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

nature neuroscience



# Modeling behaviorally relevant neural dynamics enabled by preferential subspace identification

Omid G. Sani[1], Hamidreza Abbaspourazad[1], Yan T. Wong[2,5], Bijan Pesaran[2] and Maryam M. Shanechi[1,3,4] ✉

**Neural activity exhibits complex dynamics related to various brain functions, internal states and behaviors. Understanding how neural dynamics explain specific measured behaviors requires dissociating behaviorally relevant and irrelevant dynamics, which is not achieved with current neural dynamic models as they are learned without considering behavior. We develop preferential subspace identification (PSID), which is an algorithm that models neural activity while dissociating and prioritizing its behaviorally relevant dynamics. Modeling data in two monkeys performing three-dimensional reach and grasp tasks, PSID revealed that the behaviorally relevant dynamics are significantly lower-dimensional than otherwise implied. Moreover, PSID discovered distinct rotational dynamics that were more predictive of behavior. Furthermore, PSID more accurately learned behaviorally relevant dynamics for each joint and recording channel. Finally, modeling data in two monkeys performing saccades demonstrated the generalization of PSID across behaviors, brain regions and neural signal types. PSID provides a general new tool to reveal behaviorally relevant neural dynamics that can otherwise go unnoticed.**

Modeling how behavior is encoded in the dynamics of neural activity over time is a central challenge in neuroscience. Here, we specifically define behavior as a behavioral signal that is of interest and is measured within a given task; for example, measured arm kinematics during a motor task. We also use terms such as 'behaviorally relevant' and 'behaviorally irrelevant' only with respect to such measured behavioral signals. We refer to the dynamics of neural activity regardless of the neural signal type—whether neural population spiking activity or local field potentials (LFPs)—as neural dynamics. Modeling neural dynamics is essential for investigating or decoding behaviorally measurable brain functions such as movement[1–4], speech and language[5], mood[4,6] and decision-making[7], as well as neurological dysfunctions such as movement tremor[8]. However, building such models is challenging for two main reasons. First, in addition to the specific behavior being measured and studied, the dynamics in recorded neural activity encode other brain functions/behaviors, inputs from other neurons and internal states with brain-wide representations such as thirst[2,9–14]. If these other dynamics do not influence the measured behavior during neural recordings, we term them behaviorally irrelevant neural dynamics. Second, many natural behaviors such as movements or speech are temporally structured. Thus, understanding their neural representation is best achieved by learning a dynamic model, which explicitly characterizes the temporal evolution of neural population activity[2,15–17]. Given these two challenges, answering increasingly sought-after and fundamental questions about neural dynamics, such as their dimensionality[2,13,18–20], and important temporal features, such as rotations[14,21–23], requires a novel dynamic modeling framework that can prioritize extracting those neural dynamics that are related to a specific measured behavior of interest. This would ensure that behaviorally relevant neural dynamics are not masked or confounded by other dynamics and will broadly affect the study of diverse brain functions. Developing such a dynamic modeling framework has remained elusive to date.

Currently, dynamic modeling of neural activity is largely performed according to two alternative conceptual frameworks. In both, a state space model formulation is used, but these frameworks differ regarding which elements within the brain state—a latent signal describing the overall state of the brain—they can learn and model as their state (Fig. 1a). In the first framework, often termed representational modeling (RM), behavioral measurements such as movement kinematics, choices or tremor intensity at each point in time are assumed to be directly represented in the neural activity at that time[2,3,8,24,25]. By making this assumption, RM implicitly assumes that the dynamics of neural activity are the same as those in the behavior of interest, and the RM framework therefore takes behavior as the state in the model and learns its dynamics without considering the neural activity (Fig. 1a and Methods). This assumption, however, may not hold, since neural activity in many cortical regions, including the prefrontal[7,26], motor[9,21,27] and visual[13] cortices and other brain structures such as the amygdala[12,28], often contains components that are responsive to different behavioral and task parameters[7,9,12,13,21,26–28] and is therefore not fully explained by the RM framework[2,10,16,17,21,27]. Motivated by this complex neural response, a second framework, known as neural dynamic modeling (NDM), has received growing attention[2,6,15,17,21–23,29–34] and has led to recent findings, for example, about movement generation[2,21] and mood[6]. In NDM, the dynamics of neural activity are modeled in terms of a latent variable that constitutes the state in the model and is learned purely using the recorded neural activity and agnostic to the behavior (Fig. 1a). When the neural encoding of behavioral measurements such as movement kinematics[22,30,32] or mood variations[6] is of interest, a second step is needed to relate the already learned latent state to behavior. Because NDM does not use behavior to guide the learning of neural dynamics, it may miss or

[1]Ming Hsieh Department of Electrical and Computer Engineering, Viterbi School of Engineering, University of Southern California, Los Angeles, CA, USA. [2]Center for Neural Science, New York University, New York City, NY, USA. [3]Neuroscience Graduate Program, University of Southern California, Los Angeles, CA, USA. [4]Department of Biomedical Engineering, University of Southern California, Los Angeles, CA, USA. [5]Present address: Department of Physiology, and Electrical and Computer Systems Engineering, Monash University, Melbourne, Victoria, Australia. ✉e-mail: shanechi@usc.edu

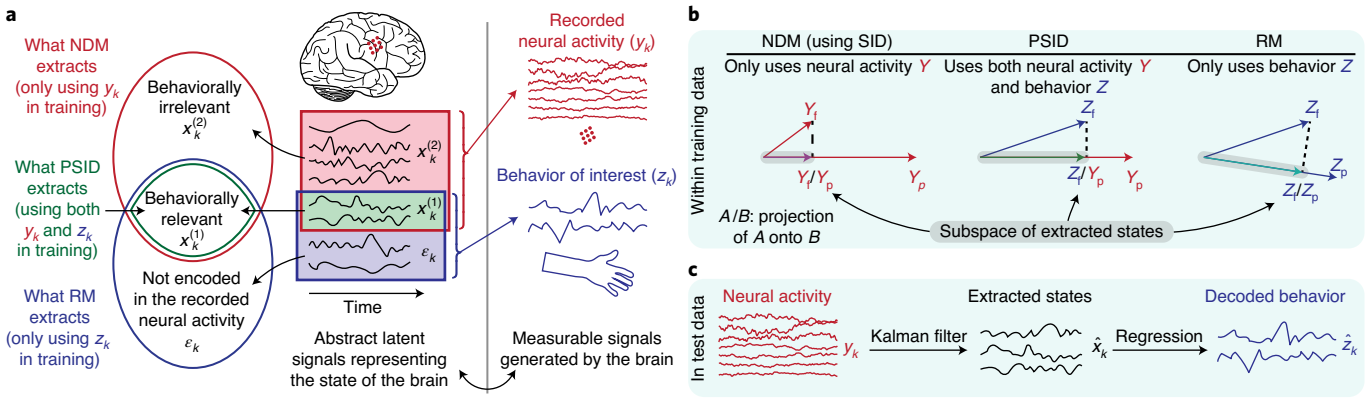



**Fig. 1 | PSID enables learning of dynamics shared between the recorded neural activity and the measured behavior. a**, Schematic of how the state of the brain can be thought of as a high-dimensional time-varying variable of which some dimensions ($x_k^{(1)}$ and $x_k^{(2)}$) drive the recorded neural activity ($y_k$), some dimensions ($x_k^{(1)}$ and $\epsilon_k$) drive the measured behavior ($z_k$) and some dimensions ($x_k^{(1)}$) drive both and are therefore shared between them (a numerical simulation is shown in Supplementary Fig. 1). The choice of the learning method affects which elements of the brain states can be extracted from the neural activity. NDM extracts states regardless of their relevance to behavior and RM extracts states regardless of their relevance to the recorded neural activity. PSID enables extraction of states that are related to both the recorded neural activity and a specific measured behavior. **b**, Schematic of how PSID achieves its goal in learning the dynamic model (see also Extended Data Fig. 1) in comparison to a representative NDM method (that is, SID) and a RM method (that is, kinematic-state Kalman filter). $A/B$ denotes projecting $A$ onto $B$ (Methods). The key idea in PSID during learning is to project future behavior $z_k$ (denoted by $Z_f$) onto past neural activity $y_k$ (denoted by $Y_p$). This is unlike NDM using SID, which instead projects future neural activity (denoted by $Y_f$) onto the past neural activity $Y_p$ (Methods). It is also unlike RM, which projects future behavior onto past behavior (denoted by $Z_p$). **c**, For all three methods, after the model parameters are learned, the procedures for latent state extraction and neural decoding of behavior in the test data are the same. A Kalman filter associated with the learned model operates on the neural activity alone to extract the states, and behavior is decoded by applying a linear regression to these extracted states. The difference between the Kalman filter in the three methods is the parameter values of the filter, which are learned differently as in **b**, with PSID using its novel two-stage learning algorithm (Methods).

less accurately learn some behaviorally relevant neural dynamics that are masked or confounded by those unrelated to the behavior. Uncovering behaviorally relevant neural dynamics requires a new modeling framework to directly learn the dynamics that are shared between the recorded neural activity and measured behavior, rather than learning the prominent dynamics present in one or the other as done by current dynamic frameworks (Fig. 1a).

Here, we develop a novel learning algorithm, termed PSID, for extracting and modeling behaviorally relevant dynamics in high-dimensional neural activity. During learning, PSID uses both neural activity and behavior in training data to learn (that is, identify) a dynamic model that describes neural activity in terms of latent states while prioritizing the extraction and characterization of behaviorally relevant neural dynamics (Methods, Fig. 1b, Box 1 and Extended Data Fig. 1). After learning, the model provides a Kalman filter that operates purely on neural activity in new or test data to extract the behaviorally relevant neural dynamics (Fig. 1c).

We first show with extensive numerical simulations that PSID learns the behaviorally relevant neural dynamics significantly more accurately, with markedly lower-dimensional latent states and with orders of magnitude fewer training samples compared with standard methods. We then demonstrate the new functionalities that PSID enables by applying it to multiregional motor and premotor cortical activity recorded in two monkeys performing a rich three-dimensional (3D) reach, grasp and return task to diverse locations. PSID uniquely uncovers several new features of neural dynamics underlying motor behavior. First, PSID reveals that the dimension of behaviorally relevant neural dynamics is markedly lower than what standard methods conclude, and learns these dynamics more accurately. Second, while both NDM and PSID find rotational neural dynamics during our 3D task, PSID uncovers rotations that are in opposite directions during reach versus return epochs and are significantly more predictive of behavior compared to NDM, which, in contrast, finds rotations in the same direction. Third, compared to NDM and RM, PSID more accurately learns behaviorally relevant neural dynamics for almost all 27 upper-extremity joint angles, for 3D end-point kinematics and for almost all individual recording channels. Finally, we apply PSID to raw LFP prefrontal activity during a different saccadic eye movement task to demonstrate that PSID generalizes across behavioral tasks, brain regions and neural signal types.

## Results

**Overview of PSID and the evaluation framework.** Here, we provide a summary of PSID (Box 1, Extended Data Fig. 1 and Supplementary Note 1). We consider the state of the brain at each point in time as a high-dimensional latent variable of which some dimensions may drive the recorded neural activity, some may drive the measured behavior and some may drive both (Fig. 1a and Supplementary Fig. 1). We therefore model the neural activity ($y_k \in \mathbb{R}^{n_y}$) and behavior ($z_k \in \mathbb{R}^{n_z}$) using the following general dynamic linear state space model formulation:

$$\begin{cases} x_{k+1} = A\,x_k + w_k \\ y_k = C_y\,x_k + v_k \\ z_k = C_z\,x_k + \epsilon_k \end{cases}, \quad x_k = \begin{bmatrix} x_k^{(1)} \\ x_k^{(2)} \end{bmatrix} \tag{1}$$

where $x_k \in \mathbb{R}^{n_x}$ is the latent state that drives the recorded neural activity, and $x_k^{(1)} \in \mathbb{R}^{n_1}$ and $x_k^{(2)} \in \mathbb{R}^{n_2}$ (with $n_2 = n_x - n_1$) are its behaviorally relevant and irrelevant components, respectively. Here, the model is depicted to include both $x_k^{(1)}$ and $x_k^{(2)}$; however, PSID has a novel two-stage identification approach that allows it to learn $x_k^{(1)}$ directly from training data in its first stage without the need to also learn $x_k^{(2)}$ until an optional second stage (Box 1 and Extended Data Fig. 1). This allows PSID to prioritize learning of behaviorally relevant neural dynamics, which is key to learning these dynamics more accurately and with very low-dimensional states (that is, only $x_k^{(1)}$). Finally, $\epsilon_k$ represents behavior dynamics that are not present in the recorded neural activity, and $w_k$ and $v_k$ are noises. $A$, $C_y$ and $C_z$ and

**Box 1 | Summary of the PSID algorithm**

Given the training time-series $\{y_k : 0 \le k < N\}$ and $\{z_k : 0 \le k < N\}$, state dimension $n_x$ and parameter $n_1 \le n_x$ (the number of states extracted in the first stage), this algorithm identifies parameters of a general dynamic linear state space model as set out in equation (4) while prioritizing behaviorally relevant neural dynamics.

**Stage 1:** extract $n_1$ latent states directly via a projection of future behavior onto past neural activity as follows:

1. Form examples of future behavior ($Z_f$) and the associated past neural activity ($Y_p$) and then project the former onto the latter: $\hat{Z}_f = Z_f Y_p^T \left(Y_p Y_p^T\right)^{-1} Y_p$
2. Compute the singular value decomposition (SVD) of $\hat{Z}_f = USV^T \cong U_1 S_1 V_1^T$ and keep the top $n_1$ singular values:
3. Compute the behaviorally relevant latent state $\hat{X}_i^{(1)}$ as $\hat{X}_i^{(1)} = S_1^{\frac{1}{2}} V_1^T$

**Stage 2 (optional):** extract $n_x - n_1$ additional latent states via a projection of residual future neural activity onto past neural activity as follows:

1. Find the prediction of $Y_f$ using $\hat{X}_i^{(1)}$, and subtract this prediction from $Y_f$ and name the result $Y_f'$ (that is, residual future neural activity).
2. Project the residual future neural activity ($Y_f'$) onto past neural activity ($Y_p$): $\hat{Y}_f' = Y_f' Y_p^T \left(Y_p Y_p^T\right)^{-1} Y_p$
3. Compute the SVD of $\hat{Y}_f' = U'S'V'^T \cong U_2 S_2 V_2^T$ and keep the top $n_2 = n_x - n_1$ singular values:
4. Compute the remaining latent state $\hat{X}_i^{(2)}$ as $\hat{X}_i^{(2)} = S_2^{\frac{1}{2}} V_2^T$

**Final step:** given the extracted latent states, based on equation (4), identify model parameters via least squares as follows:

1. If stage 2 is used, concatenate $\hat{X}_i^{(2)}$ to $\hat{X}_i^{(1)}$ to get the full latent state $\hat{X}_i$; otherwise, take $\hat{X}_i = \hat{X}_i^{(1)}$
2. Repeat all steps with a shift of one step in time to extract the states at the next time step ($\hat{X}_{i+1}$)
3. Compute the least squares solution for the model parameters as follows (where the dagger symbol denotes pseudoinverse):

$$A = \hat{X}_{i+1}\hat{X}_i^{\dagger}, \quad C_y = Y_i\hat{X}_i^{\dagger}, \quad C_z = Z_i\hat{X}_i^{\dagger}$$

4. Compute the covariance of the residuals in the above least square solutions to get the statistics of the noises in equation (4).

See Extended Data Fig. 1 for a visualization and Supplementary Note 1 for details.

noise statistics are the model parameters, which PSID learns given training samples from neural activity and behavior (Methods). Special cases of this PSID model also include standard NDM (if states are not dissociated to $x_k^{(1)}$ and $x_k^{(2)}$; Methods) and RM (if $C_z$ is identity and $\epsilon_k = 0$; Methods).

To develop PSID, we showed that the behaviorally relevant latent states ($x_k^{(1)}$) lay in the intersection of the spaces spanned by past neural activity and future behavior (Methods). Thus, in the first stage of PSID, we extracted these states directly from the training data via an orthogonal projection of future behavior onto the past neural activity (Fig. 1b, Extended Data Fig. 1 and Methods). In an optional second stage, if desired, we then extracted any remaining latent states that do not directly drive behavior (that is, $x_k^{(2)}$) using an additional orthogonal projection from the residual neural activity (that is, the part not predicted by the extracted $x_k^{(1)}$) onto past neural activity (Box 1, Extended Data Fig. 1 and Methods). Finally, we learned all model parameters using the extracted latent states (Box 1). Given the learned model, we then constructed a Kalman filter that extracts the latent states from neural activity alone in test data (Fig. 1c).

We compared PSID to NDM and RM. Standard NDM describes neural activity using a latent state space model (equation (1)), with a latent state that is learned agnostic (that is, unsupervised) with respect to behavior[6,22,30,32] and may only later be regressed to behavior[6,22,32]. Since NDM methods extract the latent states and learn their dynamics without considering behavior, unlike PSID, they do not prioritize the behaviorally relevant neural dynamics. We use the standard subspace identification (SID) algorithm[35] to learn the latent state space model in linear NDM[6,31,33]. During learning, SID extracts the latent states by projecting future neural activity onto past neural activity (Fig. 1b); this is unlike PSID, which projects future behavior onto past neural activity (Fig. 1b). As control analyses, we also compared PSID to example nonlinear NDMs (for example, using recurrent neural networks (RNNs)) and to linear NDMs learned with expectation–maximization. To implement RM[3,24], we used the commonly used RM method (sometimes termed kinematic-state Kalman filter[22]), which builds an autoregressive model for behavior and directly relates behavior to neural activity using linear regression[3,24]. RM learns the state and its dynamics agnostic to neural activity (Fig. 1b); therefore, as we will show, RM may learn state dynamics that are not encoded in the recorded neural activity.

Importantly, all three linear methods (RM, NDM and PSID) describe the neural activity with a similar linear state space model (Methods). The critical difference is how model parameters and the dynamics they describe are learned using training neural data (NDM), behavior data (RM) or both (PSID), and therefore which brain states are learned (Fig. 1a). After the model parameters are learned, in all methods, an associated Kalman filter and linear regression operate on neural activity alone in test data to extract the state and decode the behavior, respectively (Fig. 1c and Methods). We used cross-validated correlation coefficient (CC) of decoding behavior using neural activity as the measure for how accurately the behaviorally relevant neural dynamics are learned. We emphasize that in PSID, behavior is only seen in the training data during learning (Fig. 1b); in the test data, behavior is in no way seen and, similar to other methods, only neural activity is used for latent state extraction and decoding (Fig. 1c).

**Neural datasets and behavioral tasks.** We first validated PSID using extensive numerical simulations and then used PSID to uncover behaviorally relevant neural dynamics in three neural signal types across two experiments employing different behavioral tasks. In the first experiment, we recorded neural activity in two monkeys as they performed 3D reach, grasp and return movements to diverse random locations within a 50-cm$^3$ workspace and continuously in time (that is, without any timing cues; Methods). The angles of 27 (monkey J) or 25 (monkey C) joints on the right shoulder, elbow, wrist and fingers at each point in time were tracked via retroreflective markers and were taken as the main behavior of interest (Methods). We also studied joint angular velocities and 3D end-point positions of the hand and elbow. We recorded neural activity from 137 (monkey J) or 128 (monkey C) electrodes covering the primary motor cortex (M1), the dorsal premotor cortex (PMd), the ventral premotor cortex (PMv) and the prefrontal cortex (PFC) in monkey J and the bilateral PMd and PMv in monkey C (Methods). We modeled the neural dynamics in two neural signal types separately: (1) LFP power in seven frequency bands and (2) multiunit spiking activity (Methods). In the second experiment, we recorded neural activity in the PFC using 32 electrodes in two

monkeys performing saccadic eye movements to targets shown on a display[36] (Methods). Here, we took the two-dimensional (2D) eye position as the behavior of interest and modeled the raw LFP activity (Methods), thereby constituting our third neural signal type.

**PSID correctly learns all model parameters.** We applied PSID to simulated data generated from 100 models with random parameters and found that it correctly identified all model parameters in equation (1) with less than 1% normalized error (Extended Data Fig. 2a), with a trend toward lower errors given more training samples (Extended Data Fig. 2b). Moreover, compared with standard SID, PSID showed a similar error and rate of convergence (Extended Data Fig. 2c,d), which indicates that even when learning all latent states is of interest rather than just the behaviorally relevant ones, PSID performs as well as SID. Furthermore, the identification error of both PSID and SID correlated with a measure of how inherently difficult it was to extract the latent states for each random model (Supplementary Fig. 2). Finally, using cross-validation, the model structure parameters $n_x$ and $n_1$ were also accurately identified (Methods and Supplementary Fig. 3).

**PSID prioritizes the identification of behaviorally relevant neural dynamics.** To confirm that PSID correctly prioritizes the identification of behaviorally relevant neural dynamics, we simulated data from 100 random models with 16 latent state dimensions ($n_x = 16$) of which 4 were behaviorally relevant ($n_1 = 4$; Fig. 2). We evaluated the accuracy for learning the behaviorally relevant eigenvalues within the state transition matrix $A$, which characterize the behaviorally relevant neural dynamics (Methods). Using a minimal total state dimension of four, PSID learned all four behaviorally relevant eigenvalues, unlike standard methods; also, standard methods did not achieve similar accuracy as PSID even when they used state dimensions that were as high as in the true model (that is, 16; Fig. 2b).

**PSID requires fewer training samples.** We found that neither RM nor NDM with a low-dimensional state learned behaviorally relevant neural dynamics even when training samples substantially increased (Extended Data Fig. 3). Furthermore, even compared to NDM with a latent state dimension as high as the true model, PSID achieved similar eigenvalue and decoding accuracy but with a lower-dimensional latent state and with orders of magnitude fewer training samples (that is, 200 times fewer; Extended Data Fig. 3). This result suggests that PSID is more data efficient for neural dimensionality reduction while preserving behavior information, which is another important advantage since experimental data are always limited.

**PSID reveals markedly lower dimensionality for behaviorally relevant neural dynamics and extracts them more accurately in motor cortex LFP activity.** Given that PSID can prioritize and dissociate behaviorally relevant neural dynamics, we used it to investigate these dynamics and their true dimensionality in neural activity during our 3D reach, grasp and return task, with behavior being all upper-extremity joint angles (Fig. 3, Extended Data Fig. 4a and Methods). We first examined the LFP power activity.

PSID revealed that behaviorally relevant neural dynamics are much lower-dimensional than implied using standard methods (Fig. 3b,i). Furthermore, PSID identified these dynamics more accurately than standard methods both at that dimension (Fig. 3c,j) and even when standard methods used much higher-dimensional states (Fig. 3d,k). The dimension of behaviorally relevant neural dynamics is defined as the minimal state dimension required to best explain the measured behavior using neural activity. To find this dimension from data with each method, we modeled the neural activity with various state dimensions in each dataset (Fig. 3a,h) and found the smallest state dimension at which the best behavior-decoding performance was achieved (averaged across all joints; Methods). First, we found that the best decoding performance for PSID was significantly higher than NDM and RM, which suggests that PSID

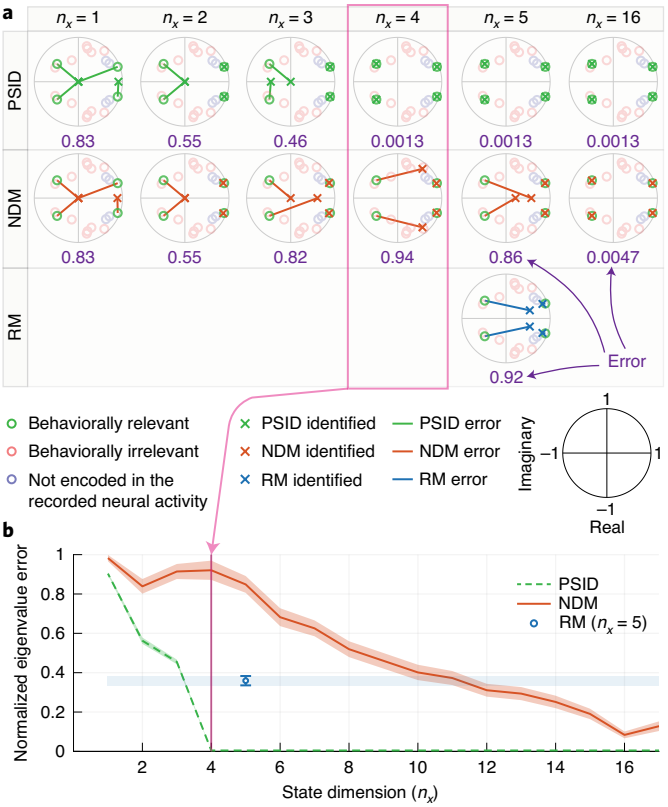



**Fig. 2 | Unlike standard methods, PSID correctly learns the behaviorally relevant neural dynamics even when using lower-dimensional latent states and performing dimensionality reduction. a**, For one simulated model, the identified behaviorally relevant eigenvalues are shown for PSID, NDM and RM and for different latent state dimensions. For RM, the state dimension can only be equal to the behavior dimension (here, $n_z = 5$). Eigenvalues are shown on the complex plane; that is, the real part on the horizontal axis and the imaginary part on the vertical axis. The unit circle is shown in gray. True model eigenvalues are shown as lightly colored circles, with the colors indicating their relevance to neural activity, behavior or both. Crosses show the identified behaviorally relevant eigenvalues when modeling the neural activity. When the state dimension $n_x$ is less than the true dimension of behaviorally relevant states ($n_1 = 4$), missing eigenvalues are taken as 0, representing an equivalent model for which some (that is, $n_1 - n_x$) latent state dimensions are always 0 (Methods). Thus, all cases have four crosses indicating four identified eigenvalues ($4 - n_x$ of which are zero when $n_x < 4$). Lines indicate the identified eigenvalue error whose normalized value—the average line length normalized by the average true eigenvalue magnitude—is noted below each plot (Methods). Unlike PSID, NDM may learn latent states that are unrelated to behavior and RM may learn latent states that are not encoded in the recorded neural activity. **b**, The normalized eigenvalue error given $10^6$ training samples is shown when using PSID, NDM and RM, averaged over 100 random models. For all random models, the total dimension of latent states ($n_x = 16$), the dimension of behaviorally relevant states ($n_1 = 4$) and the number of behavior dimensions not encoded in neural activity (that is, 4) is as in **a**. Solid lines show the average error and shaded areas show the s.e.m. ($n = 100$ random models). For NDM and PSID, the total state dimension is changed from 1 to 16 (for PSID $n_1 = 4$). Since the state dimension for RM can only be equal to the behavior dimension ($n_z = 5$), the RM s.e.m. is shown as error bars and also a horizontal shaded area for easier comparison.

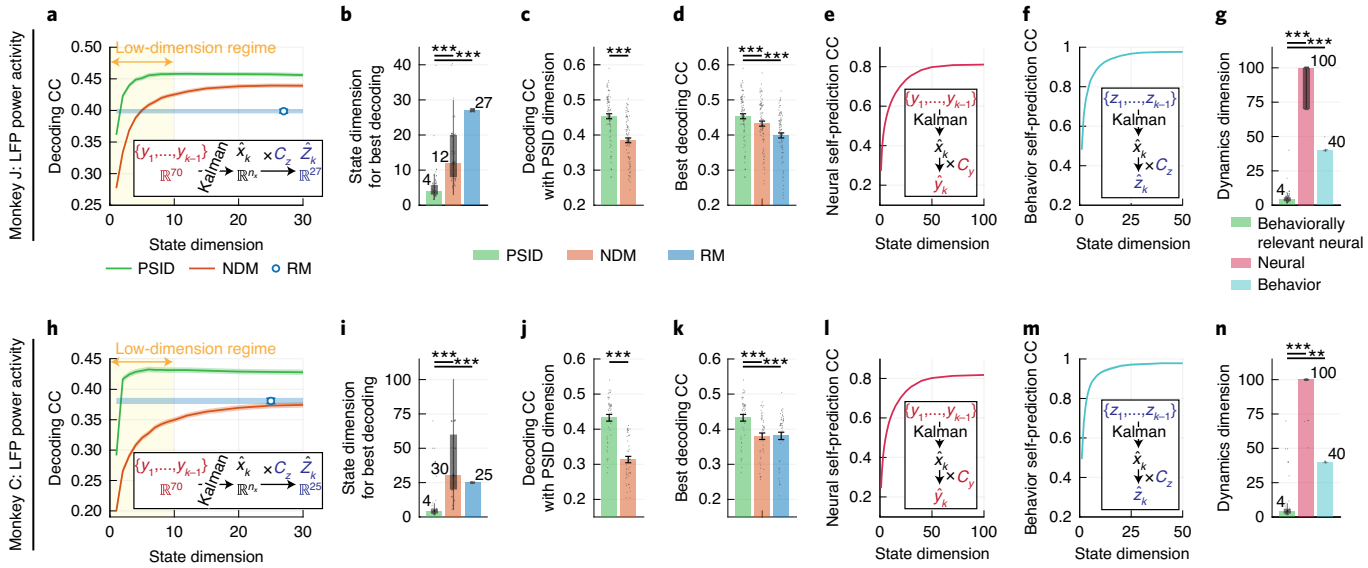



**Fig. 3 | PSID reveals a markedly lower dimension for behaviorally relevant neural dynamics and extracts them more accurately in motor cortex LFP activity during 3D reach, grasp and return movements. a**, The average joint angle decoding accuracy (that is, cross-validated CC) as a function of the state dimension using PSID, NDM and RM. The decoding CC is averaged across the datasets and the shaded area indicates the s.e.m. ($n = 91$ datasets). The dimensionality of neural activity (that is, 70) and behavior (that is, 27) are shown in a box along with the decoder structure. The yellow shaded area indicates the relatively low-dimensional regime, which is often the operating regime of interest in dynamic modeling and dimensionality reduction. **b**, The state dimension that achieves the best decoding in each dataset. Bars show the median (also written next to the bar), box edges show the 25th and 75th percentiles, and whiskers represent the minimum and maximum values (other than outliers). Outliers are the points that are more than 1.5-times the interquartile distance, that is, the box height, away from the top and bottom of the box. All data points are shown. Asterisks indicate significance of statistical tests with $^{***}P < 0.0005$. **c**, The decoding CC for each method when using the same state dimension as PSID (not available for RM since it has a fixed state dimension equal to the behavior dimension). **d**, The best decoding CC in each dataset (state dimensions from **b**). For decoding, bars show the mean and whiskers show the s.e.m. **e**, One-step-ahead self-prediction of neural activity (cross-validated CC) averaged across datasets. **f**, Same as **e**, but for behavior ($n = 7$ sessions). **g**, The behaviorally relevant neural dynamics dimension (that is, PSID result from **b**), total neural dynamics dimension (that is, state dimension from **e**) and total behavior dynamics dimension (that is, state dimension from **f**) for all datasets. **h**–**n**, Same as **a**–**g**, but for monkey C ($n = 48$ neural datasets over $n = 4$ sessions). Statistical test details and exact $P$ values are provided in Supplementary Table 1.

more accurately learns behaviorally relevant neural dynamics (Fig. 3d,k). Second, this best performance was achieved using a significantly smaller state dimension with PSID than with NDM and RM, with a median dimension of 4 versus 12–30 (Fig. 3b,i). Third, we confirmed using numerical simulations that PSID accurately estimates the true dimension of behaviorally relevant neural dynamics, whereas NDM overestimates it (Supplementary Fig. 4a,b). Fourth, at the dimension estimated by PSID, NDM had a significantly lower decoding accuracy (Fig. 3c,j), which suggests that PSID more accurately identifies the low-dimensional behaviorally relevant state. This comparison and the more accurate decoding by PSID for all latent state dimensions (Fig. 3a,h) show the advantage of PSID for dynamic dimensionality reduction while preserving behavior information.

Next, in three analyses, we confirmed that these results still held if (1) explained variance was used instead of CC as the performance measure, (2) 1–4 Hz was added to the original 7 LFP power bands and (3) NDM was implemented using expectation maximization instead of SID. PSID revealed markedly lower dimension for behaviorally relevant neural dynamics ($P < 10^{-6}$, one-sided signed-rank, $n \geq 48$) and achieved better decoding even compared to the best decoding of NDM and RM using much higher-dimensional states ($P < 10^{-9}$, one-sided signed-rank, $n \geq 48$). Furthermore, PSID achieved more accurate decoding than NDM for neural predictions of behavior at multiple time steps into the future (Supplementary Fig. 5). We also performed a comparison with nondynamic dimension reduction using principal component analysis (PCA) or factor analysis, and with direct regression from neural activity to behavior without any dimension reduction. Again, we found similar results, with PSID achieving better decoding using lower-dimensional states (Supplementary Fig. 6). In two analyses, by (1) comparing the decoding accuracy versus neural reconstruction accuracy as the dimension of the latent state increased and (2) using canonical correlation analysis to quantify the similarity of latent states across different methods, we found that PSID uniquely prioritizes the extraction of behaviorally relevant dynamics in low-dimensional states and leaves the extraction of any residual neural dynamics for higher dimensions of states (Extended Data Fig. 5).

Similar results held for each individual joint, with PSID achieving better decoding than NDM and RM for almost all joints using significantly lower-dimensional states (Supplementary Fig. 7). Also, even when a subset of joints was provided to PSID during learning, its extracted latent states in the test data were still more predictive of the remaining joints that were unseen by PSID compared to NDM-extracted latent states (Extended Data Fig. 6). This result suggests that the learned behaviorally relevant neural dynamics may generalize across different behavioral measurements related to the same task.

We next found that the dimensionality of behaviorally relevant neural dynamics was much lower than that of neural dynamics or joint-angle dynamics, which suggests that the low dimensionality that PSID finds is not simply because either neural or behavior dynamics are just as low dimensional. We found the latent state dimension required to achieve the best self-prediction of neural or behavioral signals using their own past, and defined it as the total neural or behavior dynamics dimension, respectively (Methods).

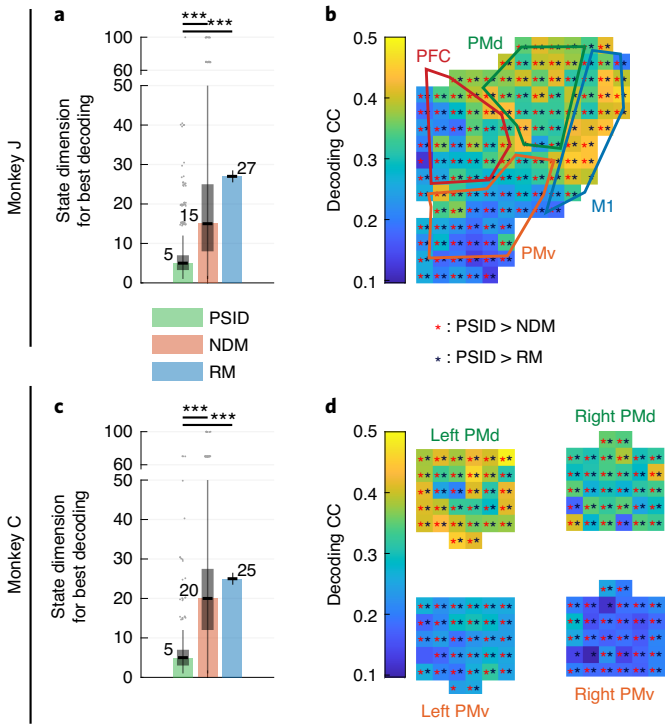



**Fig. 4 | PSID more accurately learns the behaviorally relevant neural dynamics in each recording channel across premotor, primary motor and prefrontal areas. a**, The state dimension used by each method to achieve the best decoding using the neural features from each recoding channel separately ($n = 137 \times 7 = 595$ channel datasets). As in Fig. 3b, for PSID and NDM, for each channel, 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 using that channel among all latent state dimensions. Bars, boxes and asterisks are defined as in Fig. 3b. **b**, Cross-validated CC values between the decoded and true joint angles for PSID. Asterisks mark channels for which PSID resulted in significantly ($P < 0.05$, one-sided signed-rank, $n = 35$ test folds per channel) better decoding compared to NDM (red asterisk) or RM (black asterisk). The latent state dimension for each method is chosen as in **a**. **c**, Same as **a**, but for monkey C ($n = 128 \times 4 = 512$ channel datasets). **d**, Similar to **b**, but for channels in monkey C ($n = 20$ test folds per channel), which had bilateral coverage (Methods). Statistical test details and exact *P* values are provided in Supplementary Table 1.

We confirmed in numerical simulations that this procedure correctly estimated the total latent state dimension in each signal (Supplementary Fig. 4c–e). In both monkeys, the median latent state dimension required for best self-prediction was at least 100 for neural activity (Fig. 3e,l) and was 40 for behavior joint-angles (Fig. 3f,m), which are both significantly larger than the behaviorally relevant neural dynamics dimension of four revealed by PSID (Fig. 3g,n). Moreover, the self-prediction of behavior from its own past was much better than decoding it from neural activity (Fig. 3a,f,h,m) and reached an almost perfect CC of 0.98 for both monkeys (Fig. 3f,m), which indicates that there are predictable dynamics in behavior that are not present in the recorded neural activity (corresponding to $\epsilon_k$ in Fig. 1a). These results are consistent with findings in prior studies showing that motor cortical activity can exhibit dynamics that are not related to behavior[2,9–14,28] and that parts of behavior dynamics cannot be predicted from the recorded neural activity[22,23,30,32]. Taken together, these results suggest that beyond the low-dimensional behaviorally relevant neural dynamics extracted via PSID, both recorded neural activity and behavior have significant additional dynamics that are predictable from their own past but unrelated to the other signal. PSID uniquely enables the dissociation of shared dynamics from those exclusive to one signal and reveals a much lower dimension for them (Fig. 3).

Finally, the above conclusions regarding the dimensionality of behaviorally relevant neural dynamics and decoding accuracy held irrespective of the exact behavioral signal, including for joint angles, angular velocities and 3D hand and elbow position (Supplementary Figs. 8 and 9). Last, while PSID does not have an explicit objective to separate within-trial versus trial-to-trial variability, we found that the latent states learned by PSID captured both types of variability (Supplementary Fig. 10).

**PSID-extracted dynamics are more informative of behavior for almost all recording channels across premotor, primary motor and prefrontal areas.** We found that PSID extracted more behaviorally relevant information from each recording channel rather than performing an implicit channel selection by discarding some channels with no behaviorally relevant information (Fig. 4). When modeling channels separately, PSID achieved significantly better decoding of behavior in at least 96% and 98% of individual channels compared to NDM and RM, respectively (Fig. 4b,d) while using significantly lower-dimensional states (Fig. 4a,c). Similar results held for explained variance as the performance measure (Supplementary Fig. 11). These results suggest that almost all channels contained behaviorally relevant dynamics and that PSID more accurately modeled these dynamics. Finally, we modeled groups of channels within each anatomical region using PSID separately and quantified the decoding accuracy for each individual joint (Supplementary Fig. 12). Across the joints, regions were predictive in the following order: M1 > PMd > PFC > PMv (monkey J; $P < 10^{-3}$, one-sided signed-rank, $n = 27$) and left PMd > right PMd > left PMv > right PMv (monkey C; $P < 10^{-4}$, one-sided signed-rank, $n = 25$). Interestingly, the ipsilateral (that is, right) areas in monkey C were predictive, with the ipsilateral PMd being more predictive than the contralateral PMv.

**PSID reveals behaviorally relevant rotational dynamics that otherwise go unnoticed.** Reducing the dimension of neural population activity and finding its low-dimensional representation are essential for visualizing and characterizing the relationship of neural dynamics to behavior[11,14,15,21–23]. PSID can be particularly beneficial for such applications as it can perform dimensionality reduction while preserving behaviorally relevant neural dynamics. To demonstrate this, as a special case of Fig. 3a,h, we used PSID and NDM to extract 2D representations of neural dynamics (Fig. 5), which is commonly done to visualize neural dynamics on planes[11,14,21–23]. Using both PSID and NDM, we fitted models with 2D latent states in the training data (Fig. 1b) and then used the associated Kalman filter to extract the two latent states purely from neural activity in the test data (Fig. 5b). We then plotted these two states against each other during reach and return movement epochs (Fig. 5c,d,f,g). Note that in the test data, the only input to both PSID and NDM is the same neural activity and that the learned Kalman filter for each method is continuously applied to the entire duration of the test data without breaking it into epochs (Fig. 5b).

In both monkeys, both PSID and NDM extracted neural states from LFP power activity that exhibited rotational dynamics. However, surprisingly, a clear difference emerged in the rotations uncovered by PSID compared to NDM. During the return epochs, the 2D neural dynamics extracted using PSID showed a rotation in the opposite direction of the rotation during the reach epochs (Fig. 5c,f). In contrast, similar to neural rotations extracted in prior works using PCA during forward versus backward cycling[11] or using NDM during 2D cursor control[22], neural dynamics extracted using NDM showed rotations in the same direction during both

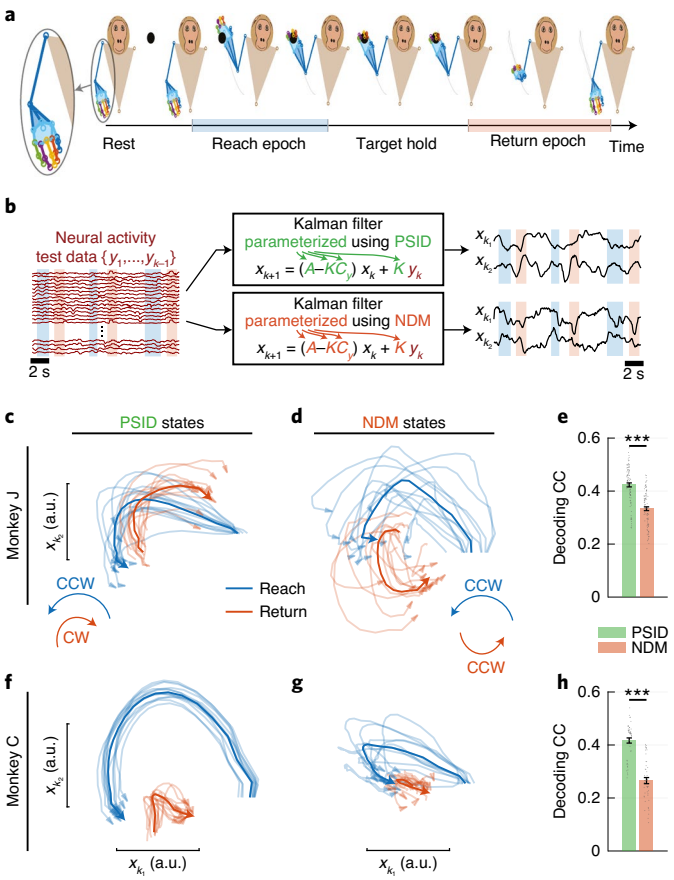



**Fig. 5 | PSID reveals rotational neural dynamics with opposite directions during 3D reach and return movements, which is not found by standard methods. a**, Example reach and return epochs in the task defined as periods of movement toward the target object and back from it, respectively. Pictures were recreated using the 3D-tracked markers and are from a view facing the monkey. **b**, Extraction of latent states for both PSID and NDM involves applying a Kalman filter to the neural activity. The difference is the value of the parameters in the Kalman filter, which are learned differently for the two methods using the training data as shown in Fig. 1b. **c**, The latent neural state dynamics during 3D reach (blue) and return (red) movements found by PSID with 2D latent states ($n_x = n_1 = 2$). We plot the states starting at the beginning of a reach/return movement epoch; the arrows mark the end of the movement epoch. Light lines show the average trace over trials in each dataset and dark lines show the overall average trace across datasets. The direction of rotation is noted by CW for clockwise or CCW for counterclockwise. States have arbitrary units (a.u.). **d**, Same as **c**, but using NDM with 2D latent states ($n_x = 2$). **e**, Cross-validated CC values between the decoded and true joint angles, decoded with the latent states extracted using PSID and NDM in **c** and **d**. Bars, whiskers and asterisks are defined as in Fig. 3d ($n = 91$ datasets). **f**–**h**, Same as **c**–**e**, but for monkey C ($n = 48$ datasets). Decoding accuracy using the PSID-extracted 2D states was only 7% (monkey J) or 4% (monkey C) worse than the best PSID decoding, whereas decoding accuracy using NDM-extracted 2D states was 23% (monkey J) or 30% (monkey C) worse than the best NDM decoding (Fig. 3a,h). Statistical test details and exact *P* values are provided in Supplementary Table 1.

reach and return epochs (Fig. 5d,g). For both PSID and NDM, their Kalman filter is not aware of when reach or return epochs happen and has the same parameter values during reach and return epochs. Yet, from the same high-dimensional neural activity, PSID and NDM extracted these different rotational patterns (an illustrative conceptual example of how different rotational patterns can coexist in high-dimensional dynamics is provided in Supplementary Video 1). As the behavior involves opposite directions of movement during reach and return epochs, these results intuitively suggest that the PSID-extracted states that reverse the direction of their rotation are more behaviorally relevant (Fig. 5a). Indeed, this suggestion was quantitatively confirmed since the PSID-extracted states also explained behavior significantly better than NDM-extracted ones and led to significantly better decoding (Fig. 5e,h). Moreover, PSID-extracted states maintained their more accurate decoding even when we used support vector regression (SVR) to learn nonlinear mappings from the extracted latent states to behavior (Supplementary Fig. 13). These results suggest that PSID can achieve dynamic dimensionality reduction and modeling in neural activity while preserving behaviorally relevant neural dynamics that may otherwise be missed.

These distinct rotational patterns also separately appeared in different anatomical regions (Extended Data Fig. 7) and when the behavior signal was taken to be different subsets of upper-extremity joints or inferred muscle activations (Supplementary Fig. 14). This result again suggests that PSID more accurately explains the encoding of various relevant behavioral signals within the task. Finally, in three additional analyses we found that (1) NDM with expectation maximization instead of SID, (2) PCA and (3) jPCA, which is another behavior agnostic method designed for extracting rotational dynamics[21], also extracted rotations similar to NDM that did not change direction during reach and return (Extended Data Fig. 8). We emphasize that beyond the above 2D results for visualization, the advantage of PSID over NDM when performing dimensionality reduction held across all dimensions (Fig. 3a,h).

**PSID reveals markedly lower dimensionality for behaviorally relevant neural dynamics and extracts them more accurately in motor cortex spiking activity.** We also applied PSID to multiunit spiking activity recorded during the same 3D reach, grasp and return task by taking neural activity $y_k$ from equation (1) to represent the spike counts in consecutive time bins (Extended Data Fig. 4b and Methods). The results reported above for LFP activity generalized to spiking activity (Fig. 6). To demonstrate that the advantages of PSID in modeling behaviorally relevant neural dynamics stem from its consideration of behavior during learning with its novel two-stage approach and hold irrespective of model structure, we also compared the results to example nonlinear NDM methods. These nonlinear NDMs, similar to linear NDM, model neural dynamics regardless of their relevance to behavior. As the first example nonlinear NDM, we used the model from linear NDM and used SVR to learn nonlinear mappings from the latent states to behavior. As the second example nonlinear NDM, we used a method based on RNNs termed latent factor analysis via dynamical systems (LFADS), which was recently successfully applied to spiking activity[23] (Methods). In both monkeys, PSID revealed that behaviorally relevant dynamics in spiking activity are markedly lower-dimensional than is implied by RM and NDM, whether linear or nonlinear (Fig. 6b,f). Moreover, PSID learned these behaviorally relevant dynamics more accurately compared to both linear and nonlinear NDM with the same dimension (Fig. 6a,c,e,g), and even compared to linear or nonlinear NDM and RM with much higher-dimensional states (Fig. 6d,h). These results also held if instead of CC, explained variance was used ($P < 0.006$, one-sided signed-rank, $n \geq 16$). Similar results held when we allowed the RNN-based nonlinear NDM to use a Poisson process observation model with faster time steps compared to PSID and/or to always use much higher-dimensional latent states (Extended Data Fig. 9).

**PSID results generalize to other behavioral tasks, neural signal types and brain regions.** To demonstrate the general utility of PSID

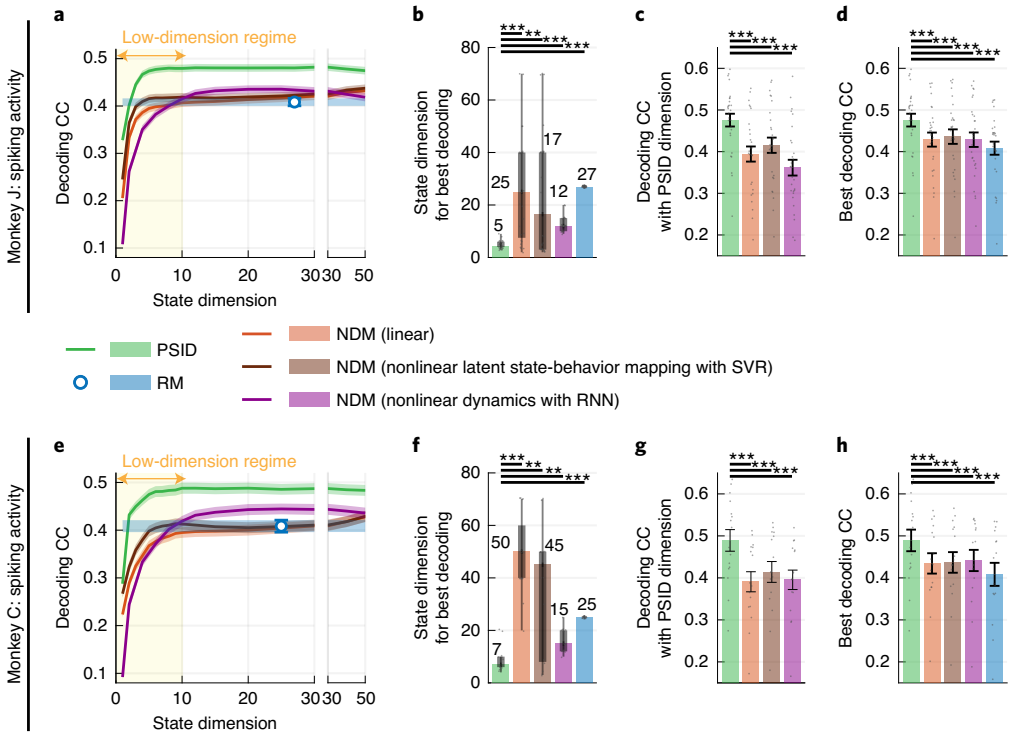



**Fig. 6 | PSID reveals a markedly lower dimension for behaviorally relevant neural dynamics and extracts them more accurately in motor cortex population spiking activity. a–h**, The figure convention is the same as in Fig. 3a–d (including definitions of bars, boxes and whiskers), shown here for modeling neural spiking activity (Methods) for monkey J (**a**–**d**) and monkey C (**e**–**h**). To show that the PSID advantage stems from its novel two-stage learning algorithm that considers behavior and therefore holds regardless of NDM model structure, results are also shown for two example nonlinear NDM methods for comparison: (1) linear NDM with nonlinear SVR mapping from latent state to behavior; (2) RNN-based NDM with nonlinear dynamics using a method termed LFADS (Methods). Similar results held for several additional configurations of the RNN-based nonlinear NDM method as provided in Extended Data Fig. 9. $n = 28$ datasets for monkey J and $n = 16$ datasets for monkey C. $^{**}P < 0.005$, $^{***}P < 0.0005$. Statistical test details and exact $P$ values are provided in Supplementary Table 1.

for different behavioral tasks, neural signal types and brain regions, we next applied it to PFC neural activity recorded in a second experiment in which two monkeys performed saccadic eye movements[36] (Extended Data Fig. 10 and Methods). Here, we modeled the raw LFP and took the 2D eye position as the behavior. Compared to standard NDM, PSID revealed markedly lower-dimensional latent representations (8–15 versus 70) for behaviorally relevant neural dynamics (Extended Data Fig. 10b,f) and more accurately learned these dynamics both at the same dimension (Extended Data Fig. 10c,g) and when NDM used much higher-dimensional latent states (Extended Data Fig. 10d,h). Finally, because the RM dimension is equal to the behavior dimension, here, RM extracted a 2D state that resulted in significantly worse decoding of behavior (Extended Data Fig. 10d,h).

## Discussion

Here, we developed PSID, a novel algorithm for dissociating and modeling behaviorally relevant neural dynamics. PSID ensures that these neural dynamics are not missed by considering both the measured behavior and the neural activity in a novel two-stage learning algorithm: the first stage exclusively learns behaviorally relevant dynamics and the optional second stage learns any remaining dynamics (Extended Data Fig. 1). Prior NDM methods with linear dynamics[6,15,22,30,33], generalized linear dynamics[30,32,37] or nonlinear dynamics[23] are all agnostic to behavior in learning the dynamics. Thus, PSID can uniquely uncover behaviorally relevant dynamics that may otherwise be discarded, as evidenced both by its better decoding at any state dimension (Figs. 3 and 6) and its discovery of distinct reversed rotational dynamics in neural activity during return epochs in our motor task (Fig. 5).

Prior work has reported low-dimensional rotational neural dynamics during different motor tasks, often involving 2D control of a cursor[14,21–23]. Interestingly, in our 3D task, NDM, PCA and jPCA all extracted neural rotations that had the same direction during reach and return epochs—similar to prior work with a center-out 2D cursor control task[22]—whereas PSID extracted neural rotations in opposite directions. Critically, the PSID rotations were significantly more predictive of behavior, which demonstrates that while both types of rotational dynamics were present in high-dimensional neural activity (Supplementary Video 1), PSID uniquely revealed those that were more behaviorally relevant. Future applications of PSID to other tasks and brain regions may similarly reveal behaviorally relevant features of neural dynamics that may otherwise go unnoticed.

Motor cortical activity strongly encodes movement, thus enabling motor brain–machine interfaces (BMIs)[1,3,4,24]. Given this strong encoding, both RM, which models behavior dynamics agnostic to neural activity[3,24], and NDM, which models neural dynamics agnostic to behavior[22,30,32,33], have been successful in decoding movement. Nevertheless, PSID still significantly outperformed these methods in motor decoding and did so using markedly lower-dimensional states (Figs. 3 and 6). Many brain functions such as memory[26] and mood[6] or brain dysfunctions such as epileptic seizures[8] could have more distributed or less targetable representations in neural activity. Using PSID in such applications may prove even more beneficial since the activity likely contains more dynamics that are unrelated to the measured behavior.

PSID is also a dimensionality reduction method that is dynamic, that is, it models the temporal structure in neural activity (equation (1)); therefore, it can aggregate information over time to optimally extract low-dimensional representations of neural activity that preserve behaviorally relevant dynamics. PSID can do this because—unlike prior dynamic dimensionality reduction methods such as Gaussian process factor analysis[38], LFADS[23] and latent state space modeling[6,15,22,30,32,33,37]—PSID takes behavior into account during learning to ensure behaviorally relevant neural dynamics are not lost. As such, PSID can benefit studies that investigate neural mechanisms underlying a behavior of interest. For example, prior works have reported that variables with 10–30 dimensions can sufficiently explain motor cortical activity using various dimensionality reduction methods, including PCA and the above dynamic methods[2,11,13,18,22,23,30,37,38]. However, unlike PSID, these methods did not aim to explicitly dissociate the behaviorally relevant parts of neural dynamics. Here, PSID revealed a dimension of around 4–7 for behaviorally relevant neural dynamics, which was significantly lower than the dimension of 12–50 implied here by other linear, nonlinear, dynamic and nondynamic dimensionality reduction methods while also being more predictive of behavior (Figs. 3 and 6 and Supplementary Fig. 6). These results demonstrate the utility of PSID for accurately estimating the dimensionality of behaviorally relevant neural dynamics, which is a fundamental sought-after question across neuroscience[2,13,18].

For datasets with discrete classes of behavioral conditions, several nondynamic dimensionality reduction methods such as linear discriminant analysis[15] and demixed PCA[26] can find low-dimensional projections of neural activity that are suitable for dissociating those classes[9,10]. However, unlike PSID, these methods are not applicable to continuous behavioral measurements, such as movements, and are not dynamic. For continuous behavioral variables, several nondynamic methods, such as reduced rank regression[13,39], partial least squares regression[40], targeted dimensionality reduction[7,36] and canonical correlation analysis[40], can build linear projections of one signal to another, for example, to find the linear subspace of largest covariations between neural activity in two cortical areas[39]. However, being nondynamic, these methods do not model the temporal patterns of neural activity or aggregate information over time in recovering the dynamics or in decoding, which are particularly important in studies of temporally structured behaviors such as movements[3,24] or speech[5]. Thus, PSID uniquely enables the extraction of behaviorally relevant low-dimensional representations for neural activity by being dynamic and applicable to continuous behaviors.

PSID uses a linear state space model whereby observations are linear functions of the latent state. A linear observation model can be used for both LFP activity[6,32,33,41] and spike counts[3,22,24], as we also showed here. Recent studies have shown that for the RM framework, modeling spikes as binary events using generalized linear state space models with nonlinearly linked Poisson observations can be more accurate in BMIs[42,43]. Consistent with these studies[42,43], in one of the two subjects, using a Poisson observation model improved the nonlinear NDM decoding, even though PSID still outperformed it (Extended Data Fig. 9). A variation of NDM using SID has been developed for Poisson observation models[37], and an interesting future direction is to generalize PSID to learn such generalized linear models with behaviorally relevant latent states. Moreover, given the growing interest in modeling simultaneous spike–field activity[32,41,44–46] and new multiscale approaches[32,41,45,46], developing a multiscale or nonlinear version of PSID that can model observations with linear and nonlinear relations to the states and from multiple timescales together would be interesting. Finally, PSID models behavior as a Gaussian process and could therefore be approximately applicable to ordinal categorical behaviors. An interesting future direction is to extend PSID to non-ordinal categorical behaviors.

PSID may also enhance future neurotechnologies such as BMIs[1,3,4] and closed-loop deep-brain stimulation systems[4,8] in decoding and modulating behaviorally relevant brain states—especially those encoded across distributed brain networks that are likely involved in various functions and thus exhibit more dynamics that are unrelated to the measured behavior[4,6,12,13,27,31,34]. Furthermore, PSID achieved maximal decoding accuracy using markedly lower-dimensional states. As controllers designed for models with lower-dimensional states are generally more robust (Methods), PSID could benefit model-based control of brain functions with electrical or optogenetic stimulation[4,31,34]. Moreover, PSID is not only computationally efficient in learning the model, it also has a Kalman filter decoder that can process neural activity causally and efficiently. These capabilities are essential for real-time closed-loop applications[1,3,4,24]. Finally, developing adaptive learning methods that track changes in behaviorally relevant neural dynamics, for example, due to learning or stimulation-induced plasticity[3,47,48], and can appropriately select the adaptation learning rate[49] are important future directions.

Here, we described PSID as a tool for extracting and modeling behaviorally relevant dynamics from neural activity. In this application, neural activity is taken as the primary signal and behavior is taken as a secondary signal encoded by the primary signal. Nevertheless, the mathematical derivation of PSID and numerical simulations here do not depend on the nature of the two signals. For example, PSID could be applied to neural signals from two brain areas, in which case it will prioritize and dissociate their shared dynamics from those that are exclusive to one area. The two signals in PSID could even be from different sources. For example, when studying interpersonal neural and behavioral synchrony[50], applying PSID to neural and/or behavioral signals that are synchronously recorded from different individuals may enable the extraction and modeling of common dynamics between them. In general, when two signals are suspected to have shared dynamics (for example, because they may be driven by common dynamic inputs), PSID can be used to extract and model the shared dynamics.

In conclusion, this novel PSID modeling algorithm can advance our understanding of how behaviorally measurable brain functions are encoded in neural activity across broad tasks and brain regions. Also, PSID may prove to be particularly beneficial for studying distributed brain functions such as those involved in emotion, memory and social behaviors.

## Online content

Any methods, additional references, Nature Research reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41593-020-00733-0.

Received: 4 September 2019; Accepted: 2 October 2020;
Published online: 9 November 2020

## References

1. Schwartz, A. B., Cui, X. T., Weber, D. J. & Moran, D. W. Brain-controlled interfaces: movement restoration with neural prosthetics. *Neuron* **52**, 205–220 (2006).
2. Shenoy, K. V., Sahani, M. & Churchland, M. M. Cortical control of arm movements: a dynamical systems perspective. *Annu. Rev. Neurosci.* **36**, 337–359 (2013).
3. Shanechi, M. M. Brain–machine interface control algorithms. *IEEE Trans. Neural Syst. Rehabil. Eng.* **25**, 1725–1734 (2017).
4. Shanechi, M. M. Brain–machine interfaces from motor to mood. *Nat. Neurosci.* **22**, 1554–1564 (2019).
5. Herff, C. & Schultz, T. Automatic speech recognition from neural signals: a focused review. *Front. Neurosci.* **10**, 429 (2016).
6. Sani, O. G. et al. Mood variations decoded from multi-site intracranial human brain activity. *Nat. Biotechnol.* **36**, 954–961 (2018).

7. Mante, V., Sussillo, D., Shenoy, K. V. & Newsome, W. T. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature* **503**, 78–84 (2013).
8. Hoang, K. B., Cassar, I. R., Grill, W. M. & Turner, D. A. Biomarkers and stimulation algorithms for adaptive brain stimulation. *Front. Neurosci.* **11**, 564 (2017).
9. Kaufman, M. T. et al. The largest response component in the motor cortex reflects movement timing but not movement type. *eNeuro* https://doi.org/10.1523/ENEURO.0085-16.2016 (2016).
10. Gallego, J. A. et al. Cortical population activity within a preserved neural manifold underlies multiple motor behaviors. *Nat. Commun.* **9**, 4233 (2018).
11. Russo, A. A. et al. Motor cortex embeds muscle-like commands in an untangled population response. *Neuron* **97**, 953–966.e8 (2018).
12. Allen, W. E. et al. Thirst regulates motivated behavior through modulation of brainwide neural population dynamics. *Science* **364**, eaav3932 (2019).
13. Stringer, C. et al. Spontaneous behaviors drive multidimensional, brainwide activity. *Science* **364**, eaav7893 (2019).
14. Susilaradeya, D. et al. Extrinsic and intrinsic dynamics in movement intermittency. *eLife* **8**, e40145 (2019).
15. Cunningham, J. P. & Yu, B. M. Dimensionality reduction for large-scale neural recordings. *Nat. Neurosci.* **17**, 1500–1509 (2014).
16. Gallego, J. A., Perich, M. G., Miller, L. E. & Solla, S. A. Neural manifolds for the control of movement. *Neuron* **94**, 978–984 (2017).
17. Remington, E. D., Egger, S. W., Narain, D., Wang, J. & Jazayeri, M. A dynamical systems perspective on flexible motor timing. *Trends Cogn. Sci.* **22**, 938–952 (2018).
18. Sadtler, P. T. et al. Neural constraints on learning. *Nature* **512**, 423–426 (2014).
19. Gao, P. & Ganguli, S. On simplicity and complexity in the brave new world of large-scale neuroscience. *Curr. Opin. Neurobiol.* **32**, 148–155 (2015).
20. Gao, P. et al. A theory of multineuronal dimensionality, dynamics and measurement. Preprint at *bioRxiv* https://doi.org/10.1101/214262 (2017).
21. Churchland, M. M. et al. Neural population dynamics during reaching. *Nature* **487**, 51–56 (2012).
22. Kao, J. C. et al. Single-trial dynamics of motor cortex and their applications to brain–machine interfaces. *Nat. Commun.* **6**, 7759 (2015).
23. Pandarinath, C. et al. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nat. Methods* **15**, 805–815 (2018).
24. Kao, J. C., Stavisky, S. D., Sussillo, D., Nuyujukian, P. & Shenoy, K. V. Information systems opportunities in brain–machine interface decoders. *Proc. IEEE* **102**, 666–682 (2014).
25. Wallis, J. D. Decoding cognitive processes from neural ensembles. *Trends Cogn. Sci.* **22**, 1091–1102 (2018).
26. Kobak, D. et al. Demixed principal component analysis of neural population data. *eLife* **5**, e10989 (2016).
27. Svoboda, K. & Li, N. Neural mechanisms of movement planning: motor cortex and beyond. *Curr. Opin. Neurobiol.* **49**, 33–41 (2018).
28. Gründemann, J. et al. Amygdala ensembles encode behavioral states. *Science* **364**, eaav8736 (2019).
29. Wu, W., Kulkarni, J. E., Hatsopoulos, N. G. & Paninski, L. Neural decoding of hand motion using a linear state-space model with hidden states. *IEEE Trans. Neural Syst. Rehabil. Eng.* **17**, 370–378 (2009).
30. Aghagolzadeh, M. & Truccolo, W. Inference and decoding of motor cortex low-dimensional dynamics via latent state-space models. *IEEE Trans. Neural Syst. Rehabil. Eng.* **24**, 272–282 (2016).
31. Yang, Y., Connolly, A. T. & Shanechi, M. M. A control-theoretic system identification framework and a real-time closed-loop clinical simulation testbed for electrical brain stimulation. *J. Neural Eng.* **15**, 066007 (2018).
32. Abbaspourazad, H., Hsieh, H. & Shanechi, M. M. A multiscale dynamical modeling and identification framework for spike–field activity. *IEEE Trans. Neural Syst. Rehabil. Eng.* **27**, 1128–1138 (2019).
33. Yang, Y., Sani, O. G., Chang, E. F. & Shanechi, M. M. Dynamic network modeling and dimensionality reduction for human ECoG activity. *J. Neural Eng.* **16**, 056014 (2019).
34. Yang, Y. et al. Model-based prediction of large-scale brain network dynamic response to direct electrical stimulation. *Nat. Biomed. Eng.* (in the press).
35. Van Overschee, P. & De Moor, B. *Subspace Identification for Linear Systems* (Springer US, 1996).
36. Markowitz, D. A., Curtis, C. E. & Pesaran, B. Multiple component networks support working memory in prefrontal cortex. *Proc. Natl Acad. Sci. USA* **112**, 11084–11089 (2015).
37. Buesing, L., Macke, J. H. & Sahani, M. in *Advances in Neural Information Processing Systems 25* (eds Pereira, F. et al) 1682–1690 (Curran Associates, 2012).
38. Yu, B. M. et al. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. *J. Neurophysiol.* **102**, 614–635 (2009).
39. Semedo, J. D., Zandvakili, A., Machens, C. K., Yu, B. M. & Kohn, A. Cortical areas interact through a communication subspace. *Neuron* **102**, 249–259.e4 (2019).
40. Cunningham, J. P. & Ghahramani, Z. Linear dimensionality reduction: survey, insights, and generalizations. *J. Mach. Learn. Res.* **16**, 2859–2900 (2015).
41. Hsieh, H.-L., Wong, Y. T., Pesaran, B. & Shanechi, M. M. Multiscale modeling and decoding algorithms for spike–field activity. *J. Neural Eng.* **16**, 016018 (2018).
42. Shanechi, M. M., Orsborn, A. L. & Carmena, J. M. Robust brain–machine interface design using optimal feedback control modeling and adaptive point process filtering. *PLoS Comput. Biol.* **12**, e1004730 (2016).
43. Shanechi, M. M. et al. Rapid control and feedback rates enhance neuroprosthetic control. *Nat. Commun.* **8**, 13825 (2017).
44. Stavisky, S. D., Kao, J. C., Nuyujukian, P., Ryu, S. I. & Shenoy, K. V. A high performing brain–machine interface driven by low-frequency local field potentials alone and together with spikes. *J. Neural Eng.* **12**, 036009 (2015).
45. Bighamian, R., Wong, Y. T., Pesaran, B. & Shanechi, M. M. Sparse model-based estimation of functional dependence in high-dimensional field and spike multiscale networks. *J. Neural Eng.* **16**, 056022 (2019).
46. Wang, C. & Shanechi, M. M. Estimating multiscale direct causality graphs in neural spike–field networks. *IEEE Trans. Neural Syst. Rehabil. Eng.* **27**, 857–866 (2019).
47. Yang, Y. et al. Developing a personalized closed-loop controller of medically-induced coma in a rodent model. *J. Neural Eng.* **16**, 036022 (2019).
48. Ahmadipour, P., Yang, Y., Chang, E. F. & Shanechi, M. M. Adaptive tracking of human ECoG network dynamics. *J. Neural Eng.* https://doi.org/10.1088/1741-2552/abae42 (2020).
49. Hsieh, H.-L. & Shanechi, M. M. Optimizing the learning rate for adaptive estimation of neural encoding models. *PLoS Comput. Biol.* **14**, e1006168 (2018).
50. Yun, K., Watanabe, K. & Shimojo, S. Interpersonal body and neural synchronization as a marker of implicit social interaction. *Sci. Rep.* **2**, 959 (2012).

## Methods

**Dynamic model.** *Model formulation.* We used a dynamic linear state space model to describe the temporal evolution of neural activity and behavior as follows:

$$\begin{cases} x^s_{k+1} = Ax^s_k + w_k \\ y_k = C_y x^s_k + v_k \\ z_k = C_z x^s_k + \epsilon_k \end{cases} \quad (2)$$

Here, $k$ specifies the time index, $y_k \in \mathbb{R}^{n_y}$ is the recorded neural activity, $z_k \in \mathbb{R}^{n_z}$ is the behavior (for example, movement kinematics), $x^s_k \in \mathbb{R}^{n_x}$ is the latent dynamic state variable that drives the recorded neural activity $y_k$ and can also drive the behavior $z_k$. As we show in the next section, with a change of basis, equation (2) can be written in an equivalent form that separates the behaviorally relevant and irrelevant components of $x^s_k$, as we write in equation (1) and more explicitly below in equation (4). $w_k \in \mathbb{R}^{n_x}$ and $v_k \in \mathbb{R}^{n_y}$ are zero-mean white noises that are independent of $x^s_k$; that is, $E\{x^s_k w_k^T\} = 0$ and $E\{x^s_k v_k^T\} = 0$, respectively, with the following cross-correlations:

$$E\left\{ \begin{bmatrix} w_k \\ v_k \end{bmatrix} \begin{bmatrix} w_k^T & v_k^T \end{bmatrix} \right\} \triangleq \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \quad (3)$$

$\epsilon_k \in \mathbb{R}^{n_z}$ is a general random process denoting the variations of $z_k$ that are not generated by $x^s_k$; therefore, these variations are not present in the recorded neural activity. Thus, we only assume that $\epsilon_k$ is zero-mean and independent of $x^s_k$ (that is, $E\{x^s_k \epsilon_k^T\} = 0$) and the other noises (that is, $E\{w_{k'} \epsilon_k^T\} = 0$ and $E\{v_{k'} \epsilon_k^T\} = 0$ for any $k'$), but we do not make any assumptions about the dynamics of $\epsilon_k$. In fact, $\epsilon_k$ does not need to be white and can be any general non-white (colored) random process. Note that $\epsilon_k$ is also independent of $y_k$ (since it is independent of $x^s_k$ and $v_k$), thus, observing $y_k$ does not provide any information about $\epsilon_k$. Due to the zero-mean assumption for noise statistics, it is easy to show that $x^s_k$, $y_k$ and $z_k$ are also zero-mean, which implies that in preprocessing, the mean of $y_k$ and $z_k$ should be subtracted from them and later added back to any model predictions if needed. The parameters ($A$, $C_y$, $C_z$, $Q$, $R$ and $S$) fully specify the model in equation (2) (if statistical properties of $\epsilon_k$ are also of interest, another set of latent state space parameters can be used to model it; Supplementary Note 2). There are other sets of parameters that can equivalently and fully specify the model; specifically, the set of parameters $(A, C_y, C_z, G_y, \Sigma_y \text{ and } \Sigma_x)$ with $G_y \triangleq E\{x^s_{k+1} y_k^T\}$, $\Sigma_y \triangleq E\{y_k y_k^T\}$ and $\Sigma_x \triangleq E\{x^s_k x^{sT}_k\}$ can also fully characterize the model and is more suitable for evaluating learning algorithms (Supplementary Note 3).

*Definition of behaviorally relevant and behaviorally irrelevant latent states.* $x^s_k$ is a latent state that represents all dynamics in the neural activity $y_k$, which could be due to various internal brain processes, including those related to the measured behavior of interest, other behaviors and brain functions, or internal stat es[2,9–14,27,51–57]. Without loss of generality, it can be shown (Supplementary Note 4) that equation (2) can be equivalently written in a different basis as

$$\begin{cases} \begin{bmatrix} x^{(1)}_{k+1} \\ x^{(2)}_{k+1} \end{bmatrix} = \begin{bmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x^{(1)}_k \\ x^{(2)}_k \end{bmatrix} + \begin{bmatrix} w^{(1)}_k \\ w^{(2)}_k \end{bmatrix} \\ y_k = \begin{bmatrix} C_{y_1} & C_{y_2} \end{bmatrix} \begin{bmatrix} x^{(1)}_k \\ x^{(2)}_k \end{bmatrix} + v_k \quad , \quad x_k = \begin{bmatrix} x^{(1)}_k \\ x^{(2)}_k \end{bmatrix} \\ z_k = \begin{bmatrix} C_{z_1} & 0 \end{bmatrix} \begin{bmatrix} x^{(1)}_k \\ x^{(2)}_k \end{bmatrix} + \epsilon_k \end{cases} \quad (4)$$

where $x^{(1)}_k \in \mathbb{R}^{n_1}$ is the minimal set of states that affect the specific measured behavior of interest and whose dimension $n_1$ is the rank of the behavior observability matrix (equation (42) in Supplementary Note 4). Thus, we refer to $x^{(1)}_k$ as the behaviorally relevant latent states and $x^{(2)}_k \in \mathbb{R}^{n_2}$ with $n_2 = n_x - n_1$ as the behaviorally irrelevant latent states. We interchangeably refer to the dimension of the latent states as the number of latent states (for example, $n_x$ is the total number of latent states or the total latent state dimension).

Equation (4) presents a general formulation of which special cases also include the models used in NDM and RM. If we assume that all latent states can contribute to behavior ($n_1 = n_x$ and $n_2 = 0$), equation (4) reduces to the linear state space model typically used to model the dynamics of neural activity in NDM[6,22,30,33]. If we further take $C_z$ to be the identity matrix and $\epsilon_k = 0$, the state will be set to the behavior $z_k$, and equation (4) reduces to the linear state space model used in RM[3,24,58–60]. Thus, if the assumptions of standard NDM (that is, all latent states can drive both neural activity and behavior) or RM (that is, behavior drives neural activity) hold better for a given dataset, PSID would still identify these standard models because the solution would still fall within the model in equation (4) used by PSID.

*The learning problem.* In the general learning problem, given training time-series $\{y_k : 0 \le k < N\}$ and $\{z_k : 0 \le k < N\}$, the aim is to find the dimension of the latent state $n_x$ and all model parameters $(A, C_y, C_z, G_y, \Sigma_y \text{ and } \Sigma_x)$ that generate the data according to equation (2) or equivalently equation (4). Unlike prior work, here, we critically required an identification algorithm that can dissociate the behaviorally relevant and irrelevant latent states in neural activity, and can prioritize the identification of the behaviorally relevant latent states (that is, $x^{(1)}_k$ from equation (4)). As we show in the section on PSID below (see also Extended Data Fig. 1), we achieved this prioritization by developing a novel two-stage analytical learning algorithm that can extract the behaviorally relevant latent states in its first stage without the need to extract or model the behaviorally irrelevant states; this prioritization ability led to the extraction of latent states from neural activity that are both lower-dimensional and at the same time more accurate in describing behavior. If desired, the behaviorally irrelevant latent states can then be extracted in the optional second stage. Prioritizing behaviorally relevant latent states in PSID ensured that these states are not discarded in the model even when performing dimensionality reduction; that is, when identifying a model with fewer states than the true $n_x$ (Fig. 2). Furthermore, without such prioritization, one would need models with higher-dimensional latent states to ensure that behaviorally relevant latent states are included within them, which would require more training data. Thus, given the same finite training data, PSID is more accurate compared with standard algorithms, which need to identify models with a higher-dimensional latent state and then potentially discard some state dimensions (Extended Data Fig. 3). Finally, the ability of PSID to directly learn models with low-dimensional behaviorally relevant latent states is beneficial for closed-loop controller design for controlling brain states[4,31,47] since controllers with reduced state dimensions are generally more robust[61].

As described next, once the model is learned, the extraction of behaviorally relevant states in test or new data is based purely on neural activity and is done with a Kalman filter.

*The decoding problem.* Given the model parameters, the prediction (or decoding) problem is to provide the best estimate of behavior $z_{k+1}$ given the past neural activity $\{y_n : 0 \le n \le k\}$. Given the linear state space formulation of equation (2) and to achieve the minimum mean-square error, the best prediction of neural activity $y_{k+1}$ using $y_1$ to $y_k$ (that is, neural self-prediction) and, similarly, the best prediction of behavior $z_{k+1}$ using $y_1$ to $y_k$ (that is, neural decoding)—which we denote as $\hat{y}_{k+1|k}$ and $\hat{z}_{k+1|k}$, respectively—are obtained with the well-known recursive Kalman filter[62] (Supplementary Note 5). By reformulating equation (2) to describe neural activity and behavior in terms of the latent states extracted by the Kalman filter, we show that the best prediction of future behavior using past neural activity is both a linear function of the past neural activity (which is readily measurable) and a linear function of the latent state (which is not measurable; Supplementary Note 5). In learning the model in the training data, this key insight enabled us to first directly extract the behaviorally relevant latent states via a projection of future behavior onto past neural activity (without having the model parameters), and then use these extracted states to identify the model parameters (Supplementary Note 6). Once the model was learned, in test data, we extracted the latent states and decoded the behavior using a Kalman filter associated with the identified model parameters that operates on neural activity alone.

**PSID.** We developed a novel two-stage learning algorithm, named PSID, to identify the parameters of the general dynamic model in equation (4) using the training time-series $\{y_k : 0 \le k < N\}$ and $\{z_k : 0 \le k < N\}$ while prioritizing the learning of behaviorally relevant latent states $x^{(1)}_k$ and their dynamics—or, equivalently, the dynamics of $z_k$ that are predictable from $y_k$—that is, learning them first. A high-level summary of the algorithm is provided in Box 1 and Extended Data Fig. 1 and details are provided in Supplementary Note 1. A detailed derivation is provided in Supplementary Note 6. In this section, we provide an overview of the derivation.

In learning, PSID first extracts the latent states directly using the neural activity and behavior training data, and then identifies the model parameters using the extracted latent states. The latent states are extracted in two stages. The first stage of PSID projects the future behavior ($Z_f$) onto the past neural activity ($Y_p$)—equation (10) in Supplementary Note 1, denoted as $Z_f/Y_p$ in Fig. 1b—which we show extracts the behaviorally relevant latent states (Supplementary Note 6; the row space of this projection provides the states and can be found with a singular value decomposition). Once these behaviorally relevant latent states are extracted, the dynamic model parameters in equation (4) can be learned for a model that only includes behaviorally relevant latent states $x^{(1)}_k$ (Supplementary Note 6). Alternatively, and if desired, the optional second stage of PSID can be used to extract the behaviorally irrelevant latent states $x^{(2)}_k$ and then learn the model parameters for a model that also includes these behaviorally irrelevant latent states, as in equation (4). The second stage first finds the part of the future neural activity that is not explained by the extracted behaviorally relevant latent states; that is, it does not lie in the subspace spanned by these states. This is found by subtracting the orthogonal projection of future neural activity onto the extracted behaviorally relevant latent states (equation (19) in Supplementary Note 1). This second stage then projects this unexplained future neural activity onto the past neural activity to extract the behaviorally irrelevant latent states (equation (22) in Supplementary Note 1). Note that in both stages, the latent states that PSID extracts are present in the neural activity since PSID projects all future measurable quantities onto past neural activity; so, if a latent state is just present in behavior but not in neural activity, PSID will not extract it among the latent states for neural activity (Fig. 2). After all latent states—that is, depending on the user's choice, either only behaviorally relevant

states from the first stage or both behaviorally relevant and irrelevant states from both stages—are extracted, PSID then learns all model parameters from these states (Supplementary Note 6). As a special case, one can also extract no latent state from the first stage (that is, take $n_1 = 0$ in Box 1 to only use the second stage of PSID), in which case PSID reduces to standard SID[35,62] (Supplementary Note 7). It is worth noting that future work could also develop adaptive[63,64] versions of PSID by giving smaller weights to older data samples in the projection steps[48,65,66]. Overall, PSID provides a computationally efficient solution for identifying the parameters of the model in equation (4) that only involves a small number of matrix algebra and SVD operations (Supplementary Note 6); this is unlike iterative methods used for expectation maximization or for training RNNs, for example.

**Identification of model structure parameters for PSID and NDM.** For both PSID and NDM, the total number of latent states $n_x$ is a parameter of the model structure. When learning of all dynamics in the neural activity (regardless of their relevance to behavior) is of interest, we identified the appropriate value for this parameter using the following cross-validation procedure. We fit models with different values of $n_x$, and for each model, we computed the cross-validated accuracy of one-step-ahead prediction of neural activity $y_k$ using its past (equation (45) in Supplementary Note 5). This is referred to as neural self-prediction to emphasize that the input is the past neural activity itself, which is used to predict the value of neural activity at the current time step. We used Pearson's CC to quantify the self-prediction (averaged across dimensions of neural activity). We then identified the total neural latent state dimension $n_x$ as the value that reaches within 1 s.e.m. of the best possible neural self-prediction accuracy among all considered latent state dimensions. As shown with numerical simulations, using this approach with PSID or standard SID[35,62] for NDM accurately identified the total number of latent states (Supplementary Figs. 3a–c and 4c,e). We therefore used this procedure to quantify the total neural dynamics dimension in the monkey data (Fig. 3e,l). We also used the exact same procedure on the behavioral data using the behavior self-prediction to quantify the total behavior dynamics dimension in the monkey data (Fig. 3f,m).

To learn a model with PSID with a given latent state dimension $n_x$, we also needed to specify another model structure parameter $n_1$; that is, the dimension of $x_k^{(1)}$ in equation (4). To determine a suitable value for $n_1$, we performed an inner cross-validation within the training data and fit models with the given $n_x$ and with different candidate values for $n_1$. Among the considered values for $n_1$, we selected the final value $n_1^*$ as the value of $n_1$ that, within the inner cross-validation in the training data, maximized the accuracy for decoding behavior using neural activity in the training data (equation (45) in Supplementary Note 5). We quantified the decoding accuracy using CC (averaged across dimensions of behavior). As shown with numerical simulations, this approach accurately identified $n_1$ (Supplementary Fig. 3d,e). Thus, when fitting a model with any given latent state dimension $n_x$ using PSID, unless otherwise noted, we determined $n_1$ using an inner cross-validation as detailed above (for example, Figs. 3, 4 and 6, Extended Data Figs. 5, 6, 9 and 10 and Supplementary Figs. 3–12).

**Generating random models for numerical simulations.** To validate the identification algorithms with numerical simulations, we generated random models with the following procedure. Dimensions of $y_k$ and $z_k$ were randomly selected with uniform probability from the following ranges: $5 \le n_y, n_z \le 10$. The full latent state dimension was selected with uniform probability from $1 \le n_x \le 10$, and then the number of states driving behavior ($n_1$) was selected with uniform probability from $1 \le n_1 \le n_x$. We then randomly generated matrices with consistent dimensions to be used as the model parameters $A$, $C_y$, $C_z$, $Q$, $R$ and $S$ (Supplementary Note 8). Specifically, the eigenvalues of $A$ were randomly selected from the unit circle and $n_1$ of them were then randomly selected to be used in the behaviorally relevant part of $A$ (that is, $A_{11}$ in equation (4); Supplementary Note 8). Furthermore, noise statistics were randomly generated and then scaled with random values to provide a wide range of relative state and observation noise values (Supplementary Note 8). Finally, we generated a separate state space model with a random number of latent states and parameters as the model for the independent behavior residual dynamics $\epsilon_k$ (Supplementary Note 8).

To generate a time-series realization with $N$ data points from a given model, we first randomly generated a $N$-data-point white Gaussian noise with the covariance given as equation (64) in Supplementary Note 8 and assigned these random numbers to $w_k$ and $v_k$. We then computed $x_k$ and $y_k$ by iterating through equation (2) with the initial value $x_{-1} = 0$. Finally, we generated a completely independent $N$-point time-series realization from the behavior residual dynamics model (see the previous paragraph) and added its generated behavior time-series (that is, $\epsilon_k$) to $C_z x_k$ to get the total $z_k$ (equation (2)).

**Evaluation metrics for learning of model parameters in numerical simulations.** A similarity transform is a revertible transformation of the basis in which states of the model are described and can be achieved by multiplying the states with any invertible matrix (Supplementary Note 3). For example, any permutation of the states is a similarity transform. Since any similarity transform on the model gives an equivalent model for the same neural activity and behavior (it just changes the latent state basis in which we describe the model; Supplementary Note 3), we cannot directly compare the parameters of the identified model with the true model and need to also consider all similarity transforms of the identified model. Thus, to evaluate the identification of model parameters (Extended Data Fig. 2), we first found a similarity transform that makes the basis of the latent states for the identified model as close as possible to the basis of the latent states for the true model. We then evaluated the difference between the identified and true values of each model parameter. Purely to find such a similarity transform, from the true model, we generated a new realization with $q = 1{,}000 n_x$ samples, which is taken to be sufficiently long for the model dynamics to be reflected in the states. We then used both the true and the identified models to extract the latent state using the steady-state Kalman filter (equation (45) in Supplementary Note 5) associated with each model, namely $\hat{x}_{k+1|k}^{(\text{true})}$ and $\hat{x}_{k+1|k}^{(\text{id})}$. We then found the similarity transform that minimized the mean-squared error between the two sets of Kalman-extracted states as

$$\hat{T} = \underset{T}{\operatorname{argmin}} \left( \sum_{k=1}^{q} \left| T\hat{x}_{k+1|k}^{(\text{id})} - \hat{x}_{k+1|k}^{(\text{true})} \right|_2^2 \right) = \hat{X}^{(\text{true})} \hat{X}^{(\text{id})\dagger} \tag{5}$$

where $\hat{X}^{(\text{true})}$ and $\hat{X}^{(\text{id})}$ are matrices whose $k$th column is composed of $\hat{x}_{k+1|k}^{(\text{true})}$ and $\hat{x}_{k+1|k}^{(\text{id})}$, respectively. We then applied the similarity transform $\hat{T}$ to the parameters of the identified model to get an equivalent model in the same basis as the true model. We emphasize again that the identified model and the model obtained from it using the above similarity transform are equivalent (Supplementary Note 3).

Given the true model and the transformed identified model, we quantified the identification error for each model parameter $\Psi$ (for example, $C_y$) using the normalized matrix norm as follows:

$$e_\Psi = \frac{\left| \Psi^{(\text{id})} - \Psi^{(\text{true})} \right|_{\text{F}}}{\left| \Psi^{(\text{true})} \right|_{\text{F}}} \tag{6}$$

where $|.|_{\text{F}}$ denotes the Frobenius norm of a matrix, which for any matrix $\Psi = \left[ \psi_{ij} \right]_{n \times m}$ is defined as

$$|\Psi|_{\text{F}} = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} \left| \psi_{ij} \right|^2} \tag{7}$$

This concludes the evaluation of the identified model parameters.

**Evaluation metrics for learning of behaviorally relevant neural dynamics.** For both the numerical simulations and for the monkey data, we used the cross-validated accuracy of decoding behavior using neural activity as a measure of how accurately the behaviorally relevant neural dynamics are learned. In the numerical simulations, we also evaluated a more direct metric based on the eigenvalues of the state transition matrix $A$; this is because for a linear state space model, these eigenvalues specify the frequency and decay rate of the response of the latent states to excitations (that is, $w_k$) and therefore determine their dynamic characteristics[67]. Specifically, we evaluated the identification accuracy for the eigenvalues associated with the behaviorally relevant latent states (that is, eigenvalues of $A_{11}$ in equation (4)). PSID identifies the model in the form of equation (4); therefore, the first block of the identified $A$ (that is, $A_{11}$ in equation (31) from Supplementary Note 1) is associated with the behaviorally relevant states in neural activity ($x_k^{(1)}$ in equation (4)). Thus, for PSID, we simply computed the eigenvalues of $A_{11}$ and evaluated their identification accuracy (Fig. 2b and Extended Data Fig. 3). NDM identification methods do not specify which states in neural activity are behaviorally relevant. So, to find these states, we first applied a similarity transform to make the NDM-identified $A$ matrix block-diagonal with each complex conjugate pair of eigenvalues in a separate block (using the bdschur command in Matlab followed by the cdf2rdf command). We then fit a linear regression from the states associated with each block to the behavior (using the training data) and sorted the blocks by their prediction accuracy of behavior $z_k$. The behaviorally relevant eigenvalues were then taken to be the top $n_1$ eigenvalues that resulted in the most accurate prediction of $z_k$ in the training data. To evaluate the decoding accuracy for a model that only keeps the top $n_1$ eigenvalues from an NDM model, we reduced the learned model—which has a high-dimensional latent state—by only keeping the segments of model parameters that are associated with the top $n_1$ eigenvalues (Extended Data Fig. 3).

When evaluating the identified eigenvalues for models with a latent state dimension that is smaller than the true $n_1$ (for example, in Fig. 2 for $n_x < 4$), we added zeros in place of the missing eigenvalues, since a model with a low latent state dimension of $a$ is equivalent to a model with a higher latent state dimension of $b$, with $b - a$ states that are always equal to zero and have eigenvalues of zero associated with them.

Finally, given the true behaviorally relevant eigenvalues for neural activity and the identified behaviorally relevant eigenvalues, we found the closest pairing of the two sets (by comparing all possible pairings), put the true and the associated closest identified eigenvalues in two vectors and computed the normalized eigenvalue detection error using equation (6).

**Estimation of the dimensionality for behaviorally relevant neural dynamics.** To estimate the dimensionality of the behaviorally relevant neural dynamics,

we sought to find the minimal number (that is, dimension) of latent states that is sufficient to best describe behavior using neural activity. To do this, for each method, we fit models with different values of state dimension $n_x$, and computed the cross-validated accuracy of decoding behavior using neural activity (equation (45) in Supplementary Note 5). We used Pearson's CC, averaged across behavior dimensions, to quantify the decoding accuracy. We then estimated the dimension of the behaviorally relevant neural dynamics as the smallest latent state dimension that reached within 1 s.e.m. of the best possible cross-validated decoding accuracy among all considered latent state dimensions (for example, Figs. 3b, 4a and 6b, Extended Data Figs. 9b and 10b and Supplementary Figs. 6b, 8b and 9b).

**Neural datasets and behavioral tasks in nonhuman primates.** We studied the neural activity and behavior in two different datasets with different behavioral tasks, recording coverage, neural signal types and subjects. All surgical and experimental procedures were performed in compliance with the National Institutes of Health Guide for Care and Use of Laboratory Animals and were approved by the New York University Institutional Animal Care and Use Committee. No statistical methods were used to predetermine sample sizes, but our sample sizes were similar to those reported in previous publications[7,9,11,21–23,36,58]. Our analyses did not require experimental intervention, and thus randomization was not needed in the assignment of animals to experiments. For a similar reason, data collection and analyses were not performed blind to the conditions of the experiments. No animals or analyzed data were excluded from this study.

*The first dataset: diverse 3D reach, grasp and returns, modeling LFP power or spiking activity.* In the first dataset, neural activity was recorded from the motor cortical areas in two adult male rhesus macaques (monkeys J and C, aged 5 and 8 years, respectively) while they were performing 3D reach, grasp and return movements[41,68] to diverse locations. A cube (2.5 $cm^3$) or cylinder (6-mm radius, 1.5 cm in length) was used as the target object, which was fixed during each recording session. The target object was manually moved by the experimenter to diverse random locations spanning a wide 3D area within the reach of the monkey. Without any timing cues, the monkey was required to reach for the object, grasp it, release it and return the hand to the resting position. The object and its movements remained visible between reaches.

For monkey J, neural activity was recorded from 137 electrodes on a microdrive (Gray Matter Research) covering parts of the M1, the PMd, the PMv and the PFC on the left hemisphere (with 28, 32, 45 and 32 channels in each area, respectively). For monkey C, activity was recorded from 128 electrodes on 4, 32-electrode microdrives (Gray Matter Research) covering the PMd and the PMv on both the left and right hemispheres (with 32 channels in each area and hemisphere). Prior work has shown that the ipsilateral (that is, here, right) hemisphere may also contain information about movements[69,70]. Using 3D-tracked retroreflective markers, the movement of various points on the torso, chest, right arm, hand and fingers were tracked within a 50-$cm^3$ workspace via the Cortex software package (Motion Analysis)[68]. These markers were used to extract—via the SIMM toolkit (MusculoGraphics)—the angular position of the 27 (monkey J) or 25 (monkey C) joints of the upper-extremity, consisting of 7 joints in the shoulder, elbow, wrist, and 20 (monkey J) or 18 (monkey C) joints in fingers (4 in each, except 2 missing finger joints in monkey C)[68,71]. All subsequent analyses were implemented in Matlab. We analyzed the neural activity during seven (monkey J) or four (monkey C) recording sessions. For most of our analyses (unless otherwise specified), to further increase the sample size, we randomly divided the electrodes into non-overlapping groups of 10 or 30 electrodes for LFP and spiking analyses, respectively, and performed modeling in each group separately. We refer to each random electrode group in each recording session as one dataset.

To model the recorded LFPs, we performed common average referencing and then, as the neural features, extracted signal log-powers (that is, in dB units) from 7 frequency bands[41,72,73] (theta: 4–8 Hz; alpha: 8–12 Hz; low beta: 12–24 Hz; high beta: 24–34 Hz; low gamma: 34–55 Hz; high gamma 1: 65–95 Hz; and high gamma 2: 130–170 Hz) within sliding 300-ms windows at a time step of 50 ms using Welch's method (using 8 subwindows with 50% overlap)[74]. To model the recorded multiunit spiking activity, we counted the spikes in 10-ms non-overlapping bins, applied a Gaussian kernel smoother (with s.d. of 50 ms)[26,38,75–77], and, unless otherwise noted, downsampled to a time step of 50 ms. The extracted features were taken as the neural activity time-series $y_k$ ($y_k \in \mathbb{R}^{70}$ in each LFP dataset and $y_k \in \mathbb{R}^{30}$ in each spike dataset). Unless otherwise noted, the behavior time-series $z_k$ was taken as the joint angles at the end of each 50-ms time step of neural activity ($z_k \in \mathbb{R}^{27}$ in monkey J and $z_k \in \mathbb{R}^{25}$ in monkey C).

*The second dataset: saccadic eye movements, modeling raw LFP activity.* In the second dataset, for two other adult male rhesus macaques (monkeys A and S, aged 8 and 6 years, respectively), neural activity was recorded from the PFC while the monkeys performed saccadic eye movements from a central fixation point toward one of eight targets on a display[36]. In both monkeys, neural activity was recorded using a 32-electrode microdrive (Gray Matter Research)[36]. The eye position was measured with an infrared optical eye tracking system (ISCAN)[36] and was taken as the behavior to be modeled. In each trial, monkeys performed delayed saccadic eye movement to one of eight targets for a liquid reward[36,78]. The visual stimuli were controlled via custom LabVIEW (National Instruments) software[36]. We analyzed the neural activity during more than 4,000 (monkey A) or 22,000 (monkey S) trials collected over 27 (monkey A) or 43 (monkey S) days. To model the recorded LFPs, we performed common average referencing, applied a causal low-pass anti-aliasing filter with a cut-off of 8 Hz (order 4 IIR Butterworth filter) and then downsampled the LFP signals to a 20-Hz sampling rate (that is, 50-ms sampling time step). We took the downsampled raw LFP activity as the neural activity time-series $y_k$ ($y_k \in \mathbb{R}^{32}$). We took the eye position, similarly downsampled to a 20-Hz sampling rate, as the behavior time-series $z_k$ ($z_k \in \mathbb{R}^2$).

**Cross-validated model evaluation in real neural datasets.** For each method, we performed the model identification and decoding within a fivefold cross-validation, and as the performance metric for predicting behavior, we computed the cross-validated CC between the true and predicted behavior (for example, joint angles). For all methods, in each cross-validation fold, we first $z$-scored each dimension of neural activity and behavior based on the training data to ensure that learning methods did not discount any behavior or neural dimensions due to a potentially smaller natural variance. Nevertheless, we found that results were almost identical even without this $z$-scoring step (not shown), which suggests that in our datasets, variabilities in variance across data dimensions had no major impact. In fitting the models with PSID, for each latent dimension $n_x$, unless specified otherwise, $n_1$ was selected using a fourfold inner cross-validation within the training data. For PSID and standard SID[35,62], a horizon parameter of $i = 5$ was used in all analyses, except for per channel analyses (Fig. 4), spiking activity analyses (Fig. 6) and raw LFP activity analyses (Extended Data Fig. 10), for which a horizon of $i = 20$ was used due to the smaller neural feature dimension. For some control analyses with NDM, we used the expectation maximization algorithm[79,80]. Once models were learned in the training data, each method applied a Kalman filter—corresponding to the learned model—on neural activity in the test data to extract the latent states and to predict the behavior or self-predict the neural activity from these extracted states.

**Implementation details for alternative nonlinear methods.** The new functionality offered by PSID is the ability to dissociate and model behaviorally relevant neural dynamics by considering both neural activity and behavior when learning the dynamic model in contrast to NDM (whether linear[6,21,22,29,30,33], generalized linear[30,32,37] or nonlinear[23,81]), which only considers neural activity in learning the dynamic model. To show that the advantage of PSID is due to its learning approach and thus holds regardless of the NDM model structure, we also performed a comparison with example nonlinear NDM methods. As the first example for nonlinear NDM, we learned a model using linear NDM and then used nonlinear SVR[82] to regress the latent state to behavior. We used the LIBSVM[82] library to learn the SVR with a radial basis function kernel and an epsilon-SVR loss with default parameters[82]. As the second example of a nonlinear NDM, we used a RNN-based method termed LFADS[23], which was recently successfully applied to neural spiking activity[23]. LFADS fits a RNN autoencoder that can take fixed-length segments of data as input and encode the dynamics in each segment into an initial condition. Given this initial condition, a RNN termed the generator then generates a factor time-series that can linearly reconstruct a smoothed copy of the input data segment. Thus, we can apply LFADS on our neural data by cutting the neural feature time-series in fixed-length non-overlapping 1-s segments[23], applying LFADS smoothing to each segment to denoise the neural activity during that segment and getting a corresponding factor time-series. To decode behavior, a linear regression can be fitted from the factor time-series to the behavior[23]. We ran the LFADS model fitting and subsequently learned the linear regression from factors to the behavior using the training data and tested the decoding accuracy in the test data. For the LFADS model, unless otherwise noted, all hyperparameters were selected to be as in row 2 of supplementary table 1 in ref. [23], given that the number of trials in that dataset from ref. [23] was closest to our data. In particular, we always set the dimension of the initial condition to 64 as in row 2 of supplementary table 1 in ref. [23]. In the LFADS formulation, the generator RNN models the dynamics of neural data[23]. The dimension of the state vector of the generator RNN[23]—also referred to as the number of units in the generator[23]—specifies the memory of the generator RNN or how many numbers are used to represent the state of the generator at a given time step and to generate the dynamics in the next time step[83]. Since the same concept is represented by the state dimension in a state space model (equation (1)), to be comparable with other methods, we refer to the generator state dimension in LFADS as the state dimension or dimension of dynamics. Nevertheless, for completeness, we tried two configurations for the generator state dimension of LFADS. First, to provide directly comparable results with other methods, as we swept over a different number of factors, we always set the generator state dimension (that is, the dimension of dynamics) to be equal to the number of factors. Hence, in this configuration, the number of factors in LFADS also indicates the dimension of its dynamics and is therefore comparable with the state dimension for other methods (Fig. 6). Second, in a control analysis, we always set the state dimension of the LFADS generator to 64 as in row 2 of supplementary table 1 in ref. [23] and then only swept the number of factors, but the results were similar (Extended Data Fig. 9). When applying LFADS to the same Gaussian smoothed spiking activity features that were used by PSID (Fig. 6), we used a Gaussian observation model for LFADS. In another control

analysis, we additionally applied LFADS to non-smoothed spike counts in 10-ms non-overlapping bins (Extended Data Fig. 9), in which case we used a Poisson observation model for LFADS. Again, all conclusions were similar. It is worth noting that given its analytical form, PSID was computationally more efficient both in fitting the models in the training data (per dataset: 3.2 min for PSID versus 120 h for LFADS, which is 2,250 times more efficient) and in estimating the latent representation from neural activity in the test data (per dataset: 0.53 s for PSID versus 35 min for LFADS, which is 4,000 times more efficient).

**Statistics.** We used the Wilcoxon signed-rank or rank-sum test for all paired and nonpaired statistical tests, respectively. These tests are nonparametric and do not assume a specific distribution for the data. To correct for multiple comparisons when comparing the performance of methods for different joints or channels, we corrected the *P* values within the test data using the false discovery rate control[84]. Statistical test details and exact *P* values for relevant figures are provided in Supplementary Table 1.

**Reporting Summary.** Further information on research design is available in the Nature Research Reporting Summary linked to this article.

## Data availability

The data used to support the results are available upon reasonable request from the corresponding author.

## Code availability

The code for the PSID algorithm is available online at https://github.com/ShanechiLab/PSID.

## References


51. Thura, D. & Cisek, P. Deliberation and commitment in the premotor and primary motor cortex during dynamic decision making. *Neuron* **81**, 1401–1416 (2014).
52. Haroush, K. & Williams, Z. M. Neuronal prediction of opponent's behavior during cooperative social interchange in primates. *Cell* **160**, 1233–1245 (2015).
53. Herzfeld, D. J., Kojima, Y., Soetedjo, R. & Shadmehr, R. Encoding of action by the Purkinje cells of the cerebellum. *Nature* **526**, 439–442 (2015).
54. Ramkumar, P., Dekleva, B., Cooler, S., Miller, L. & Kording, K. Premotor and motor cortices encode reward. *PLoS ONE* **11**, e0160851 (2016).
55. Whitmire, C. J., Waiblinger, C., Schwarz, C. & Stanley, G. B. Information coding through adaptive gating of synchronized thalamic bursting. *Cell Rep.* **14**, 795–807 (2016).
56. Christophel, T. B., Klink, P. C., Spitzer, B., Roelfsema, P. R. & Haynes, J.-D. The distributed nature of working memory. *Trends Cogn. Sci.* **21**, 111–124 (2017).
57. Takahashi, K. et al. Encoding of both reaching and grasping kinematics in dorsal and ventral premotor cortices. *J. Neurosci.* **37**, 1733–1746 (2017).
58. Menz, V. K., Schaffelhofer, S. & Scherberger, H. Representation of continuous hand and arm movements in macaque areas M1, F5, and AIP: a comparative decoding study. *J. Neural Eng.* **12**, 056016 (2015).
59. Wu, W., Gao, Y., Bienenstock, E., Donoghue, J. P. & Black, M. J. Bayesian population decoding of motor cortical activity using a Kalman filter. *Neural Comput.* **18**, 80–118 (2006).
60. Bansal, A. K., Truccolo, W., Vargas-Irwin, C. E. & Donoghue, J. P. Decoding 3D reach and grasp from hybrid signals in motor and premotor cortices: spikes, multiunit activity, and local field potentials. *J. Neurophysiol.* **107**, 1337–1355 (2011).
61. Obinata, G. & Anderson, B. D. O. *Model Reduction for Control System Design* (Springer Science & Business Media, 2012).
62. Katayama, T. *Subspace Methods for System Identification* (Springer Science & Business Media, 2006).
63. Shenoy, K. V. & Carmena, J. M. Combining decoder design and neural adaptation in brain–machine interfaces. *Neuron* **84**, 665–680 (2014).
64. Yang, Y. & Shanechi, M. M. An adaptive and generalizable closed-loop system for control of medically induced coma and other states of anesthesia. *J. Neural Eng.* **13**, 066019 (2016).
65. Yang, Y., Chang, E. F. & Shanechi, M. M. Dynamic tracking of non-stationarity in human ECoG activity. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* 1660–1663 (2017).
66. Ahmadipour, P., Yang, Y. & Shanechi, M. M. Investigating the effect of forgetting factor on tracking non-stationary neural dynamics. In *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)* 291–294 (2019).
67. Fu, Z.-F. & He, J. *Modal Analysis* (Elsevier, 2001).
68. Wong, Y. T., Putrino, D., Weiss, A. & Pesaran, B. Utilizing movement synergies to improve decoding performance for a brain machine interface. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* 289–292 (2013).
69. Cisek, P., Crammond, D. J. & Kalaska, J. F. Neural activity in primary motor and dorsal premotor cortex in reaching tasks with the contralateral versus ipsilateral arm. *J. Neurophysiol.* **89**, 922–942 (2003).
70. Ames, K. C. & Churchland, M. M. Motor cortex signals for each arm are mixed across hemispheres and neurons yet partitioned within the population response. *eLife* **8**, e46159 (2019).
71. Putrino, D., Wong, Y. T., Weiss, A. & Pesaran, B. A training platform for many-dimensional prosthetic devices using a virtual reality environment. *J. Neurosci. Methods* **244**, 68–77 (2015).
72. Flint, R. D., Ethier, C., Oby, E. R., Miller, L. E. & Slutzky, M. W. Local field potentials allow accurate decoding of muscle activity. *J. Neurophysiol.* **108**, 18–24 (2012).
73. Bundy, D. T., Pahwa, M., Szrama, N. & Leuthardt, E. C. Decoding three-dimensional reaching movements using electrocorticographic signals in humans. *J. Neural Eng.* **13**, 026021 (2016).
74. Oppenheim, A. V. & Schafer, R. W. *Discrete-Time Signal Processing* (Pearson Higher Education, 2011).
75. Williams, A. H. et al. Unsupervised discovery of demixed, low-dimensional neural dynamics across multiple timescales through tensor component analysis. *Neuron* **98**, 1099–1115.e8 (2018).
76. Trautmann, E. M. et al. Accurate estimation of neural population dynamics without spike sorting. *Neuron* **103**, 292–308.e4 (2019).
77. Gallego, J. A., Perich, M. G., Chowdhury, R. H., Solla, S. A. & Miller, L. E. Long-term stability of cortical population dynamics underlying consistent behavior. *Nat. Neurosci.* **23**, 260–270 (2020).
78. Sadras, N., Pesaran, B. & Shanechi, M. M. A point-process matched filter for event detection and decoding from population spike trains. *J. Neural Eng.* **16**, 066016 (2019).
79. Ghahramani, Z. & Hinton, G. E. *Parameter Estimation for Linear Dynamical Systems*. Technical Report CRG-TR-92-2, 1–6 (University of Toronto, 1996); https://www.cs.toronto.edu/~hinton/absps/tr96-2.html
80. Bishop, C. M. *Pattern Recognition and Machine Learning* (Springer, 2011).
81. Archer, E. W., Koster, U., Pillow, J. W. & Macke, J. H. Low-dimensional models of neural population activity in sensory cortical circuits. In *Advances in Neural Information Processing Systems* 27 (eds Ghahramani, Z. et al.) 343–351 (Curran Associates, 2014).
82. Chang, C.-C. & Lin, C.-J. LIBSVM: a library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2**, 1–27 (2011).
83. Medsker, L. & Jain, L. C. *Recurrent Neural Networks: Design and Applications* (CRC Press, 1999).
84. Benjamini, Y. & Hochberg, Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Stat. Soc. Ser. B Methodol.* **57**, 289–300 (1995).


## Acknowledgements

This work was supported in part by the following organizations and grants: the Army Research Office (ARO) under contract W911NF-16-1-0368 as part of the collaboration between the US DOD, the UK MOD and the UK Engineering and Physical Research Council (EPSRC) under the Multidisciplinary University Research Initiative (MURI) (to M.M.S.); the Office of Naval Research (ONR) Young Investigator Program (YIP) under contract N00014-19-1-2128 (to M.M.S.); the National Science Foundation (NSF) CAREER Award CCF-1453868 (to M.M.S.); ARO contract W911NF1810434 under the Bilateral Academic Research Initiative (BARI) (to M.M.S.); US National Institutes of Health (NIH) BRAIN grant R01-NS104923 (to B.P. and M.M.S.); and a University of Southern California Annenberg Fellowship (to O.G.S).

## Author contributions

O.G.S. and M.M.S. conceived the study and developed the new PSID algorithm. O.G.S. performed all the analyses. H.A. performed the muscle activation inference used in Supplementary Fig. 14. Y.T.W. and B.P. provided all the nonhuman primate data. O.G.S. and M.M.S. wrote the manuscript with input from B.P.

## Competing interests

The authors declare no competing interests.

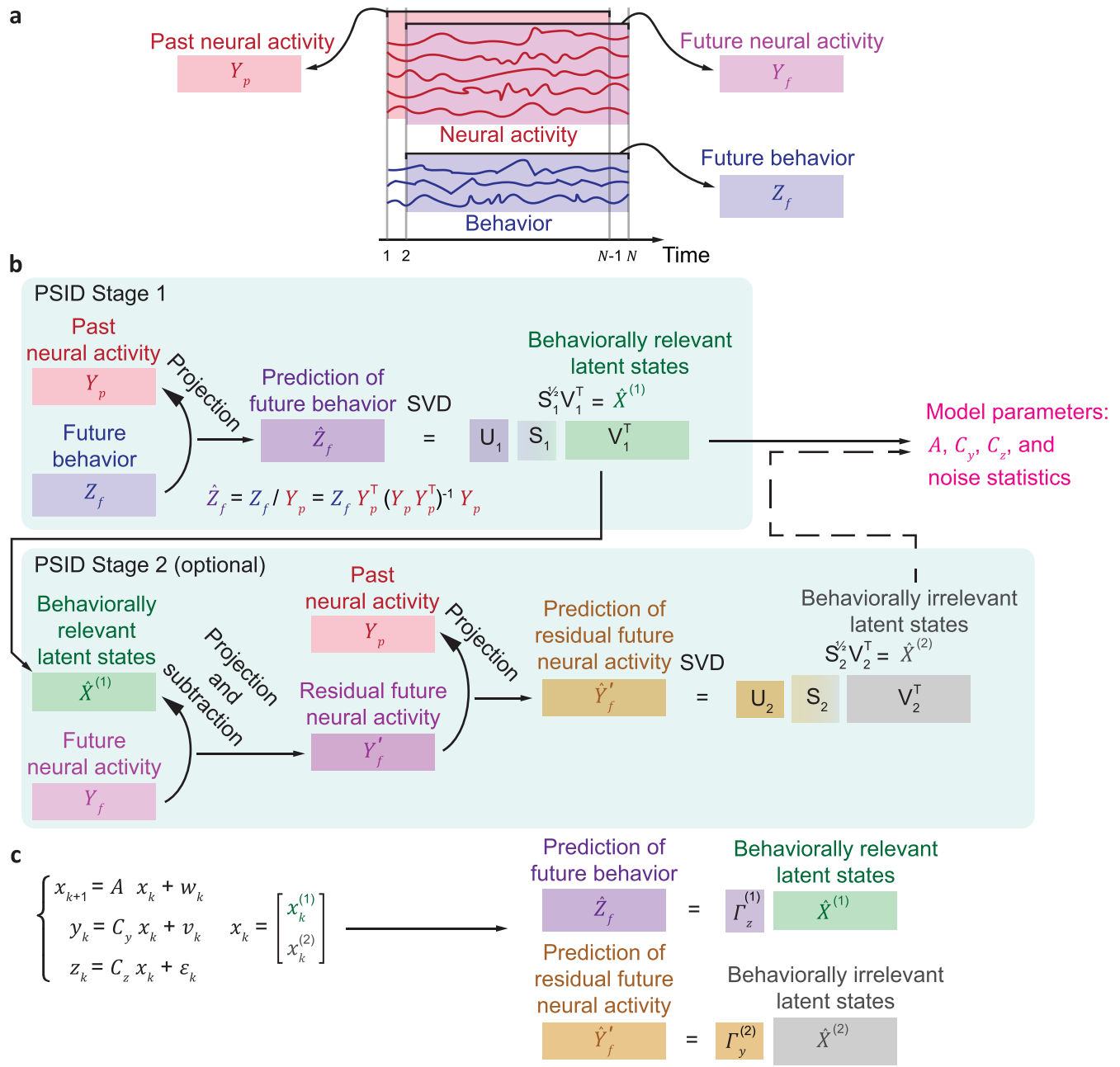
## Additional information

**Extended data** is available for this paper at https://doi.org/10.1038/s41593-020-00733-0.

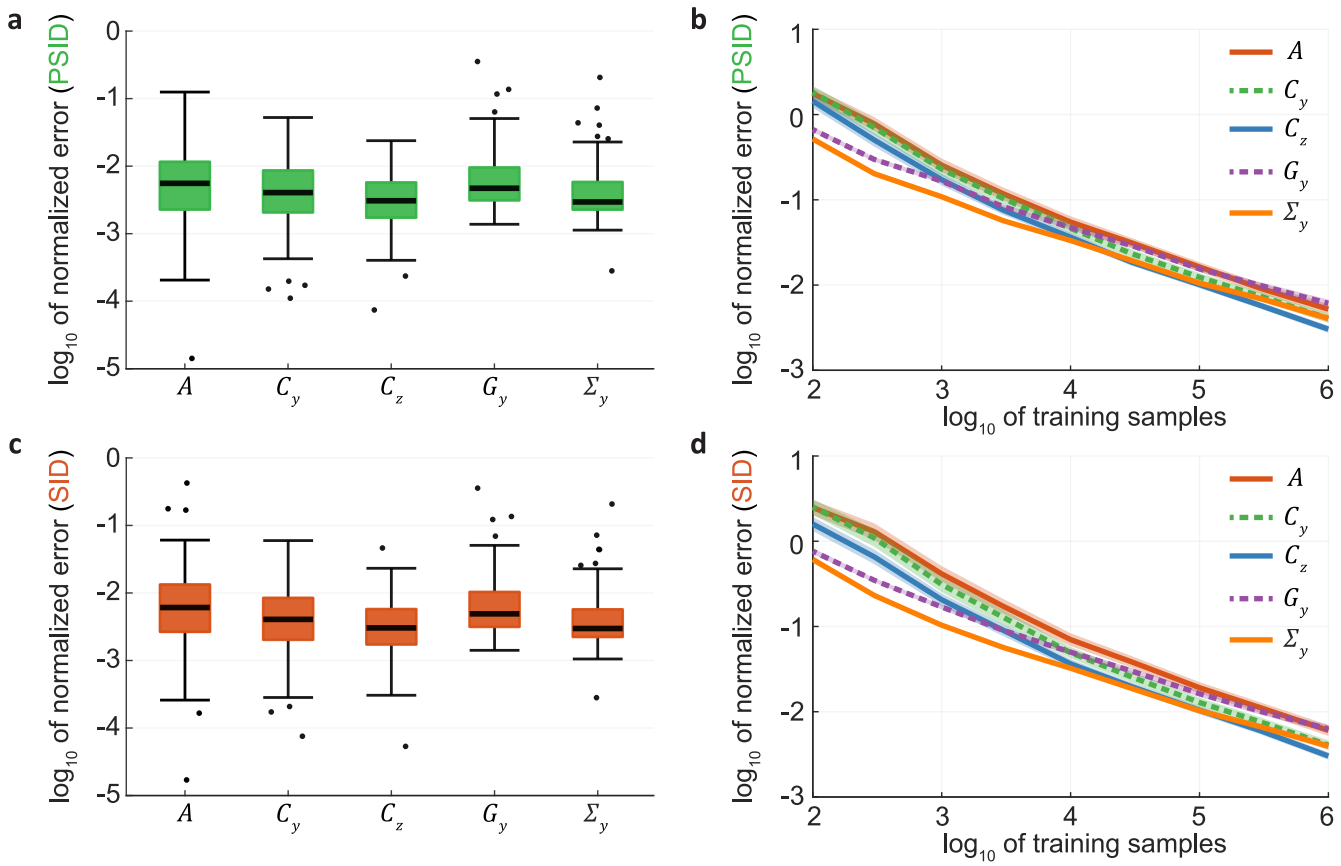
**Supplementary information** is available for this paper at https://doi.org/10.1038/s41593-020-00733-0.

**Correspondence and requests for materials** should be addressed to M.M.S.

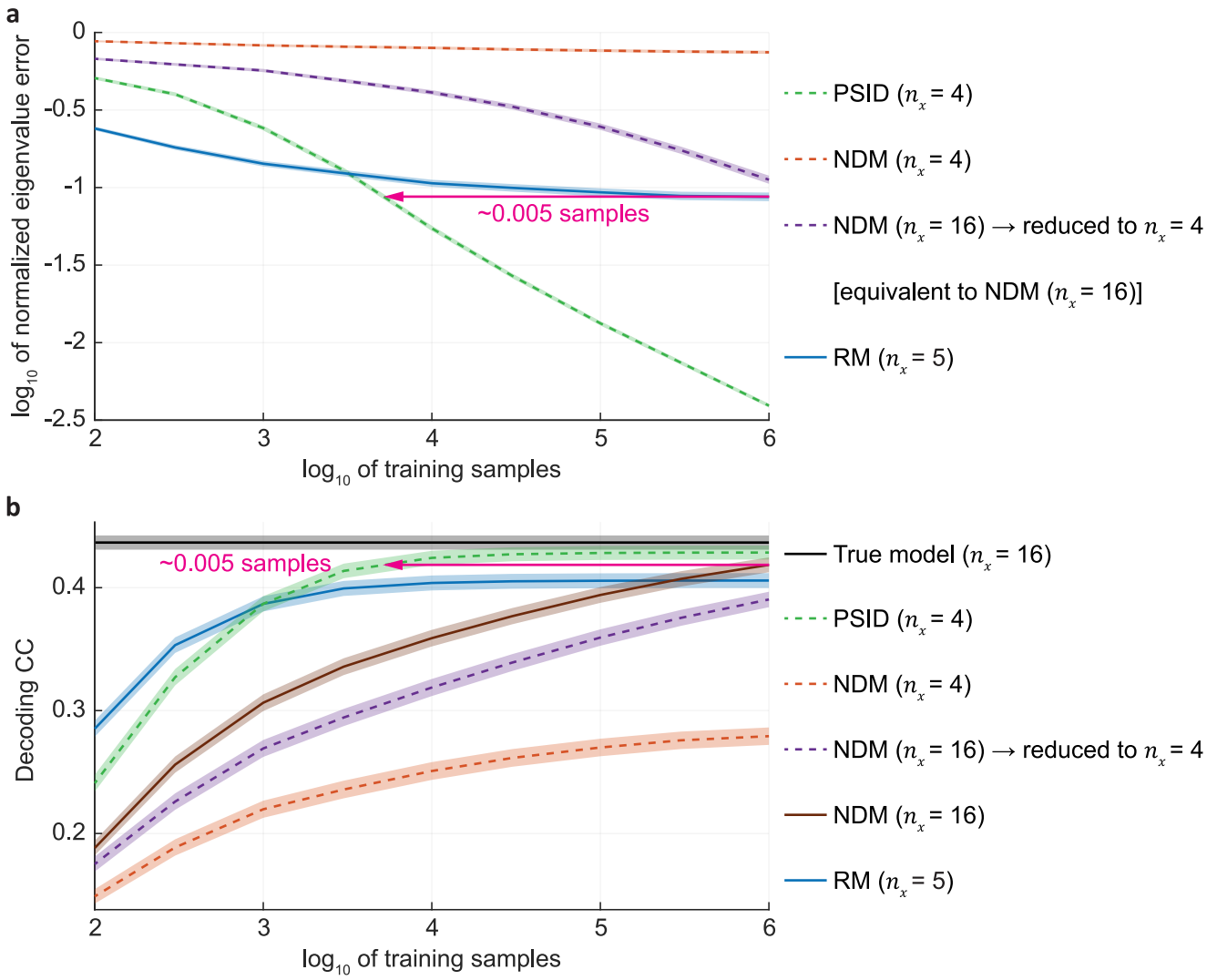
**Peer review information** *Nature Neuroscience* thanks Carsen Stringer and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.



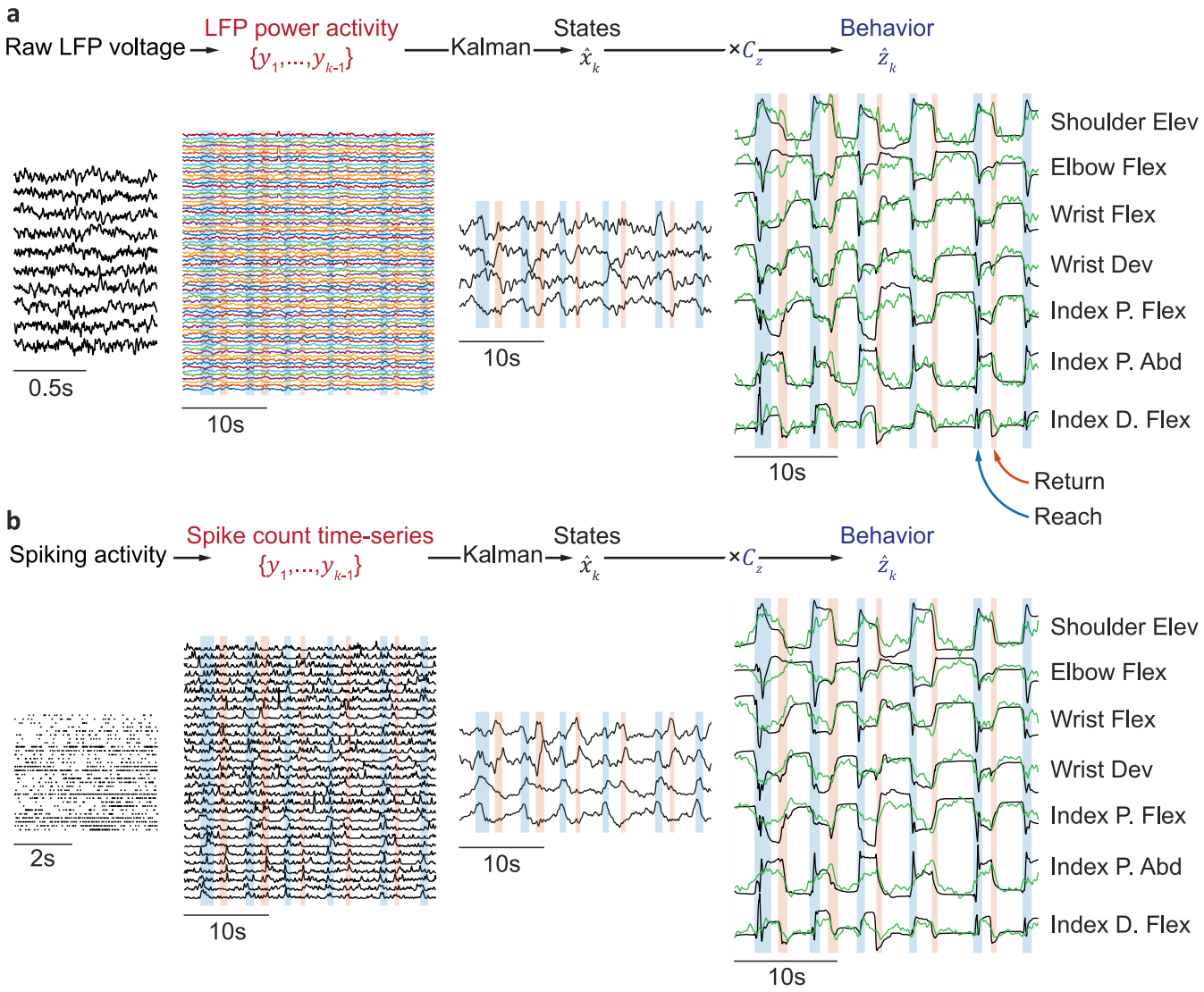

**Extended Data Fig. 1 | Visualization of the PSID algorithm.** (**a**) The extraction of future and past neural activity and future behavior from data is shown (see Supplementary Note 1 for the general definition). Matrices are depicted as colored rectangles. Past and future neural activity matrices $Y_p$ and $Y_f$ are of the same size, with columns of $Y_f$ containing neural data for one step into the future relative to the corresponding columns of $Y_p$. Future behavior matrix $Z_f$ includes the time-series of behavior at the same time steps as $Y_f$. (**b**) PSID learning algorithm. In stage one of PSID, performing SVD on the projection of future behavior $Z_f$ onto past neural activity $Y_p$ gives the behaviorally relevant latent states $\hat{X}^{(1)}$. These states can be used on their own to learn the parameters for a model that only includes behaviorally relevant latent states. Optionally, stage two of PSID can be used to also extract behaviorally irrelevant latent states $\hat{X}^{(2)}$. In stage two, residual future neural activity $Y_f'$ is obtained by subtracting from $Y_f$ its projection onto $\hat{X}^{(1)}$. Performing SVD on the projection of residual future neural activity $Y_f'$ onto past neural activity $Y_p$ gives the behaviorally irrelevant latent states $\hat{X}^{(2)}$. These states can then be used together with the behaviorally relevant latent states $\hat{X}^{(1)}$ to learn the parameters for a model that includes both sets of states. Once model parameters (Equation. 1) are learned using only the neural and behavior training data, the extraction of latent states and the decoding of behavior in the test data are done purely from neural activity and using a Kalman filter and linear regression as shown in Fig. 1c (the Kalman filter and linear regression are specified by the learned model parameters). (**c**) A brief sketch of the main derivation step to obtain the PSID algorithm in (b). In the derivation of PSID (Supplementary Note 6), we show that for the model in Equation. 1, the prediction of future behavior $Z_f$ using past neural activity $Y_p$ (that is $\hat{Z}_f$) has the same row space as the behaviorally relevant latent states $\hat{X}^{(1)}$. Similarly, we show that the prediction of the residual future neural activity $Y_f'$ using past neural activity $Y_p$ (that is $\hat{Y}_f'$) has the same row space as the behaviorally irrelevant latent states $\hat{X}^{(2)}$ (Supplementary Note 6). Thus, in (b), we can empirically extract the latent states $\hat{X}^{(1)}$ and $\hat{X}^{(2)}$ from training data by first computing the predictions $\hat{Z}_f$ and $\hat{Y}_f'$ as shown in (b) via projections, and then finding their row space using SVD.



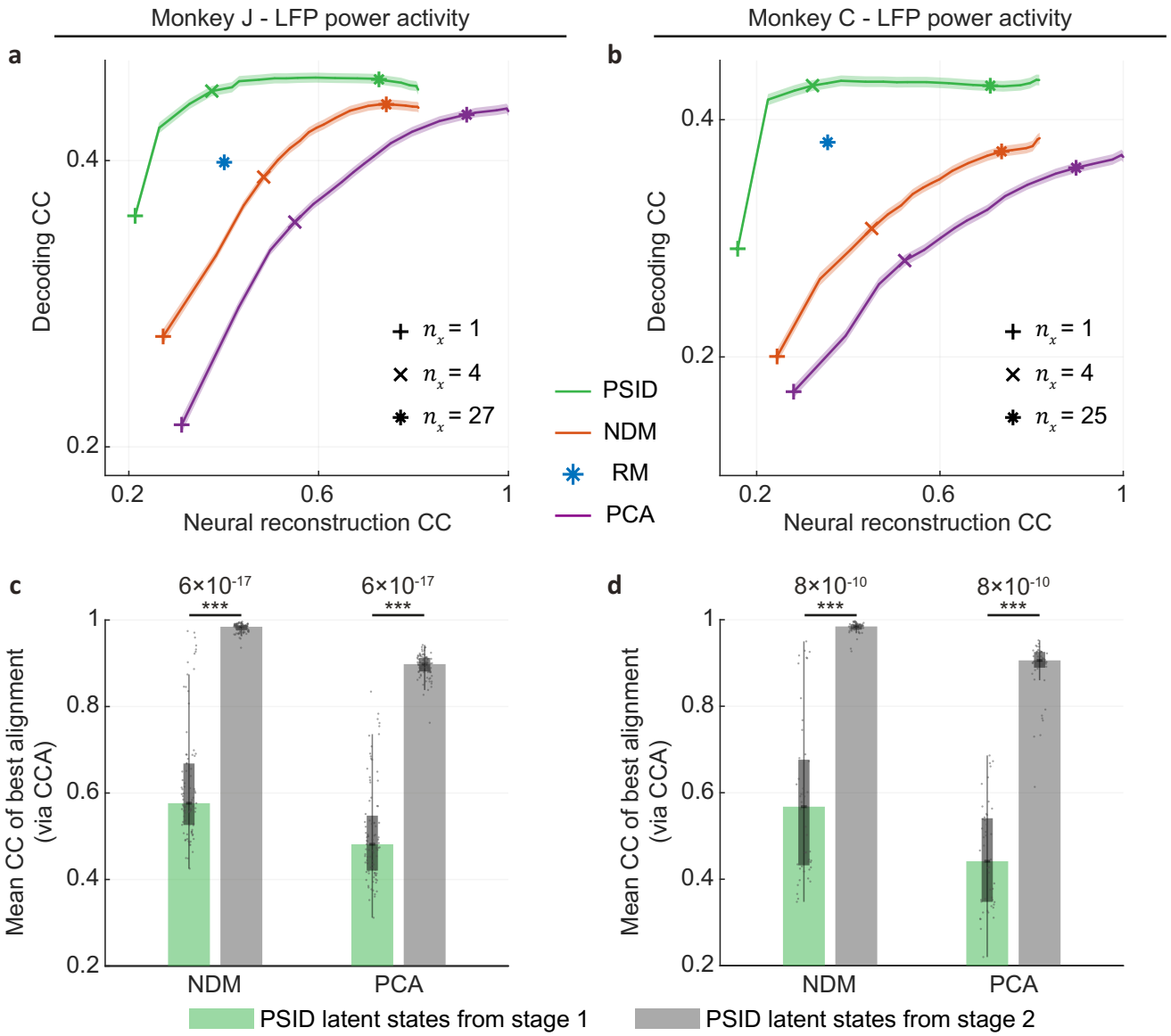

**Extended Data Fig. 2 | PSID correctly learns model parameters at a rate of convergence similar to that of SID while also being able to prioritize behaviorally relevant dynamics.** (**a**) Normalized error for identification of each model parameter using PSID (with $10^6$ training samples) across 100 random simulated models. Each model had randomly selected state, neural activity, and behavior dimensions as well as randomly generated parameters (Methods). The parameters $A$, $C_y$, $C_z$ from Equation. 1 together with the covariance of neural activity $\Sigma_y \triangleq \boldsymbol{E}\{y_k y_k^T\}$ and the cross-covariance of neural activity with the latent state $G_y \triangleq \boldsymbol{E}\{x_{k+1} y_k^T\}$ fully characterize the model (Methods). Here, the same model structure parameters $n_x$ (total latent state dimension) and $n_1$ (dimension of the latent states extracted during the first stage of PSID) as the true model were used when applying PSID to data for each model (see Supplementary Fig. 3 on how these model structure parameters are also accurately identified from data). The horizontal dark line on the box shows the median, box edges show the 25th and 75th percentiles, whiskers represent the minimum and maximum values (other than outliers) and the dots show the outlier values. Outliers are defined as in Fig. 3b. Using $10^6$ samples, all parameters are identified with a median error smaller than 1%. (**b**) Normalized error for all parameters as a function of the number of training samples for PSID. The normalized error consistently decreases as more samples are used for identification. Solid line shows the average $\log_{10}$ of the normalized error and the shaded area shows the s.e.m. (**c**)-(**d**) Same as (a)-(b), shown for the standard SID algorithm. The rate of convergence for both PSID and SID, and for all parameters is around 10 times smaller error for 100 times more training samples (that is slope of −0.5 on (b), (d)). $n = 100$ random models in all panels.



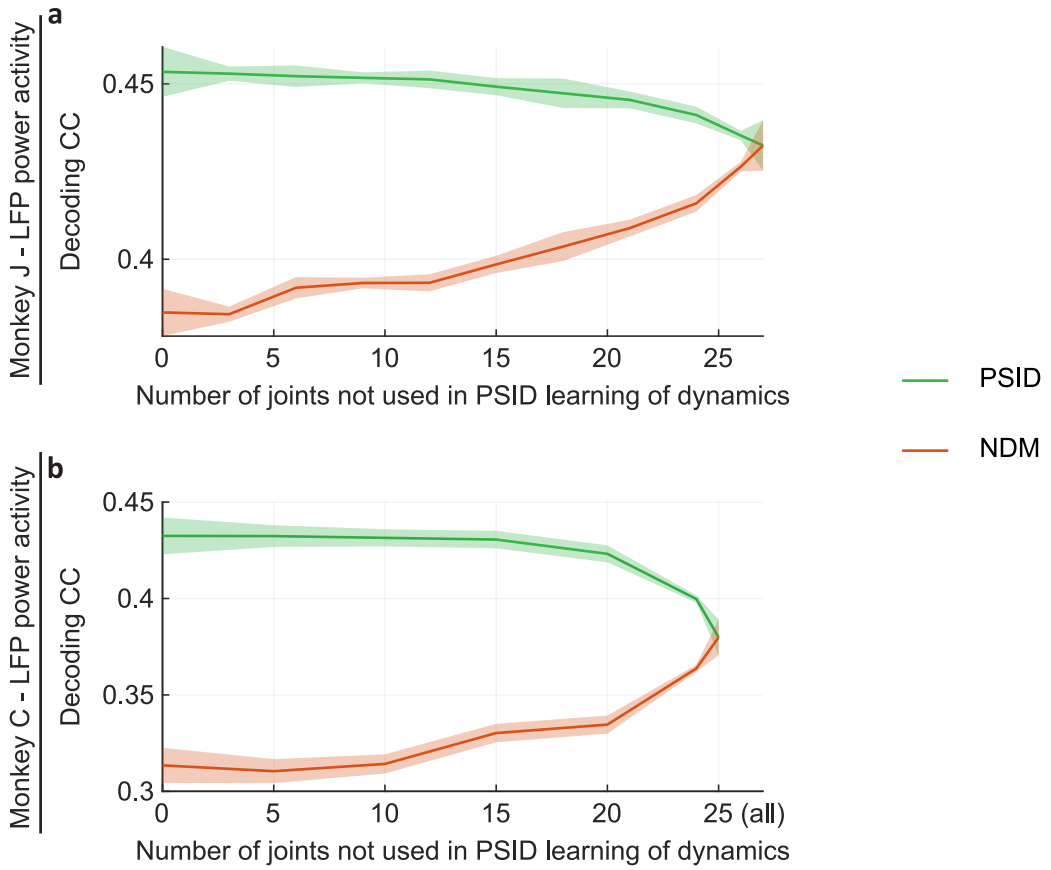

**Extended Data Fig. 3 | PSID requires orders of magnitude fewer training samples to achieve the same performance as NDM that uses a larger latent state dimension, and NDM with the same latent state dimension as PSID or RM do not achieve a comparable performance to PSID even with orders of magnitude more samples.** (**a**) Normalized eigenvalue error is shown for 1000 random simulated models with 16-dimensional latent states out of which 4 are behaviorally relevant, when using RM, PSID, or NDM with similar or larger latent state dimension than PSID. Solid lines show the average and shaded areas show the s.e.m. ($n = 1000$ random models). For NDM, to learn the behaviorally relevant dynamics using a model with a high-dimensional latent state ($n_x = 16$), we first identify this model, then sort the dimensions of the extracted latent state in order of their decoding accuracy, and then reduce the model to keep the 4 most behavior predictive latent state dimensions (Methods). These operations provide the estimate of the 4 behaviorally relevant eigenvalues (Methods). For RM, the state dimension is the behavior dimension (here $n_z = 5$). (**b**) Cross-validated behavior decoding CC for the models in (a). Figure convention and number of samples are the same as in (a). Note that unlike in (a), here we provide decoding results using the NDM with a 16-dimensional latent state both with and without any model reduction, as the two versions result in different decoding while they don't differ in their most behavior predictive dimensions and thus have the same eigenvalue error in (a). Optimal decoding using the true model is shown as black. For NDM with a 4-dimensional latent state (that is in the dimension reduction regime) and RM, eigenvalue identification in (a) and decoding accuracies in (b) almost plateaued at some final value below that of the true model, indicating that the asymptotic performance of having unlimited training samples has almost been reached. In both (a) and (b), even for an NDM with a latent state dimension as large as the true model (that is not performing any dimension reduction and using $n_x = 16$), (i) NDM was inferior in performance compared with PSID with a latent state dimension of only 4 when using the same number of training samples, and (ii) NDM required orders of magnitude more training samples to reach the performance of PSID with the smaller latent state dimension as shown by the magenta arrow. Parameters are randomized as in Methods except for the state noise ($w_t$), which is about 30 times smaller (that is $-2.5 \leq \alpha_1 \leq -0.5$), and the behavior signal-to-noise ratio, which is 2 times smaller (that is $-0.3 \leq \alpha_3 \leq +1.7$), both adjusted to make the decoding performances more similar to the results in real neural data (Fig. 3).



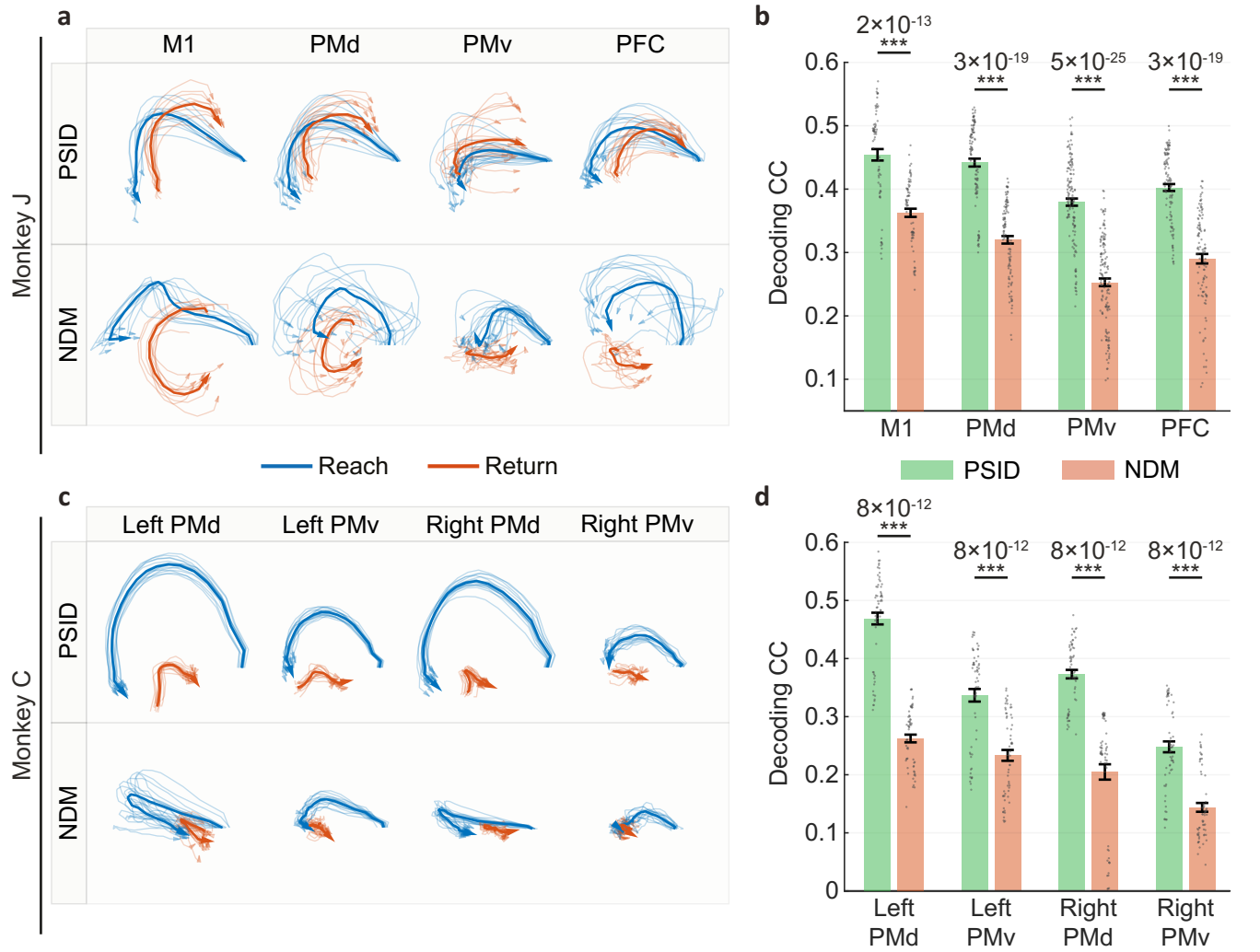

**Extended Data Fig. 4 | PSID can be used to model neural activity for different neural signal types including LFP power activity or population spiking activity.** Modeling neural activity using PSID is demonstrated with example signals, extracted latent states, and decoded behavior for (**a**) LFP power activity (that is signal power in different frequency bands, which are shown with different colors, Methods) and (**b**) Population spiking activity (Methods). In both cases, regardless of neural signal type, after extracting the neural feature time-series, decoding consists of two steps: 1) applying Kalman filter to extract the latent states given the neural feature time-series, 2) computing a linear combination of the states to get the decoding of behavior. By learning the dynamic model parameters, PSID specifies the Kalman filter parameters as well as the linear combination. Joint name abbreviations are as in Supplementary Fig. 12.



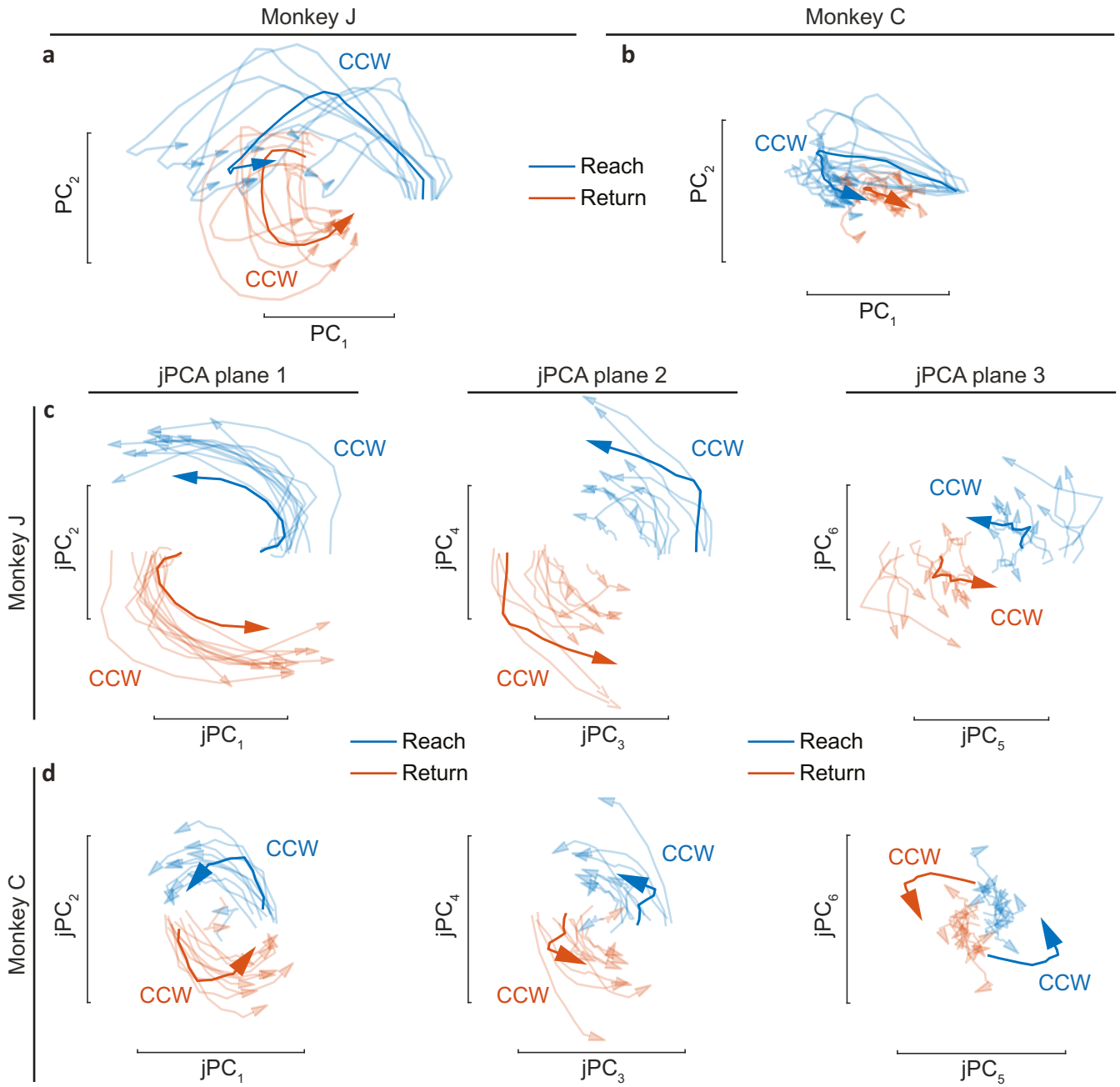

**Extended Data Fig. 5 | As the dimension of the latent state extracted by PSID increases, it first covers the subspace of neural dynamics that are behaviorally relevant and then covers the subspace of residual neural dynamics.** (**a**) For different state dimensions (or different number of principal components (PCs) in the case of PCA), the cross-validated behavior decoding CC is shown versus the cross-validated accuracy of reconstructing neural activity using the same states/PCs quantified by CC. For PSID, NDM, and RM, reconstruction of neural activity is done using a Kalman filter for one time step into the future (that is one-step-ahead self-prediction, Methods). For PCA, reconstruction is done for the same time step by multiplying the extracted PCs by the transpose (that is inverse) of the PCA decomposition matrix. Solid lines show the average decoding CC and shaded areas show the s.e.m. ($n = 91$ datasets). Multiple points on the curves associated with equal number of states/PCs are marked with the same symbol (plus/cross/asterisks). (**b**) Same as (a) for monkey C ($n = 48$ datasets). (**c**) Using canonical correlation analysis (CCA), average CC for the best linear alignment between the latent states extracted in the first and second stages of PSID with the latent states/PCs extracted using NDM/PCA is shown (see also Extended Data Fig. 1). The state/PC dimension for NDM/PCA was the same as the state dimension in the first stage of PSID. Bars, boxes and asterisks are defined in as in Fig. 3b. (**d**) Same as (c) for monkey C. Statistical tests in panels c,d are one-sided signed-rank with $n$ (number of datasets) as in panels a,b, respectively, with the $P$ values noted above asterisks in the plot. As expected, compared with the second stage of PSID, the latent states extracted in the first stage of PSID are significantly less aligned with latent states from NDM and PCA (panels c,d). This is consistent with the first few state dimensions extracted by the first stage of PSID being significantly more aligned to behavior compared with the states extracted by NDM or PCA in panels a,b; it is also consistent with PSID reaching similar neural self-prediction as NDM when also using those states extracted in the second stage and thus higher overall latent state dimension (panels a,b). The first stage of PSID learns behaviorally relevant neural dynamics resulting in better PSID decoding using lower-dimensional latent states while its second stage learns the residual dynamics in neural activity (panels a,b). That is why latent states from the first stage are significantly less aligned with states from PCA and NDM, which simply aim to fit the dynamics in neural activity agnostic to their relevance to behavior.



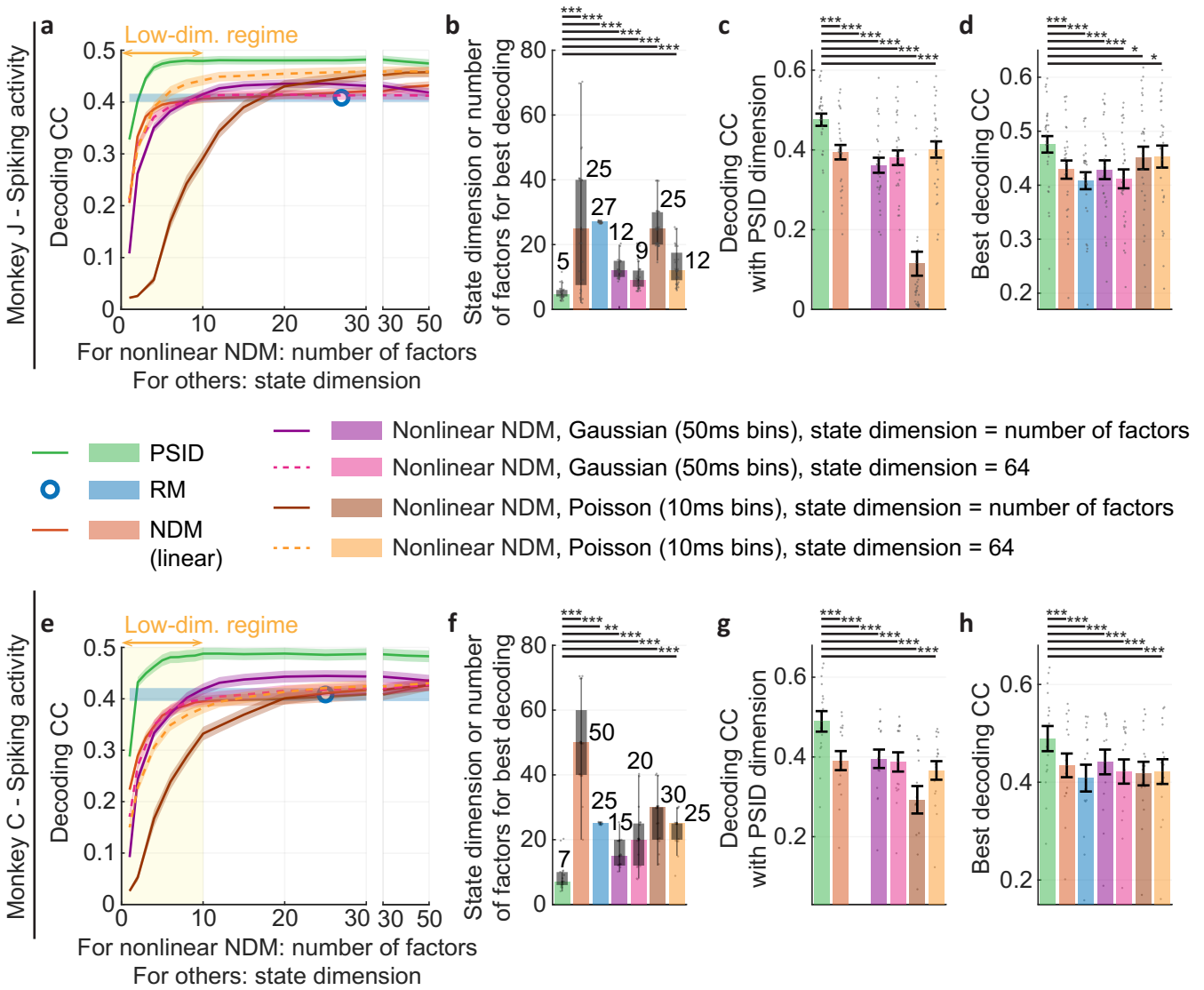

**Extended Data Fig. 6 | Dynamic model learned by PSID using a subset of joints in the training data was more predictive of the remaining joints in the test data compared with the dynamic model learned by NDM.** We selected a subset of joints and excluded them from the PSID modeling. After learning the dynamic model and extracting the latent states in the training data, we fitted a linear regression from these latent states to predict the remaining joints that were unseen by PSID (that is the regression solution constituting the parameter $C_z$ in the model). Similarly, NDM learned its dynamic model and extracted the latent states in the training data, and then fitted a linear regression from these latent states to predict the joints. We then evaluated the final learned models in the test data. We repeated this procedure for multiple random joint subsets while ensuring that overall, all joints are a member of the unseen subsets equal number of times. (**a**) The peak cross-validated decoding accuracy (CC) is shown for PSID as a function of the number of joints that were unseen when learning the dynamic model. In each dataset, the same latent state dimension as PSID is used for NDM. In NDM, joints are never used in learning the dynamic model, equivalent to having all joints in the unseen subset. Indeed, PSID reduces to NDM in the extreme case when no joint is provided to PSID in learning the dynamic model as evident from the green and red curves converging at the end (in this case only stage 2 of PSID is performed, Methods). Solid lines show the average decoding CC and shaded areas show the s.e.m. ($n \geq 91$ joint subset datasets). (**b**) Same as (a), for monkey C ($n \geq 48$ joint subset datasets). For both monkeys and in all cases (other than PSID not seeing any joint for which it reduces to NDM), PSID decoding was significantly better than NDM decoding ($P < 10^{-6}$; one-sided signed-rank; $n \geq 91$ and $n \geq 48$ joint subset datasets in monkeys J and C, respectively). To investigate why training PSID with a subset of joints helps in decoding of a different unseen subset of joints in the test data, we computed the correlation coefficient between each pair of joint angles within our datasets and found an absolute correlation coefficient value of 0.31 ± 0.0097 (mean ± s.e.m., $n = 351$ joint pairs) and 0.32 ± 0.011 ($n = 300$ joint pairs), for monkeys J and C respectively. This result may suggest that since all joints are engaged in the same task, there are correlations between them that allow PSID to improve decoding even for joints that it does not observe during learning the dynamic model in training data.



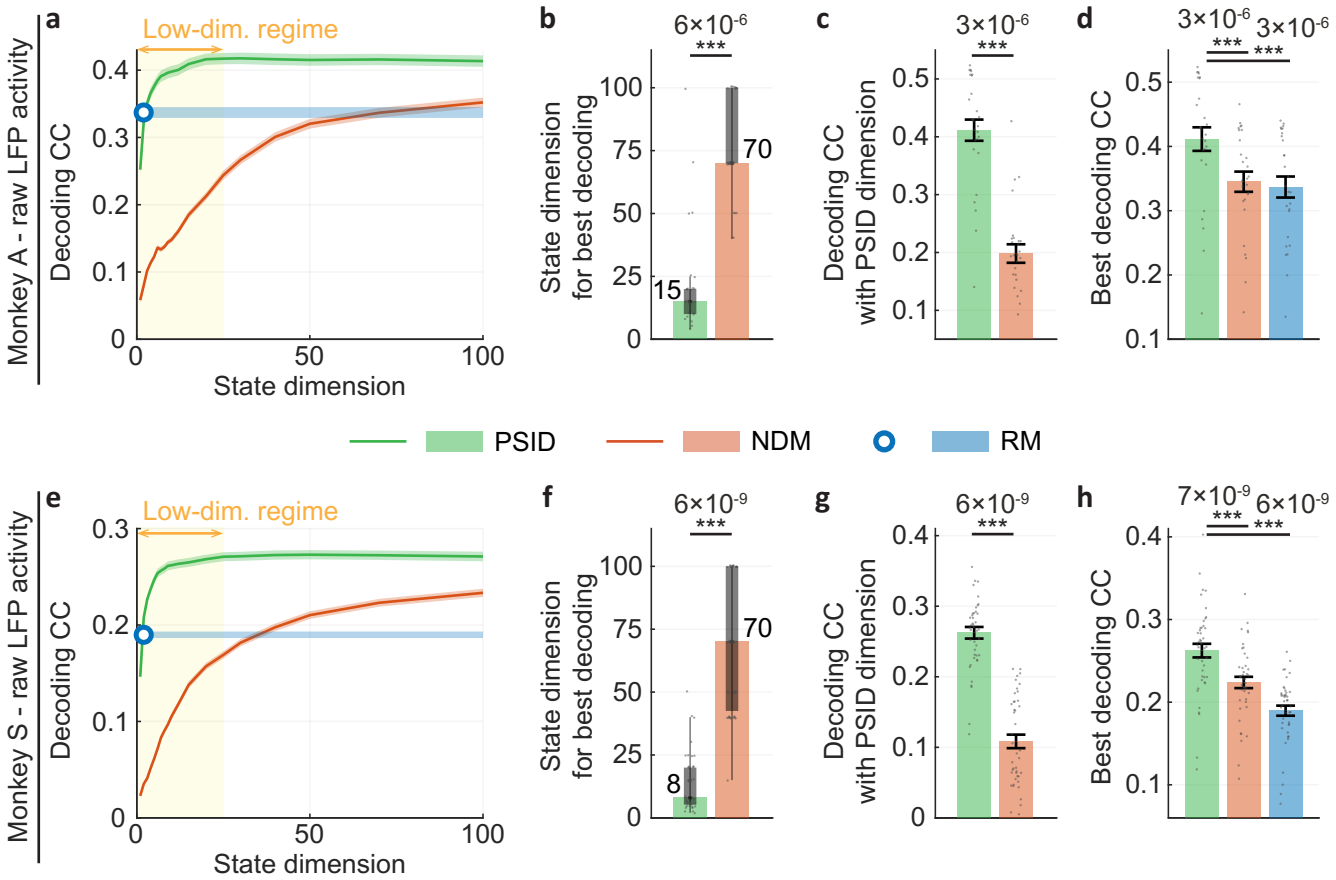

**Extended Data Fig. 7 | Extraction of bidirectional rotational dynamics using PSID was robust to brain region and held also when modeling neural activity within different cortical regions separately.** (**a**) Average trajectory of 2D states identified by PSID during reach and return epochs, when neural activity within different cortical areas is modeled separately. Figure convention is the same as in Fig. 5c. (**b**) Decoding accuracy of using the 2D PSID states or the 2D NDM states (from (a)) to decode behavior. Figure convention is the same as in Fig. 5e. (**c**)-(**d**) Same as (a)-(b) for monkey C. In both monkeys, 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) and resulted in more accurate decoding of behavior (panels b,d; $P < 10^{-11}$ with the exact values noted above asterisks in the plot; one-sided signed-rank; $n = 70$ and $n = 60$ datasets for monkeys J and C, respectively).




**Extended Data Fig. 8 | Similar to NDM, PCA and jPCA extract rotations that are in the same direction during reach and return epochs.** (**a**) Figure convention is the same as in Fig. 5c for projections to the 2D space extracted using PCA (that is top two PCs). Decoding for these and higher-dimensional PCA-extracted states is provided in Supplementary Fig. 6. (**b**) Same as (a) for monkey C. (**c**) Same as (a) for projections to 2D spaces extracted using jPCA[21]. (**d**) Same as (c) for monkey C.




**Extended Data Fig. 9 | PSID again achieved better decoding using lower-dimensional latent states when RNN-based nonlinear NDM always used a dynamic latent state with much higher dimension of 64 and/or when RNN-based nonlinear NDM used a Poisson observation model with a faster time step.** (**a**)-(**h**) Figure convention and number of datasets in all panels is the same as in Fig. 6, with additional configurations for the RNN-based nonlinear NDM method (that is LFADS) added to the comparison (Methods). As in Fig. 6, the dimension of the initial condition for LFADS is always 64. The alterations from Fig. 6 are as follows. First, in Fig. 6, the state dimension for LFADS—which we use to refer to generator RNN's state dimension[23] since it has the same role as the state dimension in a state-space model and determines how many numbers are used to represent the generator state at a given time step and generate the dynamics at the next time step (Methods)—was set to the number of factors to provide a directly comparable result with other methods (Methods; with this choice, number of LFADS factors is equal to its state dimension and is thus comparable with the state dimension in other methods). Here, instead, we also consider always setting the LFADS generator state dimension to 64 regardless of the number of factors. Thus, for this configuration of LFADS, the horizontal axis in panels a,e and the vertical axis in panels b,f only refer to number of factors, which is always smaller than the LFADS state dimension of 64. Again in this case where nonlinear NDM always uses 64-dimensional states to describe the dynamics, PSID reveals a markedly lower dimension than the number of factors in nonlinear NDM, and achieves better decoding than nonlinear NDM. Second, in Fig. 6, to provide a directly comparable result with PSID, the same Gaussian smoothed spike counts with 50 ms bins were used for both PSID and LFADS as input (Methods). Here, instead, we also allow LFADS to use non-smoothed spike counts with 10 ms bins and a Poisson observation model (Methods). For nonlinear NDM, switching the observation model from Gaussian to Poisson improved the peak decoding in monkey J ($P < 10^{-3}$; one-sided signed-rank; $n = 26$ datasets), while both observation models achieved similar decoding in monkey C ($P > 0.07$; two-sided signed-rank; $n = 16$ datasets). Nevertheless, comparisons with PSID remained as before for all these nonlinear NDM configurations (regardless of it using Poisson or Gaussian observations): PSID revealed a markedly lower dimension than the number of factors in nonlinear NDM (panels b,f; $P < 0.004$; one-sided signed-rank; $n \geq 16$ datasets) and achieved better decoding than even a nonlinear NDM with a larger number of factors than the PSID state dimension (panels a,c,d,e,g,h; $P < 0.03$; one-sided signed-rank; $n \geq 16$ datasets). $^{*}P < 0.05$, $^{**}P < 0.005$, $^{***}P < 0.0005$. Statistical test details and exact $P$-values are as in Fig. 6 for linear NDM and RM and are provided in Supplementary Table 1 for the nonlinear NDM variations. This result is because nonlinear NDM, similar to linear NDM and unlike PSID, only considers neural activity when learning the dynamic model. This shows that the PSID advantage is in its novel formulation and two-stage approach for *learning* the dynamic model by considering both neural activity and behavior.




**Extended Data Fig. 10 | PSID reveals low-dimensional behaviorally relevant dynamics in prefrontal raw LFP activity during a task with saccadic eye movements.** (**a**)-(**h**) Figure convention for all panels is the same as in Fig. 3a–d, shown here for a completely different behavioral task, brain region, and neural signal type. Here monkeys perform saccadic eye movements while PFC activity is being recorded (Methods). Raw LFP activity is modeled and the behavior consists of the 2D position of the eye. Similar results hold with PSID more accurately identifying the behaviorally relevant neural dynamics than both NDM and RM. PSID again reveals a markedly lower dimension for behaviorally relevant neural dynamics than NDM. Also, note that RM provides no control over the dimension of dynamics and is forced to use a state dimension equal to the behavior dimension ($n_z$ =2), which in this case is an underestimation of dimension of behaviorally relevant dynamics in neural activity as evident by RM's much worse decoding accuracy compared with PSID. Statistical tests are one-sided signed-rank for which the *P*-values are noted above the asterisks ($n = 27$ and $n = 43$ datasets in monkeys A and S, respectively).

nature research

Corresponding author(s): Maryam M. Shanechi

Last updated by author(s): 9/21/2020

# Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see our Editorial Policies and the Editorial Policy Checklist.

## Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

| n/a | Confirmed |
|---|---|
| ☐ | ☒ The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement |
| ☐ | ☒ A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly |
| ☐ | ☒ The statistical test(s) used AND whether they are one- or two-sided<br>*Only common tests should be described solely by name; describe more complex techniques in the Methods section.* |
| ☐ | ☒ A description of all covariates tested |
| ☐ | ☒ A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons |
| ☐ | ☒ A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals) |
| ☐ | ☒ For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted<br>*Give P values as exact values whenever suitable.* |
| ☐ | ☒ For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings |
| ☐ | ☒ For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes |
| ☐ | ☒ Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated |

*Our web collection on statistics for biologists contains articles on many of the points above.*

## Software and code

Policy information about availability of computer code

| | |
|---|---|
| Data collection | Kinematic data were acquired using the Cortex software package (version 5.3) to track retroreflective markers in 3D (Motion Analysis, Inc USA). Joint angles were solved from the 3D marker data using a Rhesus macaque musculoskeletal model via the SIMM toolkit (version 4.0, MusculoGraphics Inc., USA). The visual stimuli in the task with saccadic eye movements were controlled via custom LabVIEW (version 9.0, National Instruments) software executed on a real-time embedded system (NI PXI-8184, National Instruments). |
| Data analysis | Custom code (MATLAB version R2019a) for the PSID algorithm is available online at https://github.com/ShanechiLab/PSID<br>Support vector regression analyses (Fig. 6 and Supplementary Fig. 13) were done using the open source LIBSVM library (version 321).<br>RNN analyses (Fig. 6 and Extended Data Fig. 9) were done using the open source LFADS tensorflow library (version 2019).<br>Muscle activations (Supplementary Fig. 14) were inferred using the open source OpenSim library (version 4.0). |

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research guidelines for submitting code & software for further information.

## Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A list of figures that have associated raw data
- A description of any restrictions on data availability

The data used to support the results are available upon reasonable request from the corresponding author.

# Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

# Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

| | |
|---|---|
| Sample size | A sample size of four non-human primate (NHP) subjects was used, consisting of two NHP subjects for each of the two behavioral tasks. Demonstration in two NHPs is standard for NHP electrophysiology studies and similar to those reported in previous publications (a list of such publications is provided in Online Methods). All results held for all subjects. |
| Data exclusions | No data was excluded from the study. |
| Replication | Results were replicated in two subjects for each experimental task and all attempts at replication were successful. |
| Randomization | Not relevant for this study. Identical analyses were performed on data from each subject and the results were reported for each subject. There was no grouping of subjects. |
| Blinding | Not relevant for this study. There was no group allocation. |

# Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

## Materials & experimental systems

| n/a | Involved in the study |
|---|---|
| ☒ | ☐ Antibodies |
| ☒ | ☐ Eukaryotic cell lines |
| ☒ | ☐ Palaeontology and archaeology |
| ☐ | ☒ Animals and other organisms |
| ☒ | ☐ Human research participants |
| ☒ | ☐ Clinical data |
| ☒ | ☐ Dual use research of concern |

## Methods

| n/a | Involved in the study |
|---|---|
| ☒ | ☐ ChIP-seq |
| ☒ | ☐ Flow cytometry |
| ☒ | ☐ MRI-based neuroimaging |

# Animals and other organisms

Policy information about studies involving animals; ARRIVE guidelines recommended for reporting animal research

| | |
|---|---|
| Laboratory animals | Four adult male rhesus macaques (macaca mulatta) ages 5 (monkey J), 6 (monkey S), 8 (monkey A), and 8 (monkey C) years old. |
| Wild animals | This study did not involve wild animals. |
| Field-collected samples | This study did not involve field-collected samples. |
| Ethics oversight | All surgical and experimental procedures were performed in compliance with the National Institute of Health Guide for Care and Use of Laboratory Animals and were approved by the New York University Institutional Animal Care and Use Committee. |

Note that full information on the approval of the study protocol must also be provided in the manuscript.