Anatomically-based skeleton kinetics and pose estimation in freely-moving rodents

Arne Monsees[1], Kay-Michael Voit[1], Damian J. Wallace[1], Juergen Sawinski[1], Edyta Leks[2,3], Klaus Scheffler[2,3], Jakob H. Macke[4,5,6] & Jason N. D. Kerr[1,6]

[1] Department of Behavior and Brain Organization, Research Center caesar, Bonn, Germany.

[2] High-Field MR Center, Max Planck Institute for Biological Cybernetics, Tübingen, Germany.

[3] Department for Biomedical Magnetic Resonance, Eberhard Karls University of Tübingen, Tübingen, Germany.

[4] Machine Learning in Science, Eberhard Karls University of Tübingen, Tübingen, Germany.

[5] Empirical Inference, Max Planck Institute for Intelligent Systems, Tübingen, Germany.

[6] denotes equal last author

Correspondence to:

Jason N. D. Kerr

e-mail: jason.kerr@caesar.de, arne.monsees@caesar.de

Abstract

Forming a complete picture of the relationship between neural activity and body kinetics requires quantification of skeletal joint biomechanics during behavior. However, without detailed knowledge of the underlying skeletal motion, inferring joint kinetics from surface tracking approaches is difficult, especially for animals where the relationship between surface anatomy and skeleton changes during motion. Here we developed a videography-based method enabling detailed three-dimensional kinetic quantification of an anatomically defined skeleton in untethered freely-behaving animals. This skeleton-based model has been constrained by anatomical principles and joint motion limits and provided skeletal pose estimates for a range of rodent sizes, even when limbs were occluded. Model-inferred joint kinetics for both gait and gap-crossing behaviors were verified by direct measurement of limb placement, showing that complex decision-making behaviors can be accurately reconstructed at the level of skeletal kinetics using our anatomically constrained model.

Introduction

The relationship between neural activity patterns and body motion is complex as neuronal activity patterns are dependent on factors such as the intended behavioral outcome[1], task familiarity[2], changes in the environment but also exact limb trajectories[3-5] and motion kinetics[6]. Much of the motion kinematic data forming our view of the sensorimotor control of movement was collected during short behavioral epochs where the animal was in various forms of restraint[3-8], but how these findings relate to kinematics during free behavior, where the relationship between the environment and body motion is continuously changing, is largely unknown[9-11]. While there have been methodological advances recently enabling detailed neural population activity recordings[12-15] and surface tracking of an animal's body[16-24], a major challenge still remains for generating detailed kinetics of individual body parts, such as limbs, and how they interact with the environment during free behavior[18,25]. This poses an especially difficult problem as limb motions involving muscles, bones and joints are biomechanically complex given their three-dimensional (3D) translational and rotational co-dependencies[26,27].

More recently, advances in the development of machine learning approaches have enabled limb tracking in both freely-moving[23] and head-restrained insects[28] as the limb exoskeleton not only provides joint angle limits and hard limits of limb position, but can be tracked as a surface feature during behavior. When studying vertebrates, like rats, the entire skeleton is occluded by the animal's fur and inferring bone positions and calculating joint kinetics becomes more complicated since the spatial relationship between skeleton and overlying soft tissues are less apparent[29,30]. Despite this limitation, recent approaches have extended two-dimensional surface tracking methods[21,23,24] to include 3D pose reconstructions[31] using a multi-camera cross-validation approach and hand-marked ground-truth data sets[19] allowing general kinematic representation of animal behaviors and poses for multiple species[32]. Extending these approaches to obtain the skeleton kinetics relies on knowledge of the skeleton anatomy and biomechanics as well as motion restrictions of joints[27] because animal poses are limited by both bone lengths and joint angle limits. Here, we developed an anatomically constrained skeleton model incorporating mechanistic knowledge of bone locations, anatomical limits of bone rotations, and temporal constraints to track 3D joint positions and their kinetics in freely moving rats. Together the fully constrained skeleton enabled the reconstruction of skeleton poses and kinetic quantification during gap-crossing tasks and throughout spontaneous behavioral sequences.

Results

Constraining the model

Here we tracked 3D joint positions and their kinetics in freely moving rats (Fig. 1a) over a large size range (N = 6 animals, average weight: 321 g, range: 71-735 g) using videography and a anatomically constrained skeleton model (ACM) incorporating mechanistic knowledge of bone locations, anatomical limits of bone rotations, and temporal constraints. Together, both the temporal and anatomical constraints, i.e. the fully constrained ACM, enabled the reconstruction of skeleton poses of behaving animals with single joint precision (Fig. 1b) as well as smooth limb and joint transitions during gait cycles (Fig. 1c) allowing the quantification joint kinetics and spatial positions of the limbs throughout behavioral sequences. At the core of this approach was a generalized rat skeleton based on rat bone anatomy[33] (Fig. 1d) modeled as a mathematical graph with vertices representing individual joints and edges representing bones (Fig. 1c, see methods for details and Supplementary Fig. 1). For example, a single edge was used to represent the animal’s head, the spinal column was approximated using four edges based on cervical, thoracic and lumbar sections of the column as well as the sacrum[33] and the tail by five edges (Fig. 1d, Supplementary Fig. 1). To constrain the model we applied angle limits for each joint based on measured rotations[34] (Fig. 1e, see Methods “Constraining poses based on joint angle limits”) as well as anatomical constraints based on measured relationships between bone lengths and animal weight and size[35] (see Methods “Constraining bone lengths based on allometry”). Finally, as vertebrates are symmetrical around the mid-sagittal plane we applied a further anatomical constraint to ensure symmetry for bone lengths and surface marker locations (Supplementary Fig. 3). Together, this approach established a unique skeleton model for each animal. To generate probabilistic estimates of 3D joint positions and provide temporal constraints, we implemented a temporal unscented Rauch-Tung-Striebel (RTS) smoother[36,37], an extension of a Kalman filter[38], which is suitable for nonlinear dynamics models and also incorporates information from future marker locations (see Supplementary Methods “Probabilistic pose estimation” for details). Parameters of the smoother were learned via the expectation-maximization (EM) algorithm[39], by iteratively fitting poses of the entire behavioral sequence (see Methods “Performing probabilistic pose reconstruction”).

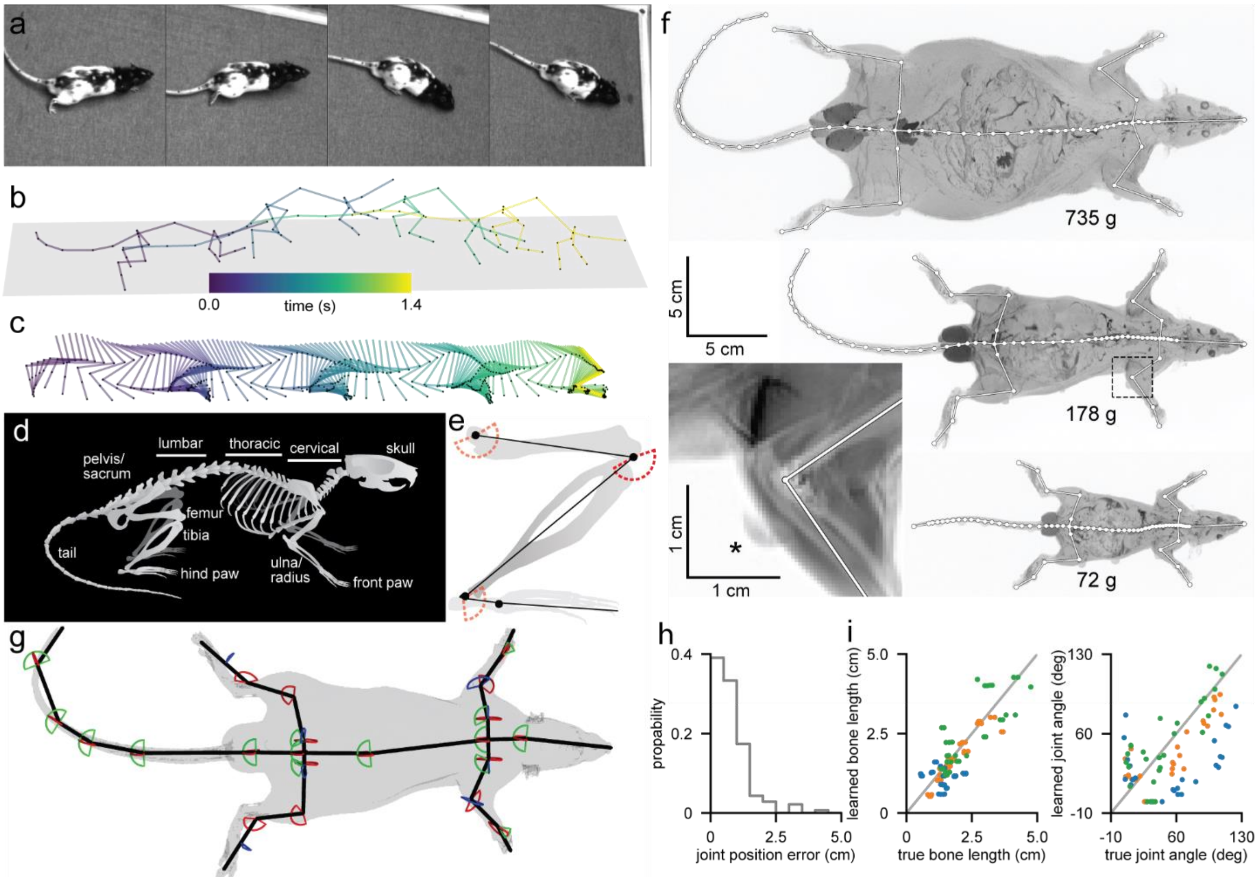


Fig. 1. Leaning anatomically informed skeleton models allows for accurate 3D pose reconstruction during free behavior. a, Four recorded frames from an overhead camera showing a freely moving rat with painted surface labels. **b**, Reconstructed animal poses of the entire skeleton during gait. Poses correspond to the images shown in a. **c**, Enlargement of the reconstructed right hind limb during the sequence shown in b. **d**, Schematic image of a rat skeleton showing anatomical landmarks. **e**, Schematic image of a hind limb with modeled bones (black lines) and joints (black dots) as well as enforced joint angle limits for flexion/extension (red dashed lines). **f**, MRI scans of three differently sized animals (maximum projection) and an enlargement of a right elbow joint (lower left, mean projection, same area as in left dashed box) with manually labeled bone (white lines) and joint (white dots) positions. Note visible MRI surface marker (asterisk). **g**, 3D representation of a rat's MRI scan showing the animal's surface (gray) and the aligned skeleton model (black lines) and joint angle limits for flexion/extension (red lines), abduction/adduction (green lines) and internal/external rotation (blue lines). **h**, Probability histogram of the joint position error. **i**, Learned bone lengths (left) and joint angles (right) compared to MRI bone lengths and joint angles (N = 6 animals). Colors represent small (blue), medium (orange) and large (green) animal sizes (blue: 71 g & 72 g, orange: 174 g & 178 g, green: 699 g & 735 g).

Learning the skeleton

To relate the animal's surface to the underlying skeleton we used a grid of rationally-placed surface markers, which were either distinct anatomical landmarks like the snout or were painted on the animal's fur (Fig 1a, 14 landmarks and 29 spots total per animal, Supplementary Fig. 2). To individually tailor the skeleton to each animal we used gradient decent optimization to learn varying bone lengths and surface marker locations for each animal (see Methods "Learning bone lengths and surface marker positions"). Visible surface markers were manually annotated from a fraction of all recorded images to learn the skeleton that could then be used for all behavioral data acquired from that animal. As the rigid spatial relationship between surface markers and the underlying joints remained constant the algorithm could learn individual bone lengths as well as surface marker locations by adjusting both via gradient descent optimization (Supplementary Fig. 3). New poses were iteratively generated for each time point by applying a global translation to the generalized skeleton model and subsequently modifying positions of joints and rigidly attached surface markers by rotating each bone (Supplementary Video 1). Errors were established and minimized by projecting inferred 3D surface marker locations onto calibrated overhead camera sensors and subsequently comparing them to manually labeled ground-truth data (Supplementary Fig. 4).

To evaluate the accuracy of both the skeleton model and inference of joint positions over a large range of animal sizes, we obtained high-resolution MRI scans for each animal (Fig. 1f, N = 6 animals, see methods "Comparison of skeleton parameters with MRI data") and aligned the skeleton model to measured positions of 3D surface markers (Fig. 1g). Errors for inferred spine and limb joint positions were low (Fig. 1h, 138 joint positions total, joint position error: 0.79 +/- 0.69 & 0.65 cm [mean +/- s.d. & median]) and inferred limb bone lengths and bone angles were not significantly different from those measured in MRI scans (Fig. 1i, 108 bone lengths total, range of measured bone lengths: 0.53 cm to 4.76 cm, bone length error: 0.46 +/- 0.34 & 0.36 cm [mean +/- s.d. & median], Spearman correlation coefficient: 0.75, two-tailed p-value testing non-correlation: 5.00×10^{-21}; 84 bone angles total, range of measured bone angles: 4.13° to 123.77°, bone angle error: 27.80 +/- 18.98 & 26.72° [mean +/- s.d. & median], Spearman correlation coefficient: 0.47, two-tailed p-value testing non-correlation: 5.29×10^{-6}). Together this demonstrated first, that the anatomically constrained skeleton model generated by our algorithm was highly accurate when compared with the animal's actual skeleton across the range of animal sizes, and second, that accurate joint positions could be reconstructed in a single static pose from this approach.

Accurate behavior reconstructions required both the temporal and anatomical constraints. To reconstruct behavioral sequences using the ACM, we first tracked 2D surface marker locations in the recorded movies using DeepLabCut[24], which is specifically designed for surface landmark detection of laboratory animals. As the ACM contained both joint angle limits and temporal constraints, we evaluated the role of these by reconstructing poses without either the joint angle limits or the temporal constraints. The resulting temporal model, only temporally constrained, and the joint angle model, only constrained by joint angle limits, and the naïve skeleton model, constrained by neither, were compared to the ACM. To measure animal paw positions and orientations during gait we used a modified frustrated total internal reflection (FTIR) touch sensing approach[40,41] (Fig. 2a-c, Supplementary Video 2) and compared these measurements to the paw positions and orientations inferred by each model (Fig. 2d,e, N = 6 animals, 29 sequences, 181.25 s per 145000 frames total from 4 cameras). The ACM produced significantly smaller positional errors compared to all other models (Fig. 2g, left; 10410 positions total; p-values of one-sided Kolmogorov-Smirnov test: ACM vs. joint angle model: $9.84\text{x}10^{-21}$; ACM vs temporal model: $4.38\text{x}10^{-35}$; ACM vs. naïve skeleton model: $9.03\text{x}10^{-37}$), whereas orientation errors were only significantly smaller when comparing the ACM to the temporal and naïve skeleton model (Fig. 2g, right; 7203 and 6969 orientations total for the ACM/anatomical model and the temporal/naïve skeleton model respectively; p-values of one-sided Kolmogorov-Smirnov test: ACM vs temporal model: $3.20\text{x}10^{-39}$; ACM vs. naïve skeleton model: $2.51\text{x}10^{-50}$). While orientation errors were significantly reduced by the anatomical constraints, including temporal constrains limited abrupt pose changes over time compared to either the naïve skeleton model or joint angle model (Fig. 2f, Supplementary Video 3-8). As a result, ACM-generated joint velocities and accelerations (Fig. 2h, 576288 velocities and accelerations total) were significantly smaller when compared to all other models (p-values of one-sided Kolmogorov-Smirnov test: ACM vs. joint angle model (velocity): numerically 0; ACM vs. temporal model (velocity): numerically 0; ACM vs. naïve skeleton model (velocity): numerically 0; ACM vs. joint angle model (acceleration): numerically 0; ACM vs. temporal model (acceleration): numerically $3.71\text{x}10^{-90}$; ACM vs. naïve skeleton model (acceleration): numerically 0). The temporal and anatomical constraints each had an advantage over the naïve skeleton model, and both constraints applied simultaneously improved positional accuracy as well as motion trajectories and prevented anatomically infeasible bone orientations and abrupt paw relocations. Moreover, the fraction of position errors exceeding 4 cm increased when constraints were not considered (fraction of errors exceeding 4 cm: ACM: 2.72%; joint angle model: 3.64%; temporal model: 4.42%; naïve skeleton model: 6.44%), and the same was observed for orientation errors exceeding 60° (fraction of errors exceeding 60°: ACM: 7.78%; joint

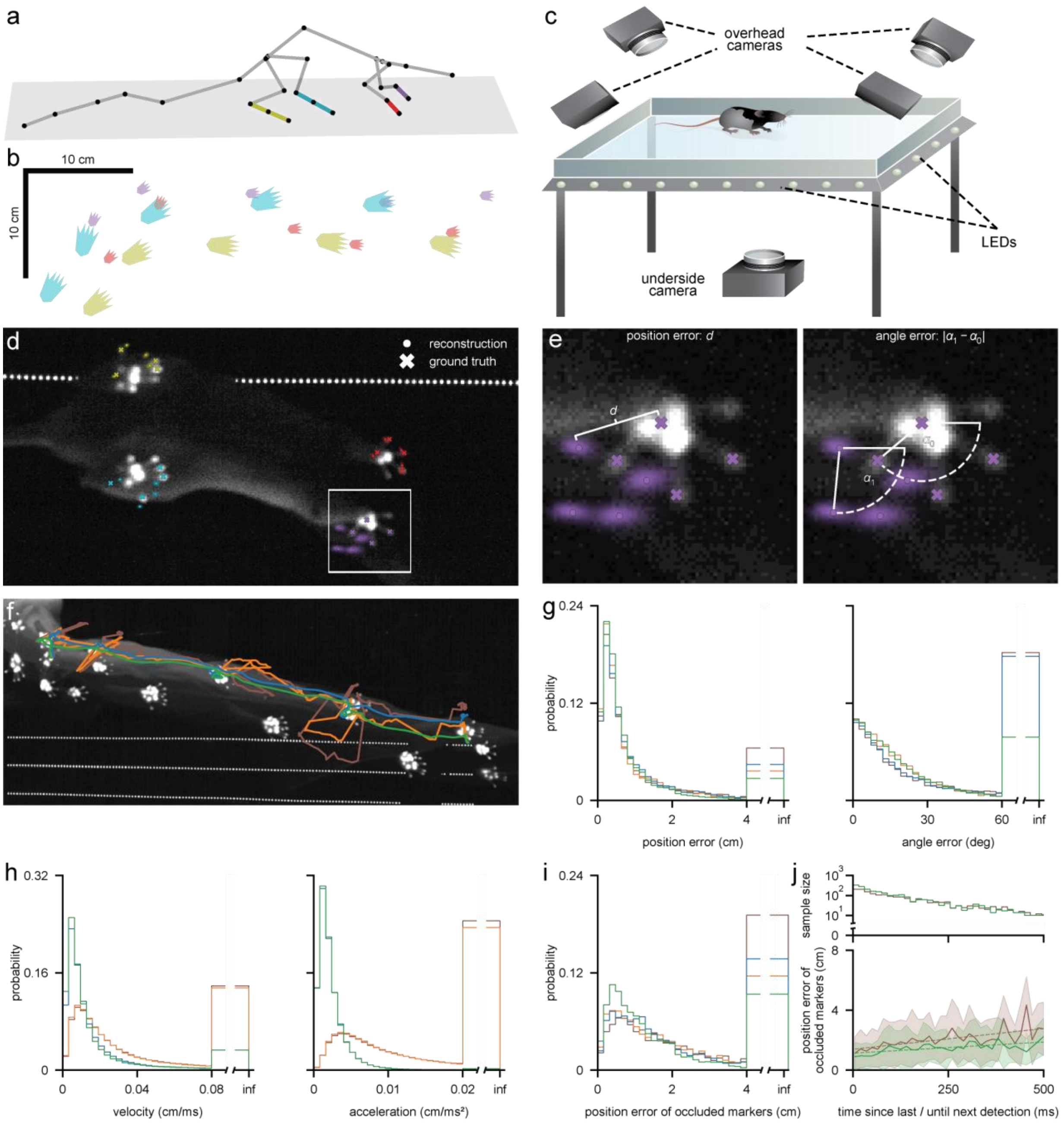


Fig. 2. Comparison between inferred and measured paw positions during free behavior. **a**, Reconstructed animal pose based on a learned skeleton model with highlighted left front (purple), right front (red), left hind (cyan) and right hind paw (yellow). **b**, Reconstructed xy-positions of the paws during gait. Colors as in a. **c**, Schematic image of the FTIR touch sensing setup with one underneath and four overhead cameras. **d**, Single image from the underneath camera with reconstructed (x) and ground truth (filled circle) xy-positions of the paw's centers and fingers/toes for the all four paws. Colors as in a. Large point clouds around landmark locations indicate high uncertainty. **e**, Enlarged view of the left front paw in d (white box) showing calculation of position error (left) and the angle error (right). **f**, Maximum intensity projection from

the underneath camera of a 2.5 s long sequence with trajectories for the reconstructed xy-positions of the right hind paw using the ACM (green), temporal- (blue), joint angle- (orange) and naïve skeleton (brown) models. **g**, Probability histograms for paw position (left) and angle errors (right) comparing different model constraint regimes. Color-coding as in f. **h**, Probability histograms for paw velocities (left) and accelerations (right) comparing different model constraint regimes. Color-coding as in f. **i**, Probability histograms for paw position errors whereas only undetected surface markers are used for the calculation comparing different model constraint regimes. Color-coding as in f. **j**, Position errors of occluded markers (bottom) and corresponding binned sample sizes (top) as a function of time since last / until next marker detection comparing different model constraint regimes. Color-coding as in f. Sample sizes differ depending on whether reconstructed poses were obtained via the unscented RTS smoother (green) or not (brown).

angle model: 7.81%; temporal model: 17.77%; naïve skeleton model: 18.22%). Likewise, enforcing constraints also lowered the percentage of velocities exceeding 0.08 cm/ms (fraction of errors exceeding 0.08 cm/ms: ACM: 3.29%; joint angle model: 13.49%; temporal model: 3.28%; naïve skeleton model: 13.85%) and accelerations exceeding 0.02 cm/ms^2 (fraction of errors exceeding 0.02 cm/ms^2: ACM: 0.22%; joint angle model: 23.43%; temporal model: 0.25%; naïve skeleton model: 24.55%). To test ACM robustness to missing surface markers, position errors were calculated for inferred paw positions from data in which surface markers were undetected (Fig. 2i, 2797 position errors total). Compared to all other models the ACM produced significantly lower errors (p-values of one-sided Kolmogorov-Smirnov test: ACM vs. joint angle model: 9.67x10^{-23}; ACM vs temporal model: 2.83x10^{-22}; ACM vs. naïve skeleton model: 3.91x10^{-47}) as well as the smallest number of error values above 4 cm (ACM: 9.36%; joint angle model: 11.61%; temporal model: 13.72%; naïve skeleton model: 19.12%). Paw placement errors increased the longer a surface marker remained undetected for the ACM and the naïve skeleton model (Fig. 2j, linear regression: ACM: slope: 1.49 cm/s, intercept: 1.13 cm; naïve skeleton model: slope: 2.77 cm/s, intercept: 1.39 cm) and errors were significantly lower when comparing both models (p-values of one-sided Mann-Whitney rank test: ACM vs. naïve skeleton model: 3.91x10^{-47}).

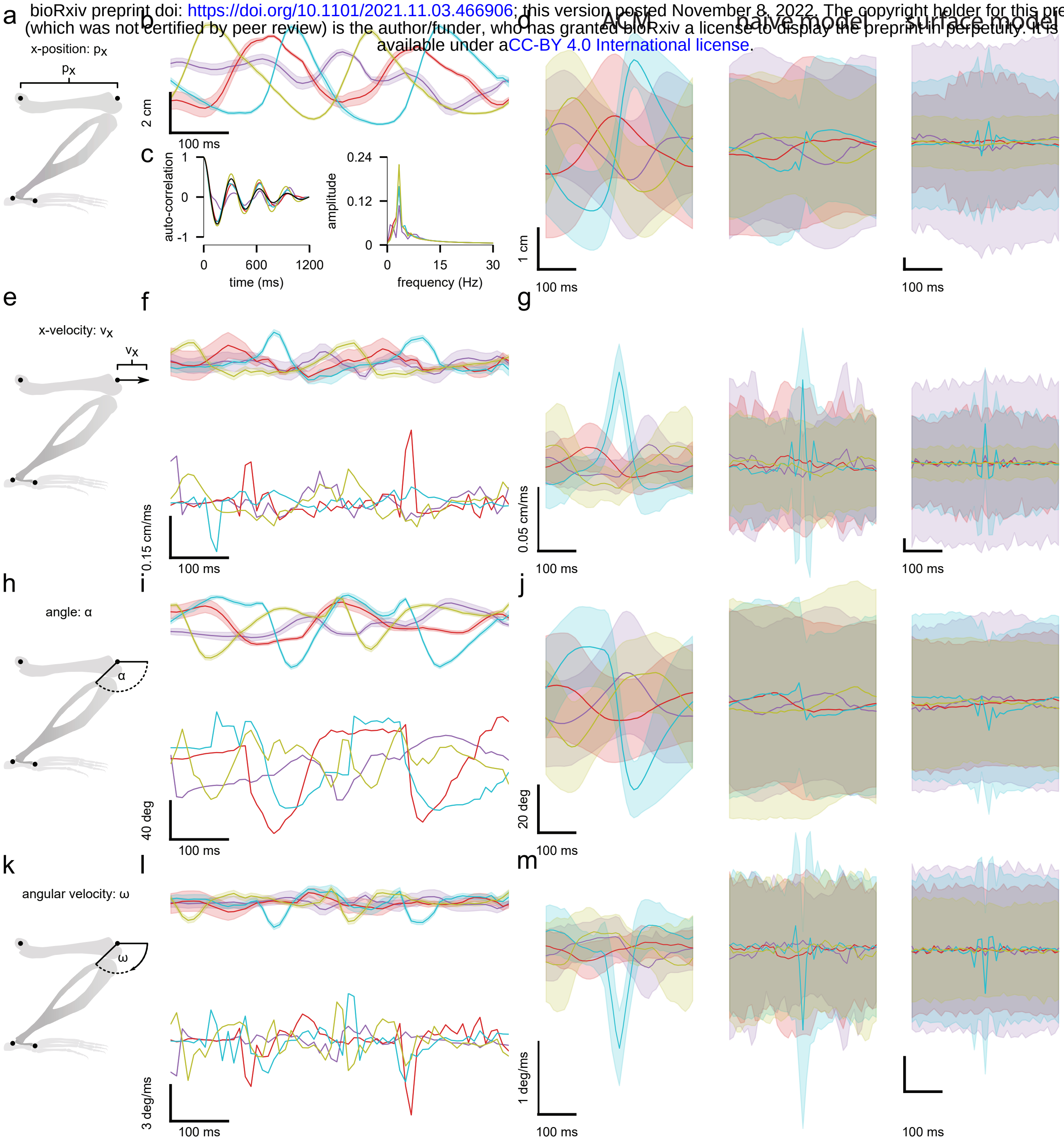

a
x-position: p_x
p_x
b
2 cm
100 ms
c
auto-correlation
1
0
-1
0
600
1200
time (ms)
amplitude
0.24
0.12
0
0
15
30
frequency (Hz)
d
ACM
naive model
surface model
1 cm
100 ms
100 ms
e
x-velocity: v_x
v_x
f
0.15 cm/ms
100 ms
g
0.05 cm/ms
100 ms
100 ms
h
angle: α
α
i
40 deg
100 ms
j
20 deg
100 ms
k
angular velocity: ω
ω
l
3 deg/ms
100 ms
m
1 deg/ms
100 ms
100 ms

Fig. 3. Influence of temporal and anatomical constraints on periodic gait cycles. a, Schematic of the normalized x-position for a single joint. **b**, Trajectories of the normalized x-position as a function of time for the left wrist (purple), right wrist (red), left ankle (cyan) and right ankle (yellow) joint during gait. **c**, Auto-correlations of the normalized x-position as a function of time (left) for four different limbs as well as a corresponding model fit via a damped sinusoid (black). Fourier transformed auto-correlations of all limbs (right) have their maximum peak at the same frequency. Colors as in b. **d**, Population averaged trajectories of the normalized x-position as a function of time for the ACM (left), the naïve skeleton model (center) and the surface model (right). Colors as in b. Trajectories of the ACM and the naïve skeleton model correspond to the 3D joint locations, whereas trajectories of the surface model correspond to the 3D locations of the associated surface markers. **e**, Schematic of the of the first temporal derivative of the normalized x-position (i.e. normalized x-velocity) for a single joint. **f**, Normalized x-velocity as a function of time for the ACM (top) and the naïve skeleton model (bottom) during gait (colors as in b). **g**, Population averaged trajectories of the normalized x-velocity as a function of time for the ACM (left), the naïve skeleton model (center) and the surface model (right). Colors as in b. Trajectories as in d. **h**, Schematic of the normalized bone angle for a single joint. **i**, Bone angle as a function of time for the ACM (top) and the naïve skeleton model (bottom) during gait (colors as in b). **j**, Population averaged trajectories of the bone angle as a function of time for the ACM (left), the naïve skeleton model (center) and the surface model (right). Colors as in b. Trajectories as in d. **k**, Schematic of the first temporal derivative of the bone angle (angular velocity) for a single joint. **l**, Angular velocity as a function of time for the ACM (top) and the naïve skeleton model (bottom) during gait (colors as in b). **m**, Population averaged trajectories of bone angular velocity as a function of time for the ACM (left), the naïve skeleton model (center) and the surface model (right). Colors as in b. Trajectories as in d.

Kinetics of cyclic gait behavior.

Smooth and periodic reconstruction of an animal's average gait cycles during walking or running is only possible with robust and accurate tracking of animal limb positions. To establish whether the ACM could generate an average gait cycle from freely moving data, we next extracted individual gait cycles from multiple behavioral sequences (Fig. 3a,b, Supplementary Fig. 5, Supplementary Video 9-11) where joint velocities exceeded 25 cm/s (left; N = 2 animals, 27 sequences, 146.5 s per 58600 frames total from 4 cameras). The ACM extracted gait cycles were stereotypical and rhythmic (Fig. 3b,c), showing clear periodicity in autocorrelations of extracted limb movement (Fig 3c, left; damped sinusoid fit: frequency: 3.14 Hz, decay rate: 2.49 Hz, R^2-value: 0.90) and a common peak for all limbs in Fourier transformed data (Fig. 3c, right; max. peak at 3.33 Hz, sampling rate: 0.83 Hz). Averaged ACM extracted gait cycles (Fig. 3d, left, Supplementary Fig. 6-9) were significantly less variable than those obtained from the naïve skeleton model (Fig. 3d, center) throughout the entire gait cycle (p-value of one-sided Mann-Whitney rank test: 1.40x10^{-49}). When gait cycles were obtained from only tracking surface markers alone via a deep neural network without any form of underlying skeleton (surface model), high noise levels even made the periodic nature of the gait cycles vanish in its entirety (Fig. 3d, right). The robustness and accuracy of limb tracking was even more apparent when analyzing joint velocities (Fig. 3e-g), joint angles (Fig. 3h-j), and joint angular velocities (Fig. 3k-m), as traces generated without the ACM constraints were dominated by noise in individual examples (Fig. 3f,i,l, lower) and the cyclic nature of gait was less prominent when compared to traces obtained from the ACM (Fig. 3f,i,l, top). Consistent with this, ACM averaged traces (Fig. 3g,j,m, left, Supplementary Fig. 6-9) had significantly less variance compared to those obtained from the naïve skeleton model (Fig. 3g,j,m, right, Supplementary Fig. 6-9) for all metrics (p-values of one-sided Mann-Whitney rank test: velocity: 2.28x10^{-55}; angle: 1.42x10^{-55}; angular velocity: 1.44x10^{-5}). Additionally, for all metrics the periodicity of the gait cycles in the form of equidistant peaks was more variable for the naïve skeleton model (12 peaks total; sampling rate: 10 ms; time difference between minimum/maximum peaks: position (min. peaks): 64.16 +/- 56.78 ms; velocity (max. peaks): 80.83 +/- 54.99 ms; angle (max. peaks): 74.16 +/- 33.53 ms; angular velocity (min. peaks): 53.33 +/- 47.78 ms [avg. +/- s.d.]), when compared to the ACM (12 peaks total; sampling rate: 10 ms; time difference between minimum/maximum peaks: position (min. peaks): 75.00 +/- 29.01 ms; velocity (max. peaks): 78.33 +/- 10.67 ms; angle (max. peaks): 78.33 +/- 23.74 ms; angular velocity (min. peaks): 75.00 +/- 10.40 ms [avg. +/- s.d.]). Together this shows that the ACM can objectively extract behaviors, such as gait, from freely moving animals and quantify complex relationships between limb-bones by inferring 3D joint positions over time as well as their first derivatives.

Kinetics of complex behavior

We next used the ACM to analyze motion kinetics and segment a more complex decision-making behavior, the gap crossing task, in which the distances between two separate platforms are changed forcing the animal to re-estimate the distance to jump for each trial (Fig. 4a). Reconstructed poses during gap-estimation and jump-behaviors consisted of sequences where animals either approached or waited at the edge of the track and jumped (N = 42, Supplementary Fig. 10, Supplementary Video 12,13) or reached with a front paw to the other side of the track before jumping (N = 2, Supplementary Video 14,15). As with the inference of paw placement during gait (Fig. 2b,f), hind-paw spatial position could be inferred throughout the jump and compared to skeletal parameters during the behavior, such as the angle of the thoracolumbar joint at jump onset compared to the paw positions upon landing (Fig. 4b; 44 trials, N = 2 animals). As rats jumped stereotypically, we next tested whether the jump-related pattern of movements could be analyzed using the ACM to objectively define decision points in the behavior, such as time of jump, from each individual trial. The changes in joint angles in the spine segments and hind limbs around the time of the jump were highly consistent. Averaging these joint angles to give an averaged joint-angle trace provided a metric with a global minimum (Fig. 4c) during the jump that was independent of whether the animal crossed the gap immediately, paused and waited at the track edge or reached across the gap (Supplementary Fig. 11). This approach enabled objective identification of jump start-, mid- and end-points, from each individual jump. Traces of joint angles averaged across joints and trials (Fig. 4d) and average ACM poses (Fig. 4e) illustrate the consistency of the pose changes through the jump. We next used this to quantify relationships between joints and changes in joint kinetics throughout a jump sequence. Auto-correlations for spatial and angular limb velocities allowed quantification of the interdependency of joint movements at any point within the jumping behavior, for example at the start-point of a jump (Fig. 4f,g). This displayed relationships like a significant correlation between the spatial velocity of the right elbow and wrist joints (Fig. 4h left, Spearman correlation coefficient: 0.95, two-tailed p-value testing non-correlation: $5.40\text{x}10^{-24}$), as well as joint interactions across the midline, such as a significant correlation between spatial velocity of the right and left knee joints (Fig 4h, right, Spearman correlation coefficient: 0.93, two-tailed p-value testing non-correlation: $6.79\text{x}10^{-20}$). As the animal jumped across the gap, changes in the bone angles and their derivatives (Fig. 4i) were correlated with distance that the animal jumped (Fig. 4j). For example, angular velocity of the thoracolumbar joint and vertical velocity (z-velocity) of the thoracocervical joint were significantly correlated with jump distance 205 ms and 175 ms respectively, before the animal landed (Fig. 4k,l, Spearman correlation coefficient: -0.73, two-tailed p-value testing non-correlation: $1.13\text{x}10^{-8}$, and 0.81, two-tailed p-value testing non-correlation: $1.12\text{x}10^{-11}$).

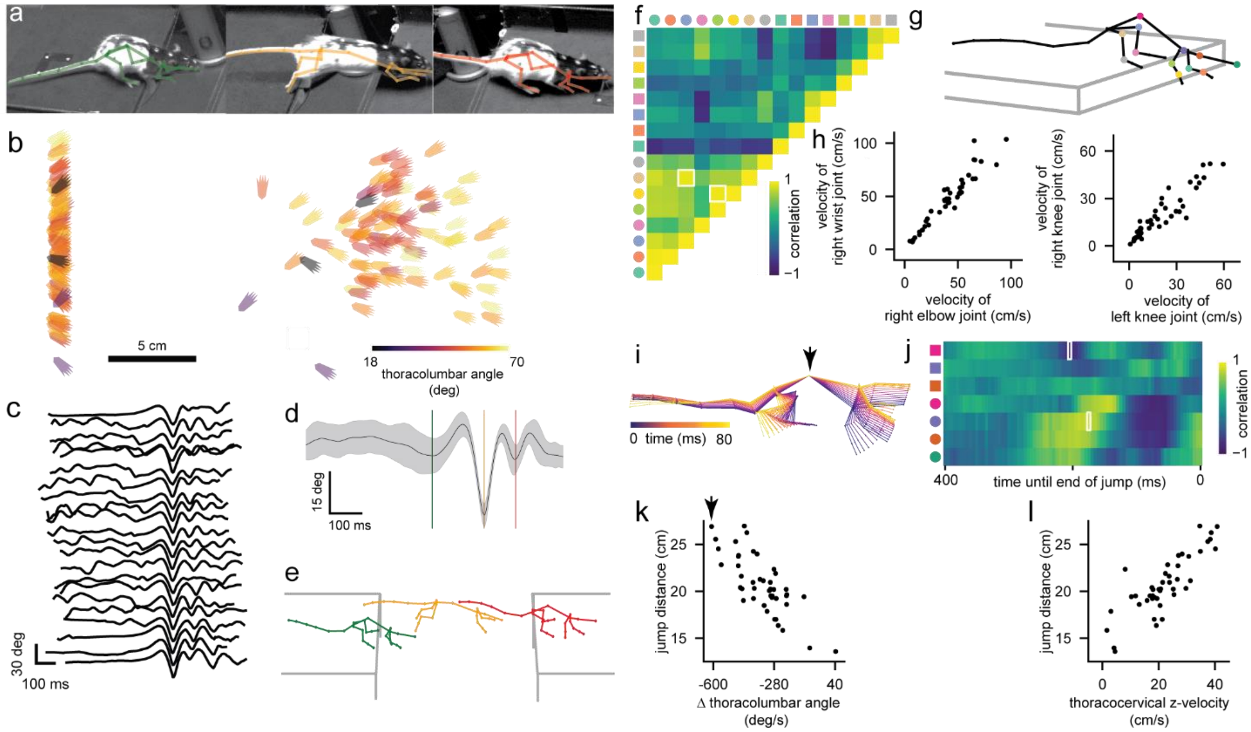


Fig. 4. 3D pose reconstruction of skeletons allows for detailed quantification of complex behavior. **a**, Images of a rat performing the gap crossing task for a trial. **b**, Reconstructed xy-positions of the hind paws at the start and end of the jump color-coded by the joint angle of the thoracolumbar joint for each gap crossing event of the population. **c**, Averaged joint-angle traces (spine and hind limb joint angles) from 22 out of 44 jump trials. **d**, Joint-angle trace averaged across joints and all jump trials. **e**, Average poses at the start- (green), mid- (orange) and end-point (red) of the jump from all jump trials. The three different time points are indicated by colored lines in d. **f**, Cross-correlation of the spatial and angular velocities of the limb joints at the start-point of a jump. Different marker shapes indicate whether rows/columns represent spatial or angular velocities (circles and squares respectively). Marker color corresponds to joint markers in g. **g**, Average pose at the start of a jump calculated from all jump trials. Joint colors are consistent with the marker colors in f and j. **h**, High correlation examples for spatial velocities of different limb joints as a function of each other for both animals. The data shown represents the correlation values highlighted in white in f. **i**, Overlaid poses of a single animal 240 ms to 160 ms before the end of a jump. Arrow indicates the thoracolumbar joint. **j**, Correlations of the z- and angular velocities of the head and spine joints for time points up to 400 ms before the end-point of a jump. Marker conventions as in f. **k**, Jump distance as a function of angular velocity of the thoracolumbar joint for both animals 205 ms before the end of the jump. Poses corresponding to the single data point highlighted with the arrow are shown in i. Displayed data represents the correlation value highlighted with a white rectangle in j. **l**, Jump distance as a function of z-velocity of the thoracocervical joint for both animals 175 ms before the end of the jump. Displayed data represents the correlation value highlighted with a white rectangle in j.

Discussion

We developed an anatomically constrained model (ACM) for tracking skeletal poses of untethered freely-moving animals, at the resolution of single joints, that enabled the quantification of joint kinetics during gait and gap-crossing behaviors. From these kinetic measurements the ACM was able to build a comparative map of the kinetic-sequences throughout decision-making behaviors that could be compared to the behavioral outcome. Accurate generation of skeleton kinetics relied on incorporating skeleton anatomy, requiring smoothness of rotations and imposing motion restrictions of joints[27], as animal poses are limited by both bone lengths and joint angle limits[26]. In addition, we generated ground truth data to quantify both the accuracy of the algorithm used to fit the model skeleton to the behavioral data and also the performance of the ACM at estimating limb and joint trajectories. Comparing the bone lengths of the fitted skeleton to the actual bone lengths measured from anatomical MRI scans for animals of a range of sizes we showed that accurate model fits could be obtained for animals with an order of magnitude difference in weight, with equally good fitting results independent of animal size. By directly measuring the animals paw positions and comparing with positions returned by the ACM, we showed that the combination of both anatomical and temporal constraints significantly reduced the errors relative to the naïve skeleton model or either constraint alone. This combination allowed accurate estimation not only of the location and orientation of the paws but also the accelerations and velocities of the joints during the measured behaviors. The ACM was capable of accurately quantifying limb kinetics during cyclic gait behaviors and more complex behaviors, such as gap-crossing, even when limbs were partially occluded.

Lastly, the ACM remained accurate over a large range of animal sizes, 72 g – 735 g, with the expectation that the ACM approach would also work for smaller rodents, such as mice. Our approach ushers in a suite of new possibilities for studying the biomechanics of motion during complex behaviors in freely-moving animals and complements recent developments in detailed surface tracking[42]. It opens up future investigations to also model forces applied by tendons and muscles[26,27] and starts bridging the gap between neural computations in the brain[3-8,13] and the mechanistic implementation of complex behavior[9-11], such as rodent emotion[43].

Recently, various studies relying on deep neural networks approached the problem of detecting an animal's pose in the form of 2D features from an image without anatomically constrained skeleton models[21,23,24]. 3D poses can be inferred from these 2D features by means of classical calibrated camera setups[44], however the 2D detection in one camera image does not benefit from the information from other cameras and the triangulation may suffer from resulting mislabeling of

2D features as well as missing detections due to occluded features. A recent approach[32] overcomes many of these issues by mapping from recorded images directly to 3D feature locations, again using deep learning, and is capable of classifying animal behaviors across many species[32]. However, it does not possess explicit inherent temporal connections between frames and thus no persistent skeletal model with fixed bone lengths over time or anatomical constraints on joint angles. In contrast, the ACM uses a different approach: With DLC[24] we used an existing method to detect 2D anatomical markers and inferred 3D positions and kinetics of movement with the RTS smoother based on anatomical constraints and mechanistic knowledge of bone rotations[26,27], considering the trajectory of 3D positions over time. While the goal of the current study was to infer skeletal kinematics of freely behaving animals but not real-time behavior tracking[23,45], we expect future work in the field of 3D animal pose estimation to combine both supervised learning techniques[32,42] and mechanistic model constraints[26,27], to simultaneously capitalize on their different strengths, e.g. by applying a smoother with anatomical knowledge like the ACM directly to 3D positions from an image-to-3D framework[32]. Our approach has the capacity to extend existing methods and not only to enhance the detail in which animal behavior can be studied and quantified, but it also provides an objective and accurate quantification of limb and joint positions for comparison with neuronal recordings.

Acknowledgments

We thank David Greenberg for developing initial video acquisition software, Florian Franzen for help with an external camera trigger, Jan-Matthis Lückmann for initial code for reading single frames, Michael Bräuer, Rolf Honnef, Michael Straussfeld and Bernd Scheiding from the mechanical workshop for fabrication of the setup components, Kristina Barragan, Abhilash Cheekoti, Nada Eiadeh, Gizem Görünmez, Yolanda Mabuto, Anastasiia Nychyporchuk, Aarya Pawar and Nurit Zorn for manually labelling of images and Julia Kuhl for illustrations. Funding was obtained from Stiftung caesar and the Max Planck Society. A.M. is a graduate student with the International Max Planck Research School (IMPRS) for Brain and Behavior. K.S. and E.L. were funded in part by DFG Reinhard Koselleck Project, DFG SCHE 658/12. J.H.M. was supported by the German Research Foundation (DFG) through Germany's Excellence Strategy (EXC-Number 2064/1, Project number 390727645).

Author contributions: Development of anatomically constrained model concept: A.M., J.H.M., and J.N.D.K. Algorithm design and implementation: A.M. and J.H.M. Experimental design and setup: A.M., K-M.V., D.J.W., J.S., and J.N.D.K. Animal preparation and data collection: A.M. and D.J.W. MRI sequences and data collection: A.M. E.L. and K.S. Analysis design and implementation: A.M. and J.N.D.K. Manuscript preparation: A.M. and J.N.D.K.

Methods

Obtaining video data of behaving animals. All experiments were performed in accordance with German guidelines for animal experiments and were approved by the Landesamt für Natur, Umwelt und Verbraucherschutz, North Rhine-Westphalia, Germany. Six Lister Hooded rats (Charles River Laboratories), weighting 174 g (animal #1), 178 g (animal #2), 71 g (animal #3), 72 g (animal #4) , 735 g (animal #5) and 699 g (animal #6) on the day of the experiment, were used. Anatomical landmarks for tracking limb and body positions consisted of black or white ink spots (5-8 mm diameter, black markers: Edding 3300, white markers: Edding 751, Edding, Ahrensburg, Germany) which were painted onto the fur in a stereotypical pattern that was near-symmetrical around the animals' mid-sagittal axis (Supplementary Fig. 2). For application of the anatomical markers, animals were anesthetized with isoflurane (2-3%) and body temperature maintained around 37.5°C using a heating pad and temperature probe. After this labeling procedure the animals were allowed to recover for approx. 45 min before datasets were acquired on a gap-crossing track and open arena. The open arena was 80x105 cm^2 with 50 cm high walls colored gray to promote contrast with the animals and markers. The gap-crossing track consisted of two 50x20 cm^2 platforms, mounted 120 cm off the ground on a slide mechanism to allow manual adjustment of the distance between the platforms in the range from 0 to 60 cm. The platforms were positioned such that with the gap closed they met along one of the 20 cm edges. The edges of the platform, apart from the edge along which the two platforms met, were equipped with a 2.5 cm tall wall. The floor of the platforms was covered with a layer of neoprene material to promote a secure grip for the animals' feet. A water delivery spout was located in the center of the 20 cm track edge opposite of where the platforms met. To encourage gap-crossing behavior, animals were water-restricted, having full access to water two days per week, and otherwise having access to water only on the gap-crossing track. Fifty to one hundred microliters of water was available at the delivery spout after each successful cross of the gap. Animals received a minimum of 50% of their daily *ad libertum* water consumption either during the training or recording sessions or as a supplement after the last session of the day if they had not already consumed at least this amount. Gap-crossing training commenced approximately two weeks prior to the recording, and consisted of two daily sessions. Gap distances were pseudo-random, with the gap distance reduced in cases where the animal refused to cross. Both setups were homogeneously illuminated using eight 125 cm long white LED strips with 700 lm/m (PowerLED, Berkshire, United Kingdom), arranged equidistantly in a patch of 125x80 cm^2 and 125x55 cm^2 at a distance of 130 cm and 150 cm above the ground of the open arena and the gap-crossing track, and data were acquired using four synchronously triggered digital cameras (ace acA1300-200um, Basler, Ahrensburg, Germany) mounted above the setups and set in such a way that all parts of the setup were covered by at least two cameras, with the majority of both setups covered by all four. Datasets consisted of 1280x1024 px^2 image frames with an acquisition time of 2.5 ms recorded at 100 Hz for the gait dataset in the open arena and 200 Hz for the gap-crossing dataset. For quantification of the animals' foot positions we used a custom-made FTIR plate, consisting of a single sheet of 60x60 cm^2, along the edges of which an IR-LED strip (Solarox 850 nm LED strip infrared 850 nm, Winger Electronics GmbH & Co. KG, Germany) was mounted such that IR light could propagate through the FTIR plate from two opposing sites. Animal position data were acquired from the overhead cameras at 200 Hz for these experiments, and paw placements were recorded using two additional cameras (ace acA1300-200um, Basler, Ahrensburg, Germany), synchronized with the overhead cameras, mounted underneath the plate and equipped with infrared-highpass filters (Near-IR Bandpass Filter, part: BP850, useful range: 820-910 nm, FWHM: 160 nm, Midwest Optical Systems, Inc., Palatine, USA). These cameras were set to acquire 1280x1024 px^2 frames with an acquisition time of 2.5 ms recorded at 200 Hz. The total FTIR dataset consisted of 29 sequences with a total of 36250 frames in each of the four cameras and a total duration of 181.25 s. The gait dataset consisted of 27 sequences with a total of 14650 frames in each of the four cameras and a total duration of 146.5 s. The gap-crossing dataset consisted of 44 sequences with a total of 8800 frames in each of the four cameras and a total duration of 44 s.

Obtaining MRI scans to evaluate learned skeleton models. To locate labeled surface markers, custom-made MRI markers (Premium sanitary silicone DSSA, fischerwerke GmbH & Co. KG, Waldachtal, Germany) were attached to the respective positions on the surface of the animals' bodies. MR imaging was performed at a field strength of 3T (Magnetom Prisma, Siemens Healthineers, Erlangen, Germany), using the integrated 32-channel spine coil of the manufacturer. The data was acquired *ex vivo* in six rats using a 3D turbo-spin-echo sequence with variable-flip-angle echo trains (3D *TSE-VFL*). Detailed MR protocol parameters for 3D TSE-VFL imaging with a turbo factor of 98 were as follows: a repetition time of 3200 ms, an effective echo time of 284 ms, an echo train duration of 585 ms, and an echo spacing of 6.3 ms using a readout bandwidth of 300 Hz/px for one slab with 208 slices covering the whole rat at an isotropic resolution of 0.4x0.4x0.4 mm^3.

Calibrating multi-camera setups. We based the calibration of multiple cameras on a pinhole camera model with 2nd order radial distortions (Supplementary Text) and OpenCV[46] functions for detection of checkerboard corners. The checkerboards we used had additional ArUco[47] markers printed on them. To obtain the calibration an objective function penalizing mismatches between detected and projected corners was defined and minimized via gradient descent optimization using the Trust Region Reflective algorithm[48] (Supplementary Text).

Defining a 3D skeleton model. The generalized skeleton model consisted of joints, modeled as vertices, and inter-joint segments, modeled as edges and which could represent multiple bones from the true skeleton (Supplementary Fig. 1). The front limbs were modeled as four edges, representing the clavicle, humerus, radius/ulna and metacarpal/phalanges. The associated vertices corresponded to the shoulder, elbow and wrist, with the last vertex representing the tip of the middle phalanx. The hind limbs were modeled as five edges representing the pelvis, femur, tibia/fibula, tarsus and phalanges, with the associated vertices representing the hip, knee, ankle and metatarsophalangeal joints, with the last vertex representing the tip of the middle tarsal. The tail was modeled as five edges and five vertices, with the last vertex representing the tip of the tail. The spine was modeled as four edges, representing the cervical, thoracic and lumbar spinal regions and the sacrum, with three intervening vertices. The head was modeled as a single edge, with a vertex at the tip to represent the nose, and a second vertex representing the joint to the first cervical vertebra. The resting pose of the 3D skeleton model was defined as the pose generated by the pose reconstruction scheme, when all the parameters encoding bone rotations were set to zero. In this pose all edges (i.e. bones) pointed towards the positive z-direction of the right-handed world coordinate system, except the four edges approximating the clavicle/collarbone and sacrum/pelvis, where edges of the right limb faced towards and edges of the left limb faced against the positive x-direction of the world coordinate system (Supplementary Fig. 1). The configuration of these four edges was also kept constant during pose reconstruction, so that edges representing the cervical and lumbar vertebrae were always orthogonal to the edges representing the clavicle and sacrum. The y-coordinates of all vertices were equal to zero, locating the entire 3D skeleton model in the world's xz-plane while situated in the resting pose. Besides the world coordinate system each edge also had its own right-handed coordinate system located at the start vertex of the corresponding edge, e.g. the coordinate system of the edge representing the left humerus was located at the position of the vertex representing the left shoulder joint. The z-direction of these edge coordinate systems were always identical to the direction in which the associated edges faced. Additionally, anatomical rotations were defined in the edge coordinate systems, so that a rotation around the x-direction became equivalent to flexion/extension, rotations around the y-direction were identical to abduction/adduction and a rotation around the z-direction coincided with internal/external rotation.

Constraining poses based on joint angle limits. We implemented joint angle limits based on measured minimum and maximum values for flexion/extension, abduction/adduction and internal/external rotation in domestic house cats[21]. A comparable set of measured values is to our knowledge not available for rat. For vertices approximating head, spine or tail joints data for joint angle limits was not available, so that we modeled corresponding edges without the capacity to rotate around the z-direction of their associated edge coordinate systems, whereas joint angle limits for rotations around the x- and y-direction were set to +/- 90°. This allowed a respective child-vertex to reach any point within an area spanned by a hemisphere pointing in the direction of the associated edge with radius identical to the length of this edge. Since the resting pose of our 3D skeleton model was not necessarily identical to the pose in which the published joint angle limits were measured in, we calculated the correct rotational limits which were consistent with our resting pose based on the mean of the published values. The resulting joint angle limits were set as follows:

joint	**x (°)**	**y (°)**	**z (°)**
left shoulder	[25,205]	[-85,25]	[-35,35]
right shoulder	[25,205]	[-25,85]	[-35,35]
left elbow	[2.5,145]	[0,0]	[-100,45]
right elbow	[2.5,145]	[0,0]	[-45,100]
left wrist	[-135,35]	[-12.5,37.5]	[0,0]
right wrist	[-135,35]	[-37.5,12.5]	[0,0]
left hip	[35,195]	[-65,25]	[-85,40]
right hip	[35,195]	[25,65]	[-40,85]
left knee	[-145,15]	[0,0]	[0,0]
right knee	[-145,15]	[0,0]	[0,0]
left ankle	[-10,145]	[0,0]	[0,0]
right ankle	[-10,145]	[0,0]	[0,0]
left metatarsophalangeal	[0,0]	[0,0]	[-15,35]
right metatarsophalangeal	[0,0]	[0,0]	[-35,15]

While the published joint angles referred to Euler angles, we used Rodrigues vectors to parameterize rotations (Supplementary Text), since the latter are better suited for pose reconstruction[49]. However, both parameterizations become identical when only a single type of rotation, e.g. flexion/extension, is present at a vertex, which was the case for the measured joint angles[49]. Parameterizing rotations with Rodrigues vectors therefore allowed us to obtain smooth transitions between different types of bone rotations.

Constraining surface marker positions based on body symmetry. When learning surface maker positions and bone lengths we constrained the former to comply with the symmetrically applied surface marker pattern by enforcing box constraints for each spatial dimension, e.g. markers on the left side of an animal were prevented from being placed on the right side. This reduced the total number of free parameters during learning. To reduce this number further we also mirrored surface marker positions in the yz-plane of the associated edge coordinate system when there was a left/right correspondence, i.e. we only learned surface marker positions for the left side which then also determined right-sided surface maker positions due to the mirroring (Supplementary Text). Resulting box constraints for central and left-sided surface maker locations were then defined in the coordinate system of the associated edges and set as follows (Supplementary Fig. 1,2):

marker	**attached joint**	**x**	**y**	**z**
head #1	spine #5	[0,0]	[0,inf)	(-inf,inf)
head #2	spine #5	[0,0]	[0,inf)	(-inf,inf)
head #3	head (leaf)	[0,0]	[0,0)	[0,0]
spine #1	spine #2	[0,0]	[0,inf)	(-inf,inf)
spine #2	spine #2	[0,0]	[0,inf)	(-inf,inf)
spine #3	spine #3	[0,0]	[0,inf)	(-inf,inf)
spine #4	spine #3	[0,0]	[0,inf)	(-inf,inf)
spine #5	spine #4	[0,0]	[0,inf)	(-inf,inf)
spine #6	spine #5	[0,0]	[0,inf)	[0,0]
tail #1	tail #1 (leaf)	[0,0]	[0,0]	[0,0]
tail #2	tail #2	[0,0]	[0,inf)	(-inf,inf)
tail #3	tail #3	[0,0]	[0,inf)	(-inf,inf)
tail #4	tail #4	[0,0]	[0,inf)	(-inf,inf)
tail #5	tail #5	[0,0]	[0,inf)	(-inf,inf)
tail #6	spine #1	[0,0]	[0,inf)	(-inf,inf)
shoulder	shoulder	[0,0]	[0,inf)	[0,inf)
elbow	elbow	(-inf,0]	[0,0]	[0,0]
wrist	wrist	[0,0]	(-inf,0]	[0,0]

finger #1	finger (leaf)	(-inf,inf)	[0,0]	(-inf,inf)
finger #2	finger (leaf)	[0,0]	[0,0]	[0,0]
finger #3	finger (leaf)	(-inf,inf)	[0,0]	(-inf,inf)
side	spine #3	(-inf,0]	(-inf,inf)	(-inf,inf)
hip	hip	[0,0]	[0,inf)	[0,inf)
knee	knee	(-inf,0]	[0,0]	[0,0]
ankle	ankle	(-inf,0]	[0,0]	[0,0]
metatarsophalangeal	metatarsophalangeal	[0,0]	(-inf,0]	[0,0]
toe #1	toe (leaf)	(-inf,inf)	[0,0]	(-inf,inf)
toe #2	toe (leaf)	[0,0]	[0,0]	[0,0]
toe #3	toe (leaf)	(-inf,inf)	[0,0]	(-inf,inf)

The only exception from this was the upper bound of the left-sided surface marker on the shoulder in z-direction for the two large animals (animals #5 and #6), which was also set to 0 in order to prevent the bone lengths of the collarbones to become zero during learning.

Constraining bone lengths based on allometry. We applied loose constraints on the length of limb bones based on the published linear relationships between body weight and bone lengths in rats[35]. The lengths of the following list of limb bones were constrained according to measured slope estimates[35]. Box constraints for bone lengths were calculated from weight-matched lengths plus or minus 10 times the standard deviation, based on the following proportionality factors:

bone name	**slope avg. +/- s.d. (cm/g)**
humerus	0.0075 +/- 0.0005
radius	0.0069 +/- 0.0004
metacarpal	0.0023 +/- 0.0001
femur	0.0102 +/- 0.0006
tibia	0.0114 +/- 0.0006
metatarsal	0.0053 +/- 0.0003

For bones that were not part of the limbs no constraints were enforced, such that corresponding box constraints were set to [0,inf). To ensure that bone lengths of the left and right limbs were identical, we only learned bone lengths of the left-sided limbs, which then also determined right-sided limb bone lengths (Supplementary Text).

Training deep neural networks to detect 2D locations of surface markers. To automatically detect 2D locations of surface makers we used DeepLabCut[24]. For each rat in each dataset an individual neural network was trained on manually labeled images obtained from four different cameras, six trained networks in total. For each image that was used for training a background-subtracted image was generated by subtracting the image acquired 200 ms prior to the frame of interest for the FTIR and gap-crossing datasets and 125 ms prior for the gait dataset. Subsequently, approximate 2D locations of the recorded rats on the background-subtracted images were obtained by calculating the median indices of pixels above a threshold-value of 5 times the standard deviation of each pixel, where the standard deviations were calculated from the first 100 images of each recorded video, which were acquired with the arena or track empty and free of any moving objects. These 2D locations where then used as a center-point to crop the original images to 600x600 px^2. To minimize the influence of visible movements of the experimentors on this center-point detection in the recorded FTIR data set, pixel values of pixels, which did not show the FTIR plate, were set to zero for the recordings of animals #3 to #6. For the FTIR datasets the networks were trained on 4068 images for animal #1, 3980 images for animal #2, 752 images for animal #3, 1100 images for animal #4, 992 images for animal #5 and 1128 images for animal #6. For the gait and gap-crossing datasets 2404 and 3608 images were used respectively for each analyzed animal (animal #1 and #2). Resulting images that did not contain any manually annotated 2D positions of surface markers due to the preprocessing steps not leading to correct cropping, were not used during training. We used DeepLabCut's default settings, with the only two exceptions being that we changed the network architectures to ResNet-152 and enabled mirroring of images for which we paired surface markers with a left/right correspondence[50]. Training was conducted via DeepLabCut 2.1.6.4 downloaded from GitHub (https://github.com/DeepLabCut/DeepLabCut/commit/2f5d32884da2e5c3e4b6ef2a2126f6bb61579060). Once the networks were trained, we used them to obtain 2D locations of surface markers for images of analyzed behavioral sequences, where we set DeepLabCut's pcutoff-parameter[50] to 0.9 and treated detected marker positions below this value as missing measurements.

Performing probabilistic pose reconstruction. To perform probabilistic 3D pose reconstruction, which allows for generating poses using non-linear mathematical operations and where information of an entire behavioral sequence is processed, we implemented an unscented Rauch-Tung-Striebel (RTS) smoother[36,37], whose fundamental principles are based on the ordinary Kalman filter formulation[38]. In this approach, time series data is modelled as a stochastic process generated by a state space model where at each time point hidden states give rise to observable measurements and fulfill the Markov property, i.e. each hidden state only depends on the preceding one (Supplementary Fig. 12). This formalism allowed us to represent each pose as a low-dimensional state variable, corresponding to the location of a reconstructed skeleton as well as the individual bone rotations (dimension of hidden state variable: 50; 3 variables for 3D location of the skeleton plus 47 variables for bone rotations). The measurable 2D locations of surface markers (which were given by the outputs of the trained neural network) had a higher dimensionality and were represented via measurement variables (dimension of measurement variable: maximal 344; 43 surface markers times 4 cameras times 2 variables for the 2D location of a surface marker). We assumed the hidden states to be (conditionally) normally distributed, whereby temporal constraints are implicitly modeled through the transition kernel of the Markov process (i.e. the probabilistic mapping between one state and the next). Our formalism allows for non-linearities in our pose reconstruction scheme, e.g. introduced by the usage of trigonometric functions when applying bone rotations. The unscented RTS smoother can be used to perform probabilistic pose estimation in such a nonlinear state space model, considering both past and future (Supplementary Text). We learned the unknown model parameters (i.e. the initial mean and covariance of the state variables as well as the covariances of the transition and measurement noise) via an expectation-maximization (EM) algorithm[39] (maximal 2944 model parameters total; 50 parameters for mean of initial hidden state variable plus 1275 parameters for covariance matrix of initial hidden state variable plus 1275 parameters for covariance matrix of transition noise plus maximal 334 parameters for diagonal covariance matrix of measurement noise), which aims to maximize a lower bound of the state space model's evidence, i.e. the evidence lower bound (ELBO), accounting for each pose within a behavioral sequence (Supplementary Text). This is achieved by alternating between an expectation step (Supplementary Text), in which we obtain the expected values of the state variables given a fixed set of model parameters via the unscented RTS smoother, and a maximization step (Supplementary Text), in which these model parameters are updated in closed form in order to maximize the ELBO[51]. After convergence of the EM algorithm, final poses were obtained by applying the unscented RTS smoother using the learned model parameters.

Accounting for missing measurements during pose reconstruction. Detecting 2D positions of surface markers via a trained deep neural network was not always successful, e.g. due to marker occlusions. As a result, we only had access to different subsets of all 2D positions during smoothing. This forced us to apply modifications to the plain unscented RTS smoother formulation and the EM algorithm, i.e. we set rows and/or columns of the measurement covariance matrices to zero during the filtering path of the smoother[52,53] and proceeded equivalently with the covariance matrix of the measurement noise when maximizing the ELBO (Supplementary Text).

Enforcing joint angle limits during pose reconstruction. The plain formulation of the unscented RTS smoother does not account for box constraints, so that state variables representing bone rotations are not bounded. To still allow for anatomically constrained pose estimation we instead optimized unbound state variables, which could be mapped onto the correct lower and upper bounds for joint angle limits via sigmoidal functions, i.e. error functions (Supplementary Text). These functions had slope one at the origin and were asymptotically converging towards the lower and upper bounds of the respective joint angle limits.

Learning bone lengths and surface marker positions. To learn bone lengths and surface marker positions we simultaneously fitted our generalized 3D skeleton model to manually labeled 2D positions of surface markers at different time points for each animal. Fitting of the generalized 3D skeleton model was achieved via gradient decent optimization using the L-BFGS-B algorithm[54] in order to minimize an objective function, which penalized mismatches between manually labeled 2D locations of surface markers and those generated via the pose reconstruction scheme (Supplementary Text). Bone lengths, surface marker positions and pose parameters were optimized, while only the pose parameters were unique for every time point and the rest were shared throughout the entire sequence. For this we used sequences of freely-behaving animals recorded via four different cameras totaling to 2404 training frames for animal #1 and #2, 752 training frames for animal #3, 1100 training frames for animal #4, 992 training frames for animal #5 and 1128 training frames for animal #6. Bone lengths were initialized by the mean of their upper and lower bounds or zero when there were no constraints and surface marker positions were initialized to be identical to the joints they were attached too. Initial poses were identical to the resting pose but global skeleton locations and rotations were adjusted prior to the fitting to loosely align with the locations of an animal’s body as seen by the cameras. Once values for bone lengths and surface marker positions were learned, we used them for all further pose reconstructions.

Comparison of skeleton parameters with MRI data. To estimate the quality of these skeleton parameters, we aligned learned 3D skeleton models to manually labeled 3D locations of surface markers obtained from an MRI scan for each animal (Fig. 1b, bottom). To determine the 3D positions of the respective spine joints in the MRI scan, we counted vertebrae such that each modeled spine segment matched its anatomical counterpart with respect to the number of contained vertebrae[33]. One MRI surface marker was not recoverable in the MRI dataset from one animal (right metatarsophalangeal marker, animal #1), and in this case we labeled the 3D location on the animal's body closest to the position of the missing marker. Again, we used gradient decent optimization of an objective function, so that manually labeled 3D markers locations were matched with the ones given by our model. Skeleton parameters were kept constant and only pose parameters changed during optimization. All ground truth joint positions except those for the metatarsophalangeal joints could be identified manually in the MRI scan (4 joint locations total). These missing locations were assumed to be identical to the positions of the corresponding metatarsophalangeal markers.

Defining four different models to evaluate the influence of anatomical and temporal constraints. In the ACM anatomical and temporal constraints were enforced and poses were reconstructed using the unscented RTS smoother together with the EM algorithm. This was also the case for the temporal model but joint angle limits of limb joints, which were not equal to [0°,0°], were set to [-180°,180°], effectively allowing full 360° rotations at the respective joints. Pose parameters for these two models were initialized by fitting the pose of the first time point of a behavioral sequence equivalently to how we learned the skeleton parameters with the only exception that automatically detected instead of manually labeled 2D locations of surface markers were used. The covariance matrices for the initial state variables as well as the state and the measurement noise, which were learned via the EM algorithm, were initialized by setting all diagonal entries to 0.001 and off-diagonal entries to zero, while the latter were also kept constant for the measurement noise covariance matrix during the maximization step of the EM algorithm. For the joint angle and the naïve skeleton model, where only anatomical or no constrains were enforced, we did not use the unscented RTS smoother but reconstructed every pose in the same way we initialized the ACM and the temporal model. Here poses within a behavioral sequence at a certain time point were initialized with the reconstructed pose of the previous time point and joint angle limits of limb joints were set to [-180°,180°] in the naïve skeleton model.

Evaluating pose reconstruction accuracy via a FTIR touch sensing system. To obtain ground truth data paw centers and three individual fingers/toes were manually labeled for each limb in every 40th frame of the FTIR dataset. Images from the calibrated underneath cameras were used and paw centers were identified as the interpolated intersection of the three fingers/toes. Manually labeled marker locations were then projected onto the surface of the transparent floor and xy-positions were calculated as the intersection between this surface and the corresponding epipolar lines. Paw positions and orientation errors where then calculated in the coordinate system of the transparent floor based on these xy-coordinates. Velocity and acceleration values for the four different models were derived from central finite differences (order of accuracy: 8) based on the reconstructed 3D positions of the metatarsophalangeal/wrist and finger/toe markers. Paw position errors of undetected markers where obtained by only using paw position errors of surface markers that were not detected by the trained neural network (i.e. pcutoff < 0.9). When ordering these errors according to the time spans that passed since a respective maker was successfully detected, differentiating between the ACM/temporal and joint angle/naïve skeleton model was necessary. As the unscented RTS smoother incorporates information from the past as well as from the future, time spans until the next detection need to be treated equally to time spans since the last detection, i.e. the direction of the time axis becomes irrelevant. For the ACM/temporal model time spans were calculated as the minimum of the time spans since the last or until the next detection, whereas for the joint angle/naïve skeleton model only time spans since the last detection were relevant, as the smoother was not used here. For the resulting analysis we only included errors for which the corresponding sample size was at least 10.

Analyzing gait data. In order to extract gait periodicity, we normalized reconstructed poses by applying a coordinate transformation on 3D joint locations, such that the new origin was identical to the joint which connects lumbar vertebrae with the sacrum and the new x-direction pointed towards the xy-position of the joint linking cervical with thoracic vertebrae. Given the new joint coordinates we calculated normalized x-positions and bone angles as well as their first temporal derivatives (i.e. normalized x-velocity and angular velocity) of limb joints, where bone angles were defined as the angle between the new x-direction and a respective bone. To model auto-correlations of normalized x-positions, we minimized an objective function penalizing mismatches between data from the four different traces of each limb and the corresponding estimate calculated using a single damped sinusoid via gradient decent optimization. To obtain population averages of normalized x-positions, bone angles and their temporal derivatives, we detected midpoints of swing phases by identifying maximum peaks of normalized x-velocities above 25 cm/s. An individual trace was extracted containing data up to +/- 200 ms around each peak. These traces were then aligned with respect to their associated velocity peaks and then averaged across the entire population. To obtain traces from only tracking surface markers alone, 3D positions of surface markers where interpolated based on their inferred 2D locations given by the trained neural network, whereas for each 3D position only the two most likely 2D locations where taken into account, i.e. the two 2D locations with the highest pcutoff-value.

Analyzing gap-crossing data. Each of the 44 gap-crossing sequences was 1 s long and contained 200 frames per camera totaling 35200 frames. Due to the limited number of gap-crossing events and recorded frames, we used 20% of the frames to train the neural network, i.e. we took every 5th frame of the recorded gap crossing sequences for its training and deployed it to automatically process all frames once training was completed. Similar to the analysis of gait data, velocity values were derived from central finite differences (order of accuracy: 8) of reconstructed 3D joint positions and joint angles were defined as the angle between two connected bones. To obtain start-, mid- and end-point for each jump we averaged joint angles of all spine and hind limb joints and identified the time point where this averaged metric reached its global minimum for each gap-crossing sequence. The averaged metric was characteristic for each jump, i.e. distinct peaks were always present in the following order: local minimum, local maximum, global minimum, local maximum, local minimum. This allowed us to extract the start- and end-point of each jump by finding the first and last local minimum of this sequential pattern. Resulting jump start- and end-points were in close agreement with those obtained from manual assessments of gap crossing sequences by a human expert. Jump distances were calculated as the absolute xy-difference of the average hind paw positions, i.e. average of left/right ankle, metatarsophalangeal and toe joint positions, at the start- and end-point of each jump. To obtain population averaged poses for the jump start-, mid- and end-points, we normalized each pose equivalently to the analysis of gait data. This aligned the resulting poses at their origin and we were able to calculate characteristic jump poses by averaging them across the entire population. For the population averaged mean angle traces we aligned each individual trace according to the mid-point of each jump and then averaged across the entire population. To highlight the diversity of the data given by the reconstructed poses, we calculated distance- and auto-correlations of several different metrics and joints: jump distances were correlated with spatial z-velocities and angular velocities of spine joints at time points up to 400 ms before the end of a jump and absolute spatial velocities and angular velocities of hind limbs joints were correlated with each other at the start-point of a jump. Since differences in bone lengths for each animal dominated the correlation for spatial position and joint angle we only focused on their first temporal derivatives.

Computing hardware. All pose reconstructions and analyzes were conducted on a workstation equipped with an AMD Ryzen 7 2700x CPU, 32 GB DDR4 RAM, Samsung 970 EVO 500 GB SSD, and a single NVIDIA GeForce RTX 1080 Ti (11 GB) GPU. The installed operating system was Ubuntu 18.04.5 LTS. Training DeepLabCut was either conducted on a NVIDIA GeForce RTX 1080 Ti (11 GB) GPU, using CUDA version 10.0 and NVIDIA driver version 410.48, or a NVIDIA

GeForce RTX 2080 Ti (11 GB) GPU, using CUDA version 11.0 and NVIDIA driver version 450.80.02.

Code availability. Code for performing pose reconstructions will be made publicly available on GitHub: https://github.com/bbo-lab/ACM

Data availability. Raw data available on request.

References:

1 Shadmehr, R., Smith, M. A. & Krakauer, J. W. Error correction, sensory prediction, and adaptation in motor control. *Annu Rev Neurosci* **33**, 89-108, doi:10.1146/annurev-neuro-060909-153135 (2010).

2 Kawai, R. *et al.* Motor cortex is required for learning but not for executing a motor skill. *Neuron* **86**, 800-812, doi:10.1016/j.neuron.2015.03.024 (2015).

3 Maynard, E. M. *et al.* Neuronal interactions improve cortical population coding of movement direction. *J Neurosci* **19**, 8083-8093 (1999).

4 Georgopoulos, A. P., Kettner, R. E. & Schwartz, A. B. Primate motor cortex and free arm movements to visual targets in three-dimensional space. II. Coding of the direction of movement by a neuronal population. *J Neurosci* **8**, 2928-2937 (1988).

5 Georgopoulos, A. P., Schwartz, A. B. & Kettner, R. E. Neuronal population coding of movement direction. *Science* **233**, 1416-1419, doi:10.1126/science.3749885 (1986).

6 Moran, D. W. & Schwartz, A. B. Motor cortical representation of speed and direction during reaching. *J Neurophysiol* **82**, 2676-2692, doi:10.1152/jn.1999.82.5.2676 (1999).

7 Churchland, M. M. *et al.* Neural population dynamics during reaching. *Nature* **487**, 51-56, doi:10.1038/nature11129 (2012).

8 Wagner, M. J. *et al.* A neural circuit state change underlying skilled movements. *Cell* **184**, 3731-3747 e3721, doi:10.1016/j.cell.2021.06.001 (2021).

9 Parker, P. R. L., Brown, M. A., Smear, M. C. & Niell, C. M. Movement-Related Signals in Sensory Areas: Roles in Natural Behavior. *Trends Neurosci* **43**, 581-595, doi:10.1016/j.tins.2020.05.005 (2020).

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

11 Cisek, P. & Kalaska, J. F. Neural mechanisms for interacting with a world full of action choices. *Annu Rev Neurosci* **33**, 269-298, doi:10.1146/annurev.neuro.051508.135409 (2010).

12 Skocek, O. *et al.* High-speed volumetric imaging of neuronal activity in freely moving rodents. *Nat Methods* **15**, 429-432, doi:10.1038/s41592-018-0008-0 (2018).

13 Klioutchnikov, A. *et al.* Three-photon head-mounted microscope for imaging deep cortical layers in freely moving rats. *Nat Methods* **17**, 509-513, doi:10.1038/s41592-020-0817-9 (2020).

14 Anikeeva, P. *et al.* Optetrode: a multichannel readout for optogenetic control in freely moving mice. *Nat Neurosci* **15**, 163-170, doi:10.1038/nn.2992 (2011).

15 Luo, T. Z. *et al.* An approach for long-term, multi-probe Neuropixels recordings in unrestrained rats. *Elife* **9**, doi:10.7554/eLife.59716 (2020).

16 de Chaumont, F. *et al.* Computerized video analysis of social interactions in mice. *Nat Methods* **9**, 410-417, doi:10.1038/nmeth.1924 (2012).

17 Hoogland, T. M., De Gruijl, J. R., Witter, L., Canto, C. B. & De Zeeuw, C. I. Role of Synchronous Activation of Cerebellar Purkinje Cell Ensembles in Multi-joint Movement Control. *Curr Biol* **25**, 1157-1165, doi:10.1016/j.cub.2015.03.009 (2015).

18 Machado, A. S., Darmohray, D. M., Fayad, J., Marques, H. G. & Carey, M. R. A quantitative framework for whole-body coordination reveals specific deficits in freely walking ataxic mice. *Elife* **4**, doi:10.7554/eLife.07892 (2015).

19 Marshall, J. D. *et al.* Continuous Whole-Body 3D Kinematic Recordings across the Rodent Behavioral Repertoire. *Neuron* **109**, 420-437 e428, doi:10.1016/j.neuron.2020.11.016 (2021).

20 Ohayon, S., Avni, O., Taylor, A. L., Perona, P. & Roian Egnor, S. E. Automated multi-day tracking of marked mice for the analysis of social behaviour. *J Neurosci Methods* **219**, 10-19, doi:10.1016/j.jneumeth.2013.05.013 (2013).

21 Pereira, T. D. *et al.* Fast animal pose estimation using deep neural networks. *Nat Methods* **16**, 117-125, doi:10.1038/s41592-018-0234-5 (2019).

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

23 Graving, J. M. *et al.* DeepPoseKit, a software toolkit for fast and robust animal pose estimation using deep learning. *Elife* **8**, doi:10.7554/eLife.47994 (2019).

24 Mathis, A. *et al.* DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nat Neurosci* **21**, 1281-1289, doi:10.1038/s41593-018-0209-y (2018).

25 Pereira, T. D., Shaevitz, J. W. & Murthy, M. Quantifying behavior to understand the brain. *Nat Neurosci* **23**, 1537-1549, doi:10.1038/s41593-020-00734-z (2020).

26 Charles, J. P., Cappellari, O., Spence, A. J., Wells, D. J. & Hutchinson, J. R. Muscle moment arms and sensitivity analysis of a mouse hindlimb musculoskeletal model. *J Anat* **229**, 514-535, doi:10.1111/joa.12461 (2016).

27 Charles, J. P., Cappellari, O. & Hutchinson, J. R. A Dynamic Simulation of Musculoskeletal Function in the Mouse Hindlimb During Trotting Locomotion. *Front Bioeng Biotechnol* **6**, 61, doi:10.3389/fbioe.2018.00061 (2018).

28 Gunel, S. *et al.* DeepFly3D, a deep learning-based approach for 3D limb and appendage tracking in tethered, adult Drosophila. *Elife* **8**, doi:10.7554/eLife.48571 (2019).

29 Bauman, J. M. & Chang, Y. H. High-speed X-ray video demonstrates significant skin movement errors with standard optical kinematics during rat locomotion. *J Neurosci Methods* **186**, 18-24, doi:10.1016/j.jneumeth.2009.10.017 (2010).

30 Camomilla, V., Dumas, R. & Cappozzo, A. Human movement analysis: The soft tissue artefact issue. *J Biomech* **62**, 1-4, doi:10.1016/j.jbiomech.2017.09.001 (2017).

31 Huang, K. *et al.* A hierarchical 3D-motion learning framework for animal spontaneous behavior mapping. *Nat Commun* **12**, 2784, doi:10.1038/s41467-021-22970-y (2021).

32 Dunn, T. W. *et al.* Geometric deep learning enables 3D kinematic profiling across species and environments. *Nat Methods* **18**, 564-573, doi:10.1038/s41592-021-01106-6 (2021).

33 Maynard, R. L. & Downes, N. in *Anatomy and Histology of the Laboratory Rat in Toxicology and Biomedical Research* (eds Robert Lewis Maynard & Noel Downes) 23-39 (Academic Press, 2019).

34 Newton, C. D. & Nunamaker, D. *Textbook of small animal orthopaedics*. (J.B. Lippintott Co., 1985).

35 Lammers, A. R. & German, R. Z. Ontogenetic allometry in the locomotor skeleton of specialized half-bounding mammals. *Journal of Zoology* **258**, 485-495, doi:10.1017/S0952836902001644 (2002).

36 Šimandl, M. & Duník, J. DESIGN OF DERIVATIVE-FREE SMOOTHERS AND PREDICTORS. *IFAC Proceedings Volumes* **39**, 1240-1245, doi:https://doi.org/10.3182/20060329-3-AU-2901.00200 (2006).

37 Särkkä, S. *Bayesian Filtering and Smoothing*. (Cambridge University Press, 2013).

38 Kalman, R. E. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering* **82**, 35-45, doi:10.1115/1.3662552 (1960).

39 Dempster, A. P., Laird, N. M. & Rubin, D. B. Maximum Likelihood from Incomplete Data Via the EM Algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)* **39**, 1-22, doi:https://doi.org/10.1111/j.2517-6161.1977.tb01600.x (1977).

40 Ambrose, E. J. A surface contact microscope for the study of cell movements. *Nature* **178**, 1194, doi:10.1038/1781194a0 (1956).

41 Mendes, C. S. *et al.* Quantification of gait parameters in freely walking rodents. *BMC Biol* **13**, 50, doi:10.1186/s12915-015-0154-0 (2015).

42 Bolanos, L. A. *et al.* A three-dimensional virtual mouse generates synthetic training data for behavioral analysis. *Nat Methods* **18**, 378-381, doi:10.1038/s41592-021-01103-9 (2021).

43 Dolensek, N., Gehrlach, D. A., Klein, A. S. & Gogolla, N. Facial expressions of emotion states and their neuronal correlates in mice. *Science* **368**, 89-94, doi:10.1126/science.aaz9468 (2020).

44 Zhang, Z. A Flexible New Technique for Camera Calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**, 1330-1334 (2000).

45 Kane, G. A., Lopes, G., Saunders, J. L., Mathis, A. & Mathis, M. W. Real-time, low-latency closed-loop feedback using markerless posture tracking. *Elife* **9**, doi:10.7554/eLife.61909 (2020).

46 Bradski, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).

47 Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F. J. & Marín-Jiménez, M. J. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition* **47**, 2280-2292, doi:https://doi.org/10.1016/j.patcog.2014.01.005 (2014).

48 Branch, M. A., Coleman, T. F. & Li, Y. A Subspace, Interior, and Conjugate Gradient Method for Large-Scale Bound-Constrained Minimization Problems. *SIAM J. Sci. Comput.* **21**, 1-23 (1999).

49 Moll, G. P. & Rosenhahn, B. in *2009 Workshop on Applications of Computer Vision (WACV).* 1-8.

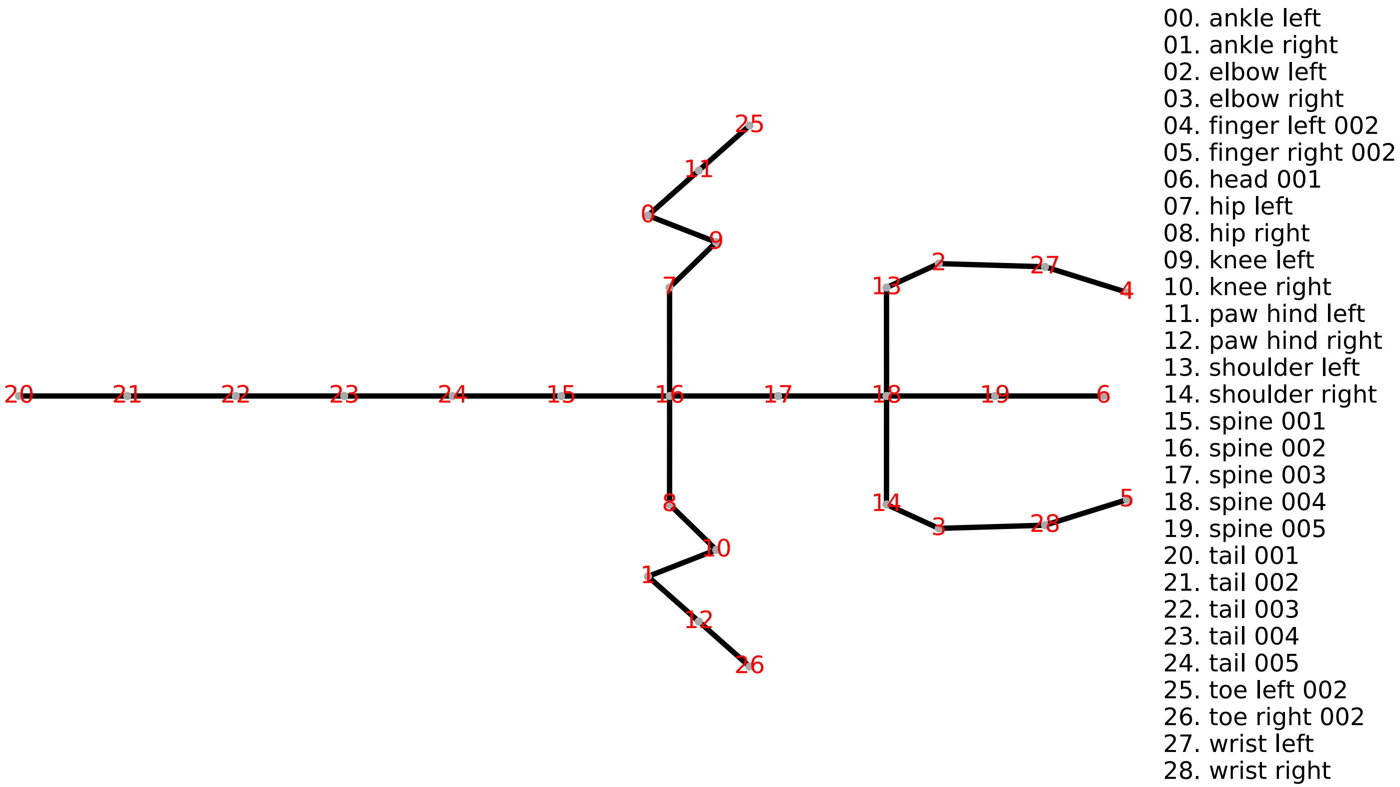
50 Nath, T. *et al.* Using DeepLabCut for 3D markerless pose estimation across species and behaviors. *Nat Protoc* **14**, 2152-2176, doi:10.1038/s41596-019-0176-0 (2019).

51 Kokkala, J., Solin, A. & Sarkka, S. Sigma-Point Filtering and Smoothing Based Parameter Estimation in Nonlinear Dynamic Systems. *arXiv: Methodology* (2015).

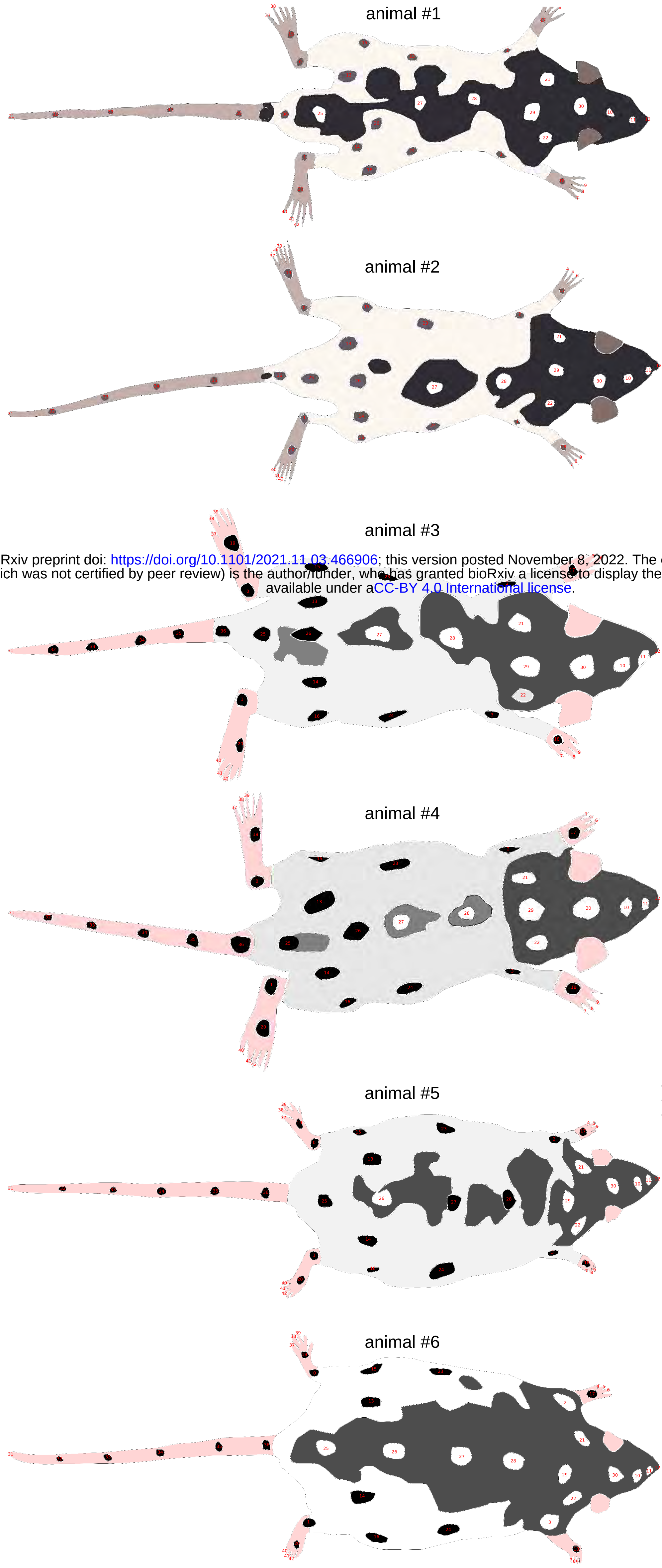
52 Shumway, R. H. & Stoffer, D. S. AN APPROACH TO TIME SERIES SMOOTHING AND FORECASTING USING THE EM ALGORITHM. *Journal of Time Series Analysis* **3**, 253-264, doi:https://doi.org/10.1111/j.1467-9892.1982.tb00349.x (1982).

53 Shumway, R. H. & Stoffer, D. S. *Time Series Analysis and Its Applications: With R Examples*. (Springer International Publishing, 2017).

54 Byrd, R. H., Lu, P., Nocedal, J. & Zhu, C. A Limited Memory Algorithm for Bound Constrained Optimization. *SIAM Journal on Scientific Computing* **16**, 1190-1208, doi:10.1137/0916069 (1995).

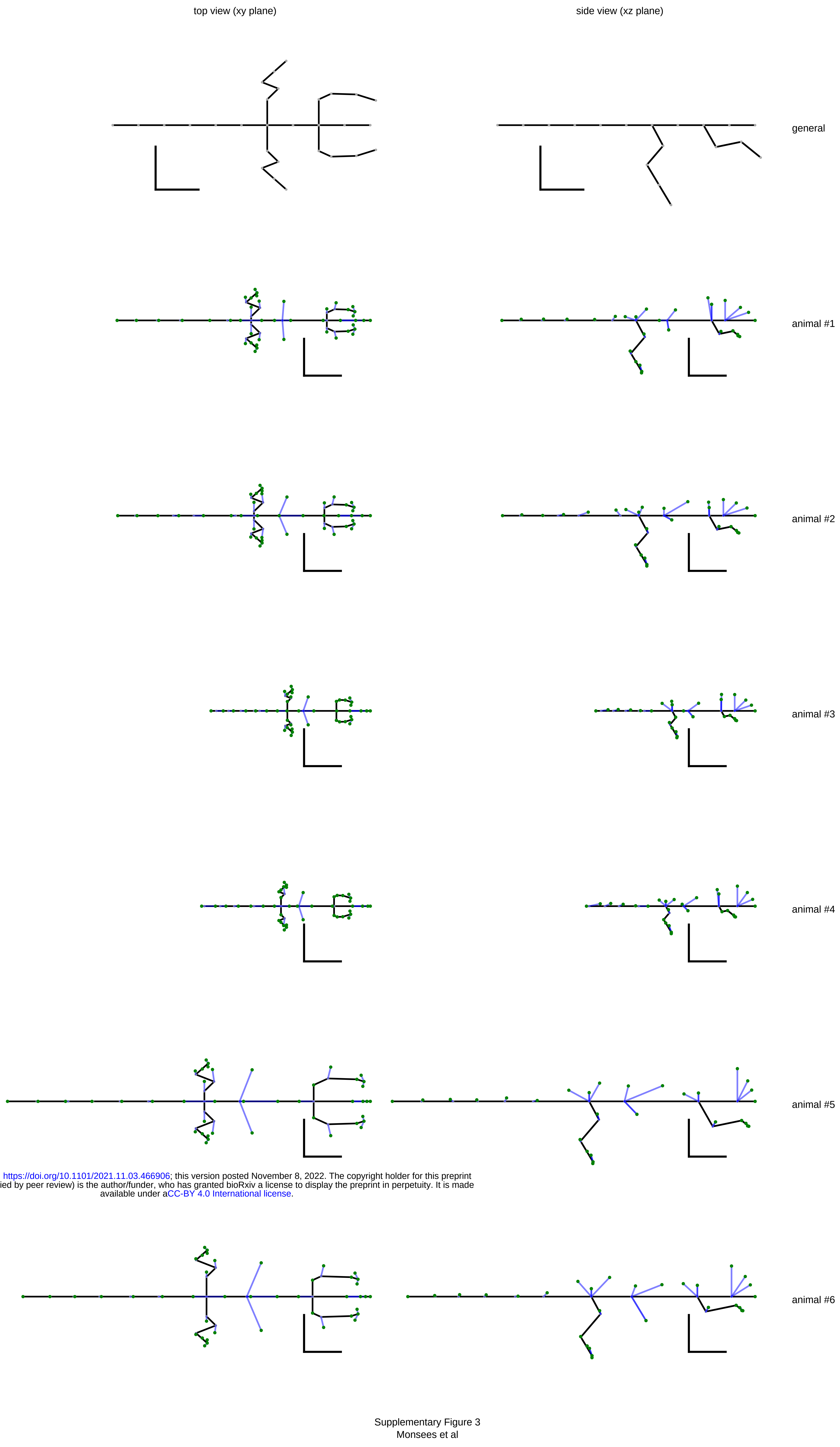


Supplementary Fig. 1 | Projection of the generalized skeleton model in the xy-plane with index and name for each individual joint. For this figure, all bone lengths are equal and all bone rotations have been set to the mean of their upper and lower bounds.

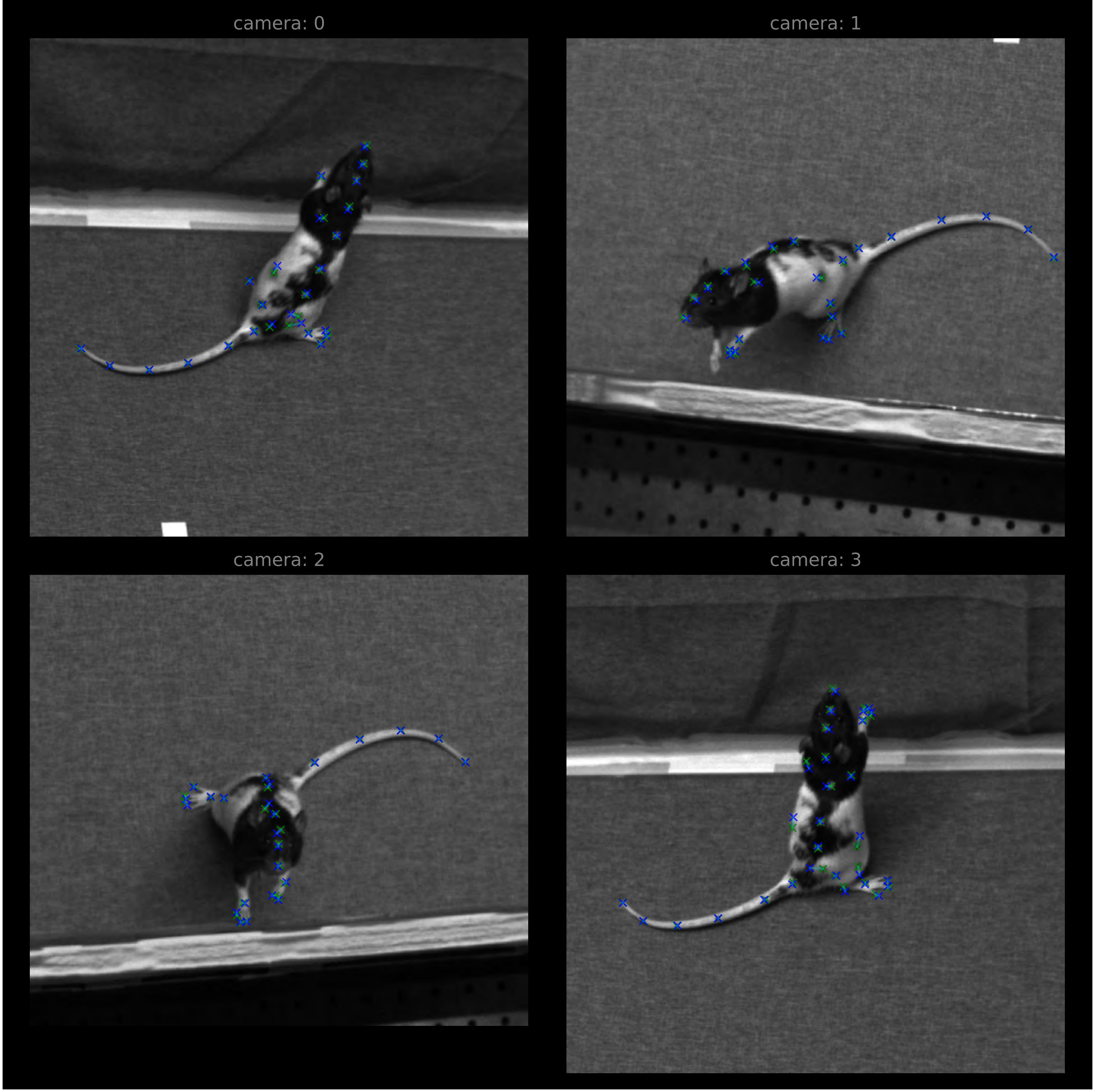


00. ankle (left)
01. ankle (right)
02. elbow (left)
03. elbow (right)
04. finger (left) 001
05. finger (left) 002
06. finger (left) 003
07. finger (right) 001
08. finger (right) 002
09. finger (right) 003
10. head 001
11. head 002
12. head 003
13. hip (left)
14. hip (right)
15. knee (left)
16. knee (right)
17. paw front (left)
18. paw front (right)
19. paw hind (left)
20. paw hind (right)
21. shoulder (left)
22. shoulder (right)
23. side (left)
24. side (right)
25. spine 001
26. spine 002
27. spine 003
28. spine 004
29. spine 005
30. spine 006
31. tail 001
32. tail 002
33. tail 003
34. tail 004
35. tail 005
36. tail 006
37. toe (left) 001
38. toe (left) 002
39. toe (left) 003
40. toe (right) 001
41. toe (right) 002
42. toe (right) 003

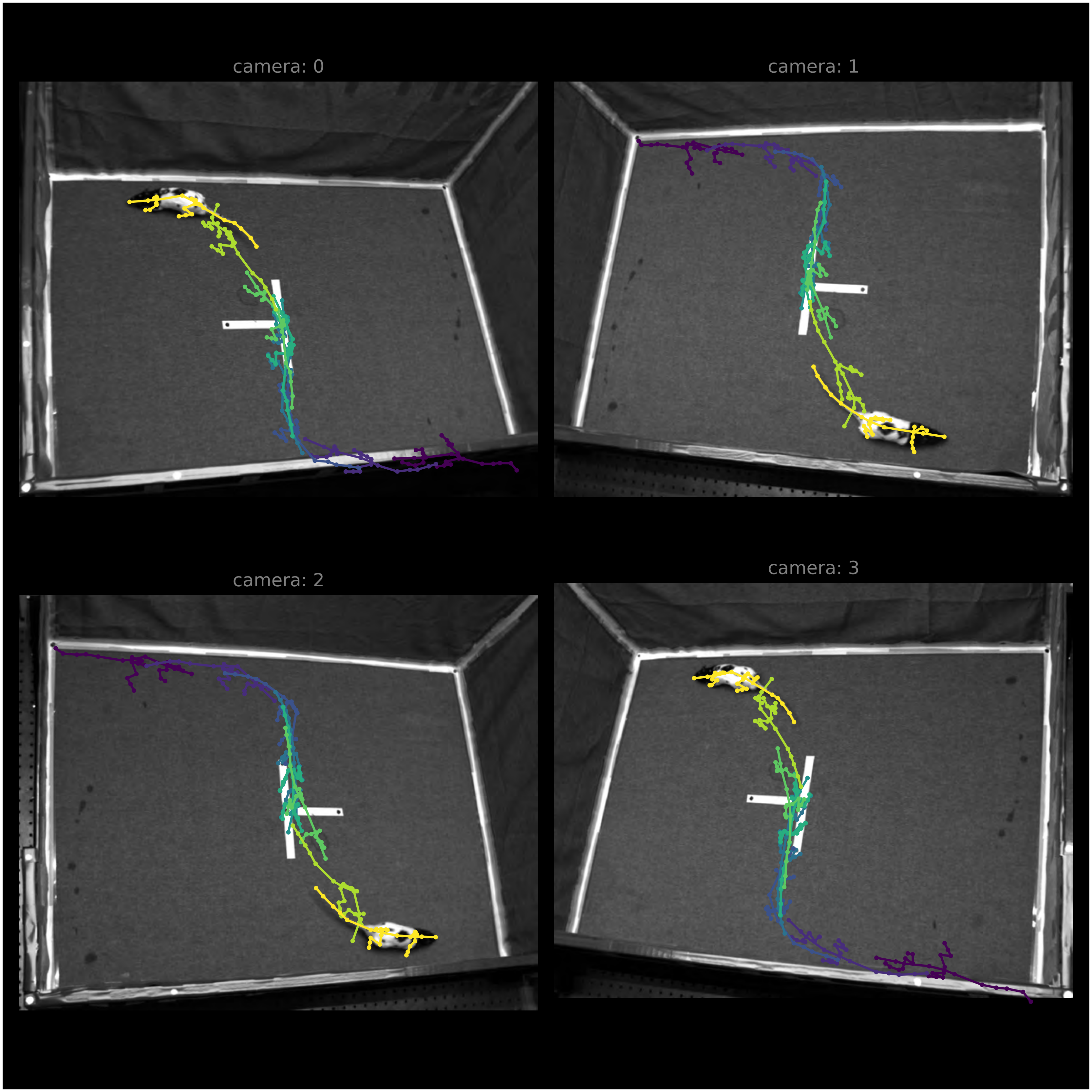
Supplementary Fig. 2 | Schematic images of the six labeled animals with index and name for each individual surface marker



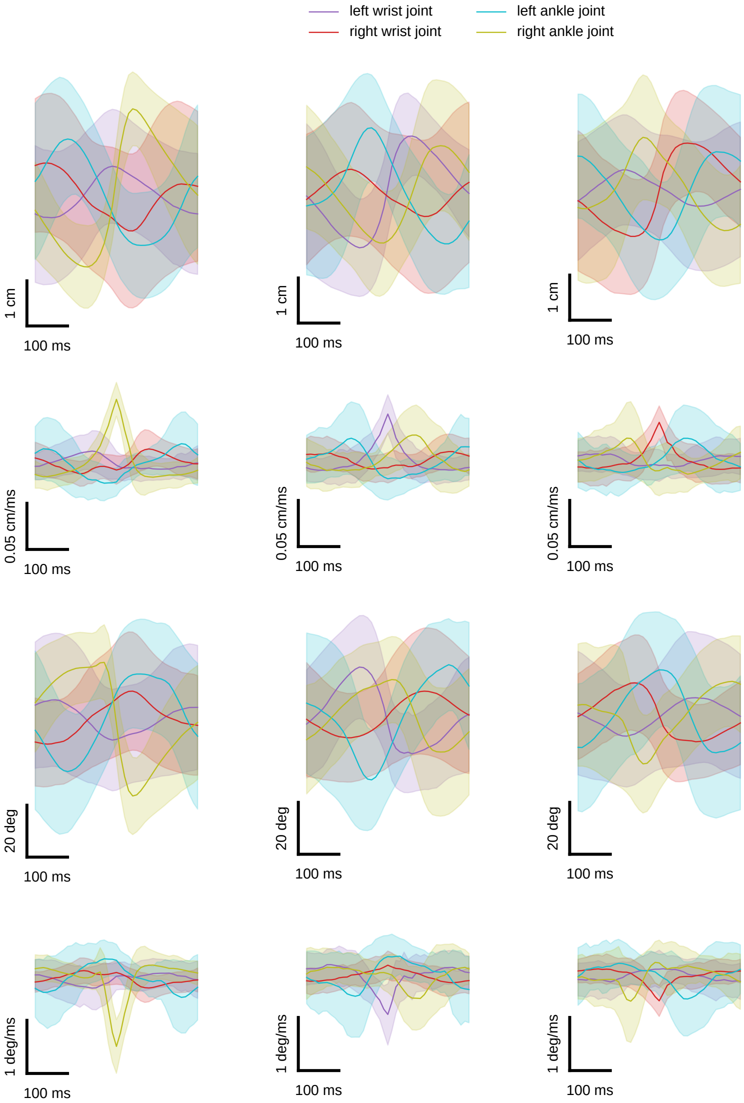
Supplementary Fig. 3 | Projections of the generalized and learned skeleton models for each animal as viewed from the top (xy-view, left column) and from the side (xz-view, right column). All bone rotations were set to the mean of their upper and lower bounds. Green dots indicate the learned positions of surface markers and blue lines join paired joints and surface markers. All scale bars 5 cm.



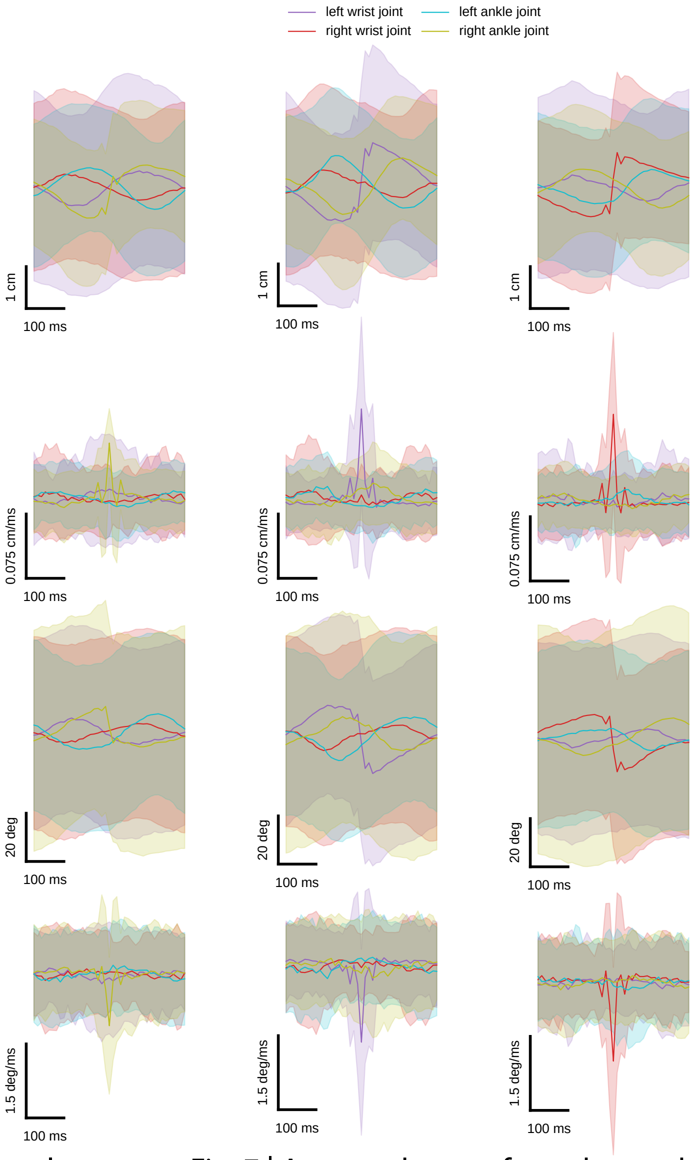
Supplementary Fig. 4 | Synchronous training frames from four calibrated cameras used for learning skeleton lengths and surface maker positions. The figure shows the manually labeled surface marker positions (green), their locations after the skeleton model is learned (blue) and the discrepancies between the two (green lines).



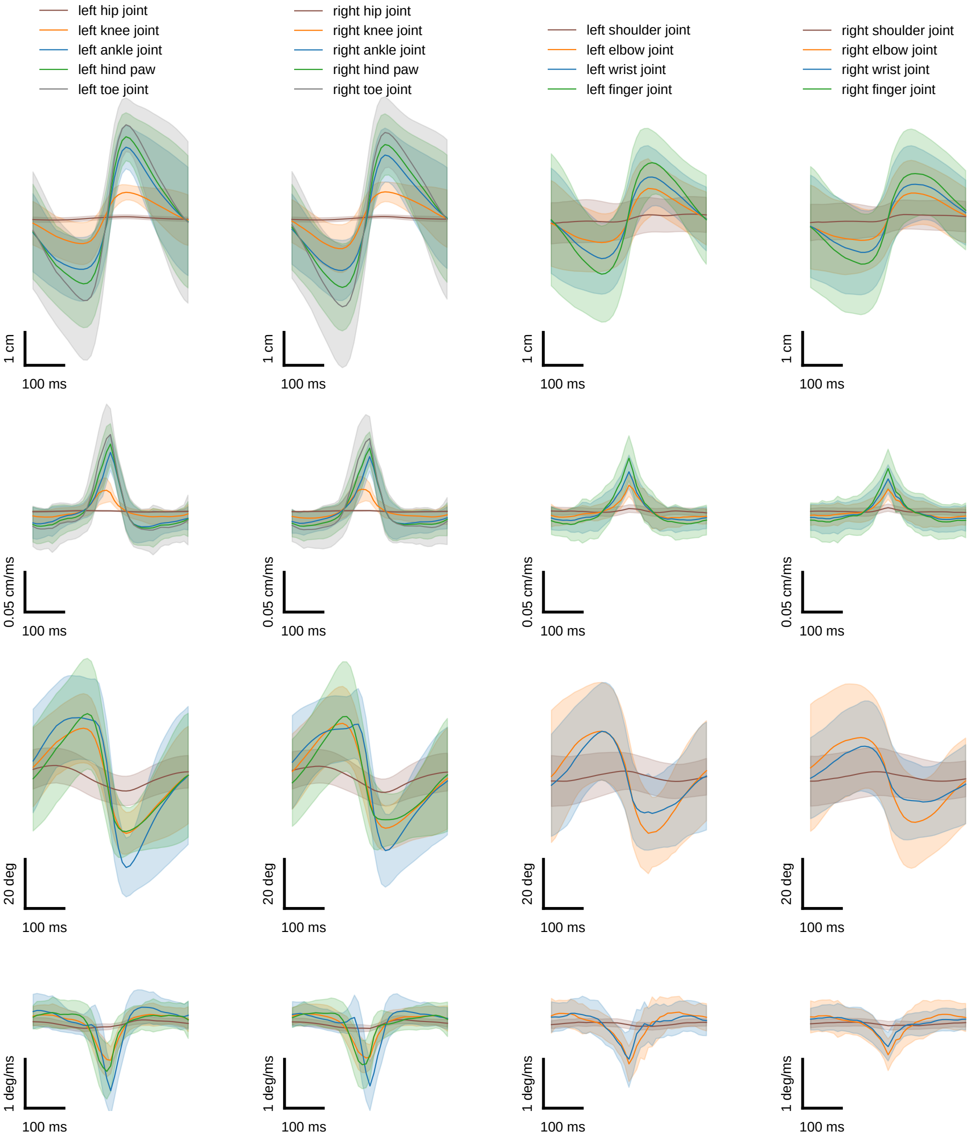
Supplementary Fig. 5 | Synchronous frames from four calibrated cameras which were part of the gait data set. Reconstructed skeleton poses are shown for different time points of the gait sequence. The time difference between poses is 1 s.



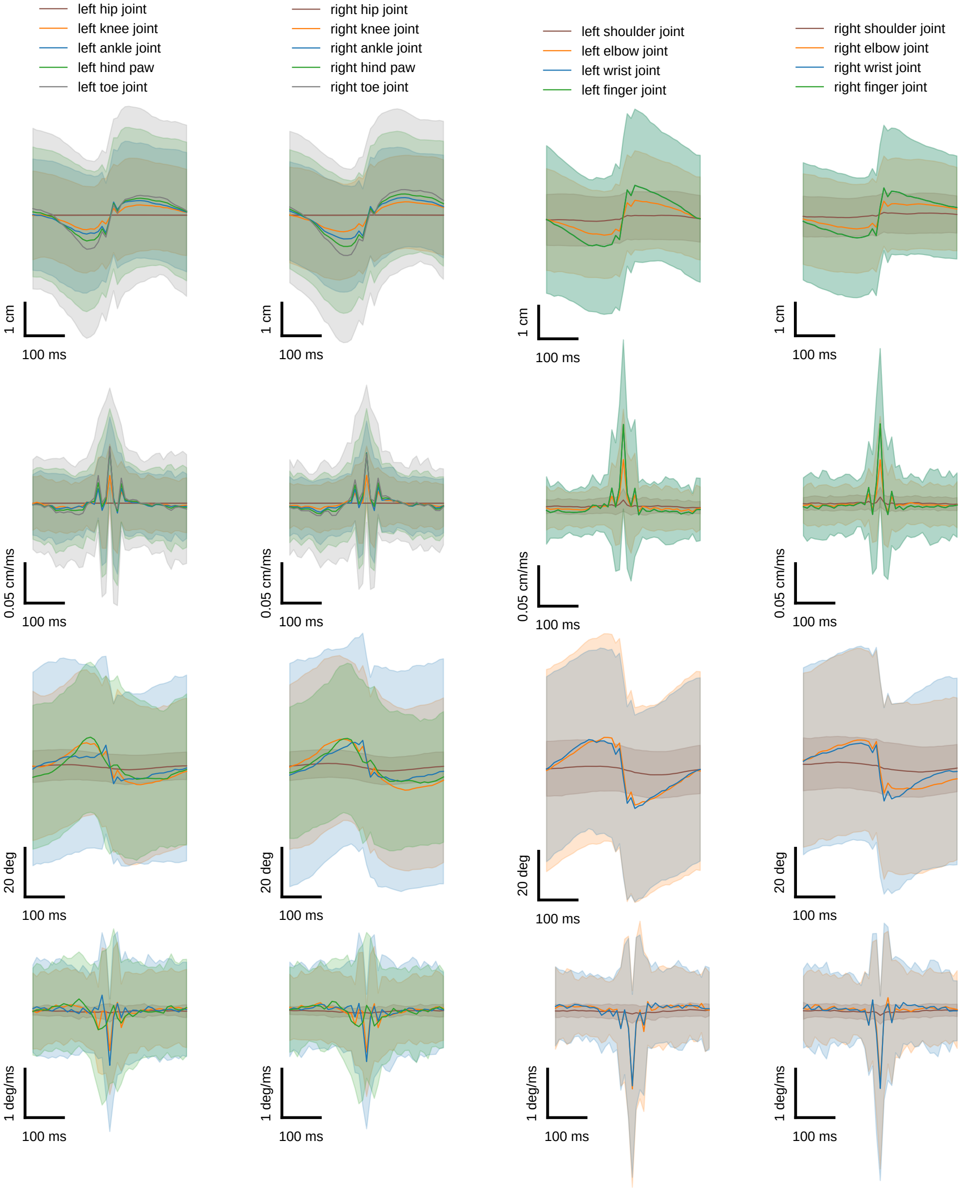
Supplementary Fig. 6 | Averaged traces from the ACM as in Figure 3d,g,j,m (left), but with trace alignment based on the velocity peaks for the right ankle (right column), left wrist (center column) and right wrist joint (right column), instead of aligning to the velocity peak of the left ankle joint.



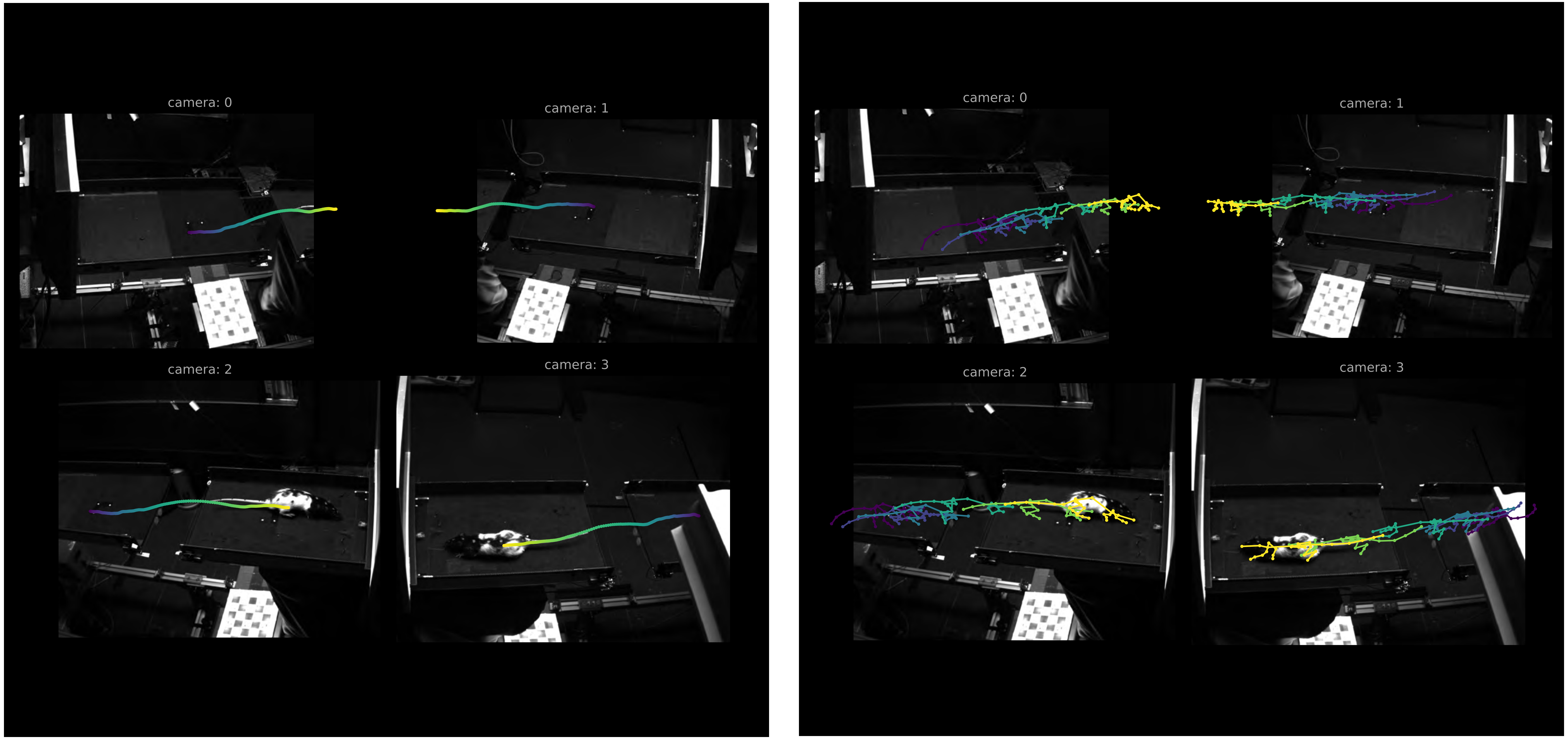
Supplementary Fig. 7 | Averaged traces from the model not constrained by either temporal or joint limit constraints as in Figure 3d,g,j,m (right) aligned to velocity peaks for the right ankle (right column), left wrist (center column) and right wrist joint (right column), instead of aligning to the velocity peak of the left ankle joint.



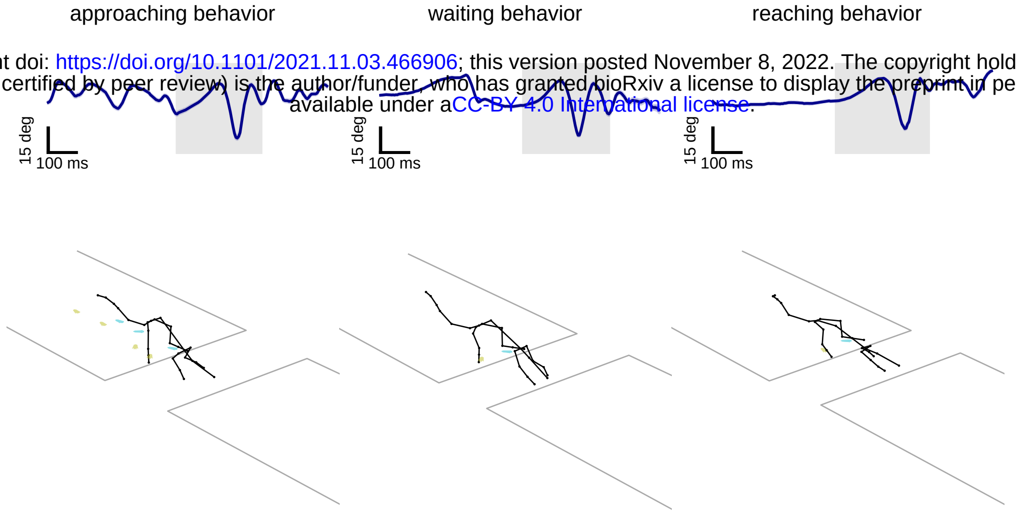
Supplementary Fig. 8 | Averaged traces from the ACM for all joints of the left hind (left column), right hind (center left column), left front (center right column) and right front limb (right column). Traces were aligned to velocity peaks from the left ankle, right ankle, left wrist and right wrist joint of the respective limb.



Supplementary Fig. 9 | Averaged traces from the model not constrained by either temporal or joint limit constraints for all joints of just the left hind (left column), right hind (center left column), left front (center right column) and right front limb (right column). Traces were aligned to velocity peaks from the left ankle, right ankle, left wrist and right wrist joint of the respective limb.

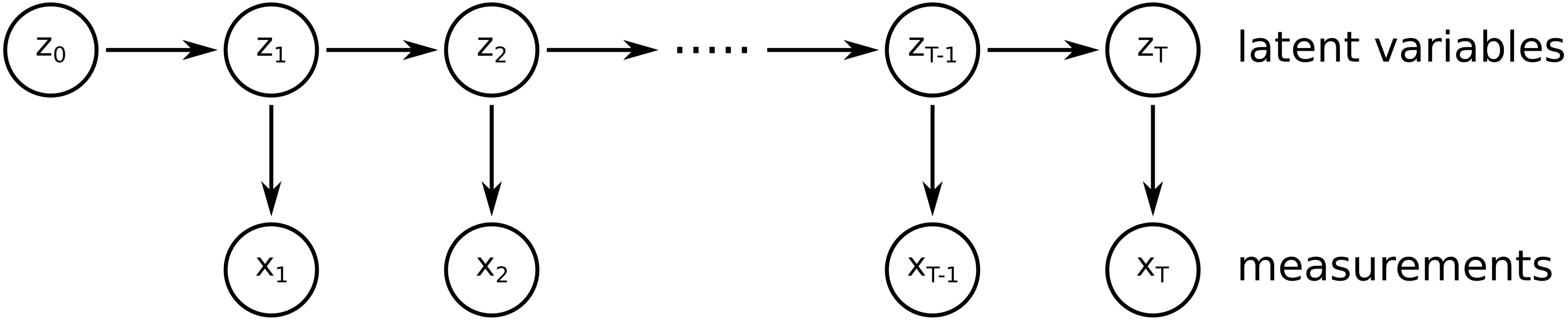


Supplementary Fig. 10 | Synchronous frames from four calibrated cameras which were part of the gap-crossing data set, with overlay of the center of mass (left) and reconstructed skeleton poses (right) shown for different time points.



Supplementary Fig. 11 | Averaged joint-angle trace (average angles of all spine and hind paw joints) as a function of time (top) and reconstructed poses (bottom), showing hind paw footprints (cyan/yellow), at the start of a jump for three different gap crossing events. The animal was approaching the gap fast, with a smooth transition from walking to jumping behavior (left), waiting at the edge (center) or reaching to the other side of the track with its right front limb (right) before crossing the gap.

State Space Model



$$z_t = z_{t-1} + \epsilon_z$$
$$x_t = g(z_t) + \epsilon_x$$

$$z_0 \sim \mathcal{N}(\mu_0, V_0)$$
$$\epsilon_z \sim \mathcal{N}(0.0, V_z)$$
$$\epsilon_x \sim \mathcal{N}(0.0, V_x)$$

model parameters:

$$\theta = \{\mu_0, V_0, V_z, V_x\}$$

Supplementary Fig. 12 | Illustration of the state space model used for describing behavioral time series'.

Contents

1 Parameterizing rotations

We choose to parameterize rotations with Rodrigues vectors as they are well suited for the description of bone rotations with three rotational degrees of freedom [10]. A Rodrigues vector r is formed by combining the axis of rotation $\omega \in \mathbb{R}^3$ and the rotation angle $\theta \in \mathbb{R}$:

$$r = \theta\omega = \theta(\omega_1, \omega_2, \omega_3)^T \tag{1}$$

where $||\omega|| = 1$. To calculate the associated rotation matrix R from a given Rodrigues vector r we can use the following function:

$$\mathrm{f}_{\mathrm{r}\to\mathrm{R}}(r) = I + \hat{\omega}\sin(\theta) + \hat{\omega}^2(1 - \cos(\theta)) = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix} = R \tag{2}$$

where $I \in \mathbb{R}^{3\times3}$ is the identity matrix and $\hat{\omega} \in \mathbb{R}^{3\times3}$ is given by:

$$\hat{\omega} = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}. \tag{3}$$

2 Camera calibration

2.1 Pinhole camera model

To project an arbitrary three-dimensional joint or surface marker location $m_{3\mathrm{D}} \in \mathbb{R}^3$ onto a camera sensor to obtain the corresponding two-dimensional data point $m_{2\mathrm{D}} \in \mathbb{R}^2$, we are using a pinhole camera model [6], which gives the following relationship between the two:

$$\mathrm{f}_{3\mathrm{D}\to2\mathrm{D}}\left(m_{3\mathrm{D}}, \tilde{r}, \tilde{t}, \tilde{k}, \tilde{A}\right) = \tilde{A}\,\mathrm{f}_{\mathrm{distort}}\left(\mathrm{f}_{\mathrm{r}\to\mathrm{R}}(\tilde{r})\,m_{3\mathrm{D}} + \tilde{t}, \tilde{k}\right) = m_{2\mathrm{D}} \tag{4}$$

where $\tilde{r} \in \mathbb{R}^3$ is the Rodrigues vector and $\tilde{t} \in \mathbb{R}^3$ the translation vector of the respective camera, such that the expression $\mathrm{f}_{\mathrm{r}\to\mathrm{R}}(\tilde{r})\,m_{3\mathrm{D}} + \tilde{t}$ maps $m_{3\mathrm{D}}$ from the world coordinate system into the coordinate system of the camera. Given the camera's distortion vector $\tilde{k} \in \mathbb{R}^2$, the function $\mathrm{f}_{\mathrm{distort}}$ applies radial distortions according to

$$\mathrm{f}_{\mathrm{distort}}\left(y, \tilde{k}\right) = \begin{pmatrix} \frac{y_1}{y_3}\left(1 + \tilde{k}_1\tilde{c} + \tilde{k}_2\tilde{c}^2\right) \\ \frac{y_2}{y_3}\left(1 + \tilde{k}_1\tilde{c} + \tilde{k}_2\tilde{c}^2\right) \\ 1 \end{pmatrix} \tag{5}$$

with $y = (y_1, y_2, y_3)^T$ and $\tilde{c} = \left(\frac{y_1}{y_3}\right)^2 + \left(\frac{y_2}{y_3}\right)^2$. The final mapping onto the two-dimensional camera sensor is done using the camera matrix $\tilde{A} \in \mathbb{R}^{2\times3}$ given by

$$\tilde{A} = \begin{pmatrix} \tilde{A}_{11} & 0 & \tilde{A}_{13} \\ 0 & \tilde{A}_{22} & \tilde{A}_{23} \end{pmatrix} \tag{6}$$

where $\tilde{A}_{11}$ and $\tilde{A}_{22}$ are the focal lengths and $\tilde{A}_{13}$ and $\tilde{A}_{23}$ are the x- and y-location of the camera's optical center.

2.2 Calibration of multiple cameras

Given a multi-camera setup with several cameras and overlapping fields of view, we need to infer the initially unknown location and camera parameters of every individual camera in the setup as this allows us to predict where a three-dimensional point in space will be visible on each camera sensor.

This can be achieved by generating a sequence of images showing an object whose physical structure and dimensions are known to us. Hereby, the images are taken synchronously in all cameras, such that the spatial location and orientation of the shown object is identical for a given set of images at a certain time point. For this purpose checkerboards are suited objects as edges of individual tiles can be detected automatically in recorded image frames and the description of their spatial structure requires only a single parameter, i.e. the length of a quadratic tile. Given a multi-camera setup with n_{cam} cameras and n_{time} time points at which we used each camera to record images, which show a checkerboard that has a total of n_{edge} detectable edges, we can calibrate the setup by minimizing a respective objective function via gradient decent optimization using the Trust Region Reflective algorithm [2]:

$$\underset{\substack{\tilde{r}_i,\tilde{t}_i,\tilde{k}_i,\tilde{A}_i,\hat{r}_\tau,\hat{t}_\tau \\ \forall i\in\{1,\ldots,n_{\text{cam}}\} \\ \forall \tau\in\{1,\ldots,n_{\text{time}}\}}}{\arg\min} \sum_{\tau=1}^{n_{\text{time}}} \sum_{i=1}^{n_{\text{cam}}} \sum_{j=1}^{n_{\text{edge}}} \delta_{\tau ij} \left|\left| m_{\tau ij} - \mathrm{f}_{\text{3D}\to\text{2D}} \left(\mathrm{f}_{\text{r}\to\text{R}} \left(\hat{r}_\tau\right) \hat{m}_j + \hat{t}_\tau, \tilde{r}_i, \tilde{t}_i, \tilde{k}_i, \tilde{A}_i \right) \right|\right|^2 \tag{7}$$

where $\tilde{r}_i$ is the Rodrigues vector, $\tilde{t}_i$ is the translation vector, $\tilde{k}_i$ is the distortion vector and $\tilde{A}_i$ is the camera matrix of camera i. The Rodrigues vector $\hat{r}_\tau$ and the translation vector $\hat{t}_\tau$ encode the orientation and translation of the checkerboard at time point τ. Since the checkerboard is a planar object each edge j is given by a three-dimensional point $\hat{m}_j = c_{\text{tile}}(x_j, y_j, 0)^T$ with the known length of a single tile c_{tile} and $x_j \in \mathbb{N}$ as well as $y_j \in \mathbb{N}$. Furthermore, the two-dimensional edge j in camera i at time point τ is denoted as $m_{\tau ij} \in \mathbb{R}^2$ and the delta function $\delta_{\tau ij}$ indicates whether this edge is detected successfully, i.e. $\delta_{\tau ij} = 1$, or not, i.e. $\delta_{\tau ij} = 0$.

3 Skeleton model

3.1 Modifying the skeleton model to obtain new poses

Given a three-dimensional skeleton model, we need to adjust joint locations by rotating each bone of the model, such that resulting three-dimensional positions of rigidly attached surface markers match the respective two-dimensional locations in our video data. Assuming our skeleton model has a total of n_{bone} bones and n_{marker} surface markers, we want to generate the three-dimensional locations of the joints $p \in \mathbb{R}^{n_{\text{bone}}\times 3}$ and surface markers $m \in \mathbb{R}^{n_{\text{marker}}\times 3}$, which can be obtained according to Algorithm 1.

Algorithm 1

```
function $\mathrm{f}_{\text{pose}}(t, r, l, v)$
    for $j \in \{1, ..., n_{\text{bone}}\}$ do
        $R_j \leftarrow I$                                    ▷ Initialize each bone rotation $R_j$
    for $i \in \{1, ..., n_{\text{bone}}\}$ do
        for $j \in \{1, ..., n_{\text{bone}}\}$ do
            if $j_1$ is child of $i_0$ then                   ▷ Check if rotation of bone $i$ affects end joint $j_1$
                $R_j \leftarrow \mathrm{f}_{\text{r}\to\text{R}}\left(r_i\right)^T R_j$      ▷ Update rotation of bone $j$
    for $j \in \{1, ..., n_{\text{bone}}\}$ do
        $R_j \leftarrow {R_j}^T \bar{R}_j$                    ▷ Apply bone rotation $R_j$ to resting pose $\bar{R}_j$
    $p_{1_0} \leftarrow t$                                    ▷ Initialize root joint location $p_{1_0}$
    for $j \in \{1, ..., n_{\text{bone}}\}$ do
        $p_{j_1} \leftarrow p_{j_0} + \left(R_{j_{13}}, R_{j_{23}}, R_{j_{33}}\right)^T l_j$   ▷ Calculate end joint location $p_{j_1}$ of bone $j$
        for $k \in \{1, ..., n_{\text{marker}}\}$ do
            if $j_1$ is connected to $k$ then                 ▷ Check if end joint $j_1$ is connected to marker $k$
                $m_k \leftarrow p_{j_1} + R_j v_k$            ▷ Calculate absolute marker location $m_k$
    return $m$
```

Here, it is assumed that the set $\{1, ..., n_{\text{bone}}\}$ is sorted, such that one iterates through the skeleton graph beginning with the bone whose start joint is the root joint 1_0 and then proceed with the bones further down the skeleton graph. Thus, it is always guaranteed that for $j > i$, the start joint i_0 of bone i is never a child of the start joint j_0 of bone j. It is also assumed that the bone coordinate systems of the skeleton model are constructed such that their z-directions encode the directions in which the respective bones are pointing. Furthermore, the global translation vector $t \in \mathbb{R}^3$ corresponds to the three-dimensional location of the skeleton's root joint, the rows of the tensor $r \in \mathbb{R}^{n_{\text{bone}} \times 3}$ contain Rodrigues vectors encoding the bone rotations, the vector $l \in \mathbb{R}^{n_{\text{bone}}}$ contains the bone lengths and the rows of the tensor $v \in \mathbb{R}^{n_{\text{marker}} \times 3}$ contain the relative maker locations, i.e. the locations of the markers when the position of the attached joints are assumed to be the origin. The resting pose $\bar{R} \in \mathbb{R}^{n_{\text{bone}} \times 3 \times 3}$ of the animal describes the orientation of the bones when no additional rotations are applied, i.e. $r_i = (0, 0, 0)^T \; \forall \; i \in \{1, ..., n_{\text{bone}}\}$. Here, the frequent usage of the transpose operation allows to first rotate bones, which are the closest to the leaf joints of the skeleton graph [4]. This has the advantage that we can enforce constraints on bone rotations with reference to a global coordinate system that corresponds to the three main axes of the animal's body. Assume we only model a single front limb where we only have rotations around the shoulder, elbow and wrist, i.e. R_{shoulder}, R_{elbow} and R_{wrist}, and would like to obtain the new orientation R_{new} of the bone whose start joint is identical to the animal's wrist given its resting pose $\bar{R}_{\text{wrist}}$ while iterating through the skeleton graph starting from the root joint, i.e. the shoulder. Then we can obtain R_{new} according to

$$R_{\text{new}} = \left({R_{\text{wrist}}}^T {R_{\text{elbow}}}^T {R_{\text{shoulder}}}^T \right)^T \bar{R}_{\text{wrist}} = R_{\text{shoulder}} R_{\text{elbow}} R_{\text{wrist}} \bar{R}_{\text{wrist}} \tag{8}$$

Thus, we can iterate through the skeleton graph from the root to the leaf joints but actually apply the respective bone rotations in the reversed order.

3.2 Inferring bone lengths and surface marker positions

Reconstructing poses for n_{time} time points can be archived equivalently to the calibration of a multi-camera setup as discussed in Section 2.2, i.e. we need to minimize a respective objective function via gradient decent optimization using the L-BFGS-B algorithm [3]:

$$\underset{\substack{t_\tau, r_\tau, l, v \\ \forall \tau \in \{1, ..., n_{\text{time}}\}}}{\arg\min} \sum_{\tau=1}^{n_{\text{time}}} \sum_{i=1}^{n_{\text{cam}}} \sum_{j=1}^{n_{\text{marker}}} \delta_{\tau ij} ||m_{\tau ij} - \hat{m}_{\tau ij}||^2 \tag{9}$$

where $m_{\tau ij}$ is the two-dimensional location of marker j in camera i at time point τ and $\delta_{\tau ij}$ indicates whether this marker location was successfully detected, i.e. $\delta_{\tau ij} = 1$, or not, i.e. $\delta_{\tau ij} = 0$. The corresponding projected two-dimensional marker location $\hat{m}_{\tau ij}$ can be obtained by propagating the absolute marker positions calculated via Algorithm 1 through the projection function $\mathrm{f}_{\text{3D}\to\text{2D}}$:

$$\hat{m}_{\tau ij} = \mathrm{f}_{\text{3D}\to\text{2D}} \left(\mathrm{f}_{\text{pose}} \left(t_\tau, r_\tau, l, v \right)_j, \tilde{r}_i, \tilde{t}_i, \tilde{k}_i, \tilde{A}_i \right) \tag{10}$$

where $t_\tau \in \mathbb{R}^3$ and $r_\tau \in \mathbb{R}^{n_{\text{bone}} \times 3}$ denote the translation vector and the bone rotations at time point τ. Note how there is a set of pose-encoding parameters t_τ and r_τ for each time point τ whereas the bone lengths l and the relative surface marker positions v, which encode the animal's skeletal structure and configuration, are shared across all time points. Thus, if we provide enough time points where the animal is visible in many different poses, which ideally cover the entire spectrum of the animal's behavioral space, we can not only reconstruct the pose of the animal for the given time points but are also able to learn the structure of the animal's skeleton, by inferring the unknown parameters l and v.

3.3 Scaling of input and output variables

In general, we always scale the translation vector t and the bone rotations r as well as the resulting two-dimensional marker locations $\hat{m}$, such that all of them roughly lie within the same range, i.e.

$[-1, 1]$. Particularly, we define the normalization constants $c_t = 50$ cm and $c_r = \frac{\pi}{2}$ rad as well as $c_1 = 640$ px and $c_2 = 512$ px, which we use to normalize r and t as well as $\hat{m}$. The choice for c_t was based on the dimensions of the largest arena we used in our experiments, where the maximum distance to an arena's edge from the origin of the world coordinate system, located at the center of the arena, was around 50 cm. The choice for c_r was based on the maximum bone rotation of the naively constrained spine and tail joints in our skeleton model, which was equal to $\frac{\pi}{2}$ rad. The choice for c_1 and c_2 were based on the sensor sizes of the cameras we used in our experiments, which were all equal to 1280×1024 px^2. Using the normalization constants we obtain the normalized translation vector $t^* = \frac{t}{c_t}$ and the normalized bone rotations $r^* = \frac{c_r}{r}$ as well as the normalized two-dimensional marker locations

$$\hat{m}^* = \begin{pmatrix} \hat{m}_1^* \\ \hat{m}_2^* \end{pmatrix} = \begin{pmatrix} \frac{\hat{m}_1}{c_1} - 1 \\ \frac{\hat{m}_2}{c_2} - 1 \end{pmatrix} \tag{11}$$

for a single two dimensional marker location $\hat{m} \in \mathbb{R}^2$, such that $\hat{m}_1$ represents its x- and $\hat{m}_2$ its y-coordinate. These normalized variables were used instead of their non-normalized counterparts in all depicted optimization and pose reconstruction steps.

3.4 Enforcing body symmetry

To improve the inference of bone lengths and surface marker positions we took advantage of the symmetric properties of an animal's body, i.e. for every left-sided limb there exists a corresponding limb on the right side. Furthermore, we also placed the surface markers onto the animal's fur, such that the marker-pattern itself was symmetrical, e.g. for a marker that was placed to a position close to the left hip joint there was a corresponding marker on the right side of the animal. By incorporating this knowledge into Algorithm 1 we reduced the number of free parameters, i.e. we only optimized the reduced bone lengths $l^* \in \mathbb{R}^{n^*_{\text{bone}}}$ and relative marker positions $v^* \in \mathbb{R}^{n^*_{\text{marker}} \times 3}$, where n^*_{bone} is the number of asymmetrical bones, i.e. bones along the head, spine and tail, plus the number of limb bones on the animal's left side and, equivalently, n^*_{marker} denotes the number of the asymmetrical and left-sided markers. The excluded right-sided limb bones were then enforced to have the same lengths as the corresponding limb bones on the left side. Additionally, we also applied this concept for the relative marker locations by mirroring the x-component of the left-sided markers at the yz-plane to obtain the relative marker locations of the markers on the right side. To implement this we defined Algorithm 2, which maps the reduced bone lengths l^* to the original parameter l.

Algorithm 2

```
function $\mathrm{f}_{l^* \to l}(l^*)$
    $c \leftarrow 1$                                        ▷ Initialize counter $c$ for right-sided bones
    for $i \in \{1, ..., n^*_{\text{bone}}\}$ do
        $l_i \leftarrow l^*_i$                              ▷ Set asymmetric/left-sided bone length $l_i$
        if $i$ is left-sided bone then                      ▷ Check if bone $i$ is on the left side
            $l_{n^*_{\text{bone}}+c} \leftarrow l^*_i$      ▷ Set right-sided bone length $l_{n^*_{\text{bone}}+c}$
            $c \leftarrow c + 1$                            ▷ Increase counter $c$ for right-sided bones
    return $l$
```

Equivalently, we also defined the corresponding Algorithm 3, which maps the reduced relative marker positions v^* to their original counterpart v.

Algorithm 3

```
function $\mathrm{f}_{v^* \to v}(v^*)$
    $c \leftarrow 1$                                              ▷ Initialize counter $c$ for right-sided markers
    for $j \in \{1, ..., n^*_{\text{marker}}\}$ do
        $v_j \leftarrow v^*_j$                                    ▷ Set asymmetric/left-sided rel. marker position $v_j$
        if $j$ is left-sided marker then                          ▷ Check if marker $j$ is on the left side
            $v_{n^*_{\text{marker}}+c} \leftarrow \left(-v^*_{j\,1}, v^*_{j\,2}, v^*_{j\,3}\right)^T$   ▷ Set right-sided rel. marker position $v_{n^*_{\text{marker}}+c}$
            $c \leftarrow c + 1$                                  ▷ Increase counter $c$ for right-sided markers
    return $v$
```

To learn the underlying three-dimensional skeleton model while also enforcing body symmetry, we then redefined $\hat{m}_{\tau ij}$ from equation 10 as follows:

$$\hat{m}_{\tau ij} = \mathrm{f}_{\text{3D}\to\text{2D}} \left(\mathrm{f}_{\text{pose}} \left(t_\tau, r_\tau, \mathrm{f}_{\mathrm{l}^* \to \mathrm{l}} \left(l^* \right), \mathrm{f}_{\mathrm{v}^* \to \mathrm{v}} \left(v^* \right) \right)_j, \tilde{r}_i, \tilde{t}_i, \tilde{k}_i, \tilde{A}_i \right) \tag{12}$$

and minimized equation 9 with respect to the parameters l^* and v^* instead of l and v.

4 Probabilistic pose estimation

4.1 Using a state space model to describe behavioral time series'

To allow for probabilistic pose reconstruction of entire behavioral sequences of length T, which ensures that poses of consecutive time points are similar to each other, we deploy a state space model, given by a transition and an emission equation

$$z_t = z_{t-1} + \epsilon_z \tag{13}$$

$$x_t = \mathrm{g}\left(z_t\right) + \epsilon_x \tag{14}$$

where at time point $t \in \{1, ..., T\}$ the state variable $z_t \in \mathbb{R}^{n_z}$ encodes the position of the animal as well as the bone rotations and the measurement variable $x_t \in \mathbb{R}^{n_x}$ represents the two-dimensional surface marker locations in all cameras given by a trained neural network. Thus, the state variable z_t contains the global translation vector t as well as the pose-encoding tensor r for time point t and the measurement variable x_t is a constant quantity given for all time points t. The function g, given by Algorithm 4, computes the noise-free measurements of the two-dimensional surface marker locations x^*_t given the state variable z_t. At this point the bone lengths l and relative maker locations v are already inferred and therefore given. The same applies to the Rodrigues vector $\tilde{r}_i$, the translation vector $\tilde{t}_i$, the distortion vector $\tilde{k}_i$ and the camera matrix $\tilde{A}_i$ of camera i, which we obtained from calibrating the multi-camera setup. The normalization constants c_t, c_r as well as c_1 and c_2 are the same as in Section 3.3. The probabilistic nature of the model is given by incorporating the two normally distributed random variables $\epsilon_z \sim \mathcal{N}\left(0, V_z\right)$ and $\epsilon_x \sim \mathcal{N}\left(0, V_x\right)$, simulating small pose changes over time and measurement noise, as well as the initial state $z_0 \sim \mathcal{N}\left(\mu_0, V_0\right)$, which is also assumed to be a normally distributed random variable. Thus, the state space model is entirely described by the model parameters $\Theta = \{\mu_0, V_0, V_z, V_x\}$. This allows for inferring a set of expected state variables $z = \{z_1, ..., z_T\}$ given our measurements $x = \{x_1, ..., x_T\}$ in case we have a good estimate for the model parameters Θ. Alternatively, we are also able to calculate a set of model parameters Θ, which, given an estimate for the state variables z, maximizes a lower bound of the model's evidence, i.e. the evidence lower bound (ELBO). The former is equivalent to the expectation step (E-step) of the expectation-maximization (EM) algorithm, which can be performed by applying the unscented Rauch-Tung-Striebel (RTS) smoother, whereas the latter is identical to the algorithm's maximization step (M-step), in which new model parameters are calculated in closed form to maximize the ELBO [8].

Algorithm 4

```
function $g(z_t)$
    $t \leftarrow c_t(z_{t1}, z_{t2}, z_{t3})^T$                              ▷ Obtain global translation $t$
    $r_0 \leftarrow c_r(z_{t3}, z_{t4}, z_{t5})^T$                            ▷ Obtain global translation $r_0$
    for $i \in \{1, .., n_{\text{bone}}\}$ do
        $k \leftarrow 3\,(i+1)$                                               ▷ Calculate correct index $k$
        $r_i \leftarrow (z_{tk}, z_{tk+1}, z_{tk+2})^T$                       ▷ Obtain bone rotation $r_i$
    $m_{3\text{D}} \leftarrow \mathrm{f}_{\text{pose}}\,(t, r, l, v)$          ▷ Obtain 3D marker locations given $l$ and $v$
    for $i \in \{1, .., n_{\text{cam}}\}$ do
        for $j \in \{1, .., n_{\text{marker}}\}$ do
            $k \leftarrow 2n_{\text{marker}}\,(i-1)+j$                        ▷ Calculate correct index $k$
            $m_{2\text{D}} \leftarrow \mathrm{f}_{3\text{D}\rightarrow 2\text{D}}\left(m_{3\text{D}j}, \tilde{r}_i, \tilde{t}_i, \tilde{k}_i, \tilde{A}_i\right)$   ▷ Obtain 2D marker locations given $\tilde{r}$, $\tilde{t}$, $\tilde{k}$ and $\tilde{A}$
            $x^*_{t\,k} \leftarrow \frac{m_{2\text{D}1}}{c_1} - 1$            ▷ Normalize x-coordinates
            $x^*_{t\,k+n_{\text{marker}}} \leftarrow \frac{m_{2\text{D}2}}{c_2} - 1$   ▷ Normalize y-coordinates
    return $x^*_t$
```

4.2 Theory of the expectation-maximization algorithm

While the EM algorithm was first introduced by Dempster et al. [5], we follow the concepts and notations stated by Bishop [1] and Murphy [9]. To derive a formulation of the ELBO we first note that the model's joint distribution $\mathrm{p}\,(x, z)$ is equal to the product of the model's likelihood $\mathrm{p}\,(x|z)$ and prior $\mathrm{p}\,(z)$:

$$\mathrm{p}\,(x, z) = \mathrm{p}\,(x|z)\,\mathrm{p}\,(z)\,. \tag{15}$$

Additionally, we also note that the mutual dependency of the model's marginal likelihood $\mathrm{p}\,(x)$, posterior $\mathrm{p}\,(z|x)$, likelihood $\mathrm{p}\,(x|z)$ and prior $\mathrm{p}\,(z)$ is given by Bayes' theorem:

$$\mathrm{p}\,(z|x)\,\mathrm{p}\,(x) = \mathrm{p}\,(x|z)\,\mathrm{p}\,(z)\,. \tag{16}$$

We now define an arbitrary probability density function $\mathrm{q}\,(z)$ over our state variables z, for which we know the following statement is true by definition:

$$\int \mathrm{q}\,(z)\,dz = 1. \tag{17}$$

Multiplying equation 17 with an arbitrary constant c yields:

$$c\int \mathrm{q}\,(z)\,dz = \int c\,\mathrm{q}\,(z)\,dz = c. \tag{18}$$

We can now replace the constant c with a function independent from the state variables z without loss of generality. If we choose this function to be the model's marginal log-likelihood $\ln \mathrm{p}\,(x)$, we obtain:

$$\int \mathrm{q}\,(z) \ln \mathrm{p}\,(x)\,dz = \ln \mathrm{p}\,(x) \tag{19}$$

and note that, due to equation 17 and 18 respectively, the marginal log-likelihood $\ln \mathrm{p}\,(x)$ is actually independent of the probability density function $\mathrm{q}\,(z)$. Next, we can use equation 15 and 16 to derive a relationship between the marginal log-likelihood $\ln \mathrm{p}\,(x)$, the Kullback–Leibler (KL) divergence

$\mathrm{KL}\left(q||p\right)$ and the ELBO $\mathcal{L}$, starting from equation 19:

$$\begin{aligned}
\ln \mathrm{p}\left(x\right) &= \int \mathrm{q}\left(z\right) \ln \mathrm{p}\left(x\right) dz && (20)\\
&= \int \mathrm{q}\left(z\right) \ln \frac{\mathrm{p}\left(z|x\right)\mathrm{p}\left(x\right)}{\mathrm{p}\left(z|x\right)} dz && (21)\\
&= \int \mathrm{q}\left(z\right) \ln \frac{\mathrm{p}\left(x|z\right)\mathrm{p}\left(z\right)}{\mathrm{p}\left(z|x\right)} dz && (22)\\
&= \int \mathrm{q}\left(z\right) \ln \frac{\mathrm{p}\left(x,z\right)}{\mathrm{p}\left(z|x\right)} dz && (23)\\
&= \int \mathrm{q}\left(z\right) \ln \frac{\mathrm{p}\left(x,z\right)\mathrm{q}\left(z\right)}{\mathrm{p}\left(z|x\right)\mathrm{q}\left(z\right)} dz && (24)\\
&= \int \mathrm{q}\left(z\right) \left(\ln \frac{\mathrm{p}\left(x,z\right)}{\mathrm{q}\left(z\right)} - \ln \frac{\mathrm{p}\left(z|x\right)}{\mathrm{q}\left(z\right)} \right) dz && (25)\\
&= \int \mathrm{q}\left(z\right) \ln \frac{\mathrm{p}\left(x,z\right)}{\mathrm{q}\left(z\right)} dz - \int \mathrm{q}\left(z\right) \ln \frac{\mathrm{p}\left(z|x\right)}{\mathrm{q}\left(z\right)} dz && (26)\\
&= \mathcal{L} + \mathrm{KL}\left(q||p\right) && (27)
\end{aligned}$$

with $\mathcal{L} = \int \mathrm{q}\left(z\right) \ln \frac{\mathrm{p}(x,z)}{\mathrm{q}(z)} dz$ and $\mathrm{KL}\left(q||p\right) = -\int \mathrm{q}\left(z\right) \ln \frac{\mathrm{p}(z|x)}{\mathrm{q}(z)} dz$. The KL divergence is a distance measure between the probability density functions q and p and as such always larger or equal to zero:

$$\mathrm{KL}\left(q||p\right) \geq 0 \tag{28}$$

with equality $\mathrm{KL}\left(q||p\right) = 0$ if $q = p$. When we add the ELBO $\mathcal{L}$ to equation 28 and combine the result with the derived definition of $\ln \mathrm{p}\left(x\right)$, it becomes clear that the ELBO $\mathcal{L}$ is a lower bound of the marginal log-likelihood:

$$\ln \mathrm{p}\left(x\right) = \mathcal{L} + \mathrm{KL}\left(q||p\right) \geq \mathcal{L}. \tag{29}$$

If we now acknowledge that we also require the model parameters Θ to compute the above quantities, i.e.

$$\begin{aligned}
\ln \mathrm{p}\left(x|\Theta\right) &= \mathcal{L}\left(\mathrm{q},\Theta\right) + \mathrm{KL}\left(q||p\right) && (30)\\
&= \int \mathrm{q}\left(z\right) \ln \frac{\mathrm{p}\left(x,z|\Theta\right)}{\mathrm{q}\left(z\right)} dz - \int \mathrm{q}\left(z\right) \ln \frac{\mathrm{p}\left(z|x,\Theta\right)}{\mathrm{q}\left(z\right)} dz && (31)\\
&= \left(\int \mathrm{q}\left(z\right) \ln \mathrm{p}\left(x|\Theta\right) dz + \int \mathrm{q}\left(z\right) \ln \frac{\mathrm{p}\left(z|x,\Theta\right)}{\mathrm{q}\left(z\right)} dz \right) - \int \mathrm{q}\left(z\right) \ln \frac{\mathrm{p}\left(z|x,\Theta\right)}{\mathrm{q}\left(z\right)} dz && (32)\\
&\geq \mathcal{L}\left(\mathrm{q},\Theta\right) && (33)\\
&= \int \mathrm{q}\left(z\right) \ln \mathrm{p}\left(x|\Theta\right) dz - \mathrm{KL}\left(q||p\right), && (34)
\end{aligned}$$

we can start building an understanding for how the EM algorithm works. In the E-step we are holding Θ constant and maximize $\mathcal{L}\left(\mathrm{q},\Theta\right)$ with respect to q, i.e. given a current estimate for the model parameters Θ_k we infer the probability density functions of our state variables $\mathrm{p}\left(z|x,\Theta_k\right)$, such that $\mathrm{q}\left(z\right) = \mathrm{p}\left(z|x,\Theta_k\right)$, making the KL divergence $\mathrm{KL}\left(q||p\right)$ become zero, i.e. $\mathrm{KL}\left(q||p\right) = \mathrm{KL}\left(p||p\right) = 0$, and the marginal log-likelihood $\ln \mathrm{p}\left(x|\Theta_k\right)$ become equal to the ELBO $\mathcal{L}\left(q,\Theta\right)$. Here, setting $\mathrm{q}\left(z\right) = \mathrm{p}\left(z|x,\Theta_k\right)$ maximizes the ELBO $\mathcal{L}\left(\mathrm{q},\Theta\right)$ due to the equality given by equation 34 and the previously mentioned fact that the marginal log-likelihood $\ln \mathrm{p}\left(x\right)$ is actually independent of the probability density function $\mathrm{q}\left(z\right)$. Subsequently, in the M-step we are holding q constant and maximize $\mathcal{L}\left(\mathrm{q},\Theta\right)$ with respect to Θ in order to obtain a new set of model parameters Θ_{k+1}, leading to an increased marginal log-likelihood $\ln \mathrm{p}\left(x|\Theta_{k+1}\right)$, as the KL divergence becomes greater then zero again, i.e. $\mathrm{KL}\left(q||p\right) \geq 0$

and $\mathcal{L}(\mathrm{q}, \Theta_{k+1}) \geq \mathcal{L}(\mathrm{q}, \Theta_k)$. Thus, the starting point in the M-step is the following:

$$\begin{aligned}
\ln \mathrm{p}(x|\Theta) &\geq \mathcal{L}(\mathrm{q}, \Theta) && (35)\\
&= \int \mathrm{p}(z|x, \Theta_k) \ln \frac{\mathrm{p}(x, z|\Theta)}{\mathrm{p}(z|x, \Theta_k)} dz && (36)\\
&= \int \mathrm{p}(z|x, \Theta_k) \ln \mathrm{p}(x, z|\Theta)\, dz - \int \mathrm{p}(z|x, \Theta_k) \ln \mathrm{p}(z|x, \Theta_k)\, dz && (37)\\
&= \mathcal{Q}(\Theta, \Theta_k) - \int \mathrm{p}(z|x, \Theta_k) \ln \mathrm{p}(z|x, \Theta_k)\, dz && (38)
\end{aligned}$$

with $\mathcal{Q}(\Theta, \Theta_k) = \int \mathrm{p}(z|x, \Theta_k) \ln \mathrm{p}(x, z|\Theta)\, dz$. We note that the latter term is independent of Θ and can be omitted since our goal is to optimize the ELBO $\mathcal{L}(\mathrm{q}, \Theta)$ with respect to Θ. Therefore, instead of maximizing the ELBO $\mathcal{L}(\mathrm{q}, \Theta)$ directly, we can just maximize the function $\mathcal{Q}(\Theta, \Theta_k)$. We furthermore notice that $\mathcal{Q}(\Theta, \Theta_k)$ has the form of an expectation value, i.e. we can obtain $\mathcal{Q}(\Theta, \Theta_k)$ by taking the expectation of $\ln \mathrm{p}(x, z|\Theta)$ with respect to z:

$$\mathcal{Q}(\Theta, \Theta_k) = \mathbb{E}[\ln \mathrm{p}(x, z|\Theta)] \tag{39}$$

where $\mathbb{E}[\ln \mathrm{p}(x, z|\Theta)]$ is conditioned on x and Θ_k, i.e. both quantities are given. With this we finally arrive at the essence of what is done during the M-step, i.e. maximizing $\mathcal{Q}(\Theta, \Theta_k)$ with respect to Θ to obtain new model parameters Θ_{k+1}:

$$\Theta_{k+1} = \underset{\Theta}{\arg\max}\, \mathcal{Q}(\Theta, \Theta_k) \tag{40}$$

4.3 The unscented transformation

We are required to approximate expectation values to perform the E-step, i.e. when applying the unscented Kalman filter and the unscented Kalman smoother (Algorithm 7 and 9), as well as the M-step, i.e. when maximizing $\mathcal{Q}(\Theta, \Theta_k)$ (equation 39), as we can not compute them analytically [8]. These expectation values are of the form:

$$\mathbb{E}[\mathrm{h}(y)] = \int \mathrm{p}(y)\, \mathrm{h}(y)\, dy \tag{41}$$

where h is an arbitrary function and $y \in \mathbb{R}^d$ an arbitrary normally distributed random variable, i.e. $y \sim \mathcal{N}(m, \Sigma)$. We can obtain such approximations using the unscented transformation f_{ut}, which was first introduced by Julier et al. [7] and is defined in Algorithm 5. Given the mean m and the covariance Σ, the unscented transformation f_{ut} generates so called sigma points $\mathcal{Y} \in \mathbb{R}^{2d+1 \times d}$, whose locations are systematically spread around the mean m based on the covariance Σ:

Algorithm 5

```
function $\mathrm{f}_{\mathrm{ut}}(m, \Sigma)$
    $L \leftarrow \mathrm{f}_{\mathrm{cholesky}}(\Sigma)$
    $\mathcal{Y}_1 \leftarrow m$
    for $i \in \{2, ..., d+1\}$ do
        $\mathcal{Y}_i \leftarrow m + \sqrt{d+\lambda}\, {L^T}_i$
    for $i \in \{d+2, ..., 2d+1\}$ do
        $\mathcal{Y}_i \leftarrow m - \sqrt{d+\lambda}\, {L^T}_i$
    return $\mathcal{Y}$
```

Here $\mathrm{f}_{\mathrm{cholesky}}(\Sigma)$ denotes the Cholesky decomposition of matrix Σ, which computes a lower triangular matrix L such that $LL^T = \Sigma$, and λ can be calculated as follows:

$$\lambda = \alpha^2 (d + \kappa) - d \tag{42}$$

where we set the parameters $\alpha = 1$ and $\kappa = 0$ [8], such that $\lambda = 0$. Using the sigma points $\mathcal{Y}$, we can approximate $\mathbb{E}\left[\mathrm{h}\left(y\right)\right]$ as follows:

$$\mathbb{E}\left[\mathrm{h}\left(y\right)\right] \approx \sum_{i=1}^{2d+1} w_i \,\mathrm{h}\left(\mathcal{Y}_i\right) = \sum_{i=1}^{2d+1} w_i \,\mathrm{h}\left(\mathrm{f}_{\mathrm{ut}}\left(m, \Sigma\right)_i\right) \tag{43}$$

with the weights w:

$$w_1 = \frac{\lambda}{d+\lambda} \tag{44}$$

$$w_i = \frac{1}{2\left(d+\lambda\right)} \forall i \in \{2, ..., 2d+1\}. \tag{45}$$

which due to our choice of α and κ simplifies to:

$$w_1 = 0 \tag{46}$$

$$w_i = \frac{1}{2d} \forall i \in \{2, ..., 2d+1\}. \tag{47}$$

4.4 Expectation step

In the E-step we need to infer the probability density function of the latent variable z_t for all time points t of a behavioral sequence, given the set of all measurements x and the model parameters Θ, noted as $\mathrm{p}\left(z_t|x, \Theta\right)$. Since all random variables of the model are assumed to be normally distributed, this property is maintained for the latent variable z_t as well. Therefore, z_t is drawn form a normal distribution with mean μ_t and covariance V_t, i.e. $z_t \sim \mathcal{N}\left(\mu_t, V_t\right)$. By using all measurements x of the sequence for the inference of $\mathrm{p}\left(z_t|x, \Theta\right)$ at time point t, information of the past as well as of the future is processed, which is what the unscented RTS smoother is used for. However, to derive the equations of the smoother we first need to focus on the inference when only information of the past is available, i.e. we want to infer $\mathrm{p}\left(z_t|x_1, ..., x_t, \Theta\right)$ where only measurements until time point t are given, which can be achieved by utilizing the unscented Kalman filter. To avoid confusions, we denote mean values and covariance matrices obtained from the unscented Kalman smoother as $\tilde{\mu}_t$ and $\tilde{V}_t$, whereas those calculated via the unscented RTS smoother are denoted as $\hat{\mu}_t$ and $\hat{V}_t$.

4.4.1 The unscented Kalman filter

The unscented Kalman filter is an iterative algorithm, which calculates the filtered values for the mean $\tilde{\mu}_t$ and covariance $\tilde{V}_t$ at a time point t, based on the filter output for these values $\tilde{\mu}_{t-1}$ and $\tilde{V}_{t-1}$ at the previous time point $t-1$ as well as the measurement variable x_t for time point t. The inference scheme for obtaining $\mathrm{p}\left(z_t|x_1, ..., x_{t-1}, \Theta\right)$ is given by Algorithm 6 and 7 [11, 12]:

Algorithm 6

```
function $\mathrm{f}_{\mathrm{ukf}_0}(\tilde{\mu}_{t-1}, \tilde{V}_{t-1}, V_z, V_x)$
    $\mathcal{Z} \leftarrow \mathrm{f}_{\mathrm{ut}}\left(\tilde{\mu}_{t-1}, \tilde{V}_{t-1}\right)$    ▷ Form sigma points $\mathcal{Z}$
    $\bar{z} \leftarrow \sum_{i=1}^{2n_z+1} w_i \mathcal{Z}_i$    ▷ Compute predicted mean $\bar{z}$
    $P \leftarrow V_z + \sum_{i=1}^{2n_z+1} w_i(\mathcal{Z}_i - \bar{z})(\mathcal{Z}_i - \bar{z})^T$    ▷ Compute predicted covariance $P$
    $\mathcal{Z} \leftarrow \mathrm{f}_{\mathrm{ut}}(\bar{z}, P)$    ▷ Form sigma points $\mathcal{Z}$
    $\mathcal{X} \leftarrow \mathrm{g}(\mathcal{Z})$    ▷ Propagate sigma points through emission function $\mathrm{g}$
    $\bar{x} \leftarrow \sum_{i=1}^{2n_z+1} w_i \mathcal{X}_i$    ▷ Compute predicted mean $\bar{x}$
    $S \leftarrow V_x + \sum_{i=1}^{2n_z+1} w_i(\mathcal{X}_i - \bar{x})(\mathcal{X}_i - \bar{x})^T$    ▷ Compute predicted covariance $S$
    for $i \in \{1, ..., n_x\}$ do
        if $x_{ti}$ is missing measurement then
            for $j \in \{1, ..., n_x\}$ do
                $S_{ij} \leftarrow 0$    ▷ Set rows of missing measurements to 0
                $S_{ji} \leftarrow 0$    ▷ Set columns of missing measurements to 0
            $S_{ii} \leftarrow 1$    ▷ Set diagonal entries to 1 to allow computing $S^{-1}$
    $C \leftarrow \sum_{i=1}^{2n_z+1} w_i(\mathcal{Z}_i - \bar{z})(\mathcal{X}_i - \bar{x})^T$    ▷ Compute cross-covariance $C$
    for $i \in \{1, ..., n_x\}$ do
        if $x_{ti}$ is missing measurement then
            for $j \in \{1, ..., n_z\}$ do
                $C_{ji} \leftarrow 0$    ▷ Set columns of missing measurements to 0
    $K \leftarrow CS^{-1}$    ▷ Compute filter gain $K$
    $\bar{x} \leftarrow x_t - \bar{x}$
    for $i \in \{1, ..., n_x\}$ do
        if $x_{ti}$ is missing measurement then
            $\bar{x}_i \leftarrow 0$    ▷ Set entries of missing measurements to 0
    $\tilde{\mu}_t \leftarrow \bar{z} + K\bar{x}$    ▷ Compute filtered mean $\tilde{\mu}_t$
    $\tilde{V}_t \leftarrow P - KC^T$    ▷ Compute filtered covariance $\tilde{V}_t$
    return $\tilde{\mu}_t, \tilde{V}_t$
```

To obtain values for filtered means $\tilde{\mu} = \{\tilde{\mu}_0, ..., \tilde{\mu}_T\}$ and covariances $\tilde{V} = \{\tilde{V}_0, ..., \tilde{V}_T\}$ for all time points one needs to iterate through the entire behavioral sequence:

Algorithm 7

```
function $\mathrm{f}_{\mathrm{ukf}}(\mu_0, V_0, V_z, V_x)$
    $\tilde{\mu}_0 \leftarrow \mu_0$
    $\tilde{V}_0 \leftarrow V_0$
    for $t \in \{1, ..., T\}$ do
        $\tilde{\mu}_t, \tilde{V}_t \leftarrow \mathrm{f}_{ukf_0}\left(\tilde{\mu}_{t-1}, \tilde{V}_{t-1}, V_z, V_x\right)$
    return $\tilde{\mu}, \tilde{V}$
```

4.4.2 The unscented RTS smoother

The unscented RTS smoother is also an iterative algorithm, which calculates the smoothed values for the mean $\hat{\mu}_t$ and covariance $\hat{V}_t$ at a time point t, based on the smoother output for these values $\hat{\mu}_{t+1}$ and $\hat{V}_{t+1}$ at the next time point $t+1$ as well as the corresponding output from the unscented Kalman filter $\tilde{\mu}_t$ and $\tilde{V}_t$ for time point t. The inference scheme for obtaining $\mathrm{p}(z_t|x, \Theta)$ is given by Algorithm 8 and 9 [12]:

Algorithm 8

```
function $\mathrm{f}_{\mathrm{uks}_0}(\tilde{\mu}_t, \tilde{V}_t, \hat{\mu}_{t+1}, \hat{V}_{t+1}, V_z)$
    $\mathcal{Z} \leftarrow \mathrm{f}_{\mathrm{ut}}\left(\tilde{\mu}_t, \tilde{V}_t\right)$    ▷ Form sigma points $\mathcal{Z}$ (1)
    $\bar{z} \leftarrow \sum_{i=1}^{2n_z+1} w_i \mathcal{Z}_i$    ▷ Compute predicted mean $\bar{z}$ (3.1)
    $P \leftarrow V_z + \sum_{i=1}^{2n_z+1} w_i (\mathcal{Z}_i - \bar{z})(\mathcal{Z}_i - \bar{z})^T$    ▷ Compute predicted covariance $P$ (3.2)
    $D \leftarrow \sum_{i=1}^{2n_z+1} w_i (\mathcal{Z}_i - \tilde{\mu}_t)(\mathcal{Z}_i - \bar{z})^T$    ▷ Compute cross-covariance $D$ (3.3)
    $G_t \leftarrow DP^{-1}$    ▷ Compute smoother gain $G_t$ (4.1)
    $\hat{\mu}_t \leftarrow \tilde{\mu}_t + G_t\left(\hat{\mu}_{t+1} - \bar{z}\right)$    ▷ Compute smoothed mean $\hat{\mu}_t$ (4.2)
    $\hat{V}_t \leftarrow \tilde{V}_t + \left(G_t \hat{V}_{t+1} - D\right) G_t{}^T$    ▷ Compute smoothed covariance $\hat{V}_t$ (4.3)
    return $\hat{\mu}_t, \hat{V}_t, G_t$
```

To obtain values of the smoothed means $\hat{\mu} = \{\hat{\mu}_0, ..., \hat{\mu}_T\}$ and covariances $\hat{V} = \{\hat{V}_0, ..., \hat{V}_T\}$ for all time points one needs to run the forward filtering path and then iterate backwards through the entire behavioral sequence:

Algorithm 9

```
function $\mathrm{f}_{\mathrm{uks}}(\mu_0, V_0, V_z, V_x)$
    $\tilde{\mu}, \tilde{V} \leftarrow \mathrm{f}_{ukf}\left(\mu_0, V_0, V_z, V_x\right)$
    $\hat{\mu}_T \leftarrow \tilde{\mu}_T$
    $\hat{V}_T \leftarrow \tilde{V}_T$
    for $t \in \{T-1, ..., 0\}$ do
        $\hat{\mu}_t, \hat{V}_t, G_t \leftarrow \mathrm{f}_{uks_0}\left(\tilde{\mu}_t, \tilde{V}_t, \mu_{t+1}, V_{t+1}, V_z\right)$
    return $\hat{\mu}, \hat{V}, G$
```

Here, the set of all smoother gains $G = \{G_0, ..., G_{T-1}\}$ is needed for performing the M-step later on.

4.4.3 Enforcing anatomical constraints

The plain formulation of the unscented RTS smoother does not allow constraining the state variables. However, in order to enforce joint angle limits we need to ensure that Rodrigues vectors encoding bone rotations stay within specified limits. Therefore, we introduce a mapping function $\mathrm{f}_{\mathrm{z}^* \to \mathrm{z}}$, which allows for mapping a redefined state variable $z_t^* \in \mathbb{R}^{n_z}$ onto the original one $z_t \in \mathbb{R}^{n_z}$, while enforcing that entries of z_t corresponding to bone rotations stay within their respective lower and upper bounds. The respective mapping function $\mathrm{f}_{\mathrm{z}^* \to \mathrm{z}}$ is given by Algorithm 10.

Algorithm 10

```
function $\mathrm{f}_{\mathrm{z}^* \to \mathrm{z}}(z_t^*)$
    $t^* \leftarrow (z^*_{t\,1}, z^*_{t\,2}, z^*_{t\,3})^T$    ▷ Obtain normalized global translation $t^*$
    $r_0^* \leftarrow (z^*_{t\,3}, z^*_{t\,4}, z^*_{t\,5})^T$    ▷ Obtain normalized global rotation $r_0^*$
    for $i \in \{1, .., n_{\mathrm{bone}}\}$ do
        $k \leftarrow 3\,(i+1)$    ▷ Calculate correct index $k$
        $r_i^* \leftarrow \left(z^*_{t\,k}, z^*_{t\,k+1}, z^*_{t\,k+2}\right)^T$    ▷ Obtain normalized bone rotation $r_i^*$
        for $j \in \{1, .., 3\}$ do
            $n \leftarrow \mathrm{erf}\left(\frac{\sqrt{\pi}}{2} r^*_{i\,j}\right)$    ▷ Map $r^*_{i\,j} \in (-\inf, \inf)$ to $n \in (-1, 1)$
            $r_{ij} \leftarrow b_{0ij} + \frac{1}{2}\left(b_{1ij} - b_{0ij}\right)(1+n)$    ▷ Compute $r_{ij} \in \left(b_{0ij}, b_{1ij}\right)$
    $z_t \leftarrow (t^*, r_0^*, r_1, ..., r_{n_{\mathrm{bone}}})^T$    ▷ Obtain $z_t$ via concatenation
    return $z_t$
```

Here, $b_{0,ij}$ and $b_{1,ij}$ denote the lower and upper bound corresponding to entry j of the Rodrigues vector r_i, which encodes the rotation of bone i, and erf is a sigmoidal function, i.e. the error function given by:

$$\mathrm{erf}\left(y\right) = \frac{2}{\sqrt{\pi}} \int_0^y \exp\left(-t^2\right) dt \tag{48}$$

for $y \in \mathbb{R}$. In order to enforce joint angle limits we just replace the original transmission and emission equation in our state space model given by equations 13 and 14 with:

$$z_t^* = z_{t-1}^* + \epsilon_z \tag{49}$$

$$x_t = \mathrm{g}\left(\mathrm{f}_{\mathrm{z}^* \to \mathrm{z}}\left(z_t^*\right)\right) + \epsilon_x. \tag{50}$$

In the following we always refer to the state space model given by equations 49 and 50 and therefore also to the redefined state variables z^* but we drop * in the notation.

4.5 Maximization step

In the M-step we find a new set of model parameters Θ_{k+1} by maximizing the ELBO $\mathcal{L}$, given the smoothed means $\hat{\mu}$ and covariances $\hat{V}$ as well as the smoother gains G, which we obtained in the E-step using a current estimate of the model parameters Θ_k.

4.5.1 Obtaining new model parameters by maximizing the evidence lower bound

We can take advantage of the specific structure of the state space model when maximizing the ELBO $\mathcal{L}$ [8]. In the state space model the state variables fulfill the Markov property, i.e. each state variable z_t only depends on the previous one z_{t-1}. Based on this we can compute the model's joint distribution:

$$\mathrm{p}\left(x, z\right) = \mathrm{p}\left(z_0\right) \prod_{t=1}^{T} \mathrm{p}\left(z_t | z_{t-1}\right) \mathrm{p}\left(x_t | z_t\right). \tag{51}$$

When we now take the logarithm of the joint distribution and acknowledge that the model parameters Θ are also required for computing the joint distribution we obtain:

$$\ln \mathrm{p}\left(x, z | \Theta\right) = \ln \mathrm{p}\left(z_0 | \mu_0, V_0\right) + \sum_{t=1}^{T} \ln \mathrm{p}\left(z_t | z_{t-1}, V_z\right) + \sum_{t=1}^{T} \ln \mathrm{p}\left(x_t | z_t, V_x\right). \tag{52}$$

However, to maximize $\mathcal{Q}\left(\Theta, \Theta_k\right)$ we actually need to consider the expectation value of $\ln \mathrm{p}\left(x, z | \Theta\right)$:

$$\begin{aligned} \mathcal{Q}\left(\Theta, \Theta_k\right) &= \mathbb{E}\left[\ln \mathrm{p}\left(x, z | \Theta\right)\right] & (53) \\ &= \mathbb{E}\left[\ln \mathrm{p}\left(z_0 | \mu_0, V_0\right)\right] + \sum_{t=1}^{T} \mathbb{E}\left[\ln \mathrm{p}\left(z_t | z_{t-1}, V_z\right)\right] + \sum_{t=1}^{T} \mathbb{E}\left[\ln \mathrm{p}\left(x_t | z_t, V_x\right)\right] & (54) \\ &= I_0 + I_z + I_x & (55) \end{aligned}$$

with $I_0 = \mathbb{E}\left[\ln \mathrm{p}\left(z_0 | \mu_0, V_0\right)\right]$, $I_z = \sum_{t=1}^{T} \mathbb{E}\left[\ln \mathrm{p}\left(z_t | z_{t-1}, V_z\right)\right]$ and $I_x = \sum_{t=1}^{T} \mathbb{E}\left[\ln \mathrm{p}\left(x_t | z_t, V_x\right)\right]$. If we now acknowledge that all random variables in our state space model are normally distributed, i.e. $z_t \sim \mathcal{N}\left(\hat{\mu}_t, \hat{V}_t\right)$, it becomes clear that computing $\mathcal{Q}\left(\Theta, \Theta_k\right)$ only involves evaluating the expectation values of log-transformed normal distributions (see Appendix A). Consequently, we can obtain simplified terms for the individual components I_0, I_z and I_x of $\mathcal{Q}\left(\Theta, \Theta_k\right)$ using the smoothed means $\hat{\mu}$ and covariances $\hat{V}$ as well as the smoother gains G. For I_0 we get:

$$\begin{aligned} I_0 &= -\frac{1}{2} \ln \det\left(2\pi V_0\right) - \frac{1}{2} \mathrm{tr}\left({V_0}^{-1} \mathbb{E}\left[\left(z_0 - \mu_0\right)\left(z_0 - \mu_0\right)^T\right]\right) & (56) \\ &= -\frac{1}{2} \ln \det\left(2\pi V_0\right) - \frac{1}{2} \mathrm{tr}\left({V_0}^{-1}\left(\hat{V}_0 + \left(\hat{\mu}_0 - \mu_0\right)\left(\hat{\mu}_0 - \mu_0\right)^T\right)\right). & (57) \end{aligned}$$

To obtain a simplified expression for I_z we need to form pairwise sigma points $\mathcal{P}_t$, as there are always two random variables involved simultaneously, $z_t \sim \mathcal{N}\left(\hat{\mu}_t, \hat{V}_t\right)$ and $z_{t-1} \sim \mathcal{N}\left(\hat{\mu}_{t-1}, \hat{V}_{t-1}\right)$, when evaluating the expectation values of the underlying log-transformed normal distributions in I_z. For each of the T transition steps we generate the pairwise mean vector $\check{\mu}_t \in \mathbb{R}^{2n_z}$:

$$\check{\mu}_t = \begin{pmatrix} \hat{\mu}_t \\ \hat{\mu}_{t-1} \end{pmatrix} \tag{58}$$

as well as the pairwise covariance matrix $\check{V}_t \in \mathbb{R}^{2n_z \times 2n_z}$:

$$\check{V}_t = \begin{pmatrix} \hat{V}_t & \hat{V}_t {G_{t-1}}^T \\ G_{t-1}\hat{V}_t & \hat{V}_{t-1} \end{pmatrix} \tag{59}$$

and calculate the pairwise sigma points $\mathcal{P}_t$ as follows:

$$\mathcal{P}_t = \begin{pmatrix} \mathcal{B}_t \\ \mathcal{A}_t \end{pmatrix} = \mathrm{f_{ut}}\left(\check{\mu}_t, \check{V}_t\right) \tag{60}$$

where concatenating the incomplete pairwise sigma points $\mathcal{B}_t \in \mathbb{R}^{4n_z+1\times n_z}$ and $\mathcal{A}_t \in \mathbb{R}^{4n_z+1\times n_z}$ gives $\mathcal{P}_t \in \mathbb{R}^{4n_z+1\times 2n_z}$. Consequently, the weights $\check{w}$ associated with the pairwise sigma points $\mathcal{P}_t$ are then given in accordance with the concepts discussed in Section 4.3:

$$\check{w}_1 = 0 \tag{61}$$

$$\check{w}_i = \frac{1}{4n_z} \forall i \in \{2, ..., 4n_z + 1\} \tag{62}$$

A simplified term for I_z is then given by:

$$I_z = -\frac{T}{2}\ln\det\left(2\pi V_z\right) - \frac{1}{2}\sum_{t=1}^{T}\mathrm{tr}\left({V_z}^{-1}\mathbb{E}\left[\left(z_t - z_{t-1}\right)\left(z_t - z_{t-1}\right)^T\right]\right) \tag{63}$$

$$= -\frac{T}{2}\ln\det\left(2\pi V_z\right) - \frac{T}{2}\,\mathrm{tr}\left({V_z}^{-1}\left(\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left(z_t - z_{t-1}\right)\left(z_t - z_{t-1}\right)^T\right]\right)\right) \tag{64}$$

$$\approx -\frac{T}{2}\ln\det\left(2\pi V_z\right) - \frac{T}{2}\sum_{t=1}^{T}\mathrm{tr}\left({V_z}^{-1}\left(\frac{1}{T}\sum_{i=1}^{4n_z+1}\check{w}_i\left(\mathcal{B}_{ti} - \mathcal{A}_{ti}\right)\left(\mathcal{B}_{ti} - \mathcal{A}_{ti}\right)^T\right)\right). \tag{65}$$

To evaluate the expectation value in I_x it is sufficient to just use the normal sigma points $\mathcal{Z}_t = \mathrm{f_{ut}}\left(\hat{\mu}_t, \hat{V}_t\right)$ and propagate them through our emission function g:

$$I_x = -\frac{T}{2}\ln\det\left(2\pi V_x\right) - \frac{1}{2}\sum_{t=1}^{T}\mathrm{tr}\left({V_x}^{-1}\mathbb{E}\left[\left(x_t - \mathrm{g}\left(z_t\right)\right)\left(x_t - \mathrm{g}\left(z_t\right)\right)^T\right]\right) \tag{66}$$

$$= -\frac{T}{2}\ln\det\left(2\pi V_x\right) - \frac{T}{2}\,\mathrm{tr}\left({V_x}^{-1}\left(\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left(x_t - \mathrm{g}\left(z_t\right)\right)\left(x_t - \mathrm{g}\left(z_t\right)\right)^T\right]\right)\right) \tag{67}$$

$$\approx -\frac{T}{2}\ln\det\left(2\pi V_x\right) - \frac{T}{2}\sum_{t=1}^{T}\mathrm{tr}\left({V_x}^{-1}\left(\frac{1}{T}\sum_{i=1}^{2n_z+1}w_i\left(x_t - \mathrm{g}\left(\mathcal{Z}_{ti}\right)\right)\left(x_t - \mathrm{g}\left(\mathcal{Z}_{ti}\right)\right)^T\right)\right). \tag{68}$$

To finally obtain new model parameters $\Theta_{k+1} = \{\mu_{0,k+1}, V_{0,k+1}, V_{z,k+1}, V_{x,k+1}\}$ we still need to differentiate $\mathcal{Q}\left(\Theta, \Theta_k\right)$ with respect to μ_0, V_0, V_z and V_x, set the resulting derivatives to zero and solve them for μ_0, V_0, V_z and V_x, respectively. The required derivatives have the following form (see Appendix B):

$$\frac{d}{d\mu_0}\mathcal{Q}\left(\Theta,\Theta_k\right) = \frac{d}{d\mu_0} I_0 \tag{69}$$

$$= {V_0}^{-1}\left(\hat{\mu}_0 - \mu_0\right) \tag{70}$$

$$\frac{d}{dV_0}\mathcal{Q}\left(\Theta,\Theta_k\right) = \frac{d}{dV_0} I_0 \tag{71}$$

$$= -\frac{1}{2}{V_0}^{-1} + \frac{1}{2}{V_0}^{-1}\left(\hat{V}_0 + \left(\hat{\mu}_0 - \mu_0\right)\left(\hat{\mu}_0 - \mu_0\right)^T\right){V_0}^{-1} \tag{72}$$

$$\frac{d}{dV_z}\mathcal{Q}\left(\Theta,\Theta_k\right) = \frac{d}{dV_z} I_z \tag{73}$$

$$= -\frac{T}{2}{V_z}^{-1} + \frac{T}{2}\sum_{t=1}^{T}{V_z}^{-1}\left(\frac{1}{T}\sum_{i=1}^{4n_z+1}\breve{w}_i\left(\mathcal{B}_{ti} - \mathcal{A}_{ti}\right)\left(\mathcal{B}_{ti} - \mathcal{A}_{ti}\right)^T\right){V_z}^{-1} \tag{74}$$

$$\frac{d}{dV_x}\mathcal{Q}\left(\Theta,\Theta_k\right) = \frac{d}{dV_x} I_x \tag{75}$$

$$= -\frac{T}{2}{V_x}^{-1} + \frac{T}{2}\sum_{t=1}^{T}{V_x}^{-1}\left(\frac{1}{T}\sum_{i=1}^{2n_z+1} w_i\left(x_t - \mathrm{g}\left(\mathcal{Z}_{ti}\right)\right)\left(x_t - \mathrm{g}\left(\mathcal{Z}_{ti}\right)\right)^T\right){V_x}^{-1}. \tag{76}$$

Setting these derivatives to zero and solving for μ_0, V_0, V_z and V_x yields the following:

$$\mu_0 = \hat{\mu}_0 \tag{77}$$

$$V_0 = \hat{V}_0 + \left(\hat{\mu}_0 - \mu_0\right)\left(\hat{\mu}_0 - \mu_0\right)^T \tag{78}$$

$$V_z = \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{4n_z+1}\breve{w}_i\left(\mathcal{B}_{ti} - \mathcal{A}_{ti}\right)\left(\mathcal{B}_{ti} - \mathcal{A}_{ti}\right)^T \tag{79}$$

$$V_x = \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{2n_z+1} w_i\left(x_t - \mathrm{g}\left(\mathcal{Z}_{ti}\right)\right)\left(x_t - \mathrm{g}\left(\mathcal{Z}_{ti}\right)\right)^T. \tag{80}$$

The resulting values for $\mu_{0,k+1}$, $V_{z,k+1}$ and $V_{x,k+1}$ are then given by equations 77, 79 and 80. To obtain $V_{0,k+1}$ we need to substitute $\mu_{0,k+1}$ into equation 78, giving $V_{0,k+1} = \hat{V}_0$. Lastly, we still need to adjust the solution for $V_{x,k+1}$ to also account for missing measurements. Besides that, we note that it is sufficient to only compute the diagonal entries of $V_{x,k+1}$, since we enforce the covariance matrix of the measurement noise V_x to be a diagonal matrix. Thus, the final solution for a diagonal entry $j \in \{1, ..., n_x\}$ of $V_{x,k+1}$ is given by:

$$\operatorname{diag}\left(V_{x,k+1}\right)_j = \frac{1}{T_j}\sum_{t=1}^{T}\delta_{tj}\sum_{i=1}^{2n_z+1} w_i\Big(x_{tj} - \mathrm{g}\left(\mathcal{Z}_{ti}\right)_j\Big)^2 \tag{81}$$

where the function diag gives the diagonal entries of the input matrix, δ_{tj} indicates if at time point t the entry j of x_t is associated with a missing measurement, i.e. $\delta_{tj} = 0$, or not, i.e. $\delta_{tj} = 1$, and T_j is the total number of successful measurements for entry j in the entire behavioral sequence, i.e. $T_j = \sum_{t=1}^{T}\delta_{tj}$.

4.6 Convergence of the expectation-maximization algorithm

We calculate the changes in the model parameters Θ in each iteration k of the EM algorithm to check for convergence [1]. Particularly, we are computing the vectors $\Delta\mu_0 \in \mathbb{R}^{n_z}$, $\Delta\operatorname{diag}\left(V_0\right) \in \mathbb{R}^{n_z}$, $\Delta\operatorname{diag}\left(V_z\right) \in \mathbb{R}^{n_z}$ and $\Delta\operatorname{diag}\left(V_x\right) \in \mathbb{R}^{n_x}$, which contain the relative changes of μ_0, V_0, V_z and V_x at iteration k:

$$\Delta \mu_{0i} = \operatorname{abs}\left(\frac{\mu_{0,k_i} - \mu_{0,k-1_i}}{\mu_{0,k-1_i}}\right) \ \forall\ i \in \{1, ..., n_z\} \tag{82}$$

$$\Delta \operatorname{diag}\left(V_0\right)_i = \operatorname{abs}\left(\frac{V_{0,k_{ii}} - V_{0,k-1_{ii}}}{V_{0,k-1_{ii}}}\right) \ \forall\ i \in \{1, ..., n_z\} \tag{83}$$

$$\Delta \operatorname{diag}\left(V_z\right)_i = \operatorname{abs}\left(\frac{V_{z,k_{ii}} - V_{z,k-1_{ii}}}{V_{z,k-1_{ii}}}\right) \ \forall\ i \in \{1, ..., n_z\} \tag{84}$$

$$\Delta \operatorname{diag}\left(V_x\right)_i = \operatorname{abs}\left(\frac{V_{x,k_{ii}} - V_{x,k-1_{ii}}}{V_{x,k-1_{ii}}}\right) \ \forall\ i \in \{1, ..., n_x\} \tag{85}$$

where abs is a function returning the absolute value of its input argument and $\mu_{0,k}$, $V_{0,k}$, $V_{z,k}$ and $V_{x,k}$ are the model parameters at iteration k whereas $\mu_{0,k-1}$, $V_{0,k-1}$, $V_{z,k-1}$ and $V_{x,k-1}$ are those at iteration $k-1$. We only focus on the diagonal entries of the covariances V_0 and V_z since a fraction of their off-diagonal entries is expected to be zero. Using these relative changes we construct a vector $\Delta v \in \mathbb{R}^{3n_z+n_x}$ containing all relative changes via concatenation:

$$\Delta v = \left(\Delta \mu_0, \Delta \operatorname{diag}\left(V_0\right), \Delta \operatorname{diag}\left(V_z\right), \Delta \operatorname{diag}\left(V_x\right)\right)^T \tag{86}$$

and assume convergence is reached when the mean $\Delta \bar{v}$ of Δv falls below a threshold ϵ_{tol}:

$$\Delta \bar{v} = \frac{1}{3n_z + n_x} \sum_{i=1}^{3n_z+n_x} \Delta v_i < \epsilon_{\text{tol}} \tag{87}$$

where we set $\epsilon_{\text{tol}} = 0.05$.

4.7 Implementation of the expectation-maximization algorithm

We initialize the mean of the state variables μ_0 by minimizing the objective function given by equation 9 but keep the bone lengths l and the surface marker positions v constant and set $n_{\text{time}} = 1$, i.e. we only include a single time point in the optimization, which is identical to the first time point of a respective behavioral sequence. The covariances V_0, V_x and V_z are initialized as matrices whose diagonal elements all equal 0.001 and off-diagonal entries are set to zero. To learn new model parameters μ_0, V_0, V_x and V_z we run the EM algorithm, given by Algorithm 11, with the stated initial values using measurements x obtained from the behavioral sequence. Finally, once the EM algorithm converged, we use the unscented RTS smoother with the resulting learned model parameters to reconstruct poses of the behavioral sequence.

Algorithm 11

```
function $\mathrm{f}_{\mathrm{EM}}(\mu_0, V_0, V_z, V_x)$
    $k \leftarrow 0$                                                  ▷ Initialize iteration number $k$
    $\mu_{0,k} \leftarrow \mu_0$                                      ▷ Initialize state mean
    $V_{0,k} \leftarrow V_0$                                          ▷ Initialize state covariance
    $V_{z,k} \leftarrow V_z$                                          ▷ Initialize covariance of transition noise
    $V_{x,k} \leftarrow V_x$                                          ▷ Initialize covariance of measurement noise
    $\Delta \bar{v} \leftarrow$ inf
    while $\Delta \bar{v} \geq \epsilon_{\text{tol}}$ do
        $\hat{\mu}, \hat{V}, G \leftarrow \mathrm{f}_{\mathrm{uks}}\left(\mu_{0,k}, V_{0,k}, V_{z,k}, V_{x,k}\right)$       ▷ Perform E-step
        $\mu_{0,k+1}, V_{0,k+1}, V_{z,k+1}, V_{x,k+1} \leftarrow \mathrm{f}_{\mathrm{M}}\left(\hat{\mu}, \hat{V}, G\right)$     ▷ Perform M-step
        $k \leftarrow k + 1$                                          ▷ Increase iteration number $k$
        $\Delta \bar{v} \leftarrow \mathrm{f}_{\mathrm{tol}}\left(\mu_{0,k-1}, V_{0,k-1}, V_{z,k-1}, V_{x,k-1}, \mu_{0,k}, V_{0,k}, V_{z,k}, V_{x,k}\right)$   ▷ Compute change in $\Theta$
    return $\mu_{0,k}, V_{0,k}, V_{z,k}, V_{x,k}$
```

Here, in accordance to the concepts stated in Section 4.5 and 4.6, function $\mathrm{f_M}$, given by Algorithm 12, performs the M-step and function $\mathrm{f_{tol}}$, given by Algorithm 13, computes the mean $\Delta\bar{v}$ of the relative changes of the model parameters Θ.

Algorithm 12

```
function $\mathrm{f_M}(\hat{\mu}, \hat{V}, G)$
    for $t \in \{1, ..., T\}$ do
        $\begin{pmatrix} \mathcal{B}_t \\ \mathcal{A}_t \end{pmatrix} \leftarrow \mathrm{f_{ut}}\left(\begin{pmatrix} \hat{\mu}_t \\ \hat{\mu}_{t-1} \end{pmatrix}, \begin{pmatrix} \hat{V}_t & \hat{V}_t {G_{t-1}}^T \\ G_{t-1}\hat{V}_t & \hat{V}_{t-1} \end{pmatrix}\right)$
        $\mathcal{Z}_t \leftarrow \mathrm{f_{ut}}\left(\hat{\mu}_t, \hat{V}_t\right)$
    $\mu_{0,k+1} \leftarrow \hat{\mu}_0$
    $V_{0,k+1} \leftarrow \hat{V}_0$
    $V_{z,k+1} \leftarrow \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{4n_z+1} \breve{w}_i \left(\mathcal{B}_{ti} - \mathcal{A}_{ti}\right)\left(\mathcal{B}_{ti} - \mathcal{A}_{ti}\right)^T$
    for $j \in \{1, ..., n_x\}$ do
        ${V_{x,k+1}}_{jj} \leftarrow \frac{1}{T_j}\sum_{t=1}^{T} \delta_{tj} \sum_{i=1}^{2n_z+1} w_i \left(x_{tj} - \mathrm{g}\left(\mathcal{Z}_{ti}\right)_j\right)^2$
    return $\mu_{0,k+1}, V_{0,k+1}, V_{z,k+1}, V_{x,k+1}$
```

Algorithm 13

```
function $\mathrm{f_{tol}}(\mu_{0,k-1}, V_{0,k-1}, V_{z,k-1}, V_{x,k-1}, \mu_{0,k}, V_{0,k}, V_{z,k}, V_{x,k})$
    for $i \in \{1, ..., n_z\}$ do
        $\Delta{\mu_0}_i \leftarrow \mathrm{abs}\left(\frac{{\mu_{0,k}}_i - {\mu_{0,k-1}}_i}{{\mu_{0,k-1}}_i}\right)$
        $\Delta\,\mathrm{diag}\left(V_0\right)_i \leftarrow \mathrm{abs}\left(\frac{{V_{0,k}}_{ii} - {V_{0,k-1}}_{ii}}{{V_{0,k-1}}_{ii}}\right)$
        $\Delta\,\mathrm{diag}\left(V_z\right)_i \leftarrow \mathrm{abs}\left(\frac{{V_{z,k}}_{ii} - {V_{z,k-1}}_{ii}}{{V_{z,k-1}}_{ii}}\right)$
    for $i \in \{1, ..., n_x\}$ do
        $\Delta\,\mathrm{diag}\left(V_x\right)_i \leftarrow \mathrm{abs}\left(\frac{{V_{x,k}}_{ii} - {V_{x,k-1}}_{ii}}{{V_{x,k-1}}_{ii}}\right)$
    $\Delta v \leftarrow \left(\Delta\mu_0, \Delta\,\mathrm{diag}\left(V_0\right), \Delta\,\mathrm{diag}\left(V_z\right), \Delta\,\mathrm{diag}\left(V_x\right)\right)^T$
    $\Delta\bar{v} \leftarrow \frac{1}{3n_z+n_x}\sum_{i=1}^{3n_z+n_x} \Delta v_i$
    return $\Delta\bar{v}$
```

Appendix

A Evaluating expected values of log-transformed normal distributions

Given a d-dimensional normal distribution $\mathrm{p}_{\mathrm{norm}}$ with mean $\mu_y \in \mathbb{R}^d$ and covariance $V_y \in \mathbb{R}^{d \times d}$, evaluating it for a normally distributed random variable $y \sim \mathcal{N}(m, \Sigma)$ takes the following form:

$$\mathrm{p}_{\mathrm{norm}}\left(y|\mu_y, V_y\right) = (2\pi)^{-\frac{d}{2}} \det\left(V_y\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(y-\mu_y)^T {V_y}^{-1}\left(y-\mu_y\right)\right) \tag{88}$$

where $\det\left(V_y\right) \in \mathbb{R}$ denotes the determinant of matrix V_y. Applying a logarithmic transformation yields:

$$\ln \mathrm{p}_{\mathrm{norm}}\left(y|\mu_y, V_y\right) = -\frac{1}{2}\ln\det\left(2\pi V_y\right) - \frac{1}{2}\operatorname{tr}\left({V_y}^{-1}\left(y-\mu_y\right)\left(y-\mu_y\right)^T\right) \tag{89}$$

where $\operatorname{tr}\left(V_y\right) \in \mathbb{R}$ denotes the trace of matrix V_y. Noticing that $\mathbb{E}\left[yy^T\right] = \Sigma + mm^T$ [8], we can take the expected value of equation 89 with respect to y and obtain:

$$\begin{aligned} \mathbb{E}\left[\ln \mathrm{p}_{\mathrm{norm}}\left(y|\mu_y, V_y\right)\right] &= -\frac{1}{2}\ln\det\left(2\pi V_y\right) - \frac{1}{2}\operatorname{tr}\left({V_y}^{-1}\mathbb{E}\left[\left(y-\mu_y\right)\left(y-\mu_y\right)^T\right]\right) && (90)\\ &= -\frac{1}{2}\ln\det\left(2\pi V_y\right) - \frac{1}{2}\operatorname{tr}\left({V_y}^{-1}\left(\Sigma + \left(m-\mu_y\right)\left(m-\mu_y\right)^T\right)\right) && (91) \end{aligned}$$

B Derivatives

Given a d-dimensional vector $v \in \mathbb{R}^d$, two symmetric matrices $M \in \mathbb{R}^{d \times d}$ and $C \in \mathbb{R}^{d \times d}$ as well as a scalar $c \in \mathbb{R}$, we can obtain the following derivatives:

$$\begin{aligned} \frac{d}{dv}\operatorname{tr}\left(Cvv^T\right) &= Cv + C^T v = 2Cv && (92)\\ \frac{d}{dM}\ln\det\left(cM\right) &= M^{-1} && (93)\\ \frac{d}{dM}\operatorname{tr}\left(M^{-1}C\right) &= -\left(M^T\right)^{-1}C^T\left(M^T\right)^{-1} = -M^{-1}CM^{-1}. && (94) \end{aligned}$$

References

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[2] Mary Ann Branch, Thomas F. Coleman, and Yuying Li. A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems. *SIAM Journal on Scientific Computing*, 21(1):1–23, 1999.

[3] Richard H. Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal of Scientific Computing*, 16:1190–1208, September 1995.

[4] J. Chen, S. Nie, and Q. Ji. Data-free prior model for upper body pose estimation and tracking. *IEEE Transactions on Image Processing*, 22(12):4627–4639, 2013.

[5] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[6] Intel. *The OpenCV Reference Manual*, 4.5.2 edition, 2021.

[7] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proceedings of 1995 American Control Conference - ACC'95*, volume 3, pages 1628–1632 vol.3, 1995.

[8] Juho Kokkala, Arno Solin, and Simo Särkkä. Sigma-point filtering and smoothing based parameter estimation in nonlinear dynamic systems, 2015.

[9] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[10] Gerard Pons-Moll and Bodo Rosenhahn. Ball joints for marker-less human motion capture. In *IEEE Workshop on Applications of Computer Vision (WACV)*, December 2009.

[11] Robert H. Shumway and David S. Stoffer. *Time Series Analysis and Its Applications*. Springer, 4th edition, 2017.

[12] Simo Särkkä. *Bayesian Filtering and Smoothing*. Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2013.