Supplemental Materials

5.1 Motif description and time binned evolution

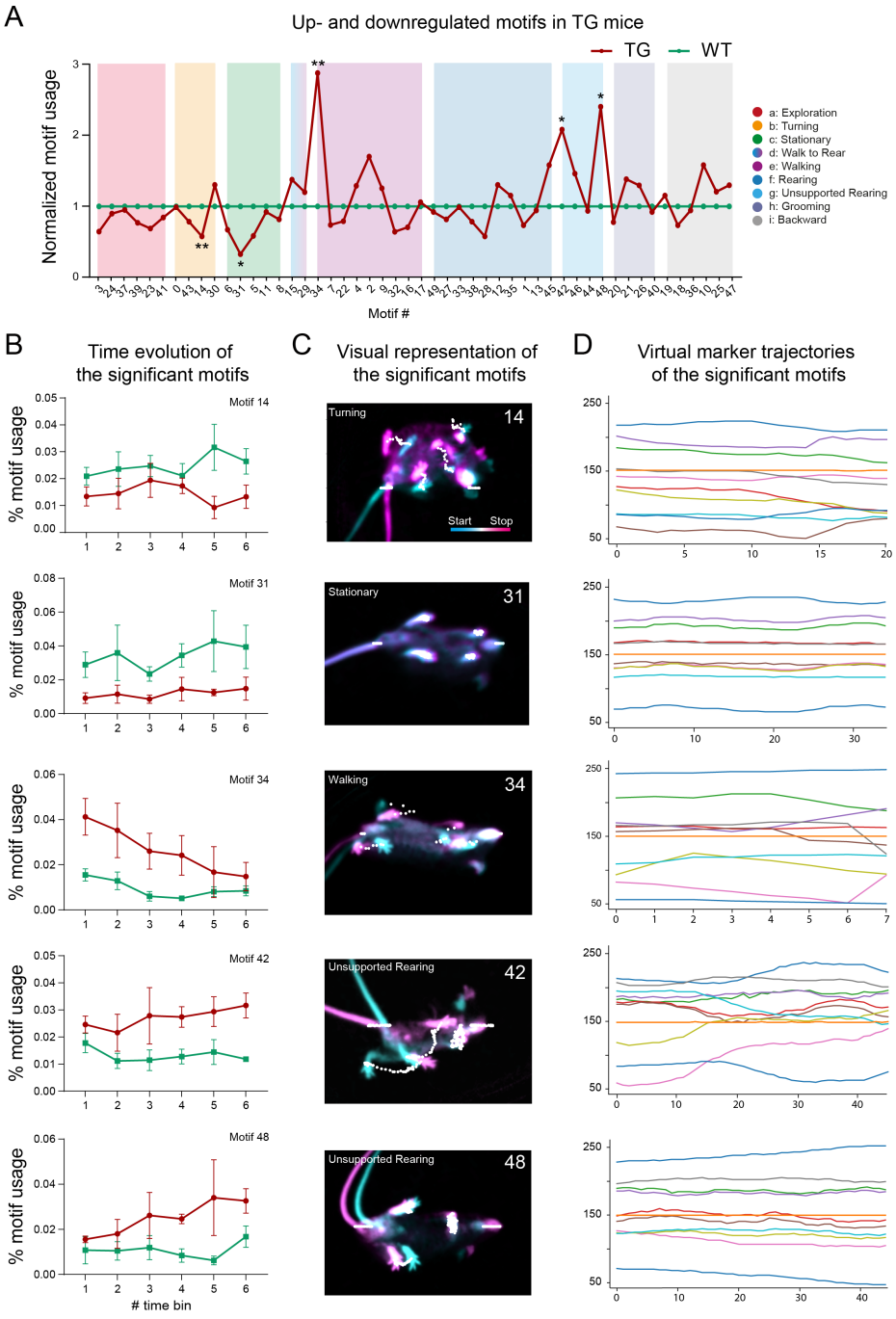

Figure S.1: **Phenotype specific and time dependent modulation of behavioral motifs.** (A) Depiction of up- and downregulation of motifs and communities in TG animals (red line) compared to WT animals (green line) and ordered by communities. (B) Binned time evolution of significant motifs between both phenotypes. (C) Visual representation of significant motifs. The start frame is colored in cyan and the end frame is colored in magenta. White dots represent the DLC virtual marker points. (D) Corresponding exemplary (x,y)-virtual marker trajectory of the visual representation shown in C.

Figure S.1 A shows an alternative way to represent behavioral differences between two groups or phenotypes. Here, the usage of a motif for the tg group is calculated as a ratio and normalized against the wt group. This representation depicts the up- or downregulation in motifs or communities with respect to the tg group. The *Exploration*, *Turning* and *Stationary* communities are downregulated, while the *Walk to Rear* and *Unsupported Rearing* communities are upregulated. In the *Walking* community, the biggest peak of upregulation happens in the significant changed motif 34. Other communities have certain motifs which are differently used but with no significant group difference.

In order to investigate how stable the significant differences are during the experiment, we binned the experiment into six equal blocks (Figure S.1 B). We find that the motif usage is constantly up- or downregulated for all significantly changed motifs. In Figure S.1 C we visualized the these motifs by taking the start (cyan color) and end (magenta color) frame for a random episode of motif occurrence. White dots are representing the DeepLabCut virtual marker positions over time. Panel D of Figure S.1 shows the corresponding DeepLabCut trajectory of the visualized motif.

To describe the motifs we investigated the single motif videos (Examples are given in supplemental video 1-5). Motif 14 (*Turning* community) is characterized by a rotational motion with the front paws further away from each other. Motif 31 (*Stationary* community) shows no paw movement but some nose motion. Motif 34 (*Walking* community) shows a push into a slow walk motion and as shown on the hierarchical tree Figure 2, C, it is grouped further away from the other walk motifs, which suggests that it does not belong to the classical walk cycle. Motif 42 and Motif 48 (*Unsupported Rearing* community) show a stationary standing animal inside the arena with some nose motion. Stationary here refers to almost no movement in the back paws.

5.2 Model comparison

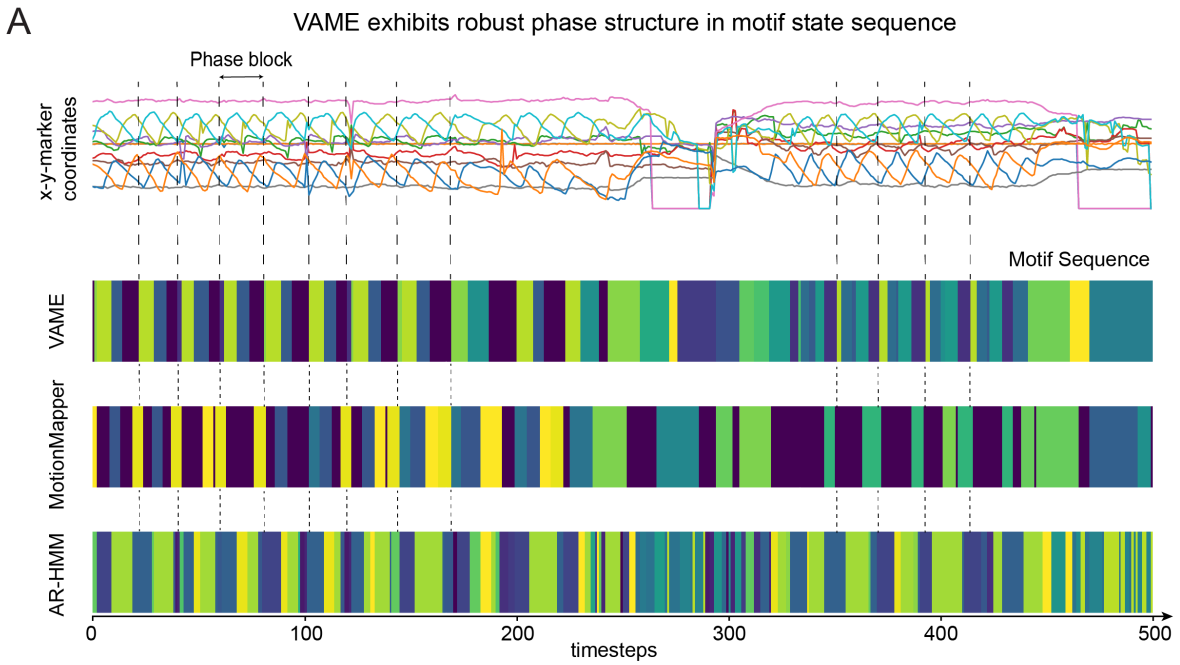

Figure S.2: **Qualitative comparison with MotionMapper and AR-HMM.** (A) Exemplary trace of the input time series together with the motif segmentations obtained from the VAME, MotionMapper and AR-HMM methods.

Figure S.2 A shows a qualitative comparison between VAME and current methods (MotionMapper and AR-HMM, (Berman et al., 2014; Wiltschko et al., 2015)). We trained all models on our data and carried out segmentation into approximately the same amount of behavioral motifs (VAME, AR-HMM n=50, MotionMapper n=51). The exemplary trace is aligned to the motif sequences (Figure 4, top) and shows two walking and rearing episodes. As reference, we selected the x-coordinate of the left hind paw (orange marker) to define a phase block, indicated by the dashed lines between time step 0 and 200 (Figure 4, top). Visual inspection between the methods revealed that VAME motifs match the phase of the signal more precisely. The motif sequence obtained from AR-HMM exhibits more rapid state switches while MotionMapper has more longer lasting motifs.

Table S.1 reports the results on Purity, NMI and Homogeneity for smaller cluster sizes of $k = \{10, 30\}$ for all three models (VAME, MotionMapper, AR-HMM) on our benchmark dataset.

Model	Purity	NMI	Homogeneity %
MotionMapper ($k = 10$)	62.14	10.91	15.14
AR-HMM ($k = 10$)	68.12	18.69	26.15
VAME ($k = 10$)	**74.20**	**29.65**	**42.00**
MotionMapper ($k = 30$)	66.37	14.13	22.98
AR-HMM ($k = 30$)	73.99	21.77	37.97
VAME ($k = 30$)	**79.19**	**26.10**	**51.16**

Table S.1: Model comparison based on an annotated dataset with $k = 10$ and $k = 30$ motifs.

5.3 Downprojection and trajectories

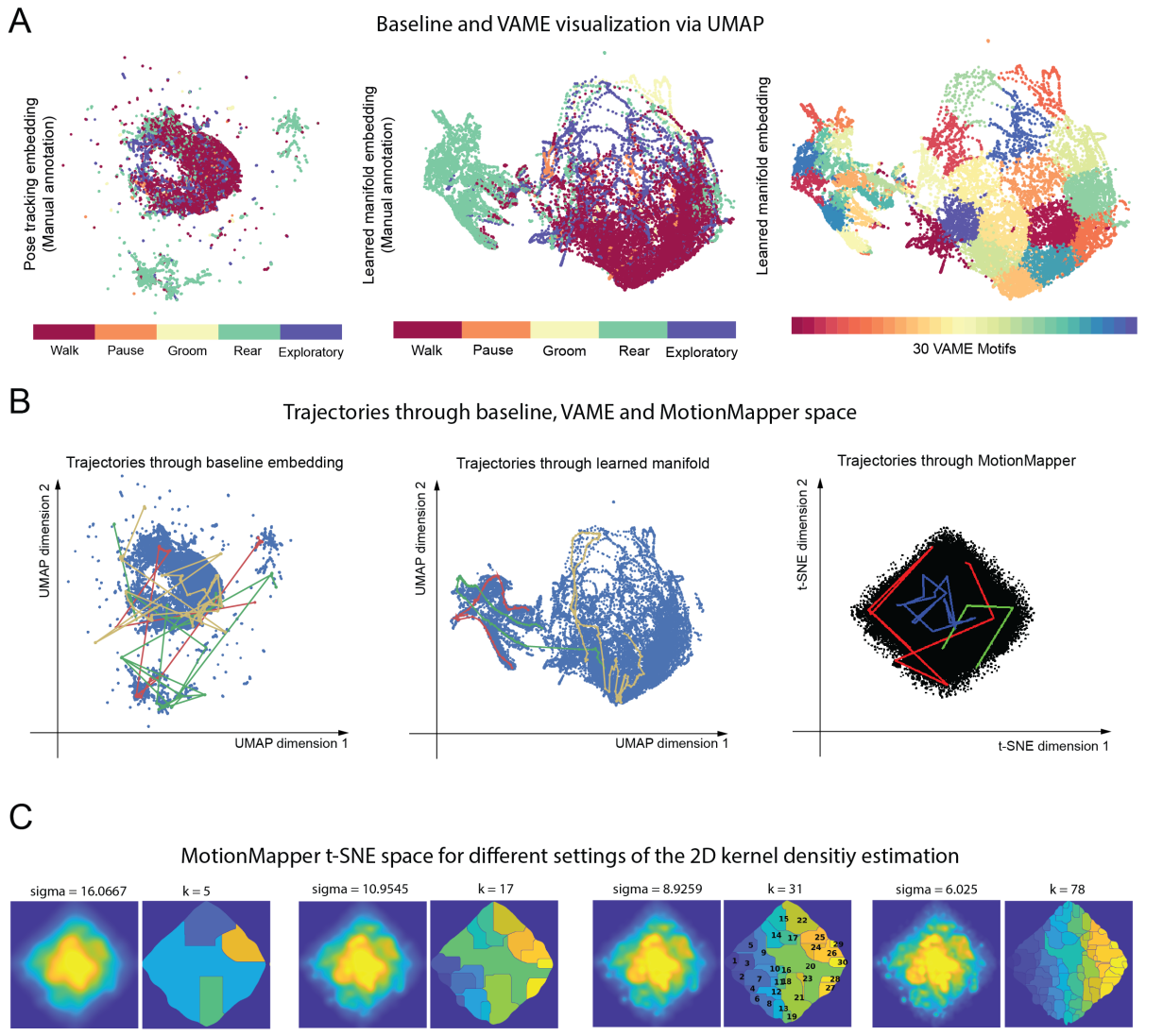

Figure S.3: **Model embedding and trajectories.** (A) Unifold Manifold Approximation (UMAP) embeddings of the marker position input series (left), the latent representation encoded from the RNN color-coded for 5 manually labeled behavioral classes (middle) and color-coded according to assignment into 30 VAME motifs (right). (B) Three exemplary paths of consecutive video frames through the UMAP embedding space of the spatial input series (left) and through the embedding space visualizing the spatiotemporal representation (Middle). (Right) t-SNE embedding obtained by MotionMapper with three examplary paths. (C) Embeddings and segmentations obtained from MotionMapper for different settings of the 2D kernel density estimation parameter (sigma).

In Figure S.3 A (left) we show the Unifold Manifold Approximation (UMAP) projection of the original egocentrically aligned virtual marker signal (baseline) for the manual annotated dataset and (middle, right) the UMAP projection of the latent vectors obtained by VAME for the same dataset (middle: human label, right: VAME motifs). It can be observed that the VAME embedding is less scattered and more densely represented.

To test the spatiotemporal connectivity of the latent representation for VAME, we sampled three random behavioral sequences with a length of 1.5 seconds and plotted their trajectories on top of both visualizations. Moreover, we also plotted this trajectory on top of the t-SNE embedding of MotionMapper as comparison. We observed that the course of trajectories within the embedding of VAME followed a smooth path through the projected manifold while trajectories

through the embedding of the baseline signal appears scattered. For MotionMapper, we also observed more scatter (Figure S.3 B).

Lastly, we investigated the embedding of MotionMapper more closely to understand the model behavior on our dataset (Figure S.3, C). More specifically, we show the effect of modifying the standard deviation of the smoothing Gaussian of the 2D kernel density estimation, a free parameter of MotionMapper, on the obtained t-SNE embedding as well as on obtained the watershed segmentations. As depicted here, the choice of the parameter has a strong impact on the clusterability; a high setting of the standard deviation implies a lower number of clusters that are detectable by the watershed segmentation, and vice-versa. Here, we observe that the overrepresented density in the middle of the embedding exists for segmentations to 5, 17 and 31 clusters, and is only resolved for a large setting of the cluster size (k=78). This suggests that the largest mass of the behavioral space is mostly contained in a single motif located in the center of the space. As discussed previously, this finding is likely observable for pose estimation data obtained from mice, as the sinusoidal signals have a higher similarity in the frequency space, when compared to behavioral signals measured for, for example, fruit flies.

5.4 VAME model selection

The VAME model consists of one biRNN encoder and two biRNN decoder. A HMM is used to identify hidden states (motifs) within our embedding space, as described in section 2.1 (also see Methods 4.3). The model was chosen after we tested four different variations of the architecture and compared the HMM against a k-Means algorith. In table S.2, we show the different choices and validated their outcome based on our benchmark dataset.

Our architectural choices were either a standard variational autoencoder consisting of a biRNN encoder and a biRNN decoder or with an additional biRNN prediction decoder. Furthermore, we applied to both variants spectral regularization of the latent space (Ma et al., 2019) to see if this could lead to improved clusterability. We applied three metrics (Purity, NMI, Homogeneity, see section 2.4) to identify the best model. In both cases (k-Means or HMM), the variational autoencoder model without spectral regularization and an additional decoder had the highest scores. The model with HMM led to the best scores and hence, we chose this as the primary model in our manuscript.

Model (segmentation)	Purity	NMI	Homogeneity %
VAME single decoder (k-Means)	74.66	22.26	43.02
VAME single decoder + spectral regularization (k-Means)	76.17	22.13	43.20
VAME two decoder (k-Means)	**76.23**	**23.58**	**45.55**
VAME two decoder + spectral regularization (k-Means)	75.56	23.12	44.69
VAME single decoder (HMM)	78.44	26.70	49.82
VAME single decoder + spectral regularization (HMM)	79.81	26.98	51.98
VAME two Decoder (HMM)	**80.66**	**28.61**	**54.85**
VAME two Decoder + spectral regularization (HMM)	79.15	27.64	52.84

Table S.2: VAME model selection with two different segmentation algorithm (k-Means and HMM) for $k = 50$. Reported is the mean of five repeated training and inference runs.

5.5 Motif clustering number

Deciding the number of cluster (e.g. motifs) k present in a data set is hard task when no a-priori knowledge is available. To determine this number for our dataset, we used a similar technique as presented in (Wiltschko et al., 2015). We took our trained VAME model and let the HMM infer 100 motifs and treated motifs that had less than 1% usage as noise. This mark was 50 motifs, which we used throughout the paper. Figure S.4 visualizes the motif usage for all eight animals (blue lines), the mean usage (red line) and the 1% threshold (dashed line). The first 10 motifs have a high usage, which then drops slowly to the 1% threshold line (48 motifs) and continues to decreae to almost 0% usage (100 motifs).

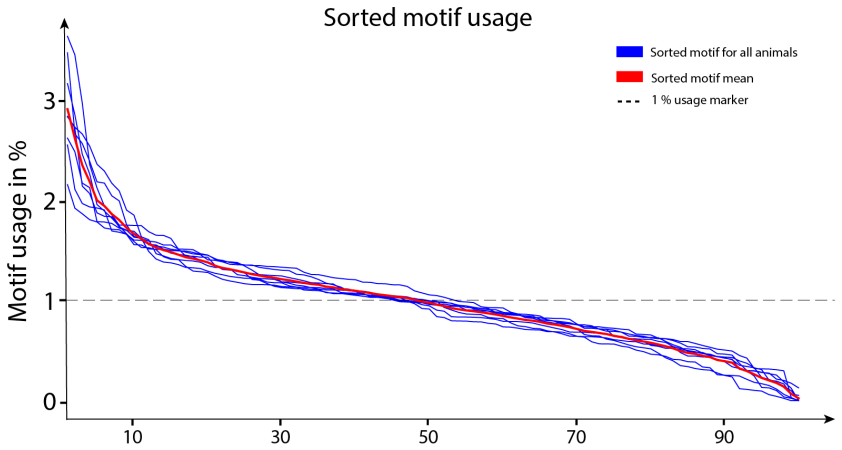

Figure S.4: **Sorted motif usage for all animals.**

5.6 Parameters and biases

Choice	Our setting
What is the pixel resolution of the animal body in the camera image	In our case, the animal body was covered with approximately 100×300 pixels.
What is the temporal resolution of the camera?	The employed camera operates at 60 Hz.
How many virtual markers are used and how are they placed on the animal body parts?	We placed 6 virtual markers on paws, the nose and the tail root (see Figure 1).
Which software is used for keypoint detection?	We used DeepLabCut 1.1 (Mathis et al., 2018).
How many frames are manually labeled?	We labeled the keypoints in 600 uniformly chosen frames.
How long is the keypoint detection CNN trained?	We trained the keypoint detection for 350.000 iterations, the resulting training errors was 2.14 pixels and the test error was 2.51 pixels.
How is the obtained keypoint time series aligned to egocentric coordinates?	We use the procedure defined in the Methods section.
How is missing data handled?	Missing data points for periods where pose estimation could not reliably detect the position of the corresponding virtual marker shorter then 200 ms were linearly interpolated while the values for longer periods were set to an arbitrary negative value.
Which size of the time window was used for the reconstruction and prediction decoder?	We used 30 frames (500 ms) for the reconstruction and 10 frames (166 ms) for the prediction decoder.
Which other hyperarameters have been used to train VAME?	All other hyperparameter settings can be found in the default configuration file available at the project website.
How many datapoints are used to train VAME?	We trained our model with 1.3 million data points.
How is the clustering carried out?	We used a Hidden Markov Model for state detection as described in the Methods section.

Table S.3: Table of choices for our approach and settings for our model.

Although the proposed quantification method is unsupervised, the choice of parameters as well as processing steps integrate biases into the quantification result. In Table S.3 we summarize the biases of our approach and state the parameter setting that have been used for our data.

5.7 Community visualization and description

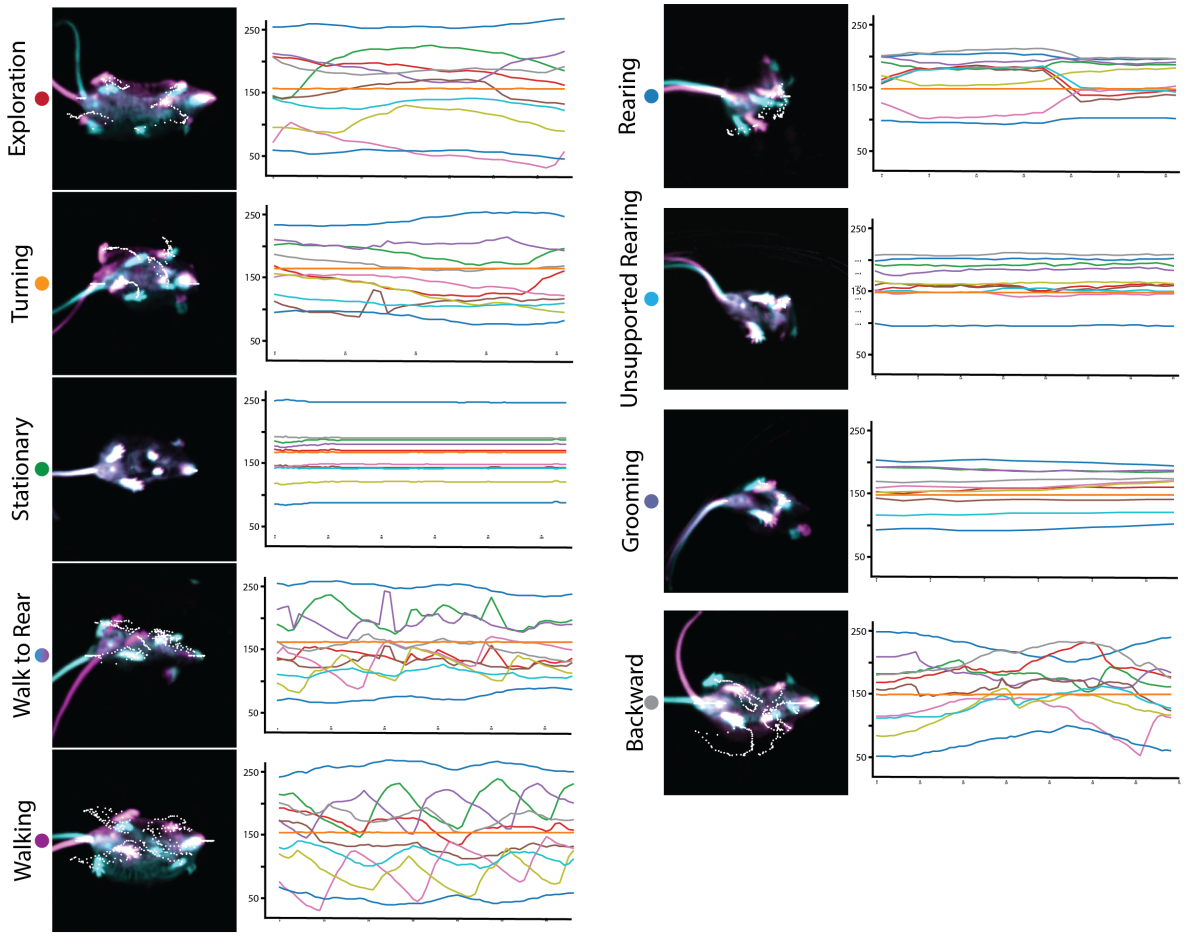

Figure S.5: **Visualization of Communities with their respective DLC trace.**

In Figure S.5 we visualized all nine communities by taking the start (cyan color) and end (magenta color) frame for a random community episode. White dots are representing DeepLabCut marker. Next to the visual representation, the DeepLabCut trace for this episode is shown. Community a contains motifs with exploration characteristics such as slow walking and a lot of nose movement which could be interpreted as sniffing. Community b shows mainly events in which motifs express rotational behavior. In c, the motifs display almost no movement of any body part. Community d consists of two motifs which depict transitional behavior from walk to rear or vice versa. In community e, we found that all motifs express a specific part of the walking behavior. Community f contains motifs which are mainly showing rears along the wall of the arena while g contains motifs depicting rears within the arena. Community h belongs to the same branch as g but portrays mainly motifs with grooming activity. Lastly, community i shows motifs in which the animal performs a backward motion e.g after rearing.

5.8 Generative model aspects

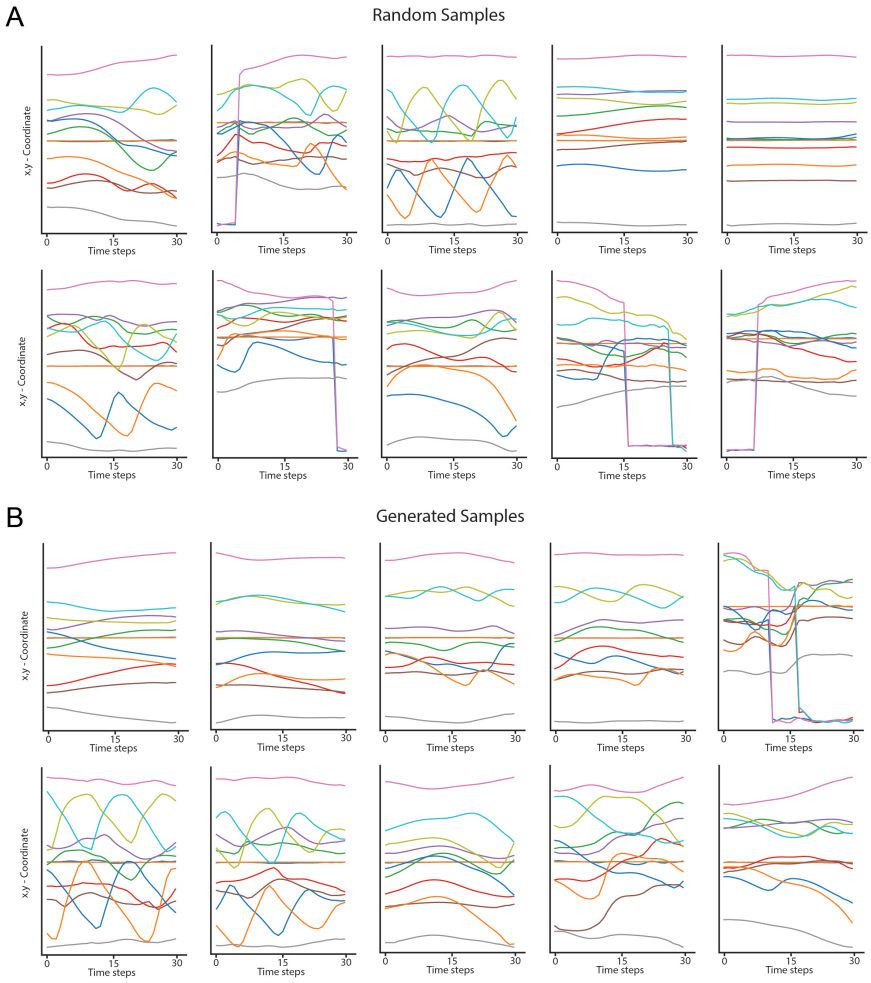

Figure S.6: **Random sample reconstruction using generative modeling** (A) Reconstruction of randomly drawn samples from the input data. (B) Generated samples obtained by fitting a Gaussian mixture model on the latent space, and decoding points sampled from the fitted density via the reconstruction decoder.

The VAME framework is based on the concept of variational autoencoder, which belongs to the class of generative models. The power of a generative models lies in its ability to learn the data distribution and create new, unseen samples from this distribution. This capability can be used to show that the model is able to represent the data distribution. In Figure S.6 A, we first show VAME's capability to reconstruct random samples form the input data. To demonstrate that we can generate accurate samples from VAME's latent space, we carried out ex-post density estimation using a 10-dimensional GMM, as proposed in (Ghosh, Sajjadi, Vergari, Black, & Scholkopf, 2020). From the fitted density, we are able to sample datapoints that can be transformed to input-traces using the decoder of the trained model. With this functionality, users of VAME can verify if the model has succesfully learned the data distribution to generate realistic synthetic samples. Note that this approach can be also used to validate single clusters, if the density is fitted on regions belonging to individual motifs only.

5.9 Sequence detection in locomotion

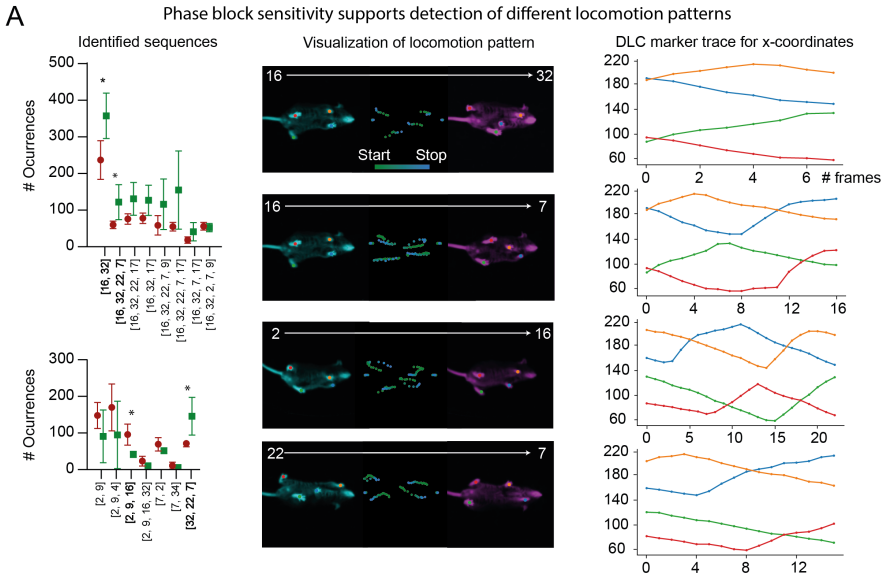

Figure S.7: **(A) Identfication of the locomotion structure in both groups (left) and visualization of four significant different patterns (middle, right).**

Given the results shown in Figure 3 C, we used the representation learned by VAME to identify locomotion patterns within the *Walking* community. Briefly, we built an algorithm that first identified reoccurring motif patterns in the community and then counts their occurrences. We detected 22 sub-patterns, which were used by all animals. Within these sub-patterns, we were able to identify four specific sub-second sequences that were significantly more used in either wt or tg mice (unpaired t-Test). The strongest sub-pattern for wt animals is the sequence [16, 32] and [32, 22, 7], which can be also seen in the difference graph as a strong transition. For the tg, the strongest pattern involves the sequence [2, 9], which again can be seen in the difference graph as a strong transition for the tg group. These results demonstrate the capabaility of VAME to robustly capture patterns for transitions between motifs, especially for locomotion behavior based on its phase sensitivity.

5.10 Community transitions

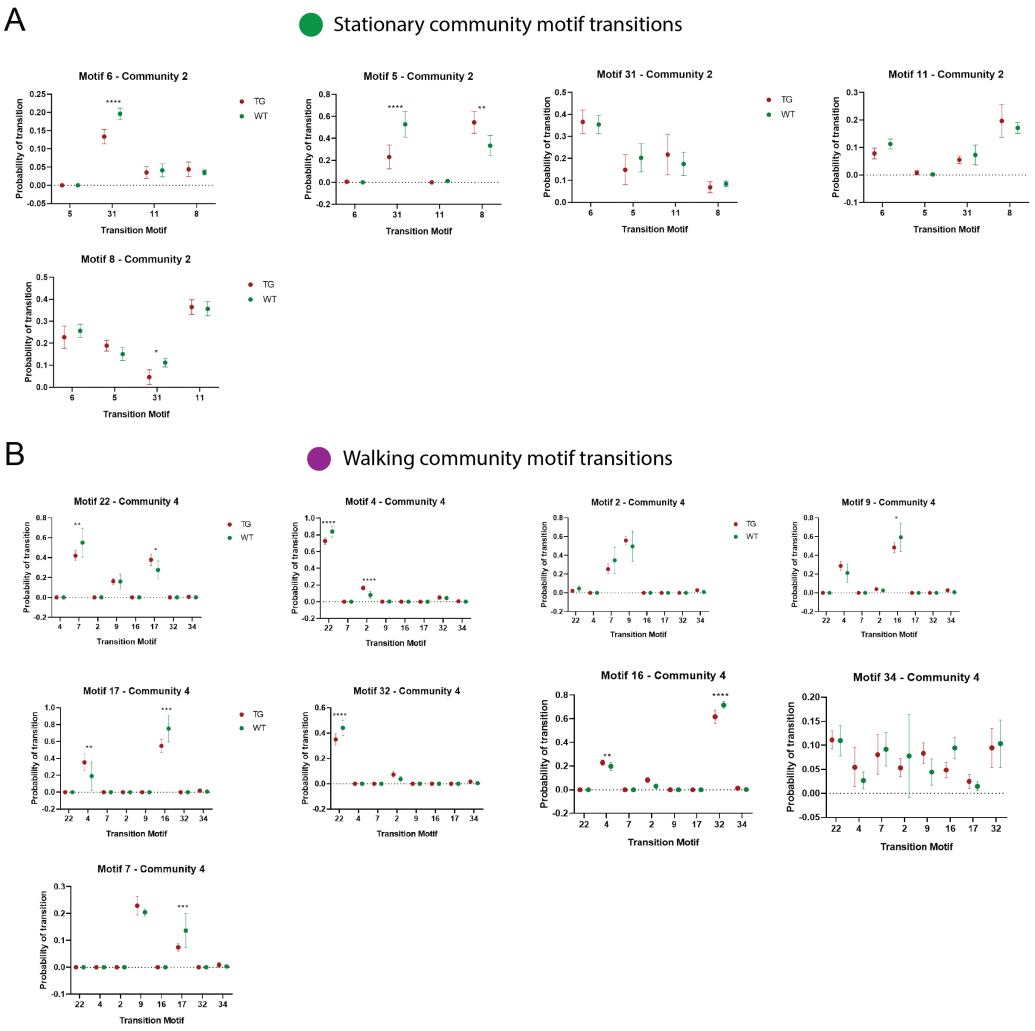

Figure S.8: **Transitions of motif within the *Stationary* and *Walking* community.**

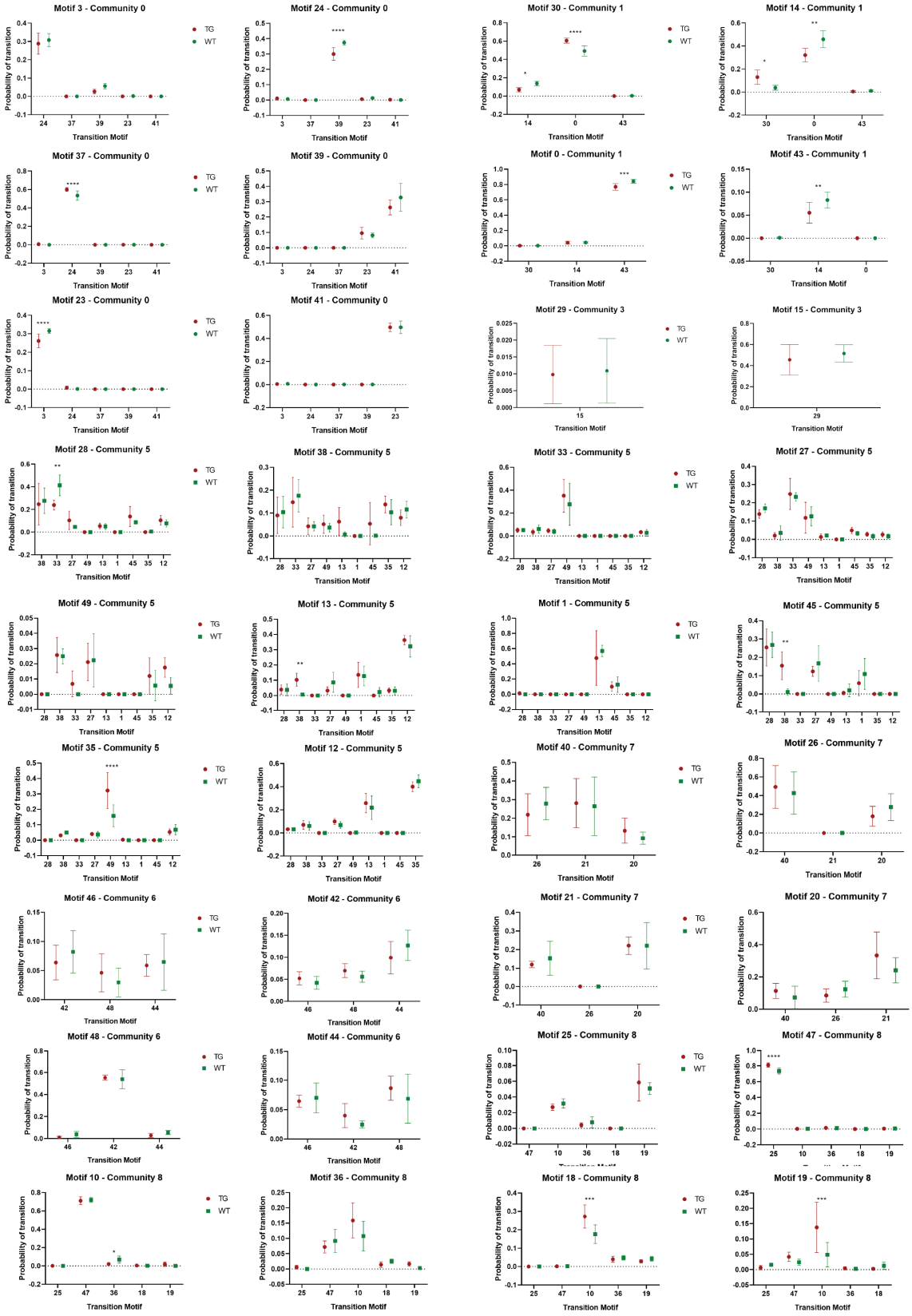

Figure S.9: **Within community transition statistics.**