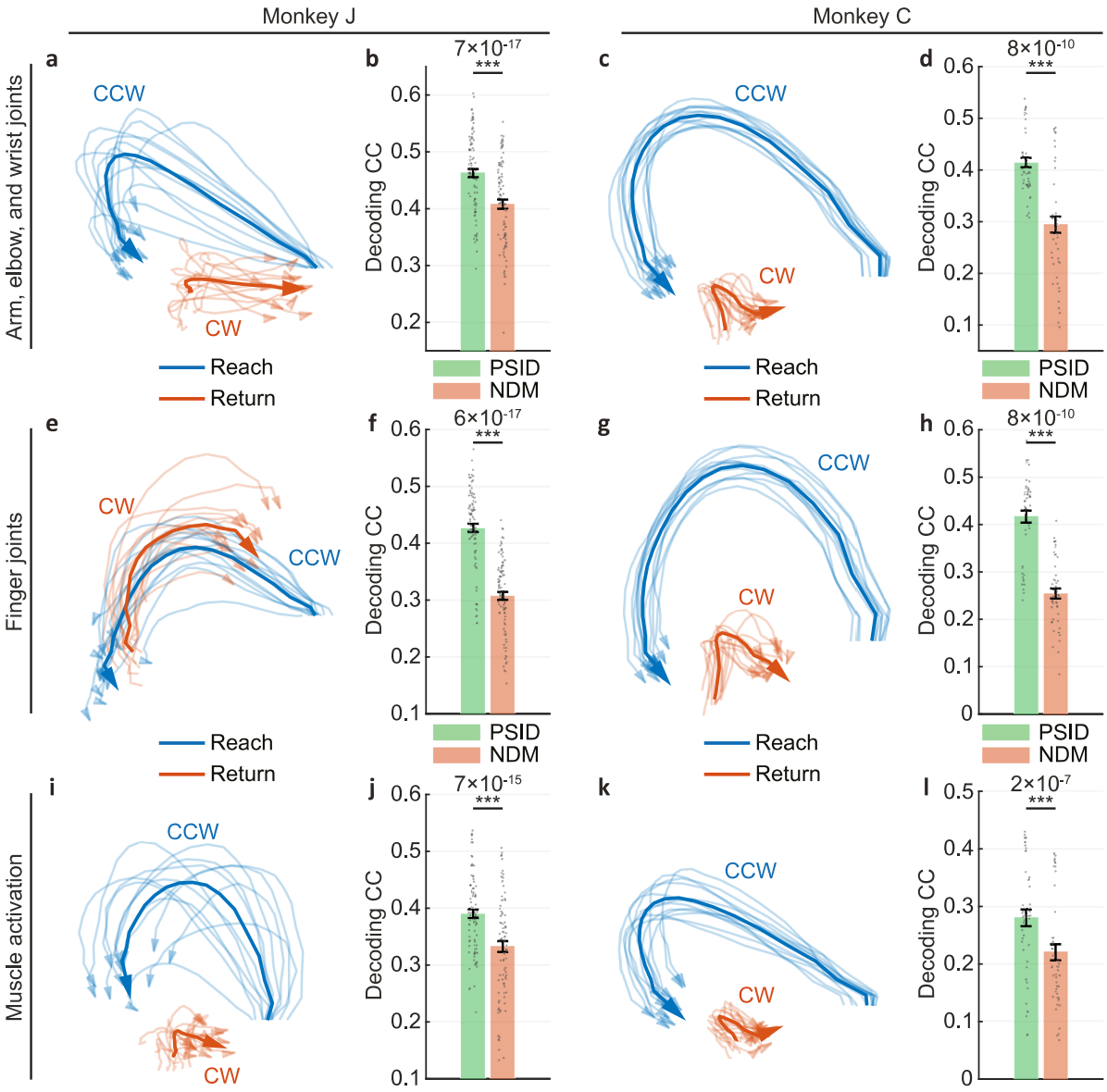
Supplementary Fig. 12 | Behaviorally relevant information varies across different brain regions in similar ways for different joints, with M1 and PMd consistently containing more information than PMv or PFC.

(**a**) Decoding accuracy using PSID is shown when channels within different anatomical regions are used for decoding. Decoding for each arm, elbow and wrist joint is shown, as well as average decoding over different joint groups. Bars and whiskers are defined as in Fig. 3d. *n* = 14, 21, 28, and 21 datasets for M1, PMd, PMv, and PFC, respectively. (**b**) Same as (a) for all finger joints separately and averaged within joints of each finger. (**c**)-(**d**) Same as (a)-(b) for monkey C (*n* = 12 datasets for each region). The ipsilateral (right) hemisphere in monkey C was also predictive of behavior, which is consistent with prior work[69,70].



Supplementary Fig. 13 | Even when nonlinear Support Vector Regression (SVR) is used to map the latent state to behavior, the rotational states extracted by PSID are more predictive of behavior than those extracted by standard NDM.

(**a**) Nonlinear SVR decoding for exactly the same rotational latent states shown in Fig. 5c,d. Figure convention is the same as in Fig. 5e. (**b**) Same as (a) for monkey C. PSID-extracted latent states were significantly more predictive of NDM-extracted latent states even when nonlinear mapping using SVR was allowed ($P = 6\times10^{-17}$ and $P = 8\times10^{-10}$; one-sided signed-rank; $n = 91$ and $n = 48$ datasets in monkeys J and C, respectively).

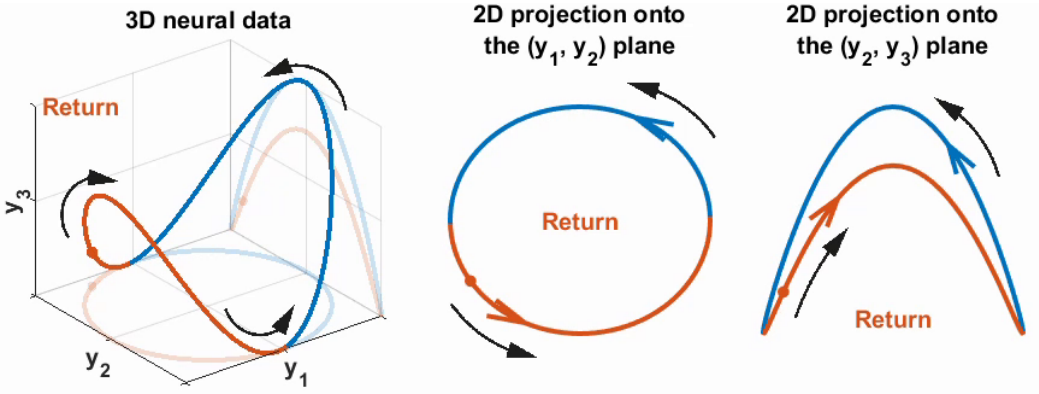


Supplementary Fig. 14 | Extraction of bidirectional rotational dynamics using PSID was robust to behavioral signals and held also when modeling different subsets of joints and simulated muscle activations.

(**a**) Average trajectory of 2D states identified by PSID during reach and return epochs, when only arm, elbow, and wrist joints are taken as the behavior. Figure convention is the same as in Fig. 5c. For standard NDM, extracted 2D states do not depend on the behavior being modeled and are always the same as those shown in Fig. 5d,g, regardless of the behavior that is being modeled. (**b**) Decoding accuracy of using PSID 2D states (from (a)) or NDM 2D states (from Fig. 5d) to decode behavior. Figure convention is the same as in Fig. 5e. (**c**)-(**d**) Same as (a)-(b) for monkey C. (**e**)-(**h**) Same as (a)-(d) for when only finger joints are taken as the behavior. (**i**)-(**l**) Same as (a)-(d) for when inferred muscle activations for upper and lower arm muscles (inferred from joint angles using OpenSim*) are taken as the behavior. For all behavioral signals, similar to the results in Fig. 5, PSID again extracted latent states that, unlike the latent states extracted using NDM, rotated in opposite directions during reach and return (panels a,c,e,g,i,k) and resulted in more accurate decoding of behavior (panels b,d,f,h,j,l; $P < 10^{-6}$ with the exact values noted above asterisks in the plot; one-sided signed-rank; n = 91 and n = 48 datasets for monkeys J and C, respectively).

*Delp, S. L. et al. OpenSim: Open-Source Software to Create and Analyze Dynamic Simulations of Movement. IEEE Trans. Biomed. Eng. 54, 1940–1950 (2007).

Supplementary Videos



Supplementary Video 1 | Visualization of how high-dimensional neural dynamics may contain 2D rotations both in the same and in opposite directions.

The presented simulation depicts a hypothetical scenario where 3 dimensions of neural activity traverse a manifold in 3D space of which different projections reveal rotations in the same or opposite directions during reach vs. return epochs. Among all projections, PSID can find the projection corresponding to the behaviorally relevant neural dynamics (e.g. here the (y_2, y_3) plane, if behavior is best predicted using the activity in this plane) whereas the standard behavior-agnostic NDM methods may find other projections (e.g. the (y_1, y_2) plane). A similar hypothetical manifold squeezed to varying degrees—resulting in 3 versions of the manifold—has been used in prior work to demonstrate the concept of tangling[11]. Here our goal is instead to demonstrate a distinct concept of how for the same traversal on exactly the same manifold in 3D, different 2D projections can show rotations in different directions—i.e. either rotations that keep the same direction or rotations that reverse their direction during the traversal. This observation is similar to how neural dynamics extracted by PSID in our motor dataset show different rotations compared to those extracted by NDM (Fig. 5). This result demonstrates the importance of PSID in performing dynamic dimensionality reduction while preserving behavior information, which in this example corresponds to which lower-dimensional 2D projection plane to pick for modeling the dynamics of the 3D neural activity. Here, for simplicity, projections are visualized as static but the same concept holds for PSID projections, which are dynamic—i.e. to get the projected latent variable at a given time, PSID can aggregate information not only from the same time step of neural activity but also from all the past neural activity. Moreover, unlike this simple hypothetical example, the dimension of the neural space is in general much higher than 3. Thus, discovering and modeling the low-dimensional dynamic projection that preserves behaviorally relevant dynamics is a major challenge because these dynamics are hidden within the overall high-dimensional neural space. PSID addresses this challenge by discovering these behaviorally relevant dynamics within the high-dimensional neural space—i.e. *where (which subspace)* they are in this high-dimensional dynamic space—, finding their *dimensionality*, and finally explicitly *modeling* their temporal evolution (i.e. dynamics).

Supplementary Tables

Supplementary Table 1 | Statistical test details.

SR: signed rank. RS: rank-sum. All tests are one-sided. EDF: Extended Data Fig.

Fig.	Test	*n*	*P*-values
3b	SR	91 datasets	9×10^{-15} (NDM), 5×10^{-17} (RM)
3c	SR	91 datasets	7×10^{-17} (NDM)
3d	SR	91 datasets	2×10^{-16} (NDM), 6×10^{-17} (RM)
3g	RS	91 neural datasets and 7 sessions	8×10^{-33} (neural), 5×10^{-6} (behavior)
3i	SR	48 datasets	2×10^{-8} (NDM), 4×10^{-6} (RM)
3j	SR	48 datasets	8×10^{-10} (NDM)
3k	SR	48 datasets	8×10^{-10} (NDM), 8×10^{-10} (RM)
3n	RS	48 neural datasets and 4 sessions	5×10^{-19} (neural), 3×10^{-3} (behavior)
4a	SR	959 channel datasets	10^{-124} (NDM), 3×10^{-155} (RM)
4b	SR	35 test folds	$P < 0.05$ for each marked significant channel
4c	SR	512 channel datasets	10^{-70} (NDM), 2×10^{-80} (RM)
4d	SR	20 test folds	$P < 0.05$ for each marked significant channel
5e	SR	91 datasets	6×10^{-17}
5h	SR	48 datasets	8×10^{-10}
6b	SR	28 datasets	7×10^{-6} (NDM), 8×10^{-4} (SVR), 2×10^{-6} (RNN), 2×10^{-6} (RM)
6c	SR	28 datasets	2×10^{-6} (NDM), 4×10^{-6} (SVR), 2×10^{-6} (RNN)
6d	SR	28 datasets	2×10^{-6} (NDM), 3×10^{-6} (SVR), 4×10^{-5} (RNN), 2×10^{-6} (RM)
6f	SR	16 datasets	2×10^{-4} (NDM), 10^{-3} (SVR), 4×10^{-3} (RNN), 2×10^{-4} (RM)
6g	SR	16 datasets	2×10^{-4} (NDM), 2×10^{-4} (SVR), 2×10^{-4} (RNN)
6h	SR	16 datasets	2×10^{-4} (NDM), 4×10^{-4} (SVR), 4×10^{-4} (RNN), 2×10^{-4} (RM)
EDF 9b	SR	28 datasets	2×10^{-6}, 7×10^{-6}, 2×10^{-6}, 3×10^{-6} (RNNs, from left to right)
EDF 9c	SR	28 datasets	2×10^{-6}, 2×10^{-6}, 2×10^{-6}, 3×10^{-6} (RNNs, from left to right)
EDF 9d	SR	28 datasets	4×10^{-5}, 3×10^{-6}, 2×10^{-2}, 3×10^{-2} (RNNs, from left to right)
EDF 9f	SR	16 datasets	4×10^{-3}, 6×10^{-5}, 3×10^{-4}, 6×10^{-5} (RNNs, from left to right)
EDF 9g	SR	16 datasets	2×10^{-4}, 2×10^{-4}, 2×10^{-4}, 2×10^{-4} (RNNs, from left to right)
EDF 9h	SR	16 datasets	4×10^{-4}, 2×10^{-4}, 3×10^{-4}, 4×10^{-4} (RNNs, from left to right)

Supplementary Notes

Supplementary Note 1 | Preferential subspace identification (PSID) algorithm

Here we provide the step by step description of PSID. For a visualization see Extended Data Fig. 1 and for the detailed derivation see Supplementary Note 6. Given the training time-series $\{y_k: 0 \leq k < N, y_k \in \mathbb{R}^{n_y}\}$ and $\{z_k: 0 \leq k < N, z_k \in \mathbb{R}^{n_z}\}$, state dimension n_x and parameters $n_1 \leq n_x$ (number of states extracted in the first stage) and $i \geq 2$ (projection horizon), the PSID algorithm identifies parameters of a general dynamic linear state-space model as in equation (4), using the following steps.

1. Form the following matrices (visualized in Extended Data Fig. 1a for the simplified case of $i = 1$):

$$\begin{bmatrix} Y_p \\ \hline Y_f \end{bmatrix} \triangleq \left[\begin{array}{cccc} y_0 & y_1 & \cdots & y_{j-1} \\ y_1 & y_2 & \cdots & y_j \\ \vdots & \vdots & \ddots & \vdots \\ y_{i-1} & y_i & \cdots & y_{j+i-1} \\ \hline y_i & y_{i+1} & \cdots & y_{j+i} \\ y_{i+1} & y_{i+2} & \cdots & y_{j+i+1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{2i-1} & y_{2i} & \cdots & y_{j+2i-1} \end{array}\right] = \left[\begin{array}{cccc} y_0 & y_1 & \cdots & y_{j-1} \\ y_1 & y_2 & \cdots & y_j \\ \vdots & \vdots & \ddots & \vdots \\ y_{i-1} & y_i & \cdots & y_{j+i-1} \\ y_i & y_{i+1} & \cdots & y_{j+i} \\ \hline y_{i+1} & y_{i+2} & \cdots & y_{j+i+1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{2i-1} & y_{2i} & \cdots & y_{j+2i-1} \end{array}\right] \triangleq \begin{bmatrix} Y_p^+ \\ \hline Y_f^- \end{bmatrix} \triangleq \begin{bmatrix} Y_p \\ \hline Y_i \\ \hline Y_f^- \end{bmatrix} \tag{8}$$

$$\begin{bmatrix} Z_p \\ \hline Z_f \end{bmatrix} \triangleq \left[\begin{array}{cccc} z_0 & z_1 & \cdots & z_{j-1} \\ z_1 & z_2 & \cdots & z_j \\ \vdots & \vdots & \ddots & \vdots \\ z_{i-1} & z_i & \cdots & z_{j+i-1} \\ \hline z_i & z_{i+1} & \cdots & z_{j+i} \\ z_{i+1} & z_{i+2} & \cdots & z_{j+i+1} \\ \vdots & \vdots & \ddots & \vdots \\ z_{2i-1} & z_{2i} & \cdots & z_{j+2i-1} \end{array}\right] = \left[\begin{array}{cccc} z_0 & z_1 & \cdots & z_{j-1} \\ z_1 & z_2 & \cdots & z_j \\ \vdots & \vdots & \ddots & \vdots \\ z_{i-1} & z_i & \cdots & z_{j+i-1} \\ z_i & z_{i+1} & \cdots & z_{j+i} \\ \hline z_{i+1} & z_{i+2} & \cdots & z_{j+i+1} \\ \vdots & \vdots & \ddots & \vdots \\ z_{2i-1} & z_{2i} & \cdots & z_{j+2i-1} \end{array}\right] \triangleq \begin{bmatrix} Z_p^+ \\ \hline Z_f^- \end{bmatrix} \triangleq \begin{bmatrix} Z_p \\ \hline Z_i \\ \hline Z_f^- \end{bmatrix} \tag{9}$$

   Number of rows in the matrices defined above are as follows:

   Y_p and Y_f have $i \times n_y$ rows. Y_p^+ has $(i+1) \times n_y$ rows. Y_f^- has $(i-1) \times n_y$ rows. Y_i has n_y rows.

   Z_p and Z_f have $i \times n_z$ rows. Z_p^+ has $(i+1) \times n_z$ rows. Z_f^- has $(i-1) \times n_z$ rows. Z_i has n_z rows.

   Number of columns in all these matrices is $j = N - 2i + 1$.

2. If $n_1 = 0$ (no behaviorally relevant latent states), skip to step 9

3. [Begins stage 1 of PSID]: Compute the least squares prediction of Z_f from Y_p, and Z_f^- from Y_p^+ as:

$$\hat{Z}_f = Z_f Y_p^T \left(Y_p Y_p^T\right)^{-1} Y_p \tag{10}$$

$$\hat{Z}_f^- = Z_f^- Y_p^{+T} \left(Y_p^+ Y_p^{+T}\right)^{-1} Y_p^+ \tag{11}$$

4. Compute the singular value decomposition (SVD) of $\hat{Z}_f$ and keep the top n_1 singular values:

$$\hat{Z}_f = USV^T \cong U_1 S_1 V_1^T \tag{12}$$

5. Compute the behavior observability matrix $\Gamma_{z_i}^{(1)}$ and the behaviorally relevant latent state $\hat{X}_i^{(1)}$ as (.† denotes pseudoinverse):

$$\Gamma_{z_i}^{(1)} = U_1 S_1^{\frac{1}{2}} \tag{13}$$

$$\hat{X}_i^{(1)} = \Gamma_{z_i}^{\dagger} \hat{Z}_f \tag{14}$$

6. Remove the last n_z rows of $\Gamma_{z_i}^{(1)}$ to get $\Gamma_{z_{i-1}}^{(1)}$ and then compute the behaviorally relevant latent state at the next time step ($\hat{X}_{i+1}^{(1)}$) as:

$$\Gamma_{z_{i-1}}^{(1)} = \Gamma_{z_i}^{(1)}{}_{(1:(i-1)\times n_z,:)} \tag{15}$$

$$\hat{X}_{i+1}^{(1)} = \Gamma_{z_{i-1}}^{(1)\,\dagger} \hat{Z}_f^- \tag{16}$$

7. Compute the least squares solution for A_{11} using the latent states as:

$$A_{11} = \hat{X}_{i+1}^{(1)} \hat{X}_i^{(1)\dagger} \tag{17}$$

8. If $n_x = n_1$ (no additional states), set $A = A_{11}$, $\hat{X}_i = \hat{X}_i^{(1)}$ and $\hat{X}_{i+1} = \hat{X}_{i+1}^{(1)}$ and skip to step 17

9. [Begins stage 2 of PSID]: If $n_1 > 0$, find the neural observability matrix $\Gamma_{y_i}^{(1)}$ for $\hat{X}_i^{(1)}$ as the least squares solution of predicting Y_f using $\hat{X}_i^{(1)}$, and subtract this prediction from Y_f (otherwise set $Y_f' = Y_f$).

$$\Gamma_{y_i}^{(1)} = Y_f \hat{X}_i^{(1)T} \left(\hat{X}_i^{(1)} \hat{X}_i^{(1)T}\right)^{-1} \tag{18}$$

$$Y_f' = Y_f - \Gamma_{y_i}^{(1)} \hat{X}_i^{(1)} \tag{19}$$

10. If $n_1 > 0$, remove the last n_y rows of $\Gamma_{y_i}^{(1)}$ to find the neural observability matrix for $\hat{X}_{i+1}^{(1)}$ and subtract the corresponding prediction from Y_f^- (otherwise if $n_1 = 0$, set $Y_f^{-\prime} = Y_f^-$).

$$\Gamma_{y_{i-1}}^{(1)} = \Gamma_{y_i}^{(1)}{}_{(1:(i-1)\times n_y,:)} \tag{20}$$

$$Y_f^{-\prime} = Y_f^- - \Gamma_{y_{i-1}}^{(1)} \hat{X}_{i+1}^{(1)} \tag{21}$$

11. Compute the least squares prediction of Y_f' from Y_p, and $Y_f^{-\prime}$ from Y_p^+ as:

$$\hat{Y}_f' = Y_f' Y_p^T \left(Y_p Y_p^T\right)^{-1} Y_p \tag{22}$$

$$\hat{Y}_f^{-\prime} = Y_f^{-\prime} Y_p^{+T} \left(Y_p^+ Y_p^{+T}\right)^{-1} Y_p^+ \tag{23}$$

12. Compute the SVD of $\hat{Y}_f'$ and keep the top $n_2 = n_x - n_1$ singular values:

$$\hat{Y}_f' = U' S' V'^T \cong U_2 S_2 V_2^T \tag{24}$$

13. Compute the remaining neural observability matrix $\Gamma_{y_i}^{(2)}$ and the corresponding latent state $\hat{X}_i^{(2)}$ as:

$$\Gamma_{y_i}^{(2)} = U_2 S_2^{\frac{1}{2}} \tag{25}$$

$$\hat{X}_i^{(2)} = \Gamma_{y_i}^{\dagger} \hat{Y}_f' \tag{26}$$

14. Remove the last n_y rows of $\Gamma_{y_i}^{(2)}$ to get $\Gamma_{y_{i-1}}^{(2)}$ and then compute the remaining latent states at the next time step ($\hat{X}_{i+1}^{(2)}$) as:

$$\Gamma_{y_{i-1}}^{(2)} = \Gamma_{y_i}^{(2)}{}_{(1:(i-1)\times n_y,:)} \tag{27}$$

$$\hat{X}_{i+1}^{(2)} = \Gamma_{y_{i-1}}^{(2)}{}^{\dagger} \hat{Y}_f^{-\prime} \tag{28}$$

15. If $n_1 > 0$, concatenate $\hat{X}_i^{(2)}$ to $\hat{X}_i^{(1)}$ and $\hat{X}_{i+1}^{(2)}$ to $\hat{X}_{i+1}^{(1)}$ to get the full latent state (otherwise set $\hat{X}_i = \hat{X}_i^{(2)}$ and $\hat{X}_{i+1} = \hat{X}_{i+1}^{(2)}$):

$$\hat{X}_i = \begin{bmatrix} \hat{X}_i^{(1)} \\ \hat{X}_i^{(2)} \end{bmatrix}, \quad \hat{X}_{i+1} = \begin{bmatrix} \hat{X}_{i+1}^{(1)} \\ \hat{X}_{i+1}^{(2)} \end{bmatrix} \tag{29}$$

16. Compute the least squares solution for A_{21} and A_{22} using the latent states and form the full A as:

$$[A_{21} \quad A_{22}] = \hat{X}_{i+1}^{(2)} \hat{X}_i^{\dagger} \tag{30}$$

$$A = \begin{bmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{bmatrix} \tag{31}$$

17. Compute the least squares solution for C_y and C_z using the latent states and the observations as:

$$C_y = Y_i \hat{X}_i^{\dagger} \tag{32}$$

$$C_z = Z_i \hat{X}_i^{\dagger} \tag{33}$$

18. Compute the residuals as:

$$\begin{bmatrix} W_i \\ V_i \end{bmatrix} = \begin{bmatrix} \hat{X}_{i+1} \\ Y_i \end{bmatrix} - \begin{bmatrix} A \\ C_y \end{bmatrix} \hat{X}_i \tag{34}$$

19. Compute the noise statistics as the sample covariance of the residuals:

$$\begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} = \frac{1}{j} \begin{bmatrix} W_i \\ V_i \end{bmatrix} \begin{bmatrix} W_i \\ V_i \end{bmatrix}^T \tag{35}$$

20. Solve equation (47) to find the steady-state solution $\tilde{P}$, and substitute $\tilde{P}$ in equation (46) to get the steady-state Kalman gain K (equations in Supplementary Note 5).

21. If parameters Σ_y and G_y are of interest, solve the Lyapunov equation (37) to get Σ_x, and then use equations (38) and (39) to compute Σ_y and G_y, respectively (equations in Supplementary Note 3). These parameters are not needed for Kalman filtering or for decoding behavior from neural activity (equation (45) in Supplementary Note 5).

Supplementary Note 2 | The distinction between primary and secondary signals

To clarify the difference between the signals y_k and z_k in equation (2), it is worth noting that in the formulation of equation (2), y_k is taken as the primary signal in the sense that the latent state x_k describes the complete dynamics of y_k that also includes its shared dynamics with the secondary signal z_k. The designation of the primary and secondary signals (e.g. taking y_k to be the neural activity and z_k to be the behavior or vice versa) is interchangeable as far as the shared dynamics of the two signals are of interest and the choice of the primary signal only determines which signal's dynamics are fully described beyond the shared dynamics. In this work we take the primary signal y_k to be the neural activity and the secondary signal z_k to be the behavior. This is motivated by the typical scenario in

neuroscience and neuroengineering where the neural activity is often considered the primary signal and the goal is to learn how behavior is encoded in it or to decode behavior from it.

The term $C_z\, x_k^s$ in equation (2), which we refer to as

$$z_{1_k} = C_z\, x_k^s, \tag{36}$$

represents the part of the secondary signal z_k that is contributed by x_k^s and thus shared with the primary signal. Any additional dynamics of the secondary signal that are not shared with the primary signal are modeled as the general independent signal ϵ_k. If modeling these dynamics of the secondary signal is also of interest, after learning the parameters of equation (2), one could use the model to estimate z_{1_k} (Supplementary Note 5) and thus ϵ_k (as $\epsilon_k = z_k - z_{1_k}$) in the training data and then use standard dynamic modeling techniques (e.g. SID) to characterize the dynamics of ϵ_k in terms of another latent state-space model. But since these dynamics are independent of y_k, such characterization would not be helpful in describing the encoding of z_k in y_k or in decoding of z_k from y_k and thus we will not discuss their identification, and only discuss their generation in our numerical simulations (Supplementary Note 8).

Supplementary Note 3 | Equivalent sets of parameters that can fully describe the model

We define $G_y \triangleq \boldsymbol{E}\{x_{k+1}^s y_k^T\}$ specifying the cross-covariance of y_k with the state at the next time step, $\Sigma_x \triangleq \boldsymbol{E}\{x_k^s x_k^{s^T}\}$ specifying the covariance of x_k^s and $\Sigma_y \triangleq \boldsymbol{E}\{y_k y_k^T\}$ specifying the covariance of y_k. From equation (2), it is straight forward to show that these covariances are related to the model noise statistics (equation (3)) via

$$\Sigma_x = A\Sigma_x A^T + Q \tag{37}$$

$$\Sigma_y = C_y \Sigma_x C_y^T + R \tag{38}$$

$$G_y = A\Sigma_x C_y^T + S \tag{39}$$

where equation (37) is also known as the Lyapunov equation[35,62]. The Lyapunov equation (37) has a unique solution for Σ_x if A is stable (i.e. the absolute value of all its eigenvalues are less than 1)[62]. For stable systems (models with a

stable A), it is clear from equations (37)-(39) that there is a one to one relation between the set of parameters $(A, C_y, C_z, G_y, \Sigma_y, \Sigma_x)$ and the set (A, C_y, C_z, Q, R, S), and thus both sets can be used to describe the model in equation (2).

Equation (2) is known as the forward stochastic formulation for a linear state-space model. Given that only y_k and z_k are measurable real quantities and that the stochastic latent state x_k^s is not directly accessible, equation (2) is called an *internal* description for the signals y_k and z_k[62]. This internal description is not unique and a family of infinitely many models with different x_k^s can describe the same y_k and z_k. For example, any non-singular matrix T' can transform equation (2) to an equivalent model with $x_{k_{new}}^s = T' x_k^s$, a process known as a similarity transform (or a change of basis). Moreover, Faurre's stochastic realization problem shows that even beyond similarity transforms, there are non-unique sets of noise statistics (Q, R, and S) that give the exact same second order statistics for y_k[35,62]. The unique and complete *external* description for y_k and z_k consists of their second order statistics. Thus, in the model learning problem, all models that give the correct external description are equally valid solutions. The set of parameters $(A, C_y, C_z, G_y, \Sigma_y, \Sigma_x)$ are thus more suitable (compared with the equivalent set of parameters (A, C_y, C_z, Q, R, S)) for evaluating model learning because among this set, all parameters other than Σ_x are uniquely determined from second order statistics of y_k and z_k, up to within a similarity transform[35,62].

Supplementary Note 4 | Equivalent model formulation with behaviorally relevant states separated from the other states giving rise to equation (4)

Given the second order statistics of y_k (its auto-covariances at all possible time differences, see equation (52)), any set of parameters for equation (2) that would describe how the same second order statistics could be generated from a latent state x_k^s is known as a realization for y_k[62]. We can rewrite equation (2) in an equivalent realization in which the behaviorally relevant states are clearly separated from the others. To do this, without loss of generality, we first assume that equation (2) is written as a minimal realization of y_k, defined as a realization with the smallest

possible state dimension n_x[62]. For such a minimal realization, it can be shown that the pair (A, C_y) is observable and the pair (A, G_y) is reachable (Theorem 3.12 from ref. 62). Equivalently, both the neural observability matrix

$$\Gamma_y = \begin{bmatrix} C_y \\ C_y A \\ \vdots \\ C_y A^{n_x-1} \end{bmatrix} \tag{40}$$

and the neural reachability matrix

$$\Delta_y = \begin{bmatrix} G_y & AG_y & \dots & A^{n_x-1}G_y \end{bmatrix} \tag{41}$$

are full rank with rank of n_x (Theorems 3.4 and 3.7 from ref. 62).

Since not all latent states that contribute to the neural activity are expected to also contribute to a specific behavior of interest (equations (2) and (36)), the pair (A, C_z) is not necessarily observable (i.e. it may not be possible to uniquely infer the full latent state x_k^s only from behavioral observations z_k). In other words, the behavior observability matrix

$$\Gamma_z = \begin{bmatrix} C_z \\ C_z A \\ \vdots \\ C_z A^{n_x-1} \end{bmatrix} \tag{42}$$

may not be full rank. We define $n_1 = \text{rank}(\Gamma_z)$ as the number of latent states that drive behavior because as we show next, the latent state x_k^s can be separated into two parts in a way that only n_1 dimensions contribute to the behavior z_k. We can show, by applying Theorem 3.6 from ref. 62 to the first and third rows of equation (2), that if $n_1 < n_x$, there exists a nonsingular matrix T' that via the similarity transform

$$\begin{bmatrix} x_k^{(1)} \\ x_k^{(2)} \end{bmatrix} = x_k = T' x_k^s \tag{43}$$

gives equation (4) as an equivalent formulation for equation (2). Finally, replacing the A and C_z matrices in equation (42) with their values in equation (4) gives

$$\Gamma_z = \begin{bmatrix} C_{z_1} & 0 \\ C_{z_1}A_{11} & 0 \\ \vdots & 0 \\ C_{z_1}A_{11}^{n_x-1} & 0 \end{bmatrix} = \begin{bmatrix} \Gamma_z^{(1)} & 0 \end{bmatrix}, \tag{44}$$

showing that in the formulation of equation (4), the behavior observability matrix Γ_z has exactly n_1 non-zero columns (i.e. its first n_1 columns, denoted by $\Gamma_z^{(1)}$ in equation (44)), which is consistent with its rank being n_1.

Supplementary Note 5 | Kalman filtering and the equivalent forward innovation formulation

Given the linear state-space formulation of equation (2), it can be shown that the best prediction of y_{k+1} using y_1 to y_k (denoted as $\hat{y}_{k+1|k}$) in the sense of having the minimum mean-square error, and similarly the best prediction of z_{k+1} using y_1 to y_k (denoted as $\hat{z}_{k+1|k}$) are obtained with the well-known recursive Kalman filter[62], which can be written as

$$\begin{cases} \hat{x}_{k+1|k} = A\,\hat{x}_{k|k-1} + K_k\left(y_k - C\hat{x}_{k|k-1}\right) \\ \hat{y}_{k+1|k} = C_y\hat{x}_{k+1|k} \\ \hat{z}_{k+1|k} = C_z\,\hat{x}_{k+1|k} \end{cases} \tag{45}$$

where the recursion is initialized with $\hat{x}_{0|-1} = 0$ and K_k is the Kalman gain[62] equal to

$$K_k = \left(A\tilde{P}_{k|k-1}C_y^T + S\right)\left(C_y\tilde{P}_{k|k-1}C_y^T + R\right)^{-1}. \tag{46}$$

Here $\tilde{P}_{k|k-1}$ is the covariance of the error for one-step-ahead prediction of the state (i.e. covariance of $\tilde{x}_{k|k-1} = \hat{x}_{k|k-1} - x_k$) and can be computed via the recursive Riccati equation

$$\tilde{P}_{k+1|k} = A\tilde{P}_{k|k-1}A^T + Q - \left(A\tilde{P}_{k|k-1}C_y^T + S\right)\left(C_y\tilde{P}_{k|k-1}C_y^T + R\right)^{-1}\left(A\tilde{P}_{k|k-1}C_y^T + S\right)^T \tag{47}$$

with the recursion initialized with $P_{0|-1} = R_y$. The steady-state solution of Riccati equation can be obtained by replacing $\tilde{P}_{k+1|k}$ with $\tilde{P}_{k|k-1}$ in the equation and solving for $\tilde{P}_{k|k-1}$. We will drop the subscript and denote the steady-state solution of equation (47) as $\tilde{P}$ and the associated steady-state Kalman gain as K, which is obtained by substituting $\tilde{P}$ in equation (46).

Writing the outputs in terms of the Kalman filter states gives an alternative formulation for equation (2), which is known as the forward innovation formulation and is more convenient for deriving PSID (see equation (48) below). In particular, first, this formulation shows that the optimal estimate of the latent state is a linear function of the past neural activity (see equation (49) below). Second, this formulation shows that the best prediction of future behavior and neural activity using past neural activity is a linear function of the latent state (equation (45) or equation (48) below). Together, with these two facts, we can show that linear projections of future behavior and neural activity onto the past neural activity can be used to directly extract the latent states from the data (as the latent state is a linear function of past neural activity and thus involves a projection onto the past neural activity), without knowing model parameters (Supplementary Note 6). This extracted latent state can then be used to learn the model parameters (Supplementary Note 6). The forward innovation formulation is given by

$$\begin{cases} x_{k+1} = A\,x_k + Ke_k \\ y_k = C_y x_k + e_k \\ z_k = C_z\,x_k + \varepsilon_k \end{cases}. \tag{48}$$

Here $x_k \triangleq \hat{x}_{k|k-1}$, K is the steady-state Kalman gain and e_k is the innovation process, which is the part of y_k that is not predictable from its past values[35,62]. Equations (2) and (48) have different state and noise time-series but are equivalent alternative internal descriptions for the same y_k and z_k (Supplementary Note 3). The forward innovation formulation in equation (48) is more convenient (compared with the forward stochastic formulation in equation (2)) for the derivation of PSID. Specifically, by recursively substituting the previous iteration of equation (48) into its current iteration, it can be shown that

$$\hat{x}_{k|k-1} = \Delta^c_{y_k} \Lambda^{-1}_{y_k} \begin{bmatrix} y_0 \\ \vdots \\ y_{k-1} \end{bmatrix}. \tag{49}$$

where

$$\Delta^c_{y_k} = \begin{bmatrix} A^{k-1}G_y & A^{k-2}G_y & \cdots & AG_y & G_y \end{bmatrix} \tag{50}$$

and

$$\Lambda_{y_k} \triangleq \begin{bmatrix} \Sigma_{\mathrm{y}_0} & \Sigma_{\mathrm{y}_{-1}} & \cdots & \Sigma_{\mathrm{y}_{1-k}} \\ \Sigma_{\mathrm{y}_1} & \Sigma_{\mathrm{y}_0} & \cdots & \Sigma_{\mathrm{y}_{2-k}} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{\mathrm{y}_{k-1}} & \Sigma_{\mathrm{y}_{k-2}} & \cdots & \Sigma_{\mathrm{y}_0} \end{bmatrix} \tag{51}$$

with the notation $\Sigma_{\mathrm{y_d}} \triangleq \boldsymbol{E}\{y_{k+d}y_k^T\}$ (Theorem 6 from ref. 35). This formulation reveals a key observation that enables identification of model parameters via a direct extraction of the latent state: the latent state in equation (48) (which is an equivalent formulation for equation (2)), is a linear function of the past neural activity y_k. Moreover, from equation (2), it can be shown that for $\mathrm{d} \geq 1$

$$\Sigma_{\mathrm{y_d}} \triangleq \boldsymbol{E}\{y_{k+d}y_k^T\} = C_y A^{d-1} G_y, \qquad \Sigma_{\mathrm{y_{-d}}} = \left(C_y A^{d-1} G_y\right)^T \tag{52}$$

indicating that Λ_{y_k} in equation (51) and thus the linear prediction function $\Delta_{y_k}^c \Lambda_{y_k}^{-1}$ in (49) only depend on Σ_y, A, C_y and G_y[35,62]. Thus, from equations (45) and (49) it is clear that the only parameters that are needed for optimal prediction of y_k and z_k using past y_k are A, C_y, C_z, G_y and Σ_y, which are all parameters that are uniquely identifiable within a similarity transform[35,62] (Supplementary Note 3). As we confirm with numerical simulations, all these parameters can be accurately identified using PSID (Extended Data Fig. 2).

Supplementary Note 6 | Derivations of PSID

PSID, stage 1: Extracting behaviorally relevant latent states

The central idea in PSID is that the part of z_k that is predictable from past y_k not only is a linear function of the latent state (equation (45)), but also is a linear combination of the past y_k (equations (45) and (49)) and thus must lie in a subspace of the space spanned by the past y_k (Supplementary Note 5). We can thus use an orthogonal projection from future z_k onto past y_k to extract the part of z_k that is predictable from past y_k, which leads to the *direct* extraction of the behaviorally relevant latent states from the neural and behavior data y_k and z_k, *even before the model parameters are known*. Given the extracted latent states, the model parameters can then be identified using least squares based on equation (4) as described below.

In the first stage of PSID, the part of z_k that is predictable from past y_k is extracted from the training data by projecting the future z_k values onto their corresponding past y_k values. To find the projection, for each time k, we consider the corresponding 'past' and 'future' to be the previous i samples and the next $i-1$ samples respectively, with i being a user specified parameter termed the projection horizon. For each sample y_k with $i \leq k \leq N-i$, the previous (past) i samples (from y_{k-i} to y_{k-1}) are all stacked together as the $(k-i+1)$th column of one large matrix $Y_p \in \mathbb{R}^{in_y \times j}$ (with $j = N-2i+1$); correspondingly, for each sample y_k with $i \leq k \leq N-i$, that sample together with the next (future) $i-1$ samples (from y_k, to y_{k+i-1}) are all stacked together as the $(k-i+1)$th column of one large matrix $Y_f \in \mathbb{R}^{in_y \times j}$ (equation (8)). Analogously, we form matrices $Z_p \in \mathbb{R}^{in_z \times j}$ and $Z_f \in \mathbb{R}^{in_z \times j}$ from z_k (equation (9)). Thus, Z_f and Y_p have the same number of columns with each column of Z_f containing some consecutive samples of behavior while the corresponding column in Y_p contains the previous i samples from neural activity. The goal is to find the part of Z_f that is linearly predictable from corresponding columns of Y_p (i.e. the behavior in each column of Z_f from its past neural activity). The linear least squares solution for this prediction problem has the closed form solution given in equation (10)[35,62], which is in the form of a projection from future behavior onto past neural activity. We show below that this projection can be decomposed into the multiplication of an observability matrix for behavior and a running estimate of the Kalman estimated latent states, which will thus enable these states to be directly extracted first and then the model parameters to be identified using the extracted latent states.

First, note that the least squares solution of equation (10) can also be written as

$$\hat{Z}_f = Z_f Y_p^T \left(Y_p Y_p^T\right)^{-1} Y_p = \Sigma_{z_f y_p} \Sigma_{y_p y_p}^{-1} Y_p \tag{53}$$

where $\Sigma_{z_f y_p} \triangleq \frac{1}{j} Z_f Y_p^T$ and $\Sigma_{y_p y_p} \triangleq \frac{1}{j} Y_p Y_p^T$ are sample covariance matrices for the covariance of past neural activity with future behavior and past neural activity, respectively, computed using their observed time-samples from equations (8) and (9). Sample covariance estimates are asymptotically unbiased and thus for $j \to \infty$ they would converge to their true value in the model[35,62]. Consequently, for the model in equation (2), it can be shown (by

replacing samples covariances with exact covariances from the model) that for $j \to \infty$, $\Sigma_{y_p y_p}$ converges to Λ_{y_i} defined per equation (51) and $\Sigma_{z_f y_p}$ converges to

$$\Lambda_{zy_i} \triangleq \begin{bmatrix} \Sigma_{zy_i} & \Sigma_{zy_{i-1}} & \cdots & \Sigma_{zy_1} \\ \Sigma_{zy_{i+1}} & \Sigma_{zy_i} & \cdots & \Sigma_{zy_2} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{zy_{2i-1}} & \Sigma_{zy_{2i-2}} & \cdots & \Sigma_{zy_i} \end{bmatrix} \tag{54}$$

where we are using the notation $\Sigma_{zy_{\mathrm{d}}} \triangleq \boldsymbol{E}\{z_{k+d} y_k^T\}$. From equation (2) it can be shown that

$$\Sigma_{zy_{\mathrm{d}}} \triangleq \boldsymbol{E}\{z_{k+d} y_k^T\} = C_z A^{d-1} G_y, \qquad \Sigma_{zy_{-\mathrm{d}}} = \left(C_z A^{d-1} G_y\right)^T \tag{55}$$

which has a form similar to equation (52). Substituting into the definition of Λ_{zy_i} from equation (54) gives

$$\Lambda_{zy_i} \triangleq \begin{bmatrix} \Sigma_{zy_i} & \Sigma_{zy_{i-1}} & \cdots & \Sigma_{zy_1} \\ \Sigma_{zy_{i+1}} & \Sigma_{zy_i} & \cdots & \Sigma_{zy_2} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{zy_{2i-1}} & \Sigma_{zy_{2i-2}} & \cdots & \Sigma_{zy_i} \end{bmatrix} = \begin{bmatrix} C_z \\ C_z A \\ \vdots \\ C_z A^{i-1} \end{bmatrix} \begin{bmatrix} A^{i-1} G_y & A^{i-2} G_y & \cdots & A G_y & G_y \end{bmatrix} \triangleq \Gamma_{z_i} \Delta_{y_i}^c \tag{56}$$

where Γ_{z_i} is termed the extended observability matrix for the pair (A, C_z) and $\Delta_{y_i}^c$ is termed the reversed extended controllability matrix for the pair $\left(A, G_y\right)$[35]. Moreover, based on equation (49), the Kalman filter prediction at time k using only the last i observations (y_{k-i} to y_{k-1}) can be written in terms of $\Delta_{y_i}^c$ (equation (56)) as

$$\hat{x}_{k|k-1} = \Delta_{y_i}^c \Lambda_{y_i}^{-1} \begin{bmatrix} y_{k-i} \\ \vdots \\ y_{k-1} \end{bmatrix}. \tag{57}$$

Thus, for $j \to \infty$, equation (53) can be written as

$$\hat{Z}_f = Z_f Y_p^T \left(Y_p Y_p^T\right)^{-1} Y_p = \Sigma_{z_f y_p} \Sigma_{y_p y_p}^{-1} Y_p = \Lambda_{zy_i} \Lambda_{y_i}^{-1} Y_p = \Gamma_{z_i} \Delta_{y_i}^c \Lambda_{y_i}^{-1} Y_p = \Gamma_{z_i} \hat{X}_i \tag{58}$$

where the last equality follows from equation (57) by setting columns of $\hat{X}_i$ to be the Kalman estimates obtained using the past i observations of y_k per equation (57). Note that equations (56) and (57) hold for any similarity transform of the model, including the formulation in equation (4). Rewriting equation (58) for this formulation and replacing Γ_{z_i} with its value for this formulation (defined in equation (44)) gives

$$\hat{Z}_f = Z_f Y_p^T \left(Y_p Y_p^T\right)^{-1} Y_p = \Gamma_{z_i} \hat{X}_i = \begin{bmatrix} \Gamma_{z_i}^{(1)} & 0 \end{bmatrix} \hat{X}_i = \begin{bmatrix} \Gamma_{z_i}^{(1)} & 0 \end{bmatrix} \begin{bmatrix} \hat{X}_i^{(1)} \\ \hat{X}_i^{(2)} \end{bmatrix} = \Gamma_{z_i}^{(1)} \hat{X}_i^{(1)} \tag{59}$$

where $\hat{X}_i^{(1)}$ is defined as the first n_1 rows of $\hat{X}_i$ and is thus the Kalman estimate for the behaviorally relevant states $x_k^{(1)}$ from equation (4) and $\hat{X}_i^{(2)}$ is defined as the remaining rows of $\hat{X}_i$ and is the Kalman estimate for the behaviorally irrelevant states $x_k^{(2)}$ from equation (4). Given that $\Gamma_{z_i}^{(1)}$ and $\hat{X}_i^{(1)}$ are both full rank of rank n_1, equation (59) shows that the row space of $\hat{Z}_f$ (i.e. the space spanned by the rows of $\hat{Z}_f$) is equal to the row space of the behaviorally relevant states $\hat{X}_i^{(1)}$. This is the key insight that, as we show next, allows the first stage of PSID to use SVD to recover this row space and thus extract behaviorally relevant latent states directly from data (without knowing the model parameters), without the need to extract the behaviorally irrelevant latent states (i.e. prioritize the behaviorally relevant latent states).

Before we use equation (59) to conclude the derivation of the first stage of PSID, it is useful for the derivation of the second stage to note that if we repeat the above steps for the projection of Y_f onto Y_p, we will get

$$\hat{Y}_f = Y_f Y_p^T \left(Y_p Y_p^T\right)^{-1} Y_p = \Sigma_{y_f y_p} \Sigma_{y_p y_p}^{-1} Y_p = \Gamma_{y_i} \Delta_{y_i}^c \Lambda_{y_i}^{-1} Y_p = \Gamma_{y_i} \hat{X}_i = \begin{bmatrix} \Gamma_{y_i}^{(1)} & \Gamma_{y_i}^{(2)} \end{bmatrix} \begin{bmatrix} \hat{X}_i^{(1)} \\ \hat{X}_i^{(2)} \end{bmatrix} \tag{60}$$

where Γ_{y_i} is the extended observability matrix for the pair (A, C_y), $\Gamma_{y_i}^{(1)}$ denotes the first n_1 columns of Γ_{y_i}, $\Gamma_{y_i}^{(2)}$ denotes the remaining columns of Γ_{y_i}, and $\hat{X}_i$, $\hat{X}_i^{(1)}$ and $\hat{X}_i^{(2)}$ are the same Kalman states as in equation (59). Equation (60) shows that the row space of $\hat{Y}_f$ is the union of the row spaces of both behaviorally relevant and behaviorally irrelevant latent states. As we show later in the derivation of stage 2 of PSID, this insight allows the second stage of PSID to recover the subspace of behaviorally irrelevant latent states by first subtracting the subspace of behaviorally relevant latent states—that are already extracted in stage 1—, and then applying SVD to the residual to recover the row space of behaviorally irrelevant latent states $\hat{X}_i^{(2)}$.

Equation (59) shows how $\hat{Z}_f$, which is the projection of future behavior onto past neural activity and is directly computable from data, can be decomposed into the extended behavior observability matrix $\Gamma_{z_i}^{(1)}$ and the

behaviorally relevant Kalman states $\hat{X}_i^{(1)}$. This decomposition allows us to extract the behaviorally relevant latent states even before the model parameters are learned and paves the way for subsequent learning of the model parameters. The decomposition can be performed by taking singular value decomposition (SVD) from $\hat{Z}_f$ in equation (59) (shown in equation (12)), which gives:

$$\Gamma_{z_i}^{(1)} = US^{\frac{1}{2}}, \qquad \hat{X}_i^{(1)} = S^{\frac{1}{2}}V^T \tag{61}$$

Note that the above is only one of many valid decompositions since multiplying any non-singular matrix T onto Γ_{z_i} from the right and its inverse T^{-1} onto $\hat{X}_i^{(1)}$ from the left amounts to a similarity transform and gives an equivalent model with a different basis[35]. For $j \to \infty$, as shown earlier, equation (59) holds exactly and thus the row rank of $\hat{Z}_f$ and the number of its non-zero singular values will be equal to the rank of $\hat{X}_i^{(1)}$ and $\Gamma_{z_i}^{(1)}$, which is n_1 (Supplementary Note 4). For finite data ($j < \infty$), it is expected that an approximation of this relation will hold and thus one could find n_1 via inspection of the singular values of $\hat{Z}_f$. In Methods, we instead proposed a more general method of using cross validation to find both n_1 and n_x, which doesn't require an ad-hoc determination of which singular values are notably larger than the others. Regardless of how n_1 is determined, keeping the top n_1 singular values from the SVD, we can extract $\hat{X}_i^{(1)}$ as in equation (14). Note that intuitively, keeping of the top singular values ensures that the states that describe the behavior best (i.e. best approximate $\hat{Z}_f$) are prioritized.

Having decomposed $\hat{Z}_f$ into $\Gamma_{z_i}^{(1)}$ and $\hat{X}_i^{(1)}$, determining the model parameters from these matrices is straight forward and there are multiple possible ways to accomplish this. We take an approach in the spirit of stochastic algorithm 3 from ref. 35 in SID, and use the state matrix $\hat{X}_i^{(1)}$ to identify the model parameters. This method has the advantage of guaranteeing that the identified noise statistics are positive semi-definite, which is necessary for the model to be physically meaningful[35]. We first compute the subspace for the latent states at the next time step (having observed Y_i as defined in equation (8) in addition to Y_p, i.e. having observed the past $i+1$ samples) by projecting Z_f^- onto Y_p^+ (equation (11)). Similar to equation (59), this projection can be decomposed as

$$\hat{Z}_f^- = Z_f^- Y_p^{+T}\left(Y_p^+ Y_p^{+T}\right)^{-1} Y_p^+ = \Gamma_{z_{i-1}} \Delta_{y_{i+1}}^c \Lambda_{y_{i+1}}^{-1} Y_p^+ = \Gamma_{z_{i-1}} \hat{X}_{i+1} = \Gamma_{z_{i-1}}^{(1)} \hat{X}_{i+1}^{(1)} \quad (62)$$

where $\Gamma_{z_{i-1}}$, $\Delta_{y_{i+1}}^c$, $\Gamma_{z_{i-1}}^{(1)}$ and $\Lambda_{y_{i+1}}$ are defined similar to equations (56), (44) and (51) and columns of $\hat{X}_{i+1}$ are Kalman estimates obtained using the past $i+1$ observations of y_k (from equation (57)), and $\hat{X}_{i+1}^{(1)}$ is defined as the first n_1 rows of $\hat{X}_{i+1}$. From the definition of observability matrix, it is clear that $\Gamma_{z_{i-1}}^{(1)}$ can be computed by removing the last block row of $\Gamma_{z_i}^{(1)}$ (equation (15)). $\hat{X}_{i+1}^{(1)}$ can then be computed (in the same basis as $\hat{X}_i^{(1)}$) by multiplying both sides of equation (62) with ${\Gamma_{z_{i-1}}^{(1)}}^{\dagger}$ from the left (equation (16)). This concludes the extraction of behaviorally relevant latent states. If one is not interested in behaviorally irrelevant latent states, then all model parameters for a model for the behaviorally relevant latent states can also be identified at this point by skipping the second stage of PSID and performing the parameter identification (steps 7 and 17-19 in Supplementary Note 1). Alternatively, if behaviorally irrelevant latent states are also of interest, before parameter identification, in the second stage of PSID, we extract the behaviorally irrelevant latent states (optional) and add them to the previously extracted behaviorally relevant latent states. Then parameter identification can be performed to learn all model parameters as described below.

PSID, stage 2: extracting behaviorally irrelevant latent states

So far we have extracted the behaviorally relevant latent states as the key first step toward learning the model parameters. Optionally, to also find any remaining behaviorally irrelevant states, we need to find the variations in neural activity that are not explained by the behaviorally relevant latent states. We thus first remove any variations in Y_f (and Y_f^-) that lies in the subspace spanned by the extracted behaviorally relevant states $\hat{X}_i^{(1)}$ (and $\hat{X}_{i+1}^{(1)}$) (i is horizon as defined previously), and then apply a procedure akin to the standard SID to the residual. The least squares solution for the best linear prediction of Y_f using $\hat{X}_i^{(1)}$ is given by equation (18), and is termed $\Gamma_{y_i}^{(1)}$. This solution can be thought of as the neural observability matrix associated with the behaviorally relevant states $\hat{X}_i^{(1)}$ (equation (60)). Thus, similar to equation (62), the associated observability matrix for $\hat{X}_{i+1}^{(1)}$ can be computed by removing the

last block row from the solution (equation (20)). We then subtract the best prediction of Y_f (Y_f^-) using $\hat{X}_i^{(1)}$ ($\hat{X}_{i+1}^{(1)}$) from it as shown in equation (19) (equation (21)), and call the result $Y_f{}'$ ($Y_f^{-\prime}$). In other words, $Y_f{}'$ ($Y_f^{-\prime}$) is the part of Y_f (Y_f^-) that does not lie in the space spanned by $\hat{X}_i^{(1)}$ ($\hat{X}_{i+1}^{(1)}$). Given that $\hat{X}_i^{(1)}$ and thus $\Gamma_{y_i}^{(1)}\hat{X}_i^{(1)}$ (i.e. the linear prediction of Y_f using $\hat{X}_i^{(1)}$) are of rank n_1 and that $\hat{Y}_f$ (i.e. the projection of Y_f onto Y_p) is of rank n_x (equation (60)), the projection of $Y_f{}' = Y_f - \Gamma_{y_i}^{(1)}\hat{X}_i^{(1)}$ (i.e. residual future neural activity) onto Y_p will be of rank $n_2 = n_x - n_1$. A similar procedure to what was applied to Z_f (and Z_f^-) to find $\hat{X}_i^{(1)}$ (and $\hat{X}_{i+1}^{(1)}$) can be applied to $Y_f{}'$ (and $Y_f^{-\prime}$) to extract the n_2 remaining states $\hat{X}_i^{(2)}$ (and $\hat{X}_{i+1}^{(2)}$) (steps 11-14 from Supplementary Note 1). Of note is that in this stage, keeping the top singular values after SVD (equation (24)) ensures that the remaining states that describe the unexplained neural activity best (i.e. best approximate $\hat{Y}_f'$) are prioritized.

The above concludes the extraction of behaviorally irrelevant latent states. Concatenating the states extracted from both stages (i.e. $\hat{X}_i^{(1)}$ and $\hat{X}_i^{(2)}$ as well as $\hat{X}_{i+1}^{(1)}$ and $\hat{X}_{i+1}^{(2)}$) together as in equation (29) concludes the extraction of all latent states, including behaviorally relevant and irrelevant ones.

PSID: identification of all model parameters given the extracted latent states

Next, given the extracted latent states from stage 1 and optionally stage 2, we identify the model parameters. First, we identify blocks of the A matrix in the format of equation (4). To identify A_{11}, which is the block associated with the behaviorally relevant latent states, we take columns of $\hat{X}_{i+1}^{(1)}$ and $\hat{X}_i^{(1)}$ (extracted in stage 1) as samples of the current state and the corresponding next state (i.e. $x_{k+1}^{(1)}$ and $x_k^{(1)}$ from equation (4)), respectively; based on equation (4), we can compute the least squares solution for A_{11} as given in equation (17). Second, if behaviorally irrelevant latent states are also desired to be included in the model, we follow a similar approach as was taken for A_{11} (equation (17)), to find the least squares solution for A_{21} and A_{22} (equation (30)). Then we find the least squares solution for C_y (equation (32)) and C_z (equation (33)) using both the behaviorally relevant and the behaviorally irrelevant latent states extracted in stages 1 and 2 (or using just the behaviorally relevant latent states if behaviorally irrelevant

dynamics are not of interest and thus the optional stage 2 is not performed). Finally, the residuals from the least squares solutions to equations (17), (30) and (32) provide estimated values for w_k and v_k at each time step and thus we compute the sample covariance of these residuals to identify the noise covariance parameters (equation (35)). This concludes the identification of all model parameters.

Finally, in addition to equation (33), another viable alternative for finding the parameter C_z is to learn it using linear regression, which is the procedure needed for the standard SID to relate its extracted latent state to behavior and we use in this paper for both SID and PSID. Since C_z is not involved in the Kalman filter recursions (first 2 rows of equation (45)), it does not have any effect on the extraction of latent states from y_k and it only affects the later prediction of z_k from those latent states. Consequently, we can use the other identified parameters to apply Kalman filter to the training y_k and estimate the latent states $\hat{x}_{k+1|k}$ (equation (45)). We can then use linear regression to find the C_z that minimizes the prediction of z_k using $\hat{x}_{k+1|k}$ as

$$C_z = Z_k \hat{X}^{\dagger}_{k+1|k} \tag{63}$$

where columns of Z_k contain z_k for different time steps and columns of $\hat{X}_{k+1|k}$ contain the corresponding $\hat{x}_{k+1|k}$ estimates for those time steps. The advantage of using this alternative identification of C_z is that $\hat{X}_{k+1|k}$ (used in equation (63)) are more accurate estimates of the latent states obtained using all past observations whereas $\hat{X}_i$ (used in equation (33)) are less accurate estimates obtained using only the past i observations.

Supplementary Note 7 | Standard SID as a special case of PSID and the asymptotic characteristics of PSID

As a review of the standard SID, we refer the reader to chapter 8 from ref. 62 and chapter 3 from ref. 35. For $n_1 = 0$, PSID (Supplementary Note 1) reduces to the standard SID (specifically to stochastic algorithm 3 from ref. 35). This is because if $n_1 = 0$, no behaviorally relevant states ($\hat{X}_i^{(1)}$) will be extracted leaving all variation of Y_f to be identified in stage 2 of PSID, which is similar to using standard SID. Thus, the extracted $\hat{X}_i^{(2)}$ in this case will be the

same as the $\hat{X}_i$ that is obtained from applying SID on y_k. So to compare PSID with SID, we simply use PSID with $n_1 = 0$.

As a generalization of the abovementioned version of SID (i.e. stochastic algorithm 3 from ref. 35), PSID has similar asymptotic characteristics. In some other variations of SID (for example in stochastic algorithm 2 from ref. 35 and in Algorithm A in section 8.7 from ref. 62), instead of applying SVD to $\hat{Y}_f$, SVD is applied to the empirical cross-covariance $\Sigma_{y_f y_p}$ to decompose it into Γ_{y_i} and $\Delta^c_{y_i}$ (equation (60)), giving an estimation of these matrices, which for $j \to \infty$ is unbiased[35]. From this decomposition, model parameters A, C_y, and G_y can then be extracted—C_y as the first block of Γ_{y_i}, G_y as the last block of $\Delta^c_{y_i}$, and A with a least squares solution within blocks of Γ_{y_i}—leaving the final uniquely identifiable model parameter Σ_y to be extracted as the first block of the auto-covariance $\Sigma_{y_p y_p}$ again in an asymptotically unbiased manner[35] (for details see the SID variants mentioned in the previous sentence). However, this approach cannot guarantee that for finite data ($j < \infty$) the identified A, C_y, G_y, and Σ_y will be associated with a positive real covariance sequence (i.e. Faurre's stochastic realization may have no solution)[35,62]. In the alternative approach taken by PSID (and its special case, stochastic algorithms 3 from ref. 35), A and C_y (and C_z) are computed as least squares solution of forming equation (2) with $\hat{X}_i$ taken as the value of the latent state and G_y and Σ_y are identified later based on the residuals of the least squares solution (from equations (37)-(39) as detailed in Supplementary Note 1). This approach cannot guarantee an asymptotically unbiased estimate of G_y and Σ_y (unless $i \to \infty$ in which case Kalman estimates in equation (57) will be exact), but it guarantees that even for finite data ($j < \infty$) the identified parameters will be associated with a positive real covariance sequence[35], which is essential for the model to be physically meaningful[37].

Supplementary Note 8 | Generating random model parameters for simulations

For a model with given n_x and n_1, A was generated by first randomly generating its eigenvalues and then generating a block diagonal real matrix with the randomly selected eigenvalues (using MATLAB's cdf2rdf command). We drew the eigenvalues with uniform probability from across the complex unit circle and then

randomly selected n_1 of the n_x to be later used as behaviorally relevant eigenvalues. As a technical detail, in both the original random generation of eigenvalues and in selecting n_1 of them for behavior we made sure eigenvalues are either real valued or are in complex-conjugate pairs (as needed for models with real observations). To do this, we first drew $\left\lfloor \frac{n_x}{2} \right\rfloor$ points with uniform probability from across the complex unit circle and then added the complex conjugate of each to the set of eigenvalues. If n_x was odd, we then drew an additional eigenvalue from the unit circle and set its angle to 0 or π, whichever was closer. Finally, to randomly select n_1 of the n_x eigenvalues to be used as behaviorally relevant, we repeatedly permuted the values until the first n_1 eigenvalues also formed a set of complex conjugate pairs or real values.

Next, we generated $C_y \in \mathbb{R}^{n_y \times n_x}$ by drawing each element from standard normal distribution. We generated $C_z \in \mathbb{R}^{n_z \times n_x}$ by drawing values from the standard normal distribution for the elements associated with the behaviorally relevant eigenvalues of A (or equivalently for the dimensions of x_k that drive behavior) and setting the other elements to 0 (see equation (4)).

For noise statistics Q, R, and S, we generated general random covariance matrices and applied random scaling factors to them to get a wide range of relative variances for the state noise w_k and observation noise v_k. To do this, we first generated a random square matrix Ω of the size $n_x + n_y$ by drawing elements from standard normal distribution and computed $L = \Omega\Omega^T$, which is guaranteed to be symmetric and positive semi-definite. We next selected random scaling factors for the state noise w_k and the observation noise v_k by independently selecting two real numbers a_1, a_2 with uniform probability from the range $(-1, +1)$. We then applied the following scaling to matrix L to get the noise statistics as

$$\begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} = \begin{bmatrix} 10^{a_1} I_{n_x} \\ 10^{a_2} I_{n_y} \end{bmatrix} L \begin{bmatrix} 10^{a_1} I_{n_x} & 10^{a_2} I_{n_y} \end{bmatrix} \tag{64}$$

where I_n denotes the identity matrix of the size n. This is equivalent to scaling v_k by 10^{a_1} and independently scaling w_k by 10^{a_2} and generates a wide range of state and observation noise statistics.

Finally, to build a model for generating the independent behavior residual dynamics ϵ_k (which can be a general colored signal and is not assumed to be white), we generate another random dynamic linear state-space model with independently selected latent state dimension of $1 \leq n'_x \leq 10$ and parameters generated as explained above for the main model. We will refer to this model as the behavior residual dynamics model. To diversify the ratio of behavior dynamics that are shared with neural activity (equation (36)) to the residual behavior dynamics (i.e. ϵ_k), we draw a random number α_3 in the range $(0, +2)$. We then multiply the rows of the C_z parameter in the behavior residual dynamics model with different scalar values such that for each behavior dimension m, the shared-to-residual ratio, defined as the ratio of the std of the term $z_{k_1}^{(m)}$ (equation (36)) to the std of the term $\epsilon_k^{(m)}$, is equal to 10^{α_3}.