nature methods

Article

https://doi.org/10.1038/s41592-024-02318-2

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

Received: 5 April 2023

Accepted: 22 May 2024

Published online: 12 July 2024

Check for updates

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

Keypoint tracking algorithms can flexibly quantify animal movement from videos obtained in a wide variety of settings. However, it remains unclear how to parse continuous keypoint data into discrete actions. This challenge is particularly acute because keypoint data are susceptible to high-frequency jitter that clustering algorithms can mistake for transitions between actions. 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 identify syllables whose boundaries correspond to natural sub-second discontinuities in pose dynamics. 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 also works in multiple species and generalizes beyond the syllable timescale, identifying fast sniff-aligned movements in mice and a spectrum of oscillatory behaviors in fruit flies. Keypoint-MoSeq, therefore, renders accessible the modular structure of behavior through standard video recordings.

Work from ethology demonstrates that behavior—a chain of actions traced by the body's movement over time—is both continuous and discrete[1–3]. The rapid advance of keypoint tracking methods (including SLEAP[4], DeepLabCut[5] and others[6,7]) has given researchers broad access to the continuous dynamics that underlie animal behavior[8]. But parsing these dynamics into chains of discrete actions remains an open problem[9–11]. While several action segmentation approaches exist[12–17], their underlying logic and assumptions differ, with different methods often giving distinct descriptions of the same behavior[13,15]. An important gap, therefore, exists between our access to movement kinematics and our ability to understand their underlying structure.

One method for parsing behavior in mice is Motion Sequencing (MoSeq)[16,18–21]. MoSeq uses unsupervised machine learning to transform its inputs—which are not keypoints, but three-dimensional (3D) depth videos—into a set of behavioral motifs (like rears, turns and pauses) called syllables. To identify syllables, MoSeq searches for

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

https://doi.org/10.1038/s41592-024-02318-2

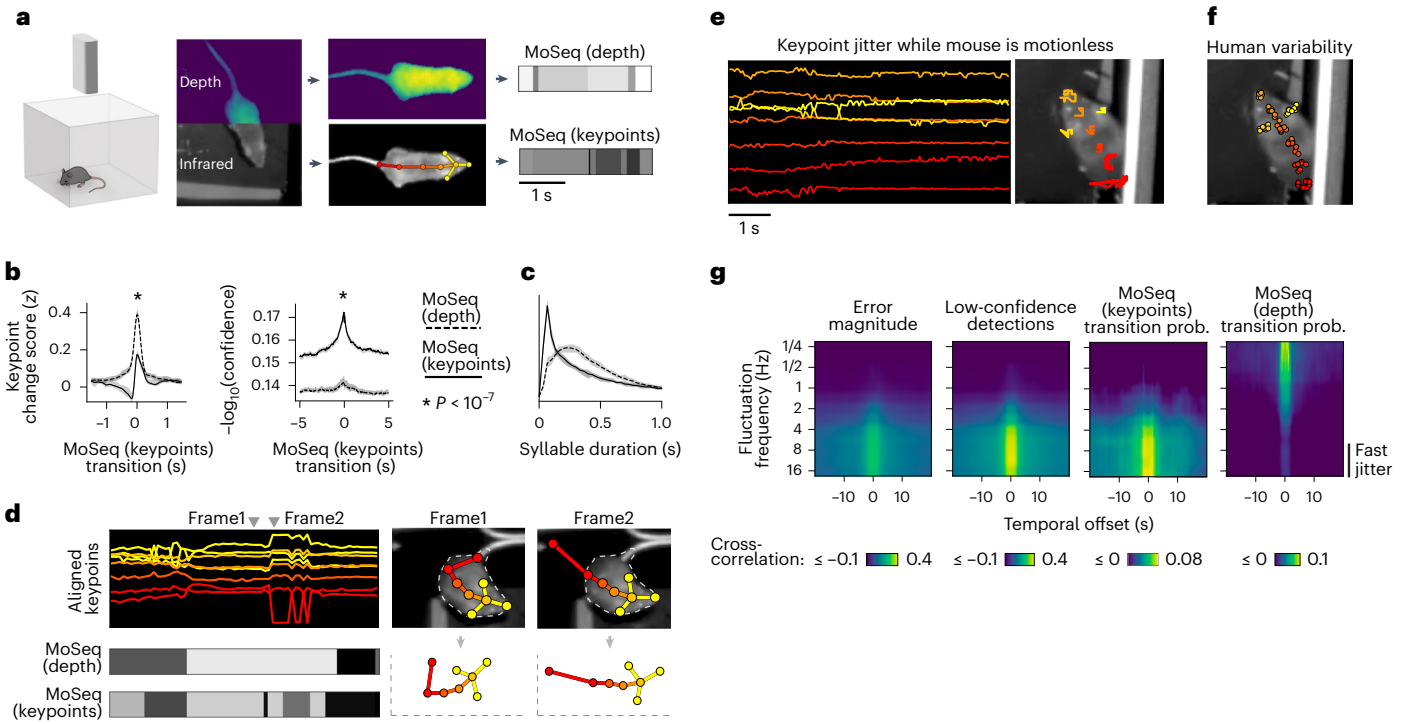


Fig. 1 | Keypoint trajectories exhibit sub-second structure. a, Left: simultaneous depth and 2D infrared (IR) recording setup. Middle: pose representations using the depth data (top) or IR (bottom, tracked keypoints indicated). Right: Example syllable sequences from MoSeq applied to depth data (referred to as 'MoSeq (depth)') or to keypoint data (referred to as 'MoSeq (keypoints)'). Figure created with SciDraw under a CC BY 4.0 license. **b**, Keypoint change scores or low-confidence detection scores, relative to the onset of MoSeq transitions (x axis) derived from either depth (gray) or keypoint (black) data. Differences in each case were significant ($P = 2 \times 10^{-7}$ over $N = 20$ model fits, Mann–Whitney U test; plots show mean and range across model fits). **c**, Comparison of syllable durations for MoSeq (keypoints) and MoSeq (depth), showing mean and inter-95% confidence interval range across $N = 20$ model fits. **d**, Left: keypoint detection errors, including high-frequency fluctuations in keypoint coordinates (top row) and error-induced syllable switches (bottom row). Right: keypoint coordinates before (frame1) and during (frame2) an example keypoint detection error. This error (occurring in the tail keypoint) causes a shift in egocentric alignment, hence changes across the other tracked keypoints. **e**, 5-s interval during which the mouse is immobile yet the keypoint coordinates fluctuate. Left: egocentrically aligned keypoint trajectories. Right: path traced by each keypoint during the 5-s interval. **f**, Variability in keypoint positions assigned by eight human labelers. **g**, Cross-correlation between various features and keypoint fluctuations at a range of frequencies. Each heat map represents a different scalar time series (such as 'transition probability'—the likelihood of a syllable transition on each frame). Each row shows the cross-correlation between that time series and the time-varying power of keypoint fluctuations at a given frequency.

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 extensively applied, pervasive discontinuities at the sub-second-to-second timescale mark boundaries between syllables, and the stickiness hyperparameter is explicitly set to capture this timescale[16].

Previous studies have applied MoSeq to characterize the effects of genetic mutations, drugs, neural manipulations and changes in the sensory or physical environment[16,22–24]. MoSeq syllables are encoded in the dorsolateral striatum (DLS)—an area important for action selection—and can be individually reinforced through closed-loop dopamine stimulation[22,23], arguing that MoSeq-identified syllables are meaningful units of behavior used by the brain to organize action sequences. But MoSeq's reliance on depth cameras is a substantial constraint; depth cameras are difficult to deploy, suffer from high sensitivity to reflections and have limited temporal resolution[25]. In principle, these limits could be overcome by applying MoSeq to keypoint data. But attempts to do so have thus far failed: researchers applying MoSeq-like models to keypoint data have reported flickering state sequences that switch much faster than the animal's actual behavior[13].

Here we confirm this finding and trace its cause to jitter in the keypoint estimates, which is mistaken by MoSeq for behavioral transitions. To address this challenge, we reformulated the model underlying MoSeq to simultaneously infer correct pose dynamics (from noisy or even missing data) and the behavioral syllables they represent. We validate this model, called keypoint-MoSeq, using accelerometry measurements, neural activity recordings and supervised behavior labels from expert observers, and show that it generalizes beyond mouse syllables to capture behaviors at multiple timescales and in several species. Because 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 parsing the structure of behavior. To facilitate broad adoption, we have directly integrated keypoint-MoSeq with widely used tracking methods (including SLEAP and DeepLabCut) and made the code freely accessible for academic users at http://www.moseq4all.org/.

Results

Mouse syllables are evident in depth-based video recordings as discontinuities of movement that reoccur with sub-second cadence[16]. To test if the same sub-second structure is present in keypoint data, we recorded conventional videos of mice exploring an open field arena and used a neural network to track eight keypoints (two ears and six points along the dorsal midline). We also captured simultaneous depth videos for comparison to depth-based MoSeq (Fig. 1a).

Similar sub-second discontinuities appeared in both the depth and keypoint data, with a keypoint-based change score (total velocity of keypoints after egocentric alignment) spiking at the transitions

https://doi.org/10.1038/s41592-024-02318-2

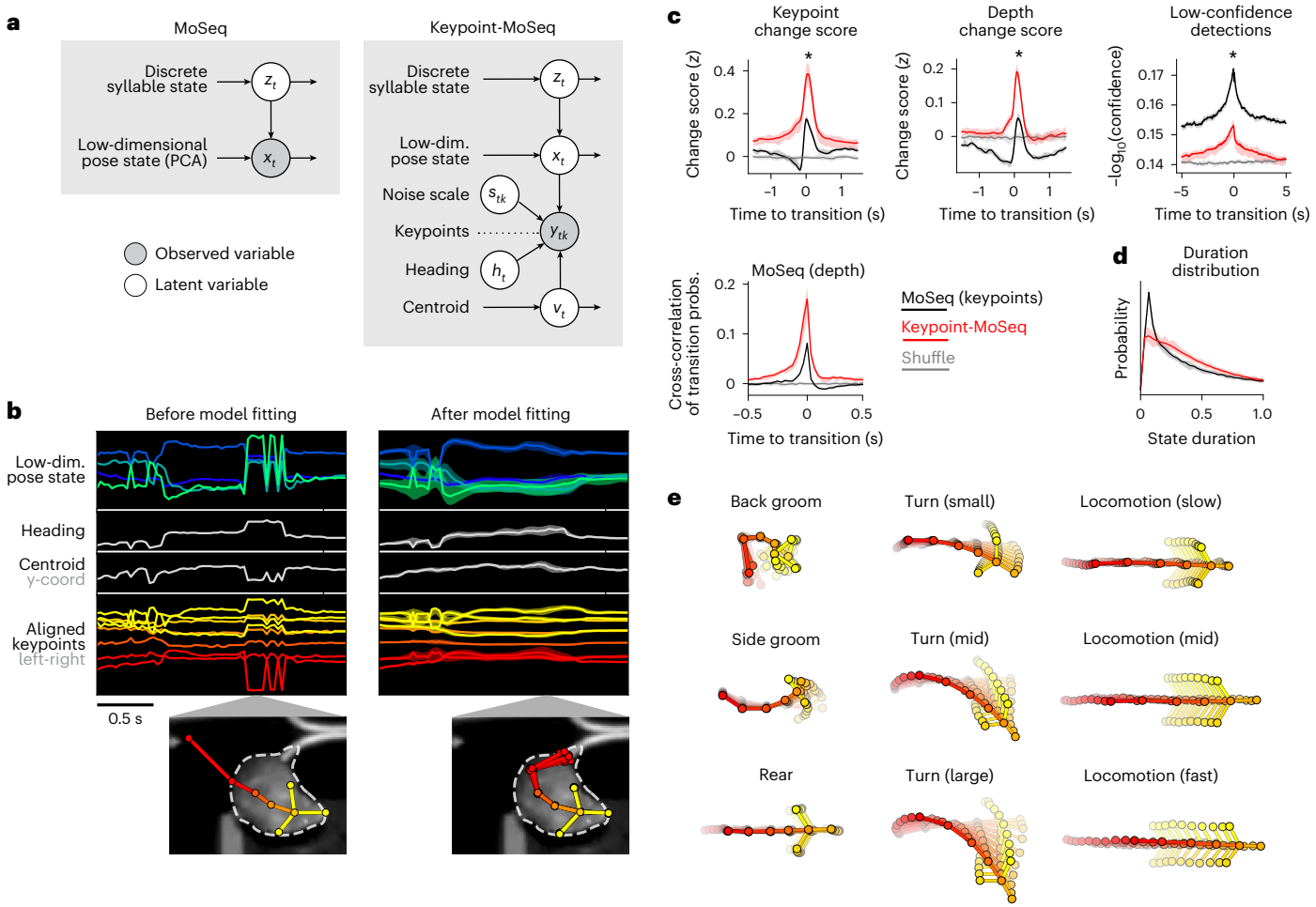


Fig. 2 | Hierarchical modeling of keypoint trajectories decouples noise from pose dynamics. a, Graphical models illustrating traditional MoSeq and keypoint-MoSeq. In both models, a discrete syllable sequence governs pose dynamics in a low-dimensional pose state; these pose dynamics are either described using principal component analysis (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 (*y* axis) are perturbed by incorrect positional assignment of the tail-base keypoint (whose erroneous location is shown in the bottom inset). Right: Keypoint-MoSeq infers plausible trajectories for each variable (shading represents the 95% confidence interval). The inset shows several likely keypoint coordinates for the tail-base inferred by the model. **c**, Top: various features averaged around syllable transitions from keypoint-MoSeq (red) versus traditional MoSeq applied to keypoint data (black), showing mean and inter-95% confidence interval range across $N = 20$ model fits. Bottom: cross-correlation of syllable transition probabilities between each model and depth MoSeq. Shaded regions indicate bootstrap 95% confidence intervals. Peak height represents the relative frequency of overlap in syllable transitions. Differences in each case were significant ($^*P = 2 \times 10^{-7}$ over $N = 20$ model fits, Mann–Whitney *U* test). **d**, Duration distribution of the syllables from each of the indicated models. Shading as in **c**. **e**, Average pose trajectories for example keypoint-MoSeq syllables. Each trajectory includes ten poses, starting 165 ms before and ending 500 ms after syllable onset.

between depth-based MoSeq syllables (Fig. 1b). Yet when we applied MoSeq directly to the keypoint data, it failed to recognize these discontinuities as syllable transitions, instead generating implausibly brief syllables that aligned poorly with the keypoint change score (Fig. 1b,c). These observations are consistent with prior work showing that MoSeq underperforms alternative clustering methods when applied to keypoints[13,26].

We wondered whether this poor performance could be explained by noise in the keypoint data, which might introduce subtle discontinuities that are falsely recognized by MoSeq as behavioral transitions. In our data, this noise took the form of high-frequency jitter that reflected errors in body part detection or rapid jumps in the inferred location of a stationary body part (Fig. 1d,e, Extended Data Fig. 1a,b and Supplementary Video 1). Much of the jitter—which was pervasive across camera angles and tracking methods—seemed to reflect inherent ambiguity in the true location of a keypoint, as frame-to-frame fluctuations in detected keypoint position had a similar scale as the variability in human labeling (Fig. 1f and Extended Data Fig. 1b–e). We confirmed that the jitter was unrelated to true movement by tracking the same body part using multiple cameras; although overall movement trajectories were almost identical across cameras, high-frequency fluctuations around those trajectories were uncorrelated, suggesting that the fluctuations are a tracking artifact (Extended Data Fig. 1f,g).

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. 1b,g). We were unable to correct this defect through simple smoothing: application of a low-pass filter—while removing jitter—also blurred true transitions, preventing MoSeq from identifying syllable boundaries (Extended Data Fig. 1h). Median filtering and Gaussian smoothing also yielded no improvement. These data reveal that high-frequency tracking noise prevents MoSeq from accurately segmenting behavior.

Hierarchical modeling decouples noise from behavior

Keypoint jitter contaminates MoSeq syllables because MoSeq assumes that each keypoint is a faithful representation of a point on the animal, and thus cannot distinguish noise from real behavior. To address this issue, we rebuilt MoSeq as a switching linear dynamical system (SLDS)—a class of model that explicitly disentangles signal from noise in time-series data[27,28]. This model—called 'keypoint-MoSeq'—has three hierarchical levels (Fig. 2a): a discrete state sequence that governs trajectories in a low-dimensional pose space, which then combines with location and heading information to yield actual keypoint coordinates. When fit to data, keypoint-MoSeq estimates for each frame the animal's location and pose, the noise in each keypoint[29] and the identity of the current behavioral syllable (Fig. 2a). Because of its structure, when a single keypoint implausibly jumps from one location to another, the model can attribute this 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 (Fig. 2b).

Unlike traditional MoSeq, keypoint-MoSeq homed in on behavioral syllables rather than noise in the keypoint data, yielding syllable transitions that overlapped more strongly with changepoints in pose, correlated better with syllable transitions from depth MoSeq and clustered less around low-confidence neural network detections (Fig. 2c). Keypoint-MoSeq also outperformed traditional MoSeq when the latter was applied to filtered keypoint data, or to keypoints inferred with a pose estimation method (Lightning Pose) that includes a jitter penalty in its training objective (Extended Data Fig. 2a,b). Furthermore, when we simulated missing data by ablating subsets of keypoints within random (0–3 s) intervals, keypoint-MoSeq was better able to preserve syllable labels and boundaries than traditional MoSeq (Extended Data Fig. 2c–f). 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 to keypoint-MoSeq resulted in models that failed to identify distinct syllables (Extended Data Fig. 2g,h).

Because keypoint-MoSeq produces slightly different syllable segmentations when run multiple times with different random seeds, we developed a likelihood-based metric that allows post hoc ranking of model runs (Extended Data Fig. 3a–g); the metric tends to be lowest for outlier models and highest for those that are consensus-like, providing a rational basis for model selection (Extended Data Fig. 3h–k). The metric revealed that 500 fitting iterations (~30 min of compute time on a GPU for ~5 h of data) are sufficient to achieve a good model fit with our open field dataset. Rather than choosing a single best model, users can also estimate an approximate probability distribution over syllable labels, although full Bayesian convergence remains impractical (Extended Data Fig. 3l).

In our open field data, keypoint-MoSeq identified 25 syllables that were easily distinguishable to human observers (Extended Data Fig. 4a and Supplementary Videos 2 and 3). These included categories of behavior (for example, rearing, grooming and walking), and variations within categories (for example, turn angle, speed; Fig. 2e). Importantly, keypoint-MoSeq preserved access to the kinematic and morphological parameters that underlie each behavioral syllable (Extended Data Fig. 4b). Thus, keypoint-MoSeq can provide an interpretable segmentation of behavior from standard two-dimensional (2D) keypoint tracking data.

Keypoint-MoSeq is sensitive to behavioral transitions

To characterize keypoint-MoSeq, we related the discovered syllables to orthogonal measures of behavior and neural activity and compared them to the behavioral states identified by alternative behavior analysis methods. These alternatives, which include VAME, MotionMapper and B-SOiD, all work by transforming keypoint data into a feature space that reflects the local dynamics around each frame, and then clustering frames according to those features[12,13,17,30].

When applied to our open field data, behavioral states from VAME, B-SOiD and MotionMapper were usually brief (median duration 33–100 ms, compared to ~400 ms for keypoint-MoSeq) and their transitions aligned poorly with changepoints in keypoint data, suggesting diminished sensitivity to the natural breakpoints in mouse behavior (Fig. 3a–c). This observation was not parameter dependent, because it remained true across a broad range of temporal windows (used by B-SOiD and MotionMapper) and after comprehensive scans over latent dimension, state number, clustering mode and preprocessing options (across all methods as applicable; Extended Data Fig. 5a).

Rearing offers a clear example of the differing sensitivity of each method to temporal structure. B-SOiD and keypoint-MoSeq both learned a specific set of rear states, and each encoded the mouse's height with comparable accuracy (Fig. 3d,e). Yet the rear states had different dynamics. Whereas keypoint-MoSeq typically detected two syllable transitions per rear (one entering the rear and one exiting), B-SOiD detected five to ten different transitions per rear, including switches between distinct rear states as well as flickering between rear and non-rear states (Fig. 3f and Extended Data Fig. 5b). Whereas mouse height increased at transitions into keypoint-MoSeq's rear state and fell at transitions out of it, height tended to peak symmetrically at transitions into and out of B-SOiD's rear states (Fig. 3g); this observation suggests that—at least in this example—B-SOiD does not effectively identify the boundaries between syllables, but instead fragments them throughout their execution.

We also evaluated each method using an orthogonal kinematic measurement: 3D head angle and acceleration from head-mounted inertial measurement units (IMUs; Fig. 3h). Behavioral transitions were identifiable in the IMU data as sudden changes in acceleration (quantified by jerk) and orientation (quantified by angular velocity). These measures tended to overlap with state transitions from keypoint-MoSeq but less so (or not at all) for B-SOiD, MotionMapper and VAME (Fig. 3i). 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 (Fig. 3i and Extended Data Fig. 6a).

The fact that keypoint-MoSeq more clearly identifies behavioral boundaries does not necessarily mean that it is better at capturing the overall content of behavior. Indeed, coarse kinematic parameters were captured equally well by all four of the tested methods (Extended Data Fig. 5c). 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, provides evidence that these methods afford fundamentally different views of temporal structure in behavior.

State transitions align with fluctuations in neural data

A core use case for unsupervised behavioral classification is to understand how the brain generates self-motivated behaviors outside a rigid task structure[9]; in this setting, boundaries between behavioral states serve as surrogate timestamps for alignment of neural data. For example, we recently used depth MoSeq to show that dopamine fluctuations in DLS are temporally aligned to syllable transitions during spontaneous behavior[22]. Here we asked whether the same result was apparent in keypoint-based segmentations of behavior (Fig. 4a).

Syllable-associated dopamine fluctuations (as captured by dLight photometry) were remarkably similar between depth MoSeq and keypoint-MoSeq, but much lower in amplitude (or nonexistent) when assessed using B-SOiD, VAME and MotionMapper (Fig. 4b and Extended Data Fig. 7a). We wondered if this apparent discrepancy in syllable-associated dopamine could be explained by differences

https://doi.org/10.1038/s41592-024-02318-2

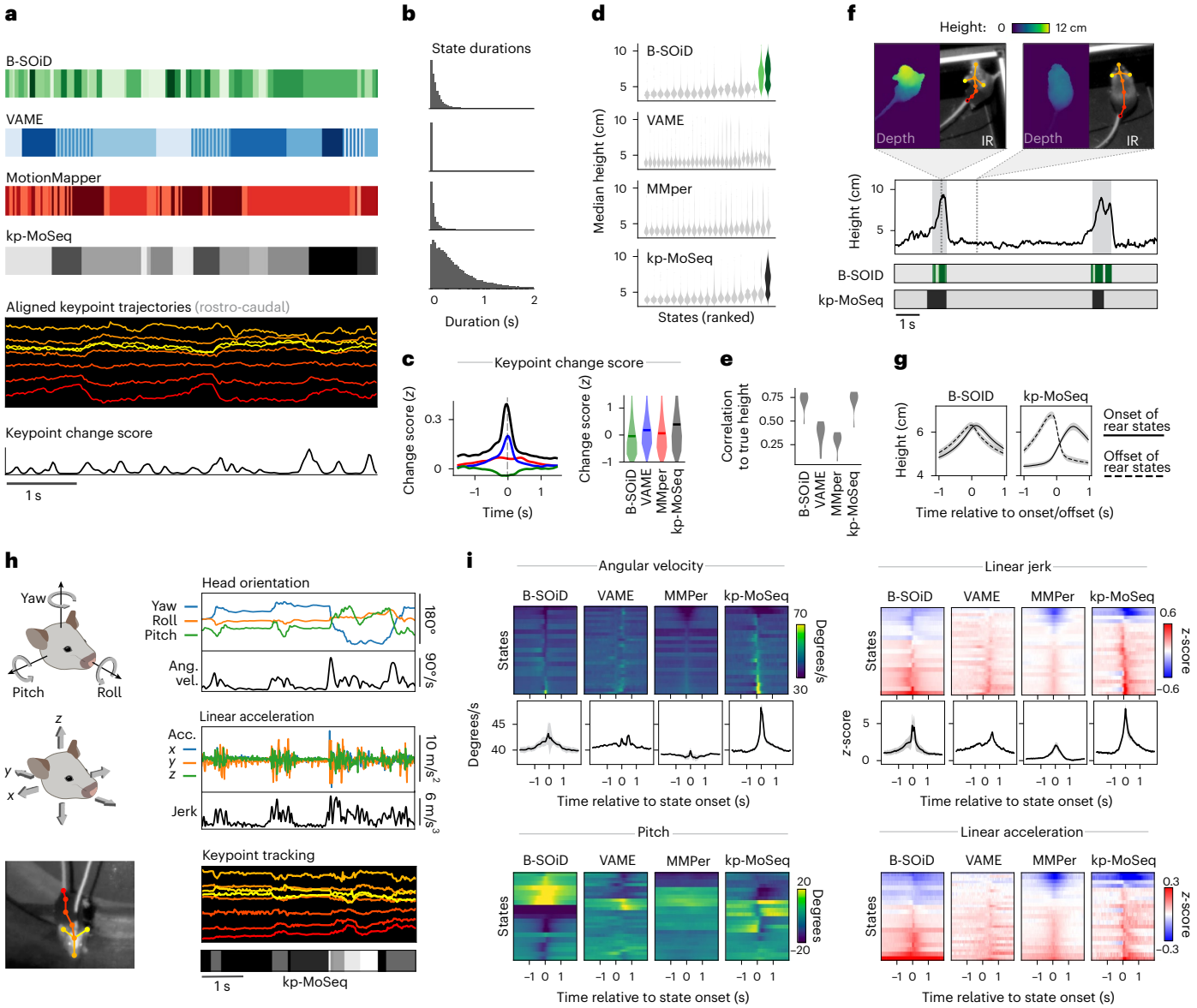


Fig. 3 | Keypoint-MoSeq captures the temporal structure of behavior. **a**, Output from four methods applied to the same 2D keypoint dataset. **b**, Distribution of state durations for each method in **a**. **c**, Left: average keypoint change scores (*z*-scored) around transitions identified by each method. Right: distribution of change scores at the transition point ('MMper' refers to MotionMapper). **d**, Distribution of mouse heights (measured by depth camera) for each unsupervised behavior state. States are classified as rear specific (and given a non-gray color in the plot) if they have median height > 6 cm. **e**, Accuracy of models trained to predict mouse height from behavior labels showing the distribution of accuracies across $N = 10$ recordings. **f**, Bottom: state sequences from keypoint-MoSeq and B-SOiD during a pair of example 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 during two example frames. **g**, Mouse height aligned to the onsets (solid lines) or offsets (dashed lines) of rear-specific states defined in **d**, showing mean and 95% confidence of the mean. **h**, Signals captured from a head-mounted 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), are plotted during a 5-s interval. Figure created with SciDraw under a CC BY 4.0 license. **i**, IMU signals aligned to the onsets of each behavioral state. Each heat map row represents a state. Line plots show the median across states for angular velocity and jerk (average and standard across $N = 10$ model fits). Keypoint-MoSeq peaks at a higher value for both signals ($P < 0.0005$, $N = 10$, Mann–Whitney U test).

in how each method represents 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 average dopamine levels should vary consistently across their behavior states but lack clear dynamics (increases or decreases) at state onsets. Indeed, for all four methods, almost every state was associated with a consistent above-average or below-average dopamine level (Fig. 4c,d and Extended Data Fig. 7b), and yet dopamine dynamics varied widely. Whereas dopamine usually increased at the initiation of keypoint-MoSeq syllables, it was usually flat (having just reached a peak or nadir) at state onsets identified by alternative methods (Fig. 4c–e). Furthermore, aligning the dopamine signal to randomly sampled times throughout the execution of each behavioral state—rather than its onset—altered state-associated dopamine dynamics for keypoint-MoSeq, but made little difference for alternative methods (Fig. 4f and Extended Data Fig. 7c,d). These results suggest that keypoint-MoSeq syllable onsets are meaningful landmarks for neural data analysis, while state onsets identified by alternative methods are often functionally indistinguishable from random timepoints during a behavior.

https://doi.org/10.1038/s41592-024-02318-2

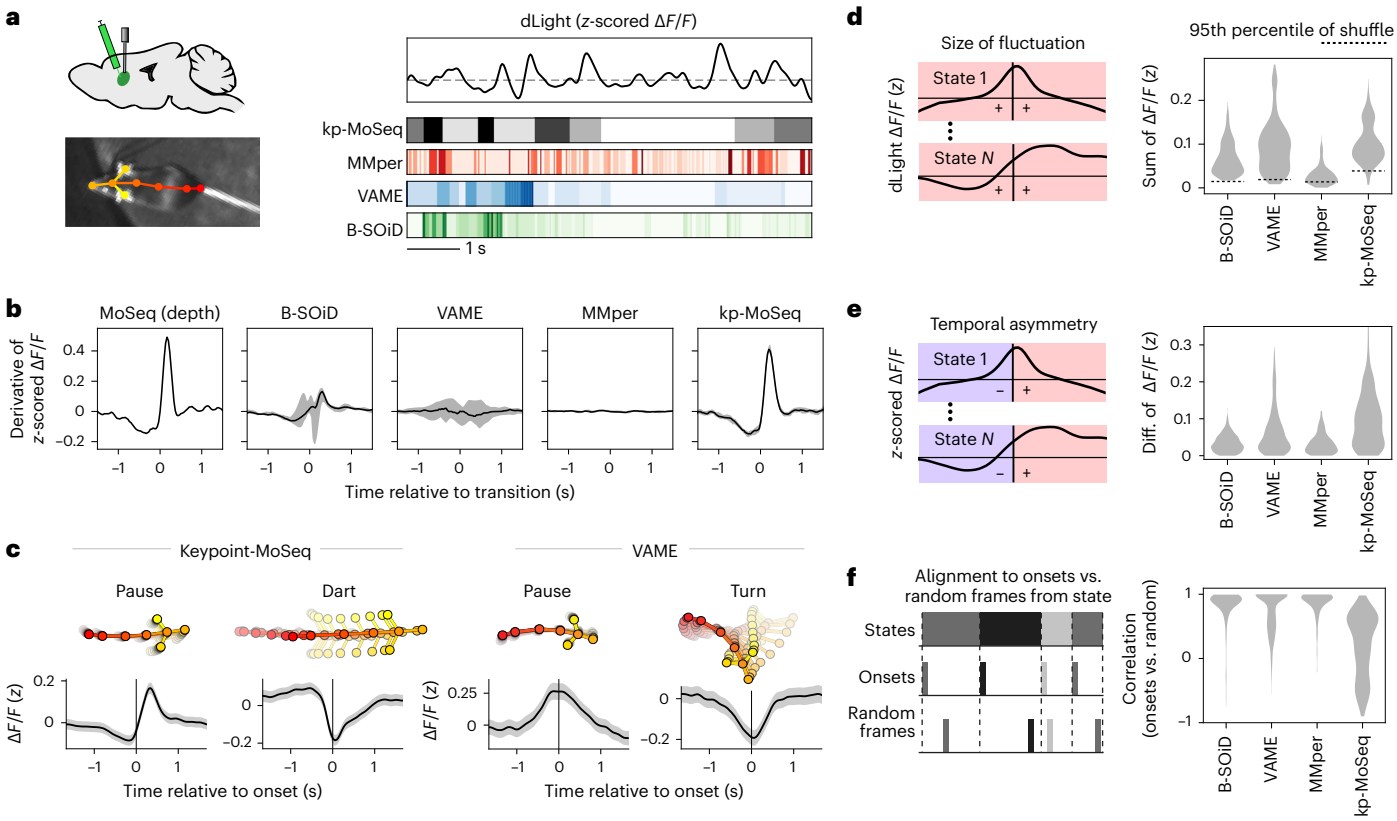


Fig. 4 | Keypoint-MoSeq syllable transitions align with fluctuations in striatal dopamine. a, Illustration depicting simultaneous recordings of dopamine fluctuations in the DLS obtained from fiber photometry (top) and unsupervised behavioral segmentation of 2D keypoint data (bottom). Adapted from ref. 22, Springer Nature Limited. **b**, Derivative of the dopamine signal aligned to state transitions from MoSeq (depth) and each keypoint-based method, showing the mean and 95% confidence of the mean. The derivative peaks at a higher value for keypoint-MoSeq compared to the non-MoSeq methods ($P < 10^{-5}$, $N = 20$ model fits per method, Mann–Whitney U test). **c**, Average dopamine signal (z-scored change in fluorescence, $\Delta F/F$) 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 (merging $N = 20$ model fits per method), where magnitude is defined as the mean total absolute value in a 1-s window centered on state onset. Box plots show median and interquartile range (IQR). **e**, Distributions capturing the temporal asymmetry of state-associated dopamine fluctuations, where asymmetry is defined as the difference in mean dopamine signal during 500 ms after versus 500 ms before state onset. Keypoint-MoSeq syllables have a higher asymmetry score on average than those from other methods ($P < 10^{-4}$, $N = 20$ model fits per method, Mann–Whitney U test). **f**, Temporal randomization affects keypoint-MoSeq-identified neurobehavioral correlations, but not those identified by other methods. Top: schematic of randomization. The dopamine signal was aligned either to the onsets of each state, as in **c**, or to random frames throughout the execution of each state. Bottom: distributions capturing the correlation of state-associated dopamine fluctuations before versus after randomization. Keypoint-MoSeq syllables have a lower correlation on average than those from other methods ($P < 10^{-4}$, $N = 20$ model fits per method, Mann–Whitney U test).

Keypoint-MoSeq generalizes across experimental setups and behaviors

Keypoint tracking is a powerful means of pose estimation because it generalizes widely across experimental setups. To test whether keypoint-MoSeq inherits this flexibility, we asked if it could quantify changes in behavior induced by environmental enrichment. Mice were recorded in either an empty arena or one that contained bedding, chew toys and a transparent shelter (Extended Data Fig. 8a). The enriched environment was too complex for traditional depth MoSeq but yielded easily to keypoint-based pose estimation. Based on these poses, keypoint-MoSeq identified 39 syllables, of which 21 varied between environments: syllables upregulated in the enriched environment tended to involve manipulation and orientation toward nearby affordances (for example, 'investigation', 'stationary right turn' and 'stop and dig'), whereas those upregulated in the empty box were limited to locomotion and rearing ('dart forward' and 'rear-up in corner'; Extended Data Fig. 8b,c). These results suggest that keypoint-MoSeq may be useful in a broad range of experimental contexts, including those whose cluttered structure precludes the effective use of depth cameras.

To test if keypoint-MoSeq can also generalize across laboratories—and to better understand the mapping between syllables and human-identified behaviors—we next analyzed a pair of published benchmark datasets[31,32]. The first dataset included human annotations for four mouse behaviors in an open field (locomotion, rearing, face grooming and body grooming) and keypoint detections from the TopViewMouse model in the DLC Model Zoo[33] (Fig. 5a–c). The second dataset (part of the CalMS21 benchmark[32]) included a set of three manually annotated social behaviors (mounting, investigation and attack) as well as keypoints for a pair of interacting mice (Fig. 5d–f). Keypoint-MoSeq recovered syllables from both datasets whose average duration was ~400 ms, while, as before, the B-SOiD, MotionMapper and VAME identified behavioral states that were much shorter (Extended Data Fig. 9a). Keypoint-MoSeq states also conformed more closely to human-identified behavioral states (Fig. 5c,f and Extended Data Fig. 9b). Although this advantage was modest overall, there were some important differences: in the CalMS21 dataset, for example, MotionMapper, B-SOiD and VAME only identified a single behavior consistently, with B-SOiD and VAME only capturing mounting and MotionMapper only capturing investigation in 100% of model fits;

https://doi.org/10.1038/s41592-024-02318-2

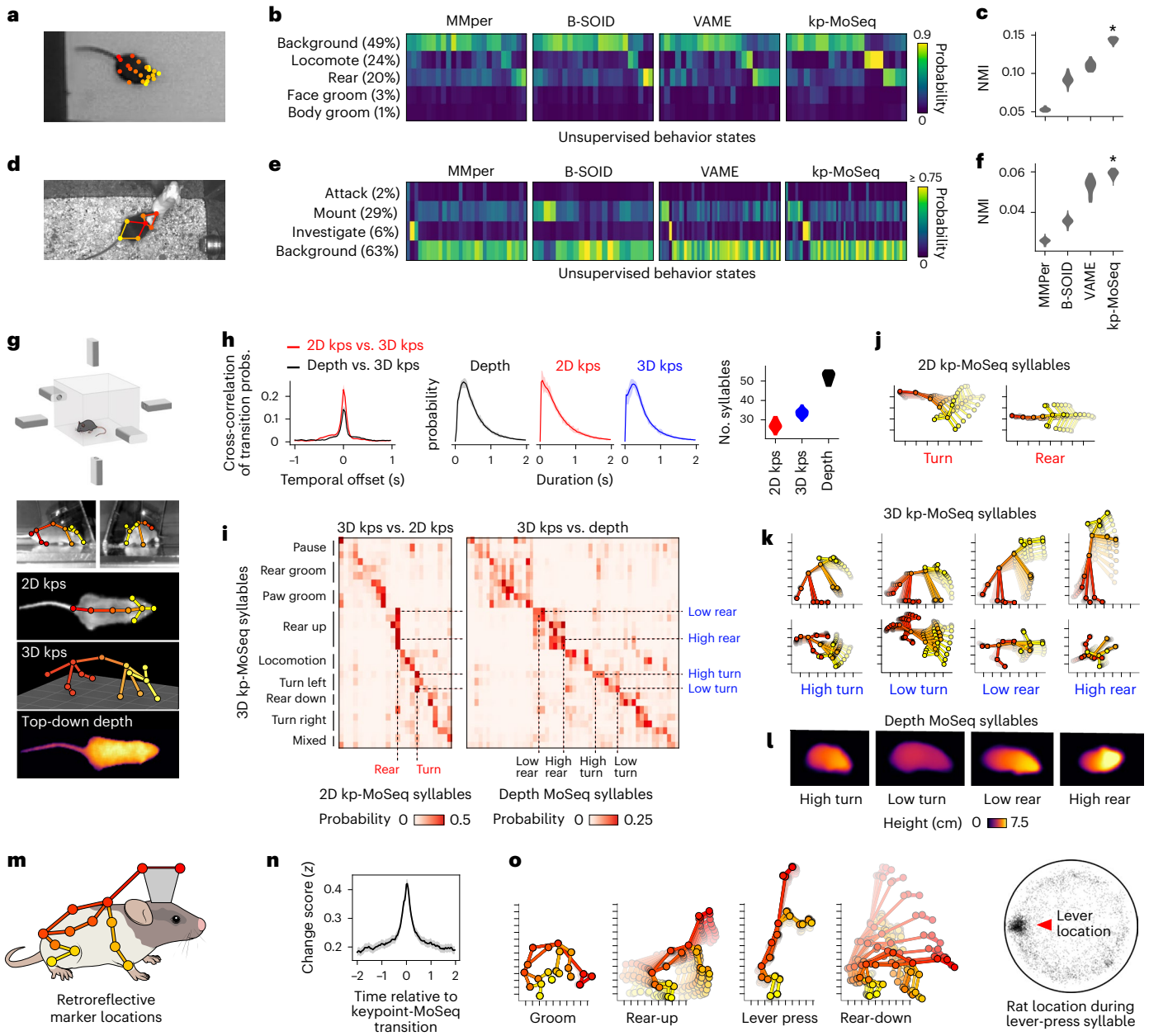


Fig. 5 | Keypoint-MoSeq generalizes across experimental setups. a, Frame from an open field benchmark dataset. **b**, Confusion matrices showing overlap between human-labeled behaviors and unsupervised states. **c**, Normalized mutual information (NMI) between supervised and unsupervised labels, showing the distribution of NMI values across $N = 20$ model fits. Keypoint-MoSeq consistently had higher NMI (*$P < 10^{-6}$, Mann–Whitney U test). **d**, Frame from the CalMS21 social behavior benchmark dataset, showing 2D keypoints of the resident mouse. **e**,**f**, Comparison between human labels and unsupervised behavior states of the resident mouse, as in **b** and **c** ($P < 10^{-5}$, Mann–Whitney U test). **g**, Multi-camera arena for simultaneous recording of 3D keypoints (3D kps), 2D keypoints (2D kps) and depth videos. Figure created with SciDraw under a CC BY 4.0 license. **h**, Comparison of MoSeq outputs from each modality. Left: cross-correlation between 3D transition probabilities and those for 2D keypoints and depth. Shading shows bootstrap 95% confidence intervals; middle: distribution of syllable durations, showing mean and inter-95% confidence interval range across $N = 20$ model fits. Right: number of states with frequency > 0.5%, showing the distribution of state counts across 20 runs of each model. **i**, Overlap of syllables from 2D keypoints (left) or depth (right) with each 3D keypoint-based syllable. **j**–**l**, Average pose trajectories for the syllables marked in **i**. **k**, 3D trajectories are plotted from the side (first row) and top (second row). **l**, Average pose (as depth image) 100 ms after syllable onset. **m**, Location of markers for rat motion capture. Figure created with SciDraw under a CC BY 4.0 license. **n**, Left: average keypoint change score (z) aligned to syllable transitions. Shading shows 95% confidence intervals of the mean. Right: durations of keypoint-MoSeq states and inter-changepoint intervals. **o**, Left: pose trajectories of example syllables learned from rat motion capture data. Right: random sample of rat centroid locations during execution of the 'lever-press' syllable.

keypoint-MoSeq, in contrast, defined at least one state specific to each of the three behaviors in 100% of model fits (Extended Data Fig. 9c).

The above benchmark datasets differed widely in the number of keypoints tracked (7 for CalMS21 versus 21 for the TopViewMouse model; Fig. 5a,d), raising the question of how the pose representation fed to keypoint-MoSeq influences its outputs. One possibility—suggested by the higher syllable count for depth MoSeq (~50) compared to keypoint-MoSeq fit to 2D keypoints (~25)—is that higher-dimensional input data allows MoSeq to make finer distinctions between behaviors. To test this rigorously, we used multiple cameras to estimate

https://doi.org/10.1038/s41592-024-02318-2

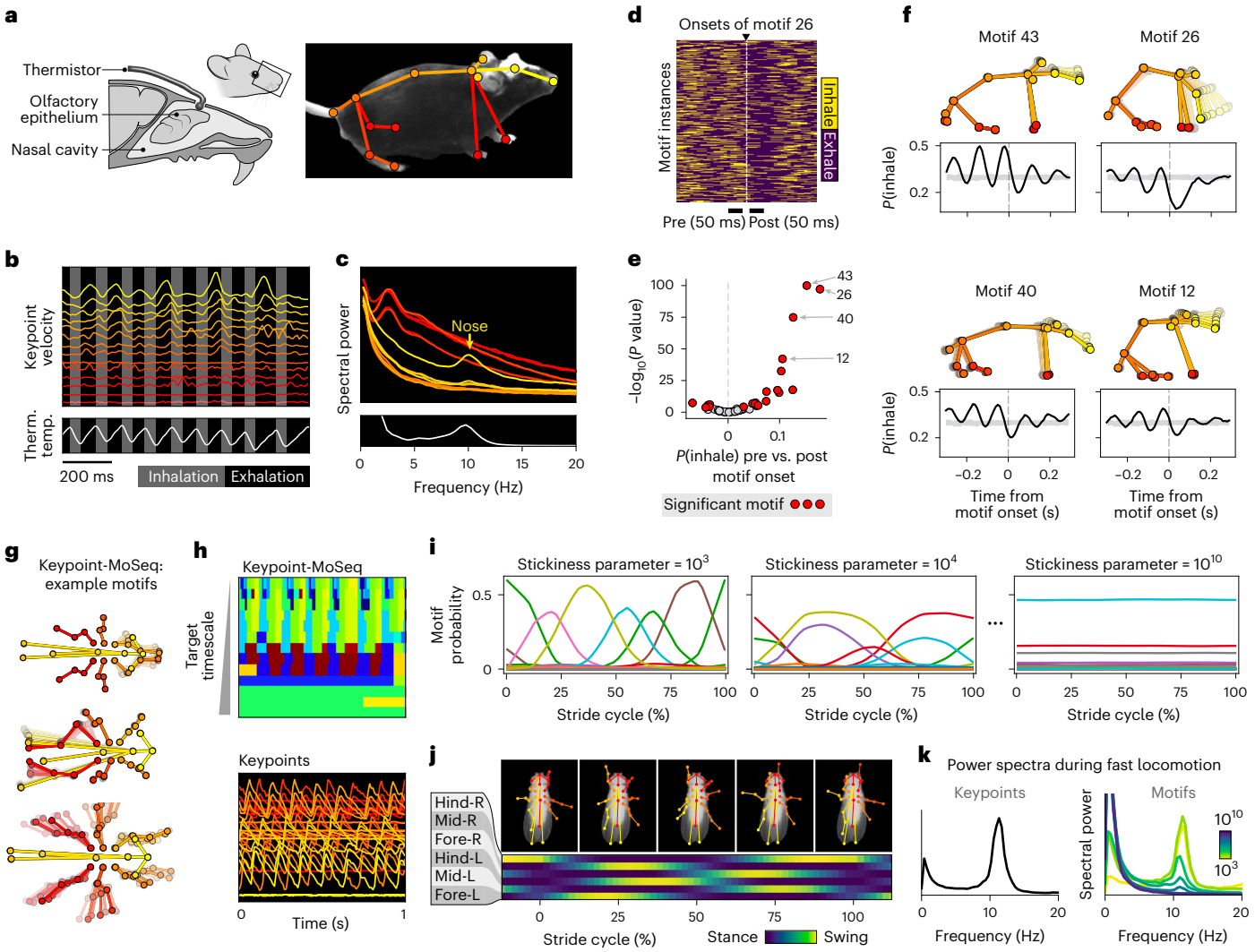


Fig. 6 | Keypoint-MoSeq segments behavior at multiple timescales. a, Setup for recording 3D pose and respiration, including location of thermistor, which monitors temperature fluctuations caused by respiration. Figure created with SciDraw under a CC BY 4.0 license. **b**, 3D keypoint velocities (top) and thermistor signal (bottom) over a 1-s interval. Keypoint traces are colored as in **a** and vertically spaced to ease visualization. **c**, Power spectra of 3D keypoint velocities (top) and thermistor signal (bottom). **d**, Example motif that aligns with inhale-to-exhale transition. The heat map shows respiration states across many instances of the motif. **e**, Volcano plot revealing respiration-aligned motifs. The *x* axis reflects change of inhalation probability during the 50 ms before versus after motif onset. **f**, Keypoint trajectories (top) and motif-aligned inhalation probabilities (bottom) for four motifs highlighted in **e**. Gray shading (bottom) shows the 2.5th-to-97.5th-percentile range of a shuffle distribution. **g**, Average pose trajectories for three fly motifs. **h**, Example of motif sequences during locomotion. Top: Keypoint-MoSeq output for models tuned to a range of timescales. Each row shows the output of a different model. Bottom: Aligned keypoint trajectories (anteroposterior coordinate). **i**, Frequency of motifs across the stride cycle during fast locomotion. Each line corresponds to one motif, and each panel represents a model with a different target timescale. **j**, Top: progression through the stride cycle. Bottom: probability that each leg is in stance or swing phase at each point in the stride; soft boundaries reflect variation in step timing. **k**, Power spectral density of keypoints (left) or motif labels (right) during fast locomotion. Colors in the right-hand plot correspond to models with a range of values for the stickiness hyperparameter, which sets the target timescale.

keypoints in 3D (including six keypoints that were not visible in the overhead-camera 2D dataset) and confirmed that the 3D keypoints had higher intrinsic dimensionality than 2D keypoints (Fig. 5g and Extended Data Fig. 9d,e). Despite this difference in dimensionality, similar changepoints were evident in both datasets, and keypoint-MoSeq identified syllables with similarly timed transitions (Fig. 5h and Extended Data Fig. 9f).

There was a bigger change, however, in how behaviors were categorized. Keypoint-MoSeq made finer-grained behavior distinctions based on 3D data as compared to 2D data, especially for behaviors that varied in height (Fig. 5i–l and Supplementary Video 4). Turning, for example, was grouped as a single state based on the 2D keypoint data but partitioned into three states with different head positions based on the 3D keypoint data (nose to the ground versus nose in the air; Fig. 5j–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. Depth-based MoSeq fractionated these behaviors still further. This analysis suggests that higher-dimensional input data permit richer descriptions of behavior, but even relatively low-dimensional 2D keypoint data still capture the timing of behavioral transitions.

Keypoint-MoSeq parses behavior across species and timescales

To test if keypoint-MoSeq generalizes across rodent species, we analyzed previously published 3D motion capture data derived from rats. In this dataset, rats were adorned with reflective body piercings

and recorded in a circular home cage arena with a lever and water spout for operant training (Fig. 5m; Rat7M dataset[34]). As with mice, keypoint-MoSeq syllables aligned with changepoints in the keypoint data (Fig. 5n) and included a diversity of behaviors, including a syllable specific to lever pressing in the arena (Fig. 5o and Supplementary Video 5).

Mice combine postural movements, respiration and whisking to sense their environment. Recent work suggests that rodents coordinate these behaviors in time, generating rhythmic head movements that synchronize with the sniff cycle[35,36]. Using an autoregressive hidden Markov model (AR-HMM), for example, head-movement motifs were discovered that align to respiration and arise during olfactory navigation[21]. Respiration, therefore, defines a fast timescale of mouse behavior that coexists with—but is distinct from—the ~400-ms timescale of behavioral syllables.

To test if keypoint-MoSeq can capture behavioral motifs at this faster timescale, we used 120-Hz cameras to track 3D keypoints of mice and measured respiration with an implanted thermistor[37] (Fig. 6a). Consistent with prior work, we observed respiration-synchronized fluctuations in nose velocity, although synchrony was weak or absent in other parts of the body (Fig. 6b,c). We then fit keypoint-MoSeq models with a range of target timescales (~35 ms to ~300 ms; Extended Data Fig. 10a). Motifs were defined as 'respiration coupled' if they consistently aligned with transitions in respiration state (inhale-to-exhale or exhale-to-inhale; Fig. 6d,e). Although respiration coupling was evident across all models, its prominence peaked at shorter timescales (Extended Data Fig. 10a), especially when fit to a subset of anterior keypoints that emphasized neck and nose movements (Extended Data Fig. 10b). The best-synchronized motifs (from the full-body model) tended to coincide with exhalation and involved isolated movements in which the nose flutters down (Fig. 6e,f). These results suggest that keypoint-MoSeq can characterize fast, sniff-aligned movements in the mouse.

Given that keypoint-MoSeq can parse two different timescales of mouse behavior, we wondered if it could also segment fly behavior, which similarly occurs at multiple well-defined timescales. Flies tend to switch between distinct, oscillatory pose trajectories[17]. These movements can be finely subdivided, as in the coordinated stance and swing phases of locomotion[38], or more coarsely segmented at the transitions between distinct oscillatory modes (for example, locomotion versus grooming), as they are by MotionMapper[17]. To capture these distinct levels of organization, we fit keypoint-MoSeq to 2D keypoints from flies exploring a flat substrate[17,39] (Extended Data Fig. 10c). The resulting behavioral motifs varied from tens to hundreds of milliseconds depending on keypoint-MoSeq's target timescale. At longer timescales, keypoint-MoSeq identified recognizable behaviors such as locomotion, head grooming or left-wing grooming, similarly to the behaviors reported by MotionMapper (Fig. 6g, Supplementary Video 6 and Extended Data Fig 10d–f).

At shorter time scales, keypoint-MoSeq divided these behaviors into their constituent phases. Fast locomotion, for example, was split between six phase-locked motifs that tiled the stride cycle (Fig. 6h). As target timescales grew longer, locomotion merged from six to two phases (corresponding to the alternating swings and stances of a canonical tripod gait) before eventually collapsing to a single motif that encompassed the full stride cycle (Fig. 6h–j and Extended Data Fig. 10g). This shift was evident in the power spectral density of keypoint-MoSeq's output, which began with a prominent peak at ~12 Hz during fast locomotion (corresponding to the stride cycle) that slowly disappeared as keypoint-MoSeq's target timescale was increased (Fig. 6k). The same hierarchy of timescales appeared for non-locomotion behaviors as well (Extended Data Fig. 10h). These results demonstrate that keypoint-MoSeq is useful as a tool for fly behavior analysis and suggest a principle for setting its target timescale that depends on whether researchers wish to subdivide the distinct phases of oscillatory behaviors.

Discussion

Syllables are broadly useful for understanding behavior[16,22–24], but their scope has been limited by the past requirement for depth data. Here we show that keypoint-MoSeq affords similar insight as depth-based MoSeq while benefiting from the generality of markerless keypoint tracking. Whereas depth MoSeq was limited to a narrow range of spatial scales and frame rates, keypoint-MoSeq can be applied to mammals and insects, parsing behaviors at the second or millisecond timescale. And because keypoint tracking is more robust to occlusion and environmental clutter, it is now possible to parse syllables amid environmental enrichment, in animals behaving alone or socially, with or without headgear and neural implants.

The core innovation enabling keypoint-MoSeq is a probabilistic model that effectively handles occlusions, tracking errors and high-frequency jitter. These noise sources are pervasive in pose tracking[5,26]; because standard methods like SLEAP and DLC process each frame separately, keypoint coordinates tend to jump from frame to frame even when the subject's pose has not discernably changed. A newer generation of pose tracking methods, such as GIMBAL[29], Deep Graph Pose[26] and Lightning Pose[40], correct for some of these errors; and two-step pipelines that build on these methods may be less prone to keypoint jitter. Here, we describe a different solution: combining noise-correction and behavior segmentation in a single end-to-end model that leverages learned patterns of animal motion to infer the most plausible pose trajectory from noisy or missing data.

Keypoint-MoSeq is somewhat resilient to noise, but it will perform best with clean keypoint data that capture most parts of the body. Although directly modeling the raw pixel intensities of depth[16] or 2D video[41] provides the most detailed access to spontaneous behavior, technical challenges like reflections, occlusions and variation in perspective and illumination remain a challenge in those settings. The development of keypoint-MoSeq—together with advances in markerless pose tracking—should enable MoSeq to be used in a variety of these adversarial circumstances, such as when animals are obstructed from a single axis of view, when multiple animals are interacting simultaneously, when the environment changes dynamically and when animals wear elaborate headgear.

Compared to keypoint-MoSeq, the alternative methods for unsupervised behavior segmentation that we tested (B-SOiD[12], MotionMapper[17] and VAME[13]) tend to emit shorter behavior motifs that often start or stop in the middle of what humans might identify as a behavioral module or motif (for example, a rear). Our analysis suggests two possible reasons for this difference. First, unlike alternative methods, MoSeq can discretize behavior at a particular user-defined timescale and, therefore, is better able to identify clear boundaries between behavioral elements that respect the natural rhythmicity in movements associated with syllables, sniffs or steps. The resulting parsimony prevents over-fractionation of individual behaviors. Second, the hierarchical structure of keypoint-MoSeq's underlying generative model means it can detect noise in keypoint trajectories and distinguish this noise from actual behavior without smoothing away meaningful behavioral transitions.

That said, we stress that there is no one best approach for behavioral analysis, as all methods involve trade-offs[42,43]. For example, keypoint-MoSeq does not yield a single fixed description of behavior, since its output is probabilistic. In principle, one could summarize this uncertainty in the form of a posterior distribution. Because proper posterior estimation is impractical using our current fitting procedure, we have defined an alternative approach whereby users generate an ensemble of candidate model fits and identify a consensus model for downstream analysis. Users wishing to better quantify model uncertainty can also apply subsequent analyses to the full ensemble of models. Keypoint-MoSeq is also limited to describing behavior at a single timescale. Although users may vary this timescale across a broad range, keypoint-MoSeq cannot simultaneously analyze behavior across

multiple timescales or explicitly represent the hierarchical nesting of behavior motifs. Finally, because keypoint-MoSeq learns the identity of syllables from the data itself, it may miss especially rare behavioral events that could otherwise be captured using supervised methods.

To facilitate the adoption of keypoint-MoSeq, we built a website (http://www.moseq4all.org/) that includes free access to the code for academics as well as extensive documentation and guidance for implementation. As demonstrated here, the model underlying MoSeq is modular and thus accessible to extensions and modifications that can increase its alignment to behavioral data. For example, a time-warped version of MoSeq was recently reported that incorporates a term to explicitly model variation in movement vigor[19]. 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.

Online content

Any methods, additional references, Nature Portfolio reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41592-024-02318-2.

References

1. Tinbergen, N. *The Study of Instinct* (Clarendon Press, 1951).
2. Dawkins, R. In *Growing Points in Ethology* (Bateson, P. P. G. & Hinde, R. A. eds.) Chap 1 (Cambridge University Press, 1976).
3. Baerends, G. P. The functional organization of behaviour. *Anim. Behav.* **24**, 726–738 (1976).
4. Pereira, T. D. et al. SLEAP: a deep learning system for multi-animal pose tracking. *Nat. Methods* **19**, 486–495 (2022).
5. Mathis, A. et al. DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nat. Neurosci.* **21**, 1281–1289 (2018).
6. Sun, J. J. et al. Self-supervised keypoint discovery in behavioral videos. *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* **2022**, 2161–2170 (2022).
7. Graving, J. M. et al. DeepPoseKit, a software toolkit for fast and robust animal pose estimation using deep learning. *eLife* **8**, e47994 (2019).
8. 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).
9. Datta, S. R., Anderson, D. J., Branson, K., Perona, P. & Leifer, A. Computational neuroethology: a call to action. *Neuron* **104**, 11–24 (2019).
10. Anderson, D. J. & Perona, P. Toward a science of computational ethology. *Neuron* **84**, 18–31 (2014).
11. Pereira, T. D., Shaevitz, J. W. & Murthy, M. Quantifying behavior to understand the brain. *Nat. Neurosci.* **23**, 1537–1549 (2020).
12. Hsu, A. I. & Yttri, E. A. B-SOiD, an open-source unsupervised algorithm for identification and fast prediction of behaviors. *Nat. Commun.* **12**, 5188 (2021).
13. Luxem, K. et al. Identifying behavioral structure from deep variational embeddings of animal motion. *Commun. Biol.* **5**, 1267 (2022).
14. Marques, J. C., Lackner, S., Félix, R. & Orger, M. B. Structure of the Zebrafish locomotor repertoire revealed with unsupervised behavioral clustering. *Curr. Biol.* **28**, 181–195 (2018).
15. Todd, J. G., Kain, J. S. & de Bivort, B. L. Systematic exploration of unsupervised methods for mapping behavior. *Phys. Biol.* **14**, 015002 (2017).
16. Wiltschko, A. B. et al. Mapping sub-second structure in mouse behavior. *Neuron* **88**, 1121–1135 (2015).
17. Berman, G. J., Choi, D. M., Bialek, W. & Shaevitz, J. W. Mapping the stereotyped behaviour of freely moving fruit flies. *J. R. Soc. Interface* https://doi.org/10.1098/rsif.2014.0672 (2014).
18. Batty, E. et al. BehaveNet: nonlinear embedding and Bayesian neural decoding of behavioral videos. in *Advances in Neural Information Processing Systems 32* (eds H. Larochelle et al.) 15706–15717 (Curran Associates, 2019).
19. Costacurta, J. C. et al. Distinguishing discrete and continuous behavioral variability using warped autoregressive HMMs. in *Advances in Neural Information Processing Systems 35* (eds S. Koyejo et al.) 23838–23850 (Curran Associates, 2022).
20. Jia, Y. et al. Selfee, self-supervised features extraction of animal behaviors. *eLife* **11**, e76218 (2022).
21. Findley, T. M. et al. Sniff-synchronized, gradient-guided olfactory search by freely moving mice. *eLife* **10**, e58523 (2021).
22. Markowitz, J. E. et al. Spontaneous behaviour is structured by reinforcement without explicit reward. *Nature* **614**, 108–117 (2023).
23. Markowitz, J. E. et al. The striatum organizes 3D behavior via moment-to-moment action selection. *Cell* **174**, 44–58 (2018).
24. Wiltschko, A. B. et al. Revealing the structure of pharmacobehavioral space through motion sequencing. *Nat. Neurosci.* https://doi.org/10.1038/s41593-020-00706-3 (2020).
25. Lin, S. et al. Characterizing the structure of mouse behavior using motion sequencing. Preprint at https://arxiv.org/abs/2211.08497 (2022).
26. Wu, A. et al. Deep Graph Pose: a semi-supervised deep graphical model for improved animal pose tracking. in *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Curran Associates, 2020).
27. Murphy, K. P. *Machine Learning* (MIT Press, 2012).
28. Linderman, S. et al. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics* Vol. 54 (eds Aarti, S. et al.) 914–922 (PMLR, Proceedings of Machine Learning Research, 2017).
29. 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).
30. Klibaite, U. et al. Deep phenotyping reveals movement phenotypes in mouse neurodevelopmental models. *Mol. Autism* **13**, 12 (2022).
31. Bohnslav, J. P. et al. DeepEthogram, a machine learning pipeline for supervised behavior classification from raw pixels. *eLife* **10**, e63377 (2021).
32. Sun, J. J. et al. Caltech mouse social interactions (CalMS21) dataset. https://doi.org/10.22002/D1.1991 (2021).
33. Ye, S., Mathis, A. & Mathis, M. W. Panoptic animal pose estimators are zero-shot performers. Preprint at https://arxiv.org/abs/2203.07436 (2022).
34. Marshall, J. D. et al. Continuous whole-body 3D kinematic recordings across the rodent behavioral repertoire. *Neuron* **109**, 420–437 (2021).
35. Moore, J. D. et al. Hierarchy of orofacial rhythms revealed through whisking and breathing. *Nature* **497**, 205–210 (2013).
36. Kurnikova, A., Moore, J. D., Liao, S. -M., Deschênes, M. & Kleinfeld, D. Coordination of orofacial motor actions into exploratory behavior by rat. *Curr. Biol.* **27**, 688–696 (2017).
37. McAfee, S. S. et al. Minimally invasive highly precise monitoring of respiratory rhythm in the mouse using an epithelial temperature probe. *J. Neurosci. Methods* **263**, 89–94 (2016).

38. DeAngelis, B. D., Zavatone-Veth, J. A. & Clark, D. A. The manifold structure of limb coordination in walking *Drosophila*. *Elife* https://doi.org/10.7554/eLife.46409 (2019).
39. Pereira, T. D. et al. Fast animal pose estimation using deep neural networks. *Nat. Methods* **16**, 117–125 (2019).
40. Dan, B. et al. Lightning Pose: improved animal pose estimation via semi-supervised learning, Bayesian ensembling, and cloud-native open-source tools. Preprint at *bioRxiv* https://doi.org/10.1101/2023.04.28.538703 (2023).
41. Batty, E. et al. In *NeurIPS* vol. 32 (eds H. Wallach et al.) (Curran Associates, 2019).
42. Berman, G. J., Bialek, W. & Shaevitz, J. W. Predictability and hierarchy in *Drosophila* behavior. *Proc. Natl Acad. Sci. USA* **113**, 11943–11948 (2016).
43. Berman, G. J. Measuring behavior across scales. *BMC Biol.* **16**, 23 (2018).

Methods

Ethical compliance

All experimental procedures were approved by the Harvard Medical School Institutional Animal Care and Use Committee (protocol number 04930) and were performed in compliance with the ethical regulations of Harvard University as well as the Guide for Animal Care and Use of Laboratory Animals.

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-h light/12-h 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 min, and then placed in the behavioral arena for 30–60 min. We recorded 6 male mice for 10 sessions (6 h) in the initial round of open field recordings; 5 male mice for 52 sessions (50 h) during the accelerometry recordings; 16 male mice for 16 sessions (8 h) during the environmental enrichment experiment; and 5 male mice for 9 sessions (6 h) during the thermistor recordings. The dopamine photometry recordings were obtained from a recent study[22]. 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. Of these, we selected a random subset of 95 sessions (~50 h) for benchmarking keypoint-MoSeq.

Stereotactic surgery procedures

For all stereotactic surgeries, mice were anesthetized using 1–2% isoflurane in oxygen, at a flow rate of 1 l min^{-1} for the duration of the procedure. Anteroposterior (AP) and mediolateral (ML) coordinates (in millimeters) were zeroed relative to bregma, and the dorsoventral (DV) coordinate was zeroed relative to the pial surface. All mice were monitored daily for 4 days following surgery and were allowed to recover for at least 1 week. Mice were then habituated to handling and brief head-fixation before beginning recordings.

For dopamine recordings, 400 nl 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. 22 for additional details).

For accelerometry recordings, we surgically attached a Mill-Max connector (DigiKey, ED8450-ND) and head bar to the skull and secured it with dental cement (Metabond). A nine-degree-of-freedom absolute orientation IMU (Bosch, BN0055) was mounted on the Mill-Max connector using a custom printed circuit board (PCB) with a net weight below 1 g.

For thermistor surgeries, we adapted a protocol previously described[37]. We first prepared the implant (GAG22K7MCD419, TE Connectivity) by stripping the leads and soldering them to two male Mill-Max pins (0.05-inch pitch, 851-93-050-10-001000). The pins and their solder joins were then entirely covered in Prime-Dent light-curable cement, and cured for 10–20 s, to ensure the longevity and stability of the electrical connection. Each implant was tested by touching two leads of a multimeter (set to measure resistance) to the female side of the Mill-Max, breathing gently on the thermistor, and checking for a resistance drop of roughly 20 kΩ to 18 kΩ.

To implant the thermistor, a midline incision was made from ~1 mm behind lambda to ~1 mm anterior to the nasal suture, and the skull cleaned and lightly scored. A craniotomy was made just anterior to the nasal suture (well posterior to the position originally reported[37]), large enough for the thermistor to fit fully inside. The thermistor was fully inserted along the AP axis so that it lay flat in the horizontal plane inside the nasal cavity. The craniotomy was then sealed with KwikSil, and the thermistor wire was secured to the skull 1–2 mm posterior to the craniotomy with cyanoacrylate glue (Loctite 454). Then dental cement (Metabond) was used to attach the Mill-Max connector in an upright position between bregma and lambda, and a head bar was cemented to the skull at lambda.

Microsoft Azure recording setup

For the initial set of open field recordings (Figs. 1, 2, 3a–g and 5g–l), mice were recorded in a square arena with transparent floor and walls (30 cm length and width). Microsoft Azure Kinect cameras captured simultaneous depth and near-IR video at 30 Hz. Six cameras were used in total: one above, one below and four side cameras at right angles at the same height as the mouse.

Accelerometry recordings

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 (45-cm diameter). Data were transferred from the IMU using a lightweight tether attached to a custom-built active commutator. 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-IR 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.

Thermistor recordings

To record mouse respiration and movement at high frame rates, we built a multi-camera recording arena using six Basler ace acA1300-200um Monochrome USB 3.0 Cameras (Edmund Optics, 33-978) that recorded from above, from below and four side views. The cameras were triggered at 120 Hz using an Arduino. Video compression was performed in real time on a GPU using a custom library (https://github.com/calebweinreb/multicamera_acquisition/). Mice were recorded in an open-top glass cube and illuminated with 32 near-IR high-power LED stars (LEDSupply, CREEXPE-FRD-3). To avoid reflections and saturations effects, the bottom camera was triggered slightly out of phase with the top cameras, and the LEDs were split into two groups: one group below the arena that turned on during the bottom camera's exposure, and one group above the arena that turned on during the top and side cameras' exposure.

To record the thermistor signal, we designed a custom PCB that used an op-amp (INA330AIDGST, Texas Instruments) to transform the thermistor's resistance fluctuations into voltages, and another circuit element to keep the voltage within the 0–3.3 V range. The PCB was connected to an Arduino (separate from the one controlling the cameras) that recorded the output. The PCB parts list, schematic and microcontroller code are available upon reasonable request to the laboratory of S.R.D.

Before behavioral recording sessions with the thermistor, mice were briefly head-fixed, and a cable with a custom headstage was inserted into the head-mounted Mill-Max adaptor. The cable was commutated with an assisted electric commutator from Doric Lenses and connected to the input of the op-amp on the custom PCB. To synchronize the thermistor and video data, we piped a copy of the camera trigger signal from the camera-Arduino to the thermistor-Arduino and recorded this signal alongside the thermistor output.

Environmental enrichment recordings

To test the effects of environmental enrichment on behavior, we built an arena for overhead video recording of an open-topped home cage. The home cage was surrounded on each side by a 16-inch vertical barrier, illuminated from above by three near-IR LED starts (LEDSupply, CREEXPE-FRD-3) and recorded with a Basler ace acA1300-200um

Monochrome USB 3.0 Camera (Edmund Optics 33-978). For half the recordings, the cage was filled with bedding, nesting material, chew sticks and a transparent, dome-shaped hut. For the other half, the cage was completely empty (except for the mouse).

Software

The following publicly available software packages were used for analysis: Python (version 3.8), NumPy (version 1.24.3), Scikit-learn (version 1.2.2), PyTorch (version 1.9), Jax (version 0.3.22), SciPy (version 1.10.1), Matplotlib (version 3.7.1), Statsmodels (version 0.13.5), Motionmapperpy (version 1.0), DeepLabCut (version 2.2.1), SLEAP (version 1.2.3), B-SOiD (version 1.5.1), VAME (version 1.1), GIMBAL (version 0.0.1), HRNet (unversioned), LightningPose (version 0.0.4) and segmentation_models_pytorch (version 0.3.3).

Statistics

All reported *P* values for comparisons between distributions were derived from Mann–Whitney *U* tests unless stated otherwise. In all comparisons to 'shuffle', the shuffle represents a cyclic permutation of the data.

Processing depth videos

Applying MoSeq to depth videos involves: (1) mouse tracking and background subtraction; (2) egocentric alignment and cropping; (3) PCA; and (4) probabilistic modeling. We applied steps 2–4 as described in the MoSeq2 pipeline[25]. For step 1, we trained a convolutional neural network (CNN) with a Unet++[44] architecture to segment the mouse from background using ~5,000 hand-labeled frames as training data.

Keypoint tracking for Microsoft Azure IR recordings

We used CNNs with an HRNet[45] architecture (https://github.com/stefanopini/simple-HRNet/) with a final stride of two for pose tracking. The networks were trained on ~1,000 hand-labeled frames each for the overhead, below-floor and side-view Microsoft Azure cameras. Frame labeling was crowdsourced through a commercial service (Scale AI). The crowdsourced labels were comparable to those from experts in our laboratory (Extended Data Fig. 1d). For the overhead camera, we tracked two ears and six 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 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 a 10-pixel (px) 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. The resulting mean absolute error (MAE) between network detections and manual annotations was 2.9 px for the training data and 3.2 px for held-out data. We also trained DeepLabCut and SLEAP models on the overhead-camera and below-floor-camera datasets. For DeepLabCut, we used version 2.2.1, setting the architecture to resnet50 architecture and the 'pos_dist_thresh' parameter to 10, resulting in train and test MAEs of 3.4 px and 3.8 px, respectively. For SLEAP, we used version 1.2.3 with the baseline_large_rf.single.json configuration, resulting in train and test MAEs of 3.5 px and 4.7 px. For Lightning Pose[40], we used version 0.0.4 and default parameters with 'pca_singleview' and 'temporal' loss terms.

Keypoint tracking for thermistor recordings

We trained separate keypoint detection networks for the Basler camera arena (used for the thermistor recordings). CNNs with an HRNet architecture were trained on ~1,000 hand-labeled frames each for the overhead and below-floor cameras and ~3,000 hand-labeled frames for the side-view cameras. The same keypoints were used as the ones for the Microsoft Azure dataset.

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[29]. To fit GIMBAL, we computed initial 3D keypoint estimates using robust triangulation (that is, by taking the median across all camera pairs, as in 3D-DeepLabCut[46]) 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. 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. Camera calibration and initial triangulation were performed using a custom library (https://github.com/calebweinreb/multicam-calibration/tree/main/multicam_calibration/).

Keypoint change score

We defined the keypoint 'change score' as the total velocity of keypoints after egocentric alignment. The goal of the change score is to highlight sudden shifts in pose. It was calculated by: (1) transforming keypoints into egocentric coordinates; (2) smoothing the transformed coordinates with Gaussian kernel (sigma = 1 frame); (3) calculating total change in coordinates across each frame; and (4) *z*-scoring. Formally, the score can be defined as:

$$\text{Change score}\,(t) = z\,\text{score}(|\,y_t - y_{t-1}|)$$

where y_t are the keypoint coordinates after Gaussian smoothing.

Spectral analysis of keypoint jitter

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 $-\log_{10}$ (neural network detection confidence), as well as the 'error magnitude' (Fig. 1g). Error magnitude was defined as the distance between the detected 2D location of a keypoint (based on a single camera angle) and a re-projection 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 probabilities (Fig. 1g). The model transition probabilities were defined for each frame as the fraction of $N = 20$ model fits in which a transition occurred on that frame. Formally, if $z^{(i)}$ denotes the syllable sequence learned by model fit i, then the transition probability at time t is calculated as

$$\frac{1}{N}\sum_{i=1}^{N}\delta\left(z_t^{(i)} \neq z_{t-t}^{(i)}\right)$$

Applying keypoint-MoSeq

Datasets were modeled separately and multiple models with different random seeds were fit for each dataset (see Supplementary Table 1 for number of fits per dataset).

Modeling consisted of two phases: (1) fitting an AR-HMM to a fixed pose trajectory derived from PCA of egocentric-aligned keypoints; and (2) fitting a full keypoint-MoSeq model initialized from the AR-HMM. References in the text to 'MoSeq applied to keypoints' or 'MoSeq (keypoints)', for example, in Figs. 1 and 2, 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 notation' 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 via https://keypoint-moseq.readthedocs.io/en/latest/. With a consumer GPU, keypoint-MoSeq requires 30–60 min of computation time to model 5 h of data. The computation time scales linearly with dataset size.

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 four PCs for the overhead-camera 2D datasets, six PCs for the below-floor camera 2D datasets and six 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 and Q, and the transition parameters π and β. The hyperparameters for this step, listed in 'mathematical notation' below, were generally identical to those in the original depth MoSeq model. The one exception was the stickiness hyperparameter κ, which we adjusted separately for each dataset to ensure a median state duration of 400 ms.

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 hyperparameters for this step are enumerated in 'mathematical notation'. In general, we used the same hyperparameter values across datasets. The two exceptions were the stickiness hyperparameter κ, which again had to be adjusted to maintain a median state duration of 400 ms, and s_0, which determines a prior on the noise scale. Because 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:

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

The κ value used for each dataset is reported in Supplementary Table 2.

Trajectory plots

To visualize the modal trajectory associated with each syllable (Fig. 2e), 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 and (3) computed a final trajectory using the nearest neighbors of the representative trajectory.

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 $(y_{T-5}, \ldots, y_{T+15})$ using the centroid and heading on time T. In other words

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

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.

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 instances of the same syllable and then computed a final trajectory as the mean across these nearest neighbors. The trajectory plots in Fig. 2e consist of ten evenly-space poses along this trajectory, that is, the poses at times $T-5, T-3, \ldots, T+13$.

Testing robustness to missing data

To test the ability of keypoint-MoSeq to infer syllables and sequences in the face of missing data, we artificially ablated random subsets of keypoints at randomly timed intervals and then modeled the ablated data (Extended Data Fig. 2c–f). The ablation intervals began on every 10th second of the recording and lasted between 33 ms and 3 s (uniformly at random). For each interval, anywhere between 1 and 8 keypoints were selected (uniformly at random). Ablation entailed (1) erasing the keypoint coordinates and then filling the gap by linear interpolation; (2) setting the corresponding confidence values to 0. We then applied keypoint-MoSeq 20 times with different random seeds, using a single, fixed set of parameters derived previously from standard model fitting on the unablated dataset. Fixing the parameters ensured that syllable labels would be comparable across repeated model fits.

Cross-syllable likelihoods

We defined each cross-syllable likelihood 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 and 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 $t_s \le t \le t_e$ and $z_{t_s-1} \neq n \neq z_{t_e+1}$. For each such instance, one can calculate the probability $P\left(x_{t_s}, \ldots, x_{t_e} \mid A_m, b_m, Q_m\right)$ 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

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

Generating synthetic keypoint data

To generate the synthetic keypoint trajectories used for Extended Data Fig. 2h, 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 notation'), except that it only had one discrete state, lacked centroid and heading variables and allowed separate noise terms for the x and y coordinates of each keypoint.

Expected marginal likelihood score

Because keypoint-MoSeq can at best produce point estimates of the model parameters—which will differ from run to run—users typically run the model several times and then rank the resulting fits. For ranking model fits, we defined a custom metric called the expected marginal likelihood score. The score evaluates a given set of autoregressive parameters (A, b, Q) by the expected value of the marginal log likelihood: $E_{x\sim P(x|\,y)} \log P(x|A,b,Q)$. In practice, given an ensemble of pose trajectories $x^{(i)}$ and parameters $\theta^{(i)} = (A, b, Q)$ derived from N separate MCMC chains, the scores are computed as:

$$\text{Score}(\theta^{(i)}) = \frac{1}{1-N}\sum_{j\neq i} \log P\left(x^{(j)}|\theta^{(i)}\right)$$

The scores shown in Extended Data Fig. 3j–m were computed using an ensemble of $N = 20$ chains. We chose this custom score instead of a more standard metric (such as held-out likelihood) because computing the latter is intractable for the keypoint-MoSeq model.

Environmental enrichment analysis

We fit a single keypoint-MoSeq model to the environmental enrichment dataset, which included recordings in an enriched home cage and control recordings in an empty cage. The transition graph (Extended Data Fig. 8b) was generated with keypoint-MoSeq's analysis pipeline (https://keypoint-moseq.readthedocs.io/en/latest/analysis.html#syllable-transition-graph/) using node positions from a force directed layout. Detection of differentially used syllables was also performed using the analysis pipeline, which applies a Kruskal–Wallis test for significant differences in the per-session frequency of each syllable (https://keypoint-moseq.readthedocs.io/en/latest/analysis.html#compare-between-groups/). Syllables were clustered into three groups by applying community detection (networkx.community.louvain_communities) to a complete graph where nodes are syllables and edges were weighted by the bigram probabilities $b_{ij} = P(z_t = i, z_{t+1} = j)$).

Applying published methods for behavior analysis

We applied B-SOiD, VAME and MotionMapper using default parameters, except for the parameter scans in Extended Data Fig. 5 (see Supplementary Table 3 for a summary for all parameter choices). In general, we were unable to uniformly improve the performance of any method by deviating from these default parameters. For example, switching VAME's state-partition method from hidden Markov model (HMM) to k-means led to higher change score alignment (Extended Data Fig. 5a) but caused a decrease in alignment to supervised behavior labels (Fig. 5e,f shows performance under an HMM; performance under k-means is not shown). Our application of each method is described in detail below.

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 uniform manifold approximation and projection; (3) clusters points in the uniform manifold approximation and projection space; and (4) uses a classifier to extend the clustering to all frames[12]. We fit B-SOiD separately for each dataset. In each case, steps 2–4 were performed multiple times with different random seeds (see Supplementary Table 1 for number of fits per dataset), 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%. Because B-SOiD uses a hardcoded window of 100 ms to calculate pose and movement features, we reran the pipeline with falsely inflated frame rates for the window-size scan in Extended Data Fig. 5a. In all analyses involving B-SOiD, rare states (frequency < 0.5%) were excluded from the analysis.

VAME is a pipeline for behavioral clustering that: (1) preprocesses keypoint trajectories and transforms them into egocentric coordinates; (2) fits a recurrent neural network; (3) clusters the latent code of the recurrent neural network[13]. We applied these steps separately to each dataset, in each case running step 3 multiple times with different random seeds (see Supplementary Table 1 for number of fits per dataset). 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 = 30 for all datasets, except for the zdim-scan in Extended Data Fig. 5a. 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 identified by keypoint-MoSeq, except when scanning over state number in Extended Data Fig. 5a. In all analyses involving VAME, rare states (frequency < 0.5%) were excluded from analysis.

MotionMapper performs unsupervised behavioral segmentation by: (1) applying a wavelet transform to preprocessed pose data; (2) nonlinearly embedding the transformed data in 2D; and (3) clustering the 2D data with a watershed transform[17]. We applied these steps separately to each dataset, in each case running steps 2–3 multiple times with different random seeds (see Supplementary Table 1 for number of fits per dataset). There are several published implementations of MotionMapper, which perform essentially the same set of transformations but differ in programming language. We obtained similar results from a recent Python implementation from the Berman laboratory (https://github.com/bermanlabemory/motionmapperpy/) and a published MATLAB implementation[30]. All results in the paper are from the Python implementation, which we applied as follows. Data were first egocentrically aligned along the tail–nose axis and then projected into eight dimensions using PCA. Ten log-spaced frequencies between 0.25 Hz and 15 Hz were used for the wavelet transform, and dimensionality reduction was performed using t-distributed stochastic neighbor embedding. The threshold for watershedding was chosen 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. 5a, we varied 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 (https://github.com/bermanlabemory/motionmapperpy/blob/master/demo/motionmapperpy_mouse_demo.ipynb/).

Predicting kinematics from state sequences

We trained decoding models based on spline regression to predict kinematic parameters (height, velocity and turn speed) from state sequences output by keypoint-MoSeq and other behavior segmentation

methods (Fig. 3e and Extended Data Fig. 5c). 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 six log-spaced knot locations (1.0, 2.0, 3.9, 7.7, 15.2 and 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' 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, aligning the first frame of B to frame t_s in the design matrix. Kinematic features were regressed against the design matrix using Ridge regression from scikit-learn and fivefold 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 hind legs and extended their head upwards, leading to a transient increase in height from its modal value of 3–5 cm to a peak of 7–10 cm. 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 5 cm to above 7 cm that occurred within the span of a second, with onset formally defined as the first frame where the height exceeded 5 cm. Next, rear offsets were defined as decreases in height from above 7 cm to below 5 cm that occurred within the span of a second, with offset formally defined as the first frame where the height fell below 7 cm. Finally, we defined complete rears as onset–offset pairs defining an interval with length between 0.5 s and 2 s. 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 and r_r (yaw, pitch and roll) and accelerations a_x, a_y and a_z (dorsal/ventral, posterior/anterior and 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 three 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:

$$\omega = \left(\frac{\mathrm{d}r_y}{\mathrm{d}t}, \frac{\mathrm{d}r_p}{\mathrm{d}t}, \frac{\mathrm{d}r_r}{\mathrm{d}t}\right)$$

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

$$\text{Jerk} = \left|\frac{\mathrm{d}\tilde{a}}{\mathrm{d}t}\right|$$

Aligning dopamine fluctuations to behavior states

For a detailed description of photometry data acquisition and preprocessing, see ref. 22. Briefly, photometry signals were: (1) normalized using $\Delta F/F_0$ with a 5-s window; (2) adjusted against a reference to remove motion artifacts and other non-ligand-associated fluctuations; (3) z-scored using a 20-s sliding window; and (4) temporally aligned to the 30-Hz 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. 4d) and temporal asymmetry (Fig. 4e) as

$$\text{Temporal asymmetry} = \frac{1}{15}\sum_{t=0}^{15} d_t - \frac{1}{15}\sum_{t=-15}^{0} d_t$$

$$\text{Total fluctuation size} = \sum_{t=-15}^{15} |d_t|$$

A third metric—the average dopamine during each state (Extended Data Fig. 7b)—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. 5a–c) were obtained from ref. 31. The dataset contains 20 videos that are each 10–20-min long. Each video includes frame-by-frame annotations of five possible behaviors: locomote, rear, face groom, body groom and defecate. We excluded 'defecate' from the analysis because it was extremely rare (<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[47]. Specifically, we used SuperAnimal-TopViewMouse that applies DLCRNet-50 as the pose estimation model. 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 (that is, the image height resized to 400, 500 and 600 with a fixed aspect ratio) and rapid self-training (model trained on zero shot predictions with confidence above 0.1) for 1,000 iterations to counter domain shift and reduce jittering predictions.

Keypoint coordinates and behavioral annotations for the supervised social behavior benchmark (Fig. 5d–f) were obtained from the CalMS21 dataset[32] (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 fitted to 2D keypoint data for the resident mouse.

We used four metrics[13] to compare supervised annotations and unsupervised states from each method. These included NMI, homogeneity, adjusted rand score and purity. All metrics besides purity were computed using the Python library scikit-learn (that is, with the function normalized_mutual_info_score, homogeneity_score, adjusted_rand_score). The purity score was defined as in ref. 13.

Thermistor signal processing

During respiration, the movement of air through a mouse's nasal cavity generates fluctuations in temperature that can be detected by a thermistor; temperature decreases during inhalations (because the mouse is warmer than the air around it) and rises between inhalations. Below we refer to the between-inhalation intervals as 'exhales' but note that they may also contain pauses in respiration—pauses and exhales likely cannot be distinguished because warming of the thermistor occurs whether or not air is flowing.

To segment inhales and exhales using the thermistor signal, we first applied a 60-Hz notch filter (scipy.signal.iirnotch, $q = 10$) and a low-pass

filter (scipy.signal.butter, order = 3, cutoff = 40 Hz, analog = false) to the raw signal, and then used a median filter to subtract the slow DC offset component of the signal. We then performed peak detection using scipy.signal.find_peaks (minimium inter-peak distance of 50 ms, minimum and maximum widths of 10 ms and 1,500 ms, respectively). To distinguish true peaks (inhalation onsets) from spurious peaks (noise), we varied the minimum prominence parameter from 10^{-4} to 1 while keeping other parameters fixed, and then used the value at which the number of peaks stabilized. Using the chosen minimum prominence, the signal was then analyzed twice—once at the chosen value, and again with a slightly more permissive minimum prominence (1/8 of the chosen value). Any low-amplitude breaths detected with the more permissive setting that overlapped with periods of breathing between 1 Hz and 6 Hz were added to the detections. This same process was then repeated to find exhale onsets but with the thermistor signal inverted. Finally, inhales and exhales were paired, and any instances of two inhales/exhales in a row were patched by inserting an exhale/inhale at the local extremum between them. Detections were then inspected manually, and any recordings with excessive noise, unusually high breathing rates (>14 Hz), or unusual autocorrelation profiles were removed from further analyses.

Classifying sniff-aligned syllables

To test whether syllables were significantly sniff aligned, we compared the probability of inhalation in the 50 ms before versus 50 ms after syllable onset. Specifically, for each syllable, we quantified the pre-inhalation versus post-inhalation fraction across all instances of that syllable, and then compared the pre-distribution and post-distribution values using a paired t-test. Syllables with $P < 0.001$ were considered significant.

Fly gait analysis

For the analysis of fly behavior, we used a published dataset of keypoint coordinates[39], which were derived from behavioral videos originally reported in ref. 17. The full dataset contains 1-h recordings (100 fps) of single flies moving freely on a backlit 100-mm-diameter arena. Keypoints were tracked using LEAP (test accuracy ~2.5 px). MotionMapper results (including names for each cluster) were also included in the published dataset. We chose four 1-h sessions (uniformly at random) for analysis with keypoint-MoSeq. All results reported here were derived from this 4-h dataset.

The analysis of syllable probabilities across the stride cycle (Fig. 6i–k) was limited to periods of 'fast locomotion', as defined by the MotionMapper labeling (state label 7). To identify the start and end of each stride cycle, we applied PCA to egocentric keypoint coordinates (restricted to fast locomotion frames). We found that the first PC oscillated in a manner reflecting the fly's gait, and thus smoothed the first PC using a one-frame Gaussian filter and performed peak detection on the smoothed signal. Each inter-peak interval was defined as one stride. Stances and swings (Fig. 6j and Extended Data Fig. 10g) were defined by backward and forward motion of the leg tips, respectively (in egocentric coordinates).

Mathematical notation

1. $\chi^{-2}(\nu, \tau^2)$ denotes the scaled inverse Chi-squared distribution.
2. $\otimes$ denotes the Kronecker product.
3. Δ^N is the N-dimensional simplex.
4. I_N is the $N \times N$ identity matrix.
5. $\mathbf{1}_{N \times M}$ is the $N \times M$ matrix of ones.
6. $\mathbf{x}_{t_1:t_2}$ denotes the concatenation $[\mathbf{x}_{t_1}, \mathbf{x}_{t_1+1}, \ldots, \mathbf{x}_{t_2}]$ where $t_1 < t_2$.

Generative model

Keypoint-MoSeq learns syllables by fitting an SLDS model[48], which decomposes an animal's pose trajectory into a sequence of stereotyped dynamical motifs. In general, SLDS models explain time-series observations $y_1, \ldots, y_T$ through a hierarchy of latent states, including continuous states $\mathbf{x}_t \in \mathbb{R}^M$ that represent the observations y_t in a low-dimensional space, and discrete states $z_t \in \{1, \ldots, N\}$ that govern the dynamics of $\mathbf{x}_t$ over time. In keypoint-MoSeq, the discrete states correspond to syllables, the continuous states correspond to pose, and the observations are keypoint coordinates. We further adapted SLDS by (1) including a sticky hierarchical Dirichlet prior (HDP); (2) explicitly modeling the animal's location and heading; and (3) including a robust (heavy-tailed) observation distribution for keypoints. Below we review SLDS models in general and then describe each of the customizations implemented in keypoint-MoSeq.

SLDSs

The discrete states $z_t \in \{1, \ldots, N\}$ are assumed to form a Markov chain, meaning

$$z_{t+1} | z_t \sim \mathrm{Cat}(\pi_{z_t})$$

where $\pi_i \in \Delta^N$ is the probability of transitioning from discrete state i to each other state. Conditional on the discrete states z_t, the continuous states x_t follow an L-order vector autoregressive process with Gaussian noise. This means that the expected value of each x_t is a linear function of the previous L states $\mathbf{x}_{t-L:t-1}$, as shown below

$$x_t | z_t, x_{t-L:t-1} \sim \mathcal{N}(A_{z_t} x_{t-L:t-1} + b_{z_t}, Q_{z_t})$$

where $A_i \in \mathbb{R}^{M \times LM}$ is the autoregressive dynamics matrix, $\mathbf{b}_i \in \mathbb{R}^M$ is the dynamics bias vector, and $Q_i \in \mathbb{R}^{M \times M}$ is the dynamics noise matrix for each discrete state $i = 1, \ldots, N$. The dynamics parameters A_i, $\mathbf{b}_i$ and Q_i have a matrix normal inverse Wishart (MNIW) prior

$$[A_i | \mathbf{b}_i], Q_i \sim \mathrm{MNIW}(\nu_0, S_0, M_0, K_0)$$

where $\nu_0 > \mathrm{M} - 1$ is the degrees of freedom, $S_0 \in \mathbb{R}^{M \times M}$ is the prior covariance matrix, $M_0 \in \mathbb{R}^{M \times (LM+1)}$ is the prior mean dynamics matrix, and $K_0 \in \mathbb{R}^{(LM+1) \times (LM+1)}$ is the prior scale matrix. Finally, in the standard formulation of SLDS (which we modify for keypoint data, as described below), each observation $\mathbf{y}_t \in \mathbb{R}^D$ is a linear function of $\mathbf{x}_t$ plus noise:

$$\mathbf{y}_t | z_t, \mathbf{x}_t \sim \mathcal{N}(C\mathbf{x}_t + \mathbf{d}, S)$$

Here we assume that the observation parameters C, $\mathbf{d}$ and S do not depend on z_t.

Sticky HDP

A key feature of depth Moseq is the use of a sticky-HDP prior for the transition matrix. In general, HDP priors allow the number of distinct states in a HMM to be inferred directly from the data. The 'sticky' variant of the HDP prior includes an additional hyperparameter κ that tunes the frequency of self-transitions in the discrete state sequence z_t, and thus the distribution of syllable durations. As in depth MoSeq, we implement a sticky-HDP prior using the weak limit approximation[49], as shown below:

$$\beta \sim \mathrm{Dir}(\gamma/N, \ldots, \gamma/N)$$
$$\pi_i | \beta \sim \mathrm{Dir}(\alpha\beta_1, \ldots, \alpha\beta_v + \kappa \ldots, \alpha\beta_N)$$

where κ is being added in the ith position. Here $\beta \in \Delta^N$ is a global vector of augmented syllable transition probabilities, and the hyperparameters γ, α and κ control the sparsity of states, the weight of the sparsity prior and the bias toward self-transitions, respectively.

SLDS for postural dynamics

Keypoint coordinates reflect not only the pose of an animal, but also its location and heading. To disambiguate these factors, we define a

canonical, egocentric reference frame in which the postural dynamics are modeled. The canonically aligned poses are then transformed into global coordinates using explicit centroid and heading variables that are learned by the model.

Concretely, let $Y_t \in \mathbb{R}^{K\times D}$ represent the coordinates of K keypoints at time t, where $D \in \{2, 3\}$. We define latent variables $\mathbf{v}_t \in \mathbb{R}^D$ and $h_t \in [0, 2\pi]$ to represent the animal's centroid and heading angle. We assume that each heading angle h_t has an independent, uniform prior and that the centroid is autocorrelated as follows:

$$h_t \sim \text{Unif}(0, 2\pi)$$
$$\mathbf{v}_t|\mathbf{v}_{t-1} \sim \mathcal{N}\left(\mathbf{v}_{t-1}, \sigma^2_{\text{loc}}\right)$$

At each time point t, the pose Y_t is generated via rotation and translation of a centered and oriented pose $\tilde{Y}_t$ that depends on the current continuous latent state $\mathbf{x}_t$:

$$Y_t = \tilde{Y}_t R(h_t) + \mathbf{1}_K \mathbf{v}_t^\top \text{ where } \text{vec}(\tilde{Y}_t) \sim \mathcal{N}((\Gamma \otimes I_D)(C\mathbf{x}_t + \mathbf{d}), S_t)$$

where $R(h_t)$ is a matrix that rotates by angle h_t in the xy plane, and $\Gamma \in R^{K\times(K-1)}$ is defined by the truncated singular value decomposition $\Gamma\Delta\Gamma^\top = I_K - \mathbf{1}_{K\times K}/K$. Note that Γ encodes a linear transformation that isometrically maps $\mathbb{R}^{(K-1)\times D}$ to the set of all centered keypoint arrangements in $\mathbb{R}^{K\times D}$, and thus ensures that $\mathbb{E}(\tilde{Y}_t)$ is always centered[50]. The parameters $C \in \mathbb{R}^{(K-1)D\times M}$ and $\mathbf{d} \in \mathbb{R}^{(K-1)D}$ are initialized using PCA applied to the transformed keypoint coordinates $\Gamma^T\tilde{Y}_t$. In principle C and $\mathbf{d}$ can be adjusted further during model fitting, and we describe the corresponding Gibbs updates in the inference section below. In practice, however, we keep C and $\mathbf{d}$ fixed to their initial values when fitting keypoint-MoSeq.

Robust observations

To account for occasional large errors during keypoint tracking, we use the heavy-tailed Student's t-distribution, which corresponds to a normal distribution whose variance is itself a random variable. Here, we instantiate the random variances explicitly as a product of two parameters: a baseline variance σ_k for each keypoint and a time-varying scale $s_{t,k}$. We assume:

$$\sigma_k^2 \sim \chi^{-2}\left(\nu_\sigma, \sigma_0^2\right)$$
$$s_{t,k}^2 \sim \chi^{-2}\left(\nu_s, s_{0,t,k}\right)$$

where $\nu_\sigma > 0$ and $\nu_s > 0$ are degrees of freedom, $\sigma_0^2 > 0$ is a baseline scaling parameter, and $s_{0,t,k} > 0$ is a local scaling parameter, which encodes a prior on the scale of error for each keypoint on each frame. Where possible, we calculated the local scaling parameters as a function of the neural network confidences for each keypoint. The function was calibrated using the empirical relationship between confidence values and error sizes. The overall noise covariance S_t is generated from σ_k and $s_{t,k}$ as follows:

$$S_t = \text{diag}\left(\sigma_1^2 s_{t,1}^2, \ldots, \sigma_K^2 s_{t,K}^2\right) \otimes I_D$$

Related work

Keypoint-MoSeq extends the model used in depth MoSeq[16], where a low-dimensional pose trajectory x_t (derived from egocentrically aligned depth videos) is used to fit an AR-HMM with a transition matrix π, autoregressive parameters A_i, $\mathbf{b}_i$ and Q_i and discrete states z_t like those described here. Indeed, conditional on $\mathbf{x}_t$, the models for keypoin-MoSeq and depth MoSeq are identical. The main differences are that keypoint-MoSeq treats $\mathbf{x}_t$ as a latent variable (that is, updates it during fitting), includes explicit centroid and heading variables, and uses a robust noise model.

Disambiguating poses from position and heading is a common task in unsupervised behavior algorithms, and researchers have adopted a variety of approaches. VAME[13], for example, isolates pose by centering and aligning data ahead of time, whereas B-SOiD[12] transforms the keypoint data into a vector of relative distances and angles. The statistical pose model GIMBAL[29], on the other hand, introduces latent heading and centroid variables that are inferred simultaneously with the rest of the model. Keypoint-MoSeq adopts this latter approach, which can remove spurious correlations between egocentric features that can arise from errors in keypoint localization.

Inference algorithm

Our full model contains latent variables $\mathbf{v}$, h, $\mathbf{x}$, z and s and parameters A, $\mathbf{b}$, Q, C, $\mathbf{d}$, σ, β and π. We fit each of these variables—except for C and $\mathbf{d}$—using Gibbs sampling, in which each variable is iteratively resampled from its posterior distribution conditional on the current values of all the other variables. The posterior distributions $P(\pi, \beta|z)$ and $P(A, b, Q|z, x)$ are unchanged from the original MoSeq paper and will not be be reproduced here (see ref. 16, pages 42–44, and note the changes of notation $Q \to \Sigma$, $z \to x$ and $\mathbf{x} \to \mathbf{y}$). The Gibbs updates for variables C, $\mathbf{d}$, σ, s, $\mathbf{v}$ and h are described below.

Resampling $P(C, d|s, \sigma, x, v, h, Y)$. Let $\tilde{\mathbf{x}}_t$ represent $\mathbf{x}_t$ with a 1 appended and define

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

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

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

with

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

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

$$s_{t,k}|C, \mathbf{d}, \sigma_k, \mathbf{x}, Y \sim \chi^{-2}\left(\nu_s + D, \left(\nu_s s_0 + \sigma_k^{-2} \| \left(\Gamma(C\mathbf{x}_t + \mathbf{d})\right)_k - \tilde{Y}_{t,k}\|^2\right)/(\nu_s + D)\right)$$

Resampling $P(\sigma|C, d, s, x, v, h, Y)$. Each σ_k is conditionally independent with posterior

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

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

Resampling $P(v|C, d, \sigma, s, x, h, Y)$. Because the translations $\mathbf{v}_1, \ldots, \mathbf{v}_T$ form an LDS, they can be updated by Kalman sampling. The observation potentials have the form $\mathcal{N}(\mathbf{v}_t|\mu, \gamma^2 I_D)$ where

$$\mu = \sum_k \frac{\gamma_t^2}{\sigma_k^2 s_{t,k}} [Y_{t,k} - R(h_t)^\top \Gamma(C\mathbf{x}_t + \mathbf{d})_k], \frac{1}{\gamma_t^2} = \sum_k \frac{1}{\sigma_k^2 s_{t,k}}$$

Resampling $P(h|C, d, \sigma, s, x, v, Y)$. 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

$$S = \sum_k \frac{1}{s_{t,k}\sigma_k^2} \Gamma(C\mathbf{x}_t + \mathbf{d})_k (Y_{t,k} - \mathbf{v}_t)^\top$$

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

$$A' = \begin{bmatrix} I & & & \\ & I & & \\ & & I & \\ A_1 & A_2 & \dots & A_L \ \mathbf{b} \end{bmatrix} \quad Q' = \begin{bmatrix} 0 & & & \\ & 0 & & \\ & & 0 & \\ & & & Q \end{bmatrix} \quad C' = \begin{bmatrix} 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ C & \mathbf{d} \end{bmatrix} \quad \mathbf{x}'_t = \begin{bmatrix} \mathbf{x}_{t-L+1} \\ \vdots \\ \mathbf{x}_t \\ 1 \end{bmatrix}$$

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

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

(Assume $\mathbf{x}'$ is left-padded with zeros for negative time indices.)

Hyperparameters

We used the following hyperparameter values throughout the paper.

Transition matrix.

$N = 100$

$\gamma = 1,000$

$\alpha = 100$

κ fit to each dataset

Autoregressive process.

M set using PCA explained variance curve

$L = 3$

$\nu_0 = M + 2$

$S_0 = 0.01 I_M$

$M_0 = [0_{M\times(L-1)}\, I_M\, 1_{M\times 1}]$

$K_0 = 10 I_{M(L+1)}$

Observation process.

$\sigma_0^2 = 1$

$\nu_\sigma = 10^5$

$\nu_s = 5$

$s_{0,t,k}$ set based on neural network confidence

Centroid autocorrelation.

$$\sigma^2_{\mathrm{loc}} = 0.4$$

Derivation of Gibbs updates

Derivation of C, d updates. To simply notation, define

$$\tilde{S}_t = \mathrm{diag}(\sigma_1^2 s_{t,1}, \dots, \sigma_K^2 s_{t,K}),\ \tilde{\mathbf{x}}_t = (\mathbf{x}_t, 1),\ \tilde{C} = (C, \mathbf{d})$$

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

$$\begin{aligned} P(\tilde{Y}|\tilde{C}, \tilde{\mathbf{x}}, \tilde{S}) &= \prod_{t=1}^{T} \mathcal{N}\left(\mathrm{vec}\left(\tilde{Y}_t\right)|\left(\Gamma \otimes I_D\right)\tilde{C}\tilde{\mathbf{x}}_t, \tilde{S}_t \otimes I_D\right) \\ &\propto \exp\left[-\frac{1}{2}\sum_{t=1}^{T}\left(\tilde{\mathbf{x}}_t^\top \tilde{C}^\top \left(\Gamma^\top \tilde{S}_t^{-1}\Gamma \otimes I_D\right)\tilde{C}\tilde{\mathbf{x}}_t - 2\mathrm{vec}(\tilde{Y}_t)^\top\left(\tilde{S}_t^{-1}\Gamma \otimes I_D\right)\tilde{C}\tilde{\mathbf{x}}_t\right)\right] \\ &\propto \exp\left[-\frac{1}{2}\sum_{t=1}^{T}\left(\mathrm{vec}(\tilde{C})^\top\left(\tilde{\mathbf{x}}_t\tilde{\mathbf{x}}_t^\top \otimes \Gamma^\top\tilde{S}_t^{-1}\Gamma \otimes I_D\right)\mathrm{vec}\left(\tilde{C}\right)\right)\right. \\ &\quad \left.\left(-2\mathrm{vec}(\tilde{C})^\top\left(\tilde{\mathbf{x}}_t^\top \otimes \tilde{S}_t^{-1}\Gamma \otimes I_D\right)\mathrm{vec}\left(\tilde{Y}_t\right)\right)\right] \\ &\propto \exp\left[-\frac{1}{2}\left(\mathrm{vec}(\tilde{C})^\top S_{x,x}\mathrm{vec}\left(\tilde{C}\right) - 2\mathrm{vec}(\tilde{C})^\top S_{x,y}\right)\right] \end{aligned}$$

where

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

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

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

where

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

Derivation of σ_k, $s_{t,k}$ updates. For each time t and keypoint k, let $\bar{Y}_{t,k} = \Gamma\left(C\mathbf{x}_t + \mathbf{d}\right)$. The likelihood of the centered and aligned keypoint location $\tilde{Y}_{t,k}$ is

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

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

$$\begin{aligned} P(s_{t,k}|\sigma_k, \tilde{Y}_{t,k}, \bar{Y}_{t,k}) &\propto \chi^{-1}(s_{t,k}|\nu_s, s_0)\,\mathcal{N}\left(\tilde{Y}_{t,k}|\bar{Y}_{t,k}, \sigma_k^2 s_{t,k} I_D\right) \\ &\propto s_{t,k}^{-1-(\nu_s+D)/2}\exp\left[\frac{-\nu_s s_0}{2s_{t,k}} - \frac{\|\tilde{Y}_{t,k} - \bar{Y}_{t,k}\|^2}{2\sigma_k^2 s_{t,k}}\right] \\ &\propto \chi^{-2}\left(s_{t,k}|\nu_s + D, \left(\nu_s s_0 + \sigma_k^{-2}\,\|\,\tilde{Y}_{t,k} - \bar{Y}_{t,k}\|^2\right)(\nu_s + D)^{-1}\right) \end{aligned}$$

$$\begin{aligned} P\left(\sigma_k|\{s_{t,k}, \tilde{Y}_{t,k}, \bar{Y}_{t,k}\}_{t=1}^{T}\right) &\propto \chi^{-1}\left(\sigma_k^2|\nu_\sigma, \sigma_0^2\right)\prod_{t=1}^{T}\mathcal{N}\left(\tilde{Y}_{t,k}|\bar{Y}_{t,k}, \sigma_k^2 s_{t,k} I_D\right) \\ &\propto \sigma_k^{-2-\nu_\sigma - DT}\exp\left[\frac{-\nu_\sigma\sigma_0^2}{2\sigma_k^2} - \frac{1}{2\sigma_k^2}\sum_{t=1}^{T}\frac{\|\tilde{Y}_{t,k} - \bar{Y}_{t,k}\|^2}{s_{t,k}}\right] \\ &\propto \chi^{-2}\left(\sigma_k^2|\nu_\sigma + DT, \left(\nu_\sigma\sigma_0^2 + S_y\right)(\nu_\sigma + DT)^{-1}\right) \end{aligned}$$

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

Derivation of v_t update. We assume an improper uniform prior on v_t, hence

$$\begin{aligned} P(\mathbf{v}_t|Y_t) &\propto P(Y_t|\mathbf{v}_t)P(\mathbf{v}_t) \propto P(Y_t|\mathbf{v}_t) \\ &\propto \mathcal{N}\left(\mathrm{vec}\left(\left(Y_t - \mathbf{1}_K\mathbf{v}_t^\top\right)R(h_t)^\top\right)|\Gamma\left(C\mathbf{x}_t + \mathbf{d}\right), S_t\right) \\ &= \prod_k \mathcal{N}\left(R(h_t)\left(Y_{t,k} - \mathbf{v}_t\right)|\Gamma\left(C\mathbf{x}_t + \mathbf{d}\right)_k, s_{t,k}\sigma_k^2 I_D\right) \\ &= \prod_k \mathcal{N}\left(v_t|Y_{t,k} - R(h_t)^\top\Gamma(C\mathbf{x}_t + \mathbf{d})_k, s_{t,k}\sigma_k^2 I_D\right) \\ &= \mathcal{N}\left(\mathbf{v}_t|\mu_t, \gamma_t^2 I_D\right) \end{aligned}$$

where

$$\mu = \sum_k \frac{\gamma_t^2}{\sigma_k^2 s_{t,k}}\left(Y_{t,k} - R(h_t)^\top\Gamma(C\mathbf{x}_t + \mathbf{d})_k\right),\ \frac{1}{\gamma_t^2} = \sum_k \frac{1}{\sigma_k^2 s_{t,k}}$$

Derivation of h_t update. We assume a proper uniform prior on h_t, hence

$$\begin{aligned} P(h_t|Y_t) &\propto P(Y_t|h_t)P(h_t) \propto P(Y_t|h_t) \\ &\propto \exp\left[\sum_k \frac{(Y_{t,k} - \mathbf{v}_t)^\top R(h_t)\Gamma(C\mathbf{x}_t + \mathbf{d})_k}{s_{t,k}\sigma_k^2}\right] \\ &= \exp\left[\frac{\mathrm{tr}\left[R(h_t)\Gamma(C\mathbf{x}_t + \mathbf{d})_k(Y_{t,k} - \mathbf{v}_t)^\top\right]}{s_{t,k}\sigma_k^2}\right] \\ &\propto \exp\mathrm{tr}\left[R(h_t)S\right] \text{ where } S = \sum_k \Gamma(C\mathbf{x}_t + \mathbf{d})_k(Y_{t,k} - \mathbf{v}_t)^\top/\left(s_{t,k}\sigma_k^2\right) \\ &\propto \exp\left[\cos(h_t)(S_{1,1} + S_{2,2}) + \sin(h_t)(S_{1,2} - S_{2,1})\right] \end{aligned}$$

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

$$P(Y_t|h_t) \propto \exp[\kappa\cos(h_t)\cos(\theta)+\sin(h_t)\sin(\theta)]$$
$$= \exp[\kappa\cos(h_t-\theta)] \propto \mathrm{vM}(h_t|\theta,\kappa)$$

Reporting summary

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

Data availability

This study used the following publicly available datasets: CalMS21 (https://data.caltech.edu/records/s0vdx-0k302)[32]; DeepEthogram benchmark data[31] (https://github.com/jbohnslav/deepethogram/); Rat7M (https://doi.org/10.6084/m9.figshare.c.5295370.v3)[51]; and fly keypoint tracking (https://doi.org/10.1038/s41592-018-0234-5). Other data raw data generated in this study have been deposited in Zenodo (https://doi.org/10.5281/zenodo.10636983)[52]. The thermistor recordings generated for this study are not publicly available at this time as they are being used for a follow-up paper. We plan to make these data publicly accessible upon publication of the follow-up study and in the meantime will provide them upon reasonable request.

Code availability

Software links and user support for both depth and keypoint data are available at http://www.moseq4all.org/. Data loading, project configuration and visualization are enabled through the keypoint-moseq[53] Python library (https://github.com/dattalab/keypoint-moseq/). We also developed a stand-alone library called jax-moseq[54] for core model inference (https://github.com/dattalab/jax-moseq/). Both libraries are freely available to the research community under an academic and non-commercial research use license. This license permits free academic and non-commercial use, explicitly prohibits redistribution and commercial use, and requires users to agree to terms including limitations on liability and indemnity. Full license details can be viewed on the respective GitHub repository pages.

References

44. Zhou, Z., et al. UNet++: a nested U-net architecture for medical image segmentation. in (eds Stoyanov, D. et al.) *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. DLMIA ML-CDS 2018. Lecture Notes in Computer Science, vol 11045, 3–11 (Springer International Publishing, 2018). https://doi.org/10.1007/978-3-030-00889-5_1
45. Sun, K., Xiao, B., Liu, D. & Wang, J. Deep high-resolution representation learning for human pose estimation. in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 5686–5696 (2019).
46. Nath, T. et al. Using DeepLabCut for 3D markerless pose estimation across species and behaviors. *Nat. Protoc.* **14**, 2152–2176 (2019).
47. Ye, S. et al. SuperAnimal pretrained pose estimation models for behavioral analysis. Preprint at https://arxiv.org/abs/2203.07436 (2023).
48. Ackerson, G. A. & Fu, K.-S. On state estimation in switching environments. *IEEE Trans. Autom. Control.* **15**, 10–17 (1970).
49. Fox, E. B., Sudderth, E. B., Jordan, M. I. & Willsky, A. S. A sticky HDP-HMM with application to speaker diarization. *Ann. Appl. Stat.* **5**, 1020–1056 (2009).
50. Andreella, A. & Finos, L. Procrustes analysis for high-dimensional data. *Psychometrika* **87**, 1422–1438 (2022).
51. Marshall, J. D. et al. Rat 7M. *figshare* https://doi.org/10.6084/m9.figshare.c.5295370.v3 (2021).
52. Weinreb, C. et al. Keypoint-MoSeq: parsing behavior by linking point tracking to pose dynamics. *Zenodo* https://doi.org/10.5281/zenodo.10636983 (2024).
53. Weinreb, C. et al. dattalab/keypoint-moseq: Keypoint MoSeq 0.4.3. *Zenodo* https://doi.org/10.5281/zenodo.10524840 (2024).
54. Weinreb, C. et al. dattalab/jax-moseq: JAX MoSeq 0.2.1. *Zenodo* https://doi.org/10.5281/zenodo.10403244 (2023).

Acknowledgements

S.R.D. is supported by National Institutes of Health (NIH) grants RF1AG073625, R01NS114020 and 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 laboratory of S.R.D. for useful comments on the paper. Portions of this research were conducted on the O2 High Performance Compute Cluster at Harvard Medical School. Mouse illustrations were downloaded from https://www.scidraw.io/.

Author contributions

C.W. and S.R.D. conceived the project and designed the experiments. C.W. and S.W.L. designed the algorithm. C.W. implemented the algorithm with contributions from S.L., M.A.M.O., L.Z. and T.P. C.W. and J.P. collected data and S.M., W.F.G., M.J., S.A., E.C. and R.H. assisted. C.W., J.P., M.A.M.O., Y.S., A.M., M.W.M. and T.P. performed analyses. C.W. and S.R.D. wrote the manuscript with input from all authors. S.R.D. supervised the project.

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. The other authors declare no competing interests.

Additional information

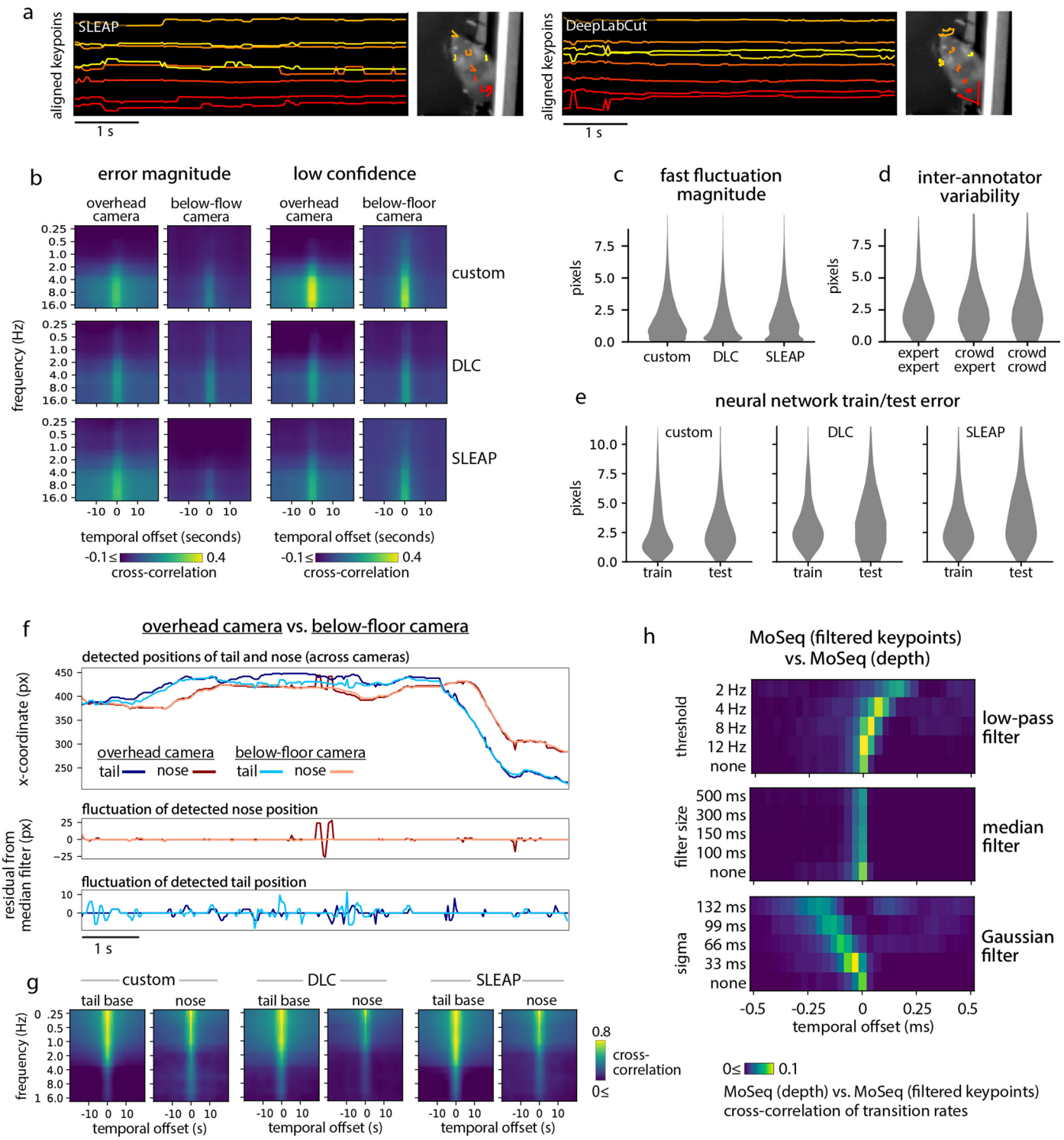
Extended data is available for this paper at https://doi.org/10.1038/s41592-024-02318-2.

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41592-024-02318-2.

Correspondence and requests for materials should be addressed to Scott W. Linderman or Sandeep Robert Datta.

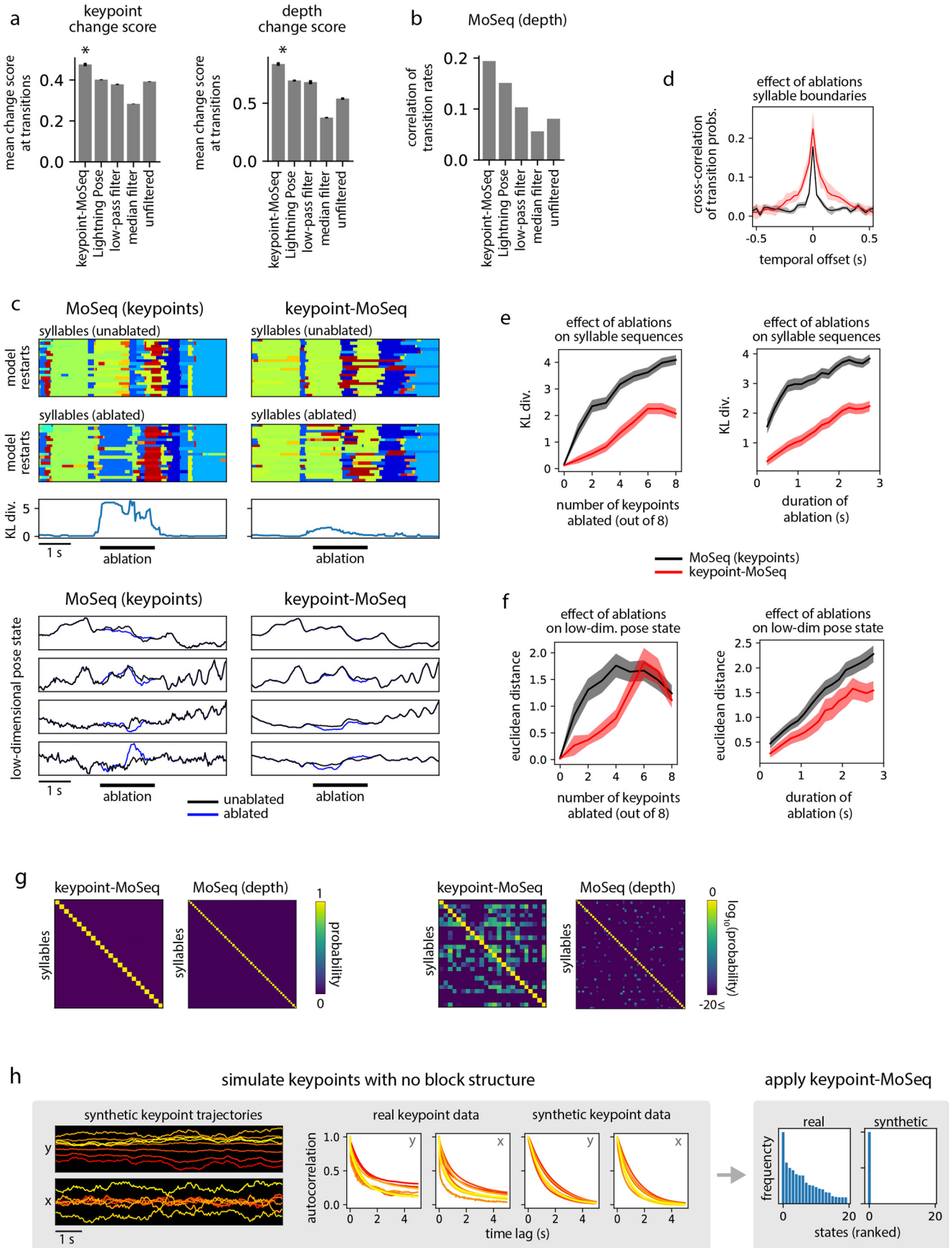
Peer review information *Nature Methods* thanks Matthew Smear and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Primary Handling Editor: Nina Vogt, in collaboration with the *Nature Methods* team.

https://doi.org/10.1038/s41592-024-02318-2



Extended Data Fig. 1 | Markerless pose tracking exhibits fast fluctuations that are independent of behavior yet affect MoSeq output. a) Example of a 5-second interval during which the mouse is still yet the keypoint coordinates fluctuate, as shown in Fig. 1e, but here for SLEAP and DeepLabCut respectively. **Left:** egocentrically aligned keypoint trajectories. **Right:** path traced by each keypoint during the 5-second interval. **b)** Cross-correlation between the spectral content of keypoint fluctuations and either error magnitude (left) or a measure of low-confidence keypoint detections (right). **c)** Magnitude of fast fluctuations in keypoint position for three different tracking methods, calculated as the per-frame distance from the detected trajectory of a keypoint to a smoothened version of the same trajectory, where smoothing was performed using a gaussian kernel with width 100ms (N=4 million keypoint detections). **d)** Inter-annotator variability, shown as the distribution of distances between multiple annotations of the same keypoint. Annotations were either crowd-sourced or obtained from experts (N=200 frames and N=4 labelers). **e)** Train- and test- error distributions for each keypoint tracking method (N=800 held out keypoint annotations). **f) 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. **g)** Cross-correlation between spectrograms obtained from two different camera angles for either the tail base or the nose, shown for each tracking method. **h)** Cross-correlation of transitions rates, comparing MoSeq applied to depth and MoSeq applied to keypoints with various levels of smoothing using a low-pass, Gaussian, or median filter (N=1 model fit per filtering parameter).

https://doi.org/10.1038/s41592-024-02318-2

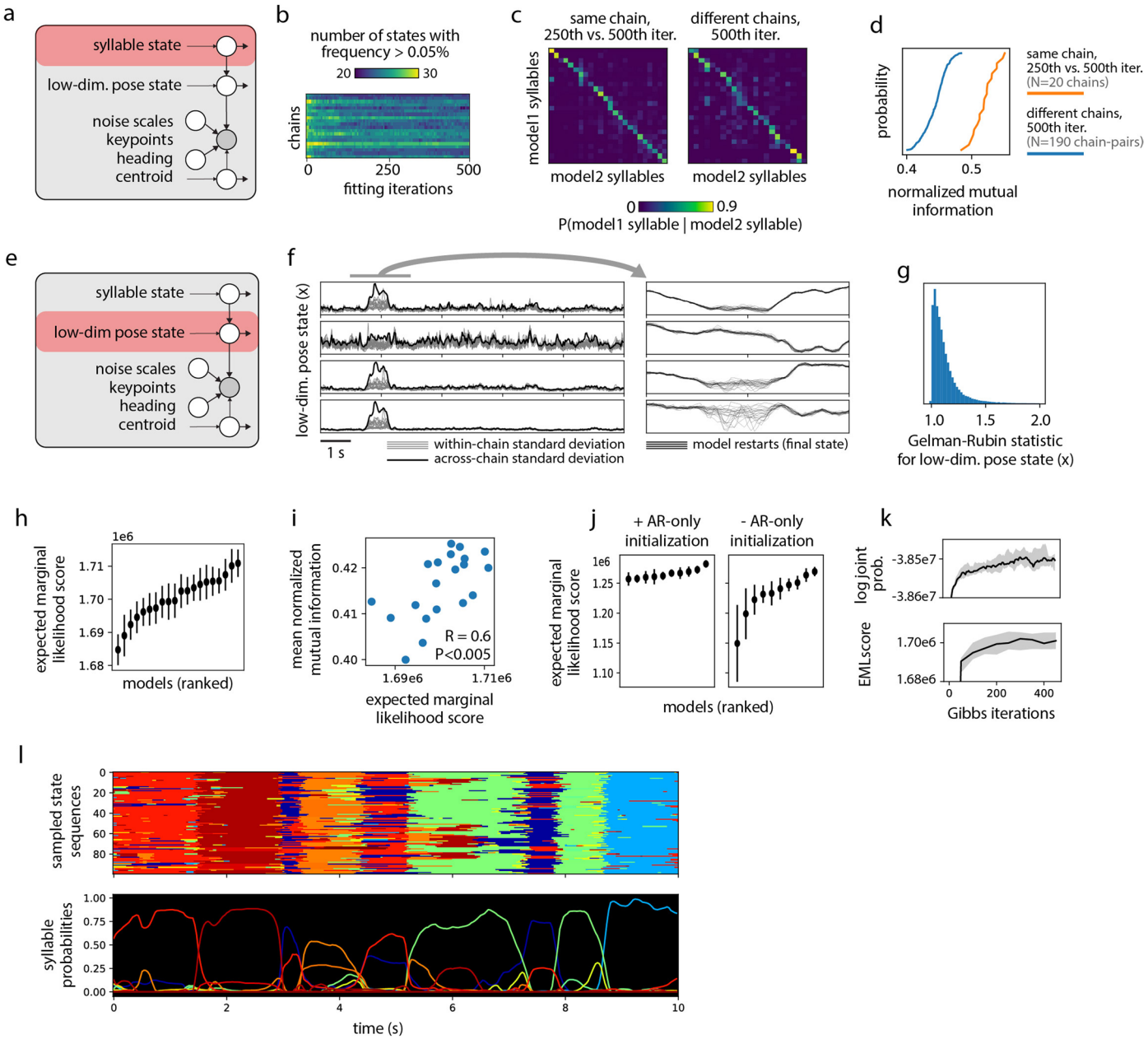


Extended Data Fig. 2 | See next page for caption.

https://doi.org/10.1038/s41592-024-02318-2

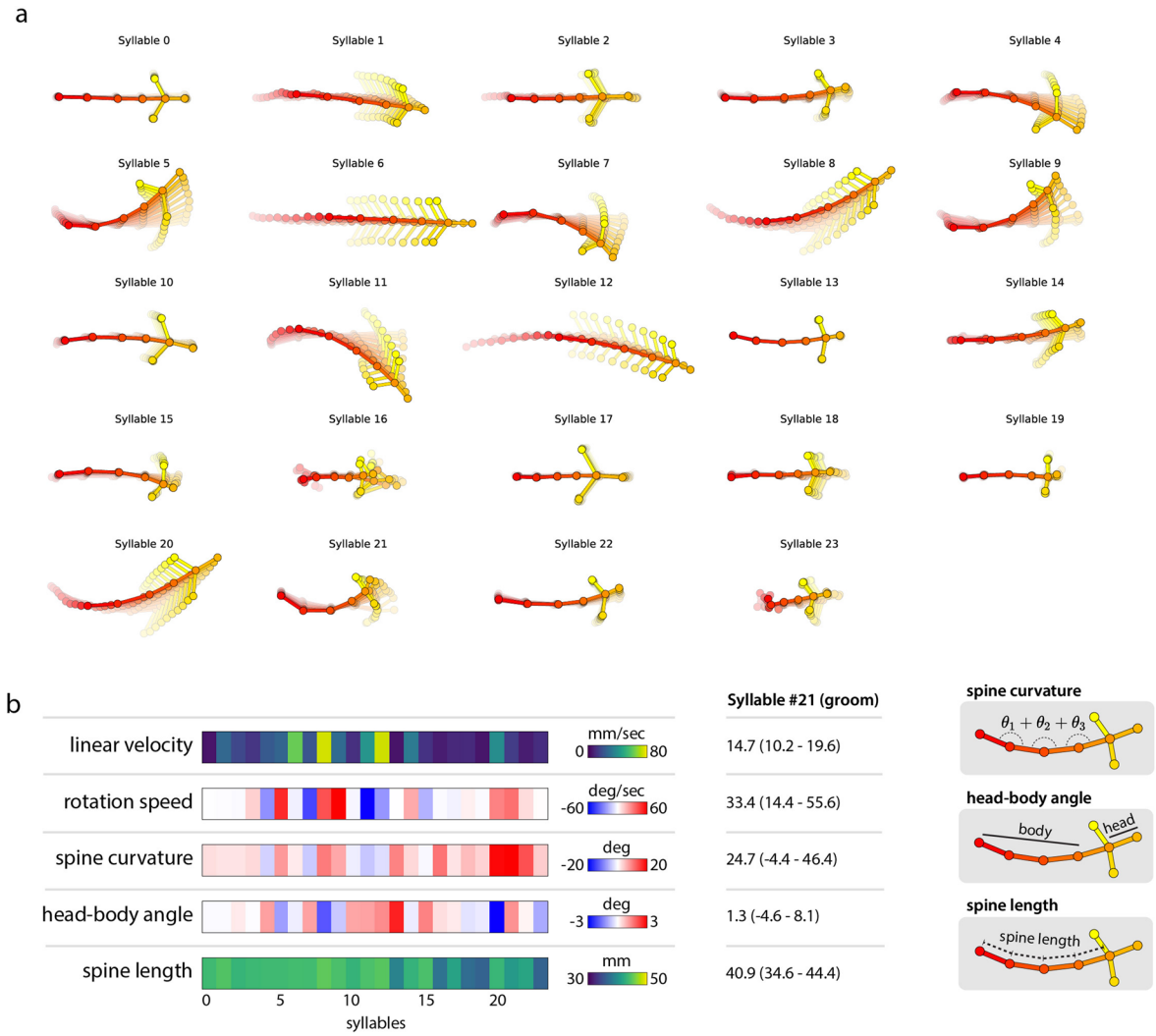
Extended Data Fig. 2 | Keypoint-MoSeq is robust to noise and missing data. **a)** Mean change score values at syllable transitions. Syllables were either derived from keypoint-MoSeq applied to (unfiltered) keypoints from our custom neural network, or from traditional MoSeq applied to several versions of the keypoint data, including keypoints inferred from Lightning Pose, or keypoints from our custom neural network followed by low-pass filtering, median filtering, or no filtering. Error bars show standard deviation across N=20 model fits. The change scores are highest for keypoint-MoSeq ($P < 10^{-4}$ over N=20 model fits, Mann-Whitney U test). **b)** Correlations of transition probabilities (that is, the probability of a new syllable starting at each frame), comparing depth MoSeq with each of the keypoint models shown in (a). **c)** Example of model responses to a one-second-long ablation of keypoint observations, shown for keypoint-MoSeq (right) and traditional AR-HMM-based MoSeq (left). **Top: Change in syllable sequences**. Each heatmap row represents an independent modeling run and each column represents a frame. The set of labels on each frame define a distribution, and the Kullback-Leibler divergence (KL div.) between the ablated and unablated distributions is plotted below. **Bottom: Change in low-dimensional pose state**. Estimated pose trajectories derived from unablated (black) or ablated (blue) data. Each dimension of the latent pose space is plotted separately. Lines reflect the mean across modeling runs. **d)** Cross-correlation of transition probabilities for ablated vs. unablated data (computed over frames that were included in an ablation), shown for keypoint-MoSeq (red) and traditional AR-HMM-based MoSeq (red), Shading shows bootstrap 95% confidence intervals for N=20 model fits. Solid line shows cross-correlation using all N=20 models (without bootstrapping). **e)** Mean Kullback-Leibler divergence [as described in (c)] across all ablation intervals, stratified by number of ablated keypoints (left) or duration of the ablation (right). Shading represents the 99% confidence interval of the mean. **f)** Mean distance between pose states estimated from ablated vs. unablated data, with colors and shading as in (e). **g)** 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). **h)** 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.

https://doi.org/10.1038/s41592-024-02318-2



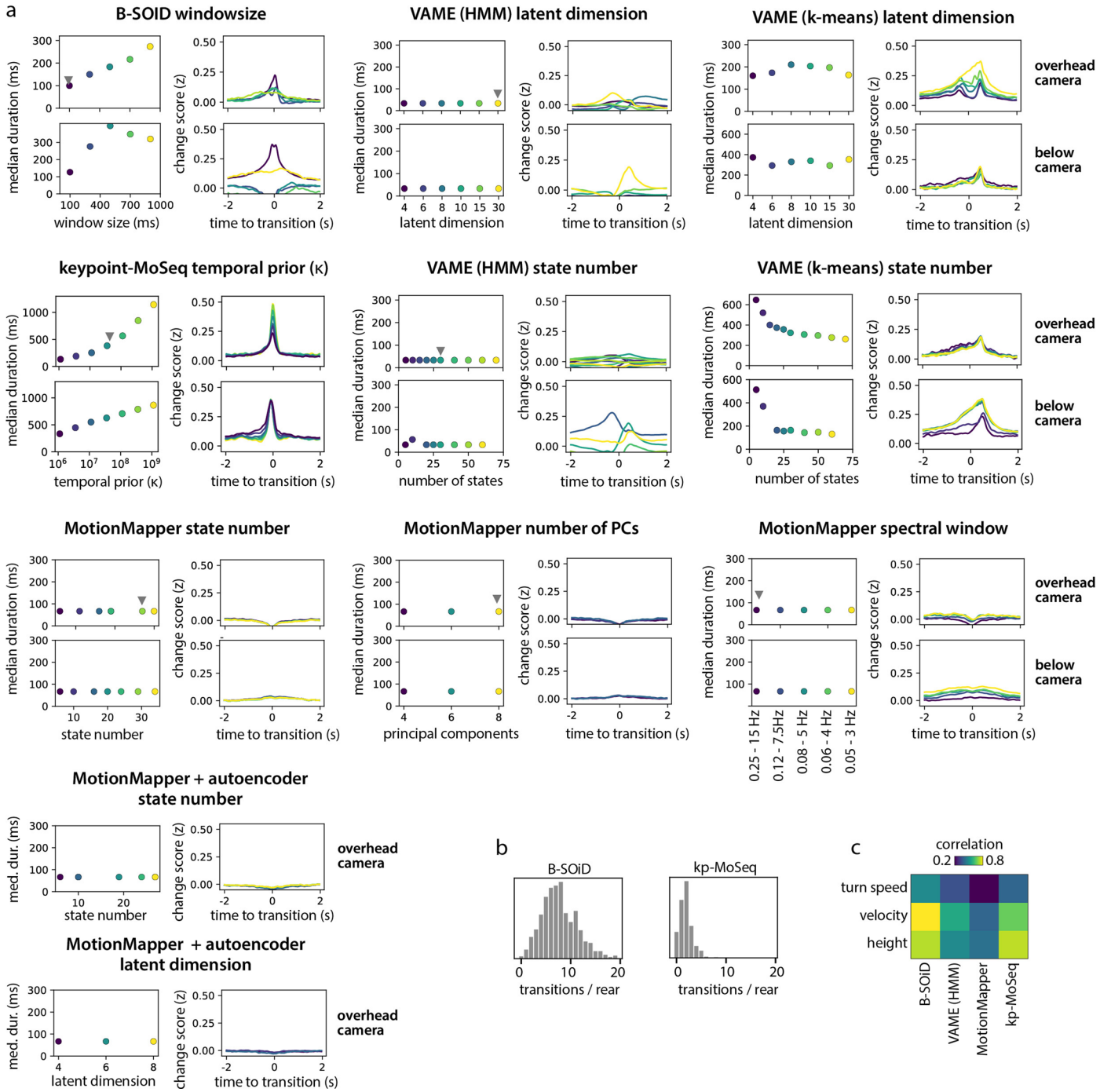
Extended Data Fig. 3 | Convergence and model selection. a) Probabilistic graphical model (PGM) for keypoint-MoSeq highlighting the discrete syllable state. **b)** Number of syllables identified by keypoint-MoSeq as a function of fitting iteration, shown for multiple independent runs of fitting (referred to as 'chains'). **c)** Confusion matrices depicting closer agreement between syllables from the same chain at different stages of fitting (left) compared to syllables from different chains at the final stage of fitting (right). **d)** Distributions of syllable sequence similarity [quantified by normalized mutual information (NMI)], either within chains at different iterations (N=20) or across chains (N=190). **e)** PGM highlighting pose state. **f) Left:** within- and between- chain variation in pose state, shown for each dimension of pose (rows) across an example 10-second interval. Gray lines represent the variation across fitting iterations within each chain, and black lines represent the total variation across chains and fitting iterations. **Right:** zoom-in on a 2-second interval showing close agreement in the final pose trajectory learned by each chain. **g)** Distribution of the Gelman-Rubin statistic (ratio of within-chain variance to total variance) across timepoints and dimensions of the pose state. **h)** Expected marginal likelihood (EML) scores (defined as a mean over marginal likelihoods) for the final model parameters learned by each chain. Vertical bars represent standard error based on N=20 chains. **i)** The scores shown in (h) correlate with mean NMI for each model, which is low when a model's syllable sequences are dissimilar from those of other models (P=0.005, Pearson test). **j)** EML scores are higher for models fit with an autoregressive-only (AR-only) initialization stage (left) compared to those without (right; P = 0.004, N=20 fits for each method, Mann-Whitney U test). Plotted as in (h). **k)** EML scores (bottom) plateau within 500 iterations of Gibbs sampling and have a similar trajectory as the model log joint probability (top). Black lines represent the median across N=20 chains and shaded regions represent inter-quartile interval. **l)** Illustration of uncertainty in syllable sequence given a fixed set of syllable definitions. **Top:** syllable sequences derived from Gibbs sampling (conditioning on fixed autoregressive parameters and transition probabilities), shown for an example 10-second window. **Bottom:** per-frame marginal probability estimates for each syllable. Each line is one syllable, with colors as in the heatmap above.

https://doi.org/10.1038/s41592-024-02318-2



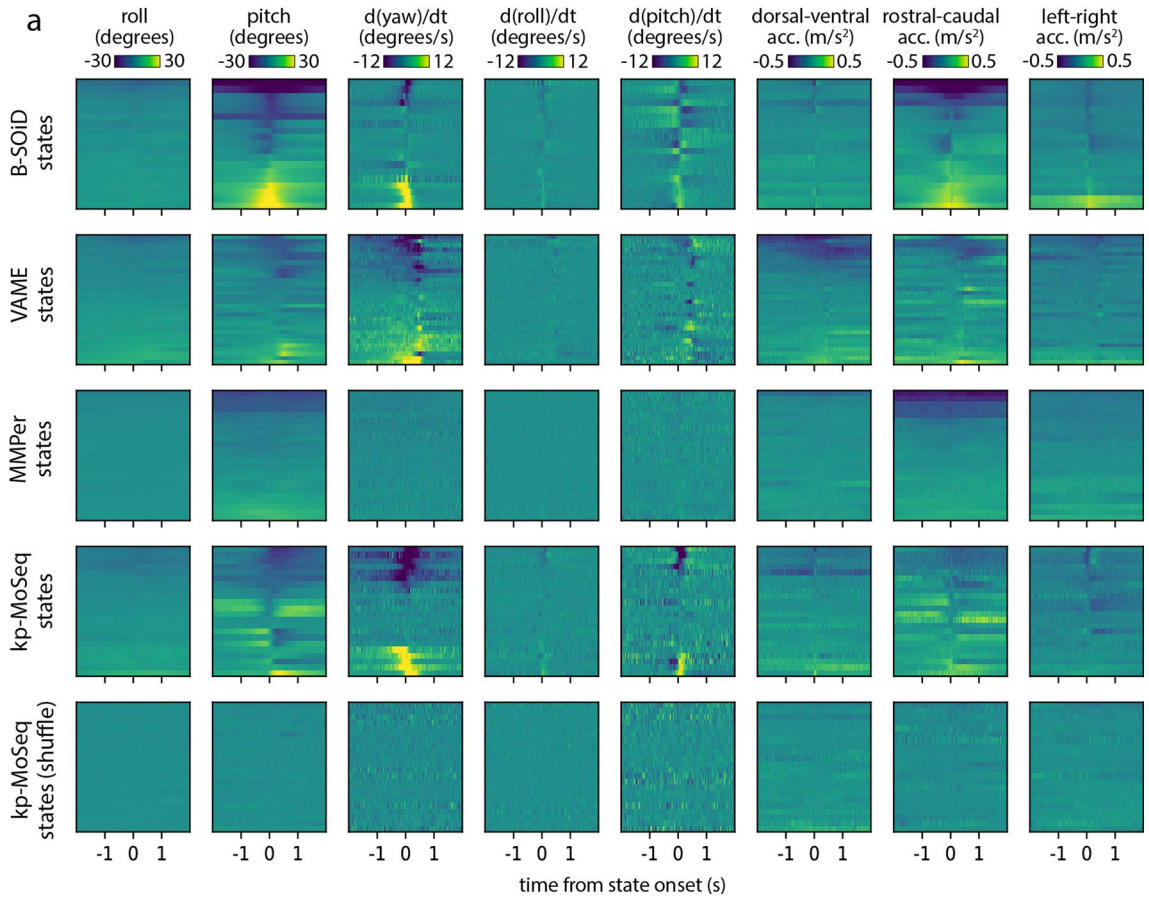
Extended Data Fig. 4 | Behaviors captured by keypoint-MoSeq syllables. **a)** Average pose trajectories for syllables identified by keypoint-MoSeq. Each trajectory includes ten evenly timed poses from 165ms before to 500ms after syllable onset. **b) Kinematic and morphological parameters for each syllable.** **Left:** Average values of five parameters (rows) for each syllable (column). **Middle:** Mean and interquartile range of each parameter for one example syllable. **Right:** cartoons illustrating the computation of the three morphological parameters.

https://doi.org/10.1038/s41592-024-02318-2



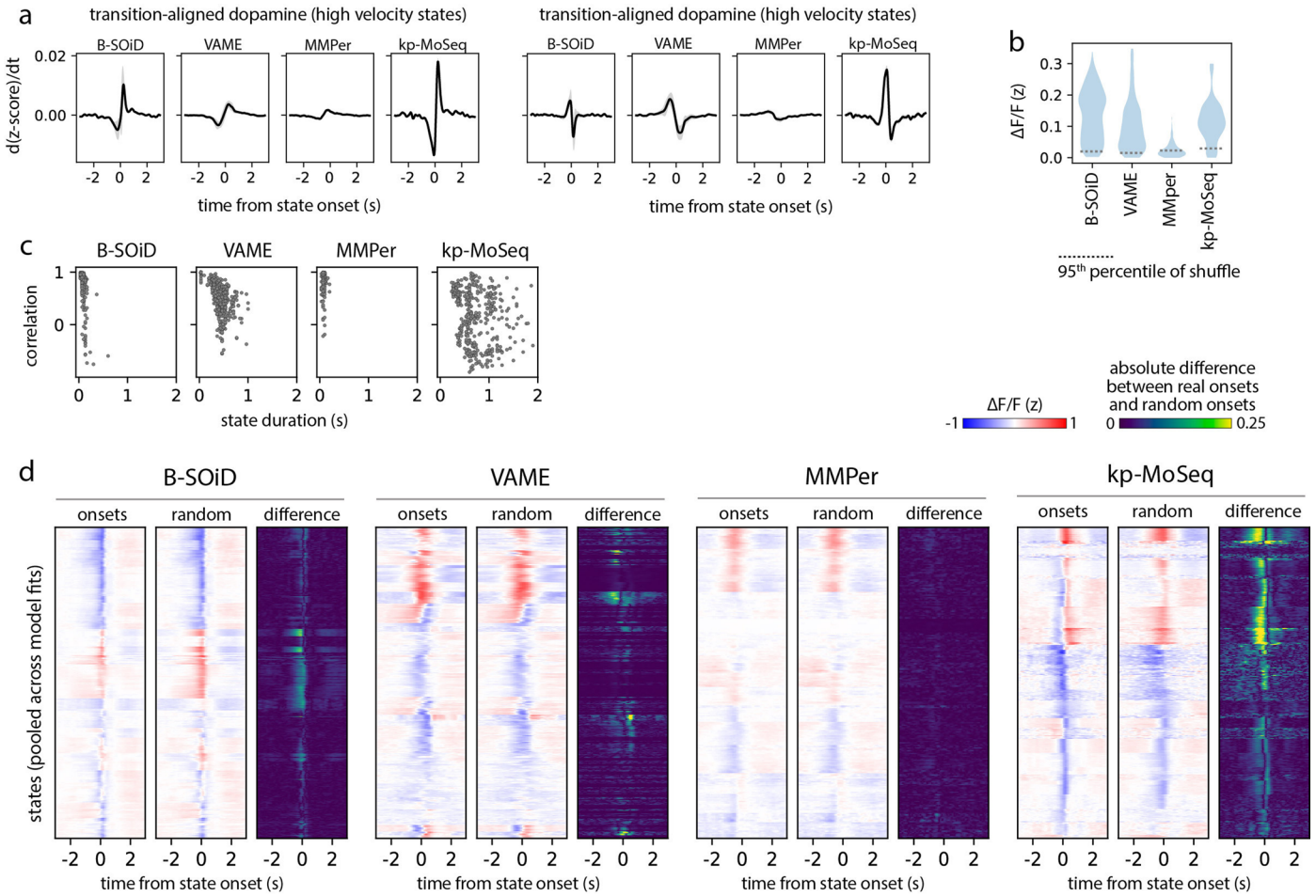
Extended Data Fig. 5 | Method-to-method differences in sensitivity to behavioral changepoints are robust to parameter settings. a) Output of unsupervised behavior segmentation algorithms across a range of parameter settings, applied to 2D keypoint data from two different camera angles (N=1 model fits per parameter set). 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 (see Supplementary Table 3 for a summary of parameters). **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.

https://doi.org/10.1038/s41592-024-02318-2



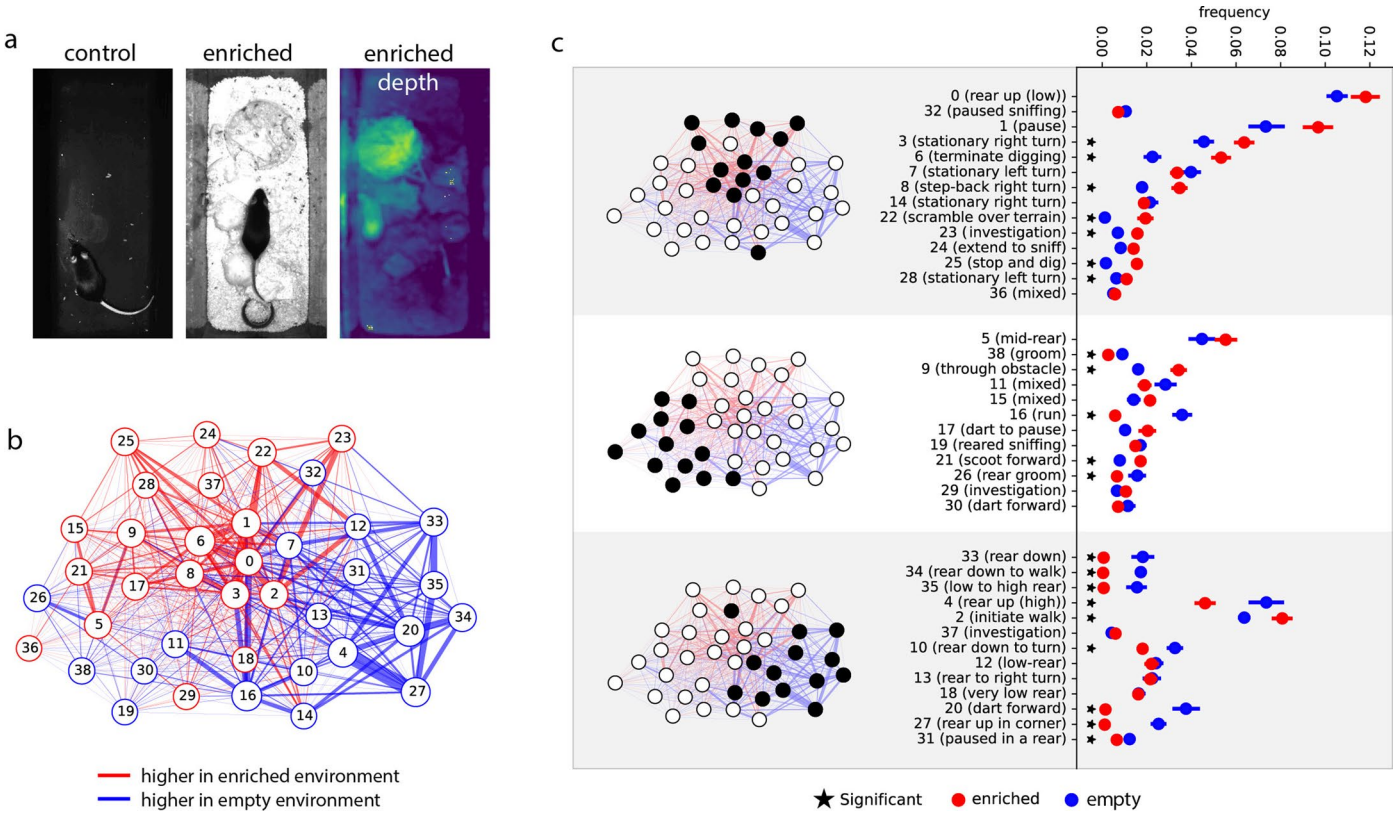
Extended Data Fig. 6 | Accelerometry reveals kinematic transitions at the onsets of keypoint-MoSeq states. a) IMU signals aligned to state onsets from several behavior segmentation methods. Each row corresponds to a behavior state and shows the average across all onset times for that state. A single model fit is shown for each method.

https://doi.org/10.1038/s41592-024-02318-2



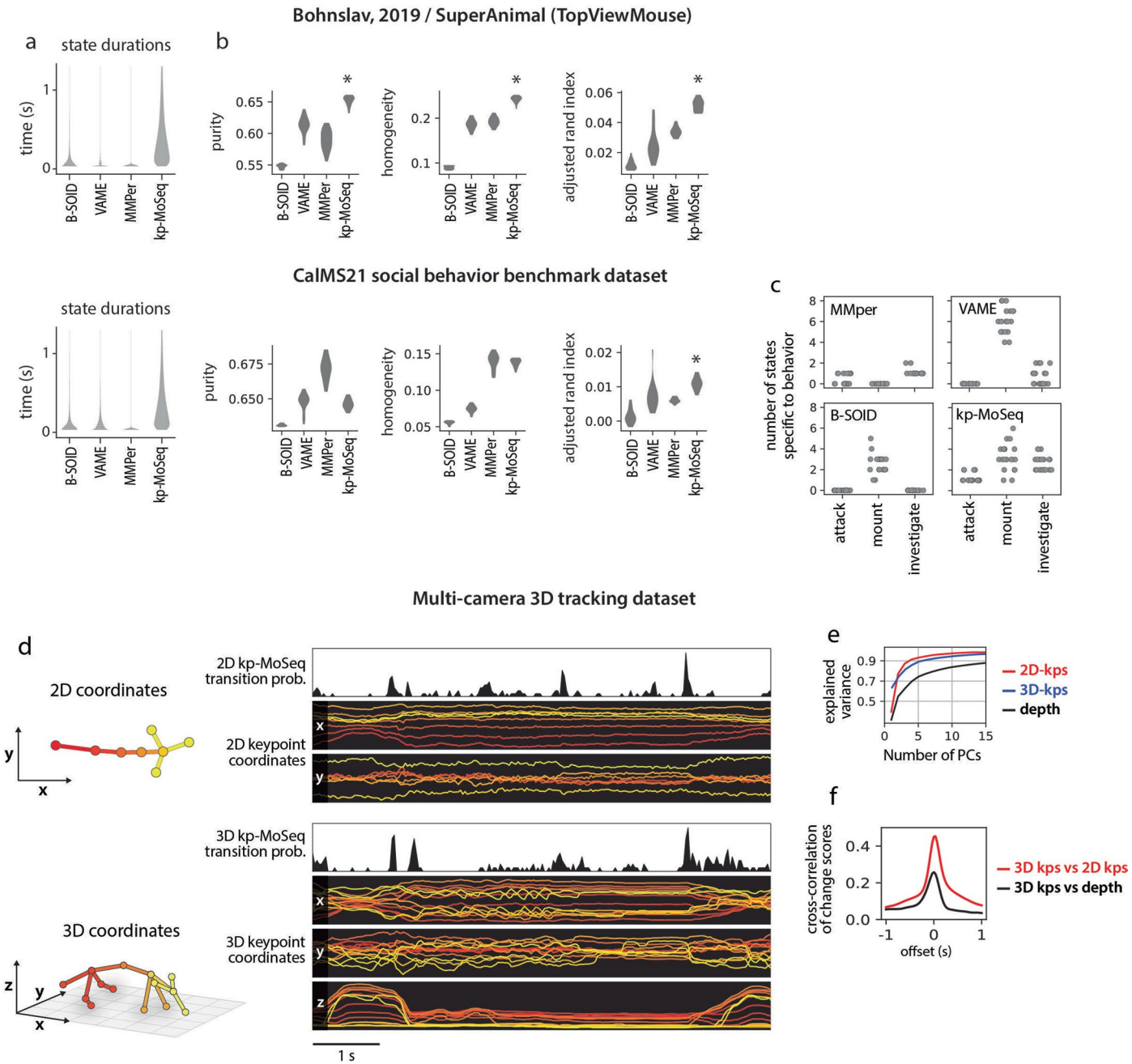
Extended Data Fig. 7 | Striatal dopamine fluctuations are enriched at keypoint-MoSeq syllable onsets. a) Derivative of the dopamine signal aligned to the onsets of high velocity or low velocity behavior states. States from each method were classified evenly as high or low velocity based on the mean centroid velocity during their respective frames. Plots show mean and inter-95% range across N=20 model fits. **b)** Distributions capturing the average absolute value of the dopamine signal across states from each method. **c)** Relationship between state durations and correlations from Fig. 5f. **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.

https://doi.org/10.1038/s41592-024-02318-2



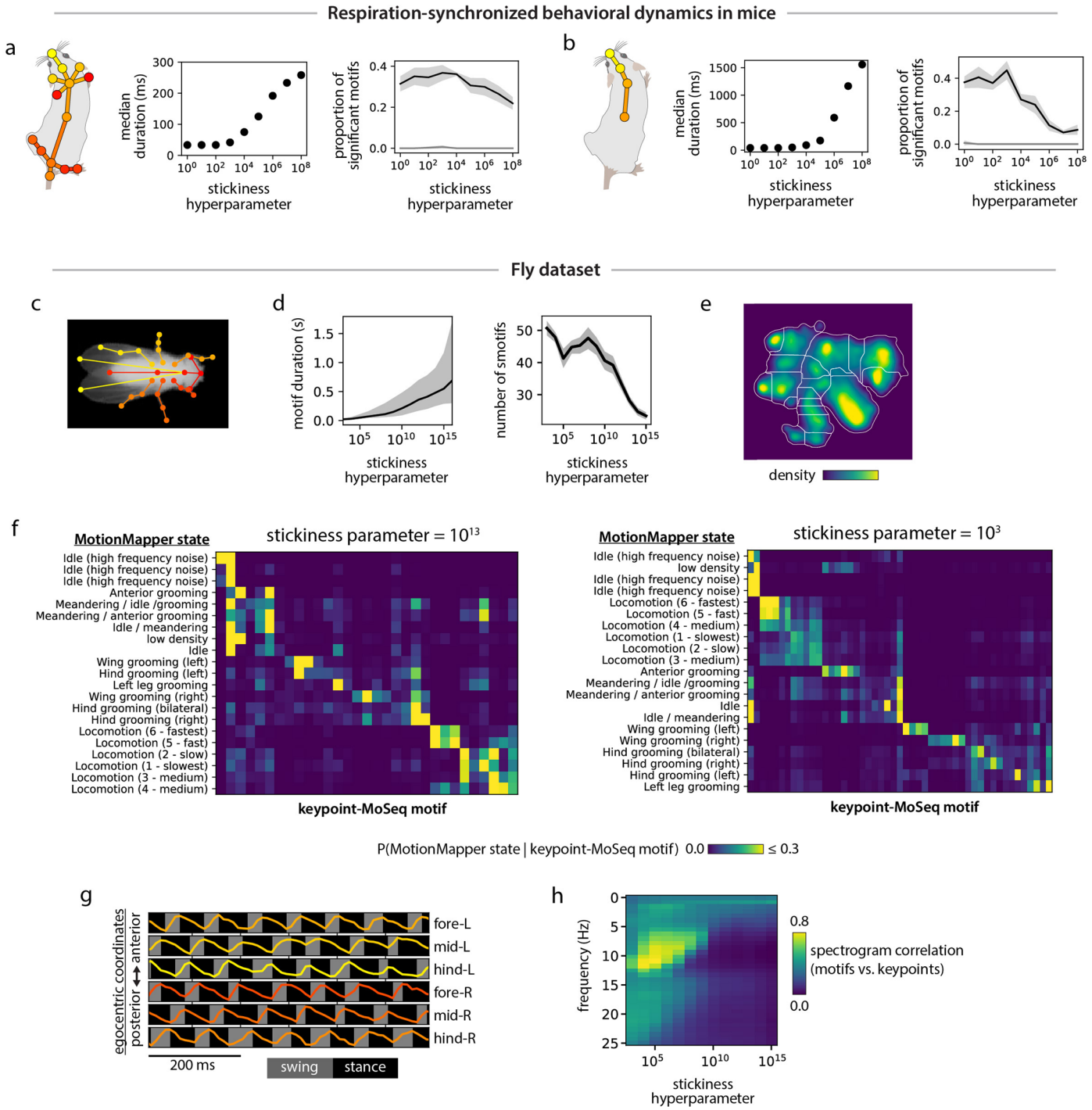
Extended Data Fig. 8 | Changes in behavior caused by environmental enrichment. a) Example frames from conventional 2D videos of the empty bin (left), and enriched environment (middle), as well as depth video of the enriched environment (right). **b)** Graph showing changes in syllable-to-syllable transition statistics across environments. Edge color and width indicate the sign and magnitude of change in the frequency of each syllable pair. **c) Right:** changes in syllable frequency across environments, with stars indicating significant differences ($P < 0.05$, N=16, Mann-Whitney U test). Error bars show standard error of the mean. **Left:** Syllable groupings defined by clustering of the transition graph shown in (b).

https://doi.org/10.1038/s41592-024-02318-2



Extended Data Fig. 9 | Supervised behavior benchmark. a) Distribution of state durations from each behavior segmentation method for the open field benchmark (top) and the CalMS21 social behavior benchmark (bottom). **b)** Three different similarity measures applied to the output of each unsupervised behavior analysis method, showing the median (gray bars) and inter-quartile interval (black lines) across independent model fits (N=20; * $P < 10^{-5}$, for keypoint-MoSeq vs. each other method, Mann-Whitney U test). **c)** 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. **d) Left:** Keypoints tracked in 2D (top) or 3D (bottom) and corresponding egocentric coordinate axes. Right: example keypoint trajectories and transition probabilities from keypoint-MoSeq. Transition probability is defined, for each frame, as the probability of a syllable transition occurring on that frame. **e)** Cumulative 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. **f)** Cross-correlation between the 3D keypoint change score and change scores derived from 2D keypoints and depth respectively (based on N=20 model fits).

https://doi.org/10.1038/s41592-024-02318-2



Extended Data Fig. 10 | Keypoint-MoSeq identifies behavioral motifs across timescales. a-b) Alignment of mouse behavior motifs to respiration. Figure created with SciDraw under a CC BY 4.0 license. **a**) Left: Keypoints used for model fitting. Middle: Median motif durations for models fit with a range of stickiness hyperparameters. Right: Proportion of significantly respiration-aligned motifs, stratified by stickiness hyperparameter, showing mean and standard deviation across N=5 model fits. **b**) As (a), but restricted to upper spine, neck, head, and nose keypoints. **c-h**) Keypoint-MoSeq partitions fly behavior across timescales. **c**) Fly keypoints used for fitting keypoint-MoSeq and MotionMapper. **d**) Motif durations (left) and number of motifs (right) for models trained with a range of target timescales. Ten separate models were fit for each timescale. For motif durations, we pooled the duration distributions from all 20 models and plotted the median duration in black and interquartile range in gray. For motif number, we counted the number of motifs with frequency above 0.5% for each of the 20 models and plotted the mean of this count in black and the standard deviation in gray. **e**) Density of points in 2D 'behavior space' generated by MotionMapper. Each white-line delimited region corresponds to a MotionMapper state label. **f**) Confusion matrices showing the frequency of each MotionMapper state during each keypoint-MoSeq motif. **g**) Example of swing and stance annotations over a 600ms window. Lines show the egocentric coordinate of each leg tip (anterior-posterior axis only). Gray shading denotes the swing phase, defined as the interval posterior-to-anterior limb motion. **h**) Cross-correlation between the spectrograms of keypoints and motif labels respectively. Heatmap rows correspond to frequency bands of the spectrograms and columns correspond to models with different target timescales.

nature portfolio

Corresponding author(s): Sandeep Robert Datta
Scott W. Linderman

Last updated by author(s): Feb 14, 2024

Reporting Summary

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

Statistics

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

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

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

Software and code

Policy information about availability of computer code

Data collection	Custom code for data acquisition and modeling was written in Python (details in Methods) and is available on github. Acquisition code: https://github.com/calebweinreb/top-bottom-moseq (unversioned), https://github.com/calebweinreb/multicamera_acquisition (version 0.1.0) Modeling code: https://github.com/dattalab/jax-moseq and https://github.com/dattalab/keypoint-moseq (version 0.0.0)
Data analysis	The following custom and publicly-available software packages were used, details are included in Methods section: Python (version 3.8), NumPy (version 1.24.3), Scikit-learn (version 1.2.2), PyTorch (version 1.9), Jax (version 0.3.22), SciPy (version 1.10.1), Matplotlib (version 3.7.1), Statsmodels (version 0.13.5), Motionmapperpy (version 1.0), DeepLabCut (version 2.2.1), SLEAP (version 1.2.3), BSOID (version 1.5.1), VAME (version 1.1), GIMBAL (version 0.0.1), HRNet (unversioned), LightningPose (version 0.0.4), segmentation_models_pytorch (version 0.3.3)

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

Data

Policy information about availability of data

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

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

This study used the following publicly available datasets: CalMS2133 (https://data.caltech.edu/records/s0vdx-0k302); DeepEthogram benchmark data32 (https://github.com/jbohnslav/deepethogram); Rat7M (https://figshare.com/collections/Rat_7M/5295370); Fly keypoint tracking (https://doi.org:10.1038/s41592-018-0234-5). Other data raw data generated in this study have been deposited on Zenodo (https://zenodo.org/records/10636983). The thermistor recordings are available from the authors upon reasonable request.

Human research participants

Policy information about studies involving human research participants and Sex and Gender in Research.

Reporting on sex and gender	*Use the terms sex (biological attribute) and gender (shaped by social and cultural circumstances) carefully in order to avoid confusing both terms. Indicate if findings apply to only one sex or gender; describe whether sex and gender were considered in study design whether sex and/or gender was determined based on self-reporting or assigned and methods used. Provide in the source data disaggregated sex and gender data where this information has been collected, and consent has been obtained for sharing of individual-level data; provide overall numbers in this Reporting Summary. Please state if this information has not been collected. Report sex- and gender-based analyses where performed, justify reasons for lack of sex- and gender-based analysis.*
Population characteristics	*Describe the covariate-relevant population characteristics of the human research participants (e.g. age, genotypic information, past and current diagnosis and treatment categories). If you filled out the behavioural & social sciences study design questions and have nothing to add here, write "See above."*
Recruitment	*Describe how participants were recruited. Outline any potential self-selection bias or other biases that may be present and how these are likely to impact results.*
Ethics oversight	*Identify the organization(s) that approved the study protocol.*

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

Field-specific reporting

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

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

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

Life sciences study design

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

Sample size	Sample sizes were not pre-determined. The number of samples (usually N=20 model runs) was chosen based on computational tractability. In practice, this sample size was easily large enough to reveal statistical significance, given the effect sizes demonstrated in the study.
Data exclusions	Recording sessions were excluded when the mice did move for the majority of the recording, i.e. of they remained in the same quadrant of the arena for >80% of frames. Thermistor recordings were excluded when the thermistor signal was excessively noisy.
Replication	Every model was run multiple times with different random seeds. No model runs were excluded.
Randomization	There were no treatment groups in our study, thus randomization was not required.
Blinding	Analysis of results did not depend on human observers, thus blinding was not necessary.

Reporting for specific materials, systems and methods

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

Materials & experimental systems

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

Methods

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

Animals and other research organisms

Policy information about studies involving animals; ARRIVE guidelines recommended for reporting animal research, and Sex and Gender in Research

Laboratory animals	This study utilized wild-type C57 mice, both male and female, between 6-15 weeks of age, housed in reverse 12 hr dark/light cycle at approximately 50% humidity and 21 degrees C.
Wild animals	This study did not involve wild animals.
Reporting on sex	The findings do not apply to only one sex.
Field-collected samples	This study did not involve samples collected from the field.
Ethics oversight	All experimental procedures were approved by the Harvard Medical School IACUC (Protocol # 04930) and were performed in compliance with ethical regulations of Harvard University and the Guide fir Animal Care and Use of Laboratory Animals.

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