Modeling and dissociation of intrinsic and input-driven neural population dynamics underlying behavior

Parsa Vahidi[1,†], Omid G. Sani[1,†], Maryam M. Shanechi[1,2,*]

1 Department of Electrical and Computer Engineering, University of Southern California

2 Neuroscience Graduate Program, Department of Computer Science, and Department of Biomedical Engineering, University of Southern California

† Equal contribution.

* Corresponding author: shanechi@usc.edu

Abstract

Neural dynamics can reflect intrinsic dynamics or dynamic inputs, such as sensory inputs or inputs from other regions. To avoid misinterpreting temporally-structured inputs as intrinsic dynamics, dynamical models of neural activity should account for measured inputs. However, incorporating measured inputs remains elusive in joint dynamical modeling of neural-behavioral data, which is important for studying neural computations of a specific behavior. We first show how training dynamical models of neural activity while considering behavior but not input, or input but not behavior may lead to misinterpretations. We then develop a novel analytical learning method that simultaneously accounts for neural activity, behavior, and measured inputs. The method provides the new capability to prioritize the learning of intrinsic behaviorally relevant neural dynamics and dissociate them from both other intrinsic dynamics and measured input dynamics. In data from a simulated brain with fixed intrinsic dynamics that performs different tasks, the method correctly finds the same intrinsic dynamics regardless of task while other methods can be influenced by the change in task. In neural datasets from three subjects performing two different motor tasks with task instruction sensory inputs, the method reveals low-dimensional intrinsic neural dynamics that are missed by other methods and are more predictive of behavior and/or neural activity. The method also uniquely finds that the intrinsic behaviorally relevant neural dynamics are largely similar across the three subjects and two tasks whereas the overall neural dynamics are not. These input-driven dynamical models of neural-behavioral data can uncover intrinsic dynamics that may otherwise be missed.

Introduction

Neural population activity exhibits rich temporal structures[1–26]. Investigating these temporal structures, i.e., dynamics, can reveal the neural computations that underlie behavior[5,6,12,15,16,19,20]. Much progress has been made in developing models that can describe the dynamics of neural population activity using a low-dimensional latent state[2–4,7,8,10–14,16]. However, a major challenge in such investigations is that neural dynamics can arise due to two distinct sources that reflect distinct computations[12,15,27]. The first source consists of the intrinsic dynamics within a given brain region. Intrinsic dynamics arise due to the recurrent connections within a region's neuronal population as it responds in a temporally structured manner to any excitations from within or outside that region[6,12,15,18,27,28]. The second source consists of input dynamics. These input dynamics are temporal structures that already exist in inputs that reach the recorded brain region, including sensory inputs or inputs from other brain regions[1,9,12,15,27,29–31]. While measuring all inputs is infeasible experimentally, measurements of sensory inputs such as task instructions or partial measurements of neural inputs into a brain region are often possible. As such, correctly interpreting how neural computations in a given brain region give rise to a specific behavior can greatly benefit from simultaneously achieving two objectives, which remain elusive.

First, given the above two sources, neural dynamics that are intrinsic to a given brain region need to be dissociated from those that are simply due to temporally structured measured inputs to that region. Second, within intrinsic neural dynamics, those that are relevant to the specific behavior of interest need to be dissociated from other intrinsic neural dynamics. This latter dissociation is important because neural dynamics of a specific behavior often constitute a minority of the total variance in the recorded neural activity[5,6,19,32–39]. Consistent with this observation, recent work has shown that learning dynamical models of neural-behavioral data together and in a way that dissociates and prioritizes their shared dynamics can unmask behaviorally relevant neural dynamics that may otherwise not be found[19,20]. We refer to this prioritized learning approach for neural-behavioral data as preferential dynamical modeling because it preferentially models the behaviorally relevant neural dynamics with priority instead of non-preferentially modeling prevalent dynamics in neural data as is typically done. But prior methods for preferential dynamical modeling cannot account for the effect of measured inputs to a given brain region[19]. Thus, the

dissociation of intrinsic and input-driven neural population dynamics that underlie specific behaviors has remained challenging.

Here, we first show how misinterpretation and incorrect identification of intrinsic behaviorally relevant dynamics could result from modeling neural activity while considering behavior but not input, or while considering input but not behavior. Indeed, modeling neural activity without considering the measured input could result in a model that mistakes the temporal structure in the input as part of the intrinsic dynamics within the recorded brain region[9,27] and consequently confound scientific conclusions. For non-preferential modeling of neural activity on its own, while not commonly done, various methods can be adapted to fit models with measured inputs[40], thus accounting for measured input and neural activity but not behavior. However, as we show, even these non-preferential methods with input can miss those intrinsic neural dynamics that are behaviorally relevant. Further, methods for preferential modeling that consider the neural-behavioral data together cannot consider measured inputs. These results motivate the critical need for developing novel methods that can simultaneously consider neural activity, behavior, and measured inputs when learning a dynamical model (see Discussion). It is also important to recognize that disentanglement of intrinsic and input dynamics is fundamentally limited by the extent to which measured inputs are available, which depends on the experimental capability for input measurement. Perfect disentanglement requires measuring all inputs, which is typically not feasible with current experimental technology. As such, our aim here is to mathematically formulate a learning problem that involves neural activity, behavior, and measured inputs simultaneously, and to dissociate the intrinsic behaviorally relevant neural dynamics from the dynamics of any measured inputs and from other intrinsic neural dynamics. As we will show in our results, even this partial dissociation using the measured inputs (e.g., task instructions) can already lead to more accurate models and inferences, and to useful new insights compared with prior methods that account for either measured input or behavior during learning but not both.

With the above motivation, we then provide a new preferential modeling approach, termed Input Preferential Subspace Identification (IPSID) that can consider both measured inputs and behaviors in the training set while learning dynamical models of neural population activity. By doing so, IPSID provides

the new capability to learn the intrinsic behaviorally relevant neural dynamics *with priority* and dissociate them both from other intrinsic neural dynamics and from the dynamics of measured inputs. We also develop a version of IPSID that achieves this capability when some input dynamics influence the behavior through pathways that are neither recorded nor downstream of the recorded neural activity. Compared with prior preferential modeling methods (i.e., PSID)[19,41], which cannot incorporate input and thus do not dissociate intrinsic and input dynamics, IPSID requires distinct mathematical operations including completely new steps (**Note S1**). We show that two new capabilities provided by IPSID are critical for accurate dissociation of intrinsic behaviorally relevant neural dynamics: prioritized learning of these dynamics in the presence of input, and ensuring all learned dynamics are directly present in the neural recordings even when inputs affect behavior.

We validate IPSID and its new capabilities in extensive numerical simulations of diverse dynamical systems and in two independent motor cortical datasets from three non-human primates (NHP) recorded during two different tasks with task instruction sensory inputs. First, we simulate a brain with fixed intrinsic dynamics that performs different behavioral tasks. IPSID correctly learns the same intrinsic behaviorally relevant neural dynamics regardless of which specific task is used to collect the simulated training neural data. In contrast, other methods learn intrinsic dynamics that are inaccurate and influenced by the specific task. Second, we apply IPSID to motor cortical population activity recorded from three NHPs in two independent datasets with two different 2-dimensional (2D) cursor-control tasks. IPSID finds intrinsic behaviorally relevant dynamics that not only predict motor behavior better than non-preferential methods even with input, but also predict neural activity better than preferential methods which cannot consider task instruction inputs. Further, IPSID reveals that intrinsic behaviorally relevant neural dynamics are largely similar across the three animals despite differences in the two cursor-control tasks and animals, while other methods miss these similar dynamics. By dissociating intrinsic behaviorally relevant dynamics from both other intrinsic dynamics and measured input dynamics, IPSID can help explore unanswered questions regarding how intrinsic and input-driven neural computations give rise to behavior across subjects and tasks.

Methods

Modeling intrinsic neural dynamics underlying behavior in the presence of inputs

To see the effect of input on misinterpretation of intrinsic neural dynamics, consider a task where a subject is instructed to follow an on-screen target with their hand while motor cortical activity that represents the hand movements is recorded (**Fig. 1a**). Here, movements of the target would result in corresponding movements in the hand that follows the target, and thus would also introduce corresponding dynamics in the neural activity that represents hand movements. Consequently, any arbitrary movement of the target will be, to some extent, reflected in the recorded neural activity. An example is shown in a numerical simulation in **Fig. 1a,b**. As another example, if the target moves up and down with a 1s period, one would expect the neural activity to also include similar periodic patterns with a 1s period. If the period of target movements changes to 2s, so would the period of the patterns in neural activity that represent the hand movements. Any neural modeling that is not informed by target movements, which serve as task instruction sensory inputs, cannot distinguish between such input dynamics and intrinsic dynamics that originate in the recorded brain region. Thus, such modeling may incorrectly conclude that there exist intrinsic dynamics originating in the recorded brain area that are periodic with a 1s period. This means that sensory inputs or inputs from other brain regions, if unaccounted for, may confound dynamical models of neural activity by being misinterpreted as intrinsic dynamics. The reflection of input dynamics in neural dynamics can also be seen in terms of the frequency domain spectrum of these signals (**Fig. 1b**). In this view, the correct dissociation of intrinsic dynamics from input dynamics requires the correct learning of the transfer function from inputs to neural signals, in a way that doesn't incorrectly attribute the input dynamics that appear in neural activity to having originated from the transfer function (**Fig. 1b**).

When performing non-preferential dynamic modeling of neural activity on its own, though not common, various methods such as subspace identification[40] can be leveraged to fit models with measured input. However, as we will show, these methods can lead to inaccurate identification of intrinsic behaviorally relevant neural dynamics as behavior is not considered during learning. Further, current methods cannot account for measured inputs in preferential modeling of neural-behavioral data together, which we now

enable by developing IPSID. To formulate the goal of IPSID, we represent the dynamical state of the recorded brain regions as a high-dimensional vector, of which each dimension may or may not contribute to generating the specific behavior of interest, i.e., be behaviorally relevant (**Fig. 1a**). Thus, in investigations concerned with behavior in the presence of a measured input, there are two major confounding factors in learning intrinsic behaviorally relevant neural dynamics: (1) the dynamics of the measured input that may be incorrectly considered part of intrinsic behaviorally relevant neural dynamics, and (2) other intrinsic neural dynamics that may mask or confound the behaviorally relevant ones.

IPSID addresses both confounding factors by accounting for neural activity, behavior, and measured input simultaneously during learning via new algebraic operations. Doing so, IPSID models and dissociates the intrinsic behaviorally relevant neural dynamics from measured input dynamics and from other intrinsic neural dynamics. Unlike IPSID, prior methods address only one or the other confound but not both. First, non-preferential neural dynamic modeling (NDM) with input (**SI Methods**), which we term INDM, accounts for neural activity and measured input but not behavior during learning. As such, INDM may miss or confound the intrinsic neural dynamics that are behaviorally relevant. Second, a method termed PSID[19,41] addresses the second confound by accounting for neural activity and behavior during learning but not input. As such, PSID cannot dissociate intrinsic and input dynamics. We thus use this naming convention for ease of exposition but the algebraic operations in IPSID are different from those in both PSID and INDM, and further IPSID requires additional new mathematical steps compared with these prior methods (**Notes S1, S2**).

In IPSID, we use the following linear state-space model to jointly describe the dynamics of neural activity (y_k) and behavior (z_k) in the presence of measured input (u_k)

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

where $x_k \in \mathbb{R}^{n_x}$ is the latent neural state composed of two parts: (1) $x_k^{(1)} \in \mathbb{R}^{n_1}$, which is the behaviorally relevant latent states in the recorded neural activity; (2) $x_k^{(2)} \in \mathbb{R}^{n_x - n_1}$, which is the other latent states in the recorded neural activity. In this model, $y_k \in \mathbb{R}^{n_y}$, $z_k \in \mathbb{R}^{n_z}$ and $u_k \in \mathbb{R}^{n_u}$ represent the recorded

neural activity, the measured behavior, and the measured input, respectively. Here $x_k^{(1)}$ being behaviorally relevant means that only those dimensions of x_k corresponding to $x_k^{(1)}$ contribute to generating behavior (z_k) in the third row of equation (1). Finally, w_k and v_k are zero mean white Gaussian noises (**SI Methods**) and ϵ_k is a general Gaussian random process representing any behavior dynamics not encoded in the recorded neural activity (i.e., not driven by x_k).

Prior works have not addressed the problem of fitting this model in a way that dissociates and prioritizes the learning of behaviorally relevant latent states. To enable such preferential/prioritized learning, we developed IPSID, which uses new linear algebraic operations to directly extract the subspace of intrinsic behaviorally relevant latent states from neural, behavioral and input training data, and then learns the model parameters. IPSID provides a new two-stage learning procedure that incorporates input as follows. In the first stage of IPSID, we develop algebraic operations that extract the behaviorally relevant latent states with *priority* via an oblique (non-orthogonal) projection of future behavior onto past neural activity and past inputs along the subspace spanned by future inputs (**Fig. S1, SI Methods**). Then, in an optional second stage, we devise algebraic operations that extract any other latent neural states by another oblique projection from any residual/unexplained future neural activity onto past neural activity and past inputs along future inputs (**Fig. S1**). Model parameters are then learned via least squares based on the extracted latent states and their relation in equation (1). This two-stage learning enables the new capability for prioritized learning of the intrinsic behaviorally relevant neural dynamics over other intrinsic neural dynamics in the presence of inputs because the former dynamics are learned first, i.e., in the first stage. The two-stage learning allows us to learn a minimally complex model of the intrinsic behaviorally relevant neural dynamics in the first stage (i.e., a model with low-dimensional states), instead of having to learn a more complex model that simultaneously includes all intrinsic neural dynamics. This leads to learning more accurate models of intrinsic behaviorally relevant dynamics for a given dataset as shown in extensive simulations and in real neural data analyses below. After the model is learned, in the test set, extraction of intrinsic behaviorally relevant neural dynamics is done without looking at behavior and via a Kalman filter associated with the learned model (**SI Methods**). Details of IPSID are provided in **SI Methods** and **Notes S1-S2**.

We found that among the learning methods, the only method that correctly learns the intrinsic behaviorally relevant neural dynamics in the presence of inputs is the new IPSID (**Fig. 1d**) as demonstrated below. Further, by implementing a numerical optimization approach that maximizes the data likelihood with block-structured constraints on model parameters (**SI Methods**), we demonstrate the critical importance of two new capabilities provided by IPSID for dissociating the intrinsic behaviorally relevant neural dynamics: i) *prioritized* learning of these dynamics enabled via the new two-stage learning algorithm in the presence of input as described above, and ii) ensuring all learned latent states are directly present in neural recordings even when inputs affect behavior. To assess the methods, we look at the eigenvalues of the latent state transition matrix A, which quantify the dynamics (**SI Methods**, **Fig. 1c,d**). We also compute the accuracy in decoding behavior from neural activity as well as in neural self-prediction – defined as predicting neural activity one-step-ahead from its own past (**SI Methods**).

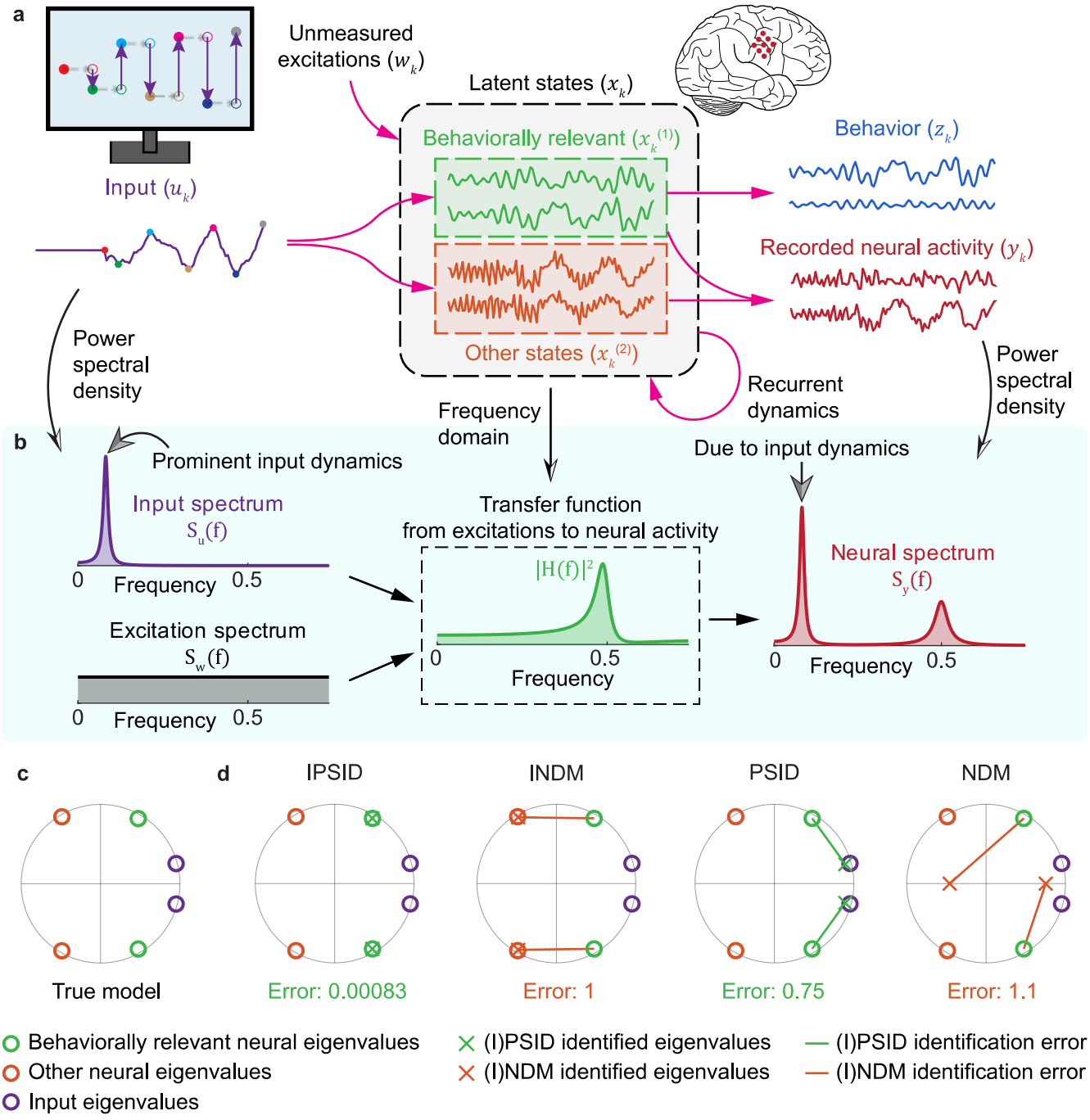

Fig. 1 | Intrinsic behaviorally relevant neural dynamics may be confounded by other intrinsic neural dynamics as well as by measured input dynamics, a challenge that the new IPSID method resolves.

(a) Data generated from a simulated brain following equation (1) with a 1D input and a 4D latent state out of which only 2 dimensions (green) drive behavior. The input is taken as the sensory input such as target position moving up and down on a screen as depicted, but input can also consist of measured activity from other upstream brain regions. Neural dynamics that arise from the recurrent dynamics of neuronal networks within the brain region constitute the intrinsic neural dynamics. Oscillating temporal pattens of the input (left) constitute the input dynamics and clearly also appear in the neural activity (right) in a way that is mixed with the intrinsic neural dynamics. **(b)** Appearance of input dynamics in neural dynamics can also be clearly seen in the frequency domain representation of (a), showing: the power spectral density (PSD), or spectrum, of input time series $S_u(f)$ (top-left); PSD of unmeasured excitations $S_w(f)$ modeled as white Gaussian noise (bottom-left); transfer function from inputs to the neural activity (middle); and PSD of neural activity (right). Neural activity exhibits two dominant frequency components. In this simulation, the lower-frequency component is the reflection of input dynamics while

the higher-frequency component represents intrinsic neural dynamics (as it is also present in the transfer function). Horizontal axes show the normalized frequency with 1 being the maximum possible frequency, i.e., π. **(c)** The eigenvalues of the latent state transition matrix A in the simulated brain model in equation (1). **(d)** Learned eigenvalues using (I)PSID or (I)NDM. Red lines indicate the error in the learned eigenvalues. The normalized error value—average line length normalized by the average true eigenvalue magnitude—is noted below each plot. Only IPSID correctly learns the intrinsic behaviorally relevant neural dynamics as quantified by the eigenvalues (**SI Methods**). Unlike IPSID, NDM or PSID may not learn the correct intrinsic dynamics but instead learn dynamics (eigenvalues) that are deflected towards the input dynamics (eigenvalues) in this example.

Results

IPSID correctly learns all model parameters in the presence of inputs

We first validated the accurate learning of intrinsic behaviorally relevant neural dynamics using IPSID in a simulated model with 4-dimensional latent states out of which only two dimensions were involved in generating the simulated behavior (**Fig. 1a**). The eigenvalues of the state transition matrix A affect the transfer function from the input to the states and neural activity (**Fig. 1b**), characterize the state response to excitations, and describe the dynamics (**Fig. 1c**, **SI Methods**). We thus use these eigenvalues to quantify the intrinsic neural dynamics (**SI Methods**). We found that the new IPSID was the only method that correctly learned the eigenvalues associated with the intrinsic behaviorally relevant neural dynamics (**Fig. 1d**). In contrast, NDM or PSID that do not consider inputs learned models that were confounded by input dynamics and INDM that does not consider behavior was confounded by other intrinsic neural dynamics beyond the behaviorally relevant ones.

To more generally validate IPSID, we applied it to data generated from 100 random models in the form of equation (1) with random parameters and dimensions (**SI Methods**). To provide input to these models, we independently simulated another 100 models without input (equation (3) from **SI Methods**) with random parameters and passed their output as the input to the original models—these inputs are thus generated by an independent dynamical system and can be thought of as activity of other brain regions or as structured sensory inputs. We found that IPSID correctly learned all model parameters in the presence of inputs (**Fig. S2**). Moreover, the rate of convergence of parameters as a function of training samples was similar to INDM (**Fig. S2b**); this suggests that despite its additional capability in dissociating

those intrinsic neural dynamics that are behaviorally relevant, IPSID does not require more training data than INDM even when modeling all latent states.

IPSID correctly prioritizes the learning of intrinsic behaviorally relevant neural dynamics in the presence of inputs

In another numerical simulation, we found that IPSID correctly prioritizes the learning of intrinsic behaviorally relevant neural dynamics in the presence of inputs, which is an important new capability for learning these dynamics accurately (**Fig. 2,** also see **Fig. 4** later). Thus, IPSID addresses the challenge of preferential dynamical modeling of neural-behavioral data with inputs (**Fig. 2**). We simulated 100 random models formulated by equation (1) with a 6D latent state, out of which only 2 dimensions were behaviorally relevant (**SI Methods**). To get the input to these models, we independently simulated 100 random models without input (equation (3) from **SI Methods**) with 2D latent states and passed their output as the input to the original models. We then learned models using (I)PSID and (I)NDM with varying latent state dimensions (n_x). In each case, we computed the error in learning the intrinsic behaviorally relevant eigenvalues, which quantifies how accurately intrinsic behaviorally relevant dynamics are learned (**Fig. 2b**, **Fig. S3**).

We found that only IPSID could learn all the intrinsic behaviorally relevant neural dynamics using the minimal latent state dimension of 2, which is the true simulated dimension of these dynamics (**Fig. 2b**, **Fig. S4**). IPSID was able to achieve this by considering both inputs and behavior in its preferential modeling of neural dynamics during learning. This meant that IPSID could simultaneously dissociate the intrinsic behaviorally relevant neural dynamics form other intrinsic neural dynamics and from input dynamics. In contrast, NDM and PSID do not consider the input and thus were unable to dissociate the intrinsic versus input dynamics, leading to a high eigenvalue error (**Fig. 2b**). Further, even though INDM considers inputs, it does not consider behavior during learning and thus it required much larger latent state dimensions to learn the intrinsic behaviorally relevant eigenvalues (**Fig. 2b**). Also, even INDM with a higher state dimension (i.e., 6) had larger eigenvalue error when using the same number of training samples as IPSID (**Fig. 2c**); this is because models with higher dimensional states are more complex

and thus more difficult to learn. Indeed, IPSID required orders of magnitude fewer training samples to learn the intrinsic behaviorally relevant neural dynamics in the presence of inputs (**Fig. 2c**).

We next found that NDM and PSID models, which do not consider input, could not accurately learn the intrinsic behaviorally relevant dynamics even by increasing their state dimension. Specifically, we first learned an NDM/PSID model with a high latent state dimension that learns a mixture of all intrinsic neural dynamics and input dynamics. We then reduced this model, as we did with INDM above, by only keeping the two dimensions that were best in decoding behavior and looking at their associated eigenvalues (**SI Methods**). Even with this approach, the reduced models were still much less accurate than low-dimensional models learned with IPSID (see **Fig. 2b** at high dimensions).

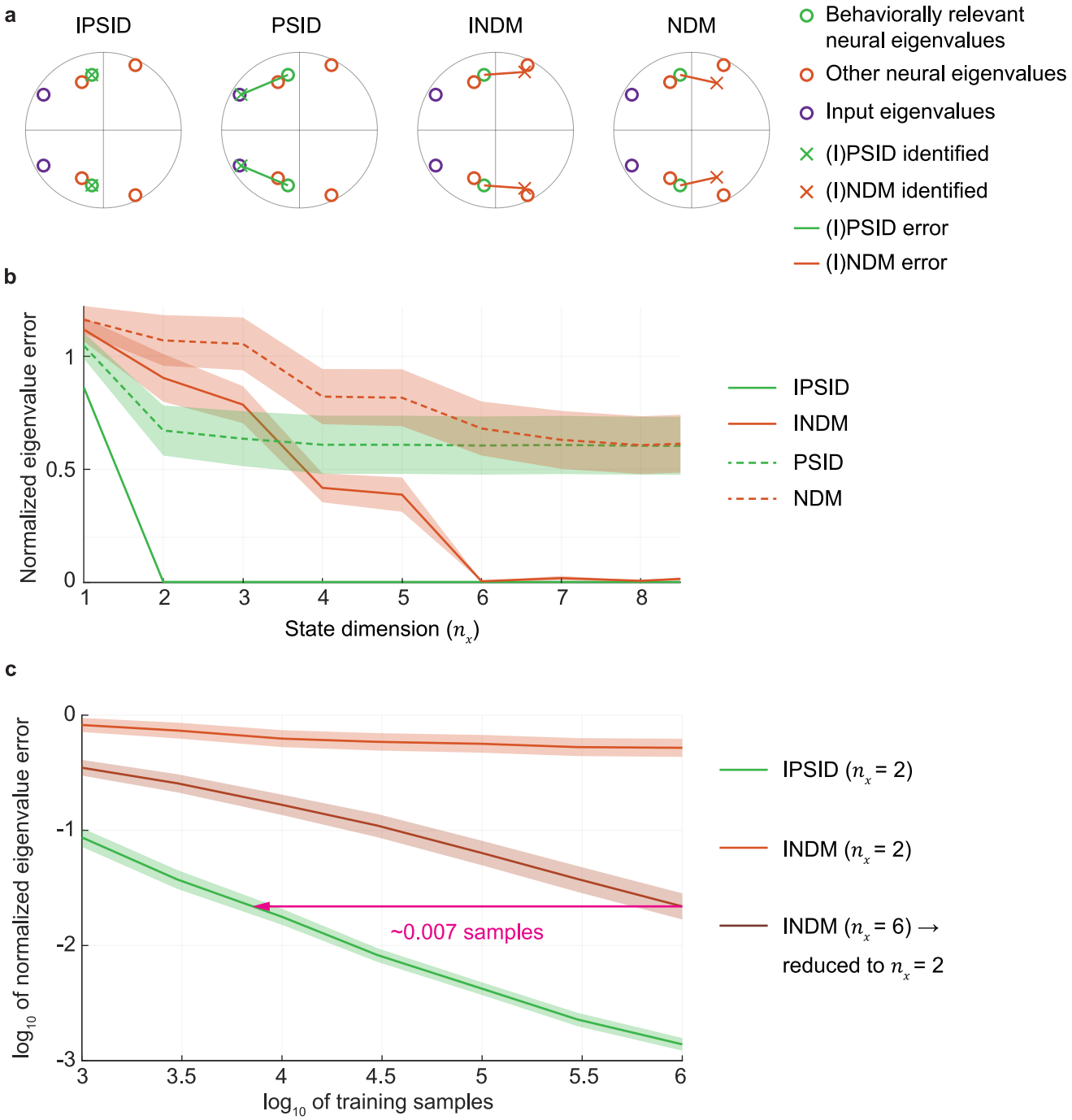

Fig. 2 | IPSID correctly prioritizes the learning of intrinsic behaviorally relevant neural dynamics thus achieving preferential neural-behavioral modeling even in the presence of input.

(a) For one simulated model (equation (1)), the identified intrinsic behaviorally relevant eigenvalues are shown for (I)PSID and (I)NDM using a 2D latent state. Eigenvalues of the state transition matrix A in the true model are shown as colored circles. Crosses show the identified behaviorally relevant eigenvalues when modeling the neural activity. **(b)** Normalized error of learning the intrinsic behaviorally relevant eigenvalues given 10^6 training samples is shown when using (I)PSID and (I)NDM, averaged over 100 random models each with total latent state dimension of n_x = 6 and behaviorally relevant state dimension of $n_1 = 2$. For all models, an independent random model with state dimension of 2 generated the input (**SI Methods**). Solid lines show the average across models and shaded areas show the s.e.m. (n = 100 random models). For all methods, we vary the state dimension n_x from 1 to 8; for $n_x > 2$, we find the 2 state dimensions that best predict behavior and evaluate their 2 associated eigenvalues (**SI Methods**). We find that to learn the intrinsic behaviorally relevant eigenvalues, IPSID only needs a minimal state dimension n_x = 2 (true n_1) whereas INDM needs a high state dimension $n_x = 6$ (true total model dimension

n_x). This also leads to INDM's higher error with the same training sample size in (c). Also, even using high-dimensional states, NDM and PSID cannot dissociate which of the eigenvalues are intrinsic and thus do not learn the correct reduced models (because they do not consider input). **(c)** Normalized error of learning the intrinsic behaviorally relevant eigenvalues vs. training samples for 100 random models. For INDM, we try i) directly learning a model with a 2D latent state and ii) first learning a model with a high enough dimension to achieve almost zero error in (b) and then reducing the model to keep the top 2 dimensions with the best behavior decoding (indicated by dimension → 2) (**SI Methods**). INDM requires orders of magnitude more training samples than IPSID to learn the intrinsic behaviorally relevant eigenvalues with similar accuracy.

IPSID can dissociate the effects of input on behavior that are reflected in the recorded neural activity from those that are not

In equation (1), all the effects of input on behavior happen through latent states that are reflected in the recorded neural activity. In this scenario, all the downstream regions of the input are either covered in the recordings or reflected in them (e.g., are downstream of the recorded regions). In addition to this scenario, we now show that IPSID can also apply to a new scenario where inputs may also influence behavior through pathways/regions that are neither recorded nor reflected in the recorded activity (**Fig. 3a**). We formulate this new scenario with the following model

$$\begin{cases} \begin{bmatrix} x_{k+1}^{(1)} \\ x_{k+1}^{(2)} \\ x_{k+1}^{(3)} \end{bmatrix} = \begin{bmatrix} A_{11} & 0 & 0 \\ A_{21} & A_{22} & 0 \\ 0 & 0 & A_{33} \end{bmatrix} \begin{bmatrix} x_k^{(1)} \\ x_k^{(2)} \\ x_k^{(3)} \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix} u_k + \begin{bmatrix} w_k^{(1)} \\ w_k^{(2)} \\ w_k^{(3)} \end{bmatrix} \\ y_k = \begin{bmatrix} C_{y_1} & C_{y_2} & 0 \end{bmatrix} \begin{bmatrix} x_k^{(1)} \\ x_k^{(2)} \\ x_k^{(3)} \end{bmatrix} + D_y u_k + v_k \\ z_k = \begin{bmatrix} C_{z_1} & 0 & C_{z_3} \end{bmatrix} \begin{bmatrix} x_k^{(1)} \\ x_k^{(2)} \\ x_k^{(3)} \end{bmatrix} + \epsilon_k \end{cases} \quad (2)$$

where compared with equation (1), an additional segment $x_k^{(3)}$ is added to the latent state x_k to represent the effects of input on behavior z_k that are not reflected in the recorded neural activity y_k. In this formulation, IPSID dissociates the latent state into three segments: (1) $x_k^{(1)} \in \mathbb{R}^{n_1}$, which is the behaviorally relevant latent state that is reflected in neural activity y_k, (2) $x_k^{(2)} \in \mathbb{R}^{n_2}$, which is the latent state that describes any other neural dynamics, and (3) $x_k^{(3)} \in \mathbb{R}^{n_x - n_1 - n_2}$, which is the behaviorally relevant latent state not reflected in the recorded neural activity y_k. These three types of latent states are shown in an example in **Fig. 3a**. Note that in this case, only $x_k^{(1)}$ and $x_k^{(2)}$ constitute the intrinsic latent

states because only these latent states drive the recorded neural activity. To add support for dissociation of these three types of latent states to IPSID, we developed two additional optional steps for IPSID (**Fig. S5**, **Note S2**).

In the first additional step, before the initial oblique projection of behavior onto neural activity and input, we project behavior onto the subspace of latent states in neural activity (i.e., neural states) irrespective of the relevance of these states to behavior; these neural states are obtained using only the second stage of IPSID (**SI Methods**, **Note S2**, **Figs. S5, S6a**). We then apply IPSID as before (**Note S1**), but now use the results of this additional projection as the behavior signal. This additional projection ensures that behavior dynamics that are not encoded in the recorded neural activity are not included in the first set of states $x_k^{(1)}$.

In the second additional step, we optionally extract $x_k^{(3)}$, which represents any behavior dynamics that are driven by the input but are not encoded in the recorded neural activity. This means that $x_k^{(3)}$ reflects processing in the downstream regions of input that are not recorded/reflected as part of neural activity. In this step, after performing the first additional step above and subsequently both stages of IPSID to extract $x_k^{(1)}$ and $x_k^{(2)}$, we compute the residual behavior that is still not predictable using $x_k^{(1)}$ and $x_k^{(2)}$. Then, using the second stage of IPSID, we build a model that predicts these residual behavior dynamics purely using the input (**SI Methods**, **Note S2**, **Fig. S5**) – this gives $x_k^{(3)}$, which summarizes the direct effect of input on behavior dynamics that are not reflected on the recorded neural activity. Together, these two additional steps enable IPSID to learn a model as in equation (2). If extraction of $x_k^{(3)}$ is not of interest, the second step can be optionally skipped, and solely the first step can be added to IPSID to ensure that $x_k^{(1)}$ and $x_k^{(2)}$ are encoded in the recorded neural activity.

We simulated models in the form of equation (2) and confirmed that with the above additional steps, IPSID correctly dissociates intrinsic behaviorally relevant neural dynamics (i.e., $x_k^{(1)}$) from other dynamics – i.e., from other intrinsic neural dynamics, input dynamics, and behavior dynamics not encoded in the recorded neural activity (**Fig. 3c**). Moreover, across 100 random models, IPSID correctly learned all

model parameters in equation (2) (**Fig. S7**). Finally, by learning $x_k^{(3)}$, which captures the behavior dynamics that are predictable from input but are not reflected in the recorded neural activity, IPSID also achieved ideal prediction of behavior from input and neural activity (**Fig. S8**).

These results demonstrate that IPSID is applicable to scenarios where the recorded neural activity does not cover all the downstream regions of the measured input. In this scenario, IPSID can also dissociate the influences of input on behavior that are reflected in the recorded neural activity from those that are not. Without this capability, some of the learned dynamics may not be present in the recorded region (**Fig. 3c***, top row comparisons*). Thus, this is another new capability by IPSID that is important for accurately dissociating intrinsic behaviorally relevant dynamics in neural recordings.

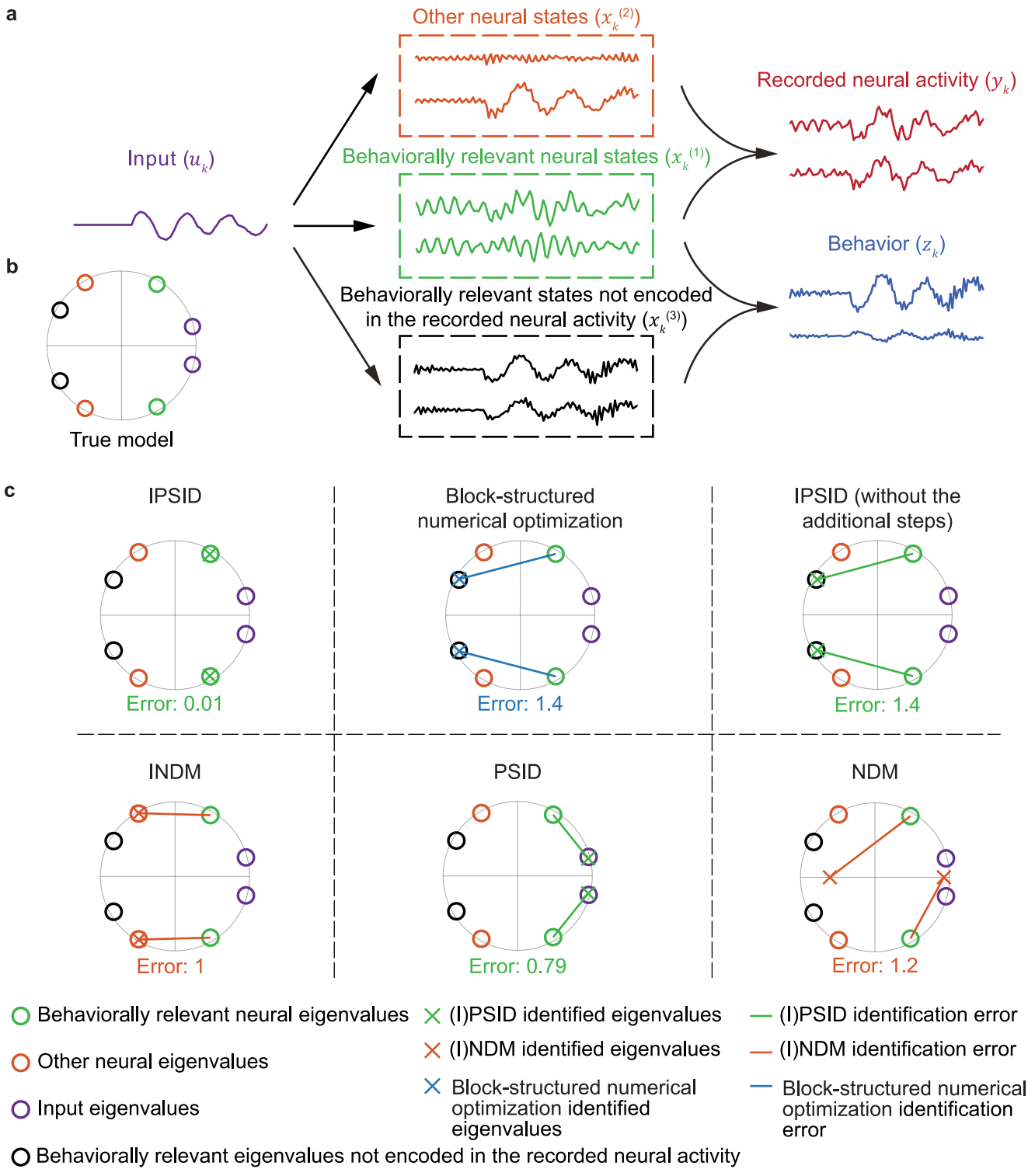

Fig. 3 | IPSID also applies to scenarios when the recorded regions do not cover all downstream regions of the input.
(a) A simulated brain (as in equation (2)) with a 6D latent state out of which only 4 dimensions drive the recorded neural activity and the other 2 dimensions just drive the behavior. **(b)** The eigenvalues of the state transition matrix A in the simulated model. The 4 eigenvalues associated with the 4 state dimensions that drive the recorded neural activity are shown as green and orange circles, depending on whether they drive behavior (green) or not (orange). Eigenvalues associated with the two additional state dimensions that only drive the behavior but not recorded neural activity are shown as black circles. **(c)** Eigenvalues of the models learned using IPSID, block-structured numerical optimization, IPSID (without additional steps), PSID and (I)NDM. A simplified schematic of key operations for each method is in **Fig. S6**. The block-structured numerical

optimization learns the model parameters via gradient descent (**SI Methods**). Notation is as in **Fig. 1**. IPSID can also address this scenario using its additional steps. Only IPSID correctly learns the intrinsic behaviorally relevant neural dynamics even in this scenario, and its new capability via the additional steps is needed to avoid the black eigenvalues/dynamics (behaviorally relevant dynamics not reflected in the recordings; see the top row comparisons).

Prioritized learning of intrinsic behaviorally relevant neural dynamics enabled with IPSID is critical for their accurate dissociation

A new capability provided by IPSID that is critical for disentangling intrinsic behaviorally relevant dynamics is to prioritize their learning over other intrinsic neural dynamics in the presence of input. This prioritized learning is enabled with IPSID's new two-stage learning procedure that incorporates input. As shown earlier in **Fig. 2**, without prioritized learning, more latent states would be needed to ensure that intrinsic behaviorally relevant neural dynamics are included in the model, which will result in much higher error in learning these dynamics for a given training dataset (**Fig. 2c**). To further show the importance of the new prioritized learning capability, we implemented a block-structured numerical optimization approach that fits a model with the same block structure as the IPSID model in equation (6) from **SI Methods**. This optimization fits all model parameters to simultaneously maximize the neural-behavioral data log-likelihood (**SI Methods**). Note that the two-stage learning procedure in IPSID enforces a distinct learning objective, which is future behavior prediction in stage 1 and future residual neural prediction in stage 2. Thus, the IPSID objective is different from the numerical optimization objective, which is to simultaneously optimize the likelihood of neural and behavioral data. We applied the numerical optimization approach to the same simulated data as in **Fig. 2c**. We found that this block-structured numerical optimization approach is significantly less accurate than IPSID for a given number of training samples. Indeed, this approach requires orders of magnitude more training samples compared to IPSID to achieve the same accuracy (**Fig. 4** blue). This analysis shows that simply imposing a set of block-structured model parameters as in equation (6) is not sufficient for accurate disentanglement; rather the new capability for prioritized learning enabled by the two-staged learning approach in IPSID is important for achieving accurate disentanglement.

We also compared the computation times of the block-structured numerical optimization approach vs. IPSID. Model fitting using IPSID, which is based on a fixed set of linear algebraic operations, was

significantly faster than model fitting using the numerical optimization approach (**Fig. S9**). Finally, this numerical optimization approach does not dissociate the effects of input on behavior that are reflected in the recorded neural activity from those that are not. Therefore, this approach may learn behavior dynamics not encoded in the recorded neural activity (**Fig. 3c**). This highlights the importance of the additional steps incorporated in IPSID as another new capability that is important for dissociating the intrinsic behaviorally relevant dynamics that are present in the recorded neural activity, as also explained in the previous section (**Fig. 3c, Fig. S5**, **Note S2**). Future work may be able to use similar ideas as the prioritized learning or these additional steps to develop numerical optimization approaches for the disentanglement problem formulated here (Discussion).

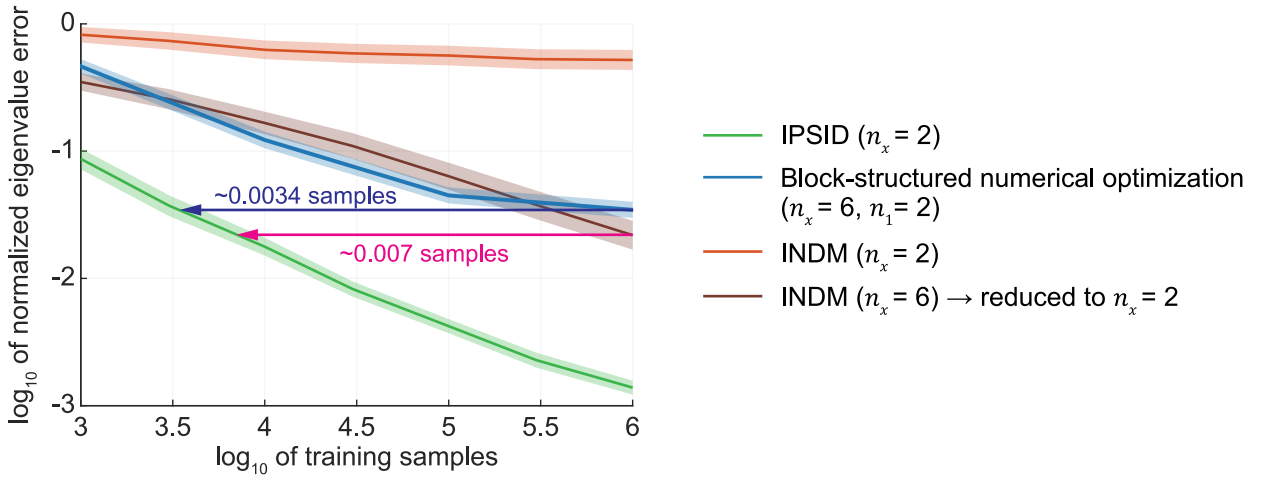

Fig. 4 | IPSID outperforms numerical optimization with block-structured model parameters in terms of accuracy for model learning, showing the importance of prioritized learning and dissociation.
Notation is as in **Fig. 2c**, but also shows the error of learning the intrinsic behaviorally relevant eigenvalues for an additional learning approach based on numerical optimization with block-structured model parameters (**SI Methods**). Similar to INDM (which is replicated from **Fig. 2c** here), this optimization approach is significantly less accurate for a given training data (i.e., number of samples). Also, it requires orders of magnitude more training samples than IPSID to learn the intrinsic behaviorally relevant eigenvalues as accurately.

Realistic motor task simulations show how sensory inputs to the brain can confound models of neural activity

As explained earlier (**Fig. 1a**), sensory inputs such as task instructions are effectively inputs to the brain that can have different dynamics from task to task, even if the intrinsic neural dynamics remain unchanged. Thus, unless accounted for during modeling, task-specific sensory inputs could confound the learned intrinsic neural dynamics. Developing a method that can learn the correct intrinsic neural

dynamics regardless of the task would allow experimenters to study any given behavioral task of interest or compare intrinsic dynamics across different tasks without worrying about confounding the results and without limiting the task design. We hypothesized that even when the intrinsic neural dynamics remain unchanged, methods that do not consider the task sensory inputs may learn different and incorrect intrinsic dynamics depending on the exact task, whereas IPSID can learn the same intrinsic dynamics regardless of the task. Here we confirm this hypothesis by simulating a brain performing various realistic cursor control motor tasks during which simulated neural data for model training is observed (**Fig. 5**; **SI Methods**).

Specifically, we modeled the brain as an optimal feedback controller[42–44] (OFC), which controls a part of its state that represents the 2D cursor kinematics such that the cursor moves to targets presented via task instructions (**SI Methods**; **Fig. 5a**). The eigenvalues of the state transition matrix for the true simulated brain model are shown in **Fig.5b**. For generality, as part of the simulated brain, we included two latent states (similar to $x_k^{(3)}$ in equation (2)) that are driven by input and affect the movement (i.e. behavior), but are not reflected in the neural dynamics (**SI Methods**). As the first task, we simulated 8 equally spaced targets around a circle and instructed the simulated brain to move the cursor to the targets in order (**Fig. 5c**, left). As the second task, we simulated a standard center-out-and-back task where in each trial the cursor needs to move from the center to a randomly-specified target among 8 targets and then return back to the center (**Fig. 5c**, middle). Lastly, we simulated a 10 by 10 grid of targets where in each trial a random target within a limited distance of the most recent target needs to be visited (**Fig. 5c**, right) similar to the tasks in our NHP datasets (**SI Methods**). For each task, we used (I)PSID and (I)NDM to learn models of neural dynamics (**Fig. 5d**).

We found that regardless of the task, IPSID correctly learned the intrinsic behaviorally relevant neural dynamics. This is evident from comparing the IPSID eigenvalues and flow fields for every task with their ground-truth (first row of **Fig. 5d** vs. **Fig. 5b**). INDM, which considers input but not behavior during training, learned an approximation of some intrinsic behaviorally relevant neural dynamics with error, and also mistakenly included some intrinsic neural dynamics that were not relevant to behavior (**Fig. 5d**, second row). PSID, which considers behavior and neural activity but not input during training, learned

biased intrinsic neural dynamics that were influenced by task instruction inputs (**Fig. 5d**, third row). Finally, NDM, which only considers neural activity during training, not only learned neural dynamics that were not related to behavior, but also learned inaccurate intrinsic behaviorally relevant neural dynamics that were influenced by task instruction inputs (**Fig. 5d**, fourth row). For example, in the first task, the biased dynamics learned by NDM and PSID were very close to the dominant frequency of the task instructions, which was around 0.2Hz (**Fig. 5d**, left column). These results demonstrate that by considering both behavior and sensory inputs such as task instructions during model training, IPSID can learn models of neural dynamics that are not confounded by the specific behavioral task during which neural data is collected. The ability to avoid these confounds is critical for comparing intrinsic neural dynamics across tasks in neuroscience investigations, as we also show in our real NHP neural data analyses below (**Fig. 8**).

We next found that models of neural dynamics learned by IPSID can generalize to other behavioral tasks that are not observed during training unlike methods that do not consider inputs. This is because by considering task instruction inputs, IPSID avoids learning models of neural dynamics that are confounded by the dynamics of instructions in a specific task. Indeed, models trained by IPSID on data from one task had minimal drop in behavior decoding performance when tested on data from a different task. In contrast, models learned by all other methods had significantly larger drops in behavior decoding performance in the other task (**Fig. S10**; $P < 0.001$; one-sided signed-rank; $n = 10$ simulations).

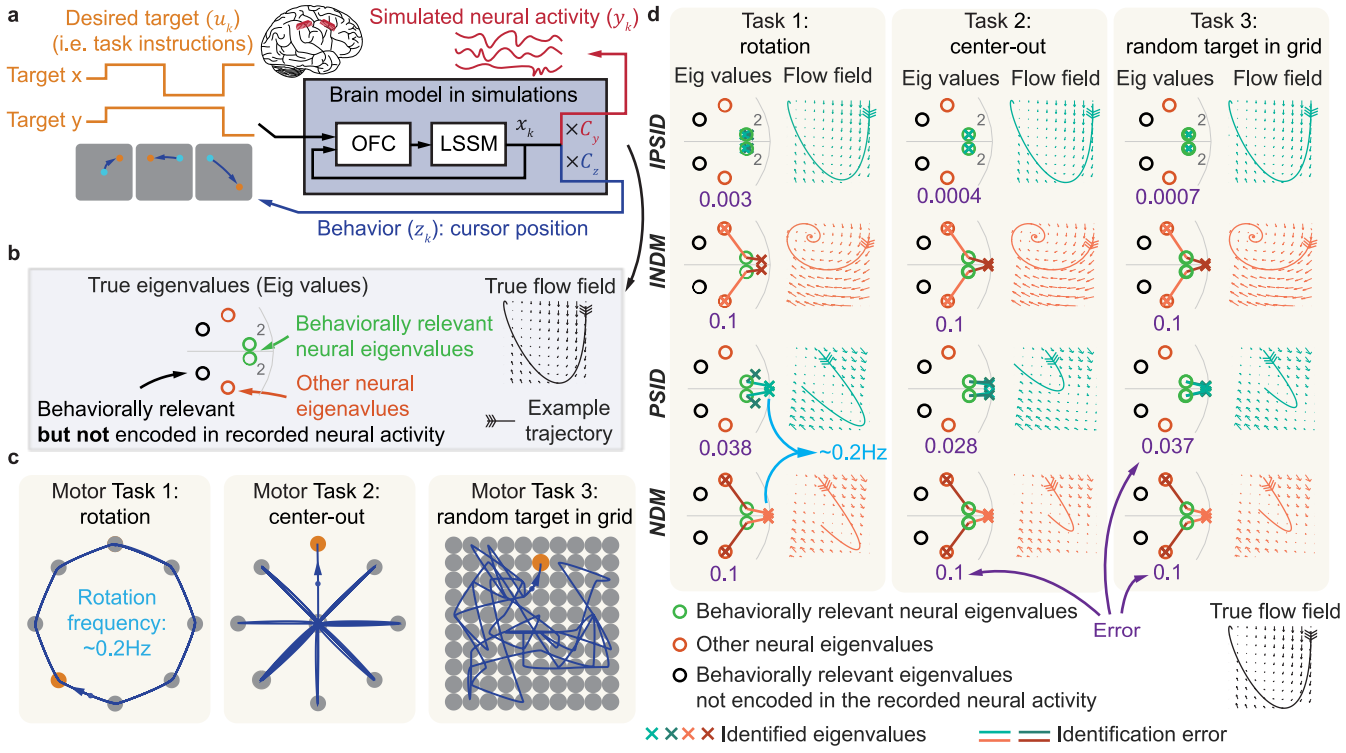

Fig. 5 | By considering task instruction inputs, IPSID learns the correct intrinsic behaviorally relevant neural dynamics regardless of the behavioral task unlike other methods.

(a) The brain model consists of an optimal feedback controller (OFC) combined with a linear state space model (LSSM). Four of the 8 latent state dimensions of the LSSM encode the 2D position and velocity of the cursor (**SI Methods**). OFC controls these 4 state dimensions such that cursor position reaches the target shown on the screen while cursor velocity goes to zero (i.e., cursor stops at target). **(b)** Eigenvalues of the state transition matrix in the full brain model (i.e., OFC together with the LSSM) and the flow field associated with the behaviorally relevant neural eigenvalues. Flow fields show the direction in which the state would change starting from various initial values. In this brain model, there are two sets of behaviorally relevant complex conjugate eigenvalues that are in the same place and thus overlapping. Each set is associated with one movement direction, horizontal and vertical, respectively. The fact that there are two overlapping sets of eigenvalues is indicated by writing a 2 next to these eigenvalues. In addition to the 4 states representing position and velocity in the 2D space, there are 2 states that only drive the neural activity, whose associated eigenvalues are depicted as orange circles. There are also 2 states that only drive the behavior, whose associated eigenvalues are depicted as black circles. **(c)** Tasks performed by the simulated brain. **(d)** Identified eigenvalues for each task using each method with a state dimension of 4. The flow field for one of the two sets of eigenvalues identified by each method (the one with the lighter green/red color) is also shown as an example. Only IPSID correctly learns the intrinsic behaviorally relevant neural eigenvalues regardless of the behavioral task used during training.

Modeling task instructions as inputs reveals distinct intrinsic behaviorally relevant neural dynamics in non-human primate neural population activity

We next used IPSID to study intrinsic behaviorally relevant neural dynamics in two independent motor cortical datasets recorded from three monkeys (monkeys I and L from the first datasets and monkey T from the second dataset) during two distinct behavioral motor tasks with planar cursor movements (**Fig. 6a**, **Fig. 7a**). In the first dataset, which was made publicly available by the Sabes lab[45], primary motor

cortical (M1) population activity was recorded while two monkeys controlled a 2D cursor to reach random targets on a grid (**Fig. 6a**, **SI Methods**). The 3D position of the monkeys' fingertip was tracked and the horizontal elements were passed on to control the cursor (**SI Methods**). In the second dataset, which was made publicly available by the Miller lab[46,47], population activity from dorsal premotor cortex (PMd) was recorded while the monkey performed sequential reaches to random target positions on a plane (**Fig. 7a**, **SI Methods**). The cursor was controlled via a manipulandum that only allowed horizontal movements. For all subjects, we modeled the population spiking activity (**SI Methods**). We took the 2D position and velocity of the cursor as the behavior signal, and the timeseries of target positions as the input task instructions (**Figs. 6a, 7a**). We modeled the smoothed spike counts[3,13,39,48] in all datasets as neural signals (**SI Methods**).

First, we found that IPSID revealed distinct intrinsic behaviorally relevant neural dynamics that were not found by other methods. Similar to our earlier simulation results (**Fig. 5**), this could be seen from the learned eigenvalues by IPSID that were different from those found by other methods (**Figs. 6b, 7b, S11b**). Second, eigenvalues found by PSID were far from those found by IPSID, whereas eigenvalues found by NDM were close to those found by INDM (**Figs. 6b**, **7b, S11b**). Note that IPSID/PSID focus on explaining the behaviorally relevant neural dynamics whereas INDM/NDM focus on explaining the overall neural dynamics regardless of relevance to behavior. Thus, the aforementioned result suggests that task instructions, which are taken as inputs in IPSID/INDM models, are highly informative of behaviorally relevant neural dynamics (seen from their effect on PSID vs IPSID), but are not very informative of the overall neural dynamics (seen from NDM and INDM results being similar). This is consistent with the vast body of work suggesting that neural dynamics relevant to any specific behavior may constitute a minority of the overall neural variance[5,6,19,32–39].

In these analyses we used the additional steps in IPSID that were designed for scenarios in which some input dynamics may affect behavior through unrecorded regions/pathways (**Fig. S5**). However, we found that even without these additional steps, the average learned eigenvalues remained almost unchanged in one subject (**Fig. S12b**) and remained relatively similar in the other two subjects (**Fig. S12a,c**). This result could suggest, particularly in the former (**Fig. S12b**), that behaviorally relevant neural

dynamics that were downstream of visual task instruction inputs were largely reflected in, or downstream of, the motor cortical recordings here. Overall, eigenvalues that were learned by IPSID were unique and were not learned by any of the other three methods. Having established their distinction, the next question was whether these distinct eigenvalues found by IPSID better describe the data, which we explored next.

IPSID finds more accurate intrinsic behaviorally relevant neural dynamics in non-human primate neural population activity

We hypothesized that as in the simulation results (**Fig. 5**), the eigenvalues learned by IPSID are more accurate descriptions of the true intrinsic behaviorally relevant neural dynamics. We performed multiple evaluations to test this hypothesis. As a measure of closeness of two sets of dynamics, we computed the Kullback–Leibler (KL) divergence between the distribution of their associated eigenvalues (**SI Methods**).

First, we explored whether IPSID can mitigate the problem of learning eigenvalues that reflect input dynamics. We characterized the input dynamics by modeling the time series of task instructions as a linear state-space model (equation (3), **SI Methods**). From this model, we found that in all three subjects and in the two tasks, the eigenvalues representing the dynamics of input task instructions were close to those learned using NDM and PSID but not to those learned using IPSID (**Figs. 6b, 7b, S11b**). Indeed, in all subjects, the KL-divergence between the input dynamics and learned dynamics was much larger for IPSID compared with PSID which cannot consider inputs during learning (**Figs. 6c, 7c, S11c**). This result shows the success of IPSID's novel algebraic operations in mitigating the influence of task instruction inputs on intrinsic dynamics unlike NDM and PSID.

Second, we demonstrated the success of preferential neural-behavioral modeling in the presence of input enabled by IPSID by comparing with INDM and NDM. In all three subjects, IPSID learned the intrinsic behaviorally relevant neural dynamics significantly more accurately than both INDM and NDM (**Figs. 6d**, **7d, S11d**). This was evident from comparing the cross-validated behavior decoding from neural activity for these methods (**Figs. 6d**, **7d, S11d**).

Third, we demonstrated the success of IPSID's new algebraic operations in accounting for inputs in preferential neural-behavioral modeling by comparing it to PSID, which is preferential yet does not

consider inputs. We found that by considering inputs, IPSID learns models that were significantly more predictive of neural dynamics compared to PSID in all three subjects, as evident by comparing the cross-validated neural self-prediction accuracy across the two methods (**Figs. 6e, 7e, S11e**). These results held even if the feedthrough term $D_y u_k$ in equation (2) – which reflects the effect of input on neural activity directly and not through the latent states x_k – was discarded when predicting neural activity using IPSID (**Figs. 6e, 7e, S11e**). This analysis demonstrates that the better prediction in IPSID is due to its latent states being more predictive of neural dynamics rather than due to a static feedthrough effect of input on neural dynamics. Overall, these consistent results from three NHPs in two independent neural datasets with two different tasks suggest that IPSID can successfully dissociate intrinsic behaviorally relevant neural dynamics from other intrinsic neural dynamics and from measured input dynamics. Moreover, these results demonstrate that not considering task instruction sensory inputs when modeling neural activity can result in less accurate models of neural dynamics and confound conclusions about intrinsic dynamics, a problem that IPSID addresses (see also next section).

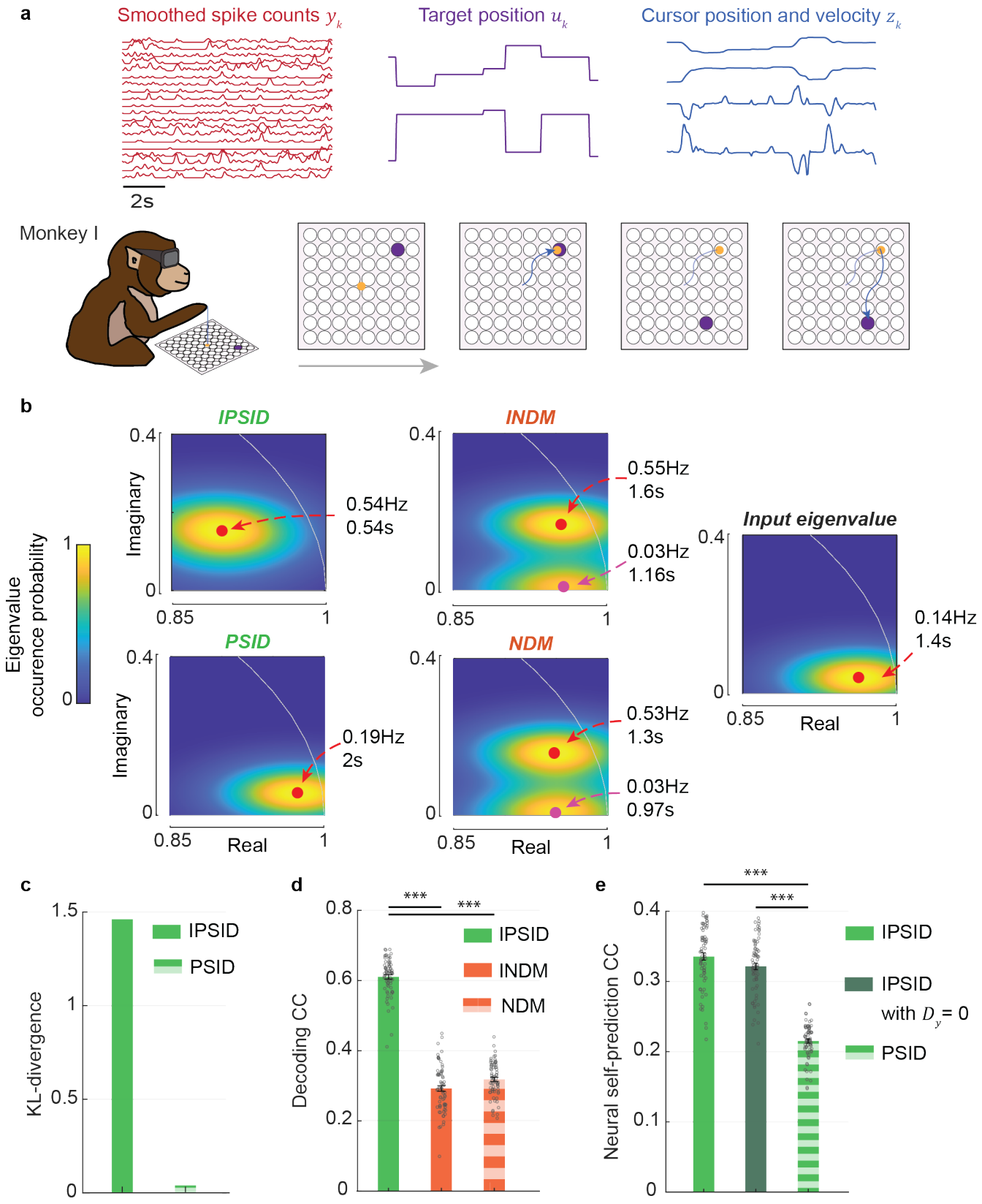

Fig. 6 | IPSID uncovers distinct and more accurate intrinsic behaviorally relevant neural dynamics in motor cortical population activity by considering task instructions as inputs to the brain.

(**a**) We modeled the population spiking activity in a monkey (monkey I) performing a 2D cursor control task (**SI Methods**). See **Fig. S11** for results from a second monkey in this task and **Fig. 7** for results in a second dataset recorded from a different

monkey in a different task. Spike counts are smoothed using a Gaussian kernel with s.d. of 50 ms (**SI Methods**). The 2D position and velocity of the cursor were taken as the behavior signal of interest and the target position time series was taken as the input to the brain. **(b)** Distribution of the eigenvalues of the state transition matrix for models learned using (I)PSID and (I)NDM across datasets. Input eigenvalue was found by applying NDM to the time-series of task instructions. Models were learned with a latent state dimension of $n_x = 4$, which is sufficient for capturing most behavior dynamics (**Fig. S13**). We estimated the probability of an eigenvalue occurring at each location on the complex plane by adding Gaussian kernels centered at locations of all identified eigenvalues (n = 70 cross-validation folds across 2 channel subsets and 7 recording sessions, **SI Methods**). Red dots indicate the location that has the maximum estimated eigenvalue occurrence probability, with the associated frequency and decay rate (**SI Methods**) noted next to each plot. Similarly, when the occurrence probability map has more than one local maximum (i.e., for NDM or INDM), pink dots indicate the location of the second local maximum. **(c)** Quantified by KL-divergence, the eigenvalues learned by PSID were much closer to input eigenvalues than the eigenvalues learned by IPSID, showing the success of the new algebraic operations in accounting for inputs in neural-behavioral modeling. We computed the KL-divergence between the probability mass function of input eigenvalues (panel b, right) and the probability mass function of eigenvalues learned by IPSID/PSID (panel b, top/bottom left). **(d-e)** Cross-validated behavior decoding (panel d) and neural self-prediction (panel e) when modeling data with dimension $n_x = 4$ and corresponding to models in (b). Triple asterisks indicate $P < 0.0005$ for a one-sided signed-rank test.

456

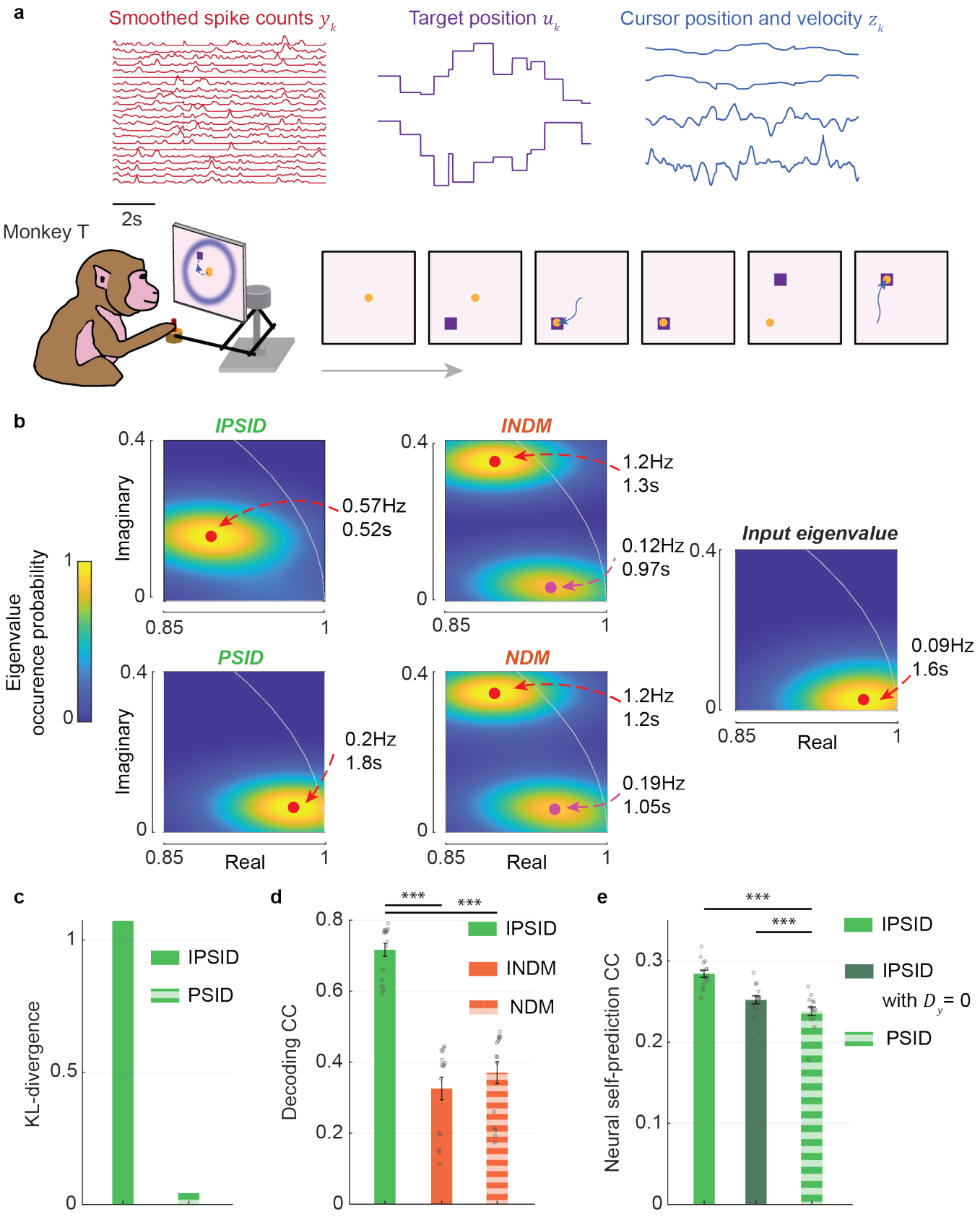

Fig. 7 | In a second dataset recorded from a different monkey and during a different task, IPSID again uncovers distinct and more accurate intrinsic behaviorally relevant neural dynamics in spiking activity by considering task instructions as inputs to the brain.

Similar to **Fig. 6** for the second subject (monkey T, n = 15 cross-validation folds across 3 recording sessions, **SI Methods**) during a different second task with sequential reaches to random targets (**SI Methods**).

IPSID uniquely revealed consistent intrinsic behaviorally relevant dynamics across three different subjects and two different tasks

While the specific task instructions are different in the two behavioral tasks in the independent datasets here – reaches to random targets on a grid vs. sequential reaches to random targets –, the two datasets also have similarities in terms of neural recordings and tasks. Specifically, both independent datasets have recordings from the motor cortical areas, and both involve cursor control tasks with targets on a 2D plane. We thus hypothesized that given these similarities, there may be similarities in the intrinsic behaviorally relevant neural dynamics across the two tasks and three subjects. To test this hypothesis, we compared the distribution of eigenvalues learned using IPSID across all pairs of the three subjects (**Fig. 8**) and quantified their average difference with three metrics: (1) symmetric KL divergence between eigenvalue distributions (**SI Methods**, **Fig. 8d**), (2) correlation coefficient (CC) between the probability mass functions of the eigenvalue distributions (**Fig. 8e**) (3) distance between the modes of the eigenvalue distributions, i.e., most probable locations (**Fig. 8f**). We also repeated these analyses for INDM.

We found that IPSID identified intrinsic behaviorally relevant dynamics that were strikingly similar across the two tasks and three subjects (**Fig. 8**). This result was clear from IPSID's learned eigenvalues both qualitatively (**Fig. 8a-c**) and quantitatively based on the three abovementioned metrics (**Fig. 8d-f**). This similarity was despite the fact that the task instruction sensory inputs were distinct between the two tasks and that these recordings were from three different animals across two independent datasets. Also, even without its additional steps (**Fig. S5**, **Note S2**), IPSID still found largely similar eigenvalues across tasks and monkeys showing the robustness of this result, but the additional steps helped it reveal this similarity slightly more strongly (**Fig. S12d-f**).

We next studied the dynamics found by INDM. INDM aims to learn the overall intrinsic neural dynamics while IPSID aims to prioritize the learning of intrinsic behaviorally relevant neural dynamics. Interestingly, unlike IPSID, the dynamics found by INDM were much more distinct across the three monkeys (**Fig. 8**), as is clear both visually (**Fig. 8a-c**) and quantitatively (**Fig. 8d-f**). Moreover, as shown in the previous section, the more similar dynamics found by IPSID were also a more accurate description of intrinsic

483 behaviorally relevant neural dynamics in each monkey (**Figs. 6d, 7d, S11d**). Together, these results
484 suggest that while the overall intrinsic neural dynamics (as found by INDM) were different across these
485 two planar motor tasks and three animals, the intrinsic behaviorally relevant neural dynamics were similar
486 as revealed by IPSID. We propose that the similarity of the intrinsic behaviorally relevant neural dynamics
487 may suggest that similar neural computations in the motor cortex underlie the planar cursor control tasks
488 despite the differences between task instructions (i.e., inputs) and between animals.

489 IPSID was the only method that revealed the above finding about similar dynamics because it not only
490 accounts for inputs (task instructions), but also prioritizes the learning of intrinsic behaviorally relevant
491 dynamics over other neural dynamics in the presence of input – which is something INDM cannot do.
492 Interestingly, this result is also consistent with our simulation study in **Fig. 5** in which IPSID was the only
493 method that correctly found the fixed intrinsic behaviorally relevant dynamics regardless of task while
494 other methods were confused by the task instructions and/or overall intrinsic dynamics. Thus, IPSID can
495 help researchers explore and compare the intrinsic neural dynamics across different behavioral tasks
496 without worrying about the specific structure of their task instruction inputs and how these inputs may
497 confound their conclusions – e.g., mitigate the confound that similarity or lack thereof in dynamics may
498 simply be due to input comparisons across tasks.

499 Together, these results highlight that the new algebraic operations in IPSID can lead to both more
500 accurate models and new useful scientific insight. These results also demonstrate that even though a
501 comprehensive measurement of all inputs to a given brain region is typically experimentally infeasible,
502 even incorporating partial input measurements (task instruction sensory inputs in this case) can already
503 yield new insights into neural computations across different tasks and subjects.

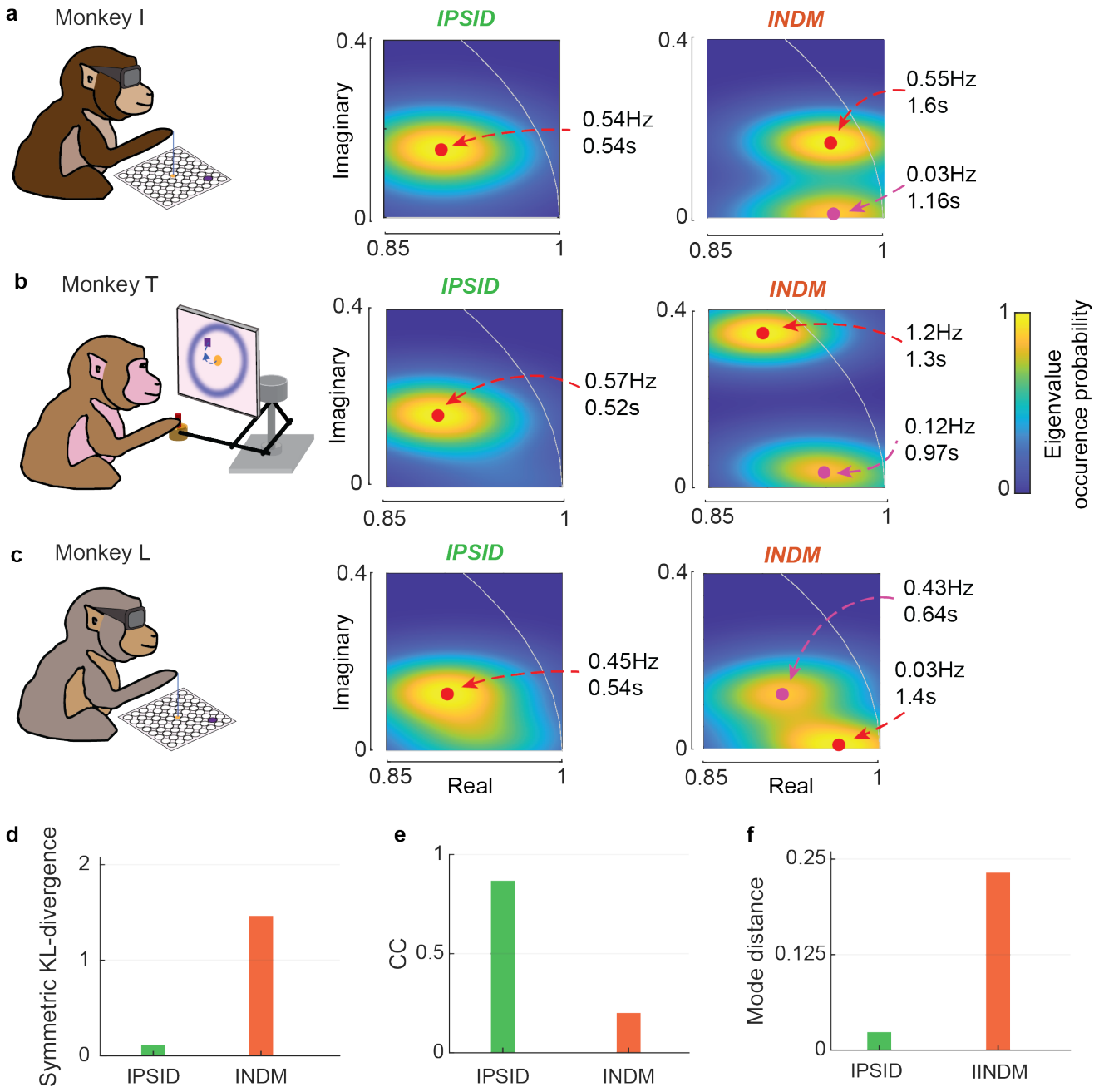

Fig. 8 | IPSID reveals largely similar intrinsic behaviorally relevant neural dynamics across three monkeys and two tasks from two independent datasets while INDM identifies different overall intrinsic neural dynamics.

(a) Same as **Fig. 6b**, showing the eigenvalues learned for IPSID and INDM. **(b-c)** Similar to (a) for the second and third monkeys, respectively (taken from **Fig. 7** and **Fig. S11**). **(d)** Average pairwise symmetric KL-divergence between the eigenvalue probability mass functions of the three monkeys is computed for the IPSID/INDM results. **(e)** Average pairwise Pearson correlation coefficient (CC) of the probability mass functions of the three monkeys used to calculate the symmetric KL-divergence in (d). **(f)** Average pairwise distance between the mode (i.e., most probable eigenvalue location) of the probability mass functions of the three monkeys used to calculate the symmetric KL-divergence in (d). Lower KL-divergence/mode distance implies more similarity across monkeys, with a minimum possible value of 0. Higher CC implies more similarity across monkeys, with a maximum possible value of 1. Based on all three metrics, IPSID finds largely similar eigenvalues across tasks and animals whereas INDM finds eigenvalues that are different across tasks and animals.

Discussion

We developed IPSID, a novel method that provides the new capability to perform preferential dynamical modeling of neural-behavioral data in the presence of measured inputs. In the IPSID formulation, a dynamical model of neural activity is learned by accounting for measured input, neural, and behavioral data simultaneously, and the learning of intrinsic behaviorally relevant neural dynamics is prioritized over other intrinsic dynamics. By doing so, IPSID can dissociate intrinsic behaviorally relevant dynamics not only from other intrinsic dynamics, but also from the dynamics of measured inputs such as task instructions or recorded activity of upstream regions. We demonstrated that without IPSID, dynamics in measured inputs to a given brain region or other intrinsic neural dynamics may be incorrectly identified as intrinsic behaviorally relevant neural dynamics within that brain region and thus confound conclusions. Indeed, in the neural data from monkeys, we showed that task instructions can act as such confounding inputs. IPSID can analytically account for such measured inputs to reveal distinct and more accurate intrinsic behaviorally relevant neural dynamics compared with existing approaches even when they considered input (as in INDM). By doing so, IPSID also provided useful scientific insights about intrinsic neural dynamics of behavior across different tasks and animals, which were not found by other methods.

IPSID could allow future studies to more easily design and compare across tasks without worrying about the temporal structure of task instruction inputs and how their reflection in neural activity may be misinterpreted as intrinsic neural dynamics. We showed this potential with experiments where a simulated brain with fixed intrinsic dynamics performed different cursor control tasks. We showed that sensory inputs in the form of task instructions could lead to learning intrinsic dynamics that incorrectly appeared task-dependent and different across tasks. IPSID addressed this issue and was the only approach that correctly found the intrinsic behaviorally relevant neural dynamics regardless of the task. Consistently, in the real motor cortical datasets and by modeling the task instructions as sensory inputs, IPSID not only learned the intrinsic behaviorally relevant neural dynamics more accurately, but also was the only method that revealed their similarity across tasks and animals.

Unexpectedly, despite differences in animals and in motor tasks and their instructions across the motor cortical datasets, we found similar intrinsic behaviorally relevant dynamics in all three animals across

both tasks/datasets using IPSID. In contrast, INDM found that the dominant overall intrinsic dynamics were different across tasks and animals. This result may suggest that motor cortical regions across different animals could have different intrinsic dynamics overall, but the part of their intrinsic dynamics that is engaged in arm movements to control 2D planar cursors may have similarity. These similar dynamics may suggest that similar intrinsic neural computations in the motor cortex underlie the performance of these two different planar cursor-reaching tasks. Prior work has found similarities in static projections of neural activity[49,50] across subjects[50] or tasks[49], but these prior works have not modeled temporal dynamics (e.g., eigenvalues) and have not disentangled the effect of task instruction input dynamics on the observed similarity. Thus, IPSID provides a new useful tool to explore whether such observed similarities reflect input dynamics or are intrinsic.

When the activity of some upstream brain regions that have inputs to the recorded region[27,31,51–53] is not measured, the learned intrinsic dynamics could also partly originate from these other regions. In the motor cortical datasets here for example, neural dynamics in upstream regions such as visual cortex—which is involved in processing the sensory input and passing it to other regions along the visual-motor pathway—may also be reflected in the learned intrinsic motor cortical dynamics. Taking the sensory instructions as input can, to some extent, account for the dynamics of inputs from these upstream visual areas. Similarly, a sensory input that is not measured or accounted for, for example the sunrise-sunset cycles during chronic recordings, may confound the modeled neural dynamics of a specific behavioral or mental state such as mood (e.g., in the form of circadian rhythms)[54,55]. Thus, recording activity from more upstream regions and measuring more sensory inputs can allow IPSID to analytically consider more comprehensive inputs during modeling to better discover intrinsic behaviorally relevant dynamics. As it is mostly experimentally infeasible to identify and record all inputs to a given brain region, a complete disentanglement of intrinsic dynamics from all input dynamics to a region becomes impractical. This experimental limitation is thus a fundamental limit on methodological efforts aimed at disentanglement. Thus, one still needs to interpret the results cautiously by noting that only dynamics of measured inputs are being disentangled from intrinsic dynamics. Nevertheless, our results show that even this partial disentanglement can lead to more accurate models and to new useful insights compared to alternative

558 models which either do not consider measured inputs, or consider measured input but not behavior during
559 learning.

560 Here, we address the challenge of preferential modeling of neural-behavioral data with measured
561 inputs, which has been unresolved. For non-preferential modeling of neural data on its own and when
562 inputs are not measured, prior studies have looked at the distinct problem of separating the recorded
563 neural dynamics into intrinsic dynamics and a dynamic input that is inferred[12,56,57]. This decomposition is
564 typically done by making certain a-priori assumptions about the input such that inputs can be inferred, for
565 example that input is constrained to be considerably less dynamic than intrinsic neural dynamics, or that
566 input is sparse or spatiotemporally independent[12,56]. In addition to preferential neural-behavioral modeling
567 with measured inputs, which is addressed here, future work can extend preferential modeling to also
568 incorporate similar input inference approaches, which could be complementary to IPSID. For example,
569 such input inference approaches can help further interpret the intrinsic behaviorally relevant dynamics
570 extracted by IPSID and hypothesize which parts of them could be due to unmeasured inputs. The results
571 from such input inference efforts can depend on the a-priori assumptions made regarding the input, since
572 mathematically both extremes are plausible when inputs are not measured: all neural dynamics could be
573 due to input from another area or they could all be intrinsic. For this reason, validating the inferred inputs
574 from these inference approaches against actually measured inputs is an important step[12,53,56,57]. Such
575 validation is also important because the underlying dynamics and inputs can have potential nonlinearities,
576 thus making the inference of unmeasured inputs challenging or infeasible due to the potential
577 unidentifiability in nonlinear systems[58].

578 One main contribution here is to formulate and highlight the problem of how intrinsic neural dynamics
579 underlying a specific behavior can be confounded by both input dynamics and other intrinsic neural
580 dynamics. We formulated this disentanglement problem that simultaneously involves measured input,
581 neural, and behavioral data during learning, and derived IPSID as a new analytical solution based on
582 subspace identification. By comparing with INDM and a block-structured numerical optimization approach
583 (**Figs. 3-4**), we showed that two new capabilities in IPSID are critical for disentanglement: prioritized
584 learning of intrinsic behaviorally relevant dynamics via the new two-stage learning operations with inputs,

585 and dissociating those behavior dynamics that are due to input but not reflected in the neural recordings
586 from those that are via the additional analytical steps (**Fig. S1**, **Fig. S5**). Prior works have proposed
587 enforcing block-structure on linear dynamic models and developed Expectation-Maximization algorithms
588 for fitting them[59,60]. But these studies have distinct goals and thus do not address the input
589 disentanglement problem, or the behaviorally relevant dissociation problem addressed here. As such,
590 they also do not enable the above two new capabilities enabled by IPSID that are critical for solving these
591 problems. Future work can utilize the ideas developed here for enabling the IPSID capabilities in order to
592 develop alternative numerical optimization solutions to the formulated disentanglement problem.

593 In addition to sensory inputs or activity in other brain regions, the input could also be any external
594 electrical or optogenetic brain stimulation, for example in a brain-machine interface (BMI). Developing
595 novel closed-loop stimulation treatments for mental disorders such as depression[61,62] hinges on building
596 dynamic models of neural activity that satisfy two criteria: (i) describe how mental states are encoded in
597 neural activity[61,62]; (ii) describe the effect of electrical stimulation on the neural activity[28,62,63]. The
598 approach developed here enables learning of models that satisfy both criteria. First, by prioritizing
599 behaviorally relevant dynamics, models accurately learn the neural dynamics relevant to behavioral
600 measurements of mental states (e.g. mood reports in depression[61]). Further, this prioritization enables
601 the learned models to have lower-dimensional latent states, which is important in developing robust
602 controllers[64]. Second, the models can explicitly learn the effect of external electrical stimulation
603 parameters on neural activity[28,63].

604 Here we used continuous valued variables with Gaussian distributions to model neural activity, as has
605 been done extensively in prior works modeling local field potentials (LFP)[14,19,30,44,61,65,66] and spike
606 counts[7,19,67,68]. However, recent works suggest that modeling spike counts as Poisson distributed
607 variables[8,12,69–72] can improve BMI performance[70,71]. Thus, an interesting direction is to extend the method
608 to support Poisson distributed neural observations, or support simultaneous Gaussian and Poisson
609 neural observations for multiscale modeling of neural modalities such as LFP and spikes together[16,44,65,73–
610 75]. Supporting general nonlinearities in the intrinsic dynamics and their relation to behavior is another
611 interesting future direction[20,41]. Finally, developing adaptive extensions that update the dynamical latent

state model to adapt to non-stationarities in neural signals or to stimulation-induced plasticity[43,76–79] will be important for BMIs and for studying learning and plasticity and their effect on intrinsic behaviorally relevant dynamics.

In conclusion, we provide a new analytical method for preferential dynamical modeling of neural-behavioral data that can account for measured inputs—whether sensory input, neural input from other regions, or external stimulation. We show the importance of doing so for correct interpretation of neural computations and dynamics that underlie behavior, for accurate modeling of intrinsic neural dynamics, and for gaining useful insights about neural computations of behavior across different tasks and subjects. These results and the new preferential modeling approach have important implications for future neuroscientific and neuroengineering studies.

Acknowledgements

This work was partly supported by NIH R01MH123770 and DP2MH126378 (M.M.S) and USC Annenberg Fellowship (O.G.S). We sincerely thank the Sabes lab at the University of California San Francisco[45] and the Miller lab at Northwestern University[46,47] for making the NHP datasets[45–47] that we used here publicly available.

Author contributions

P.V., O.G.S., and M.M.S developed the algorithms, analyzed data, and wrote the manuscript. M.M.S. supervised the study.

References

1. Buonomano, D. V. & Maass, W. State-dependent computations: spatiotemporal processing in cortical networks. *Nat. Rev. Neurosci.* **10,** 113–125 (2009).
2. 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).
3. Yu, B. M., Cunningham, J. P., Santhanam, G., Ryu, S. I., Shenoy, K. V. & Sahani, M. Gaussian-Process Factor Analysis for Low-Dimensional Single-Trial Analysis of Neural Population Activity. *J. Neurophysiol.* **102,** 614–635 (2009).
4. Macke, J. H., Buesing, L., Cunningham, J. P., Yu, B. M., Shenoy, K. V. & Sahani, M. Empirical models of spiking in neuronal populations. *Adv. Neural Inf. Process. Syst. NIPS* **24,** 1–9 (2011).
5. Churchland, M. M., Cunningham, J. P., Kaufman, M. T., Foster, J. D., Nuyujukian, P., Ryu, S. I. & Shenoy, K. V. Neural population dynamics during reaching. *Nature* **487,** 51–56 (2012).
6. Shenoy, K. V., Sahani, M. & Churchland, M. M. Cortical control of arm movements: a dynamical systems perspective. *Annu. Rev. Neurosci.* **36,** 337–359 (2013).
7. Kao, J. C., Nuyujukian, P., Ryu, S. I., Churchland, M. M., Cunningham, J. P. & Shenoy, K. V. Single-trial dynamics of motor cortex and their applications to brain-machine interfaces. *Nat. Commun.* **6,** 7759 (2015).
8. Aghagolzadeh, M. & Truccolo, W. Inference and Decoding of Motor Cortex Low-Dimensional Dynamics via Latent State-Space Models. *IEEE Trans. Neural Syst. Rehabil. Eng. Publ. IEEE Eng. Med. Biol. Soc.* **24,** 272–282 (2016).
9. Seely, J. S., Kaufman, M. T., Ryu, S. I., Shenoy, K. V., Cunningham, J. P. & Churchland, M. M. Tensor Analysis Reveals Distinct Population Structure that Parallels the Different Computational Roles of Areas M1 and V1. *PLOS Comput. Biol.* **12,** e1005164 (2016).
10. Linderman, S., Johnson, M., Miller, A., Adams, R., Blei, D. & Paninski, L. Bayesian Learning and Inference in Recurrent Switching Linear Dynamical Systems. in *Proc. 20th Int. Conf. Artif. Intell. Stat.* **54,** 914–922 (PMLR, 2017).

11. Wu, A., Roy, N. A., Keeley, S. & Pillow, J. W. Gaussian process based nonlinear latent structure discovery in multivariate spike train data. in *Adv. Neural Inf. Process. Syst.* **30,** (2017).
12. Pandarinath, C., O'Shea, D. J., Collins, J., Jozefowicz, R., Stavisky, S. D., Kao, J. C., Trautmann, E. M., Kaufman, M. T., Ryu, S. I., Hochberg, L. R., Henderson, J. M., Shenoy, K. V., Abbott, L. F. & Sussillo, D. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nat. Methods* **15,** 805–815 (2018).
13. Williams, A. H., Kim, T. H., Wang, F., Vyas, S., Ryu, S. I., Shenoy, K. V., Schnitzer, M., Kolda, T. G. & Ganguli, S. Unsupervised Discovery of Demixed, Low-Dimensional Neural Dynamics across Multiple Timescales through Tensor Component Analysis. *Neuron* **98,** 1099-1115.e8 (2018).
14. 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).
15. Vyas, S., Golub, M. D., Sussillo, D. & Shenoy, K. V. Computation Through Neural Population Dynamics. *Annu. Rev. Neurosci.* **43,** 249–275 (2020).
16. Abbaspourazad, H., Choudhury, M., Wong, Y. T., Pesaran, B. & Shanechi, M. M. Multiscale low-dimensional motor cortical state dynamics predict naturalistic reach-and-grasp behavior. *Nat. Commun.* **12,** 607 (2021).
17. Jazayeri, M. & Ostojic, S. Interpreting neural computations by examining intrinsic and embedding dimensionality of neural activity. *Curr. Opin. Neurobiol.* **70,** 113–120 (2021).
18. Shenoy, K. V. & Kao, J. C. Measurement, manipulation and modeling of brain-wide neural population dynamics. *Nat. Commun.* **12,** 633 (2021).
19. Sani, O. G., Abbaspourazad, H., Wong, Y. T., Pesaran, B. & Shanechi, M. M. Modeling behaviorally relevant neural dynamics enabled by preferential subspace identification. *Nat. Neurosci.* **24,** 140–149 (2021).
20. Sani, O. G., Pesaran, B. & Shanechi, M. M. *Where is all the nonlinearity: flexible nonlinear modeling of behaviorally relevant neural dynamics using recurrent neural networks*. 2021.09.03.458628 (2021). doi:10.1101/2021.09.03.458628
21. Chen, Y., Rosen, B. Q. & Sejnowski, T. J. Dynamical differential covariance recovers directional network structure in multiscale neural systems. *Proc. Natl. Acad. Sci.* **119,** e2117234119 (2022).

685 22. Dubreuil, A., Valente, A., Beiran, M., Mastrogiuseppe, F. & Ostojic, S. The role of population structure
686 in computations through neural dynamics. *Nat. Neurosci.* **25,** 783–794 (2022).
687 23. Michaels, J. A., Dann, B. & Scherberger, H. Neural Population Dynamics during Reaching Are Better
688 Explained by a Dynamical System than Representational Tuning. *PLoS Comput. Biol.* **12,** e1005175
689 (2016).
690 24. Remington, E. D., Egger, S. W., Narain, D., Wang, J. & Jazayeri, M. A Dynamical Systems
691 Perspective on Flexible Motor Timing. *Trends Cogn. Sci.* **22,** 938–952 (2018).
692 25. Elsayed, G. F. & Cunningham, J. P. Structure in neural population recordings: an expected byproduct
693 of simpler phenomena? *Nat. Neurosci.* **20,** 1310–1318 (2017).
694 26. Pandarinath, C., Gilja, V., Blabe, C. H., Nuyujukian, P., Sarma, A. A., Sorice, B. L., Eskandar, E. N.,
695 Hochberg, L. R., Henderson, J. M. & Shenoy, K. V. Neural population dynamics in human motor
696 cortex during movements in people with ALS. *eLife* **4,** e07436 (2015).
697 27. Sauerbrei, B. A., Guo, J.-Z., Cohen, J. D., Mischiati, M., Guo, W., Kabra, M., Verma, N., Mensh, B.,
698 Branson, K. & Hantman, A. W. Cortical pattern generation during dexterous movement is input-driven.
699 *Nature* **577,** 386–391 (2020).
700 28. Yang, Y., Qiao, S., Sani, O. G., Sedillo, J. I., Ferrentino, B., Pesaran, B. & Shanechi, M. M. Modelling
701 and prediction of the dynamic responses of large-scale brain networks during direct electrical
702 stimulation. *Nat. Biomed. Eng.* **5,** 324–345 (2021).
703 29. Ardid, S., Sherfey, J. S., McCarthy, M. M., Hass, J., Pittman-Polletta, B. R. & Kopell, N. Biased
704 competition in the absence of input bias revealed through corticostriatal computation. *Proc. Natl.*
705 *Acad. Sci.* **116,** 8564–8569 (2019).
706 30. Susilaradeya, D., Xu, W., Hall, T. M., Galán, F., Alter, K. & Jackson, A. Extrinsic and intrinsic
707 dynamics in movement intermittency. *eLife* **8,** e40145 (2019).
708 31. Chen, R., Puzerey, P. A., Roeser, A. C., Riccelli, T. E., Podury, A., Maher, K., Farhang, A. R. &
709 Goldberg, J. H. Songbird Ventral Pallidum Sends Diverse Performance Error Signals to
710 Dopaminergic Midbrain. *Neuron* **103,** 266-276.e4 (2019).
711 32. Reimer, J. & Hatsopoulos, N. G. in *Prog. Mot. Control Multidiscip. Perspect.* (ed. Sternad, D.) 243–
712 259 (Springer US, 2009). doi:10.1007/978-0-387-77064-2_12

713 33. Mante, V., Sussillo, D., Shenoy, K. V. & Newsome, W. T. Context-dependent computation by
714 recurrent dynamics in prefrontal cortex. *Nature* **503,** 78–84 (2013).
715 34. Kaufman, M. T., Seely, J. S., Sussillo, D., Ryu, S. I., Shenoy, K. V. & Churchland, M. M. The Largest
716 Response Component in the Motor Cortex Reflects Movement Timing but Not Movement Type.
717 *eNeuro* **3,** ENEURO.0085-16.2016 (2016).
718 35. Ramkumar, P., Dekleva, B., Cooler, S., Miller, L. & Kording, K. Premotor and Motor Cortices Encode
719 Reward. *PLoS ONE* **11,** (2016).
720 36. Svoboda, K. & Li, N. Neural mechanisms of movement planning: motor cortex and beyond. *Curr.*
721 *Opin. Neurobiol.* **49,** 33–41 (2018).
722 37. Allen, W. E., Chen, M. Z., Pichamoorthy, N., Tien, R. H., Pachitariu, M., Luo, L. & Deisseroth, K.
723 Thirst regulates motivated behavior through modulation of brainwide neural population dynamics.
724 *Science* **364,** eaav3932 (2019).
725 38. Stringer, C., Pachitariu, M., Steinmetz, N., Reddy, C. B., Carandini, M. & Harris, K. D. Spontaneous
726 behaviors drive multidimensional, brainwide activity. *Science* **364,** eaav7893 (2019).
727 39. Kobak, D., Brendel, W., Constantinidis, C., Feierstein, C. E., Kepecs, A., Mainen, Z. F., Qi, X.-L.,
728 Romo, R., Uchida, N. & Machens, C. K. Demixed principal component analysis of neural population
729 data. *eLife* **5,** e10989 (2016).
730 40. Van Overschee, P. & De Moor, B. *Subspace Identification for Linear Systems*. (Springer US, 1996).
731 at <http://link.springer.com/10.1007/978-1-4613-0465-4>
732 41. Sani, O. G. Modeling and control of behaviorally relevant brain states. (2020).
733 42. Todorov, E. & Jordan, M. I. Optimal feedback control as a theory of motor coordination. *Nat. Neurosci.*
734 **5,** 1226–1235 (2002).
735 43. Hsieh, H.-L. & Shanechi, M. M. Optimizing the learning rate for adaptive estimation of neural encoding
736 models. *PLOS Comput. Biol.* **14,** e1006168 (2018).
737 44. Hsieh, H.-L., Wong, Y. T., Pesaran, B. & Shanechi, M. M. Multiscale modeling and decoding
738 algorithms for spike-field activity. *J. Neural Eng.* **16,** 016018 (2018).
739 45. O'Doherty, J. E., Cardoso, M. M. B., Makin, J. G. & Sabes, P. N. Nonhuman Primate Reaching with
740 Multichannel Sensorimotor Cortex Electrophysiology. (2017). doi:10.5281/zenodo.583331

741 46. Lawlor, P. N., Perich, M. G., Miller, L. E. & Kording, K. P. Linear-nonlinear-time-warp-poisson models
742 of neural activity. *J. Comput. Neurosci.* **45,** 173–191 (2018).
743 47. Perich, M. G., Lawlor, P. N., Kording, K. P. & Miller, L. E. Extracellular neural recordings from
744 macaque primary and dorsal premotor motor cortex during a sequential reaching task. *CRCNS.org*
745 (2018). doi:https://dx.doi.org/10.6080/K0FT8J72
746 48. Gallego, J. A., Perich, M. G., Chowdhury, R. H., Solla, S. A. & Miller, L. E. Long-term stability of
747 cortical population dynamics underlying consistent behavior. *Nat. Neurosci.* **23,** 260–270 (2020).
748 49. Gallego, J. A., Perich, M. G., Naufel, S. N., Ethier, C., Solla, S. A. & Miller, L. E. Cortical population
749 activity within a preserved neural manifold underlies multiple motor behaviors. *Nat. Commun.* **9,** 1–
750 13 (2018).
751 50. Safaie, M., Chang, J. C., Park, J., Miller, L. E., Dudman, J. T., Perich, M. G. & Gallego, J. A. Preserved
752 neural population dynamics across animals performing similar behaviour. 2022.09.26.509498
753 Preprint at https://doi.org/10.1101/2022.09.26.509498 (2022)
754 51. Mao, T., Kusefoglu, D., Hooks, B. M., Huber, D., Petreanu, L. & Svoboda, K. Long-Range Neuronal
755 Circuits Underlying the Interaction between Sensory and Motor Cortex. *Neuron* **72,** 111–123 (2011).
756 52. Nashef, A., Mitelman, R., Harel, R., Joshua, M. & Prut, Y. Area-specific thalamocortical
757 synchronization underlies the transition from motor planning to execution. *Proc. Natl. Acad. Sci.* **118,**
758 e2012658118 (2021).
759 53. Kalidindi, H. T., Cross, K. P., Lillicrap, T. P., Omrani, M., Falotico, E., Sabes, P. N. & Scott, S. H.
760 Rotational dynamics in motor cortex are consistent with a feedback controller. *eLife* **10,** e67256
761 (2021).
762 54. Vadnie, C. A. & McClung, C. A. Circadian Rhythm Disturbances in Mood Disorders: Insights into the
763 Role of the Suprachiasmatic Nucleus. *Neural Plast.* **2017,** 1504507 (2017).
764 55. Logan, R. W. & McClung, C. A. Rhythms of life: circadian disruption and brain disorders across the
765 lifespan. *Nat. Rev. Neurosci.* **20,** 49–65 (2019).
766 56. Schimel, M., Kao, T.-C., Jensen, K. T. & Hennequin, G. iLQR-VAE : control-based learning of input-
767 driven dynamics with applications to neural data. 2021.10.07.463540 Preprint at
768 https://doi.org/10.1101/2021.10.07.463540 (2021)

769 57. Keshtkaran, M. R., Sedler, A. R., Chowdhury, R. H., Tandon, R., Basrai, D., Nguyen, S. L., Sohn, H.,
770 Jazayeri, M., Miller, L. E. & Pandarinath, C. A large-scale neural network training framework for
771 generalized estimation of single-trial population dynamics. 2021.01.13.426570 Preprint at
772 https://doi.org/10.1101/2021.01.13.426570 (2022)
773 58. Grewal, M. & Glover, K. Identifiability of linear and nonlinear dynamical systems. *IEEE Trans. Autom.*
774 *Control* **21,** 833–837 (1976).
775 59. Glaser, J., Whiteway, M., Cunningham, J. P., Paninski, L. & Linderman, S. Recurrent Switching
776 Dynamical Systems Models for Multiple Interacting Neural Populations. in *Adv. Neural Inf. Process.*
777 *Syst.* **33,** 14867–14878 (Curran Associates, Inc., 2020).
778 60. Semedo, J., Zandvakili, A., Kohn, A., Machens, C. K. & Yu, B. M. Extracting Latent Structure From
779 Multiple Interacting Neural Populations. in *Adv. Neural Inf. Process. Syst.* **27,** (Curran Associates,
780 Inc., 2014).
781 61. Sani, O. G., Yang, Y., Lee, M. B., Dawes, H. E., Chang, E. F. & Shanechi, M. M. Mood variations
782 decoded from multi-site intracranial human brain activity. *Nat. Biotechnol.* **36,** 954 (2018).
783 62. Shanechi, M. M. Brain–machine interfaces from motor to mood. *Nat. Neurosci.* **22,** 1554–1564
784 (2019).
785 63. Yang, Y., Connolly, A. T. & Shanechi, M. M. A control-theoretic system identification framework and
786 a real-time closed-loop clinical simulation testbed for electrical brain stimulation. *J. Neural Eng.* **15,**
787 066007 (2018).
788 64. Obinata, G. & Anderson, B. D. O. *Model Reduction for Control System Design*. (Springer Science &
789 Business Media, 2012).
790 65. Abbaspourazad, H., Hsieh, H. & Shanechi, M. M. A Multiscale Dynamical Modeling and Identification
791 Framework for Spike-Field Activity. *IEEE Trans. Neural Syst. Rehabil. Eng.* **27,** 1128–1138 (2019).
792 66. Stavisky, S. D., Kao, J. C., Nuyujukian, P., Ryu, S. I. & Shenoy, K. V. A high performing brain-machine
793 interface driven by low-frequency local field potentials alone and together with spikes. *J. Neural Eng.*
794 **12,** 036009 (2015).
795 67. Shanechi, M. M. Brain–Machine Interface Control Algorithms. *IEEE Trans. Neural Syst. Rehabil. Eng.*
796 **25,** 1725–1734 (2017).

68. 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).

69. Smith, A. C. & Brown, E. N. Estimating a State-space Model from Point Process Observations. *Neural Comput* **15,** 965–991 (2003).

70. 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,** 1–29 (2016).

71. Shanechi, M. M., Orsborn, A. L., Moorman, H. G., Gowda, S., Dangi, S. & Carmena, J. M. Rapid control and feedback rates enhance neuroprosthetic control. *Nat. Commun.* **8,** 13825 (2017).

72. 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).

73. 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).

74. 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).

75. Wang, C., Pesaran, B. & Shanechi, M. M. Modeling multiscale causal interactions between spiking and field potential signals during behavior. *J. Neural Eng.* **19,** 026001 (2022).

76. Shenoy, K. V. & Carmena, J. M. Combining decoder design and neural adaptation in brain-machine interfaces. *Neuron* **84,** 665–680 (2014).

77. Ahmadipour, P., Yang, Y., Chang, E. F. & Shanechi, M. M. Adaptive tracking of human ECoG network dynamics. *J. Neural Eng.* **18,** 016011 (2020).

78. Yang, Y., Ahmadipour, P. & Shanechi, M. M. Adaptive latent state modeling of brain network dynamics with real-time learning rate optimization. *J. Neural Eng.* **18,** 036013 (2020).

79. Yang, Y., Lee, J. T., Guidera, J. A., Vlasov, K. Y., Pei, J., Brown, E. N., Solt, K. & Shanechi, M. M. Developing a personalized closed-loop controller of medically-induced coma in a rodent model. *J. Neural Eng.* **16,** 036022 (2019).

Supplementary Information for:

Modeling and dissociation of intrinsic and input-driven neural population dynamics underlying behavior

Parsa Vahidi[1,†], Omid G. Sani[1,†], Maryam M. Shanechi[1,2,*]

1 Department of Electrical and Computer Engineering, University of Southern California

2 Neuroscience Graduate Program, Department of Computer Science, and Department of Biomedical Engineering, University of Southern California

† Equal contribution.

* Corresponding author: shanechi@usc.edu

Supplementary Information Includes:

- Supplementary Methods (**SI Methods**)
- Supplementary Notes (**Notes S1-2**)
- Supplementary Figures (**Figs. S1-12**)
- Supplementary References

Supplementary Information Methods (SI Methods)

Non-preferential neural dynamic modeling with and without input

Neural population activity exhibits rich temporal structures[1–22,26,23,25,24]. Neural dynamical modeling aims to describe all such temporal structures in neural activity[2–5,7,8,10–14,16,21,23,24,26,61,65] without prioritizing the learning of dynamics related to any particular behavior, which is why we also refer to it as non-preferential modeling. In this section we provide a brief overview of linear neural dynamical modeling with and without consideration of inputs, referred to respectively as INDM and NDM[2,7,8,14,16,19,61,65]. In NDM, neural population activity is modeled in terms of a latent state as

$$\begin{cases} x_{k+1} = A\ x_k + w_k \\ y_k = C_y\ x_k + v_k \end{cases} \tag{3}$$

where as before, $y_k \in \mathbb{R}^{n_y}$ and $x_k \in \mathbb{R}^{n_x}$ represent the neural activity and the latent state of the neural population, respectively. $v_k \in \mathbb{R}^{n_y}$ represents noises in neural activity and $w_k \in \mathbb{R}^{n_x}$ represents excitations that drive the latent state, with their covariances defined as in equation (5) below. Given that in the NDM formulation inputs are not explicitly modeled, the excitations of the latent state represented by w_k will capture a mixture of both the measured inputs and all other excitations that originate within the recorded brain region or from other brain regions. However, since w_k is modeled as noise and not a dynamical system itself, the NDM model has to capture any temporal structure in inputs via the dynamics of the latent state x_k, which is quantified through the state transition matrix A. Thus, when inputs are temporally structured, the state dynamics—which are subsequently reflected in the neural activity y_k—may incorrectly also incorporate the structured dynamics that exist in inputs, such as sensory inputs in the form of movement targets visually presented on a screen[1,9,12,27,30] (we also show this in our results). As overviewed next, when inputs are measured, although not commonly done, INDM methods can be used to incorporate them in non-preferential modeling.

In INDM, measured inputs are incorporated into equation (3) as

$$\begin{cases} x_{k+1} = A\ x_k + B\ u_k + w_k \\ y_k = C_y\ x_k + D_y u_k + v_k \end{cases} \tag{4}$$

where $u_k \in \mathbb{R}^{n_u}$ is the measured input to the neural population, e.g. inputs from other brain regions or sensory inputs. Unlike equation (3), in equation (4), the term u_k explicitly represents measured inputs to the recorded brain region and thus the state dynamics reflected in A no longer need to describe the dynamics of u_k; rather, state dynamics solely need to describe the intrinsic dynamics of the brain region in response to measured (u_k) and unmeasured excitations (w_k). Thus, this approach can dissociate the input dynamics form the intrinsic dynamics. While not commonly used in neuroscience, INDM has great utility in system identification[40,80]. However, INDM does not allow for preferential modeling of shared dynamics between two signals, such as neural activity and behavior. Here we address the unresolved challenge of modeling the effect of input within preferential modeling of two signals together (e.g., neural population activity and behavior).

Enabling the modeling of input effects in preferential dynamical modeling

Model formulation

We develop a new method termed IPSID to enable dissociating the effect of input in preferential dynamical modeling of neural-behavioral data, which has not been possible to date. Unlike non-preferential dynamical modeling—e.g., NDM/INDM and other approaches[2–4,7,8,10–14,16]—, which models the dynamics of a single signal such as neural population activity, preferential modeling dissociates the shared dynamics between two signals, such as neural population activity and behavior, and prioritizes the learning of these shared dynamics[19].

So far, input effects cannot be considered in preferential dynamical modeling, even when input is fully measured. Here we develop a preferential dynamical method that can achieve this goal and dissociate input dynamics and intrinsic dynamics. Specifically, the method simultaneously achieves two goals: It dissociates the intrinsic dynamics that originate in the recorded region from input dynamics that are simply reflected in the recorded regions but do not originate there (e.g., input from upstream regions or sensory feedback). Second, it dissociates and prioritizes the learning of intrinsic neural dynamics that are relevant to the behavior of interest from other intrinsic neural dynamics.

This new method jointly models neural activity $y_k \in \mathbb{R}^{n_y}$, behavior $z_k \in \mathbb{R}^{n_z}$, and the effect of input $u_k \in \mathbb{R}^{n_u}$ on them with the general linear state-space formulation in equation (1). In equation (1), $w_k \in \mathbb{R}^{n_x}$ and $v_k \in \mathbb{R}^{n_y}$ are taken to be zero-mean white noises that are independent of x_k, i.e. $\boldsymbol{E}\{x_k w_k^T\} = 0$ and $\boldsymbol{E}\{x_k v_k^T\} = 0$ with the following cross-correlations:

$$\boldsymbol{E}\left\{\begin{bmatrix} w_k \\ v_k \end{bmatrix}\begin{bmatrix} w_k \\ v_k \end{bmatrix}^T\right\} \triangleq \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix}. \tag{5}$$

The latent state x_k describes all intrinsic neural dynamics including those related to the given behavior and those unrelated to it. It can be shown[19,80] that equation (1) can always be written in an equivalent basis as

$$\begin{cases} \begin{bmatrix} x_{k+1}^{(1)} \\ x_{k+1}^{(2)} \end{bmatrix} = \begin{bmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{bmatrix}\begin{bmatrix} x_k^{(1)} \\ x_k^{(2)} \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u_k + \begin{bmatrix} w_k^{(1)} \\ w_k^{(2)} \end{bmatrix} \\ y_k = \begin{bmatrix} C_{y_1} & C_{y_2} \end{bmatrix}\begin{bmatrix} x_k^{(1)} \\ x_k^{(2)} \end{bmatrix} + D_y u_k + v_k \quad , \qquad x_k = \begin{bmatrix} x_k^{(1)} \\ x_k^{(2)} \end{bmatrix} \\ z_k = \begin{bmatrix} C_{z_1} & 0 \end{bmatrix}\begin{bmatrix} x_k^{(1)} \\ x_k^{(2)} \end{bmatrix} + D_z u_k + \epsilon_k \end{cases} \tag{6}$$

where $x_k^{(1)} \in \mathbb{R}^{n_1}$ and $x_k^{(2)} \in \mathbb{R}^{n_x - n_1}$ denote the behaviorally relevant and the other dimensions of x_k, respectively. In IPSID, we directly learn the model in the basis shown in equation (6) in a way that prioritizes behaviorally relevant states. n_1 is the smallest latent state dimension that is sufficient for explaining all the dynamics of behavior z_k that are encoded in neural activity y_k. The parameter set $(A, B, C_y, C_z, D_y, D_z, Q, R, S)$ fully specifies the model in equation (1) and equivalently in equation (6), except for dynamics of ϵ_k, which represent behavior dynamics that are not encoded in neural activity. As explained in **Note S2**, when desired (e.g., see **Fig. S8a**), dynamics of ϵ_k can be modeled separately by modeling the residual behavior after equation (1) is used to predict behavior[19] (**Fig. S5c**).

Learning model parameters using IPSID

In the learning problem, given neural, behavior, and input time-series – denoted by $\{y_k : 0 \le k < N\}$, $\{z_k : 0 \le k < N\}$, and $\{u_k : 0 \le k < N\}$, respectively –, and given the desired dimension of the latent state n_x, and the desired number of behaviorally relevant latent states n_1, the aim is to learn all model parameters in equation (6) while prioritizing learning of behaviorally relevant states. To do this, during

training, we first extract the intrinsic latent states directly using the neural activity, behavior and input training data; we then identify the model parameters using the extracted intrinsic latent states. The details are provided in **Note S1**. Here, we briefly explain the algorithm and the intuition behind it.

IPSID extracts the latent states from training data in two stages (**Fig. S1**). In stage 1, the subspace spanned by the behaviorally relevant latent states ($x_k^{(1)}$) is extracted with *priority* by projecting future behavior (Z_f) onto past neural activity (Y_p). However, when there exist measured inputs, the input dynamics also affect future behavior. Thus, an orthogonal projection of future behavior onto past neural activity (as is used in our prior work on PSID[19,41]) would give a mixture of the subspaces spanned by the intrinsic behaviorally relevant latent states and the input. This means that the learned states will not just reflect the intrinsic behaviorally relevant neural dynamics as these intrinsic dynamics cannot be dissociated from input dynamics. To extract the subspace associated with intrinsic behaviorally relevant latent states, we have to devise distinct algebraic operations in IPSID.

We thus devise these algebraic operations to perform an oblique (i.e., non-orthogonal) projection of future behavior onto past neural activity and past input along the subspace spanned by the future input (**Fig. S1**, **Fig. S6c**). Note that the use of oblique projections in IPSID instead of orthogonal projections is not as simple as replacing one operation with another; rather, this key change has broad consequences throughout the rest of the model learning operations that are appropriately accounted for in IPSID. For example, the learning of input parameter B (see equation (1)) has no equivalence in prior works that do not consider input (e.g., PSID) and requires distinct operations. The oblique projection ensures that the result of the projection is orthogonal to the subspace spanned by the future input, thus excluding behavior dynamics that can be directly attributed to future input dynamics rather than intrinsic neural dynamics (**Note S1**).

In an optional stage 2, we devise additional algebraic operations to extract the subspace spanned by any remaining latent states ($x_k^{(2)}$) via an oblique projection from the residual future neural activity—the part unexplained by the extracted behaviorally relevant states ($x_k^{(1)}$)—onto past neural activity and past input, again along the subspace spanned by the future input (**Fig. S1**). Finally, given the fully extracted

subspace of the latent states (either only from stage 1 or by concatenating the results from both stages), we learn all model parameters based on equation (1) via least squares as detailed in **Note S1**.

Special cases of IPSID

IPSID addresses two challenges simultaneously by allowing for input incorporation in preferential dynamical modeling of neural-behavioral data together. First, it dissociates intrinsic and measured input dynamics. Second, it dissociates intrinsic behaviorally relevant dynamics from other intrinsic neural dynamics and prioritizes the learning of the former. Special cases of IPSID cover prior methods developed by us or others, which only address one of the challenges that IPSID addresses. Briefly, when not modeling input (equivalently when assuming B, D_y, D_z are zero in equation (1)), IPSID reduces to PSID in our prior work[19]. In this case, input dynamics may be learned as part of the intrinsic dynamics in the model and thus the first challenge of dissociating the dynamics of measured input and intrinsic dynamics will not be addressed. Alternatively, when behavior is not considered during modeling, i.e., when $n_1 = 0$ such that all latent states are extracted using the second stage of IPSID, IPSID reduces to prior neural dynamical modeling with input (i.e., INDM)[40,80], which is formulated by equation (4). In this case, intrinsic behaviorally relevant dynamics are not dissociated or prioritized compared with other intrinsic neural dynamics, and thus the second challenge will not be addressed. Finally, if inputs are not considered and only the second stage is used, IPSID reduces to prior neural dynamical modeling (NDM) formulated by equation (3), which does not address either of the two challenges addressed by IPSID.

Learning using numerical optimization with block-structured parameters

To compare with IPSID and show the benefits of its two-stage learning method in prioritized learning of intrinsic behaviorally relevant dynamics, we also implement an alternative approach for fitting the same model using standard numerical optimization. In this approach we use numerical optimization[81] to learn all model parameters by maximizing the neural-behavioral data log-likelihood while imposing the same block-structure as is defined in equation (6). To do so, we use the recurrent neural network class[†] in TensorFlow[82] v2.5 to implement a linear recurrent neural network with the computation graph

[†] tf.keras.layers.RNN

corresponding to equation (6). This approach uses gradient descent via error backpropagation to fit the parameters of the recurrent neural network. Parameters are learned while maximizing the log likelihood of full output data, i.e., both neural activity and behavior $[y_k, z_k]$. We use minibatch size of 32 and run the numerical optimization on the same CPUs as was done for IPSID to enable a fair comparison in terms of the training time (**Fig. S9**). While the sequential nature of recurrent neural networks limits parallelization options during training[83], it is possible that other implementations (e.g., using a framework other than TensorFlow[82]) or computational tricks[84] may have different results and may lead to faster learning. Note that while this numerical optimization enforces the same block-structure, it cannot prioritize the learning of intrinsic behaviorally relevant dynamics and does not dissociate behavior dynamics not reflected in the neural recordings. As we show in results, both these new capabilities are critical for disentanglement.

Additional steps to add IPSID support for scenarios where neural recordings do not cover/reflect all downstream regions of the input

We also add additional optional steps to IPSID to support scenarios where some downstream regions of input that affect behavior are not covered by the recorded neural activity. In these scenarios, which we formulate as in equation (2), input may affect behavior through paths that are not reflected in the recorded neural activity. We thus add a new type of latent state in equation (2), denoted by $x_k^{(3)}$, to describe such paths as latent states that are affected by input and can affect behavior but do not contribute to generation of neural activity, i.e., columns corresponding to these states are zero in C_y and the noises driving them (i.e., $w_k^{(3)}$) are uncorrelated with the noises that drive the other latent states ($w_k^{(1)}$ and $w_k^{(2)}$). To dissociate these states from other states in IPSID, we perform two additional steps (**Note S2**, **Fig. S5**).

First, before performing the two-stage IPSID learning process described earlier (**Note S1**), we use the IPSID second stage alone to build a model for all intrinsic neural dynamics in terms of a high-dimensional latent state, denoted by $x_k^{(y)}$ in **Note S2** and **Fig. S5**. We next project the behavior onto the extracted latent state $x_k^{(y)}$ (with the result denoted by Z_f' in **Note S2** and **Fig. S5**); doing this projection removes any elements of behavior that are not encoded in neural activity (**Fig. S5a**). We then proceed with IPSID stages 1 and 2 as before but with the projected behavior signal (i.e., Z_f') used in the modeling rather than

the original behavior signal (**Fig. S5b, Fig. S6a**); using this projected behavior signal ensures that behavior dynamics that are not encoded in the neural recordings are not included in the first set of states $x_k^{(1)}$.

Second, if learning of behavior dynamics that are predictable from input but are not reflected in neural activity is of interest, we perform an additional step to learn a model for an additional latent state $x_k^{(3)}$ to describe such dynamics. To do this, we first subtract from behavior its prediction from the past neural activity and past input using the already learned model that consists of $x_k^{(1)}$ and $x_k^{(2)}$. This gives us a residual behavior signal. We then apply the IPSID second stage alone to this residual behavior to model its dynamics and the effect of input on it in terms of a latent state $x_k^{(3)}$, with a formulation akin to equation (4) but with z_k as output of the second line. This gives $x_k^{(3)}$ because $x_k^{(3)}$ summarizes the direct effect of input on behavior dynamics that are not reflected in the recorded neural activity. We then put the model learned for $x_k^{(1)}$ and $x_k^{(2)}$ together with the model learned for $x_k^{(3)}$ to get a full model in the form of equation (2). Note that the model for $x_k^{(3)}$ has no neural observation. Thus, in the final overall model, the columns of C_y corresponding to $x_k^{(3)}$ will be zero. Consequently, neural activity y_k makes no contribution to the estimation of $x_k^{(3)}$ (see equation (7) below) in the overall model, resulting in the estimated $x_k^{(3)}$ being forward predicted purely using input u_k. This concludes the learning of the full model as formulated in equation (2). The details are provided in **Note S2**.

Extracting latent states and predicting neural activity and behavior using the learned model

Given the model in equation (1), the prediction of behavior and neural activity given past neural activity and inputs are obtained using the well-known recursive Kalman filter[40,80]. Thus, once the model is learned, in test data, we extract the latent states by applying the Kalman filter associated with the identified model parameters to the neural activity and input as

$$\hat{x}_{k+1|k} = A\hat{x}_{k|k-1} + Bu_k + K\big(y_k - C_y\hat{x}_{k|k-1} - D_y u_k\big). \quad (7)$$

Here $\hat{x}_{k|k-1}$ denotes the latent state estimated at time step k using neural activity and input up to time step $k-1$ (y_n and u_n for $0 \le n < k$), with $\hat{x}_{0|-1} = 0$ taken as the initial state. K denotes the steady state

Kalman gain[40,80], which can be computed as $K = (APC^T + S)(CPC^T + R)^{-1}$ where P is the solution to the following steady-state Riccati equation: $P = APA^T + Q - (APC^T + S)(CPC^T + R)^{-1}(APC^T + S)^T$. Note that behavior is not looked at when extracting the latent states using the Kalman filter (it is only looked at during model training in the training set). Given the estimated latent states $\hat{x}_{k|k-1}$, we next compute the self-prediction of neural activity $\hat{y}_{k|k-1}$ and the decoding of behavior $\hat{z}_{k|k-1}$, both given the past neural activity (y_n for $0 \leq n < k$) and the past and current inputs (u_n for $0 \leq n \leq k$) as

$$\begin{cases} \hat{y}_{k|k-1} = C_y \hat{x}_{k|k-1} + D_y u_k \\ \hat{z}_{k|k-1} = C_z \hat{x}_{k|k-1} + D_z u_k \end{cases}. \quad (8)$$

Performance measures: behavior decoding and neural self-prediction

To evaluate the leaned models and as a measure of how well the behaviorally relevant neural dynamics are learned, we compute the accuracy of decoding behavior (equation (8)). We perform modeling within a 5-fold cross-validation. As the performance measure, we compute the correlation coefficient (CC) between the decoded and true behavior time-series in the test data, averaged across the data dimensions, as the performance measure. Similarly, to evaluate how well the neural dynamics are learned in general, irrespective of their relevance to behavior, we compute the accuracy (in terms of CC) of predicting neural activity one-step-ahead using past neural activity and past and current inputs (equation (8)), which we refer to as neural self-prediction.

General simulations with random models

To validate IPSID with numerical simulations, we generate random models and confirm that IPSID can correctly learn these models when provided with training data. We generate the random models as follows. First, we select the dimensions of y_k, z_k and u_k with uniform probability from ranges $5 \leq n_y, n_z \leq 10$ and $1 \leq n_u \leq 4$. We select the full latent state dimension n_x uniformly from $1 \leq n_x \leq 10$ and then we select the dimension of the state that derives behavior, i.e., n_1, uniformly from $1 \leq n_1 \leq n_x$. We generate the state transition matrix A based on its eigenvalues, which we randomly generate in complex conjugate pairs drawn with uniform probability across the unit disk. We randomly select a subset of n_1 complex-conjugate eigenvalues as the behaviorally relevant eigenvalues to be placed in the top-left $n_1 \times n_1$

submatrix of A (i.e., A_{11} in equation (6)). We generate matrices B, C_y, and C_z randomly with standard normal distribution for each of their elements, and then we set the right block of C_z (after the first n_1 columns) to zero as in equation (6)). For our simulation analyses in **Fig. 2a-b**, **Figs. S2-S4**, both D_y and D_z are randomly generated non-zero matrices. In all other figures, we take $D_z = 0$, and in **Figs. 1,3,5,6**, we additionally take $D_y = 0$. Finally, we generate a random positive-semi-definite square matrix with $n_x + n_y$ rows as the noise covariances Q, R, and S per equation (5). We then select two random numbers between 0.1 to 10 (with uniform probability in log scale) and scale the state and observation noises with these numbers to provide a wide range of relative state and observation noise values; we reflect that scaling in covariances Q, R, and S.

With a similar approach, we generate two other random models but this time without input: one as the behavior noise model to generate the behavior dynamics not present in neural activity, i.e. ϵ_k; and, one as the input model to generate the input u_k. The output dimension in these models is selected consistent with the behavior and input dimensions in the main model, respectively. We select the dimension of the latent state in the behavior noise model uniformly in $1 \leq n_{x_\epsilon} \leq 10$ and that of the input model uniformly from $1 \leq n_{x_u} \leq 4$. Finally, to cover a diverse range of signal to noise ratios (i.e., signal $C_z x_k$ over noise ϵ_k) for behavior, we select a random number between 1 and 100 (with uniform probability in log scale) and scale rows of C_z such that the ratio of the signal s.d. to noise s.d. for each behavior dimension becomes the selected random number. Note that the eigenvalues of the state transition matrix A_u in the input model are representative of the input dynamics and may incorrectly be learned as intrinsic dynamics by methods that do not consider input (NDM/PSID) (**Fig. 2** and **Fig. S3**). To simulate scenarios in which input affects behavior through pathways that are not reflected in the recorded neural activity (**Fig. 3, Figs. S7-S8**), we add an input term $B'u_k$ to the state equation of the model that generates ϵ_k with a B' parameter that is non-zero only in a subset of rows; this way, the input affects a random subset of dimensions of the states that generate ϵ_k, effectively changing their role to that of $x_k^{(3)}$ in equation (2).

Given the model parameters, a time series realization with N data points can be generated with the following procedure. First, the input time series u_k is generated by drawing N Gaussian noise samples

with the covariances given in the input model (similar to Q, R, and S in equation (5)) and iterating through the state-space equation (similar to equation (3), but with y_k renamed to u_k). This gives the N-sample input time-series $\{u_k: 0 \leq k < N\}$. Similarly, a realization from the behavior noise model is generated as the N-sample time-series $\{\epsilon_k: 0 \leq k < N\}$. Next, N Gaussian noise samples are randomly generated for noise covariances in the main model (Q, R, and S in equation (5)). These noise samples along with u_k and ϵ_k are used to iterate through equation (1) and produce the behavior and neural activity time series $\{y_k, z_k: 0 \leq k < N\}$. In all cases, the initial state in the state-space model iterations is taken to be $x_{-1} = 0$. Given the generated neural activity, behavior and input time series, we fit a model using (I)PSID and (I)NDM algorithms with horizon parameter of $i = 5$ (**Note S1**).

Performance measures for learning of model parameters and eigenvalues in numerical simulations

The model in equation (1) can be written with infinitely many different but equivalent sets of parameters that all give rise to the exact same statistics for neural and behavior observations y_k and z_k. Thus, to evaluate the parameter learning performance, we need to consider all equivalent sets of parameters for the identified model. An equivalent model to a given model can be obtained by a change of the latent state basis (also known as a similarity transform), which can be obtained by multiplying the latent state with an invertible matrix. Thus, to compare the identified and true models, we first solve an optimization problem to find the similarity transform that makes the basis of the identified model as similar as possible to the true model, and then compute the difference between the identified and true model parameters. Just to find this similarity transform, we generate a new realization with $q = 1000n_x$ samples from the true model, and then extract latent state $\hat{x}_{k|k-1}^{(true)}$ and $\hat{x}_{k|k-1}^{(id)}$ using the steady-state Kalman filter associated with the true and identified models (equation (7)), respectively. We then find the optimal similarity transform that minimizes the mean squared error between the two sets of latent states as

$$\hat{T} = \underset{T}{\operatorname{argmin}} \left(\sum_{k=1}^{q} \left| T\hat{x}_{k|k-1}^{(id)} - \hat{x}_{k|k-1}^{(true)} \right|^2 \right) = \hat{X}^{(true)} \hat{X}^{(id)\dagger} \tag{9}$$

where $\hat{X}^{(id)}$ and $\hat{X}^{(true)}$ are $n_x \times q$ matrices whose kth columns are constructed of $\hat{x}_{k|k-1}^{(id)}$ and $\hat{x}_{k|k-1}^{(true)}$ respectively. We apply the similarity transform[80] associated with $\hat{T}$ (i.e., the transform that changes the states from $\hat{x}_{k|k-1}^{(id)}$ to $\hat{T}\hat{x}_{k|k-1}^{(id)}$) to the identified model parameters and then quantify the identification error of each parameter as

$$e_{\Psi} = \frac{\left|\Psi^{(id)} - \Psi^{(true)}\right|_F}{\left|\Psi^{(true)}\right|_F} \tag{10}$$

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

$$|\Psi|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m}\left|\psi_{ij}\right|^2}. \tag{11}$$

In addition to computing the identification error for all model parameters, we also compute the error in how accurately the eigenvalues of the state transition matrix A are learned. This metric is an additional indication of how well the dynamics are learned since the eigenvalues of A affect the transfer function from the input to the states and to the neural activity (**Fig. 1b**), and determine the frequency and decay rate with which the latent state responds to excitations from the state noise w_k or from inputs u_k (equation (1))[85]. To measure how well the intrinsic behaviorally relevant dynamics are learned, we evaluate how well the eigenvalues of the behaviorally relevant block of A (i.e., A_{11} in equation (6)), referred to as behaviorally relevant eigenvalues, are learned. These behaviorally relevant eigenvalues are obtained for each method as follows.

IPSID/PSID learn the model in the form of equation (6). So for these IPSID/PSID models, we simply compute the eigenvalues of A_{11} (which has dimension n_1) and find their minimum normalized distance from the true behaviorally relevant eigenvalues by placing the eigenvalues in a vector and computing the error per equation (10), e.g., in **Fig. 2**. Note, in simulations, all methods know the dimension of the true behaviorally relevant latent states denoted by n_1 and the dimension of the latent state in the input model (i.e., dimension of input dynamics) denoted by n_{x_u} (see above). For PSID and because it does not consider the input and thus cannot dissociate intrinsic and input dynamics, in its first stage we use a state dimension equal to $n_1 + n_{x_u}$, so that its first stage has enough state dimensions to capture both the input

dynamics and the intrinsic behaviorally relevant neural dynamics. Then we take the top n_1 state dimensions that are best for decoding behavior as the behaviorally relevant states and evaluate the distance of their associated eigenvalues to the true behaviorally relevant eigenvalues (we refer to this procedure as model reduction, see also below for INDM/NDM).

INDM/NDM do not dissociate the behaviorally relevant latent states. Thus, for INDM/NDM, to find these latent states and their associated eigenvalues in the learned models, we proceed as follows: we first perform a similarity transform (using MATLAB's bdschur command followed by the cdf2rdf command) to find an equivalent model with block-diagonal A. We next use the Kalman estimated latent states in each block to predict behavior. We then sort all blocks in descending order of their behavior decoding performance. Finally, we take the top n_1 eigenvalues associated with the blocks with the best decoding performance as the behaviorally relevant eigenvalues in the identified model. We next compute the error in these eigenvalues using equation (10) as was explained earlier for IPSID/PSID (**Fig. 2**). We refer to this procedure as model reduction; for example, we can fit a high-dimensional INDM model and then reduce its dimension by finding those dimensions/eigenvalues that are best in decoding behavior (e.g., see **Fig. 2c**).

Motor task simulations

We devise a numerical simulation of a brain performing various cursor control tasks (**Fig. 5**). We use this simulation to demonstrate the role of sensory task instructions as inputs to the brain that affect neural dynamics and can confound the learned models of intrinsic neural dynamics. We modeled the brain with two components (**Fig. 5a**): (i) A linear state-space model (LSSM) in the form of equation (12) with $x_k^{(1)}$ corresponding to the 2D position and velocity of the cursor (overall a 4D state; $x_k^{(1)} = [p_k^{(x)}, v_k^{(x)}, p_k^{(y)}, v_k^{(y)}]^T$) , $x_k^{(2)}$ a 2D latent state corresponding to neural dynamics unrelated to behavior, and $x_k^{(3)}$ a 2D latent state corresponding to additional input driven dynamics in behavior that are absent in neural activity. (ii) An optimal feedback controller (OFC) that tries to control the part of the latent state in the LSSM that represents the cursor kinematics (i.e., $x_k^{(1)}$) such that the cursor moves to targets

presented via the task instructions[42–44,70]. The latent brain state x_k, neural activity y_k and behavior z_k evolve as

$$\begin{cases} \begin{bmatrix} x_{k+1}^{(1)} \\ x_{k+1}^{(2)} \\ x_{k+1}^{(3)} \end{bmatrix} = \begin{bmatrix} A_{11} & 0 & 0 \\ A_{21} & A_{22} & 0 \\ 0 & 0 & A_{33} \end{bmatrix} \begin{bmatrix} x_k^{(1)} \\ x_k^{(2)} \\ x_k^{(3)} \end{bmatrix} + \begin{bmatrix} B_1 \\ 0 \\ B_3 \end{bmatrix} u_k + \begin{bmatrix} w_k^{(1)} \\ w_k^{(2)} \\ w_k^{(3)} \end{bmatrix} \\ y_k = \begin{bmatrix} C_{y_1} & C_{y_2} & 0 \end{bmatrix} \begin{bmatrix} x_k^{(1)} \\ x_k^{(2)} \\ x_k^{(3)} \end{bmatrix} + v_k \\ z_k = \begin{bmatrix} C_{z_1} & 0 & 0 \\ 0 & 0 & C_{z_3} \end{bmatrix} \begin{bmatrix} x_k^{(1)} \\ x_k^{(2)} \\ x_k^{(3)} \end{bmatrix} \end{cases} \quad (12)$$

with

$$A_{11} = \begin{bmatrix} 1 & \Delta & 0 & 0 \\ 0 & \alpha & 0 & 0 \\ 0 & 0 & 1 & \Delta \\ 0 & 0 & 0 & \alpha \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 & 0 \\ \beta & 0 \\ 0 & 0 \\ 0 & \beta \end{bmatrix}, \quad C_{z_1} = I_{4\times 4}, \quad x_k^{(1)} = \begin{bmatrix} p_k^{(x)} \\ v_k^{(x)} \\ p_k^{(y)} \\ v_k^{(y)} \end{bmatrix} \quad (13)$$

where $\Delta = 0.01$ second is the duration of each time step, $\alpha = 0.99$ is the damping ratio for velocity, and $\beta = 0.01$. The last two dimensions of behavior z_k are only driven by states $x_k^{(3)}$ that are not reflected in the neural activity and are not engaged in the task, i.e. are not instructed to go to any target (similar to the vertical kinematics of finger position in real neural dataset 1, see ref. [45], because while the kinematics evolve in 3D, only 2 dimensions are used for control on a 2D plane). The OFC component of the brain controls the 2D cursor kinematics, which are the first 4 dimensions of z_k, by generating an internal (unobservable) control command time series, which we will henceforth refer to as c_k. The OFC implements a linear quadratic regulator (LQR). Briefly, let's denote by $x^{(1)^*}$ a desired target state for the behaviorally relevant state (i.e., target as dictated by task instructions). LQR determines the optimal online command c_k as a linear function of the difference between the current state and the desired target state as

$$c_k = -L\left(x_k^{(1)} - x^{(1)^*}\right) \tag{14}$$

where $x^{(1)^*} = [p^{(x)^*}, 0, p^{(y)^*}, 0]^T$, with $p^{(x)^*}$ and $p^{(y)^*}$ specifying the current target position according to task instructions, and the optimal feedback gain matrix L is obtained by solving the discrete algebraic Riccati equation[86]. Replacing the feedback equation (14) into equations (12) and (13) gives the full brain model including both the LSSM and the OFC components. Note that the overall measurable external input to the brain is the task instruction $x^{(1)^*}$, which we henceforth refer to as u_k. Moreover, based on this full brain model, the behaviorally relevant block of the state transition matrix for the full brain model can be written as $A_{11}^{Full} = A_{11} - B_1 L$. Thus, we compute the eigenvalues of A_{11}^{Full} as the ground truth for the behaviorally relevant eigenvalues (**Fig. 5b**).

To generate a random realization of the simulated brain performing the task, we randomly choose a target among the permissible targets in the task (**Fig. 5c**). Then we iterate through equations (12) and (14), starting from the initial value $x_{-1} = 0$, until the brain state $x_k^{(1)}$ reaches the desired target $u_k \equiv x^{(1)^*}$ and stays within its boundary for 0.1 s (**Fig. 5c**). We then randomly choose a new target, update u_k accordingly, and continue iterating through equations (12) and (14). Any time the desired target is reached, a new target is chosen and the data generation process continues as before. We generate $N = 2000000$ data samples with this procedure to get $\{y_k, z_k : 0 \leq k < N\}$ from equation (12). We also keep a timeseries of the desired target position values to use as the overall input ($u_k \equiv x^{(1)^*} : 0 \leq k < N$) to the brain in IPSID and INDM models. Using cross-validation, we identify the model of the brain using IPSID, INDM, PSID and NDM.

Modeling non-human primate neural population activity during non-stereotypical movements

We study two publicly available neural datasets with distinct motor tasks recorded from three macaque monkeys. As the first dataset, we use primary motor cortex (M1) neural data from the Sabes lab[45]. In this experiment, the monkeys (monkey I and monkey L) performed continuous, self-paced reaches to targets chosen randomly with uniform probability from an 8×8 or 8×17 grid, without any time gaps or pre-movement intervals (**Fig. 6a**). The cursor was controlled based on the 2D position of the monkey's

fingertip in the horizonal plane. The task interface was presented to the monkey in a virtual reality environment. We analyze the first spike dimension available for each channel—resulting in 89 to 92 units for monkey I and 91 to 96 units for monkey L—from the first 7 recording sessions for each subject. For faster computation in our analyses, we randomly partitioned the units into two non-overlapping sets of equal sizes and analyzed each set separately. We compute the fingertip's 2D velocity by taking derivative from the recorded 2D position. We take the measured 2D position as well as computed 2D velocity of the fingertip ($p_k^{(x)}, v_k^{(x)}, p_k^{(y)}, v_k^{(y)}$) as the behavior signal z_k.

As the second dataset, we use the dorsal premotor cortex (PMd) data recorded and made publicly available by the Miller lab[46,47]. In this experiment, the monkey (monkey T) performed sequential reaches to random targets on a screen by controlling a cursor via a two-link planar manipulandum (**Fig. 7a**). In this task, an on-screen visual cue (2 cm × 2 cm square) specified the target location for each reach and the monkey was given a liquid reward after making a series of four successful reaches. The location of each target was randomly chosen within 5-10 cm of the previous target. We analyzed single unit spiking activity during all 3 behavioral sessions, with 49, 46, and 57 single units. We take the measured 2D position and velocity of the arm ($p_k^{(x)}, v_k^{(x)}, p_k^{(y)}, v_k^{(y)}$) as the behavior signal z_k.

For all three subjects, we use spike counts counted within non-overlapping 50 ms time intervals and smoothed by Gaussian kernel with a 50 ms s.d.[3,13,39,48] as the neural activity y_k (**Figs. 6, 7, S11**). Smoothing is performed as is typical in the field[3,13,39,48]. Gaussian distributed variables are commonly used to model both local field potentials (LFP)[14,19,30,44,61,65,66] and spike counts[7,19,67,68]. We take the 2D target location in the task as the input time series (u_k) provided to the subject. We perform all analyses within a 5-fold cross-validation and report the cross-validated CC of predicting behavior and neural activity as the performance measures (i.e., decoding and neural self-prediction).

To choose a suitable latent state dimension for our modeling of the intrinsic behaviorally relevant neural dynamics (**Figs. 6, 7, S11**), we estimate the latent state dimension that is sufficient for capturing most of the behavioral dynamics. To do this, we model the behavior time series using INDM as

$$\begin{cases} x_{k+1}^z = A\ x_k^z + Bu_k + w_k^z \\ z_k = C_y\ x_k^z + D_y u_k + v_k^z \end{cases} \quad (15)$$

with different latent state dimensions in the range $1 \leq n_x \leq 50$ (**Fig. S13**). We then find the smallest latent state dimension for which the cross-validated one-step-ahead self-prediction of behavior reaches within 0.9 of its peak value. This dimension is found to be $n_x = 4$ in all three subjects (**Fig. S13**). We use a horizon parameter of $i = 40$ (**Note S1**) for all (I)PSID and (I)NDM modeling in real neural datasets.

As a measure of the learned dynamics, we report the distribution of the learned eigenvalues in each subject by aggregating eigenvalues of the learned models across recording sessions and cross-validation folds, and adding up gaussian kernels (with s.d. of 0.05) centered around each learned eigenvalue. The mean of all gaussian kernels, when normalized to sum up to one over the unit disk, gives a probability mass function that estimates the probability of an eigenvalue being identified at each location on the complex plane in that subject (**Figs. 6b**, **7b, S11b**). We find the mode of the eigenvalue distributions, i.e., locations on the complex plane that have peak eigenvalue identification probability, for a given method and report their associated frequency and decay rate. The frequency and decay rate describe how a complex conjugate pair of eigenvalues at a given location on the complex plane would respond to an impulse excitation, i.e., they specify the ringing frequency of the response and how fast it would decay to less than 1% of its initial value. For a point $\lambda = |\lambda|e^{j\omega}$, the associated frequency is computed as $\frac{\omega F_s}{2\pi}$ where $F_s = 20\ Hz$ is the data sampling rate, and the decay rate is computed as $\frac{n}{F_s}$ where n is the smallest integer for which $|\lambda|^n < e^{-1}$.

To quantify how close the identified eigenvalues are to the input eigenvalues (**Figs. 6c**, **7c, S11c**), we compute the pointwise *KL-divergence* between the probability mass function found using the identified method P_{method}, and the probability mass function of input eigenvalues P_{input} as $D_{KL}(P_{input} \parallel P_{method})$. Similarly, to quantify the distance between the eigenvalue distributions found using a method for two subjects *i* and *j* ($P_{subject_i}$ and $P_{subject_j}$), we computed their correlation coefficient, *symmetric KL-divergence*, defined as: $\frac{1}{2}[D_{KL}(P_{subject_i} \parallel P_{subject_j}) + D_{KL}(P_{subject_i} \parallel P_{subject_j})]$, as well as the *mode*

distance defined as the Euclidean distance between the locations with peak eigenvalue occurrence probabilities in $P_{subject_i}$ and $P_{subject_j}$.

Statistics

We use the Wilcoxon signed-rank tests for all paired statistical tests.

Supplementary Notes

Note S1 | Preferential subspace identification in presence of inputs (IPSID)

Definitions

To simplify the description of IPSID, we define some notations. First, we define the notation

$$Z/Y \triangleq ZY^T (YY^T)^\dagger Y \tag{16}$$

to denote the orthogonal projection[40] of the wide matrix $Z \in \mathbb{R}^{m\times j}$ onto another wide matrix $Y \in \mathbb{R}^{n\times j}$ – where the superscript † denotes the pseudoinverse operation. An orthogonal projection can also be thought of as the linear minimum mean squared prediction of columns of Z using columns of Y. This is because $ZY^T (YY^T)^\dagger$ is equal to cross-covariance of columns of Z and Y, multiplied by the inverse of the covariance of the columns of Y. Second, we define the shorthand notation[40]

$$\Pi_U \triangleq I - U^T (UU^T)^\dagger U \tag{17}$$

where $I \in \mathbb{R}^{j\times j}$ is the identity matrix. Π_U is a matrix that when multiplied from the left by a matrix Z, would remove the orthogonal projection of Z onto U from Z. In other words, we have $Z\Pi_U \triangleq Z - ZU^T (UU^T)^\dagger U = Z - Z/U$ for any matrix $Z \in \mathbb{R}^{m\times j}$. Third, we define the notation

$$Z/_U Y \triangleq (Z\Pi_U)(Y\Pi_U)^T[(Y\Pi_U)(Y\Pi_U)^T]^\dagger Y\Pi_U = (Z\Pi_U)/(Y\Pi_U) \tag{18}$$

to denote the oblique (i.e. non-orthogonal) projection[40] of matrix Z onto Y along matrix $U \in \mathbb{R}^{p\times j}$. Intuitively, the oblique projection in equation (18) means that we first find the part of Z that is not predictable from U (i.e. $Z\Pi_U$), and then project that part onto the part of Y that is not predictable from U (i.e. $Y\Pi_U$). This gives us the part of Z that is predictable by Y but not explained by U.

We also define the following matrices form the training neural time series $\{y_k \in \mathbb{R}^{n_y}: 0 \leq k < N\}$

$$\begin{bmatrix} Y_p \\ \hline Y_f \end{bmatrix} \triangleq \left[\begin{array}{cccc} y_0 & y_1 & \cdots & y_{j-1} \\ y_1 & y_2 & \cdots & y_j \\ \vdots & \vdots & \ddots & \vdots \\ y_{i-1} & y_i & \cdots & y_{j+i-1} \\ \hline y_i & y_{i+1} & \cdots & y_{j+i} \\ y_{i+1} & y_{i+2} & \cdots & y_{j+i+1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{2i-1} & y_{2i} & \cdots & y_{j+2i-1} \end{array}\right] = \left[\begin{array}{cccc} y_0 & y_1 & \cdots & y_{j-1} \\ y_1 & y_2 & \cdots & y_j \\ \vdots & \vdots & \ddots & \vdots \\ y_{i-1} & y_i & \cdots & y_{j+i-1} \\ \hline y_i & y_{i+1} & \cdots & y_{j+i} \\ \hline y_{i+1} & y_{i+2} & \cdots & y_{j+i+1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{2i-1} & y_{2i} & \cdots & y_{j+2i-1} \end{array}\right] \triangleq \begin{bmatrix} Y_p^+ \\ \hline Y_f^- \end{bmatrix} \triangleq \begin{bmatrix} Y_p \\ \hline Y_i \\ \hline Y_f^- \end{bmatrix} \tag{19}$$

and analogously define the following from the training behavior time series $\{z_k \in \mathbb{R}^{n_z} : 0 \leq k < N\}$

$$\begin{bmatrix} Z_p \\ \hline Z_f \end{bmatrix} \triangleq \left[\begin{array}{cccc} z_0 & z_1 & \cdots & z_{j-1} \\ z_1 & z_2 & \cdots & z_j \\ \vdots & \vdots & \ddots & \vdots \\ z_{i-1} & z_i & \cdots & z_{j+i-1} \\ \hline z_i & z_{i+1} & \cdots & z_{j+i} \\ z_{i+1} & z_{i+2} & \cdots & z_{j+i+1} \\ \vdots & \vdots & \ddots & \vdots \\ z_{2i-1} & z_{2i} & \cdots & z_{j+2i-1} \end{array}\right] = \left[\begin{array}{cccc} z_0 & z_1 & \cdots & z_{j-1} \\ z_1 & z_2 & \cdots & z_j \\ \vdots & \vdots & \ddots & \vdots \\ z_{i-1} & z_i & \cdots & z_{j+i-1} \\ \hline z_i & z_{i+1} & \cdots & z_{j+i} \\ \hline z_{i+1} & z_{i+2} & \cdots & z_{j+i+1} \\ \vdots & \vdots & \ddots & \vdots \\ z_{2i-1} & z_{2i} & \cdots & z_{j+2i-1} \end{array}\right] \triangleq \begin{bmatrix} Z_p^+ \\ \hline Z_f^- \end{bmatrix} \triangleq \begin{bmatrix} Z_p \\ \hline Z_i \\ \hline Z_f^- \end{bmatrix} \tag{20}$$

and the following from the input time series $\{u_k \in \mathbb{R}^{n_u} : 0 \leq k < N\}$

$$\begin{bmatrix} U_p \\ \hline U_f \end{bmatrix} \triangleq \left[\begin{array}{cccc} u_0 & u_1 & \cdots & u_{j-1} \\ u_1 & u_2 & \cdots & u_j \\ \vdots & \vdots & \ddots & \vdots \\ u_{i-1} & u_i & \cdots & u_{j+i-1} \\ \hline u_i & u_{i+1} & \cdots & u_{j+i} \\ u_{i+1} & u_{i+2} & \cdots & u_{j+i+1} \\ \vdots & \vdots & \ddots & \vdots \\ u_{2i-1} & u_{2i} & \cdots & u_{j+2i-1} \end{array}\right] = \left[\begin{array}{cccc} u_0 & u_1 & \cdots & u_{j-1} \\ u_1 & u_2 & \cdots & u_j \\ \vdots & \vdots & \ddots & \vdots \\ u_{i-1} & u_i & \cdots & u_{j+i-1} \\ \hline u_i & u_{i+1} & \cdots & u_{j+i} \\ \hline u_{i+1} & u_{i+2} & \cdots & u_{j+i+1} \\ \vdots & \vdots & \ddots & \vdots \\ u_{2i-1} & u_{2i} & \cdots & u_{j+2i-1} \end{array}\right] \triangleq \begin{bmatrix} U_p^+ \\ \hline U_f^- \end{bmatrix} \triangleq \begin{bmatrix} U_p \\ \hline U_i \\ \hline U_f^- \end{bmatrix} \tag{21}$$

where i is a provided quantity termed the projection horizon, and j is number of columns in the abovementioned block matrices. Given the number of data samples N, we set $j = N - 2i$ to use all data samples.

Outline of the IPSID algorithm

Here, we describe the IPSID algorithm. Given the neural, behavioral, and input training time series, i.e. $\{y_k \in \mathbb{R}^{n_y} : 0 \leq k < N\}$, $\{z_k \in \mathbb{R}^{n_z} : 0 \leq k < N\}$ and $\{u_k \in \mathbb{R}^{n_u} : 0 \leq k < N\}$, respectively, and given the state dimension n_x, parameter $n_1 \leq n_x$, and projection horizon i, IPSID learns the parameters of the model in equation (1) while prioritizing the learning of intrinsic behaviorally relevant states. Doing so, IPSID can dissociate intrinsic behaviorally relevant neural dynamics from input dynamics and from other intrinsic neural dynamics.

Stage 1: *Extract $\mathbf{n_1}$ latent states directly from data via an oblique projection of future behavior onto past neural activity and past input, along future inputs.*

1. Form examples of future behavior Z_f (equation (20)) and the associated past neural activity Y_p (equation (19)). Also form the corresponding samples of future and past input U_f and U_p (equation (21)). Project Z_f onto Y_p and U_p along U_f to get

$$\hat{Z}_f^{(O)} = Z_f /_{U_f} \begin{bmatrix} U_p \\ Y_p \end{bmatrix} \tag{22}$$

   where oblique projection is defined as in equation (18).

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

$$\hat{Z}_f^{(O)} = USV^T \approx U_1 S_1 V_1^T \tag{23}$$

3. Compute the intrinsic behaviorally relevant latent state as

$$\hat{X}_i^{(1)} = \left(U_1 S_1^{\frac{1}{2}}\right)^{\dagger} Z_f / \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix} \tag{24}$$

Stage 2 (optional): *extract $\mathbf{n_x - n_1}$ additional latent states via an oblique projection of residual future neural activity onto past neural activity and past input, along future input.*

1. Find the prediction of Y_f from $\hat{X}_i^{(1)}$ and subtract this prediction from Y_f using an oblique projection to keep the part that is predictable from U_f and U_p. Name the result Y_f' (i.e., residual future neural activity):

$$Y_f' = Y_f - Y_f /_{\begin{bmatrix} U_p \\ U_f \end{bmatrix}} \hat{X}_i^{(1)} \Pi^{-1}_{\begin{bmatrix} U_p \\ U_f \end{bmatrix}} \tag{25}$$

   Here $\Pi_{\begin{bmatrix} U_p \\ U_f \end{bmatrix}}$ is defined per equation (17).

2. Project the residual future neural activity (Y_f') onto Y_p and U_p along U_f to get

$$\hat{Y}_f'^{(O)} = Y_f' /_{U_f} \begin{bmatrix} U_p \\ Y_p \end{bmatrix} \tag{26}$$

3. Compute the SVD of $\hat{Y}_f'^{(O)}$, and keep the top $n_x - n_1$ singular values:

$$\hat{Y}_f'^{(O)} = U'S'V'^T \approx U_2 S_2 V_2^T \tag{27}$$

4. Compute the remaining latent states as

$$\hat{X}_i^{(2)} = \left(U_2 S_2^{\frac{1}{2}}\right)^{\dagger} Y_f / \begin{bmatrix} U_p \\ Y_p \\ U_f \end{bmatrix} \tag{28}$$

Final step: *given the extracted latent states, identify model parameters.*

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 ($\hat{X}_{i+1}$). To shift the time step, use Z_f^- (equation (20)), Y_f^- (equation (19)), Y_p^+ (equation (19)), U_f^- (equation (21)), and U_p^+ (equation (21)) instead of Z_f, Y_f, Y_p, U_f and U_p, respectively.
3. Compute A_{11}, A_{21}, A_{22}, C_y and C_z based on least squares solutions of equations (6) as

$$A_{11} = \hat{X}_{i+1}^{(1)} \begin{bmatrix} \hat{X}_i^{(1)} \\ U_f \end{bmatrix}^{\dagger} \Bigg|_{first\ n_1\ columns} \tag{29}$$

$$[A_{21} \quad A_{22}] = \hat{X}_{i+1}^{(2)} \begin{bmatrix} \hat{X}_i \\ U_f \end{bmatrix}^{\dagger} \Bigg|_{first\ n_x\ columns} \tag{30}$$

$$C_y = Y_i \begin{bmatrix} \hat{X}_i \\ U_f \end{bmatrix}^{\dagger} \Bigg|_{first\ n_x\ columns} \tag{31}$$

$$C_z = Z_i \begin{bmatrix} \hat{X}_i \\ U_f \end{bmatrix}^{\dagger} \Bigg|_{first\ n_x\ columns} \tag{32}$$

where Y_i and Z_i are as defined in equations (19) and (20), respectively.

4. Compute an estimate of the noise time series w_k and v_k from equation (1) based on the residuals/errors of the least squares solutions from the previous step as

$$\widehat{W}^{(1)} = \hat{X}_{i+1}^{(1)} - \hat{X}_{i+1}^{(1)} / \begin{bmatrix} \hat{X}_i^{(1)} \\ U_f \end{bmatrix} \tag{33}$$

$$\widehat{W}^{(2)} = \hat{X}_{i+1}^{(2)} - \hat{X}_{i+1}^{(2)} / \begin{bmatrix} \hat{X}_i \\ U_f \end{bmatrix} \tag{34}$$

$$\hat{V} = Y_i \ - Y_i / \begin{bmatrix} \hat{X}_i \\ U_f \end{bmatrix} \tag{35}$$

5. Compute the covariances and cross-covariance of w_k and v_k as

$$\begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} = \frac{1}{j}\begin{bmatrix} \widehat{W}^{(1)} \\ \widehat{W}^{(2)} \\ \widehat{V} \end{bmatrix}\begin{bmatrix} \widehat{W}^{(1)} \\ \widehat{W}^{(2)} \\ \widehat{V} \end{bmatrix}^T \quad (36)$$

6. Follow a procedure similar to ref. [40], pages 125-127, to find the least squares solution for the model parameters B and D_y.
7. This concludes the learning of all model parameters in the first two rows of equation (1) and thus we can now run a Kalman filter to recursively estimate the latent states (without looking at behavior) as

$$\hat{x}_{k+1} = \left(A - KC_y\right)\hat{x}_k + \left(B - KD_y\right)u_k + Ky_k \quad (37)$$

where K is the steady state Kalman gain (**SI Methods** equation (7)).

8. Finally, compute the parameter D_z, which captures the direct non-dynamic effect of input on behavior, via an orthogonal projection as

$$D_z = \{z_k - C_z\hat{x}_k\}_{1:N}/\{u_k\}_{1:N} \quad (38)$$

where $\{\alpha_k\}_{1:N}$ denotes constructing an N-column matrix with column n containing α_n.

The above concludes the learning of all model parameters using IPSID. For the special case of n_1 = 0, only stage 2 will be performed and the algorithm will reduce to INDM[40], which does not prioritize or dissociate the learning of intrinsic behaviorally relevant neural dynamics.

Note S2 | IPSID for scenarios where neural recordings do not capture all downstream regions of input that influence behavior

Here, we describe how IPSID can also support scenarios where the recorded neural activity does not cover all of the downstream regions of the input. In other words, as opposed to the main IPSID algorithm (**Note S1**), here we allow existence of latent dynamics that are affected by input and contribute to the generation of behavior but are not encoded in the recorded neural activity (i.e., $x_k^{(3)}$ in equation (2)). We now outline the key steps that need to be performed in addition to the steps described in **Note S1** to enable this extension. Compared with **Note S1**, here in addition to the state dimension n_x and the parameter n_1 specifying the dimension of the behaviorally relevant latent states, an additional parameter n_2 should also be specified by the user that determines the dimension of $x_k^{(2)}$, i.e., the intrinsic latent states extracted beyond those that are behaviorally relevant. In **Note S1**, n_2 was automatically inferred as $n_2 = n_x - n_1$; but here the user can specify any n_2 in the range $0 \leq n_2 \leq n_x - n_1$. This version of the IPSID algorithm then learns the parameters of the model in equation (2), where a new set of states denoted by $x_k^{(3)}$ with $n_x - n_1 - n_2$ dimensions describe the behavior dynamics that are predictable from input but are not reflected in recorded neural activity.

Initial projection step: *Find the part of future behavior that is encoded in neural activity.*

1. Model the neural signal y_k using only stage 2 of IPSID (i.e., INDM) to extract m latent states $\hat{X}_i^{(y)}$ that describe the neural activity, where m is the total number of latent states that drive the neural activity (sum of dimensions of $x_k^{(1)}$ and $x_k^{(2)}$ in the true model of equation (2)). The appropriate value of m for a given data can be correctly estimated by increasing n_x in the model until neural self-prediction reaches a peak performance (**Fig. S8b**). For analysis in real data (**Figs. 6, 7, S11**), we use m = 150.
2. Form examples of future behavior Z_f (equation (20)) and find its oblique projection onto $\hat{X}_i^{(y)}$ along U_f and U_p, naming the result Z_f'

$$Z_f' = Z_f /_{\begin{bmatrix} U_p \\ U_f \end{bmatrix}} \hat{X}_i^{(y)} \Pi_{\begin{bmatrix} U_p \\ U_f \end{bmatrix}}^{-1} \tag{39}$$

where oblique projection is defined as in equation (18) and $\Pi_{\begin{bmatrix} U_p \\ U_f \end{bmatrix}}$ is defined as in equation (17).

The above steps extract the part of the future behavior that is encoded in the recorded neural activity, i.e., Z_f'. We next perform IPSID stages 1 and 2 as in **Note S1**, with the only difference being the use of Z_f' instead of Z_f.

Stage 1: *perform the IPSID stage 1 as outlined in* ***Note S1*** *but use* Z_f' *(the part of future behavior that is encoded in neural activity) instead of* Z_f*, to extract* n_1 *intrinsic behaviorally relevant neural states* $\hat{X}_i^{(1)}$.

Stage 2 (optional): *perform the IPSID stage 2 as outlined in* ***Note S1****, to extract* $\boldsymbol{n_2}$ *additional neural latent states* $\hat{X}_i^{(2)}$.

Parameter learning step for neural dynamics: *given the extracted neural latent states, learn all model parameters in equation (2) that are related to the neural dynamics.*

1. Apply the final parameter learning step of **Note S1** to learn all model parameters in equation (2) that are associated with the recorded neural dynamics i.e., $A_{11}, A_{21}, A_{22}, B_1, B_2, C_y^{(1)}, C_y^{(2)}, D_y$ and noise statistics.

2. Learn the $C_z^{(1)}$ parameter from equation (2) as

$$C_z^{(1)} = Z_i \hat{X}_i^{\dagger} \tag{40}$$

where Z_i is as defined as in equation (20).

This concludes the learning of all intrinsic neural dynamics. We next proceed with an optional step that if desired can learn additional behavior dynamics that are predictable from input but are not encoded in neural activity (i.e., $x_k^{(3)}$ in equation (2)).

Learn input-driven behavior dynamics not encoded in recorded neural activity (Optional): *extract* $n_x - n_1 - n_2$ *additional latent states that are not encoded in neural activity but describe behavior dynamics predictable from input.*

1. Run the Kalman filter for the learned neural model (equation (7)) to estimate neural latent states $\hat{x}_k$ and the prediction of behavior using those latent states (equation (8)). Then subtract this prediction from the behavior signal to get the residual behavior signal z_k'' that is not predictable from the neural latent states.

2. Apply IPSID stage 2 (i.e. INDM) to z_k'' instead of y_k to extract $n_x - n_1 - n_2$ latent states $\hat{X}_i^{(3)}$ and learn their associated model parameters A_{33}, B_3, and $C_z^{(3)}$ from equation (2) to conclude this version of IPSID.

Supplementary Figures

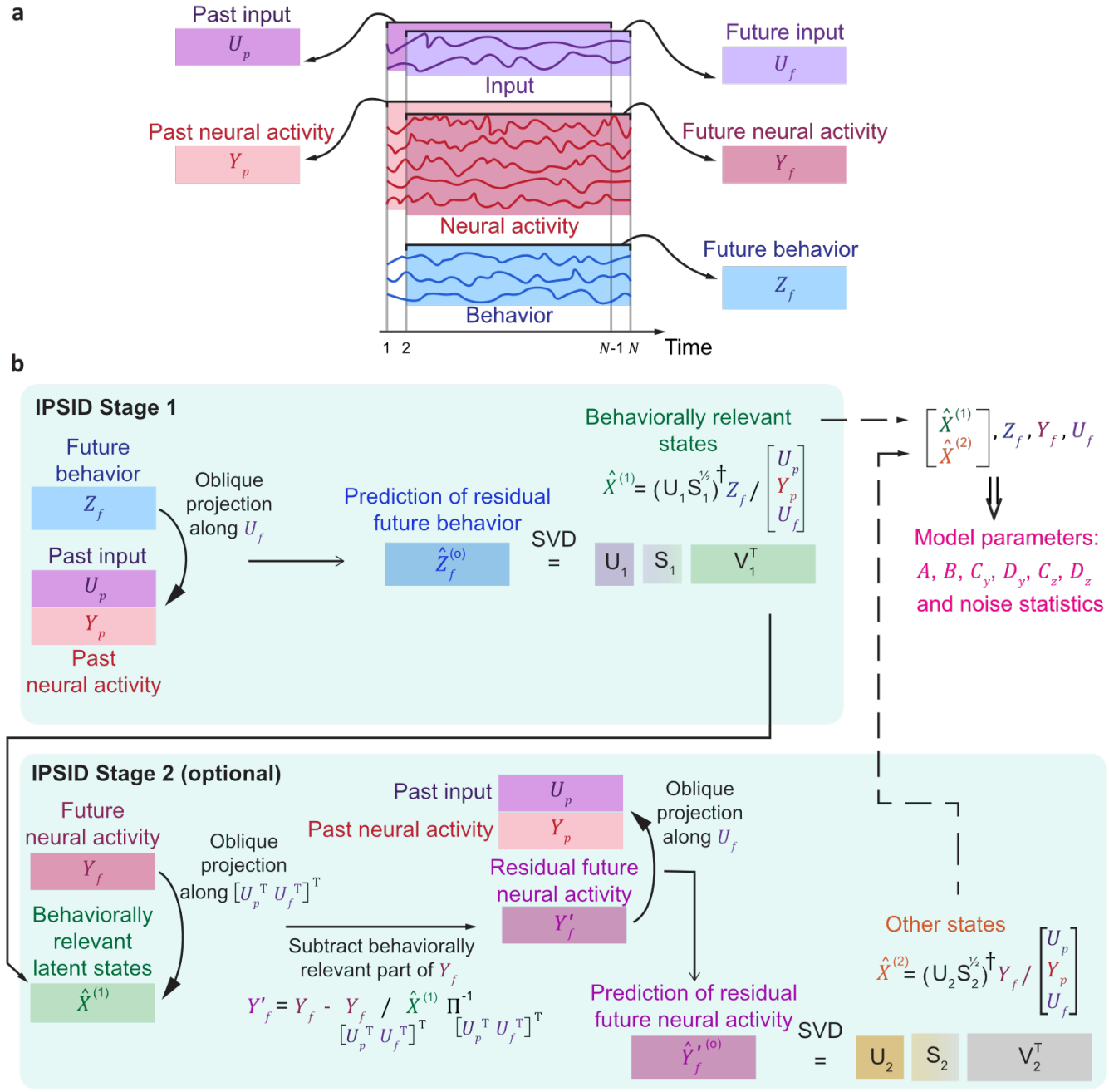

Fig. S1 | Visualization of the IPSID algorithm.

(a) The extraction of future and past training data is shown. Here, U, Y and Z denote input, neural activity and behavior, respectively. Colored rectangles represent data matrices used to extract the latent state (see **Note S1** for general definition). Future matrices U_f, Y_f and Z_f are constructed by shifting the columns of the past matrices one step ahead in time. **(b)** In the first stage of IPSID, the intrinsic behaviorally relevant states $\hat{X}^{(1)}$ are extracted from data with priority with the following procedure: Future behavior Z_f is projected onto the concatenation of past input U_p and past neural activity Y_p along the subspace spanned by the future input U_f to obtain $\hat{Z}_f^{(o)}$, which is the prediction of residual future behavior where residual refers to the part not predictable by future inputs (**Note S1**). Performing SVD on $\hat{Z}_f^{(o)}$ gives the intrinsic behaviorally relevant states $\hat{X}^{(1)}$. In the second stage of IPSID, which is optional, any remaining intrinsic latent states $\hat{X}^{(2)}$ are extracted from the data with the following procedure: The predictable part of future neural activity Y_f from $\hat{X}^{(1)}$ is removed to obtain the residual

future neural activity $Y_f{}'$, i.e., the part not predictable by $\hat{X}^{(1)}$ (operator Π is defined in **Note S1**). Further, oblique projection of $Y_f{}'$ onto concatenation of the past input and past neural activity along the subspace spanned by the future input U_f results in $\hat{Y}_f{}'^{(o)}$, which is the prediction of residual future neural activity, where residual refers to the part not predictable by future inputs or by $\hat{X}^{(1)}$. Performing SVD on $\hat{Y}_f{}'^{(o)}$ gives the remaining intrinsic latent states $\hat{X}^{(2)}$. Once the latent states $\hat{X}$ are extracted—$\hat{X} = \hat{X}^{(1)}$ if only using the first stage or $\hat{X} = [\hat{X}^{(1)^T}, \hat{X}^{(2)^T}]^T$ if also using the optional second stage—, model parameters can be learned using linear regression (**Note S1**).

1325

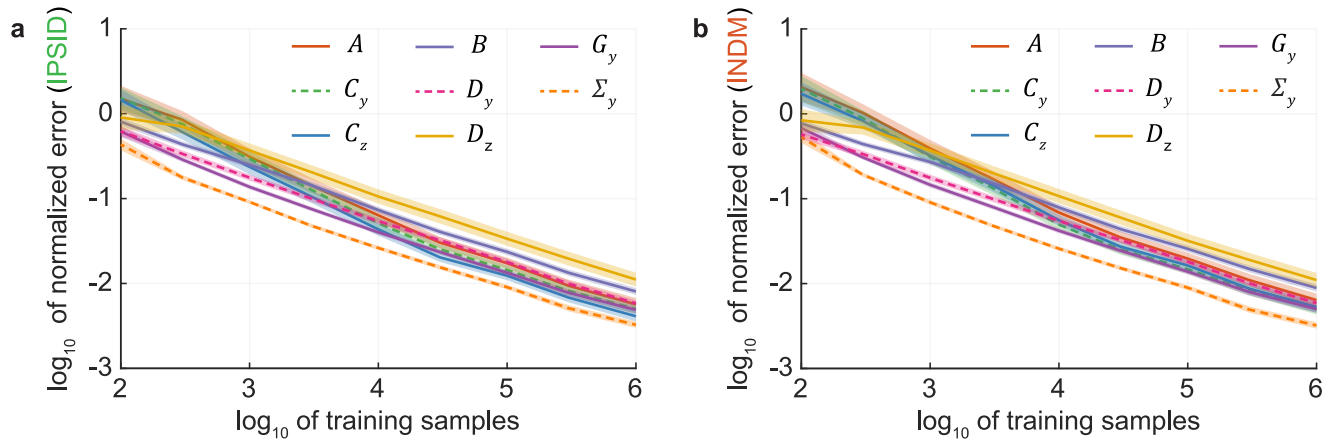

Fig. S2 | IPSID correctly learns all model parameters.

(a) Normalized error in learning each model parameter using IPSID versus the number of training samples used for learning. Models are as in equation (1). Solid lines show the mean across the random models and the shaded areas show the s.e.m. (n = 100 random models). Parameters $G_y \triangleq E\{y_{k+1}x_k^T\}$ and $\Sigma_y \triangleq E\{y_k y_k^T\}$ fully represent the noise statistics, unlike Q, R, and S from equation (5) in **SI Methods**, which are a redundant representation that is not uniquely identifiable and thus is not suitable for evaluating model identification methods[19,40,80]. **(b)** same as (a) for INDM.

1326

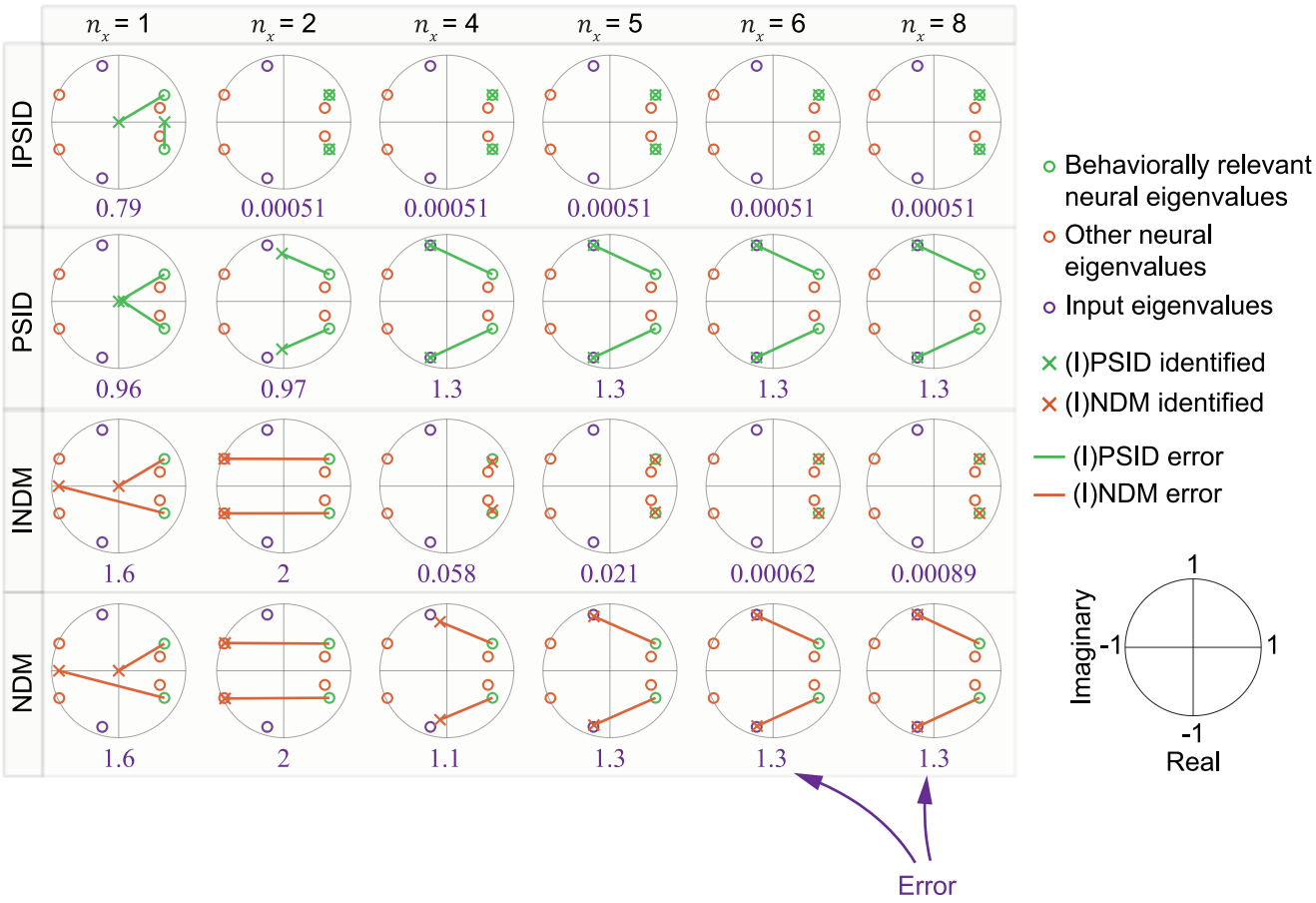

Fig. S3 | Unlike other methods, IPSID correctly learns the intrinsic behaviorally relevant neural dynamics in the presence of input even when using lower-dimensional latent states (i.e., even when performing dimensionality reduction).

For one simulated model as in equation (1), the identified intrinsic behaviorally relevant eigenvalues of the state transition matrix A are shown for (I)PSID and (I)NDM for different latent state dimensions. These eigenvalues quantify the dynamics (**SI Methods**). True model eigenvalues are shown as colored circles, with colors indicating their relevance to input, 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 = 2), missing eigenvalues are taken as 0, representing an equivalent model for which $n_1 - n_x$ latent state dimensions are always 0. Thus, all cases have 2 crosses indicating 2 identified eigenvalues ($n_1 - n_x$ of which are zero when $n_x < n_1$). Lines indicate the error of the identified eigenvalues. The normalized value of the error—average line length normalized by the average true eigenvalue magnitude—is noted below each plot (**SI Methods**). Unlike IPSID, INDM may learn dynamics that are unrelated to behavior at lower state dimensions (i.e., when performing dimensionality reduction). NDM and PSID do not consider input and thus may learn dynamics that are confounded/influenced by input dynamics.

1327

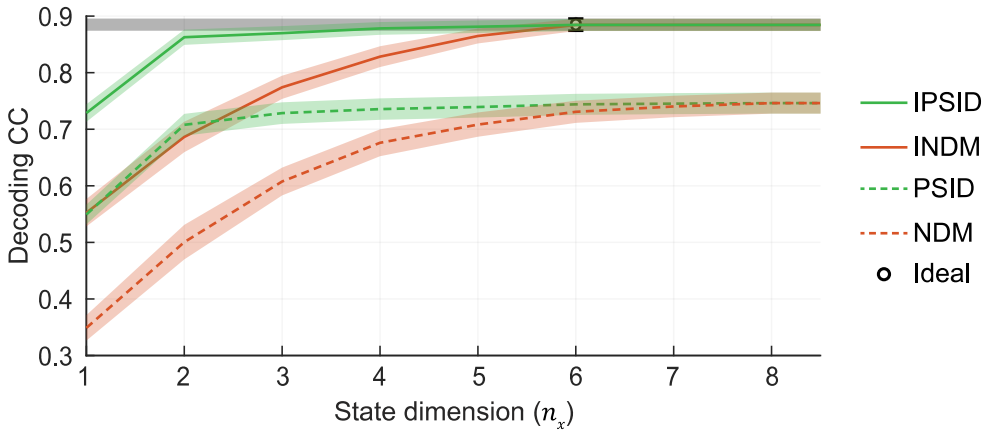

Fig. S4 | Quantified by behavior decoding accuracy, IPSID correctly prioritizes learning of intrinsic behaviorally relevant neural dynamics in the presence of input.

Cross-validated behavior decoding correlation coefficient (CC) given the same random models as **Fig. 2b**. Ideal decoding CC using true model parameters is shown in black. IPSID/INDM consider input and thus are able to achieve peak decoding similar to the ideal decoding, with IPSID achieving the peak decoding using much lower-dimensional latent states than INDM. This shows the IPSID correctly prioritizes the learning of intrinsic behaviorally relevant neural dynamics unlike INDM. NDM/PSID do not consider input and thus do not reach ideal behavior decoding accuracy even with high-dimensional latent states. Note that for models that have input (i.e., those learned with IPSID and INDM), the input is observed for decoding and thus here we use behavior decoding accuracy simply as a measure of how well the behaviorally relevant neural dynamics are explained by the model and not as a measure of pure neural decoding of behavior.

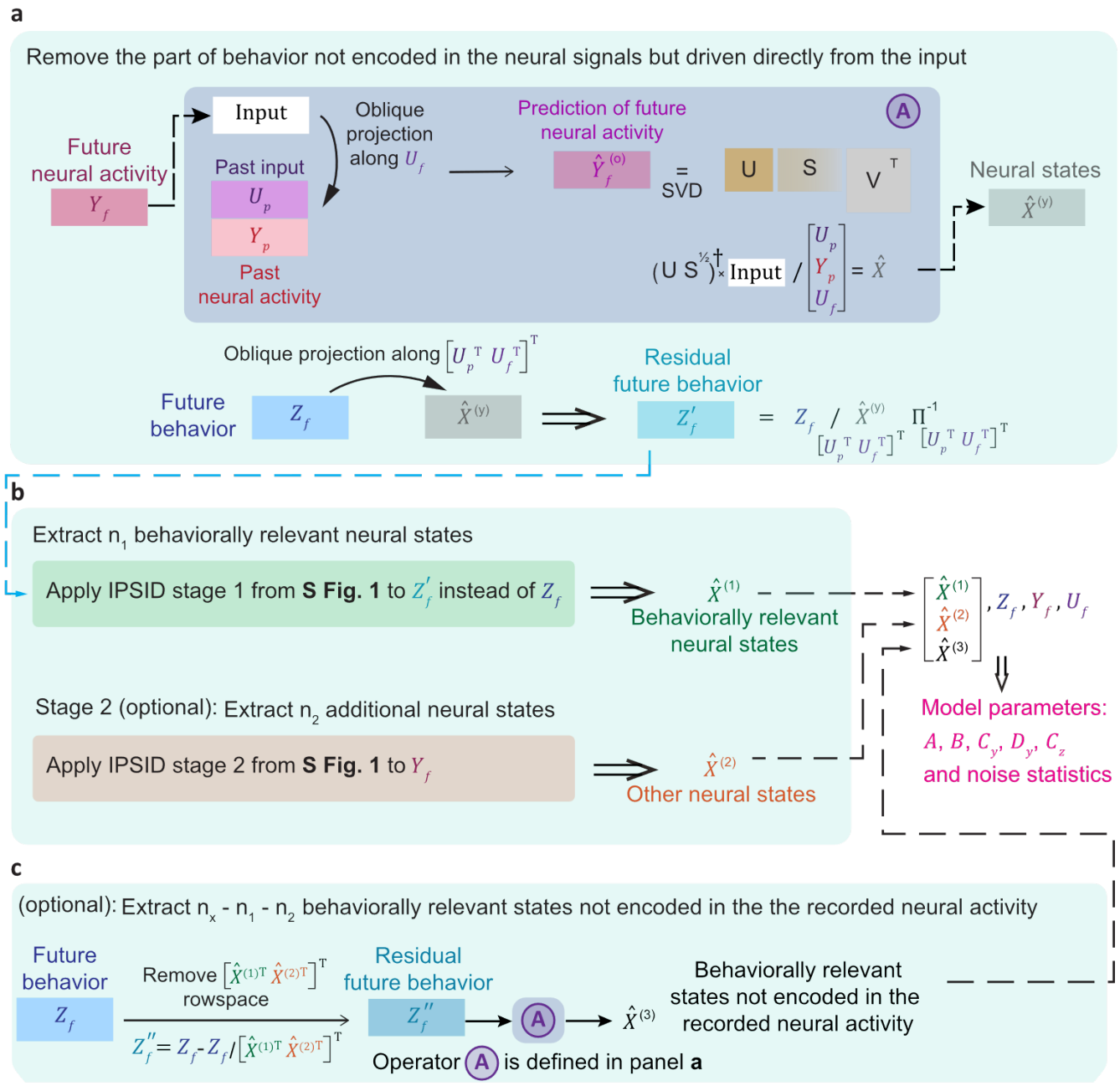

Fig. S5 | Visualization of IPSID algorithm with support for scenarios where the recorded regions do not cover all downstream regions of the input.

(a) To support the scenario where recorded regions do not cover all downstream regions of input, we add an additional step that precedes the IPSID two-stage approach presented in **Fig. S1** and **Note S1**. Specifically, before extraction of intrinsic behaviorally relevant latent states in stage 1, the future behavior Z_f is projected onto a latent state representation, termed $\hat{X}^{(y)}$, of all neural activity (operator Π is defined in **Note S1**); we extract the $\hat{X}^{(y)}$ using the second stage of IPSID (the purple box which is also called operator Ⓐ). The dimension of the latent representation $\hat{X}^{(y)}$ should be chosen high enough such that neural dynamics are well-represented in $\hat{X}^{(y)}$. In analyses of real data, we determine this dimension by forming a plot of neural self-prediction using $\hat{X}^{(y)}$ vs the dimension of $\hat{X}^{(y)}$, and finding a dimension that reaches a self-prediction close to the peak (**Fig. S8b**). We refer to the result of the projection of future behavior Z_f onto the latent representation $\hat{X}^{(y)}$ as Z'_f, which is thus the part of future behavior dynamics that is reflected in the neural recordings. This step excludes any behavior dynamics that are not represented in the recorded neural activity from being learned in stage 1. **(b)** Next, IPSID stage 1 and 2 follow as explained in **Fig. S1** and **Note S1**, but this time the residual future behavior Z'_f (from panel a) is used in these stages instead of the original future behavior Z_f **(**see **Note S2)**. This concludes the learning of intrinsic behaviorally relevant

neural dynamics (stage 1) and dissociating them from other intrinsic neural dynamics (stage 2). **(c)** Subsequently, we can optionally learn behavior dynamics that are predictable from input but are not reflected in the recorded neural activity. To do so, in a second optional step, we compute the residual behavior that is not yet predictable using the already-extracted latent states (i.e., using $\hat{X}^{(1)}$, $\hat{X}^{(2)}$), and apply the second stage of IPSID (operator Ⓐ) on that residual behavior signal. This is done by replacing future and past neural signals with future and past residual behavior signals in this second stage. This step allows us to build a model that predicts these residual behavior dynamics purely using the input via forward prediction. This concludes the dissociation of behavior dynamics that are encoded in the neural activity from those that are predictable from input but are not encoded in the recorded neural activity.

1329

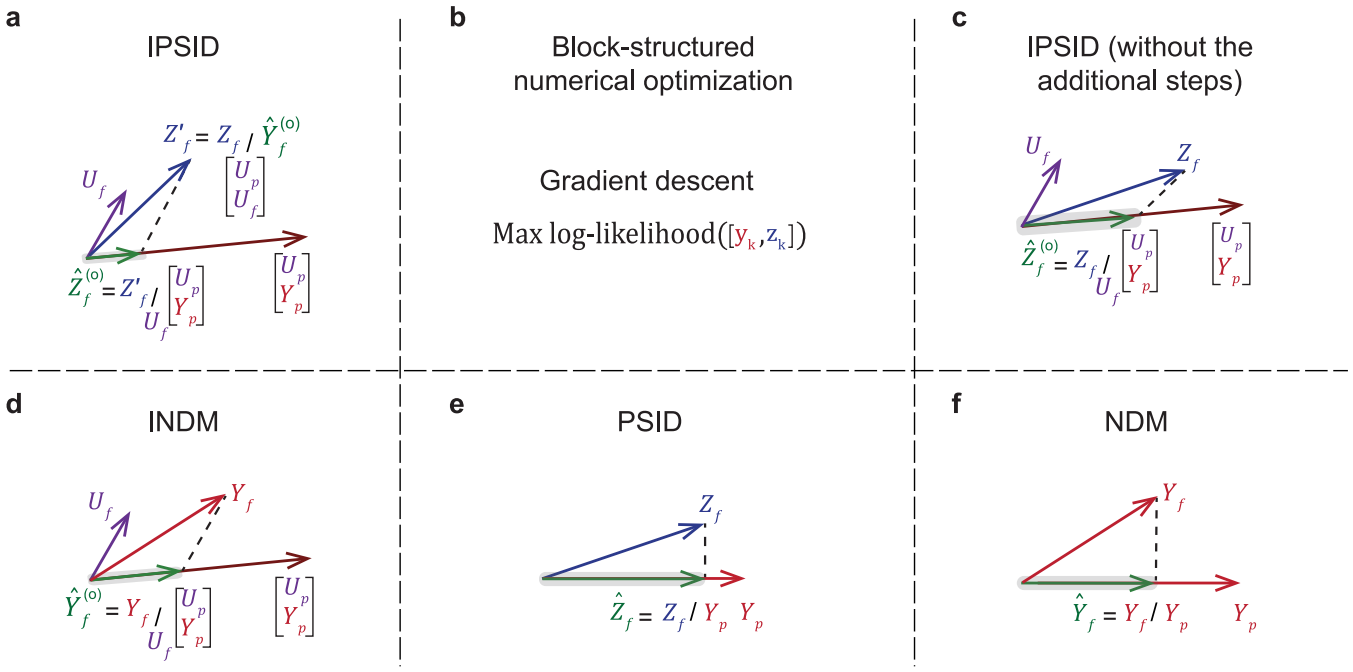

Fig. S6 | Simplified schematic overview of the key operations for subspace and numerical optimization learning methods. A schematic overview of the key operations involved in **(a)** IPSID, **(b)** block-structured numerical optimization, **(c)** IPSID (without additional steps), **(d)** INDM, **(e)** PSID and **(f)** NDM. Z_f, Y_f, Y_p, and U_f, U_p denote future behavior, future and past neural activity, and future and past input, respectively (**Note S1**, **Fig. S1**). A/B denotes an orthogonal projection of A onto B and $A/_C B$ denotes an oblique projection of A onto B along C (**Note S1**). A block-structured numerical optimization method is additionally compared which learns the model parameters of equation (6) from **SI Methods** via gradient descent (**SI Methods**).

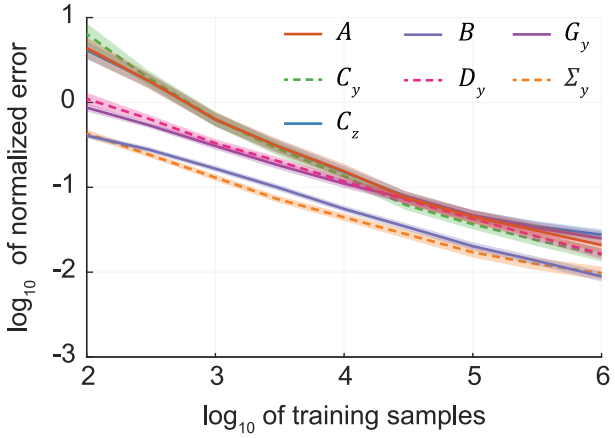

Fig. S7 | IPSID correctly learns model parameters even in scenarios where the recorded regions do not cover all downstream regions of the input.

Notation is as in **Fig. S2a**, but for simulating models where some behavior dynamics that are influenced by inputs are not reflected in the recorded neural activity. Here 100 random models are simulated according to equation (2). Evaluated parameters are for the parts of the model that are relevant to the neural states in the recordings, i.e., $x_k^{(1)}$ and $x_k^{(2)}$. Learning $x_k^{(3)}$ entails stage 2 of the same method as was validated in **Fig. S2a** (**Fig. S5c**); this $x_k^{(3)}$ learning is also validated in a different way in **Fig. S8** next.

1331

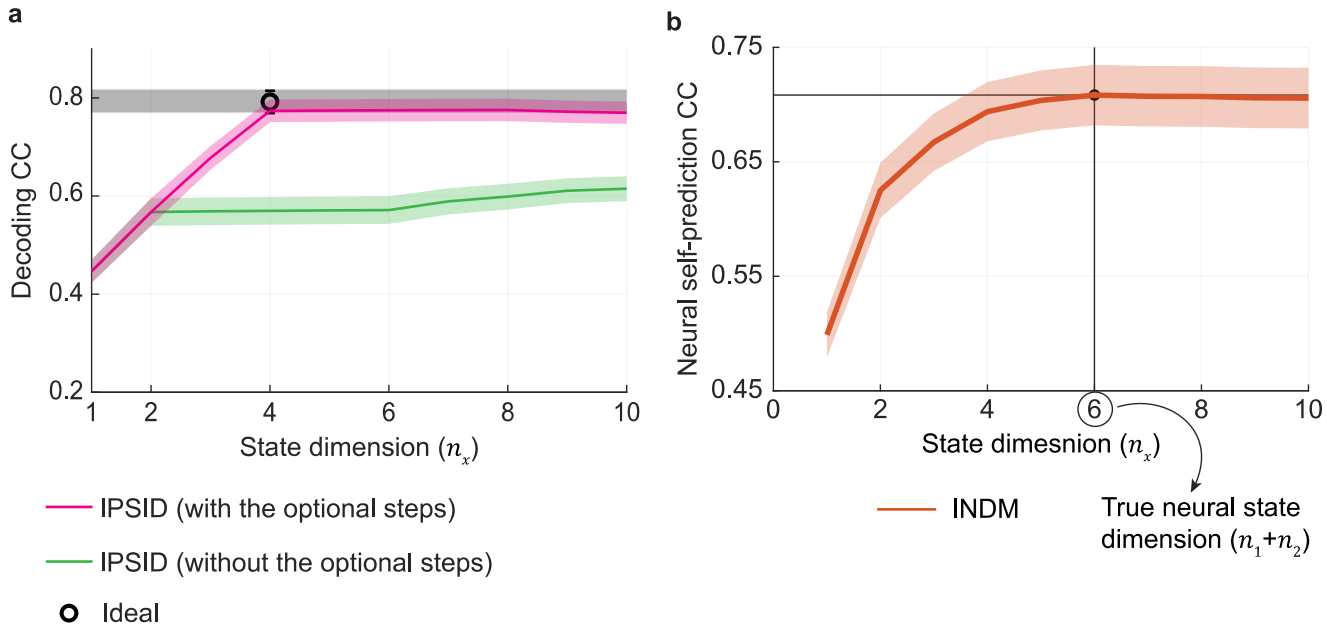

Fig. S8 | Quantified by behavior decoding accuracy, IPSID can achieve ideal prediction of behavior from input and neural activity even in scenarios where the recorded regions do not cover all downstream regions of the input.

(a) Cross-validated behavior decoding correlation coefficient (CC) given 100 random models generated according to equation (2), where some latent states $x_k^{(3)}$ reflect the influence of input on behavior but are not reflected in recorded neural activity y_k. Ideal behavior decoding CC using the true full model in equation (2), which also includes the $x_k^{(3)}$ state components, is shown in black. IPSID can also optionally (magenta) learn and dissociate any dynamics in behavior that are predictable by input but are not encoded in recorded neural activity, i.e., $x_k^{(3)}$. Thus, IPSID is able to achieve peak behavior decoding similar to the ideal model decoding (**Fig. S5c**, **Note S2**). **(b)** Neural self-prediction vs. state dimension for models learned using the second stage of IPSID alone. The latent state dimension that reaches peak self-prediction is equal to the true dimension of neural activity (i.e., $n_1 + n_2$). This procedure of finding the state dimension for peak neural self-prediction is thus used to determine the dimension of $\hat{X}^{(y)}$ in the first optional step of IPSID (see **Fig. S5**). As shown here, this procedure correctly reveals the latent state dimension required for capturing the neural dynamics in $\hat{X}^{(y)}$ because the dimension to reach the peak neural self-prediction in (b) is equal to the true neural state dimension.

1332

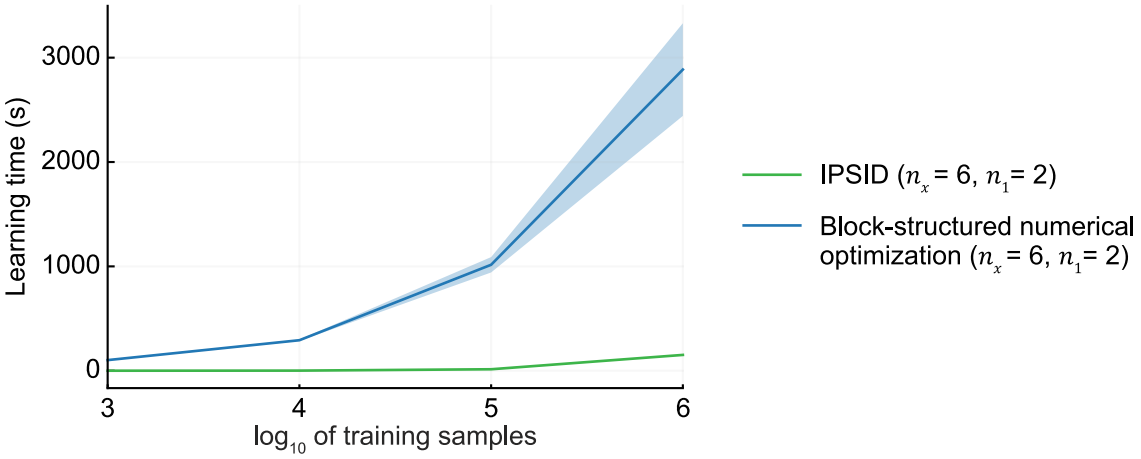

Fig. S9 | IPSID learns the model parameters faster than numerical optimization in Fig. 4.

Learning time for IPSID and block-structured numerical optimization is shown as function of training samples. For all training sample sizes, IPSID is significantly faster than the numerical optimization for learning because it involves a pre-specified set of linear algebraic operations rather than iterative learning via gradient descent. Using different deep learning libraries and adjusting other implementation details could improve the speed of these methods. Nevertheless, because IPSID uses an analytical and non-iterative method, it would likely be generally faster in terms of learning speed compared with iterative numerical optimization approaches, but the exact comparison will depend on various implementation factors and the results here are just for our implementation described in **SI Methods**.

1333

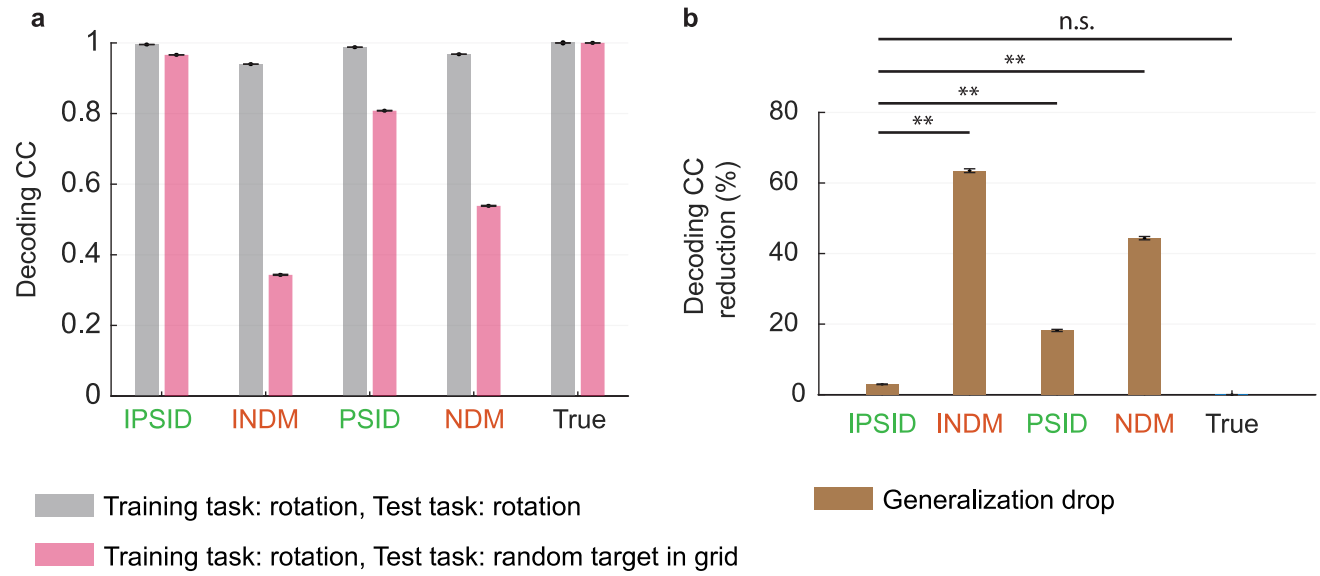

Fig. S10 | Models learned by IPSID generalized across behavioral tasks.

Models were trained using data collected during a rotation task**. (a)** Performance is shown both for test data collected during the same rotation task used in training (task 1 from **Fig. 5**) and for test data collected during a different random target in grid task (task 3 from **Fig. 5**). Performance is averaged across 10 simulations of each task. Bars show the mean and whiskers, which are very short, show the s.e.m. **(b)** Relative drop (%) in behavior decoding correlation coefficient when generalizing a model trained on data from a rotation task to data from a random target in grid task. Double asterisks indicate $P < 0.005$ and n.s. indicates $P > 0.05$ for a one-sided signed-rank test. IPSID is the only method that generalizes well across tasks.

1334

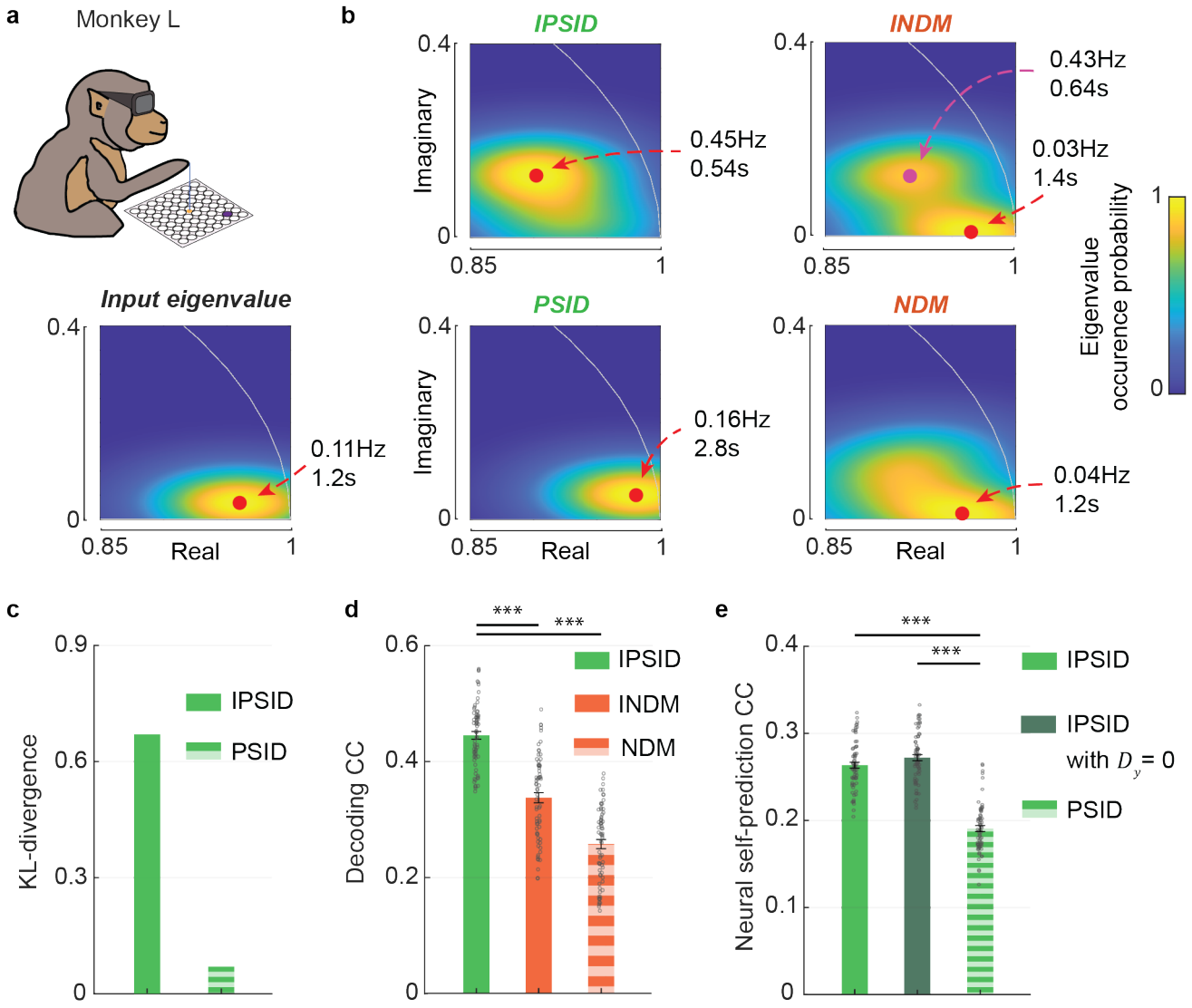

Fig. S11 | In a third monkey, IPSID again uncovers distinct and more accurate intrinsic behaviorally relevant neural dynamics in spiking activity by considering task instructions as inputs to the brain.

Similar to **Fig. 6** for the third subject (monkey L, $n = 70$ cross-validation folds across 2 channel subsets and 7 recording sessions, **SI Methods**). The task is the same as in **Fig. 6** for a different first monkey while the task is different from **Fig. 7** for a different second monkey.

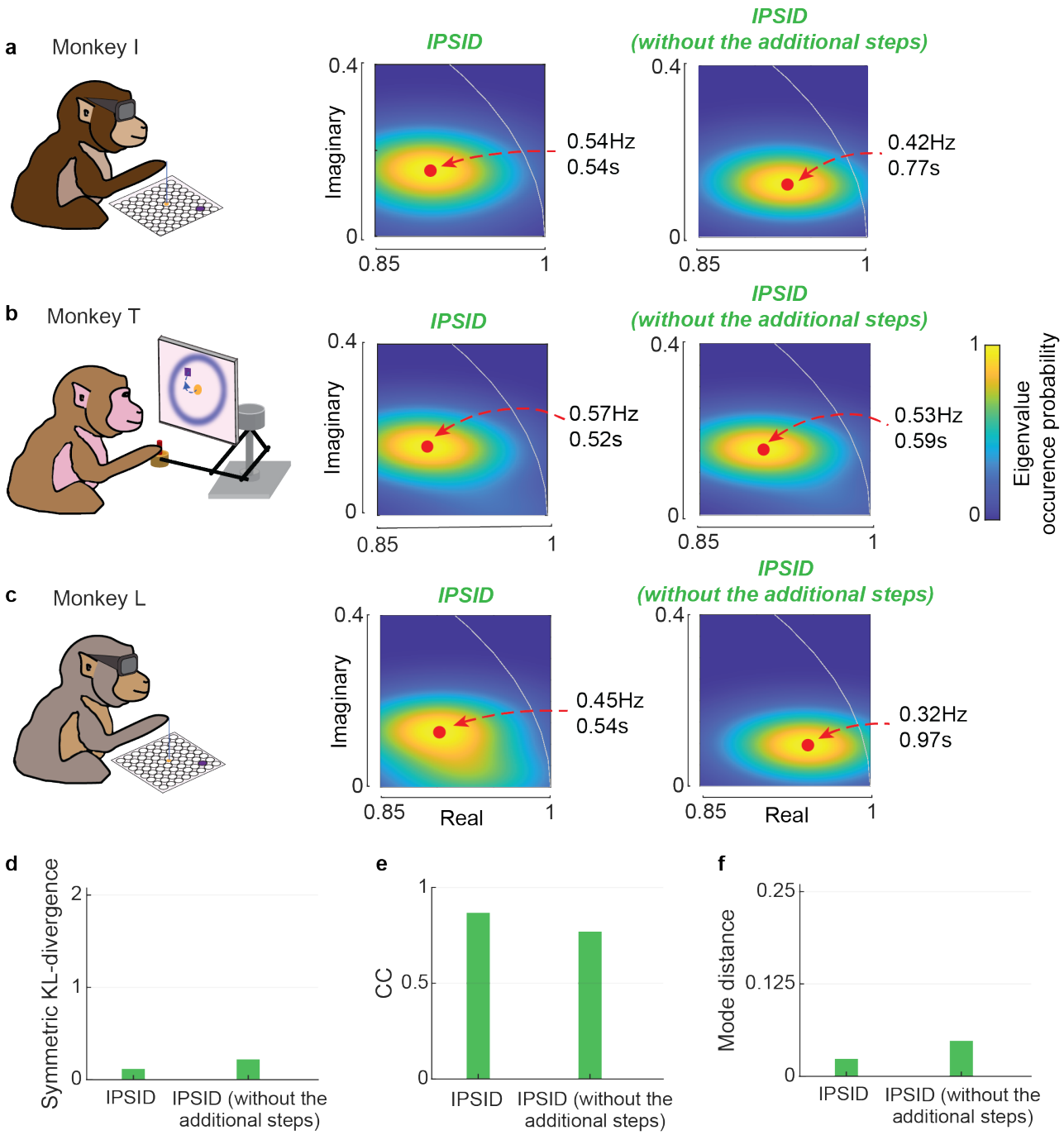

Fig. S12 | Even without the additional steps that ensure the model only include paths from input to behavior that are encoded in the recorded neural activity, IPSID finds largely similar eigenvalues across three monkeys and two tasks from two independent datasets.

(a) Same as **Fig. 6b**, showing the eigenvalues learned for both versions of IPSID, with and without the additional optional steps (**Fig. S5** vs. **Fig. S1**, respectively). **(b-c)** Similar to (a) for the second and third subjects (**Fig. 7, Fig. S11**). Both versions of IPSID largely find similar eigenvalues. Nevertheless, to ensure only paths from input to behavior that are encoded in the recorded neural activity are included in the learned models, IPSID with the additional steps that ensures this property is used in **Figs. 6-8** and **Fig. S11**. **(d-f)** The similarity of the eigenvalues across three monkeys from two independent datasets with two distinct tasks when learned by IPSID with vs. without its additional steps as quantified by the symmetric KL-divergence, correlation coefficient, and mode distance. Notation is as in **Fig. 8d-f**. As quantified by all three metrics, both versions of IPSID find largely consistent eigenvalues across the two monkeys/tasks, with the additional steps helping IPSID reveal the similarity slightly more clearly (compare with the much larger distance/divergence and much smaller CC for INDM in **Fig. 8**).

1335

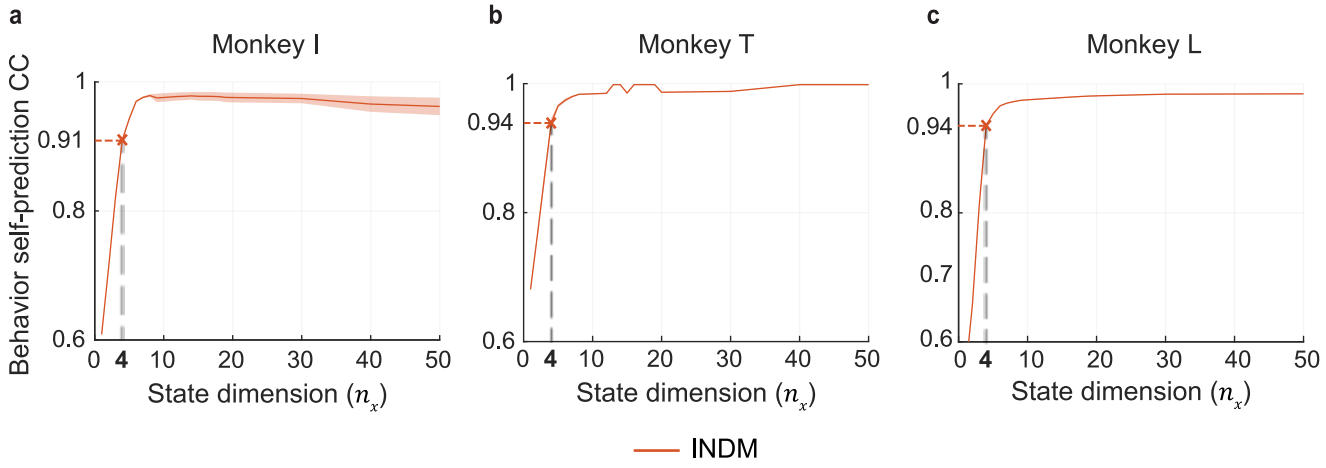

Fig. S13 | Latent state dimension of 4 is sufficient to capture most of the behavior dynamics.

(a) Cross-validated behavior self-prediction versus latent state dimension for the first subject (monkey I) with reaches to random targets on a grid (**Fig. 6a**). Here INDM is applied to behavioral data and the learned model is used to predict the current behavior signal from its past. Using a latent state dimension of 4, behavior self-prediction CC reaches 91% of the ideal value (CC=1). **(b)** Similar to (a) for the second subject (monkey T) with sequential reaches to random targets (**Fig. 7a**). Using a latent state dimension of 4, behavior self-prediction CC reaches 94% of the ideal value (CC=1). **(c)** Similar to (a) for the third subject (monkey L) performing the same task as in (a). Using a latent state dimension of 4, behavior self-prediction CC reaches 94% of the ideal value (CC=1).

1336

Supplementary References

80. Katayama, T. *Subspace Methods for System Identification*. (Springer Science & Business Media, 2006). at <https://link.springer.com/book/10.1007%2F1-84628-158-X>

81. Hardt, M., Ma, T. & Recht, B. Gradient Descent Learns Linear Dynamical Systems. *J. Mach. Learn. Res.* **19,** 1–44 (2018).

82. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y. & Zheng, X. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. (2015). at <https://www.tensorflow.org/>

83. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. Attention is All you Need. in *Adv. Neural Inf. Process. Syst.* **30,** (Curran Associates, Inc., 2017).

84. Särkkä, S. & García-Fernández, Á. F. Temporal Parallelization of Bayesian Smoothers. *IEEE Trans. Autom. Control* **66,** 299–306 (2021).

85. Fu, Z.-F. & He, J. *Modal Analysis*. (Elsevier, 2001).

86. Åström, K. J. & Wittenmark, B. *Computer-Controlled Systems: Theory and Design, Third Edition*. (Courier Corporation, 2013).