Keypoint-MoSeq: parsing behavior by linking point tracking to pose dynamics

Caleb Weinreb[1], Mohammed Abdal Monium Osman[1], Libby Zhang[2,5,6], Sherry Lin[1], Jonah Pearl[1], Sidharth Annapragada[1], Eli Conlin[1], Winthrop F. Gillis[1], Maya Jay[1], Shaokai Ye[3], Alexander Mathis[3], Mackenzie Weygandt Mathis[3], Talmo Pereira[4], Scott W. Linderman[5,6,*] and Sandeep Robert Datta[1,*]

[1]Department of Neurobiology, Harvard Medical School, Boston, MA, USA
[2]Department of Electrical Engineering, Stanford University, Stanford, CA, USA.
[3]Brain Mind and Neuro-X Institute, School of Life Sciences, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland.
[4]Salk Institute for Biological Studies, La Jolla, USA
[5]Wu Tsai Neurosciences Institute, Stanford University, Stanford, CA, USA.
[6]Department of Statistics, Stanford University, Stanford, CA, USA.
*email: scott.linderman@stanford.edu; srdatta@hms.harvard.edu

Abstract

Keypoint tracking algorithms have revolutionized the analysis of animal behavior, enabling investigators to flexibly quantify behavioral dynamics from conventional video recordings obtained in a wide variety of settings. However, it remains unclear how to parse continuous keypoint data into the modules out of which behavior is organized. This challenge is particularly acute because keypoint data is susceptible to high frequency jitter that clustering algorithms can mistake for transitions between behavioral modules. Here we present keypoint-MoSeq, a machine learning-based platform for identifying behavioral modules ("syllables") from keypoint data without human supervision. Keypoint-MoSeq uses a generative model to distinguish keypoint noise from behavior, enabling it to effectively identify syllables whose boundaries correspond to natural sub-second discontinuities inherent to mouse behavior. Keypoint-MoSeq outperforms commonly-used alternative clustering methods at identifying these transitions, at capturing correlations between neural activity and behavior, and at classifying either solitary or social behaviors in accordance with human annotations. Keypoint-MoSeq therefore renders behavioral syllables and grammar accessible to the many researchers who use standard video to capture animal behavior.

Introduction

Work from ethology demonstrates that behavior — a chain of actions traced by the body's movement over time — is both continuous and discrete. Keypoint tracking methods (which including SLEAP[1], DeepLabCut[2] and others[3,4]) enable users to specify and track points corresponding to body parts in videos of behaving animals, and thereby to quantify movement kinematics. These methods are simple to implement and applicable to a wide range of video data; because of their ease of use and generality, keypoint tracking approaches are revolutionizing our access to the continuous dynamics that underlie many aspects of animal behavior in a wide variety of settings[5].

In contrast, it remains less clear how to best cluster behavioral data into the discrete modules of movement that serve as building blocks for more complex patterns of behavior[6-8]. Identifying these modules is essential to the creation of an ethogram, which describes the order in which behavioral modules are expressed in a given context or experiment. While several methods exist that can automatically transform high-dimensional behavioral data into an ethogram[9-14], their underlying logic and assumptions differ, with different methods often giving distinct descriptions of the same behavior[10,13]. An important gap therefore exists between our access to movement kinematics and our ability to understand how these kinematics are organized to impart structure upon behavior; filling this gap is essential if we are to understand how the brain builds complex patterns of action.

One widely deployed and well validated method for identifying behavioral modules and their sequencing is Motion Sequencing (MoSeq)[14]. MoSeq uses unsupervised machine learning methods to transform its inputs — which are not keypoints, but instead data from depth cameras that "see" in three dimensions from a single axis of view — into a set of behavioral motifs (like rears, turns and pauses) called syllables. MoSeq identifies behavioral syllables through a probabilistic generative model that instantiates the ethological hypothesis that behavior is composed of repeatedly used modules of action that are stereotyped in form and placed flexibly into at least somewhat predictable sequences. One important aspect of MoSeq is that it seeks to identify syllables by searching for discontinuities in behavioral data at a timescale that is set by the user; this timescale is specified through a "stickiness" hyperparameter that influences the frequency with which syllables can transition. In the mouse, where MoSeq has been most extensively applied, pervasive discontinuities at the sub-second-to-second timescale mark the boundaries between syllables, and the stickiness hyperparameter is explicitly set to match this timescale. Given a timescale and a depth dataset to analyze, MoSeq automatically identifies the set of syllables out of which behavior is composed in a given experiment without human supervision.

MoSeq-based analysis has been shown to capture meaningful changes in spontaneous, exploratory rodent behaviors induced by genetic mutations, changes in the sensory or physical environment, direct manipulation of neural circuits and pharmacological agents[14-17]. Importantly, MoSeq does not simply provide a useful description of behavior, but also reveals biologically important brain-behavior relationships. For example, the behavioral transitions identified by MoSeq correspond to systematic fluctuations in neural activity in both dopaminergic neurons and their targets in dorsolateral striatum (DLS)[15], and the behavioral syllables identified by MoSeq have explicit neural correlates in DLS spiny projection neurons[16]. Furthermore, dopamine fluctuations in DLS causally influence the use and sequencing of MoSeq-identified syllables over time, and individual syllables can be reinforced (without any alteration in their underlying kinematic content) through closed-loop dopamine manipulations[15].

However, MoSeq has a significant constraint: as currently formulated MoSeq is tailored for input data from depth cameras, which are typically placed over simple behavioral arenas in which single mice are recorded during behavior. Although depth cameras afford a high dimensional view of ongoing pose dynamics, they also suffer from high sensitivity to reflections, limited temporal resolution, and are often difficult to deploy[18]. In principle these limits could be overcome if MoSeq could instead be applied to keypoint data, which can much more flexibly be derived from recordings using standard video cameras. However, attempts to do so have thus far failed, with researchers reporting flickering state sequences that switch much faster than the animal's actual behavior[10,19].

Here we confirm this finding and identify its cause: jitter in the keypoint estimates themselves, which is mistaken by MoSeq for behavioral transitions. To address this challenge, we have reformulated the model underlying MoSeq to simultaneously infer correct pose dynamics (from noisy or even missing data) and the set of expressed behavioral syllables. We benchmark keypoint-MoSeq by comparing its performance on 2D keypoint data to both standard depth camera-based MoSeq and to alternative behavioral clustering methods (including B-SOiD[9], VAME[10] and MotionMapper[20]). We find that keypoint-MoSeq preserves important information about behavioral timing — despite being fed behavioral data that are relatively low dimensional — and identifies similar sets of behavioral transitions as depth MoSeq; furthermore, keypoint-MoSeq outperforms alternative methods at demarcating behavioral transitions in kinematic data, capturing systematic fluctuations in neural activity, and identifying complex features of solitary and social behavior highlighted by expert observers. We also demonstrate that keypoint-MoSeq works on either 2D or 3D keypoints, with increasing dimensionality of the input data yielding richer sets of behavioral syllables.

Our results demonstrate that the ethograms articulated by keypoint-MoSeq effectively identify the natural boundaries present in both neural and behavioral data at the syllable timescale. Given that keypoint tracking can be applied in diverse settings (including natural environments), requires no specialized hardware, and affords direct control over which body parts to track and at what resolution, we anticipate that keypoint-MoSeq will serve as a general tool for understanding the structure of behavior in a wide variety of settings. To facilitate broad adoption of this approach, we have built keypoint-MoSeq to be directly integrated with widely-used keypoint tracking methods (including SLEAP and DeepLabCut), and have made keypoint-MoSeq code freely accessible for academic users at www.MoSeq4all.org; this modular codebase includes novice-friendly Jupyter notebooks to enable users without extensive computational experience to use keypoint-MoSeq, methods for motif visualization in 2D and 3D, a pipeline for post-hoc analysis of the outputs of keypoint-MoSeq, and a hardware-accelerated and parallelization-enabled version of the code for analysis of large datasets.

Results

Simple inspection of depth-based behavioral video data reveals a block-like structure organized at the sub-second timescale[14] (Fig. 1); this observation previously inspired the development of MoSeq, which posits that these blocks encode serially-expressed behavioral syllables. To ask whether keypoint data possess a similar block-like structure, we recorded simultaneous depth and conventional two-dimensional (2D) monochrome videos (using the Microsoft Azure, which has depth and IR-sensitive sensors that operate in parallel to acquire data at 30 Hz) while mice explored an open field arena; we then used a convolutional neural network to track eight keypoints in the 2D video (two ears and six points along the dorsal midline; Fig 1a).

Analysis of the depth videos (independent of MoSeq) revealed the familiar sub-second blocks of smooth behavioral dynamics punctuated by sharp transitions, and applying MoSeq to these videos segmented these blocks into a series of stereotyped behavioral syllables (Fig. 1b). Block-like structure was also apparent in the keypoint data; changepoint analysis (which identifies discontinuities in the underlying data) revealed that block durations were similar for the keypoint data, the depth data, and the syllables identified by MoSeq; furthermore, changepoints in the keypoint data matched both changepoints in the depth data and transitions in behavior identified by MoSeq (Fig 1c-d). This structure is not an accident of camera or keypoint placement, as similar results were obtained when tracking 10 keypoints (including the limbs and ventral midline) using a camera placed below the mouse (Extended Data Fig. 1). The reappearance of a common sub-second organization across depth and keypoint data suggests that this temporal structure is intrinsic to mouse behavior.

MoSeq models behavior as sequence of discrete states, where each state is defined as an autoregressive (AR) trajectory through pose space (corresponding to a syllable), and transitions between states are specified by a modified hidden Markov model (HMM). MoSeq therefore identifies syllables as repeated trajectories through pose space, and transitions between syllables as discontinuities in the pose dynamics. MoSeq includes a stickness hyperparameter that in effect allows it to foveate on a single timescale at which it seeks to explain behavior; this feature enables MoSeq to identify syllables from depth data whose average duration is ~400ms, although there is a broad distribution of mean durations across syllables, and each syllable is associated with its own duration distribution.

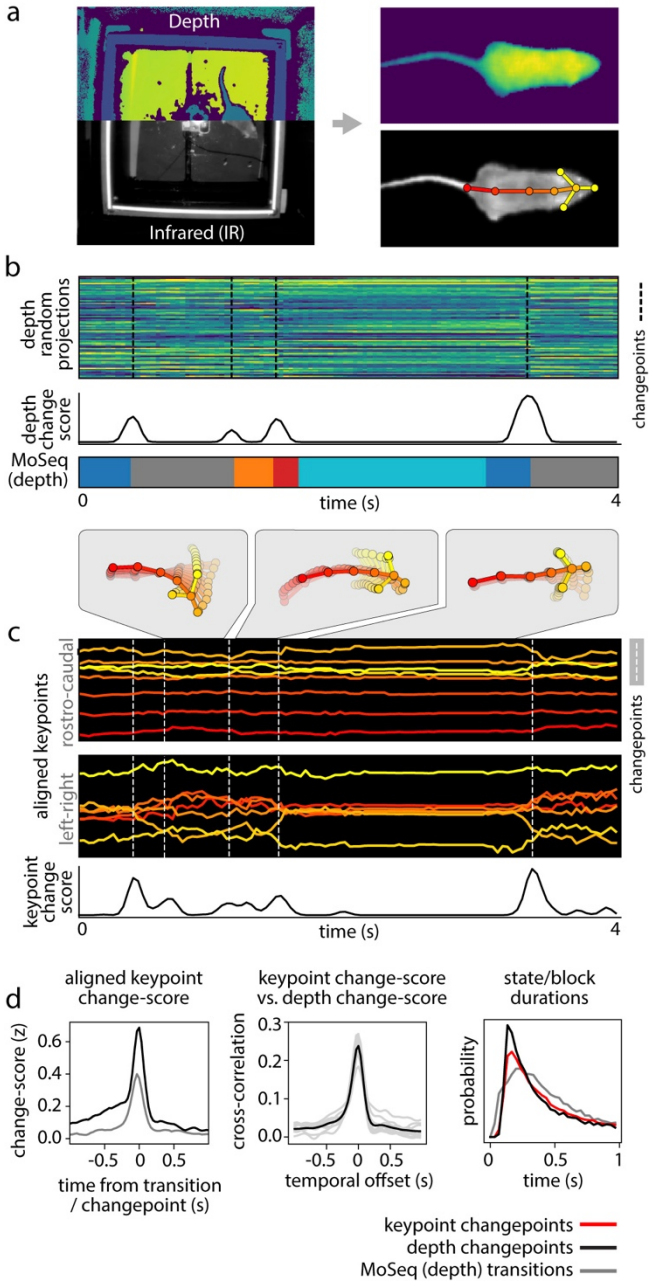


Figure 1: Keypoint trajectories exhibit sub-second to second structure during spontaneous behavior. a) Left: sample frame from simultaneous depth and infrared recordings. **Right:** centered and aligned pose representations featurized by depth (top) or keypoints (bottom). **b-c)** Features extracted from depth or 2D keypoint data within a 4-second window. All rows are temporally aligned. **b) Top:** Representation of the mouse's pose based on depth video. Each row shows a random projection of the high-dimensional depth time-series. Discontinuities in the visual pattern capture abrupt changes in the mouse's movement. **Middle:** Overall rate of change in the depth signal. **Bottom:** color-coded syllable sequence from MoSeq applied to the depth data [referred to as "MoSeq (depth)"]. **c)** Position of each keypoint in egocentric coordinates, plotted above the keypoint change-score. Vertical lines mark changepoints, defined as peaks in the change-score. **d) Left:** average keypoint change-score (z-scored) aligned to MoSeq (depth) transitions (gray), or to changepoints in the depth signal (gray). **Middle:** cross-correlation between depth- and keypoint-change scores, shown for the whole dataset (black line) and for each session (gray lines). **Right:** Distribution of syllable durations, based either on modeling or changepoint analysis.

However, when applied to keypoint data, MoSeq failed to identify syllables at this characteristic ~400ms timescale, instead producing a set of brief syllables whose durations were often just one or two frames, and a prominent tail of aberrently long syllables that merged multiple behaviors; futhermore, the transitions between these syllables aligned poorly to changepoints derived from the keypoint data (Fig. 2a-b). These observations are consistent with prior work demonstrating that feeding keypoints to MoSeq generates behavioral representations that are less informative than those generated by alternative clustering methods[10,19].

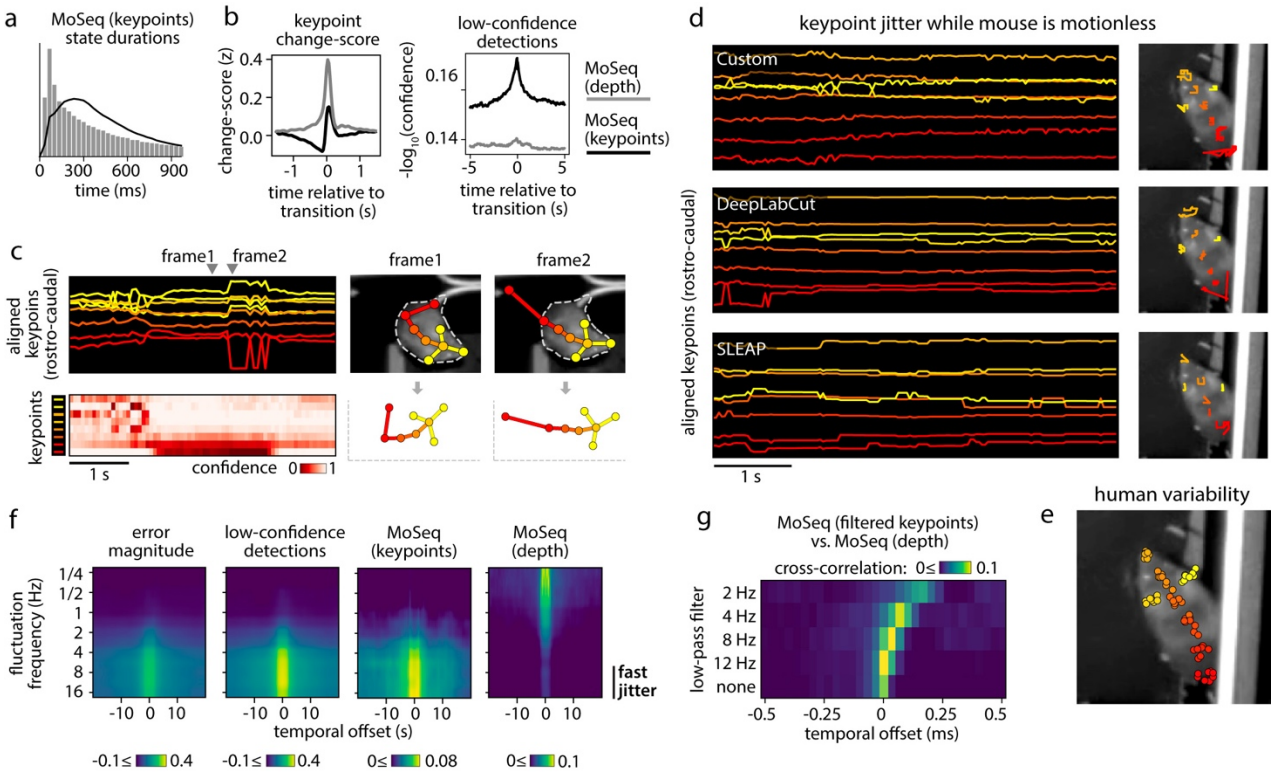


Figure 2: Keypoint tracking noise challenges syllable inference . a) Applying MoSeq to keypoint trajectories [referred to as "MoSeq (keypoints)"] produces abnormally brief syllables when compared to MoSeq applied to depth data ["MoSeq (depth)"]. **b)** Z-scored keypoint change-score (left) and a low-confidence detection score (right) relative to MoSeq transitions derived from either keypoints or depth. The low-confidence score is computed from neural network confidences on each frame as the mean of $-\log_{10}(\text{confidence}_k)$ across keypoints k. **c) Left:** example of keypoint detection errors, including high-frequency fluctuations in keypoint coordinates (top row) that coincide with low neural network confidence (bottom row). **Right:** detected keypoint coordinates before the error (frame1) and during the error (frame2). Displacement of the tail-base keypoint causes a shift in egocentric alignment, leading to coordinate changes across the other keypoints. **d)** Example of keypoint jitter from three different keypoint tracking methods over a 5-second interval during which the mouse was motionless. **Left:** egocentrically aligned keypoint trajectories. **Right:** path traced by each keypoint during the 5-second interval. **e)** Variability across eight human labelers. **f)** Cross-correlations with keypoint fluctuations at a range of frequencies. Each heatmap represents a different time-series (see Methods section "Spectral Analysis" for detailed descriptions). **g)** Cross-correlation of transitions rates, comparing MoSeq (depth) and MoSeq applied to keypoints with various levels of smoothing by a low-pass filter. Transition rate is defined as the posterior probability of a transition occurring on each frame.

We wondered whether the poor performance of MoSeq could be explained by noise in the keypoint data, which in principle could introduce subtle discontinuities that are falsely recognized by MoSeq as behavioral transitions. Indeed, mouse keypoint data exhibited high-frequency (>8Hz) jitter in position regardless of whether we tracked keypoints with our custom neural network or commonly used platforms like DeepLabCut

(DLC) and SLEAP (Fig. 2c-d, see Methods). Inspection of videos revealed that high frequency keypoint jitter is often associated with local tracking errors or rapid switching in the inferred location of an ambiguously positioned keypoint, rather than discernable changes in pose (Fig 2d, Extended Data Fig. 2a). Indeed, frame-to-frame fluctuations in the keypoints had a similar scale as the variability in human labeling (Fig 2e, Extended Data Fig. 2b). We confirmed that keypoint flicker was unrelated to true movement by tracking the same body part using multiple cameras; though overall movement trajectories were almost identical, the high-frequency fluctuations around these trajectories were uncorrelated across cameras (Extended Data Fig. 2c-d). Consistent with the possibility that keypoint noise dominates MoSeq's view of behavior, syllable transitions derived from keypoints – but not depth – frequently overlapped with jitter and low-confidence estimates of keypoint position (Fig. 2f). Though one might imagine that simple smoothing could ameliorate this problem, application of a low-pass filter had the additional consequence of blurring actual transitions, preventing MoSeq from identifying syllable boundaries (Fig 2g). Median filtering and Gaussian smoothing similarly yielded no improvement (Extended Data Fig 2e). These data reveal that high-frequency tracking noise can be pervasive across point-tracking algorithms and demonstrate that this noise impedes the ability of MoSeq to accurately segment behavior.

Hierarchical modeling of keypoint trajectories decouples noise from behavior

MoSeq syllables reflect keypoint jitter because MoSeq assumes that each keypoint is a faithful and accurate representation of the position of a point on the animal. We therefore sought an alternative approach that could treat the keypoints as noisy observations rather than the truth. Switching linear dynamical systems (SLDS), which extend the AR-HMM model that underlies MoSeq, offer a principled way to decouple keypoint noise from behavior[21,22]. We therefore formulated an SLDS-based version of MoSeq whose architecture enables joint inference of pose and syllable structure. This new SLDS model has three hierarchical levels (Fig. 3a): a discrete state sequence (top level) that governs the dynamics of keypoint trajectories in a low-dimensional pose space (middle level), which is then projected into the keypoint space itself (bottom level). The three levels of this model therefore correspond to syllables, pose states, and keypoint coordinates respectively.

We further adapted the SLDS model to keypoint data by adding three additional variables: centroid and heading (which capture the animal's overall position in allocentric coordinates) and a noise metric for each keypoint in each frame[23]. When fit to data, the SLDS model estimates for each frame the animal's location and pose, as well as the identity and content of the current behavioral syllable (Fig. 3b). Because of its structure, when a single keypoint implausibly jumps from one location to another, the

SLDS model can attribute the sudden displacement to noise and preserve a smooth pose trajectory; if all the keypoints suddenly rotate within the egocentric reference frame, the model can adjust the inferred heading for that frame and restore a plausible sequence of coordinates. Since in the special case of zero keypoint noise our new model reduces to the same AR-HMM used in depth MoSeq[14], we refer to this new method as “keypoint-MoSeq” for the remainder of the paper.

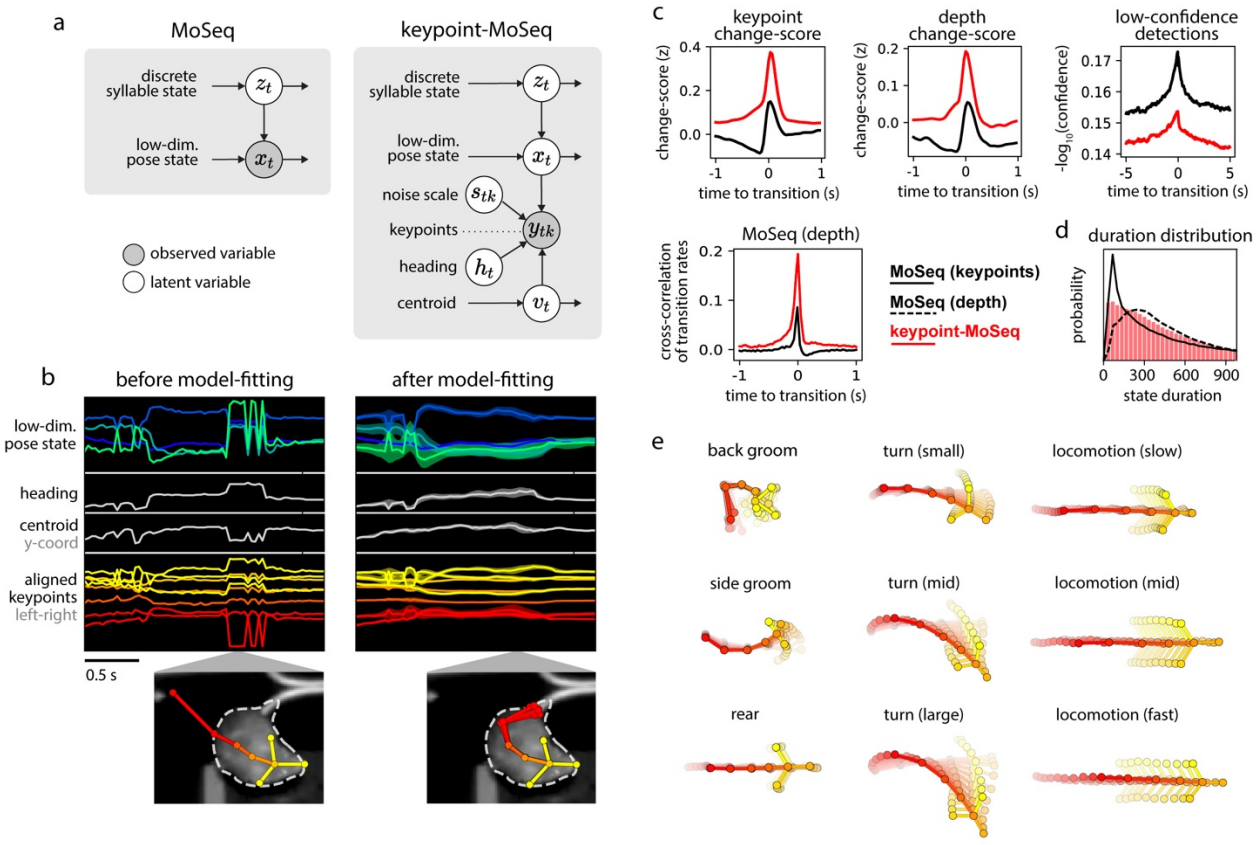


Figure 3: Hierarchical modeling of keypoint trajectories decouples noise from pose dynamics. a) Graphical models illustrating MoSeq and a novel hierarchical model called “keypoint-MoSeq”. In both models, a discrete syllable sequence governs the dynamics of a low-dimensional pose state. The pose state is either fixed using PCA (as in “MoSeq”, left) or inferred from keypoint observations in conjunction with the animal’s centroid and heading, as well as a noise scale that discounts keypoint detection errors (as in “keypoint-MoSeq”, right). b) Example of error-correction by keypoint-MoSeq. **Left:** Before fitting, all variables are perturbed by displacement of the tail-base keypoint, in the callout. **Right:** Keypoint-MoSeq infers plausible trajectories for each variable. Shading indicates uncertainty in the model posterior (95% confidence). The callout shows likely keypoint coordinates inferred by the model. **c)** Average trajectory of features aligned to transitions from each modeling approach. **d)** Durations distribution of syllables from each model. **e)** Average pose trajectories for example keypoint-MoSeq syllables. Each trajectory includes ten evenly timed poses from 165ms before to 500ms after syllable onset.

Unlike traditional MoSeq, keypoint-MoSeq appeared to effectively identify behavioral syllables rather than noise in the keypoint data. State transitions identified by keypoint-MoSeq overlapped with transitions in the raw depth data, with depth MoSeq-derived syllable transitions, and with transitions in the keypoints as identified by changepoint analysis; syllable boundaries identified by keypoint-MoSeq also overlapped less with low-confidence neural network detections for individual keypoints (Fig. 3c). Furthermore, the duration distribution of syllables identified by keypoint-MoSeq more closely matched that generated by conventional MoSeq using depth data (Fig 3d, Extended Data Fig 3a). From a modeling perspective the output of MoSeq was sensible: cross-likelihood analysis revealed that keypoint-based syllables were mathematically distinct trajectories in pose space, and submitting synthetic keypoint data that lacked any underlying block structure resulted in keypoint-MoSeq models that failed to identify distinct syllables (Extended Data Fig 3b,c). These analyses suggest that keypoint-MoSeq effectively addresses the syllable switching problem, nominating it as a candidate for parsing keypoint data obtained from conventional 2D cameras into syllables.

For our open field data, keypoint-MoSeq identified 25 syllables (Extended Data Fig 3d). Inspection of movies depicting multiple instances of the same syllable revealed that each syllable was a distinct, stereotyped motif of behavior that could be easily labeled by human observers. Keypoint-MoSeq differentiated between categories of behavior (e.g., rearing, grooming, walking), and variations within each category (e.g., turn angle, speed) (Fig 3e). Importantly, keypoint-MoSeq preserves access to the kinematic and morphological parameters that underlie each behavioral syllable (Extended Data Fig 3e), thereby enabling explicit comparisons and analysis. These data demonstrate that keypoint-MoSeq provides an interpretable segmentation of behavior captured by standard 2D videos, which are used in most behavioral neuroscience experiments.

Keypoint-MoSeq better captures the fast temporal structure of behavior than alternative behavioral clustering methods

We wished to validate keypoint-MoSeq by demonstrating that it generates the kinds of outputs that would be predicted for a time-series model of behavior, and by showing that this output is useful for addressing questions typically posed by users of unsupervised methods in behavioral classification. As part of this validation process, we compared keypoint-MoSeq to alternative unsupervised methods for clustering keypoints, reasoning that this comparison might highlight strengths and weaknesses that are particular to each method. Such alternative methods include VAME, MotionMapper and B-SOiD, all of which first transform keypoint data into a feature

space that reflects the dynamics in a small window around each frame, and then cluster those features to distinguish a set of behavioral states[9,10,20,24].

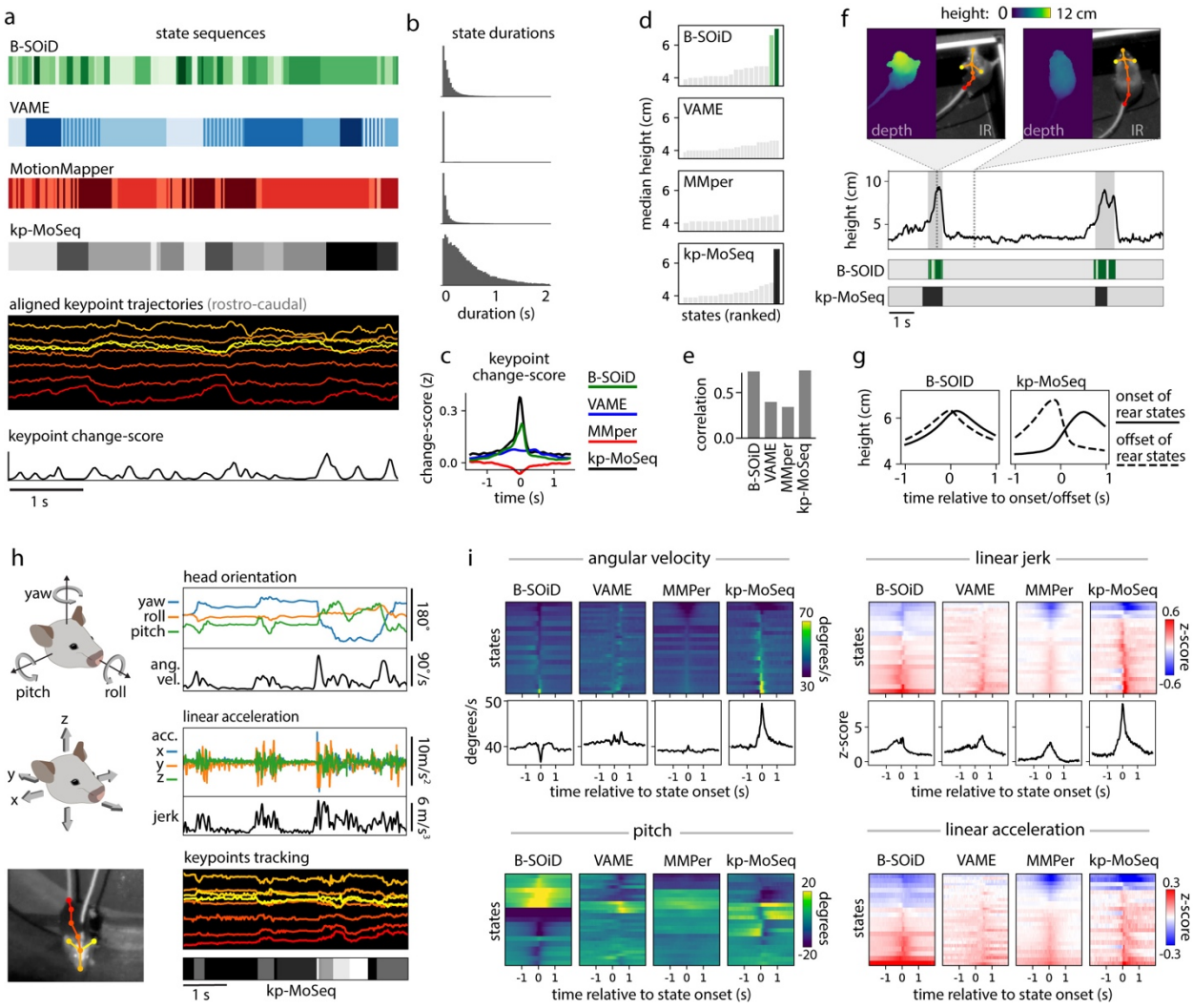


Figure 4: Keypoint-MoSeq captures the temporal structure of behavior. a) Example behavioral segmentations from four methods applied to the same 2D keypoint dataset. Keypoint-MoSeq transitions (fourth row) are sparser than those from other methods and more closely aligned with peaks in the keypoint change-score (bottom row). **b)** Distribution of state durations for each method in (a). **c)** Average keypoint change-score (z-scored) relative to transitions from each method ("MMper" refers to MotionMapper). **d)** Median mouse height (measured by depth camera) for each unsupervised behavior state. Rear-specific states (shaded bars) are defined as those with median height > 6cm. **e)** Accuracy of mouse-height decoding models that were fit to state sequences from each method. **f) Bottom:** state sequences from keypoint-MoSeq and B-SOiD during a pair of rears. States are colored as in (d). **Top:** mouse height over time with rears shaded gray. Callouts show depth- and IR-views of the mouse at two example frames. **g)** Average mouse height aligned to the onsets (solid line) or offsets (dashed line) of rear-specific states defined in (d). **h)** Signals captured

from a head-mounted inertial measurement unit (IMU), including absolute 3D head-orientation (top) and relative linear acceleration (bottom). Each signal and its rate of change, including angular velocity (ang. vel.) and jerk (the derivative of acceleration), is plotted during a five second interval. **i)** IMU signals aligned to the onsets of each behavioral state. Each heatmap row represents a state. Line plots show the median across states for angular velocity and jerk.

.

As mentioned above, by design MoSeq identifies boundaries between behavioral syllables that correspond to abrupt transitions in the keypoint or depth data. To ask whether these alternative methods identify similar boundaries between discrete behaviors, we applied them to the identical 2D keypoint dataset. Behavioral states from VAME, B-SOiD and MotionMapper were usually brief (median duration 33-100ms, compared to ~400ms for keypoint-MoSeq) and their transitions aligned significantly less closely with changepoints in keypoint data than did syllable transitions identified by keypoint-MoSeq (Fig 4a-c). To ensure these results were the consequence of the methods themselves rather than specific parameters we chose, we performed a comprehensive parameter scan for all methods, including up to an order of magnitude dilation of the temporal windows used by B-SOiD and MotionMapper, as well as scans over latent dimension, state number, clustering mode, and preprocessing options across all methods (where applicable); this analysis revealed some parameter combinations that yielded longer state durations, but these combinations tended to have a similar or worse alignment to changepoints in the keypoint data, a finding we replicated for both overhead and bottom-up camera angles (Extended Data Figure 4a).

Rearing affords a particularly clear example of the differences between unsupervised behavioral methods with respect to time. B-SOiD and keypoint-MoSeq both learned a specific set of rear states/syllables (Fig 4d; no rear-specific states were identified by VAME or MotionMapper) and each encoded the mouse's height with comparable accuracy (B-SOiD: R=0.73, keypoint-MoSeq: R=0.74 for correlation between predicted and true mouse height; Fig 4e). Yet the rear states from each method differed dramatically in their dynamics. Whereas keypoint-MoSeq typically detected two syllable transitions that surrounded each rearing behavior (one entering the rearing syllable, the second exiting the rearing syllable), B-SOiD typically detected five to ten different transitions during the execution of a single rear, including switches between distinct rear states as well as flickering between rear- and non-rear-states (Fig 4f; Extended Data Fig 4b). This difference was made further apparent when we aligned mouse height to rearing states identified by the different methods (Fig 4g). Mouse height increased at transitions into keypoint-MoSeq's rear state and fell at transitions out of it, producing a pair of height trajectories into and out of the rearing syllable that differed from each other and were asymmetric in time. In contrast, height tended to peak at transitions into and out of B-SOiD's rear states, with a temporally symmetric

trajectory that was only slightly different for ingoing versus outgoing transitions; this observation suggests that — at least in this example — B-SOiD does not effectively identify the boundaries between syllables, but instead tends to fragment sub-second behaviors throughout their execution.

The observation that keypoint-MoSeq effectively identifies behavioral boundaries has so far relied exclusively on analysis of video data. We therefore sought to validate keypoint-MoSeq and compare it to other unsupervised behavioral algorithms using a more direct measure of movement kinematics. To carefully address this issue, we asked about the relationship between algorithm-identified behavioral transitions and behavioral changepoints identified by head-mounted inertial measurement units (IMUs), which allow us to capture precise 3D head orientation and linear acceleration while we record mice exploring an open field arena using an overhead 2D camera (Fig 4h). Behavioral transitions were identifiable in the IMU data as transient increases in the rates of change for acceleration (quantified by jerk) and orientation (quantified by angular velocity). Both measures correlated with state transitions identified by keypoint-MoSeq but failed to match transitions in behavioral states identified by B-SOiD, MotionMapper and VAME (Fig. 4i). Furthermore, IMU-extracted behavioral features (like head pitch or acceleration) typically rose and fell symmetrically around B-SOiD, MotionMapper and VAME-identified transitions, while keypoint-MoSeq identified asymmetrical changes in these features. For example, acceleration tended to be highest in the middle of B-SOiD-identified behavioral states, while acceleration tended to sharply change at the boundaries of keypoint-MoSeq-identified behavioral syllables (Fig 4i; Extended Data Fig 5a-b).

The fact that keypoint-MoSeq more clearly identifies behavioral boundaries does not necessarily mean that it is better at capturing the instantaneous content of behavior. Indeed, a spline-based linear encoding model was able to effectively reconstruct a panel of coarse kinematic parameters from all four of the explored methods with comparable accuracy (Extended Data Fig 4c). However, the fact that movement parameters – as measured by accelerometry – change suddenly at the onset of keypoint-MoSeq syllables, but not at the onset of B-SOiD, VAME or MotionMapper states, provide additional evidence that these methods afford fundamentally different views of temporal structure in behavior. The coincidence of behavioral transitions identified by keypoint-MoSeq (which are ultimately based on video data) and IMU data (which is based in movement per se) further validates the segmentation of behavior generated by keypoint-MoSeq.

Keypoint-MoSeq state transitions align with fluctuations in neural data

Understanding the relationship between brain and behavior requires timestamps that enable researchers to align neural and behavioral data to moments of change. During traditional head-fixed behavioral tasks, such timestamps naturally arise out of task structure, in which time is divided up into clear, experimenter-specified epochs relating to e.g., the presentation of sensory cues or reward, the moment of behavioral report, etc. One of the main use cases for unsupervised behavioral classification is to understand how the brain generates spontaneous behaviors that arise outside of a rigid task structure[6]; in this setting, the boundaries between behavioral states serve as surrogate timestamps to allow alignment of neural data.

We have recently used depth MoSeq to show that the levels of the neuromodulator dopamine fluctuate within the dorsolateral striatum (DLS) during spontaneous behavior, and that these fluctuations are temporally aligned to syllable transitions[15]: On average, dopamine levels rise rapidly at the onset of each syllable, and then decline toward the end of the syllable. Furthermore, the average magnitude of dopamine fluctuations varies across syllables. We wondered whether we could recapitulate these previously observed relationships between syllable transitions and dopamine fluctuations using keypoint-MoSeq or alternative methods for fractionating keypoint data into behavioral states (Fig 5a).

Syllable-associated dopamine fluctuations (as captured by dLight photometry) were remarkably similar between depth MoSeq and keypoint-MoSeq; aligning the derivative of the dopamine signal to syllable transitions revealed a trajectory that was almost identical between depth MoSeq and keypoint-MoSeq, with a shallow dip prior to syllable onset and sharp rise after onset (Fig 5b). State-related dopamine fluctuations were much lower in amplitude (or non-existent), however, when assessed using B-SOiD, VAME and MotionMapper (Fig 5b). Given the association between striatal dopamine release and movement[25], it is possible that method-to-method variation can be explained by differences in how each method represents stationary vs. locomotory behavior. Yet, the transition-associated dopamine fluctuations highlighted by keypoint-MoSeq remained much more prominent than those from other methods when analysis was restricted to high or low velocity states (Extended Data Fig 6a).

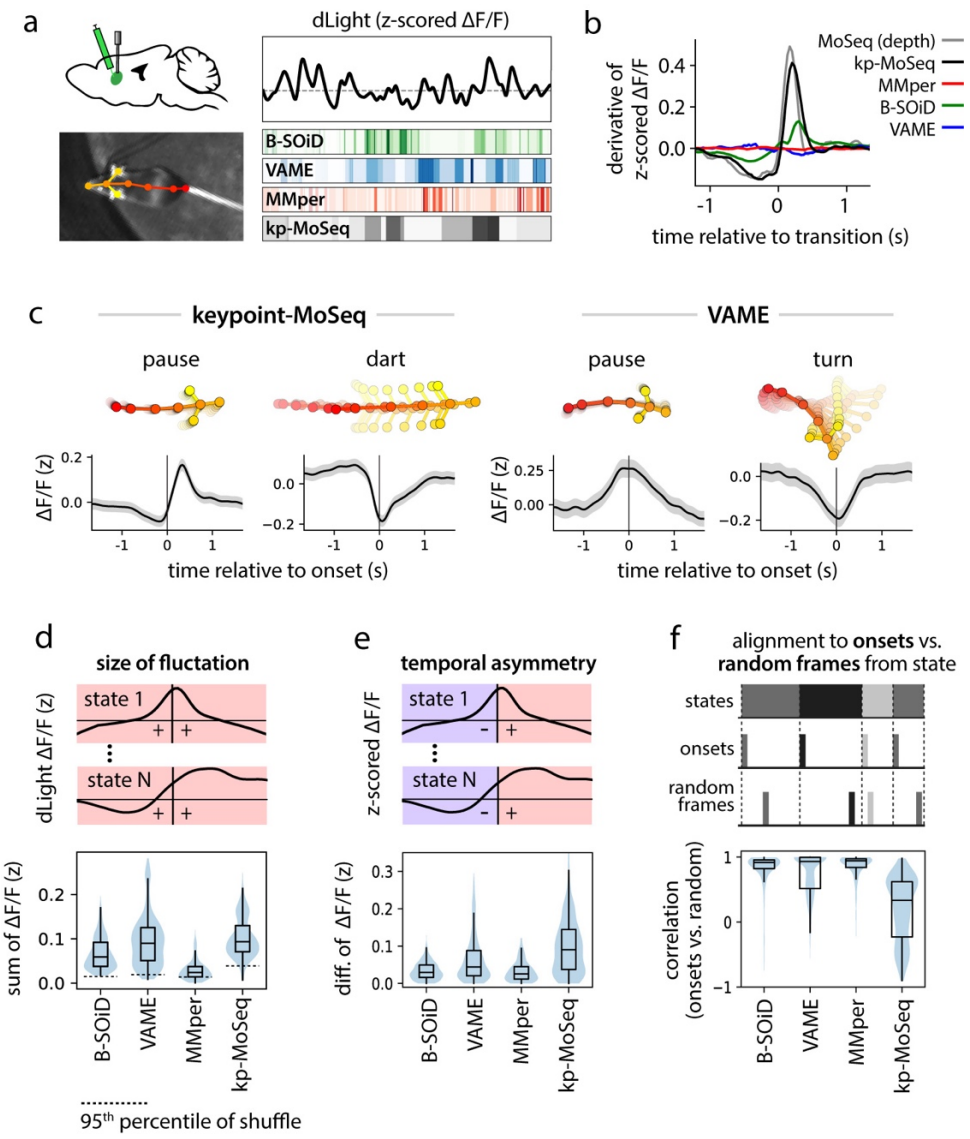


Figure 5: Keypoint-MoSeq syllable transitions align with fluctuations in striatal dopamine. a) Neural-behavioral dataset, including dopamine fluctuations in the dorsolateral striatum (DLS) obtained from fiber photometry (top) and unsupervised behavioral segmentations of 2D keypoint data (bottom). **b)** Derivative of the dopamine signal aligned to state transitions from each method. **c)** Average dopamine signal (z-scored ΔF/F) aligned to the onset of example states identified by keypoint-MoSeq and VAME. Shading marks the 95% confidence interval around the mean. **d)** Distributions capturing the magnitude of state-associated dopamine fluctuations across states from each method, where magnitude is defined as mean total absolute value in a one-second window centered on state onset. **e)** Distributions capturing the temporal asymmetry of state-associated dopamine fluctuations, where asymmetry is defined as the difference in mean dopamine signal during 500ms after versus 500ms before state onset. **f) Top:** schematic of randomization. The dopamine signal was either aligned to the onsets of each state, as in (c), or to random frames throughout the execution of each state. **Right:** distributions capturing the correlation of state-associated dopamine fluctuations before vs. after randomization.

We wondered whether the inability of alternative clustering methods to identify a clear relationship between behavior and dopamine could be explained by differences in how they represent the temporal structure of behavior. If, as we have shown, B-SOiD, VAME and MotionMapper can capture the content of behavior but not the timing of transitions, then one might expect average dopamine levels to vary consistently across the different behavioral states identified by these alternative methods. To test this prediction, we computed the average dopamine trace aligned to state onset separately for each state (Fig 5c). Across all methods almost every state was associated with a consistent average increase or decrease in dopamine levels (Fig 5c-d, Extended Data Fig 6b).

However, the specific pattern of fluctuation identified by each method substantially varied. Dopamine tended to increase at the initiation of keypoint-MoSeq-identified behavioral syllables, with dopamine baselines and amplitudes varying across syllables. In contrast, dopamine signals were typically at a peak or nadir at the beginning of each state identified by alternative methods, forming a trajectory that was symmetric around state onset (Fig 5c). This symmetry tended to wash out dopamine dynamics, with the average change in the dopamine signal being approximately three times larger for keypoint-MoSeq than for alternative methods (Fig 5e). Similarly, the number of states where the dopamine signal changed sign before vs. after state onset was ~2-fold greater for keypoint-MoSeq than for alternatives. Furthermore, aligning the dopamine signal to randomly-sampled times throughout the execution of each behavioral state – rather than its onset – radically altered the state-associated dopamine dynamics observed using keypoint-MoSeq, but made little difference for alternative methods (Fig 5f, Extended Data Fig 6c-d), a result that could not be explained simply by differences in each state's duration (Extended Data Fig 6c). These results suggest that the onsets of keypoint-MoSeq-identified behavioral syllables are meaningful landmarks for neural data analysis, while state onsets identified by alternative methods are often functionally indistinguishable from timepoints randomly chosen from throughout the duration of a behavior.

Keypoint-MoSeq generalizes across pose representations and behaviors

Keypoint tracking is a powerful means of pose estimation because it is so general: available methods can be flexibly applied to a wide variety of experimental setups, can capture diverse behaviors, and afford the experimenter broad latitude in the choice of which parts to track and at what resolution. To test the ability of keypoint-MoSeq to generalize across laboratories — and to better understand the mapping

between syllables and human-identified behaviors — we used keypoint-MoSeq and alternative methods to analyze a pair of published benchmark datasets[26,27]. The first dataset included conventional 2D videos of a single mouse behaving in an open field, with human annotations for four commonly occurring behaviors (locomote, rear, face groom and body groom) (Fig 6a-c). To identify keypoints in this dataset we used DeepLabCut, specifically the TopViewMouse SuperAnimal network from the DLC Model Zoo[28], which automatically identifies keypoints without the need for annotation data or training. The second dataset (part of the CalMS21 benchmark[27]) included a set of three manually annotated social behaviors (mounting, investigation, and attack) as well as keypoints for a pair of interacting mice (Fig 6d-f).

Changepoints analysis of keypoint data from both datasets identified block-like structure whose mean duration was ~400ms, consistent with the presence of a behavioral rhythm organized at the sub-second timescale (Extended Data Fig 7a-b). Consistent with this, Keypoint-MoSeq recovered syllables from both datasets whose average duration was ~400ms while, as before, the B-SOiD, MotionMapper and VAME identified behavioral states that were much shorter (Extended Data Fig 7c-d). Keypoint-MoSeq was also better at conveying information about which human-identified behavioral states were occuring at each moment than alternative methods; that said, the different methods were not dramatically different in terms of quantitative performance, consistent with each doing a reasonable job of capturing broad information about behavior (Fig 6c,f, Extended Data Fig 7e-f). However, there were some important differences: in the CalMS21 dataset, for example, neither VAME nor B-SOiD ever defined an attack-specific state, and VAME only defined an investigation-specific state in 53% of model fits. Keypoint-MoSeq, in contrast, defined at least one state specific to each of these behaviors in 100% of model fits (Extended Data Fig 7g). These results demonstrate that keypoint-MoSeq can identify temporal structure in diverse 2D keypoint datasets and reveal consistency between keypoint-MoSeq and supervised labels for behavioral states.

Finally, we noted that the number of syllables identified in our open field data by keypoint-MoSeq using 2D keypoints (25) was substantially fewer than the number identified by depth MoSeq (52). Furthermore, the most rarely used syllables identified by depth MoSeq were used relatively more than the most rarely used syllables identified by keypoint-MoSeq (i.e., the distribution describing how often each syllable is used during an experiment is right shifted for depth data). These findings suggest that the higher dimensionality of the depth data (relative to the 8 keypoints identified in the 2D data) affords MoSeq more information about pose during spontaneous behavior, which in turn yields a richer behavioral description. To test this hypothesis, we used multiple cameras to estimate the position of keypoints in 3D (including 6 keypoints that were not

visible in the overhead camera 2D dataset) (Fig 6g). Compared to the 2D data, the new 3D keypoint pose representation was higher dimensional, had smoother trajectories and exhibited oscillatory dynamics related to gait (Extended Data Fig 8a-b). Yet the temporal structure of both the data and the syllables that emerged from keypoint-MoSeq was surprisingly similar: the 3D data contained similar changepoints to both the 2D and depth data (Extended Data 8c-d), and after processing with keypoint-MoSeq the resulting syllable duration distributions were almost identical between the 2D, 3D and depth datasets, and syllable transitions tended to occur at the same moments in time (Fig 6h).

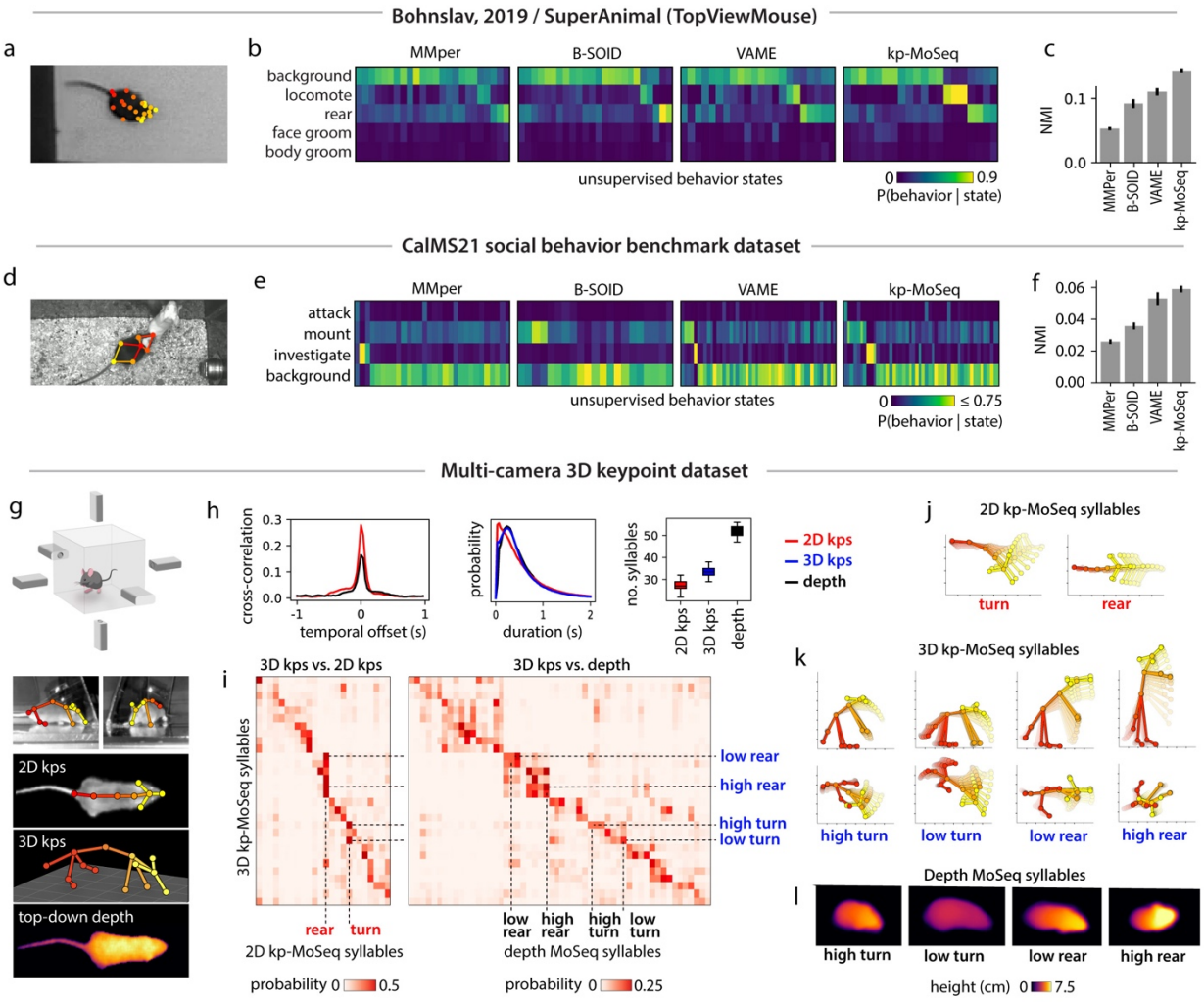


Figure 6: Keypoint-MoSeq generalizes across pose representations and behaviors. a) Example frame from a benchmark open field dataset (Bohnslav, 2019). **b)** Frequency of human-annotated behaviors during states inferred from unsupervised analysis of 2D keypoints. **c)** Normalized mutual information (NMI) between human annotations and unsupervised behavior labels from each method. **d)** Example frame from the CalMS21 social behavior benchmark dataset, showing 2D

keypoint annotations for the resident mouse. **e-f)** Overlap between human annotations and unsupervised behavior states inferred from 2D keypoint tracking of the resident mouse. **g)** Multi-camera arena for simultaneous recording of 3D keypoints (3D kps), 2D keypoints (2D kps) and depth videos. **h)** Comparison of model outputs across tracking modalities. 2D and 3D keypoint data were modeled using keypoint-MoSeq, and depth data were modeled using original MoSeq. **Left:** cross correlation of transition rates, comparing 3D keypoints to 2D keypoints and depth respectively. **Middle:** distribution of syllable durations; **Right:** number of states with frequency > 0.5%. Boxplots represent the distribution of state counts across 20 independent runs of each model. **i)** Propability of syllables inferred from 2D keypoints (left) or depth (right) during each 3D keypoint-based syllable. **j-l)** Average pose trajectories for the syllables marked in (i). **k)** 3D trajectories are plotted in side view (first row) and top-down view (second row). **l)** Average pose (as depth image) 100ms after syllable onset.

There was a bigger change, however, in the way syllables were categorized when comparing 2D and 3D data. Keypoint-MoSeq tended to distinguish more syllable states in the 3D data (52±3 syllables for depth MoSeq, 33±2 syllables for 3D keypoints vs. 27±2 syllables for 2D keypoints; Fig 6h), especially for behaviors that varied in the mouse's height (Fig 6i). Turning, for example, was grouped as a single state with the 2D keypoint data but partitioned into three states with different head positions with the 3D keypoint data (nose to the ground vs. nose in the air), and seven different states in the depth data (Fig 6j-l). Rearing was even more fractionated, with a single 2D syllable splitting six ways based on body angle and trajectory in the 3D keypoint data (rising vs. falling) and 8 ways in the depth data. These data demonstrate that keypoint-MoSeq works well on both 2D and 3D keypoint data; furthermore, our analyses suggest that higher-dimensional sources of input data to MoSeq give rise to richer descriptions of behavior, but that even relatively low-dimensional 2D keypoint data can be used to usefully identify behavioral transitions.

Discussion

MoSeq is a well-validated method for behavioral segmentation that leverages natural sub-second discontinuities in rodent behavior to automatically identify the behavioral syllables out of which spontaneous behavior is assembled[14-17]. However, the conventional MoSeq platform is unable to directly accept keypoint data, as pervasive keypoint jitter (a previously-characterized limitation of neural network-based pose tracking[2,19]) causes MoSeq to identify false behavioral transitions[10,19]. To address this challenge, here we reformulate MoSeq as an SLDS model, which enables joint inference of keypoint positions and associated behavioral syllables. Keypoint-MoSeq effectively estimates syllable structure in a variety of datasets, including mice with implants, filmed from above or below, using either 2D or 3D keypoints. We validate keypoint-MoSeq by demonstrating that the identified behavioral syllables are interpretable; that the identified behavioral transitions match changepoints in depth and kinematic data; and that the identified syllables capture systematic fluctuations in neural activity and complex behaviors identified by expert observers. Thus keypoint-MoSeq affords much of the same insight into behavioral structure as depth MoSeq, while rendering behavioral syllables and grammar accessible to researchers who use standard video to capture animal behavior.

There are now many techniques for unsupervised behavior segmentation[6,29]. The common form of their outputs – a sequence of discrete labels – belies profound variation in how they work and the kinds of biological insight one might gain from applying them. To better understand their relative strengths and weaknesses when applied to mouse keypoint data, here we perform a detailed head-to-head comparison between keypoint-MoSeq and three alternative methods (B-SOiD[9], MotionMapper[20] and VAME[10]). All these methods similarly encode the kinematic content of mouse behavior. The methods differed radically, however, in the temporal structure of their outputs. Keypoint-MoSeq syllables lasted almost an order of magnitude longer on average than states identified by alternative clustering methods, and transitions between B-SOiD, MotionMapper and VAME states often occurred in the middle of what a human might identify as a behavioral module or motif (e.g., a rear). Our analysis suggest three possible reasons for this difference. First, unlike alternative methods, MoSeq seeks to explain behavior at a particular timescale, and therefore is better able to identify clear boundaries between behavioral elements that respect the natural sub-second rhythmicity in both neural activity and mouse behavior itself. Second, MoSeq assumes that syllables are continuous trajectories through pose space, which prevents the kind of within-module fractionation observed when keypoint data is clustered using alternative methods. Finally, the formulation of MoSeq as a probabilistic generative model means it

can infer keypoint noise and distinguish this noise from actual behavior without smoothing away meaningful behavioral transitions.

The fact that MoSeq is a probabilistic generative model means that its descriptions of behavior are constrained by the model structure and its parameters: it seeks to describe behavior as composed of auto-regressive trajectories through a pose space with switching dynamics organized at a single main timescale. Because MoSeq instantiates an explicit model for behavior, there are certainly problems in behavioral analysis for which keypoint-MoSeq may be ill-suited. For example, as has been previously noted, keypoint-MoSeq cannot integrate dynamics across a wide range of timescales, as would be possible with methods such as MotionMapper[30,31]. In addition, some behaviors — like the leg movements of walking flies — may be better captured by methods whose design emphasizes oscillatory dynamics. It is important to note that, despite its structural constraints, MoSeq-based methods are not *only* useful for capturing fine timescale structure in behavior; indeed, MoSeq has repeatedly been shown to be performant at tasks that pervasively influence the structure of behavior, including changes in behavior due to genetic mutations or drug treatments[17,32]. That said, we stress that there is no one “best” approach for behavioral analysis, as all methods involve trade-offs: methods that work for one problem (for example, identifying fast neurobehavioral correlates) may not be well suited for another problem.

The outputs of MoSeq depend upon the type of data it is fed. While similar behavioral boundaries are identified from 2D keypoints, 3D keypoints and depth data, increasing the dimensionality of the input data also increases the richness of the syllables revealed by MoSeq. Though directly modeling the raw pixel intensities of depth[14] or 2D video[33] recordings provides the most detailed access to spontaneous behavior, there are significant technical challenges (ranging from reflection sensitivity to relatively low temporal resolution) that make depth cameras difficult to use in many experimental settings. Similarly, occlusions and variation in perspective and illumination remain a challenge for direct 2D video modeling. The development of keypoint-MoSeq — together with the extraordinary advances in markerless pose tracking — should enable MoSeq to be used in a variety of these adversarial circumstances, such as when mice are obstructed from a single axis of view, or when the environment changes dynamically. Keypoint-MoSeq could therefore be an important tool moving forward in our collective attempts to explore brain-behavior relationships in increasingly ethologically relevant settings. Conversely, keypoint-MoSeq can also be applied to the petabytes of legacy data sitting fallow on the hard drives of investigators who have already done painstaking behavioral experiments using conventional video cameras. Going forward, increasingly sophisticated pose tracking approaches[19,34] and methods

that combine keypoint tracking with direct video analysis[35] may eventually close the gap in dimensionality between keypoint- and (depth) video-based pose tracking.

To facilitate the adoption of keypoint-MoSeq we have built a website (www.MoSeq4all.org) that includes free access to the code for academics as well as extensive documentation and guidance for implementation. As demonstrated by this paper, the model underlying MoSeq is modular and therefore accessible to extensions and modifications that can increase its alignment to behavioral data. For example, Costacurta et al., recently reported a time-warped version of MoSeq that incorporates a term to explicitly model variation in movement vigor[36]. We anticipate that the application of keypoint-MoSeq to a wide variety of experimental datasets will both yield important information about the strengths and failure modes of model-based methods for behavioral classification, and prompt continued innovation.

References

1 Pereira, T. D. *et al.* SLEAP: A deep learning system for multi-animal pose tracking. *Nature Methods* **19**, 486-495 (2022). https://doi.org:10.1038/s41592-022-01426-1 PMID - 35379947

2 Mathis, A. *et al.* DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nature Publishing Group* **21**, 1281-1289 (2018). https://doi.org:papers3://publication/doi/10.1038/s41593-018-0209-y

3 Graving, J. M., Chae, D., Naik, H., Li, L. & bioRxiv, B. K. Fast and robust animal pose estimation. *biorxiv.org*, http:--dx.doi.org- 10.1101 - 620245 (2019). https://doi.org:papers3://publication/doi/10.1101/620245

4 Sun, J. J. *et al.* Self-Supervised Keypoint Discovery in Behavioral Videos. *2022 Ieee Cvf Conf Comput Vis Pattern Recognit Cvpr* **00**, 2161-2170 (2022). https://doi.org:10.1109/cvpr52688.2022.00221 PMID - 36628357

5 Mathis, A., Schneider, S., Lauer, J. & Mathis, M. W. A Primer on Motion Capture with Deep Learning: Principles, Pitfalls, and Perspectives. *Neuron* **108**, 44-65 (2020). https://doi.org:10.1016/j.neuron.2020.09.017 PMID - 33058765

6 Datta, S. R., Anderson, D. J., Branson, K., Perona, P. & Leifer, A. Computational Neuroethology: A Call to Action. *Neuron* **104**, 11 - 24 (2019). https://doi.org:10.1016/j.neuron.2019.09.038

7 Anderson, D. J. & Perona, P. Toward a science of computational ethology. *Neuron* **84**, 18-31 (2014). https://doi.org:10.1016/j.neuron.2014.09.005

8 Pereira, T. D., Shaevitz, J. W. & Murthy, M. Quantifying behavior to understand the brain. *Nature Neuroscience* **23**, 1537-1549 (2020). https://doi.org:10.1038/s41593-020-00734-z PMID - 33169033

9 Hsu, A. I. & Yttri, E. A. B-SOiD, an open-source unsupervised algorithm for identification and fast prediction of behaviors. *Nature Communications* **12**, 5188 (2021). https://doi.org:10.1038/s41467-021-25420-x PMID - 34465784

10 Luxem, K. *et al.* Identifying behavioral structure from deep variational embeddings of animal motion. *Commun Biol* **5**, 1267 (2022). https://doi.org:10.1038/s42003-022-04080-7

11 Berman, G. J., Choi, D. M., Bialek, W. & Shaevitz, J. W. Mapping the structure of drosophilid behavior. (2013).

12 Marques, J. C., Lackner, S., Félix, R. & Orger, M. B. Structure of the Zebrafish Locomotor Repertoire Revealed with Unsupervised Behavioral Clustering. *Current Biology* **28**, 181 - 195.e185 (2018). https://doi.org:10.1016/j.cub.2017.12.002

13 Todd, J. G., Kain, J. S. & de Bivort, B. L. Systematic exploration of unsupervised methods for mapping behavior. *Physical Biology* **14**, 015002 (2017). https://doi.org:10.1088/1478-3975/14/1/015002

14 Wiltschko, A. B. *et al.* Mapping Sub-Second Structure in Mouse Behavior. *Neuron* **88**, 1121-1135 (2015). https://doi.org:10.1016/j.neuron.2015.11.031

15 Markowitz, J. E. *et al.* Spontaneous behaviour is structured by reinforcement without explicit reward. *Nature* **614**, 108-117 (2023). https://doi.org:10.1038/s41586-022-05611-2
16 Markowitz, J. E. *et al.* The Striatum Organizes 3D Behavior via Moment-to-Moment Action Selection. *Cell* **174**, 44-58.e17 (2018).
17 Wiltschko, A. B. *et al.* Revealing the structure of pharmacobehavioral space through motion sequencing. *Nat. Neurosci.* (2020). https://doi.org:10.1038/s41593-020-00706-3
18 Lin, S. *et al.* Characterizing the structure of mouse behavior using Motion Sequencing. (2022). https://doi.org:10.48550/ARXIV.2211.08497
19 Wu, A. *et al. Deep Graph Pose: a semi-supervised deep graphical model for improved animal pose tracking*. (2020).
20 Berman, G. J., Choi, D. M., Bialek, W. & Shaevitz, J. W. Mapping the stereotyped behaviour of freely moving fruit flies. *Journal of the Royal Society, Interface / the Royal Society* **11** (2014). https://doi.org:papers3://publication/doi/10.1098/rsif.2014.0672
21 Murphy, K. P. *Machine Learning*. (MIT Press, 2012).
22 Linderman, S. *et al.* in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics* Vol. 54 (eds Singh Aarti & Zhu Jerry) 914--922 (PMLR, Proceedings of Machine Learning Research, 2017).
23 Zhang, L., Dunn, T., Marshall, J., Olveczky, B. & Linderman, S. in *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics* Vol. 130 (eds Banerjee Arindam & Fukumizu Kenji) 2800--2808 (PMLR, Proceedings of Machine Learning Research, 2021).
24 Klibaite, U. *et al.* Deep phenotyping reveals movement phenotypes in mouse neurodevelopmental models. *Molecular Autism* **13**, 12 (2022). https://doi.org:10.1186/s13229-022-00492-8
25 Panigrahi, B. *et al.* Dopamine Is Required for the Neural Representation and Control of Movement Vigor. *Cell* **162**, 1418-1430 (2015). https://doi.org:10.1016/j.cell.2015.08.014
26 Bohnslav, J. P. *et al.* DeepEthogram, a machine learning pipeline for supervised behavior classification from raw pixels. *eLife* **10**, e63377 (2021). https://doi.org:10.7554/eLife.63377
27 Sun, J. J. *et al.* Caltech Mouse Social Interactions (CalMS21) Dataset. (2021). https://doi.org:10.22002/D1.1991
28 Ye, S., Mathis, A. & Mathis, M. W. Panoptic animal pose estimators are zero-shot performers. (2022). https://doi.org:10.48550/ARXIV.2203.07436
29 Luxem, K. *et al.* Open-Source Tools for Behavioral Video Analysis: Setup, Methods, and Development. *arXiv* (2022). https://doi.org:10.48550/arxiv.2204.02842
30 Berman, G. J., Bialek, W. & Shaevitz, J. W. Predictability and hierarchy in Drosophila behavior. *Proceedings of the National Academy of Sciences* **113**, 11943-11948 (2016). https://doi.org:papers3://publication/doi/10.1073/pnas.1607601113
31 Berman, G. J. Measuring behavior across scales. *BMC biology* **16**, 23 (2018). https://doi.org:papers3://publication/doi/10.1186/s12915-018-0494-7
32 Wiltschko, A. B. *et al.* Mapping Sub-Second Structure in Mouse Behavior. *Neuron* **88**, 1121-1135 (2015).

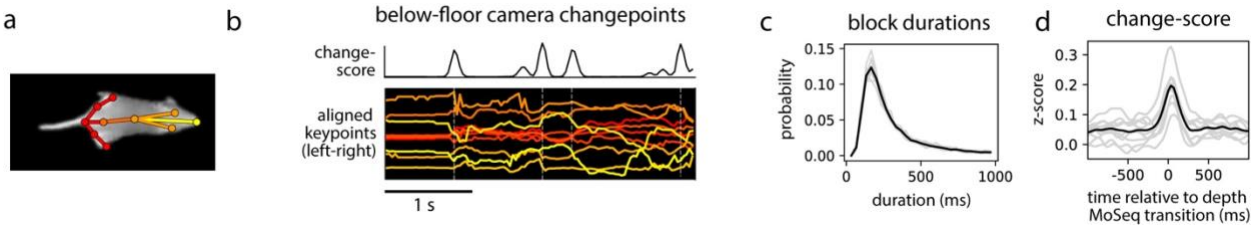
33 Batty, E. *et al.* in *Advances in Neural Information Processing Systems* Vol. 32 (eds H. Wallach *et al.*) (Curran Associates, Inc., 2019).
34 Bohnslav, J. P. *et al.* ArMo: An Articulated Mesh Approach for Mouse 3D Reconstruction. *bioRxiv*, 2023.2002.2017.526719 (2023). https://doi.org:10.1101/2023.02.17.526719
35 Whiteway, M. R. *et al.* Partitioning variability in animal behavioral videos using semi-supervised variational autoencoders. *PLOS Computational Biology* **17**, e1009439 (2021). https://doi.org:10.1371/journal.pcbi.1009439
36 Costacurta, J. C. *et al.* in *Advances in Neural Information Processing Systems* (eds Alice H. Oh, Alekh Agarwal, Danielle Belgrave, & Kyunghyun Cho) (2022).

Acknowledgements
S.R.D. is supported by NIH grants RF1AG073625, R01NS114020, U24NS109520, the Simons Foundation Autism Research Initiative, and the Simons Collaboration on Plasticity and the Aging Brain. S.R.D. and S.W.L are supported by NIH grant U19NS113201 and the Simons Collaboration on the Global Brain. C.W. is a Fellow of the Jane Coffin Childs Memorial Fund for Medical Research. W.F.G. is supported by NIH grant F31NS113385. M.J. is supported by NIH grant F31NS122155. S.W.L is supported by the Alfred P. Sloan Foundation. T.P. is supported by a Salk Collaboration Grant. We thank J. Araki for administrative support; the HMS Research Instrumentation Core, which is supported by the Bertarelli Program in Translational Neuroscience and Neuroengineering, and by NEI grant EY012196; and members of the Datta laboratory for useful comments on the paper. Portions of this research were conducted on the O2 High Performance Compute Cluster at Harvard Medical School.

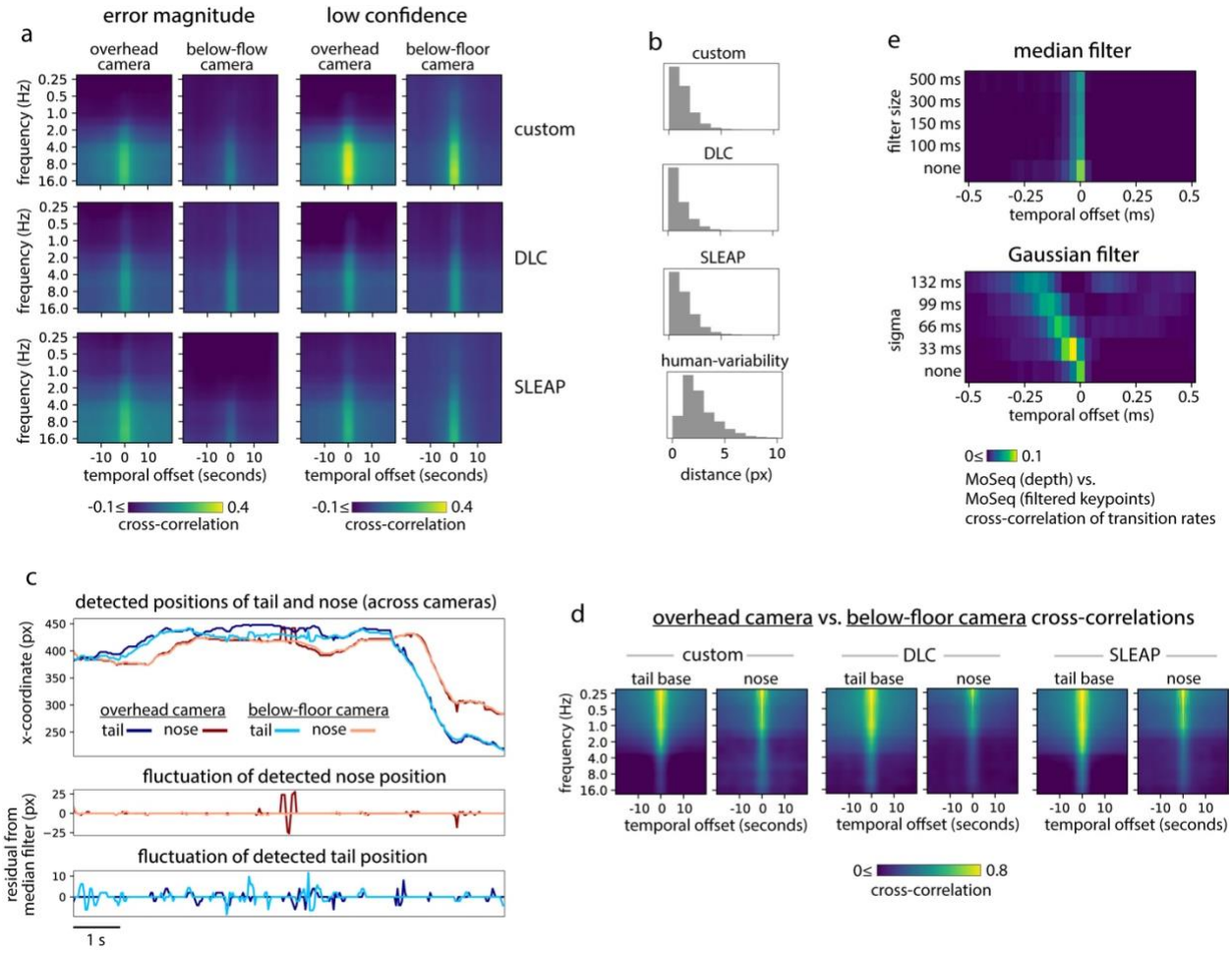
Competing interests
S.R.D. sits on the scientific advisory boards of Neumora and Gilgamesh Therapeutics, which have licensed or sub-licensed the MoSeq technology.

Code availability
Software links and user-support for both depth and keypoint data are available at the MoSeq homepage: MoSeq4all.org. Data loading, project configuration and visualization are enabled through the “keypoint-moseq” python library (https://github.com/dattalab/keypoint-moseq). We also developed a standalone library called “jax-moseq” for core model inference (https://github.com/dattalab/jax-moseq). Both libraries are freely available to the research community.

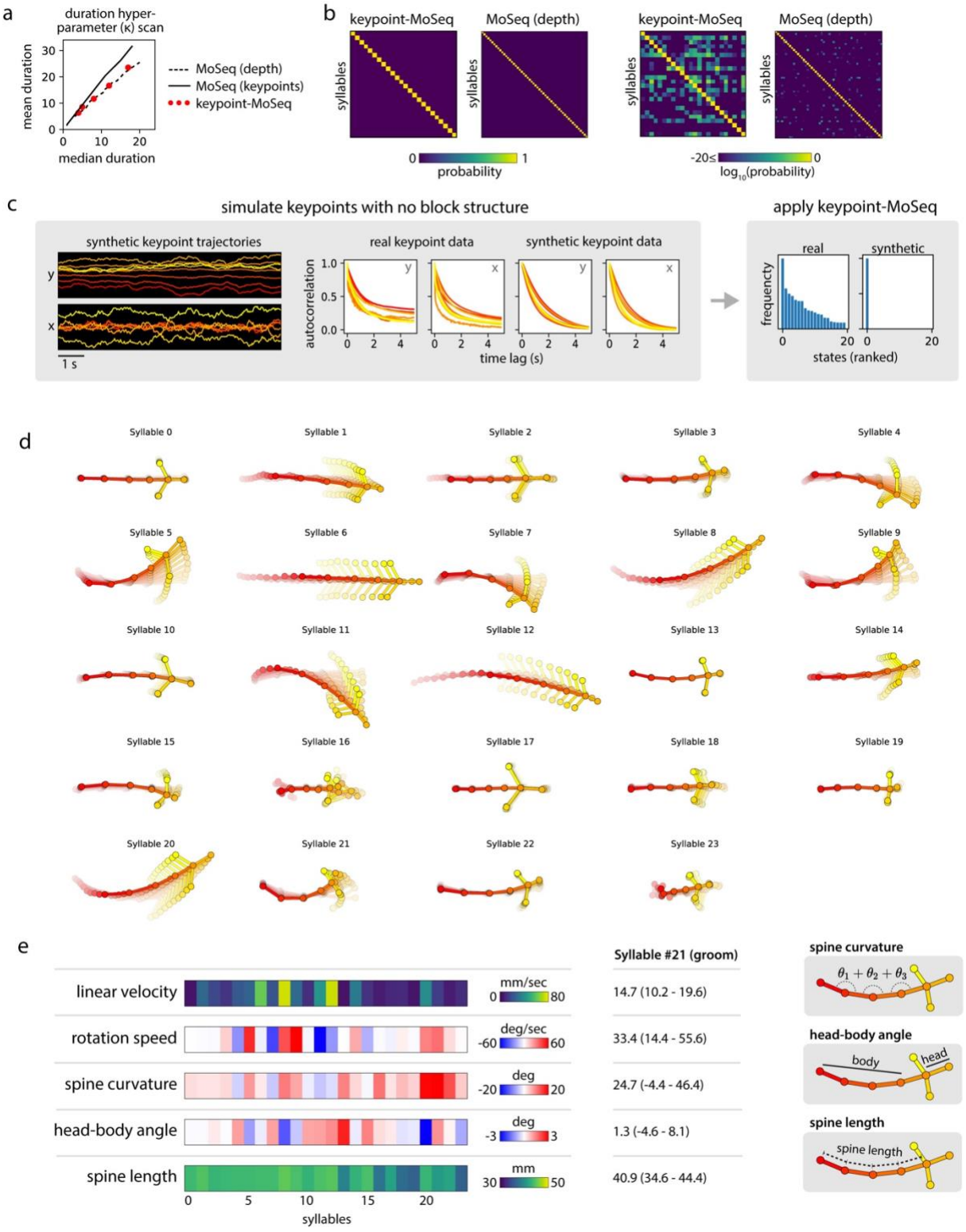
Extended Data



Extended Data Figure 1: a) 2D keypoints tracked using infrared video from a camera viewing the mouse through a transparent floor. **b)** Egocentrically aligned keypoint trajectories (bottom) and change-score derived from those keypoints (top). Vertical dashed lines represent changepoints (peaks in the change-score). **c)** Distribution of inter-changepoint intervals. **d)** Keypoint change-score aligned to syllable transitions from depth MoSeq. Results in (c) and (d) are shown for the full dataset (black lines) and for each recording session (gray lines).

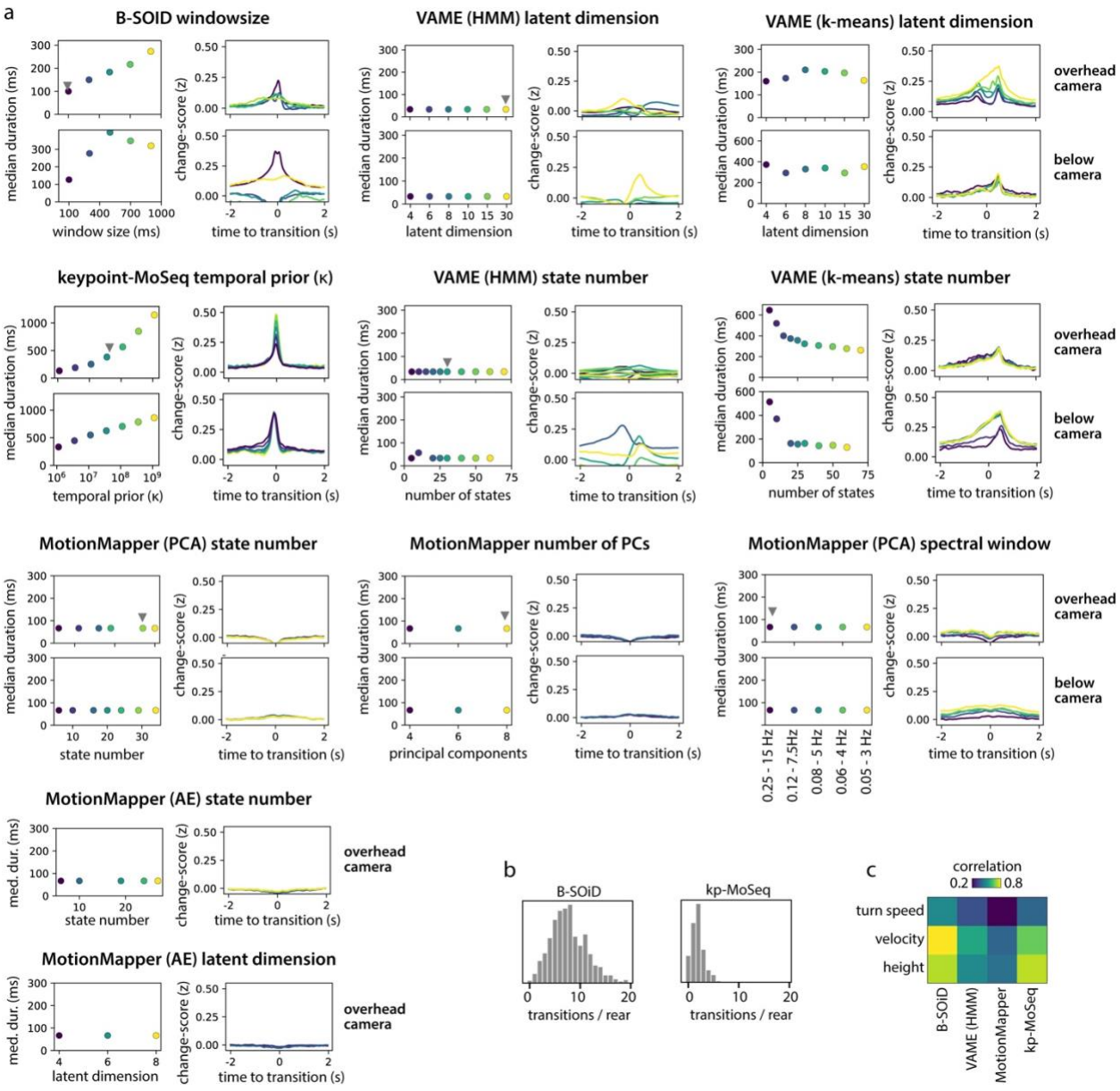


Extended Data Figure 2: a) Cross-correlation between the spectral content of keypoint fluctuations and either error magnitude (left) or a measure of low-confidence keypoint detections (right) (see Methods). **b)** Magnitude of fast fluctuations in keypoint position for three different tracking methods (top), calculated as the per-frame distance from the measured trajectory of a keypoint to a smoothened version of the same trajectory, where smoothing was performed using a gaussian kernel with width 100ms. The distribution of distances in between manually defined keypoint position across human annotators is shown on the bottom for comparison. and magnitude of variation in keypoint position across human annotators (bottom). **c)** Top: position of the nose and tail-base over a 10-second interval, shown for both the overhead and below-floor cameras. Bottom: fast fluctuations in each coordinate, obtained as residuals after median filtering. **d)** Cross-correlation between spectrograms obtained from two different camera angles for either the tail base or the nose, shown for each tracking method. **e)** Cross-correlation of transitions rates, comparing MoSeq (depth) and MoSeq applied to keypoints with various levels of smoothing using either a Gaussian or median filter.

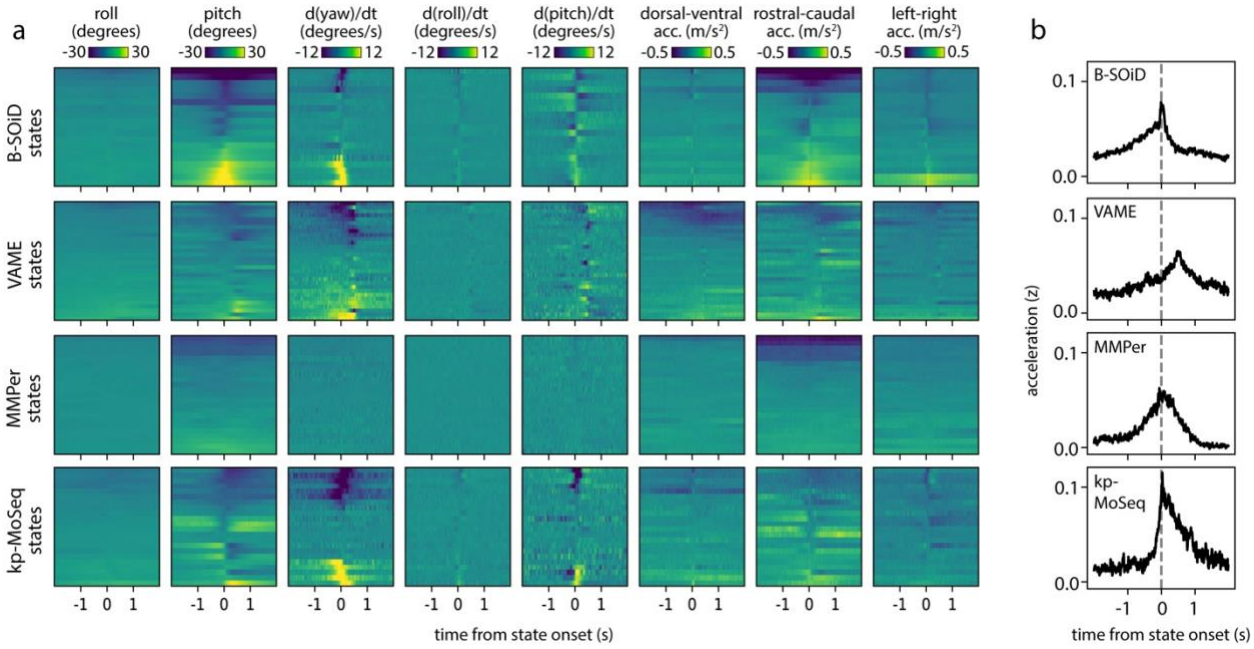


Extended Data Figure 3: a) Relationship between mean and median syllable duration as the temporal stickiness hyper-parameter κ is varied. **b)** Syllable cross-likelihoods, defined as the probability, on average, that time-intervals assigned to one syllable (column) could have arisen

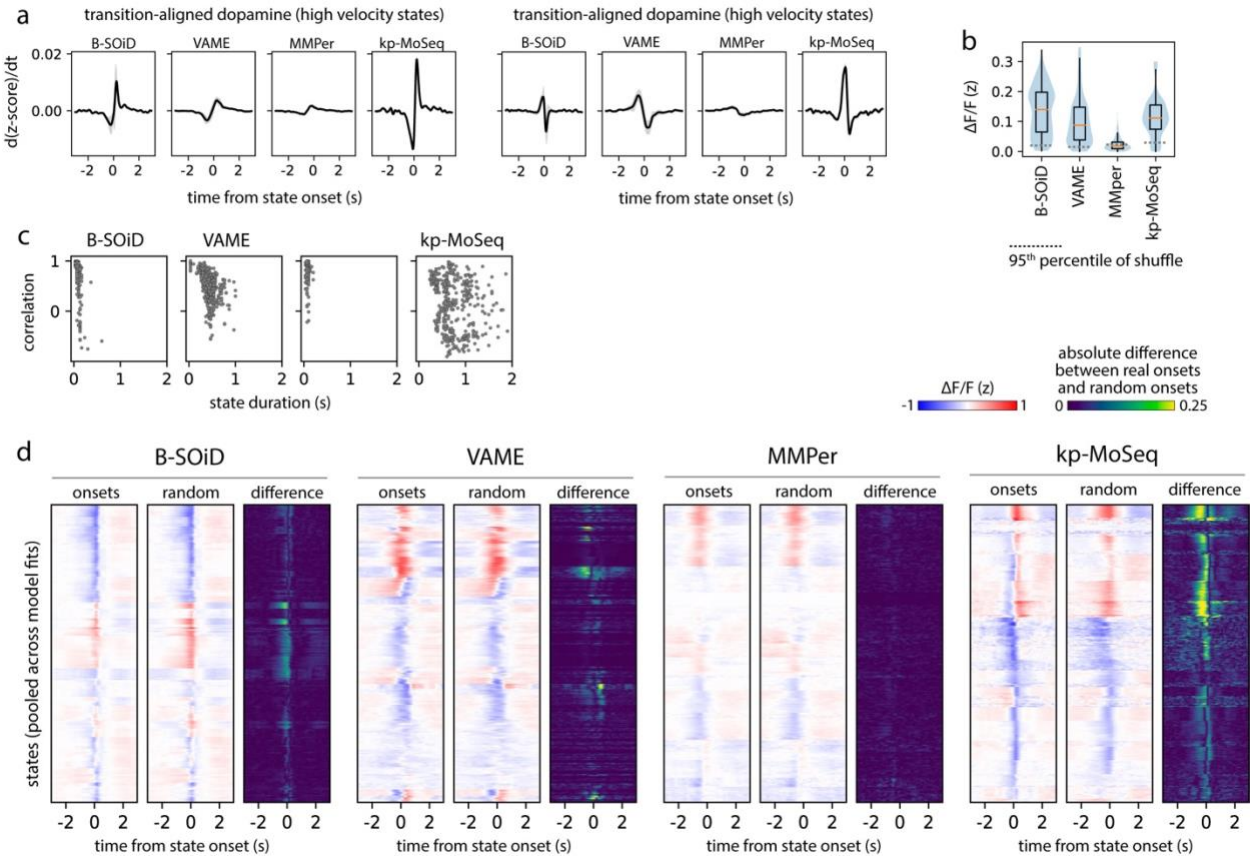
from another syllable (row). Cross-likelihoods were calculated for keypoint-MoSeq and for depth MoSeq. The results for both methods are plotted twice, using either an absolute scale (left) or a log scale (right). **c)** Modeling results for synthetic keypoint data with a similar statistical structure as the real data but lacking in changepoints. **Left:** example of synthetic keypoint trajectories. **Middle:** autocorrelation of keypoint coordinates for real vs. synthetic data, showing similar dynamics at short timescales. **Right:** distribution of syllable frequencies for keypoint-MoSeq models trained on real vs. synthetic data. **d)** Average pose trajectories for syllables identified by keypoint-MoSeq. Each trajectory includes ten evenly-timed poses from 165ms before to 500ms after syllable onset. **e)** Kinematic and morphological parameters for each syllable.



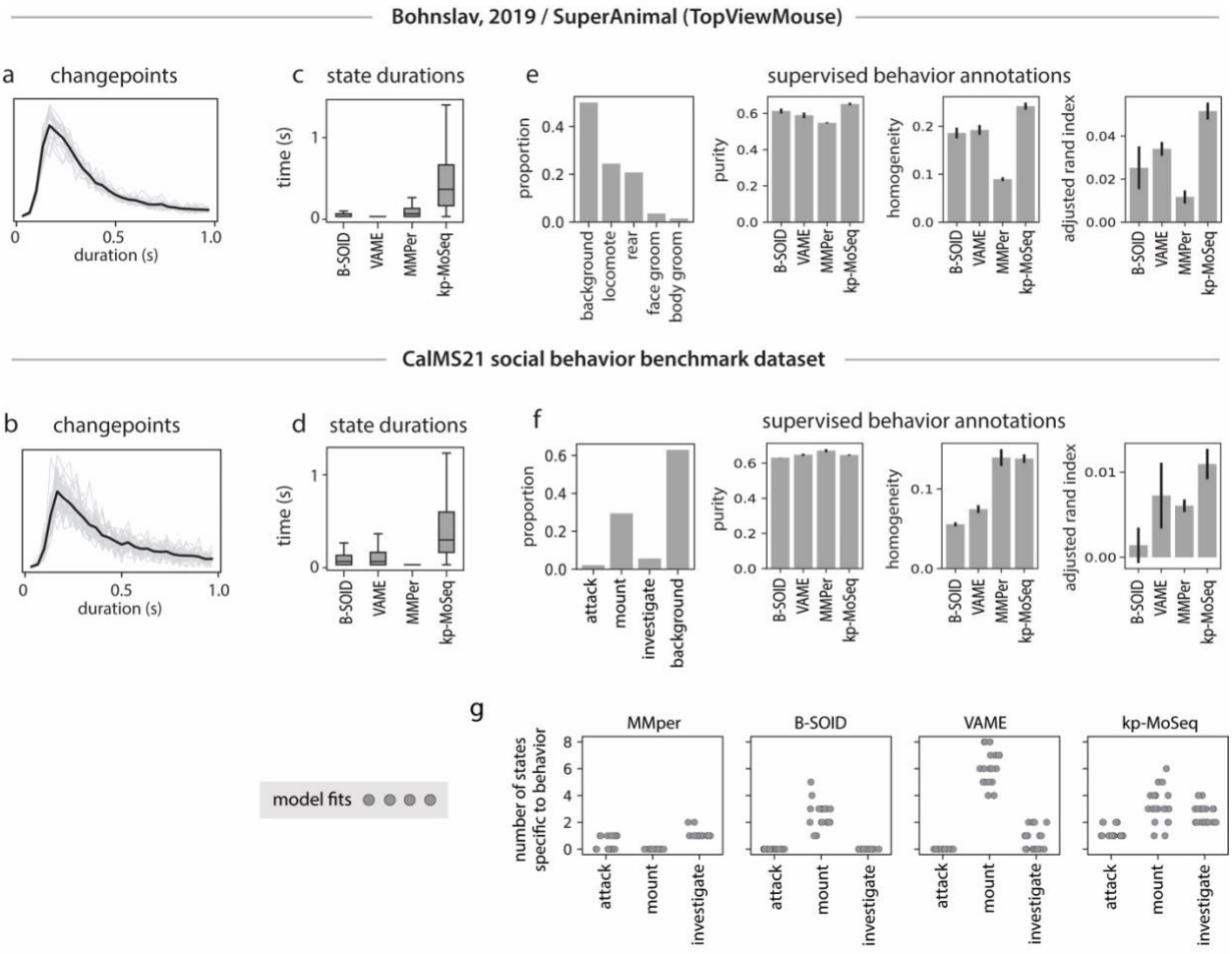
Extended Data Figure 4: a) Output of unsupervised behavior segmentation algorithms across a range of parameter settings, applied to 2D keypoint data from two different camera angles. The median state duration (left) and the average (z-scored) keypoint change-score aligned to state transitions (right) are shown for each method and parameter value. Gray pointers indicate default parameter values used for subsequent analysis. **b)** Distributions showing the number of transitions that occur during each rear. **c)** Accuracy of kinematic decoding models that were fit to state sequences from each method.



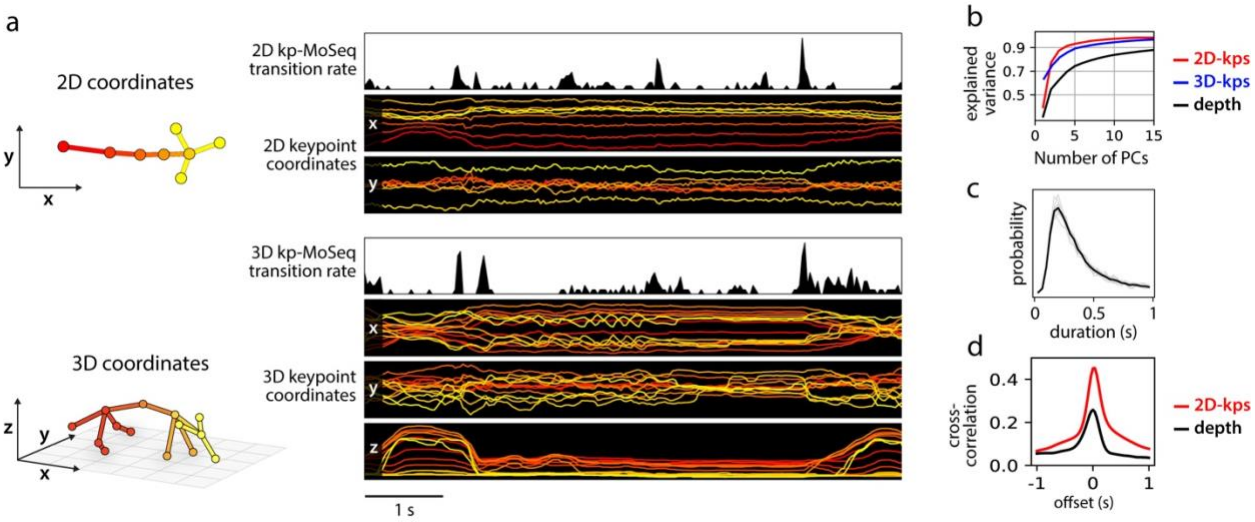
Extended Data Figure 5: a) IMU signals aligned to state onsets from several behavior segmentation methods. Each row corresponds to a behavior state and shows the average across all onset times for that state. **b)** As (a) for acceleration but showing the median across all states.



Extended Data Figure 6: a) Derivative of the dopamine signal aligned to the onsets of high velocity or low velocity behvior states. States from each method were classified evenly as high or low velocity based on the mean centroid velocity during their respective frames. **b)** Distributions capturing the average of the dopamine signal across states from each method. **c)** Relationship between state durations and correlations from Fig 5f, showing that the impact of randomization is not a simple function of state duration. **d)** Average dopamine fluctuations aligned to state onsets (left), or aligned to random frames throughout the execution of each state (middle), as well as the absolute difference between the two alignment approaches (right), shown for each unsupervised behavior segmentation approach.



Extended Data Figure 7: a,b) Distribution of inter-changepoint intervals for the (Bohnslav, 2019) open field dataset (a) and CalMS21 social behavior benchmark (b), shown respectively for the full datasets (black lines) and for each recording session (gray lines). **c,d)** Distribution of state durations from each behavior segmentation method. **e,f)** Frequency of each human-annotated behavior (left) and agreement between human-annotations and unsupervised behavior labels (right), quantified using three different metrics (see Methods). **g)** Number of unsupervised states specific to each human-annotated behavior in the CalMS21 dataset, shown for 20 independent fits of each unsupervised method. A state was defined as specific if > 50% of frames bore the annotation.



Extended Data Figure 8: a) Left: Keypoints tracked in 2D (top) or 3D (bottom) and corresponding egocentric coordinate axes. Right: example keypoint trajectories and transition rates from keypoint-MoSeq. Transition rate is defined as the posterior probability of a transition occurring on each frame. **b)** Cumul ative fraction of explained variance for increasing number of principal components (PCs). PCs were fit to egocentrically aligned 2D keypoints, egocentrically aligned 3D keypoints, or depth videos respectively. **c)** Distribution of inter-changepoint intervals in the 3D keypoint dataset, shown**. d)** Cross-correlation between the 3D keypoint change-score and change-scores derived from 2D keypoints and depth respectively.

Experimental Methods

Animal care and behavioral experiments

Unless otherwise noted, behavioral recordings were performed on 8–16-week-old C57/BL6 mice (The Jackson Laboratory stock no. 000664). Mice were transferred to our colony at 6-8 weeks of age and housed in a reverse 12-hour light/12-hour dark cycle. We single-housed mice after stereotactic surgery, and group-housed them otherwise. On recording days, mice were brought to the laboratory, habituated in darkness for at least 20 minutes, and then placed in an open field arena for 30-60 mins. We recorded 6 male mice for 10 sessions (6 hours) in the initial round of open field recordings; and 5 male mice for 52 sessions (50 hours) during the accelerometry recordings. The dopamine photometry recordings were obtained from a recent study[1]. They include 6 C57/BL6 mice and 8 DAT-IRES-cre (The Jackson Laboratory stock no. 006660) mice of both sexes, recorded for 378 sessions (180 hours). In a subset of these recordings, specific syllables (detected in real-time using depth video) were targeted for closed-loop reinforcement, as described in ref[1].

Stereotactic surgery procedures

For all stereotactic surgeries, mice were anaesthetized using 1–2% isoflurane in oxygen, at a flow rate of 1 L/min for the duration of the procedure. Anterior-posterior (AP) and medial-lateral (ML) coordinates were zeroed relative to bregma, the dorso-ventral (DV) coordinate was zeroed relative to the pial surface, and coordinates are in units of mm. For dopamine recordings, 400nL of AAV5.CAG.dLight1.1 (Addgene #111067, titer: 4.85×10^{12}) was injected at a 1:2 dilution into the DLS (AP 0.260; ML 2.550; DV −2.40) and a single 200-µm diameter, 0.37–0.57 NA fiber cannula was implanted 200 µm above the injection site (see ref[1] for additional details). For accelerometry recordings, we surgically attached a millmax connector (DigiKey ED8450-ND) and head bar to the skull and secured it with dental cement (Metabond). A 9 degree-of-freedom absolute orientation inertial measurement unit (IMU; Bosch BN0055) was mounted on the millmax connector using a custom printed circuit board (PCB) with a net weight below 1g.

Data acquisition from the IMU

The IMU was connected to a Teensy microcontroller, which was programmed using the Adafruit BNO055 library with default settings (sample rate: 100 Hz, units: m/s^2). To synchronize the IMU measurements and video recordings, we used an array of near infrared LEDs to display a rapid sequence of random 4-bit codes that updated throughout the recording. The code sequence was later extracted from the behavioral

videos and used to fit a piecewise linear model between timestamps from the videos and timestamps from the IMU.

Recording setup

For the initial set of open field recordings (Fig 1-4), mice were recorded in a square arena with transparent floor and walls (30cm length and width). Microsoft Azure Kinect cameras captured simultaneous depth and near-infrared video at 30Hz. Six cameras were used in total: one above, one below, and four side cameras at right angles at the same height as the mouse. For the accelerometry recordings, we used a single Microsoft Azure Kinect camera placed above the mouse, and an arena with transparent floor and opaque circular walls (45cm diameter). Data was transferred from the IMU using a light-weight tether attached to a custom-built active commutator. For the dopamine perturbation experiments, we used a slightly older camera model – the Microsoft Kinect 2 – to capture simultaneous depth and near-infrared at 30Hz. The recording arena was circular with opaque floor and walls (45cm diameter). Photometry signals were conveyed from the mouse using a fiber-optic patch cord attached to a passive commutator.

Computational Methods

Processing depth videos

Applying MoSeq to depth videos involves: (1) mouse tracking and background subtraction; (2) egocentric alignment and cropping; (3) principal component analysis (PCA); (4) probabilistic modeling. We applied steps (2-4) as described in the MoSeq2 pipeline[2]. For step (1), we trained a convolutional neural network (CNN) with a Unet++[3] architecture to segment mouse from background using ~5000 hand-labeled frames as training data.

Keypoint tracking

We used CNNs with an HRNet[4] architecture (https://github.com/stefanopini/simple-HRNet) with a final stride of 2 for pose tracking. The networks were trained on ~1000 hand-labeled frames each for the overhead, below-floor, and side-view camera angles. Frame-labelling was crowdsourced through a commercial service (Scale AI). For the overhead camera, we tracked two ears and 6 points along the dorsal midline (tail base, lumbar spine, thoracic spine, cervical spine, head, and nose). For the below-floor camera, we tracked the tip of each forepaw, the tip and base of each hind paw, and four points along the ventral midline (tail base, genitals, abdomen, and nose). For the side cameras, we tracked the same eight points as for the overhead camera, and also included the six limb points that were used for the below-floor camera (14 total). We trained a separate CNN for each camera angle. Target activations were formed by centering a Gaussian with 10px standard deviation on each keypoint. We used the location of the maximum pixel in each output channel of the neural network to determine keypoint coordinates, and used the value at that pixel to set the confidence score. We also trained models from DeepLabCut (version 2.2.1, resnet50 architecture, otherwise default parameters) and SLEAP (version 1.2.3, with baseline_large_rf.single.json configuration) on the overhead-camera and below-floor-camera datasets.

3D pose inference

Using 2D keypoint detections from six cameras, 3D keypoint coordinates were triangulated and then refined using GIMBAL, a model-based approach that leverages anatomical constraints and motion continuity[5]. To fit GIMBAL, we computed initial 3D keypoint estimates using robust triangulation (i.e. by taking the median across all camera pairs, as in 3D-DeepLabCut[6]) and then filtered to remove outliers using the EllipticEnvelope method from sklearn; We then fit the skeletal parameters and directional priors for GIMBAL using expectation maximization with 50 pose states (see ref[5] for details). Finally, we applied the fitted GIMBAL model to each recording, using the following parameters for all keypoints: obs_outlier_variance=1e6, obs_inlier_variance=10, pos_dt_variance=10. The latter parameters were chosen based

on the accuracy of the resulting 3D keypoint estimates, as assessed from visual inspection.

Inferring model-free changepoints

We defined changepoints as sudden, simultaneous shifts in the trajectories of multiple keypoints. We detected them using a procedure similar to the filtered derivative algorithm described in ref[7], but with changes to emphasize simultaneity across multiple keypoints. The changes account for the lower dimensionality of keypoint data compared to depth videos, and for the unique noise structure of markerless keypoint tracking, in which individual keypoints occasionally jump a relatively large distance due to detection errors. Briefly, the new procedure first defines a continuous change-score by: (1) calculating the rate of each in each keypoint coordinate; (2) quantifying simultaneity in the change-rates across keypoints; (3) transforming the signal based on statistical significance with respect to a temporally shuffled null distribution; (4) identifying local peaks in the resulting significance score. The details of each step are as follows.

1) **Calculating rates of change:** We transformed the keypoint coordinates on each frame by centering and aligned them along the tail-nose axis. We then computed the derivative of each coordinate for each keypoint, using a sliding window of length 3 as shown below, where x_t denotes the value of a coordinate at time t.

$$\dot{x}_t \approx \frac{1}{3}(x_{t+3} + x_{t+2} + x_{t+1} - x_{t-1} - x_{t-2} - x_{t-3})$$

2) **Quantifying simultaneous changes:** The derivatives for each keypoint were Z-scored and then binarized with a threshold. We then counted the number of threshold crossings on each frame and smoothed the resulting time-series of counts using a Gaussian filter with a one-frame kernel. The value of the threshold was chosen to maximize the total number of detected changepoints.

3) **Comparing to a null distribution:** We repeated step (2) for 1000 shuffled datasets, in which each keypoint trajectory was cyclically permuted by a random interval. Using the shuffles as a null distribution, we computed a P-value for each frame and defined the final change-score as $-\log_{10}(\mathrm{pval})$

4) **Identifying local peaks in the change-score:** We identified local peaks in the change-score s_t, i.e., times t for which $s_{t-1} < s_t > s_{t+1}$. Peaks were classified as statistically significant when they corresponded to a p-value below 0.01, which was chosen to control the false-discovery rate at 10%. The statistically significant peaks were reported as changepoints for downstream analysis.

Spectral Analysis

To analyze keypoint jitter, we quantified the magnitude of fluctuations across a range of frequencies by computing a spectrogram for each keypoint along each coordinate axis. Spectrograms were computed using the python function scipy.signal.spectrogram with nperseg=128 and noverlap=124. The spectrograms were then combined through averaging: each keypoint was assigned a spectrogram by averaging over the two coordinate axes, and the entire animal was assigned a spectrogram by averaging over all keypoints.

We used the keypoint-specific spectrograms to calculate cross-correlations with $-\log_{10}(\text{neural network detection confidence})$, as well as the "error magnitude" (Fig 2f). Error magnitude was defined as the distance between the detected 2D location of a keypoint (based on a single camera angle) and a reprojection of its 3D position (based on consensus across six camera angles; see "3D pose inference" above). We also computed the cross-correlation between nose- and tail-base-fluctuations at each frequency, as measured by the overhead and below-floor cameras respectively. Finally, we averaged spectral power across keypoints to compute the cross-correlation with model transition rates (Fig 2f), defined as the per-frame probability of a state transitions across 20 model restarts.

Applying keypoint-MoSeq

The initial open field recordings (Fig 1-4), as well as the accelerometry, dopamine, and two benchmark datasets were modeled separately. Twenty models with different random seeds were fit for each dataset (except for the accelerometry data, in which case one model was fit).

Modeling consisted of two phases: (1) Fitting an autoregressive hidden Markov model (AR-HMM) to a fixed pose trajectory derived from PCA of egocentric-aligned keypoints; (2) Fitting a full keypoint-MoSeq model initialized from the AR-HMM. References in the text to "MoSeq applied to keypoints" or "MoSeq (keypoints)", e.g., in Figs 2-3, refer to output of step (1). Both steps are described below, followed by a detailed description of the model and inference algorithm in the mathematical modeling section. In all cases, we excluded rare states (frequency < 0.5%) from downstream analysis. We have made the code available as a user-friendly package, available at Moseq4all.org.

1) **Fitting an initial AR-HMM:**

   We first modified the keypoint coordinates, defining keypoints with confidence below 0.5 as missing data and in imputing their values via linear interpolation,

and then augmenting all coordinates with a small amount of random noise; The noise values were uniformly sampled from the interval [-0.1, 0.1] and helped prevent degeneracy during model fitting. Importantly, these preprocessing steps were only applied during AR-HMM fitting – the original coordinates were used when fitting the full keypoint-MoSeq model.

Next, we centered the coordinates on each frame, aligned them using the tail-nose angle, and then transformed them using PCA with whitening. The number of principal components (PCs) was chosen for each dataset as the minimum required to explain 90% of total variance. This resulted in 4 PCs for the overhead camera 2D datasets, 6 PCs for the below-floor-camera 2D datasets, and 6 PCs for the 3D dataset.

We then used Gibbs sampling to infer the states and parameters of an AR-HMM, including the state sequence z, the autoregressive parameters A, b, Q, and the transition parameters π, β. The hyper-parameters for this step, listed in the mathematical modeling section below, were generally identical to those in the original depth-MoSeq model[7]. The one exception was κ which we adjusted separately for each dataset to ensure a median state duration of 400ms.

2) **Fitting a full keypoint-MoSeq model:**

We next fit the full set of variables for keypoint-MoSeq, which include the AR-HMM variables mentioned above, as well as the location v and heading h, latent pose trajectory x, per-keypoint noise level σ^2, and per-frame/per-keypoint noise scale s. Fitting was performed using Gibbs sampling for 500 iterations, at which point the log joint probability appeared to have stabilized.

The hyper-parameters for this step are enumerated in the mathematical modeling section below. In general, we used the same hyper-parameter values across datasets. The two exceptions were κ, which again had to be adjusted to maintain a median state duration of 400ms, and s_0, which determines a prior on the noise scale. Since low-confidence keypoint detections often have high error, we set s_0 using a logistic curve that transitions between a high-noise regime ($s_0 = 100$) for detections with low confidence and a low-noise regime ($s_0 = 1$) for detections with high confidence:

$$s_0 = 1 + 100\left(1 + e^{20(\text{confidence}-0.4)}\right)^{-1}$$

Trajectory plots

To visualize the modal trajectory associated with each syllable (Fig 3e), we (1) computed the full set of trajectories for all instances of all syllables (2) used a local density criterion to identify a single representative instance of each syllable (3) computed a final trajectory using the nearest neighbors of the representative trajectory.

1) **Computing the trajectory of individual syllable instances:** Let y_t, v_t, and h_t denote the keypoint coordinates, centroid and heading of the mouse at time t, and let $F(v, h; y)$ denote the rigid transformation that egocentrically aligns y using centroid v and heading h. Given a syllable instance with onset time T, we computed the corresponding trajectory X_T by centering and aligning the sequence of poses $(y_{T-5}, \dots, y_{T+15})$ using the centroid and heading on time T. In other words,

$$X_T = [F(v_T, h_T; y_{T-5}), \dots, F(v_T, h_T; y_{T+15})]$$

2) **Identifying a representative instance of each syllable:** The collection of trajectories computed above can be thought of as a set of points in a high dimensional trajectory space (for K keypoints in 2D, this space would have dimension $40K$). Each point has a syllable label, and the segregation of these labels in the trajectory space represents the kinematic differences between syllables. To capture these differences, we computed a local probability density function for each syllable, and a global density function across all syllables. We then selected a representative trajectory X for each syllable by maximizing the ratio

$$\frac{\text{local density}(X)}{\text{global density}(X)}$$

The density functions were computed as the mean distance from each point to its 50 nearest neighbors. For the global density, the nearest neighbors were selected from among all instances of all syllables. For the local densities, the nearest neighbors were selected from among instances of the target syllable.

3) **Computing final trajectories for each syllable:** For each syllable and its representative trajectory X, we identified the 50 nearest neighbors of X from among other instanes of the same syllable and then computed a final trajectory as the mean across these nearest neighbors. The trajectory plots in Fig 3e consist of 10 evenly-space poses along this trajectory, i.e., the poses at times $T-5, T-3, \dots, T+13$.

Cross-syllable likelihoods

We defined each cross-syllable likelihood[7] as the probability (on average) that instances of one syllable could have arisen based on the dynamics of another syllable. The probabilities were computed based on the discrete latent states z_t, continuous latent states x_t, and autoregressive parameters A, b, Q output by keypoint-MoSeq. The instances $I(n)$ of syllable n were defined as the set of all sequences $(t_s, \ldots, t_e)$ of consecutive timepoints such that $z_t = n$ for all $t_s \leq t \leq t_e$ and $z_{t_s-1} \neq n \neq z_{t_e+1}$. For each such instance, one can calculate the probability $P(x_{t_s}, \ldots, x_{t_e} | A_m, b_m, Q_m)$ that the corresponding sequence of latent states arose from the autoregressive dynamics of syllable m. The cross-syllable likelihood C_{nm} is defined in terms of these probabilities as

$$C_{nm} = \frac{1}{|I(n)|} \sum_{(t_s,\ldots,t_e)\in \mathrm{I(n)}} \frac{(x_{t_s}, \ldots, x_{t_e} | A_m, b_m, Q_m)}{(x_{t_s}, \ldots, x_{t_e} | A_n, b_n, Q_n)}$$

Generating synthetic keypoint data

To generate the synthetic keypoint trajectories used for Extended Data Fig 3c, we fit a linear dynamical system (LDS) to egocentrically aligned keypoint trajectories and then sampled randomly generated outputs from the fitted model. The LDS was identical to the model underlying keypoint-MoSeq (see mathematical modeling section below), except that it only had one discrete state, lacked centroid ad heading variables, and allowed separate noise terms for the x- and y- coordinates of each keypoint.

Applying B-SOiD

B-SOiD is an automated pipeline for behavioral clustering that: (1) preprocesses keypoint trajectories to generate pose and movement features; (2) performs dimensionality reduction on a subset of frames using UMAP; (3) clusters points in the UMAP space; (4) uses a classifier to extend the clustering to all frames[8]. We fit B-SOiD separately for each dataset. In each case, steps 2-4 were performed 20 times with different random seeds, and the pipeline was applied with standard parameters; 50,000 randomly sampled frames were used for dimensionality reduction and clustering, and the min_cluster_size range was set to 0.5% - 1%. Since B-SOiD uses a hardcoded window of 100ms to calculate pose and movement features, we re-ran the pipeline with falsely inflated framerates for the window-size scan in Extended Data Fig 4a. In all analyses involving B-SOiD, rare states (frequency < 0.5%) were excluded from analysis.

Applying VAME

VAME is a pipeline for behavioral clustering that: (1) preprocesses keypoint trajectories and transforms them into egocentric coordinates; (2) fits a recurrent neural network

(RNN); (3) clusters the latent code of the RNN[9]. We applied these steps separately to each dataset, in each case running step (3) 20 times with different random seeds. For step (1), we used the same parameters as in keypoint-MoSeq – egocentric alignment was performed along the tail-nose axis, and we set the pose_confidence threshold to 0.5. For step (2), we set time_window=30 and zdims=20 for all datasets, except for the zdim-scan in Extended Data Fig 4a. VAME provides two different options for step (3): fitting an HMM (default) or applying K-Means (alternative). We fit an HMM for all datasets and additionally applied K-Means to the initial open dataset. In general, we approximately matched the number of states/clusters in VAME to the number identidied by keypoint-MoSeq, except when scanning over state number in Extended Data Fig 4a. In all analyses involving VAME, rare states (frequency < 0.5%) were excluded from analysis.

Applying MotionMapper

MotionMapper performs unsupervised behavioral segmentation by: (1) applying a wavelet transform to preprocessed pose data; (2) nonlinearly embedding the transformed data in 2D; (3) clustering the 2D data with a watershed transform[10]. We applied MotionMapper separately to each dataset using the python package https://github.com/bermanlabemory/motionmapperpy. In general, the data were egocentrically aligned along the tail-nose axis and then projected into 8 dimensions using PCA. 10 log-spaced frequencies between 0.25 and 15Hz were used for the wavelet transform, and dimensionality reduction was performed using tSNE. The threshold for watershedding was chosen so as to produce at least 25 clusters, consistent with keypoint-MoSeq for the overhead camera data. Rare states (frequency < 0.5%) were excluded from analysis. For the parameter scan in Extended Data Fig 4a, we varied the each of these parameters while holding the others fixed, including the threshold for watershedding, the number of initial PCA dimensions, and the frequency range of wavelet analysis. We also repeated a subset of these analyses using an alternative autoencoder-based dimensionality reduction approach, as described in the motionmapperpy tutorial (motionmapperpy/demo/motionmapperpy_mouse_demo.ipynb).

Predicting kinematics from state sequences

We trained decoding models based on spline regression to predict kinematic parameters (height, velocity, turn speed) from state sequences output by keypoint-MoSeq and other behavior segmentation methods (Fig 4e, Extended Data Fig 4c). Let z_t represent an unsupervised behavioral state sequence and let B denote a spline basis, where $B_{t,i}$ is the value of spline i and frame t. We generated such a basis using the “bs” function from the python package “patsy”, passing in five log-spaced knot locations (1.0, 2.0, 3.9 , 7.7 , 15.2, 30.0) and obtaining basis values over a 300-frame

interval. This resulted in a 300-by-5 basis matrix B. The spline basis and state sequence were combined to form a $5N$-dimensional design matrix, where N is the number of distinct behavioral states. Specifically, for each instance $(t_s, \ldots, t_e)$ of state n (see "Cross-syllable likelihoods" section above for a definition of state instances), we inserted the first $t_e - t_s$ frames of B into dimensions $5n, \ldots, 5n + 5$ of the design matrix, alinging the first frame of B to frame t_s in the design matix. Kinematic features were regressed against the design matrix using Ridge regression from scikit-learn and 5-fold cross-validation. We used a range of values from 10^{-3} to 10^{3} for the regularization parameter α and reported the results with greatest accuracy.

Rearing analysis

To compare the dynamics of rear-associated states across methods, we systematically identified all instances of rearing in our initial open field dataset. During a stereotypical rear, mice briefly stood on their hindlegs and extended their head upwards, leading to a transient increase in height from its modal value of 3cm-5cm to a peak of 7cm-10cm. Rears were typically brief, with mice exiting and then returning to a prone position within a few seconds. We encoded these features using the following criteria. First, rear onsets were defined as increases in height from below 5cm to above 7cm that occurred within the span of a second, with onset formally defined as the first frame where the height exceeded 5cm. Next, rear offsets were defined as decreases in height from above 7cm to below 5cm that occurred within the span of a second, with offset formally defined as the first frame where the height fell below 7cm. Finally, we defined complete rears as onset-offset pairs defining an interval with length between 0.5 and 2 seconds. Height was determined from the distribution of depth values in cropped, aligned and background-segmented videos. Specifically, we used the 98th percentile of the distribution in each frame.

Accelerometry processing

From the IMU we obtained absolute rotations r_y, r_p, r_r (yaw, pitch, and roll) and accelerations a_x, a_y, a_z (dorsal/ventral, posterior/anterior, left/right). To control for subtle variations in implant geometry and chip calibration, we centered the distribution of sensor readings for each variable on each session. We defined total acceleration as the norm of the 3 acceleration components:

$$|a| = \sqrt{a_x^2 + a_y^2 + a_z^2}$$

Similarly, we defined total angular velocity as the norm $|\omega|$ of rotation derivative:

$$\omega = \left(\frac{dr_y}{dt}, \frac{dr_p}{dt}, \frac{dr_r}{dt}\right)$$

Finally, to calculate jerk, we smoothed the acceleration signal with a 50ms Gaussian kernel, generating a time-series $\tilde{a}$, and then computed the norm of its derivative:

$$\text{jerk} = \left|\frac{d\tilde{a}}{dt}\right|$$

Aligning dopamine fluctuations to behavior states

For a detailed description of photometry data acquisition and preprocessing, see ref[1]. Briefly, photometry signals were: (1) ΔF/F0-normalized using a 5-second window; (2) adjusted against a reference to remove motion artefacts and other non-ligand-associated fluctuations; (3) z-scored using a 20-second sliding window; (4) temporally aligned to the 30Hz behavioral videos.

Given a set of state onsets (either for a single state or across all states), we computed the onset-aligned dopamine trace by averaging the dopamine signal across onset-centered windows. From the resulting traces, each of which can be denoted as a time-series of dopamine signal values $(d_{-T}, \ldots, d_T)$ we defined the total fluctuation size (Fig 5d) and temporal asymmetry (Fig 5e) as

$$\text{temporal asymmetry} = \frac{1}{15}\sum_{t=0}^{15} d_t - \frac{1}{15}\sum_{t=-15}^{0} d_t\,, \qquad \text{AUC} = \sum_{t=-15}^{15} |d_t|$$

A third metric – the average dopamine during each state (Extended Data Figure 6b) – was defined simply as the mean of the dopamine signal across all frames bearing that state label. For each metric, shuffle distributions were generated by repeating the calculation with a temporally reversed copy of the dopamine times-series.

Supervised behavior benchmark

Videos and behavioral annotations for the supervised open field behavior benchmark (Fig 4a-c) were obtained from (Bohnslav, 2019)[11]. The dataset contains 20 videos that are each 10-20 minutes long. Each video includes frame-by-frame annotations of five possible behaviors: locomote, rear, face groom, body groom, and defacate. We excluded “defacate” from the analysis because it was extremely rate (< 0.1% of frames).

For pose tracking we used DLC's SuperAnimal inference API that performs inference on

videos without the need to annotate poses in those videos. Specifically, we used SuperAnimal-TopViewMouse that applies DLCRNet-50 as the pose estimation model11. Keypoint detections were obtained using DeepLabCut's API function deeplabcut.video_inference_superanimal. The API function uses a pretrained model called SuperAnimal-TopViewMouse and performs video adaptation that applies multi-resolution ensemble (i.e. the image height resized to 400, 500, 600 with a fixed aspect ratio) and rapid self-training (model trained on zeroshot predictions with confidence above 0.1) for 1000 iterations to counter domain shift and reduce jittering predictions. The code to reproduce this analysis is:

```
videos = ['path_to_video']
superanimal_name = 'superanimal_topviewmouse'
scale_list = [400, 500, 600]

deeplabcut.video_inference_superanimal(videos,
  superanimal_name,
  videotype=".mp4",
  video_adapt = True,
  scale_list = scale_list)
```

Keypoint coordinates and behavioral annotations for the supervised social behavior benchmark (Fig 4d-f) were obtained from the CalMS21 dataset[12] (task1). The dataset contains 70 videos of resident-intruder interactions with frame-by-frame annotations of four possible behaviors: attack, investigate, mount, or other. All unsupervised behavior segmentation methods were fit to 2D keypoint data for the resident mouse.

We used four metrics[9] to compare supervised annotations and unsupervised states from each method. These included normalized mutual information, homogeneity, adjusted rand score, and purity. All metrics besides purity were computed using the python library scikit-learn (i.e. with the function normalized_mutual_info_score, homogeneity_score, adjusted_rand_score). The purity score was defined as in ref[9].

Supplemental References

1 Markowitz, J. E. *et al.* Spontaneous behaviour is structured by reinforcement without explicit reward. *Nature* **614**, 108-117 (2023). https://doi.org:10.1038/s41586-022-05611-2

2 Lin, S. *et al.* Characterizing the structure of mouse behavior using Motion Sequencing. (2022). https://doi.org:10.48550/ARXIV.2211.08497

3 Zhou, Z., Rahman Siddiquee, M. M., Tajbakhsh, N. & Liang, J. in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support.* (eds Danail Stoyanov *et al.*) 3-11 (Springer International Publishing).

4 Sun, K., Xiao, B., Liu, D. & Wang, J. in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* 5686-5696.

5 Zhang, L., Dunn, T., Marshall, J., Olveczky, B. & Linderman, S. in *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics* Vol. 130 (eds Banerjee Arindam & Fukumizu Kenji) 2800--2808 (PMLR, Proceedings of Machine Learning Research, 2021).

6 Nath, T. *et al.* Using DeepLabCut for 3D markerless pose estimation across species and behaviors. *Nature Protocols* **14**, 2152-2176 (2019). https://doi.org:10.1038/s41596-019-0176-0

7 Wiltschko, A. B. *et al.* Mapping Sub-Second Structure in Mouse Behavior. *Neuron* **88**, 1121-1135 (2015).

8 Hsu, A. I. & Yttri, E. A. B-SOiD, an open-source unsupervised algorithm for identification and fast prediction of behaviors. *Nature Communications* **12**, 5188 (2021). https://doi.org:10.1038/s41467-021-25420-x PMID - 34465784

9 Luxem, K. *et al.* Identifying behavioral structure from deep variational embeddings of animal motion. *Commun Biol* **5**, 1267 (2022). https://doi.org:10.1038/s42003-022-04080-7

10 Berman, G. J., Choi, D. M., Bialek, W. & Shaevitz, J. W. Mapping the stereotyped behaviour of freely moving fruit flies. *Journal of the Royal Society, Interface / the Royal Society* **11** (2014). https://doi.org:papers3://publication/doi/10.1098/rsif.2014.0672

11 Bohnslav, J. P. *et al.* DeepEthogram, a machine learning pipeline for supervised behavior classification from raw pixels. *eLife* **10**, e63377 (2021). https://doi.org:10.7554/eLife.63377

12 Sun, J. J. *et al.* Caltech Mouse Social Interactions (CalMS21) Dataset. (2021). https://doi.org:10.22002/D1.1991

Keypoint MoSeq mathematical model

1 Generative model

Switching Linear Dynamical System We previously found that mouse behavior evolves through a sequence of short motifs that we call syllables [4]. Each motif is characterized by a stable pattern of movement dynamics, and the boundaries between motifs correspond to sudden changes in these dynamics. Switching linear dynamical systems (SLDS) are well-suited to identify these dynamical patterns and the transitions between them. SLDS models the observed data using a lower-dimensional latent time-series with linear autoregressive dynamics. The dynamics switch over time, and the sequence of transitions are assumed to form a Markov chain. Formally, if y_t, x_t, and z_t represent the observed state, continuous latent state, and discrete latent state at time t, SLDS defines the following generative model.

$$z_t \sim \text{Categorical}(\pi_{z_{t-1}}) \tag{1}$$

$$x_t \sim \mathcal{N}(A^{(z_t)}[x_{t-L}^\top, ..., x_{t-1}^\top]^\top + b^{(z_t)}, Q^{(z_t)}) \tag{2}$$

$$y_t \sim \mathcal{N}(Cx_t + d, S_t) \tag{3}$$

In depth-based MoSeq, y_t represents pixel values in a 3D depth map, and C, d, x_t are fixed ahead of time using PCA [4]. Here we relax these constraints and fit x_t simultaneously with the rest of the model. We use a matrix normal inverse Wishart (MNIW) prior for the autoregressive parameters

$$(A^{(i)}, b^{(i)}), Q^{(i)} \sim \text{MNIW}(\nu_0, S_0, M_0, K_0) \tag{4}$$

and a sticky hierarchical Dirichlet process (HDP) prior for the transition matrix π, taking the weak limit with maximum number of states N. In practice, this means π is generated via stacked Dirichlet distributions as follows.

$$\beta \sim \text{Dir}(\gamma/N, ..., \gamma/N) \tag{5}$$

$$\pi_i \sim \text{Dir}(\alpha\beta_1, ..., \alpha\beta_j + \kappa\delta_{ij}, ..., \alpha\beta_N) \tag{6}$$

The hyperparameters γ, α, κ control the sparsity of states, the weight of the sparsity prior, and the bias toward self-transitions respectively. The prior for S_t is given in Eq. 11. Note that in general, SLDS models can have state dependent observation parameters (C, d, S) and time-varying input signals. We have chosen to omit these features.

Keypoint-adapted SLDS When clustering kinematics into distinct behavior motifs, one usually wishes to ignore absolute location and heading angle. Common approaches for such affine invariance include centering and aligning data ahead of time (as in VAME [3]), or using relative distances or angles (as in B-SOiD [2]). One issue with these approaches is that they induce spurious correlations between variables. When a single keypoint moves, it may shift the egocentric reference frame, or perturb multiple distances and/or angles. To avoid this problem, we use an approach developed by Zhang et al., for the pose-inference tool GIMBAL [5]: We model the animal's centroid and heading explicitly, and then combine these variables with pose to predict keypoint locations in absolute coordinates.

Concretely, let $Y_t \in \mathbb{R}^{K \times D}$ represent the coordinates of K keypoints at time t, where $D \in \{2, 3\}$. Define latent variables $v_t \in \mathbb{R}^D$ and $h_t \in [0, 2\pi]$ to represent the animal's location and heading angle. At each time point, the pose Y_t is generated via rotation and translation of a centered and oriented pose $\tilde{Y}_t$ that depends on the current latent state x_t, as follows, where $R(h_t)$ is a matrix that rotates by angle h_t in the xy-plane.

$$Y_t = \tilde{Y}_t R(h_t) + \mathbf{1}_K v_t^\top \quad \text{where} \quad \text{vec}(\tilde{Y}_t) \sim \mathcal{N}((\Gamma \otimes I_D)(Cx_t + d), S_t) \tag{7}$$

The matrix Γ is defined by the singular value decomposition $\Gamma \Delta \Gamma^\top = I_K - \mathbf{1}_{K \times K}/K$, and ensures that $\mathbb{E}(\tilde{Y}_t)$ is always centered [1]. Γ encodes a linear transformation that isometrically maps $\mathbb{R}^{(K-1) \times D}$ to the set of all centered keypoint arrangements in $\mathbb{R}^{K \times D}$. The elements of C, d have iid priors $\mathcal{N}(0, \sigma_C^2)$ and each angle h_t has an independent uniform prior. We assume that the translations are autocorrelated as follows, where σ_{loc} is a hyper-parameter.

$$v_t \sim \mathcal{N}(v_{t-1}, \sigma_{\text{loc}}^2) \tag{8}$$

Robust Observations To account for occasional large errors in keypoint tracking data, we use a (heavy-tailed) Student's-t distribution for keypoint coordinates, and assume that the noise in these coordinates is independent and isotropic for each keypoint. These assumptions are encoded in the following generative model for S_t (defined in Eq 3).

$$\sigma_k \sim \chi^{-2}(\nu_\sigma, \sigma_0^2) \tag{9}$$
$$s_{t,k} \sim \chi^{-2}(\nu_s, s_0) \tag{10}$$
$$S_t = \text{diag}(\sigma_1^2 s_{t,1} ... \sigma_K^2 s_{t,K}) \otimes I_D \tag{11}$$

2 Hyper-parameters

We used the following hyper-parameter values throughout the paper.

Transition matrix

$$N = 100 \tag{12}$$
$$\gamma = 1000 \tag{13}$$
$$\alpha = 100 \tag{14}$$
$$\kappa \ \text{ fit to each dataset} \tag{15}$$

Autoregressive process Let $m = \dim(x)$ and $L = 3$

$$\nu_0 = \dim(x) + 2 \tag{16}$$
$$S_0 = 0.01 I_m \tag{17}$$
$$M_0 = \begin{bmatrix} 0_{m \times (L-1)} & I_m & 1_{m \times 1} \end{bmatrix} \tag{18}$$
$$K_0 = 10 I_{m*L+1} \tag{19}$$

Observation process

$$\sigma_0^2 = 1 \tag{20}$$
$$\nu_\sigma = 10^5 \tag{21}$$
$$\nu_s = 5 \tag{22}$$
$$s_0 \ \text{ set based on neural network confidence} \tag{23}$$

Animal position

$$\sigma^2_{\text{loc}} = 0.4 \tag{24}$$

3 Inference algorithm

Our full model contains latent variables v, h, x, z, s and parameters $A, b, Q, C, d, \sigma, \beta, \pi$. We fit each of these variables – with the exception of (C, d) using Gibbs sampling, wherein each variable is resampled from its posterior distribution conditional on all the other variables and on the data $Y_1, ..., Y_T$. Here we describe Gibbs updates for (C, d) even though these variables are fixed at their initial values (learned from PCA) for keypoint-MoSeq.

The posterior distributions $P(\pi, \beta \mid z)$ and $P(A, b, Q \mid z, x)$ are unchanged from the original MoSeq paper and will not be be reproduced here (see ref [4], pages 42-44, and note the changes of notation $Q \to \Sigma$, $z \to x$, and $x \to y$). The Gibbs updates for C, d, σ, s, v and h are described below.

$\boldsymbol{P(C, d \mid s, \sigma, x, v, h, Y)}$ Let $\tilde{x}_t$ represent x_t with a 1 appended and define

$$\tilde{S}_t = (\Gamma^\top \text{diag}(\sigma_1^2 s_{t,1}, ..., \sigma_K^2 s_{t,K})\Gamma) \otimes I_D \tag{25}$$

The posterior update is $(C, d) \sim \mathcal{N}(\text{vec}(C, d) \mid \mu_n, \Sigma_n)$ where

$$\Sigma_n = (\sigma_C^{-2} I + S_{x,x})^{-1} \qquad \text{and} \qquad \mu_n = \Sigma_n S_{y,x} \tag{26}$$

with

$$S_{x,x} = \sum_{t=1}^{T} \tilde{x}_t \tilde{x}_t^\top \otimes \Gamma^\top \tilde{S}_t^{-1} \Gamma \otimes I_D \qquad \text{and} \qquad S_{y,x} = \sum_{t=1}^{T} (\tilde{x}_t^\top \otimes \tilde{S}^{-1} \Gamma \otimes I_D) \text{vec}(\tilde{Y}_t)^\top \tag{27}$$

$\boldsymbol{P(s \mid C, d, \sigma, x, v, h, Y)}$ Each $s_{t,k}$ is conditionally independent with posterior

$$s_{t,k} \mid C, d, \sigma_k, x, Y \sim \chi^{-2}\big(\nu_s + D,\ (\nu_s s_0 + \sigma_k^{-2} \|\Gamma(Cx_t + d)_k - \tilde{Y}_{t,k}\|^2)/(\nu_s + D)\big) \tag{28}$$

$\boldsymbol{P(\sigma \mid C, d, s, x, v, h, Y)}$ Each σ_k is conditionally independent with posterior

$$\sigma_k^2 \sim \chi^{-2}\big(\nu_\sigma + DT,\ (\nu_\sigma \sigma_0^2 + S_y)(\nu_\sigma + DT)^{-1}\big) \tag{29}$$

where $S_y = \sum_{t=1}^{N} \|\Gamma(Cx_t + d)_k - \tilde{Y}_{t,k}\|^2 / s_{t,k}$

$\boldsymbol{P(v \mid C, d, \sigma, s, x, h, Y)}$ Since the translations $v_1, ..., v_T$ form a linear dynamical system, they can be updated by Kalman sampling. The transitions are defined by Eq. 8 and the observation potentials have the form $\mathcal{N}(v_t \mid \mu, \gamma^2 I_D)$ where

$$\mu = \sum_k \frac{\gamma_t^2}{\sigma_k^2 s_{t,k}} \big[Y_{t,k} - R(h_t)^\top \Gamma(Cx_t + d)_k\big], \quad \frac{1}{\gamma_t^2} = \sum_k \frac{1}{\sigma_k^2 s_{t,k}} \tag{30}$$

$\boldsymbol{P(h \mid C, d, \sigma, s, x, v, Y)}$ The posterior of h_t is the von-Mises distribution $\text{vM}(\theta, \kappa)$ where κ and $\theta \in [0, 2\pi]$ are the unique parameters satisfying $[\kappa \cos(\theta), \kappa \sin(\theta)] = [S_{1,1} + S_{2,2}, S_{1,2} - S_{2,1}]$ for

$$S = \sum_k \frac{1}{s_{t,k} \sigma_k^2} \Gamma(Cx_t + d)_k (Y_{t,k} - v_t)^\top \tag{31}$$

$P(x \mid C, d, \sigma, s, v, h, Y)$ To resample x, we first express its temporal dependencies as a first-order autoregressive process, and then apply Kalman sampling. The change of variables is

$$A' = \begin{bmatrix} & I & & \\ & & I & \\ & & & I \\ A_1 & A_2 & \cdots & A_L & b \end{bmatrix} \quad Q' = \begin{bmatrix} 0 & & & \\ & 0 & & \\ & & 0 & \\ & & & Q \end{bmatrix} \quad C' = \begin{bmatrix} 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ C & d \end{bmatrix} \quad x'_t = \begin{bmatrix} x_{t-L+1} \\ \vdots \\ x_t \\ 1 \end{bmatrix} \tag{32}$$

Kalman sampling can then be applied to the sample the conditional distribution,

$$P(x'_{1:T} \mid \tilde{Y}_{1:T}) \propto \prod_{t=1}^{T} \mathcal{N}(x'_t \mid A'^{(z_t)} x'_{t-1}, Q'^{(z_t)}) \; \mathcal{N}(\mathrm{vec}(\tilde{Y}_t) \mid C' x'_t, S_t). \tag{33}$$

(Assume x' is left-padded with zeros for negative time indices.)

4 Derivation of Gibbs updates

Derivation of C, d updates To simply notation, define

$$\tilde{S}_t = \mathrm{diag}(\sigma_1^2 s_{t,1}, ..., \sigma_K^2 s_{t,K}), \quad \tilde{x}_t = (x_t, 1), \quad \tilde{C} = (C, d) \tag{34}$$

The likelihood of the centered and aligned keypoint locations $\tilde{Y}$ can be expanded as follows.

$$P(\tilde{Y} \mid \tilde{C}, \tilde{x}, \tilde{S}) = \prod_{t=1}^{T} \mathcal{N}(\mathrm{vec}(\tilde{Y}_t) \mid (\Gamma \otimes I_D)\tilde{C}\tilde{x}_t, \; \tilde{S}_t \otimes I_D) \tag{35}$$

$$\propto \exp\Bigg[-\frac{1}{2} \sum_{t=1}^{T} \big(\tilde{x}_t^\top \tilde{C}^\top (\Gamma^\top \tilde{S}_t^{-1} \Gamma \otimes I_D) \tilde{C} \tilde{x}_t - 2\mathrm{vec}(\tilde{Y}_t)^\top (\tilde{S}_t^{-1} \Gamma \otimes I_D) \tilde{C} \tilde{x}_t \big) \tag{36}$$

$$\propto \exp\Bigg[-\frac{1}{2} \sum_{t=1}^{T} \big(\mathrm{vec}(\tilde{C})^\top (\tilde{x}_t \tilde{x}_t^\top \otimes \Gamma^\top \tilde{S}_t^{-1} \Gamma \otimes I_D) \mathrm{vec}(\tilde{C}) \tag{37}$$

$$- 2\mathrm{vec}(\tilde{C})^\top (\tilde{x}_t^\top \otimes \tilde{S}_t^{-1} \Gamma \otimes I_D) \mathrm{vec}(\tilde{Y}_t) \big) \Bigg] \tag{38}$$

$$\propto \exp\Bigg[-\frac{1}{2} \big(\mathrm{vec}(\tilde{C})^\top S_{x,x} \mathrm{vec}(\tilde{C}) - 2\mathrm{vec}(\tilde{C})^\top S_{x,y} \big) \Bigg] \tag{39}$$

where

$$S_{x,x} = \sum_{t=1}^{T} \tilde{x}_t \tilde{x}_t^\top \otimes \Gamma^\top \tilde{S}_t^{-1} \Gamma \otimes I_D \quad \text{and} \quad S_{x,y} = \sum_{t=1}^{T} (\tilde{x}_t^\top \otimes \tilde{S}^{-1} \Gamma \otimes I_D) \mathrm{vec}(\tilde{Y}_t) \tag{40}$$

Multiplying Eq 39 by the prior $\mathrm{vec}(\tilde{C}) \sim \mathcal{N}(0, \sigma_C^2 I)$ yields

$$P(\tilde{C} \mid \tilde{Y}, \tilde{x}, \tilde{S}) \propto \mathcal{N}(\mathrm{vec}(\tilde{C}) \mid \mu_n, \Sigma_n) \tag{41}$$

where

$$\Sigma_n = (\sigma_C^{-2} I + S_{x,x})^{-1} \quad \text{and} \quad \mu_n = \Sigma_n S_{y,x} \tag{42}$$

Derivation of $\boldsymbol{\sigma_k, s_{t,k}}$ updates For each time t and keypoint k, let $\overline{Y}_{t,k} = \Gamma(Cx_t + d)$. The likelihood of the centered and aligned keypoint location $\tilde{Y}_{t,k}$ is

$$P(\tilde{Y}_{t,k} \mid \overline{Y}_{t,k}, s_{t,k}, \sigma_k) = \mathcal{N}(\tilde{Y}_{t,k} \mid \overline{Y}_{t,k},\ \sigma_k^2 s_{t,k} I_D) \propto (\sigma_k^2 s_{t,k})^{-D/2} \exp\left[-\frac{\|\tilde{Y}_{t,k} - \bar{Y}_{t,k}\|^2}{2\sigma_k^2 s_{t,k}} \right] \tag{43}$$

We can then calculate posteriors $P(s_{t,k} \mid \sigma_k)$ and $P(\sigma_k \mid s_{t,k})$ as follows.

$$P(s_{t,k} \mid \sigma_k, \tilde{Y}_{t,k}, \overline{Y}_{t,k}) \propto \chi^{-1}(s_{t,k} \mid \nu_s, s_0)\mathcal{N}(\tilde{Y}_{t,k} \mid \overline{Y}_{t,k},\ \sigma_k^2 s_{t,k} I_D) \tag{44}$$

$$\propto s_{t,k}^{-1-(\nu_s+D)/2} \exp\left[\frac{-\nu_s s_0}{2s_{t,k}} - \frac{\|\tilde{Y}_{t,k} - \bar{Y}_{t,k}\|^2}{2\sigma_k^2 s_{t,k}} \right] \tag{45}$$

$$\propto \chi^{-2}\big(s_{t,k} \mid \nu_s + D,\ (\nu_s s_0 + \sigma_k^{-2}\|\tilde{Y}_{t,k} - \bar{Y}_{t,k}\|^2)(\nu_s + D)^{-1}\big) \tag{46}$$

$$P(\sigma_k \mid \{s_{t,k}, \tilde{Y}_{t,k}, \overline{Y}_{t,k}\}_{t=1}^T) \propto \chi^{-1}(\sigma_k^2 \mid \nu_\sigma, \sigma_0^2) \prod_{t=1}^T \mathcal{N}(\tilde{Y}_{t,k} \mid \overline{Y}_{t,k},\ \sigma_k^2 s_{t,k} I_D) \tag{47}$$

$$\propto \sigma_k^{-2-\nu_\sigma - DT} \exp\left[\frac{-\nu_\sigma \sigma_0^2}{2\sigma_k^2} - \frac{1}{2\sigma_k^2} \sum_{t=1}^T \frac{\|\tilde{Y}_{t,k} - \bar{Y}_{t,k}\|^2}{s_{t,k}} \right] \tag{48}$$

$$\propto \chi^{-2}\big(\sigma_k^2 \mid \nu_\sigma + DT,\ (\nu_\sigma \sigma_0^2 + S_y)(\nu_\sigma + DT)^{-1}\big) \tag{49}$$

where $S_y = \sum_t \|\tilde{Y}_{t,k} - \bar{Y}_{t,k}\|^2 / s_{t,k}$

Derivation of $\boldsymbol{v_t}$ update We assume an improper uniform prior on v_t, hence

$$P(v_t \mid Y_t) \propto P(Y_t \mid v_t)P(v_t) \propto P(Y_t \mid v_t) \tag{50}$$

$$\propto \mathcal{N}\big(\mathrm{vec}((Y_t - \mathbf{1}_K v_t^\top)R(h_t)^\top) \mid \Gamma(Cx_t + d),\ S_t\big) \tag{51}$$

$$= \prod_k \mathcal{N}(R(h_t)(Y_{t,k} - v_t) \mid \Gamma(Cx_t + d)_k,\ s_{t,k}\sigma_k^2 I_D\big) \tag{52}$$

$$= \prod_k \mathcal{N}(v_t \mid Y_{t,k} - R(h_t)^\top \Gamma(Cx_t + d)_k,\ s_{t,k}\sigma_k^2 I_D\big) \tag{53}$$

$$= \mathcal{N}(v_t \mid \mu_t, \gamma_t^2 I_D) \tag{54}$$

where

$$\mu = \sum_k \frac{\gamma_t^2}{\sigma_k^2 s_{t,k}} (Y_{t,k} - R(h_t)^\top \Gamma(Cx_t + d)_k), \quad \frac{1}{\gamma_t^2} = \sum_k \frac{1}{\sigma_k^2 s_{t,k}} \tag{55}$$

Derivation of $\boldsymbol{h_t}$ update We assume a proper uniform prior on h_t, hence

$$P(h_t \mid Y_t) \propto P(Y_t \mid h_t)P(h_t) \propto P(Y_t \mid h_t) \tag{56}$$

$$\propto \exp\left[\sum_k \frac{(Y_{t,k} - v_t)^\top R(h_t)\Gamma(Cx_t + d)_k}{s_{t,k}\sigma_k^2} \right] \tag{57}$$

$$= \exp\left[\frac{\mathrm{tr}[R(h_t)\Gamma(Cx_t + d)_k (Y_{t,k} - v_t)^\top]}{s_{t,k}\sigma_k^2} \right] \tag{58}$$

$$\propto \exp \mathrm{tr}[R(h_t)S] \quad \text{where} \quad S = \sum_k \Gamma(Cx_t + d)_k (Y_{t,k} - v_t)^\top / (s_{t,k}\sigma_k^2) \tag{59}$$

$$\propto \exp[\cos(h_t)(S_{1,1} + S_{2,2}) + \sin(h_t)(S_{1,2} - S_{2,1})] \tag{60}$$

Let $[\kappa\cos(\theta), \kappa\sin(\theta)]$ represent $[S_{1,1} + S_{2,2}, S_{1,2} - S_{2,1}]$ in polar coordinates. Then

$$P(Y_t \mid h_t) \propto \exp[\kappa\cos(h_t)\cos(\theta) + \sin(h_t)\sin(\theta)] \tag{61}$$

$$= \exp[\kappa\cos(h_t - \theta)] \propto \mathrm{vM}(h_t \mid \theta, \kappa) \tag{62}$$

References

[1] Angela Andreella and Livio Finos. Procrustes analysis for high-dimensional data. *Psychometrika*, 87(4):1422–1438, Dec 2022.

[2] Alexander I. Hsu and Eric A. Yttri. B-soid, an open-source unsupervised algorithm for identification and fast prediction of behaviors. *Nature Communications*, 12(1):5188, 2021.

[3] Kevin Luxem, Petra Mocellin, Falko Fuhrmann, Johannes Kürsch, Stephanie R. Miller, Jorge J. Palop, Stefan Remy, and Pavol Bauer. Identifying behavioral structure from deep variational embeddings of animal motion. *Communications Biology*, 5(1):1267, Nov 2022.

[4] Alexander B Wiltschko, Matthew J Johnson, Giuliano Iurilli, Ralph E Peterson, Jesse M Katon, Stan L Pashkovski, Victoria E Abraira, Ryan P Adams, and Sandeep Robert Datta. Mapping sub-second structure in mouse behavior. *Neuron*, 88(6):1121–1135, 12 2015.

[5] Libby Zhang, Tim Dunn, Jesse Marshall, Bence Olveczky, and Scott Linderman. Animal pose estimation from video data with a hierarchical von mises-fisher-gaussian model. In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 2800–2808. PMLR, 13–15 Apr 2021.