Automated Behavioral Analysis of Asleep Fruit Fly

Ali Osman Berk Şapcı

July 20, 2022

Contents

List of Figures

List of Tables

Abstract

Sleep is an essential behavioral program conserved across the animal kingdom. In order to understand the underlying function of sleep, careful characterization of the changes in behavior and physiology is needed in powerful genetic model systems such as *Drosophila Melanogaster*. Recent advances in machine learning have enabled tracking of the body parts and robust pose estimation; however, automated quantification of the behaviors requires providing a mapping from a pair of spatial coordinates to behavioral categories. Existing methods and pipelines are developed with behaviors defined by major postural changes in mind. The task of detecting and successfully mapping behaviors exhibited during sleep comes with its unique challenges. Our task of phenotyping sleep requires tackling with behaviors defined by unobtrusive changes, and that sparsely occur during long sleep cycles. To this end, we develop `basty` (Automated Behavioral Analysis of Asleep Fruit Fly); a novel, end-to-end pipeline, which is made public as a configurable, open source and easy-to-use software package. Our pipeline consists of multiple stages, which can be briefly described as follows: computing meaningful behavioral representations, detecting activities in long sleep experiments (14-16 hours), and nearest neighbors analysis in a low dimensional behavioral space. We evaluate our pipeline with a dataset consisting of sleep-deprivation and wild-type sleep experiments; by considering five behavioral categories. Results show that our pipeline successfully maps behavioral categories, achieving an AUC score of 0.8, and it is able to detect unseen behaviors and differences in behavioral repertoires. Furthermore, we present a brief analysis of the behavioral repertoire exhibited during sleep by examining spatio-temporal characteristics of the behaviors and their temporal organization.

Chapter 1

Introduction

Sleep is an essential behavioral program conserved across the animal kingdom, in diverse species ranging from jellyfish to humans, whose function remains unknown [Campbell and Tobler, 1984, Nath et al., 2017]. In mammals, sleep consists of multiple stages marked by physiological changes, including reductions in muscle tone and distinct electrophysiological activity patterns in the brain [Corner, 1977, Sauer et al., 2003] In invertebrates, sleep has largely been studied as a unitary process and identified by bouts of consolidated immobility. Thus, careful characterization of underlying changes in behavior and physiology is needed for understanding the functional role of the sleep, and characterizing distinct sleep stages in powerful genetic model systems such as *Drosophila Melanogaster*.

[NOTE:

- Computational Neuroethology: A Call to Action: [Datta et al., 2019]
- Quantifying behavior to understand the brain: [Pereira, 2020]
- Toward a Science of Computational Ethology: [Anderson and Perona, 2014]

] [NOTE:

- Mapping Sub-Second Structure in Mouse Behavior
- DeepLabCut: markerless pose estimation of user-defined body parts with deep learning: [Mathis et al., 2018]
- Fast animal pose estimation using deep neural networks: [Pereira et al., 2019]
- SLEAP: A deep learning system for multi-animal pose tracking: [Pereira et al., 2022]

]

[NOTE:

- A deep sleep stage in Drosophila with a functional role in waste clearance: [van Alphen et al., 2021]
- Most sleep does not serve a vital function: Evidence from Drosophila melanogaster: [Geissmann et al., 2019]
- Automated analysis of long-term grooming behavior in Drosophila using a k-nearest neighbors classifier: [Qiao et al., 2018]
- Mapping the stereotyped behaviour of freely moving fruit flies: [Berman et al., 2014]
- Systematic exploration of unsupervised methods for mapping behavior: [Todd et al., 2017]

] [NOTE:

- Predictability and hierarchy in *Drosophila* behavior: [Berman et al., 2016]
- The manifold structure of limb coordination in walking Drosophila: [DeAngelis et al., 2019]
- The 103,200-arm acceleration dataset in the UK Biobank revealed a landscape of human sleep phenotypes: [Katori et al., 2022]

] [NOTE:

- A dictionary of behavioral motifs reveals clusters of genes affecting Caenorhabditis elegans locomotion: [Brown et al., 2013]
- Deconstructing Hunting Behavior Reveals a Tightly Coupled Stimulus-Response Loop: [Mearns et al., 2020]
- Continuous Whole-Body 3D Kinematic Recordings across the Rodent Behavioral Repertoire - SI: [Marshall et al., 2021]
- Mapping Sub-Second Structure in Mouse Behavior: [Wiltschko et al., 2015]
- The Mouse Action Recognition System (MARS) software pipeline for automated analysis of social behaviors in mice: [Segalin et al., 2021]
- Neural control of affiliative touch in prosocial interaction: [Wu et al., 2021]
- Partitioning variability in animal behavioral videos using semi-supervised variational autoencoders: [Whiteway et al., 2021]
- Structure of the Zebrafish Locomotor Repertoire Revealed with Unsupervised Behavioral Clustering: [Marques et al., 2018]

]

[NOTE:

- DeepEthogram, a machine learning pipeline for supervised behavior classification from raw pixels: [Bohnslav et al., 2021]
- B-SOiD, an open-source unsupervised algorithm for identification and fast prediction of behaviors: [Hsu and Yttri, 2021]
- Ethoscopes: An open platform for high-throughput ethomics: [Geissmann et al., 2017]
- JAABA: interactive machine learning for automatic annotation of animal behavior: [Kabra et al., 2013]
- Simple Behavioral Analysis (SimBA) – an open source toolkit for computer classification of complex social behaviors in experimental animals: [Nilsson et al., 2020]

]

Chapter 2

Experiments and Data Collection

In this section, we describe how the sleep experiments are conducted and fly video recordings data were collected.

The experimental data were collected by Dr. Mehmet Fatih Keles at Wu Lab, Johns Hopkins University.

2.1 Sleep Experiments and Video Recordings

A custom imaging setup was used to perform high-resolution characterization of sleep-related behaviors in flies. This set up includes a custom 3D printed chamber (7.2X4.3X2.4 mm [WxHxL]) that is placed in front of an IR sensitive (Flir) 30 FPS camera with telecentric lens (Edmund Optics), an illustration of the experiment setup is given in the Figure 2.1.

Experiment Name	**Experiment Type**	**Fly Gender**	**# of Frames**
FlyF1-03082020164520	wild type sleep	female	1,727,979
FlyF11-01182022175505	wild type sleep	female	1,727,979
FlyF14-08172021175459	wild type sleep	female	1,727,979
FlyF15-08182021173222	wild type sleep	female	1,727,979
FlyF8-08112021174107	wild type sleep	female	1,727,979
FlyM13-08172021175457	wild type sleep	male	1,727,979
FlyM4-03062020153616	wild type sleep	male	1,727,979
FlyF19SD-11052021164243	sleep-deprived	female	2,159,979
FlyF19SD-11152020170647	sleep-deprived	female	2,159,979
FlyF41SD-11192021170807	sleep-deprived	female	2,159,979
FlyF52SD-11242021161943	sleep-deprived	female	2,159,979

Table 2.1: Details of the collected experiment data.

Flies recorded between ZT10-ZT2 (16 hours total) for wild type sleep experiments,

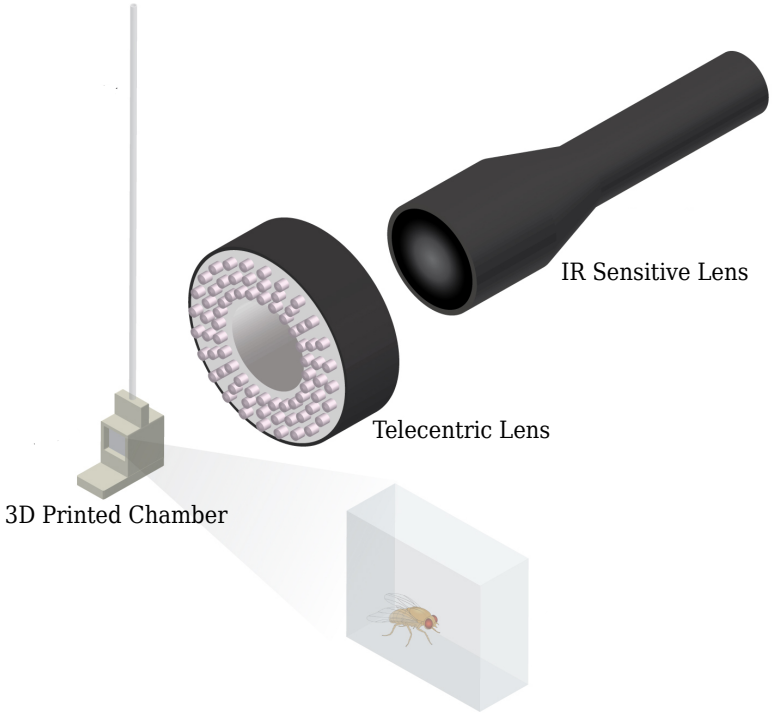


Figure 2.1: An illustration of the experimental setup which is used to perform high-resolution imaging of experiments.

and between ZT10-ZT6 (20 hours total) for sleep-deprived experiments (see Table 2.1). Each chamber has a food port (1.5 mm diameter) that allows access to liquid food (2.5% yeast, 2.5% sugar). Recording setup is in a light tight box and humidity control (60%) is achieved via a humidifier plugged into a humidity control switch. Experimental flies are loaded to individual chambers at ZT8-ZT9 via mouth pipette or a small vacuum pump. Individual chambers are sealed with a 7×7 mm acrylic windows. Windows are coated with SigmaCote to prevent flies from ventral or dorsal postural positions. 5-7 day old female and male flies are used in the experiments.

2.2 Pose Estimation and Tracking

A recently developed deep neural network based software, DeepLabCut [Mathis et al., 2018] was used to achieve markerless pose estimation. Over 20 body parts are first labeled in 1654 images from 28 animals (16 female, 12 male) to train the model with a 95/5 train/test split. An example labeled frame is given in the Figure 2.2. We used a ResNet-50 [He et al., 2016] based neural network with a batch size of 4 and 200,000 training iterations. Rest of the settings were kept default. The resulting network has a test error of 3.67 pixels.

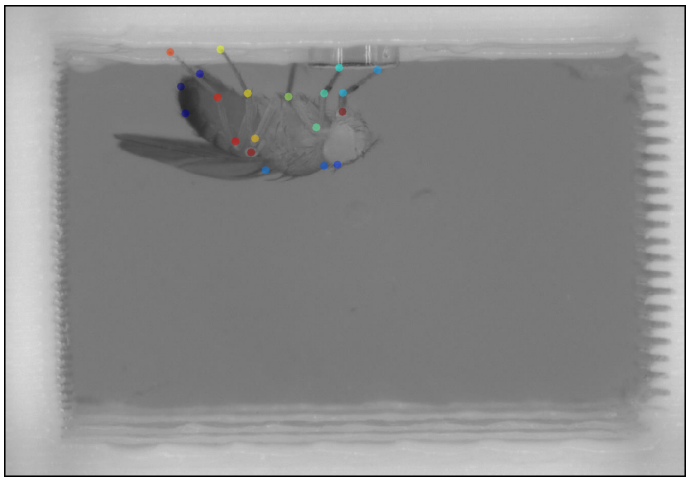
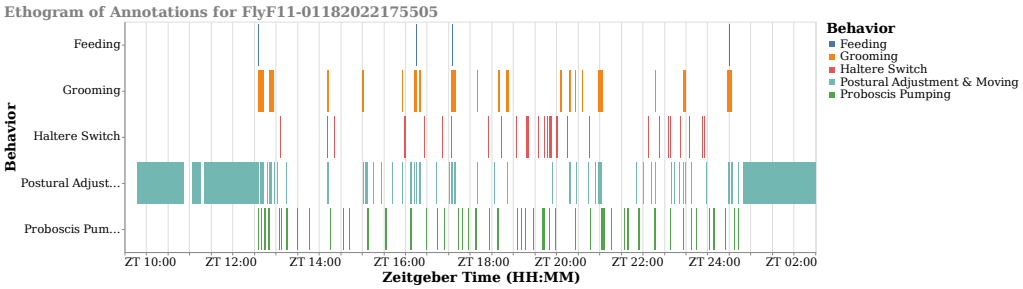


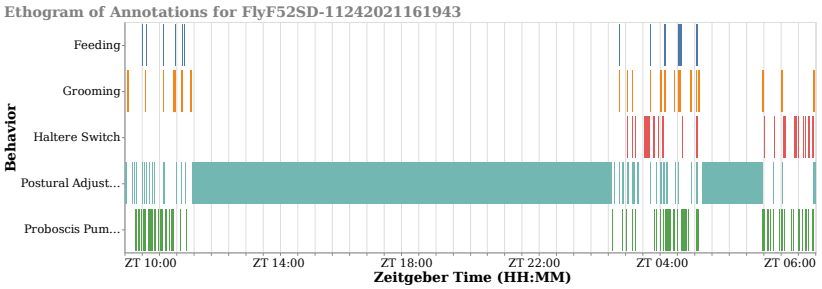
Figure 2.2: An example video recording frame of the fruit fly placed in 3D printed chamber. Colorful markers indicate the tracked body parts.

2.3 Behavior Annotations

3 human annotators labeled 5 different behaviors (feeding, grooming, moving, haltere switch, proboscis pumping) across 11 videos (14 hours and 16 hours each, respectively for 7 wild type sleep experiments and 4 sleep deprivation experiments). A single experiment is annotated by all 3 annotators to check rigor and overlap among annotators. We only used the annotations of a fly when each pair of annotators agreed at least on the 90% of the annotations. Two example ethograms generated from the annotations are given in the Figure 2.3.



(a) Wild type.



(b) Sleep-deprived.

Figure 2.3: Two example ethograms of annotated behaviors observed during the sleep experiments. Dark period starts at ZT12, and ends at ZT0.

Chapter 3

Feature Extraction

3.1 Overview of Feature Extraction

For a single experiment data, i.e., a single fruit fly recorded between ZT10 and ZT2 (zeitgeber time 10 and 2), feature extraction consists of four consecutive steps, where the input in the present stage is the output of the previous one. In this study, the input of the first step is the raw output signal of the tracking and pose estimation model which is produced by DeepLabCut, a toolbox for markerless pose estimation. The feature extraction steps are as follows:

1. Constructing pose values and preprocessing; dealing with occluded body parts, alignment of different orientations, filtering, and imputation.
2. Computing spatio-temporal features, such as distances between body parts, velocity, and angular velocity from body part positions.
3. Computing dynamic postural features by extending spatio-temporal features to multiple timescales using wavelet transformation and sliding window statistics.
4. Computing normalized high-dimensional behavioral representations.

Matrices $\mathbf{X} \in \mathbb{R}^{T \times N}$ and $\mathbf{Y} \in \mathbb{R}^{T \times N}$ denote multivariate time series for x and y cartesian components of two-dimensional video recordings. These data are collected for N tracked body parts of a fly in T consecutive time stamps by a pose estimation model. This multivariate time series matrices $\mathbf{X}$ and $\mathbf{Y}$ are the raw input data that goes into the first step of the feature extraction. Note that the number of body parts, N, must be the same among all experiments conducted with different fruit flies, but the number of time stamps, T, might differ. Each column of the $\mathbf{X} = \left[\mathbf{x}_1, \cdots, \mathbf{x}_N\right]^\top$

and $\mathbf{Y} = \left[\mathbf{y}_1, \cdots, \mathbf{y}_N\right]^\top$ can be written respectively as follows:

$$\begin{aligned} \mathbf{y}_i &= (y_{i,1}, y_{i,2}, \dots, y_{i,t-1}, y_{i,t}, y_{i,t+1}, \dots, y_{i,T}), \\ \mathbf{x}_i &= (x_{i,1}, x_{i,2}, \dots, x_{i,t-1}, x_{i,t}, x_{i,t+1}, \dots, x_{i,T}). \end{aligned} \tag{3.1}$$

Here i denotes the index of the body part, e.g., leg tip or proboscis.

In addition to $\mathbf{X}$ and $\mathbf{Y}$, a pose estimation model may report prediction scores for each tracked body part at each time step, which is the case for DeepLabCut as well. $L \in \mathbb{R}^{N \times T}$ denotes the time series of prediction scores, each column of the $L = [\mathbf{l_1}, \cdots, \mathbf{l_N}]^\top$ can be written as follows:

$$\mathbf{l}_i = (l_{i,1}, l_{i,2}, \dots, l_{i,t-1}, l_{i,t}, l_{i,t+1}, \dots, l_{i,T}). \tag{3.2}$$

The prediction scores tend to be very low when the body part is not visible. Thus, L provides valuable information about the occluded body parts. In the Section 3.2.1, how L is incorporated into construction of pose values is described in detail.

3.2 Preprocessing

This step involves preprocessing the signal by filtering and imputation of certain video frames. But in addition to this, there are a couple of optional procedures that can be beneficial for our task of learning stereotypical behaviors. These additional procedures deal with the occluded body parts of the fly, alignment of the fly orientations and defining new points of interest.

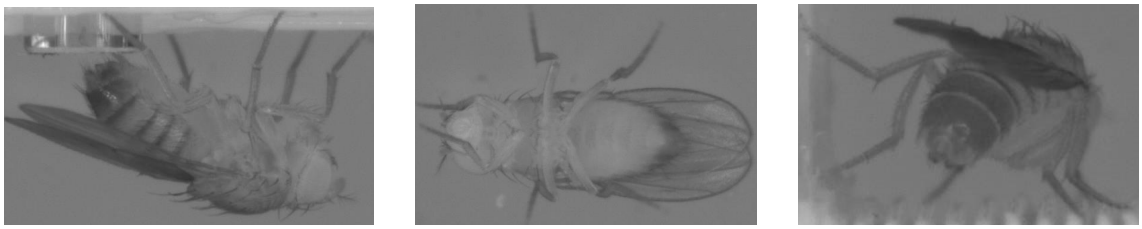
3.2.1 Occluded Body Parts

As mentioned in Chapter 1, the two-dimensional nature of the video recordings introduces several significant challenges, one of which is the occluded body parts. Occlusion describes various cases where the tracked parts are eclipsed by other parts in the view area, There are many types of occlusions. One type is short occlusions that often results from postural changes. Imputation of the time series $\mathbf{X}$ and $\mathbf{Y}$ for such short occlusions is relatively easy since the number of consecutive missing data points is limited. However, this is not the case for long occlusions, which usually occur when the fly is in dormancy for an extended period of time. Especially for the body parts with left and right counterparts, fly's orientation can result in one of the counterparts being occluded for long dormancy periods. We use imputation and elimination of the corresponding data-points to deal with the occlusions. Before

describing those approaches, we define a criterion for target occlusion.

Oriented Pose Values for Body Parts with Left & Right Counterparts

If the fly is oriented perpendicular to the camera perspective, as in Figure 3.1c, then one of the left and right body parts is often occluded. In other orientations (e.g., Figure 3.1a and Figure 3.1b), occlusion of both sides is possible as well as the case of no occlusion. However, in the conducted experiments, flies usually choose to stay dormant perpendicular to the camera perspective for long durations, as mentioned in Chapter 2. In such cases, one can concede using only one of the left or right counterparts to construct pose values. Therefore, this optional step is included in the behavioral mapping pipeline to reduce pose values of body parts with left and right counterparts to a single value.



(a) Parallel. (b) Oblique. (c) Perpendicular.

Figure 3.1: Three examples demonstrating the different orientations viewed by the camera. Orientations similar to Figure 3.1b and Figure 3.1c results in occluded body parts and hence erroneous tracking.

We use prediction scores to determine which body part should be used to compute oriented pose values. Let i and j be a pair of body parts which are left and right counterparts of each other, e.g., left and right haltere. Then, one can use the $\mathbf{l}_i$ and $\mathbf{l}_j$ to predict if one of them is occluded at a particular time step t. Let $\text{orient}(\mathbf{x}_i, \mathbf{x}_j)$ ($\text{orient}(\mathbf{y}_i, \mathbf{y}_j)$) be a new pose vector that will be computed based on $\mathbf{x}_i$ and $\mathbf{x}_j$ ($\mathbf{y}_i$ and $\mathbf{y}_j$), e.g., a vector of oriented haltere pose values, composed of left and right haltere pose values. The following conditional procedures are proposed to compute oriented pose values from the left and right pose values by deciding the orientation of the fly for a counterpart body pair. The procedures below can be used separately or successively.

- If $l_{i,t} - l_{j,t} \geq \epsilon$, then, without loss of generality, $\text{orient}(\mathbf{x}_i, \mathbf{x}_j)_t = x_{i,t}$ and $\text{orient}(\mathbf{y}_i, \mathbf{y}_j)_t = y_{i,t}$ for a threshold ϵ, typically $\epsilon > 0.5$.
- If $\left|\left\{\, t' \,:\, l_{i,t'} > l_{j,t'},\, t' \in [t-\tau \,..\, t+\tau] \,\right\}\right| > \tau$, then, without loss of generality, $\text{orient}(\mathbf{x}_i, \mathbf{x}_j)_t = x_{i,t}$ and $\text{orient}(\mathbf{y}_i, \mathbf{y}_j)_t = y_{i,t}$, for a window of size $2w+1$.
- If $l_{i,t} > l_{j,t}$ and if the nearest confident left orientation is closer than the

nearest confident right orientation, i.e.,

$$\arg\min_{t'} \left\{ |t - t'| \ : \ l_{i,t'} - l_{j,t'} \geq \epsilon \right\} > \arg\min_{t'} \left\{ |t - t'| \ : \ l_{j,t'} - l_{i,t'} \geq \epsilon \right\},$$

then, without loss of generality, $\text{orient}(\mathbf{x}_i, \mathbf{x}_j)_t = x_{i,t}$ and $\text{orient}(\mathbf{y}_i, \mathbf{y}_j)_t = y_{i,t}$.

- If simply $l_{i,t} > l_{j,t}$, then without loss of generality, $\text{orient}(\mathbf{x}_i, \mathbf{x}_j)_t = x_{i,t}$ and $\text{orient}(\mathbf{y}_i, \mathbf{y}_j)_t = y_{i,t}$.

Except for the direct comparisons based on prediction confidence scores as in the last procedure, some of the time points might be left with undecided orientations. If the number of such time points is manageably small, then direct comparisons of prediction scores for those time points are convenient and handy.

After applying the above procedures for a left and right counterpart pair i and j, we can define oriented multivariate time series as

$$\begin{aligned} \mathbf{X}^o &= \left(\left[(\mathbf{x}_k)_{k \notin \bigcup_{\{i,j\} \in \mathcal{O}} \{i,j\}} \right]^\top \,\middle|\, \left[\left(\text{orient}(\mathbf{x}_i, \mathbf{x}_j) \right)_{\{i,j\} \in \mathcal{O}} \right]^\top \right), \\ \mathbf{Y}^o &= \left(\left[(\mathbf{y}_k)_{k \notin \bigcup_{\{i,j\} \in \mathcal{O}} \{i,j\}} \right]^\top \,\middle|\, \left[\left(\text{orient}(\mathbf{y}_i, \mathbf{y}_j) \right)_{\{i,j\} \in \mathcal{O}} \right]^\top \right), \end{aligned} \tag{3.3}$$

where $\mathcal{O}$ is the set of index pairs of left and right counterparts and $\bigcup_{\{i,j\} \in \mathcal{O}} \{i,j\}$ is the union of all indexes of such body part pairs. Applying the procedures described above for each left and right counterparts results in computing $\mathbf{X}^o$ and $\mathbf{Y}^o$. Such oriented versions of the original data matrices can be used instead of $\mathbf{X}$ and $\mathbf{Y}$ in the rest of the pipeline, if desired.

Detecting Occlusions Using Prediction Scores & Anomalous Pose Values

Major postural changes and overlapping body parts during movements result in some body parts being occluded for a short interval of time. These types of occlusions may span several frames to several seconds. Since the pose estimation model makes predictions for such data points, it is necessary to detect and process such data points. Our pipeline exploits two indicators for detection: prediction scores and anomalous changes in pose values. We first mark the body parts estimated to be occluded and the corresponding time intervals. In the ensuing step, the desired imputation method is used to fill those time intervals appropriately.

The following conditions are considered to detect occluded body parts, note that they can be used separately or in combination with each other.

- If the prediction confidence score at time step t is lower than a given threshold ϵ, then the t is marked to be imputed.

- If z-score of the prediction confidence score $l_{i,t}$ computed within a window with size τ, and centered at t, is lower than a given threshold ϵ_z, then the t is marked to be imputed.
- If the second-order gradient of the estimated pose value exceeds a given threshold δ, then the t is marked to be imputed.
- If the difference between the estimated pose value at time step t and the median of pose values within the window of size τ, centered at time step t, exceeds the threshold δ, then the t is marked to be imputed.

Here, the window sizes and threshold parameters are determined separately for each condition. The conditions described above are not only useful for the occluded body parts, but are also beneficial for tackling unnatural and abnormal predictions of the pose estimation model.

3.2.2 Aligning Different Orientations

It may be desirable to align different orientations and postures in some cases. For instance, it is not possible to interpret vertical and horizontal replacements of body parts separately without aligning them into a reference frame. Flies can position themselves in different orientations while exhibiting similar actions, as it can be seen from the Figure 3.1a.

To rotationally align each frame, we transform body part coordinate values into an egocentric reference frame centered in the middle of the fly's spine, e.g., a line along the thorax. Then, we performed a linear transformation on the pose values to center the spine at the origin, oriented along the y-axis.

Alignment of the frames is an optional step in the behavioral mapping pipeline. It is potentially beneficial, and reasonable, to perform alignment when the Cartesian pose values of the body parts are included as spatio-temporal features.

3.2.3 Filtering and Imputation

Estimated pose values are highly noisy because of the reasons explained in Chapter 1. Thus, filtering the pose values and imputation of the data points marked as occluded or abnormal is an essential step before proceeding with the rest of the pipeline, which contains stages that are sensitive to noise.

One of the most practical and effective approaches for time series imputation is applying interpolation. We also benefit from univariate interpolation to replace values at marked time points, where the exact algorithm is chosen to be one of

the following; linear interpolation, spline interpolation, forward filling, or backward filling.

After imputation, we apply a median filter with appropriate window size and smooth each $\mathbf{x_i}$ and $\mathbf{y_i}$ separately. Finally, a boxcar filter (moving average) with a relatively small window size is used to filter out the rapidly changing signals by averaging. We observed that large window sizes may result in smoothing out some critical signals, which potentially define short-duration low-amplitude behaviors, e.g., switch-like haltere movement behavior.

A Rauch-Tung-Striebel Kalman smoother [Rauch et al., 1965] was employed in the development stage. However, no significant performance improvement is observed, and hence later abandoned due to its computational cost and the requirement of setting the various state parameters to reasonable values.

The configuration and actual parameters of the imputation methods and the filters are provided in Chapter 6.1.

3.3 Computation of Spatio-temporal Features

After preprocessing of pose values, learning stereotypical behaviors becomes feasible. Although tracking of relevant body parts and processing corresponding pose values is an essential step for quantifying behavior, a set of coordinate values is not sufficient to represent and capture complex spatio-temporal dynamics of animal behavior. There are thousands of unique postures, and behaviors are not even exhibited by some static set of postures. Instead, they are defined by expressive and meaningful spatio-temporal features such as distances, velocities, angles, and angular velocities. Therefore, one needs to compute such features from the coordinate values of body parts in two-dimensional space.

The second stage of the feature extraction is the computation of spatio-temporal features from pose values. Two types of features are computed in this stage, as listed below.

1. **Snapshot features:** Spatio-temporal feature values computed at a snapshot of time, listed as follows:
    - distances,
    - angles,
    - cartesian pose values (i.e., per body part features).

2. **Gradient features:** Spatio-temporal feature values computed based on how snapshot features change over time, listed as follows:
    - change of distances,
    - change of angles (i.e., angular velocities),
    - change of cartesian pose values (i.e., body part velocities).

The gradients of snapshot features are computed using second-order accurate central differences in the interior points. The resulting gradient features have the same shape, i.e., the number of features and the number of time-stamps, as the snapshot features.

3.3.1 Distances Between Body Parts

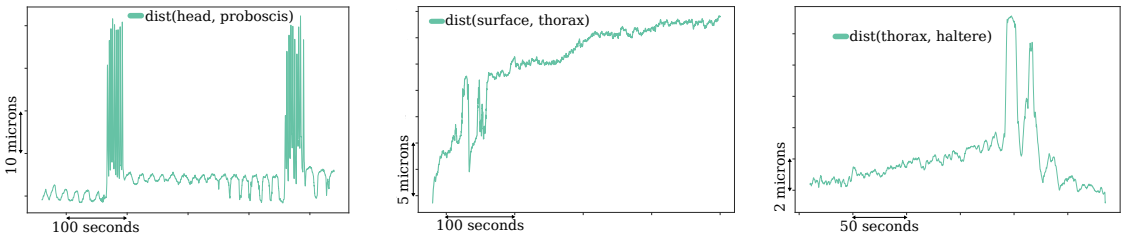
Given a body part pair (i, j), the distance between them at a time step t is calculated with the Euclidean distance, given below,

$$d_t^{i,j} = \sqrt{(x_{i,t} - x_{j,t})^2 + (y_{i,t} - y_{j,t})^2}. \tag{3.4}$$

The corresponding gradient feature, which is the change of distance between the body part i and j, is computed using the second-order gradient approximation,

$$\dot{d}_t^{i,j} = \begin{cases} \frac{|d_{t+1}^{i,j} - d_t^{i,j}|}{\Delta t} & \text{if t=0 or t=T,} \\ \frac{|d_{t+1}^{i,j} - d_{t-1}^{i,j}|}{2\Delta t} & \text{otherwise,} \end{cases} \tag{3.5}$$

where Δt is the sampling period, and it is equal to $^1/_{\text{FPS}}$ seconds.



(a) Pumping-like movement of the proboscis.

(b) Postural adjustment and movement of the thorax.

(c) Switch-like movement of the haltere.

Figure 3.2: Three spatio-temporal feature examples describing kinematics of different behaviors.

3.3.2 Joint Angles Between Body Parts

Given a triplet of body parts (i, j, k), the angle between i and k around j is calculated using the two-argument arctangent function as given below,

$$\omega_t^{i,j,k} = \text{atan2}\left(\det\begin{bmatrix} x_{i,t} - x_{j,t}, x_{k,t} - x_{j,t} \\ y_{i,t} - y_{j,t}, y_{k,t} - y_{j,t} \end{bmatrix}, \begin{bmatrix} x_{i,t} - x_{j,t} \\ y_{i,t} - y_{j,t} \end{bmatrix} \cdot \begin{bmatrix} x_{k,t} - x_{j,t} \\ y_{k,t} - y_{j,t} \end{bmatrix}\right) + \pi. \tag{3.6}$$

Then, similar to the change of distance features, the angular velocities are approximated by

$$\dot{\omega}_t^{i,j,k} = \begin{cases} \frac{|\omega_{t+1}^{i,j,k} - \omega_t^{i,j,k}|}{\Delta t} & \text{if t=0 or t=T,} \\ \frac{|\omega_{t+1}^{i,j,k} - \omega_{t-1}^{i,j,k}|}{2\Delta t} & \text{otherwise.} \end{cases} \tag{3.7}$$

3.3.3 Cartesian Pose Values of Body Parts

The cartesian pose values of a body part i are straightforwardly given by the x and y coordinate values as follows:

$$\begin{aligned} x_t^i &= x_{i,t}, \\ y_t^i &= y_{i,t}. \end{aligned} \tag{3.8}$$

Note that for such a single body part, two feature values are generated. Corresponding gradient features, namely the body part velocities along each cartesian component, are computed by

$$\begin{aligned} \dot{x}_t^i &= \begin{cases} \frac{x_{t+1}^i - x_t^i}{\Delta t} & \text{if t=0 or t=T,} \\ \frac{x_{t+1}^i - x_{t-1}^i}{2\Delta t} & \text{otherwise,} \end{cases} \\ \dot{y}_t^i &= \begin{cases} \frac{y_{t+1}^i - y_t^i}{\Delta t} & \text{if t=0 or t=T,} \\ \frac{y_{t+1}^i - y_{t-1}^i}{2\Delta t} & \text{otherwise.} \end{cases} \end{aligned} \tag{3.9}$$

In order to compute the overall two-dimensional velocity of a body part, one can always use the distance between the origin and corresponding body part.

3.3.4 Constructing Spatio-temporal Feature Matrices

Let $\mathcal{C}, \mathcal{D}$, and $\mathcal{A}$ denote the sets of body parts, body part pairs, and body part triplets; respectively defining cartesian pose values, distances, and angles, respectively. Similarly, let $\mathcal{C}', \mathcal{D}'$, and $\mathcal{A}'$ denote sets that define sets of respective gradient features. Then the snapshot feature matrix S constructed as follows:

$$\mathbf{S} = \left(\left[(\mathbf{x}^i)_{i\in\mathcal{C}}\right] \,\middle|\, \left[(\mathbf{y}^i)_{i\in\mathcal{C}}\right] \,\middle|\, \left[(\mathbf{d}^{i,j})_{\{i,j\}\in\mathcal{D}}\right] \,\middle|\, \left[(\boldsymbol{\omega}^{i,j,k})_{\{i,j,k\}\in\mathcal{A}}\right]\right), \tag{3.10}$$

where the vectors are defined as $\mathbf{x}^i = \left[x_1^i, \cdots, x_T^i\right]$, $\mathbf{y}^i = \left[y_1^i, \cdots, y_T^i\right]$, $\mathbf{d}^{i,j} = \left[d^{i,j}, \cdots, d_T^{i,j}\right]$, and $\boldsymbol{\omega}^{i,j,k} = \left[\omega_1^{i,j,k}, \cdots, \omega_T^{i,j,k}\right]$.

Similarly, for gradient features, the feature matrix is constructed by concatenating change of distances, angular velocities and body part velocities; given by

$$\mathbf{G} = \left(\left[\left(\dot{\mathbf{x}}^i\right)_{i \in \mathcal{C}'} \right] \Big| \left[\left(\dot{\mathbf{y}}^i\right)_{i \in \mathcal{C}'} \right] \Big| \left[\left(\dot{\mathbf{d}}^{i,j}\right)_{\{i,j\} \in \mathcal{D}'} \right] \Big| \left[\left(\dot{\boldsymbol{\omega}}^{i,j,k}\right)_{\{i,j,k\} \in \mathcal{A}'} \right] \right), \tag{3.11}$$

where the vectors are defined as $\dot{\mathbf{x}}^i = \left[\dot{x}_1^i, \cdots, \dot{x}_T^i\right]$, $\dot{\mathbf{y}}^i = \left[\dot{y}_1^i, \cdots, \dot{y}_T^i\right]$, $\dot{\mathbf{d}}^{i,j} = \left[\dot{d}^{i,j}, \cdots, \dot{d}_T^{i,j}\right]$, and $\dot{\boldsymbol{\omega}}^{i,j,k} = \left[\dot{\omega}_1^{i,j,k}, \cdots, \dot{\omega}_T^{i,j,k}\right]$.

The resulting two feature matrices are $\mathbf{S} \in \mathbb{R}^{T \times (2|\mathcal{C}|+|\mathcal{D}|+|\mathcal{A}|)}$, namely snapshot feature matrix, and $\mathbf{G} \in \mathbb{R}^{T \times (2|\mathcal{C}'|+|\mathcal{D}'|+|\mathcal{A}'|)}$, namely gradient feature matrix. N_S denotes the number of snapshot features, which is given by $2|\mathcal{C}| + |\mathcal{D}| + |\mathcal{A}|$ and N_G denotes the number of gradient features, which is equal to $2|\mathcal{C}'| + |\mathcal{D}'| + |\mathcal{A}'|$.

3.4 Computation of Dynamic Postural Features

Instantaneous values of spatio-temporal features do not provide a sufficient description of complex postural dynamics of behaviors. Understanding the output of a complex biological system, in our case the behavior, can only be achieved by studying multiple timescales concurrently. Previous studies attempted to search behavioral motifs, e.g., repeated sub-sequences of actions with finite length, within the behavioral time series [Ye and Keogh, 2011, Brown et al., 2013]. However, as Berman et al. [2014] states, this paradigm usually requires problems of temporal alignment and relative phasing between different scales. Alternatively, extending spatio-temporal features to capture postural dynamics at different timescales eliminate requirements of temporal alignment and motif based analysis. In order to extend the spatio-temporal features to dynamic postural features, we applied wavelet transformation (similar to Berman et al. [2014]) and computed moving statistics at different timescales (similar to Kabra et al. [2013]), respectively for the snapshot feature set ($\mathbf{S}$) and the gradient feature set ($\mathbf{G}$).

3.4.1 Moving Statistics of Gradient Features

Gradient features only reflect the instantaneous values of velocities with respect to the sampling rate. In order to capture how these values change within a given interval, the moving statistics of gradient features such as the such as the mean and the standard deviation are computed with a sliding window approach. Let τ be the window size parameter, i.e., the timescale of interest, then the moving mean of the

corresponding gradient feature $\mathbf{g}_i$ is given by the function μ_τ, as given below:

$$\mu_\tau(g_{i,t}) = {}^{1}\!/_{\min\{t+\tau,\, T\} - \max\{t-\tau,\, 1\} + 1} \sum_{t'=\max\{t-\tau,\, 1\}}^{\min\{t+\tau,\, T\}} g_{i,t'}. \tag{3.12}$$

Similarly, the moving standard deviation of a gradient feature $g_{i,t}$ by is computed by σ_τ, as in the below equation.

$$\sigma_\tau(g_{i,t}) = \left({}^{1}\!/_{\min\{t+\tau,\, T\} - \max\{t-\tau,\, 1\} + 1} \sum_{t'=\max\{t-\tau,\, 1\}}^{\min\{t+\tau,\, T\}} \left(\mu_\tau(g_{i,t}) - g_{i,t'}\right)^2 \right)^{1/2} \tag{3.13}$$

Moving statistics feature generation approach has been used to learn animal behavior by Kabra et al. [2013], and Marshall et al. [2021] has also included such features into the analysis.

3.4.2 Wavelet Transformation of Snapshot Features

The wavelet domain is a useful representation of postural dynamics due to the following reasons given by Berman et al. [2014], and the proposed spectrogram generation is used by others as well [Marshall et al., 2021, Todd et al., 2017].

- It describes dynamics over multiple timescales simultaneously by possessing a multi-resolution time-frequency trade-off.
- It eliminates the requirement of precise temporal alignment for capturing periodic behaviors by taking amplitudes of the continuous wavelet transform of each snapshot feature at different scales.

Given a function $s(t)$, the continuous wavelet transformation at a frequency $f > 0$ is expressed as the following integral:

$$W_{f,t'}\left[s\left(t\right)\right] = \frac{1}{\sqrt{a\left(f\right)}} \int_{-\infty}^{\infty} s\left(t\right) \Psi^* \left(\frac{t - t'}{a(f)}\right) \mathrm{d}t\,, \tag{3.14}$$

where Ψ is the wavelet function and a is a function for converting frequencies to wavelet scale factor. The Morlet wavelet is well suited for describing postural dynamics which is closely related to human aural and vision perception [Daugman, 1985], and it is used in the pipeline. The corresponding wavelet function is given by

$$\Psi(t) = \exp\left\{\frac{t^2}{2}\right\} \cos(w_0 t), \tag{3.15}$$

where w_0 is a non-dimensional parameter. The frequency-to-scale conversion function a for the Morlet wavelet is as follows:

$$a\left(f\right) = \frac{w_0 + \sqrt{2 + w_0^2}}{4\pi f}. \tag{3.16}$$

For the discrete sequence of snapshot feature $\mathbf{s}_i$ with sampling period Δt, $W_{f,t'}[s(t)]$ translates into

$$W_f\left(\mathbf{s}_i, t'\right) = \frac{1}{\sqrt{a(f)}} \sum_{t=1}^{T} \Delta t s_{i,t} \Psi^* \left(\frac{t - t'}{af}\right), \tag{3.17}$$

where $t', t \in \mathbb{Z}$ and $1 \leq t' \leq T$ [Torrence and Compo, 1998].

Normalization of Wavelet Power Spectrum

In order to ensure that wavelet transforms (Equation 3.17) at each frequency f are directly comparable to each other and to the other transformed time series, the transformation W_f has to be normalized at each frequency f to have unit energy. This normalization for the Morlet wavelet at frequency f is as follows:

$$C(f) = \frac{\pi^{-\frac{1}{4}}}{\sqrt{2a(f)}} \exp\left\{ 1/4 \left(w_0 - \sqrt{w_0^2 + 2}\right)^2 \right\}. \tag{3.18}$$

The resulting normalized transformation, which is also used to generate the spectrogram in Berman et al. [2014], is given by

$$W_f^0\left(\mathbf{s}_i, t'\right) = \frac{1}{C\left(f\right)} \left| W_f\left(\mathbf{s}_i, t'\right) \right|. \tag{3.19}$$

In addition to the above conventionally used normalization, we alternatively adopted the normalization proposed by Liu et al. [2007]. According to this alternative adjustment, the wavelet power spectrum should be equal to the transform coefficient squared divided by the scale it associates.

$$W_f^0\left(\mathbf{s}_i, t'\right) = \frac{W_f\left(\mathbf{s}_i, t'\right)^2}{a\left(f\right)} \tag{3.20}$$

We observed substantial improvements using this power spectrum.

Determining Spectrum Frequencies

We investigate two different approaches for computing a set of frequencies, and we include both of them in the behavior mapping pipeline. One set is dyadically spaced

frequencies between $f_{\min}$ and $f_{\max}$ via

$$f_i = f_{\max} 2^{-\frac{i-1}{N_f-1} \log \frac{f_{\max}}{f_{\min}}}, \tag{3.21}$$

where $f_{\max} = {}^{FPS}/_{2}$ Hz is the Nyquist frequency.

The other alternative set of frequencies is linearly spaced between $f_{\min}$ and $f_{\max}$ by

$$f_i = f_{\min} + \frac{f_{\max} - f_{\min}}{N_f - 1} i, \tag{3.22}$$

for $i = 1, 2, \ldots, N_f$, and their corresponding wavelet scales are computed by $a(f_i)$.

3.4.3 Constructing Dynamic Postural Feature Tensors

Let $\mathcal{T}_S = \{ {}^{1}/_{f_{\min}}, \cdots {}^{1}/_{f_{\max}} \}$ and $\mathcal{T}_G = \{ \tau_{\min}, \cdots, \tau_{\max} \}$ denote the timescale sets respectively for wavelet transforms of snapshot features and moving statistics of gradient features. Then corresponding feature tensors are given as follows:

$$\begin{aligned} \mathbf{W} &= \left(W_f^0 (\mathbf{s}_i, t) \right)_{1 \le t \le T,\, {}^{1}/_{f} \in \mathcal{T}_S,\, 1 \le i \le N_S}, \\ \mathbf{M}^{\mu} &= \left(\mu_\tau \left(g_{i,t} \right) \right)_{1 \le t \le T,\, \tau \in \mathcal{T}_G,\, 1 \le i \le N_G}, \\ \mathbf{M}^{\sigma} &= \left(\sigma_\tau \left(g_{i,t} \right) \right)_{1 \le t \le T,\, \tau \in \mathcal{T}_G, 1 \le i \le N_G}. \end{aligned} \tag{3.23}$$

The resulting extended feature tensors of dynamic postural representations are $\mathbf{W} \in \mathbb{R}^{T \times |\mathcal{T}_s| \times N_S}$, $\mathbf{M}^{\mu} \in \mathbb{R}^{T \times |\mathcal{T}_v| \times N_G}$ and $\mathbf{M}^{\sigma} \in \mathbb{R}^{T \times |\mathcal{T}_v| \times N_G}$.

3.5 Computation of Behavioral Representations

After applying wavelet transformation or computing moving statistics to extend extracted spatio-temporal features to dynamic postural features, a couple of additional operations are required to continue in the behavior mapping pipeline.

3.5.1 Flattening Dynamic Postural Feature Tensors

As constructed in Section 3.4, dynamic postural feature tensors are $\mathbf{W} \in \mathbb{R}^{T \times |\mathcal{T}_S| \times N_S}$, $\mathbf{M}^{\mu} \in \mathbb{R}^{T \times |\mathcal{T}_G| \times N_G}$, and $\mathbf{M}^{\sigma} \in \mathbb{R}^{T \times |\mathcal{T}_G| \times N_G}$. In order to apply manifold learning-based dimensionality reduction algorithms or traditional machine learning algorithms such as decision trees, the last two dimensions of these feature tensors need to be contracted to obtain a matrix representation. As a result, feature matrices are $\tilde{\mathbf{W}} \in \mathbb{R}^{T \times (N_S |\mathcal{T}_S|)}$, $\tilde{\mathbf{M}}^{\mu} \in \mathbb{R}^{T \times (N_G |\mathcal{T}_G|)}$ and $\tilde{\mathbf{M}}^{\sigma} \in \mathbb{R}^{T \times (N_G |\mathcal{T}_G|)}$ are obtained.

3.5.2 L_1 Normalization of Features

Dynamic postural feature distributions of similar behaviors may differ among flies due to different characteristics such as sex and sleep deprivation, Due to the two-dimensional nature of the video recordings, different orientations may cause observing different feature values for the same behavior. In order to have a homogeneous feature space among flies and throughout the temporal dimension, at each time step t, L_1 normalization is applied as follows:

$$\hat{\mathbf{w}}_i = \left(\frac{\tilde{w}_{t,i}}{\sum_{j=1}^{N_S|\mathcal{T}_s|} \tilde{w}_{t,j}} \right)_{1 \leq t \leq T} \quad \text{and} \quad \hat{\mathbf{W}} = \left[(\hat{\mathbf{w}}_i)_{1 \leq i \leq N_S|\mathcal{T}_S|} \right]. \tag{3.24}$$

Similarly, L_1 normalized versions of $\tilde{\mathbf{M}}^{\mu}$ and $\tilde{\mathbf{M}}^{\sigma}$, namely $\hat{\mathbf{M}}^{\mu}$ and $\hat{\mathbf{M}}^{\sigma}$ are obtained. Here $\hat{\mathbf{W}}$, $\hat{\mathbf{M}}^{\mu}$ and $\hat{\mathbf{M}}^{\sigma}$ are the final multivariate time series of normalized high dimensional behavioral representation of a single experiment data, i.e., single fruit fly recorded between ZT10 and ZT2. Notice that we may treat each time step, that is, the frame, as a discrete probability distribution after L_1 normalization.

Chapter 4

Activity Detection

4.1 Overview of Activity Detection

Each experiment comprises sixteen hours of video recording spanning both awake and asleep epochs. Since we are only interested in the behavioral repertoire exhibited during sleep, time intervals where the animal is dormant, namely the dormancy epochs, should be detected before proceeding with the behavior mapping stage. We characterize dormancy epochs by lack of macro-activities, i.e., significant postural and locational changes, which can be detected by displacement of the animal. After detecting and excluding intervals of macro-activities, we end up with time points where the fly is dormant. An additional processing step is needed for the dormancy epochs, as we are not interested in the time points where the fly is totally quiescent. Our major focus is on the micro-activity bouts manifested during a dormancy epoch. In order to detect those bouts, we should distinguish micro-activities exhibited during dormant epochs from macro-activities by quantifying them with a closer look at various body parts. We use the term "bouts" and "epochs" respectively for micro-activities and macro-activities to reflect their difference in terms of duration. As it is shown in Section 6.2, behaviors categorized in micro-activities are tends to have shorter durations, compared to macro-activities.

At this stage, our ultimate goal is to extract bouts of micro-activities exhibited during the dormancy state. Extracted micro-activity bouts constitute the data points that are subject to behavior mapping. There are several benefits of reducing the data points to this subset of dormancy and micro-activity, instead of using the entire experiment for behavior mapping. Considering the high frame rate and long length of video recordings, computational requirements are an important concern in our pipeline. Since at least the 90% of the frames are either totally quiescent or macro-activities, e.g., walking, this approach has the benefit of reducing the

computational requirements significantly. Another critical point is that the quiescence frames contain only noise energy, and normalizing each frame amplifies me normalization amplifies this low-level noise energy, generating a uniform-like probability distribution for behavioral representation [Todd et al., 2017]. Eliminating pure quiescence frames without any micro-activity avoids this. Also, as we are only interested in the behaviors exhibited during sleep, excluding macro-activity frames prevent the domination of large number of frames with walking and macro-activities in the behavioral embedding space.

In this chapter, we first describe the quantification of the macro-activities (Section 4.2.1) and micro-activities (Section 4.2.2) in the Section 4.2. After that, in the Section 4.3, we discuss two different approaches, unsupervised detection (Section 4.3.1) and supervised detection (Section 4.3.1), for detection of micro-activities and macro-activities, and constructing corresponding frame sets Dormancy, Macro-activity, and Micro-activity.

4.2 Quantifying Activities

4.2.1 Dormancy and Macro-activity Epochs

When a fly is awake, many behaviors are manifested by featuring major postural changes and displacement of the body in different ways. We categorize this type of behaviors under the umbrella term of macro-activity, and dormancy is defined as the lack of macro-activities and characterized by micro-activities. One can characterize macro-activities without considering its sub-categories by using the velocities, i.e., gradient features. In order to distinguish sub-categories of macro-activities, such as walking and rearing, more detailed and descriptive features are required. However, in our case, computing a single scalar value to capture the overall movement of a fly result is sufficient for detecting macro-activity epochs. We define this feature value by summing the gradient features for all timescales, utilizing the dynamic postural feature tensor M^{μ}, as follows:

$$v_t = \sum_{\tau \in \mathcal{T}_G} \sum_{i=1}^{N_G} \mu_\tau \left(g_i, t\right), \tag{4.1}$$

where the resulting velocity-based feature vector is $\mathbf{v} = (v_1, \cdots, v_T)$. Essentially, high and low values of v_t indicate macro-activity and dormancy, respectively. Micro-activities can not be detected by using such a straightforward and general value, and therefore, can not be distinguished from dormancy by solely using only this quantity.

4.2.2 Micro-activity Bouts

Behaviors exhibited during dormancy epochs are not necessarily accompanied by postural changes and the displacement of the animal's body. For instance, the pumping-like movement of the proboscis or switch-like movement of haltere tends to happen independent from the remaining body parts. Thus, one needs to consider more than a single scalar value, as in the previous case of macro-activity. For each snapshot feature s_i, we sum all frequency channels f, i.e., timescales, utilizing the dynamic postural feature tensor W, as follows:

$$u_{t,i} = \sum_{1/f \in \mathcal{T}_G} W_f^0(s_i, t) \quad \text{or} \quad u_{t,i} = \max_{1/f \in \mathcal{T}_G} W_f^0(s_i, t). \tag{4.2}$$

The resulting feature vectors, $\mathbf{u}_i = \left(u_{i,1}, \cdots, u_{i,T}\right)$, are representative enough for capturing the micro-activities as an umbrella category. Such micro-activities potentially occur independently from the remaining body parts. For example, using the snapshot feature of the distance between haltere and posterior thorax, we are able to detect the switch-like movement of haltere in dormancy epochs, even if the rest of the body is at rest.

4.3 Detecting Activities

4.3.1 Unsupervised Approach

The straightforward approach for detecting macro activities would be determining a global threshold based on the distribution of the values of $\mathbf{v}$ and $\mathbf{u}$. Instead of determining such a threshold value, denoted by c, we use treat the distribution of $\mathbf{v}$ and $\mathbf{u}$ as a mixture of Gaussian distributions to detect an appropriate threshold value, separately for each experiment. This approach avoids the hassle of dealing with the varying distributions of $\mathbf{v}$ and $\mathbf{u}$ among different experiments, and constitute a solid ground for threshold value.

Consider a mixture of univariate Gaussian distributions with K components, $z = 1, \cdots, K$, sorted by their corresponding mean values $\mu_1, \cdots, \mu_K$, and with full covariance matrices. Then we can define $K - 1$ many decision boundaries by:

$$\mathbb{P}(V = \delta_k \mid z = k) = \mathbb{P}(V = \delta_k \mid z = k + 1), \tag{4.3}$$

where $k = 1, \cdots, K - 1$, V denotes a random variable for values of v_t and δ_k is the threshold value for the kth decision boundary. After computing the decision boundaries for the mixture of Gaussian distributions, the macro-activity threshold

is set to either one of the decision boundaries (δ_k for $k = 1, \cdots, K-1$) or one of the means (μ_k for $k = 1, \cdots, K$). For example, a reasonable choice would be $K = 2$ and threshold $c = \mu_2$, or $K = 3$ and threshold $c = \delta_2$. Classification is done by constructing the following frame sets:

$$\begin{aligned} \mathsf{Dormancy} &= \{\, t : \; v_t \leq c, \; 1 \leq t \leq T \,\}, \\ \mathsf{Macro\text{-}activity} &= \{\, t : \; v_t > c, \; 1 \leq t \leq T \,\}. \end{aligned} \tag{4.4}$$

We follow a similar approach for detecting micro-activities. Since we look for micro-activities in dormancy epochs, now we only consider the frames from the set Dormancy. As micro-activity bouts are quantified by multiple feature vectors, we determine separate threshold values for each $\mathbf{u}_i$, using the same approach with macro-activity detection, utilizing a mixture of Gaussian distributions. After determining thresholds c_i for each $\mathbf{u}_i$, we construct the following frame sets:

$$\begin{aligned} \mathsf{Quiescence} &= \left\{ t : \; \bigwedge\nolimits_{i=1}^{N_G} u_{t,i} \leq c_i, \; t \in \mathsf{Dormancy} \right\}, \\ \mathsf{Micro\text{-}activity} &= \left\{ t : \; \bigvee\nolimits_{i=1}^{N_G} u_{t,i} > c_i, \; t \in \mathsf{Dormancy} \right\}. \end{aligned} \tag{4.5}$$

For a frame to be identified as micro-activity, it is sufficient if at least one feature to be above the corresponding threshold. This is due to the fact that, micro-activities are manifested by the displacement of only a small subset of body-parts locations.

4.3.2 Supervised Approach

Annotations contain 5 behavioral categories, namely grooming, postural adjustments, proboscis pumping, haltere switch, and feeding. These behavioral categories correspond to the micro-activities that we are interested in. We formulate a binary classification problem by considering all behavioral categories as positive class, and the rest as negative class. After constructing the Dormancy set with the unsupervised approach described in the Section 4.3.1, we train a random forest of decision trees [Breiman, 2001] with all the frames of annotated flies' Dormancy set. Similar to unsupervised approach, we use $\mathbf{u}_i$ as the training features. For a single annotated fly, the resulting training feature matrix is $\mathbf{U} \in \mathbb{R}^{T \times N_G}$. In contrast to the original publication [Breiman, 2001], we use the Scikit-learn [Pedregosa et al., 2011] implementation combines classifiers by averaging their probabilistic prediction, instead of letting each classifier vote for a single class. Micro-activity set is constructed with the frames predicted as the positive class, corresponding to the union of annotated behavioral categories, and similarly Quiescence set consists of the frames predicted as the negative class.

Configuration and the parameters of the random forest of decision trees is given in the Section 6.1, and its performance is evaluated and compared with the unsupervised approach in the same section as well.

Chapter 5

Behavior Mapping

5.1 Overview of Behavior Mapping

Our ultimate goal is to have a fine-grained categorization of the behaviors observed during dormancy. Thus, it is not sufficient to detect the activities and construct the sets Micro-activity and Macro-activity. Behavior mapping stage is the most critical and novel part of the pipeline, and in this stage, we discover and predict stereotypical behaviors by mapping each frame in the sets Micro-activity.

The behavior mapping stage starts by generating low dimensional behavioral embeddings from the high dimensional behavioral representation matrix $\hat{\mathbf{W}}$, but only the rows corresponding to the Micro-activity are included in the mapping. Rest of the frames, namely the Quiescence set and Macro-activity set directly assigned to quiescent and moving categories. In the Section 5.2, behavioral embedding generation is discussed in detail, including supervised, semi-supervised and unsupervised approaches. We use semi-supervised pair UMAP embeddings, described in the Section 5.2.3, to map frames to behaviors.

The next step is a novel nearest neighbor analysis in the generated low dimensional behavioral spaces, as we detail in the Section 5.3. Nearest neighbor analysis consists of several parts. First one is the computation of the behavioral weights (Section 5.3.1). Next, we combine behavioral weights provided by each "view", i.e., annotated experiment by forming a committee (Section 5.3.2). Finally, post-processing is described in the Section 5.3.3.

5.2 Behavioral Embeddings

The dynamic postural features are able to capture many different timescales, however their high-dimensional structure makes it challenging to directly exploit behavioral representations in analysis, learning, and visualization. For example, the behavioral representation matrix ($\hat{\mathbf{W}}$) computed from dynamic postural features ($\mathbf{W}$) with 20 spatio-temporal features and 25 timescales (i.e., frequency channels) has 20×25=500 columns. Since the correlation between different spatio-temporal features (e.g., distance between head and proboscis and cartesian pose values of proboscis) and different timescales are often strong (see Figure 5.1), one may expect that the topological structure of the high dimensional behavioral representations ($\hat{\mathbf{W}}$) can be accurately represented in a lower dimensional space. Therefore, we would like to find a low-dimensional embedding that captures the important features of the dataset. The embedding we compute should minimize local distortions, since trajectories pause near a repeatable position whenever a particular stereotyped behavior is observed [Berman et al., 2014, DeAngelis et al., 2019, Ali et al., 2019].

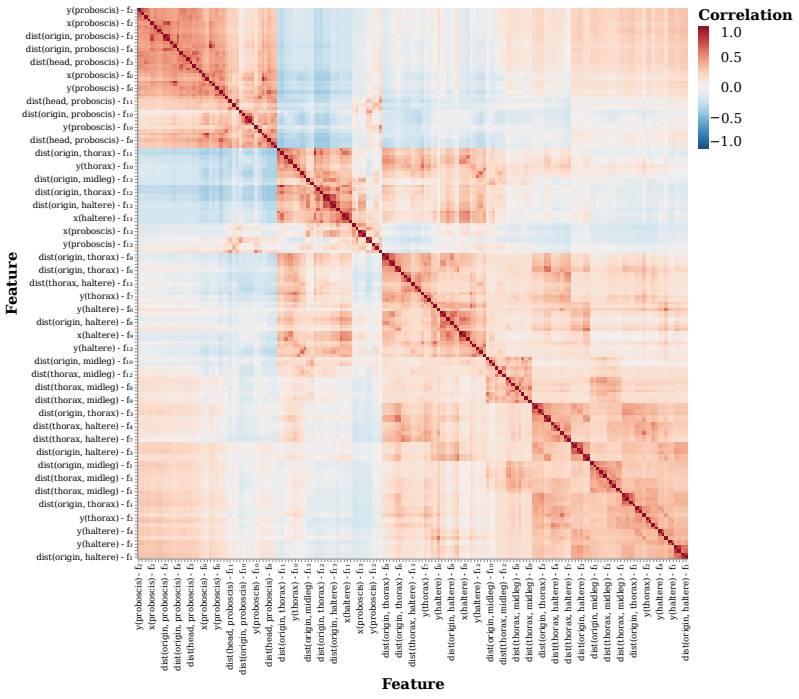


Figure 5.1: Spearman's correlation coefficient between each feature value of behavioral representations. Features are clustered and grouped based on the absolute value of the correlation values. Frequency channels $f_1, \cdots, f_20$ are dyadically spaced between 1 Hz and 20 Hz.

Many dimensionality reduction algorithms seek to preserve the pairwise distance structure among all the data samples, the well-known examples of such algorithms are PCA [Hotelling, 1933], and MDS [Kruskal, 1964]. Alternatively, some algorithms favor the preservation of local distances over global distance, such as UMAP (Uni-

form Manifold Approximation & Projection) [McInnes et al., 2020] t-SNE [Maaten and Hinton, 2008], Laplacian Eigenmaps [Belkin and Niyogi, 2003] and LargeVis [Tang et al., 2016]. The latter category of algorithms aims to achieve to preserve important local structures, and helps to improve classification performance when used in combination with learning algorithms where the function is only approximated locally, e.g., k-nearest neighbor classifier [McInnes et al., 2020]. Similarly, it has been shown that manifold learning based dimensionality reduction algorithms can improve the clustering performance [Sainburg et al., 2021]. Moreover, the trade-off of preserving local distances over global distances does not introduce any significant disadvantages in the latter stages of the behavioral pipeline.

We use UMAP for its superior performance in many aspects. McInnes et al. [2020] demonstrates that a k-nearest neighbor classifier trained on UMAP embeddings achieves higher accuracy for large k values, compared to PCA, t-SNE, LargeVis and Laplacian Eigenmaps since it captures non-local scales in a markedly effective way and local scales comparably or better. Thus, it can be argued that UMAP has captured more of the global and topological structure of the benchmark datasets than its counterparts, t-SNE and LargeVis. Another reason for choosing UMAP over other alternatives is that it tends to produce more stable embeddings. It is capable of producing sub-sample embeddings which are very close to the full embedding even for sub-samples of 5% of the dataset, outperforming the results of t-SNE and LargeVis. Finally, computational performance of UMAP scales well with the number of samples, the embedding dimensionality and the data dimensionality, compared to t-SNE LargeVis, and Eigenmaps, resulting significantly lower run-times on various datasets such as COIL-20 [Nene et al., 1996], Fashion-MNIST [Xiao et al., 2017], and GoogleNews [Mikolov et al., 2013]. Run-time performance of the dimensionality reduction algorithm is an important concern in the behavioral mapping pipeline, as the number of samples and the data dimensionality is often on the order of 10^6 and 10^2, respectively.

UMAP and other non-linear dimensionality reduction algorithms that attempt to use a mathematical structure akin to a k-nearest neighbor graph to approximate a manifold, follow a similar basic structure as below [McInnes et al., 2020].

- Graph Construction
    1. Construct a weighted k-nearest neighbor graph.
    2. Apply some transformation on the edge weights to envelop local distances.
    3. Deal with the incompatibility of the local metrics, i.e., disagreeing weights

of the edges.

- Graph Layout

    1. Define an objective function that preservers desired characteristics of the k-nearest neighbor graph.

    2. Compute a low dimensional representation by optimizing the defined objective function.

A distance metric is needed to construct a k-nearest neighbor (k-NN) graph. In our case, we normalized the feature matrices as described in Section 3.5.2, feature vector entries at each time step sum up to 1, and therefore the feature vectors can be considered as discrete probability distributions. We use the Hellinger distance [Hellinger, 1909] to quantify the similarity between "discrete probability distributions" of features, which is given by

$$\text{Hellinger}(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{k} (\sqrt{p_i} - \sqrt{q_i})^2}. \tag{5.1}$$

Then, based on the Hellinger distances, UMAP constructs a weighted k-NN graph using nearest neighbor descent [Dong et al., 2011]. Then, the k-NN graph is modified by making edges directed and defining the new weights using local Riemannian metric of each data point. After this step, there exist two edges with disagreeing weights between the nearest neighbors, as the local metrics differ. UMAP combines those weights by using t-conorm and the resulting weight can be interpreted as the probability of at least one of the two directed edge to exist.

Finally, UMAP uses a force directed graph layout algorithm in low dimensional space where attractive and repulsive forces are defined based on the gradients, optimizing the edge-wise cross-entropy between the original weighted graph and equivalent weighted graph induced by the embeddings. The algorithm proceeds by iteratively applying attractive and repulsive forces at each edge or vertex. Since the "true" graph captures the topology of the source data, the approximated graph also matches the overall topology of the data, and thus produces a meaningful low dimensional representation. A detailed mathematical and algorithmic description of the UMAP algorithm can be found in [McInnes et al., 2020], and it is skipped here as it goes beyond the scope of this work.

The algorithm described above is an unsupervised method, but it can be easily extended to work in a supervised or semi-supervised manner. Although we use a useful metric, e.g., Hellinger distance, defining the distance between a set of points, one can also define a simple metric for categorical values to extend UMAP further for

supervised and semi-supervised cases. We can obtain a second view on the source data by using a metric where distances for points in the same and different categories as well as points without a category (for the semi-supervised case) are defined appropriately. A straightforward and simple example would be defining distances as follows: 1 if the points are in the same category, 0 if the points are in different categories, 0.5 if either of the points is uncategorized. One can combine weighted graphs constructed using the two distance metrics (local metrics and defined metric for categorical values), and arrive at a shared view on the data. In the behavioral mapping pipeline, we benefit from unsupervised UMAP, and its supervised and semi-supervised extensions for different purposes.

The utilized behavioral embeddings fall into three different categories, namely "disparate embeddings", "joint embeddings" and "pair embeddings", detailed descriptions and their applications are described respectively in Section 5.2.1, Section 5.2.2, and Section 5.2.3 respectively.

The resulting behavioral embedding

$$\mathbf{E} = \left[\mathbf{e}_1, \cdots \mathbf{e}_F\right] \in F \times N, \tag{5.2}$$

is a low-dimensional representation of

$$\text{row}_{t_f} \hat{\mathbf{W}} \in \mathbb{R}^{F \times (N_S | \mathcal{T}_S|)} \ \left(t_f \in \textsf{Micro-activity}\right), \tag{5.3}$$

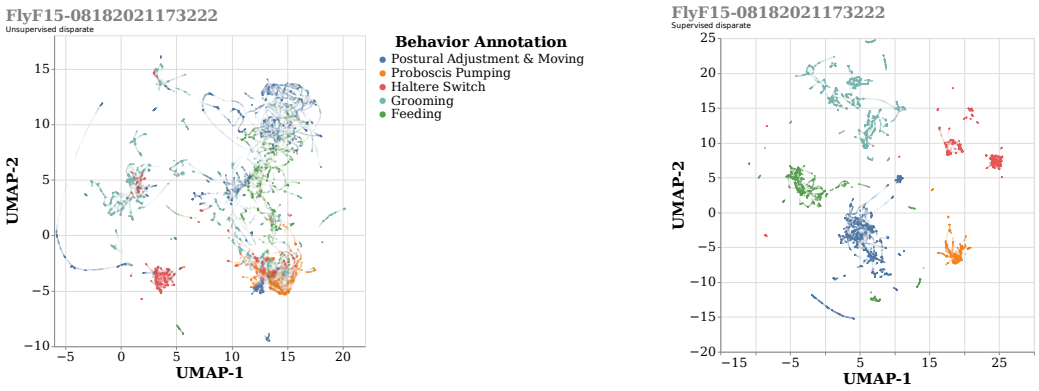
where and $N << (N_S|\mathcal{T}_S|)$ and F is the numbers of frames estimated as dormant and active, being equal to $|\textsf{Micro-activity}|$. Each frame f, corresponds to a time point $t_f \in \textsf{Micro-activity}$.

5.2.1 Disparate Embeddings

UMAP may be used to embed high-dimensional behavioral representations of each experiment separately to obtain disparate behavioral embeddings. Treating each experiment separately is useful for several purposes.

For example, using supervised UMAP for annotated experiments, we can explore behavioral sub-categories in annotations as annotations are very high-level, biased and general categorization of behaviors. For instance, one annotation category, e.g., "grooming", can be consisted of two different clusters in the behavioral embedding space, corresponding to "grooming of head" and "grooming of abdomen" (see Figure 5.2b). Defining very specific and low-level behavioral categories is not feasible and prone to error, a post-annotation analysis using disparate embeddings helps us to zoom in annotated behaviors. Another scenario of benefiting disparate embed-

dings is using unsupervised UMAP for annotated and/or unannotated experiments separately. We can visualize how behavioral repertoire is represented in the low dimensional embeddings space and analyze which features drives different regions and clusters of the behavioral embeddings (see Figure 5.2a).



(a) Unsupervised. (b) Supervised.

Figure 5.2: Two-dimensional supervised and unsupervised behavioral embeddings of FlyF15-08182021173222. Both embeddings reveal variations within annotated behavioral categories.

5.2.2 Joint Embeddings

Ideally, we would like to embed all experiments into a shared joint behavioral space, and using such an embedding would enable us to discover stereotypical and common behaviors among different flies. However, we observed that this is only possible to a certain extent and such joint embeddings tend to not mix well as the number of experiments increases. When we jointly compute a behavioral space using all available experiments, the resulting embedding is usually not totally homogeneous in terms of flies and experiments. We observed that similar behaviors might end up embedded in different regions, and multiple clusters consisting of highly similar behaviors emerges.

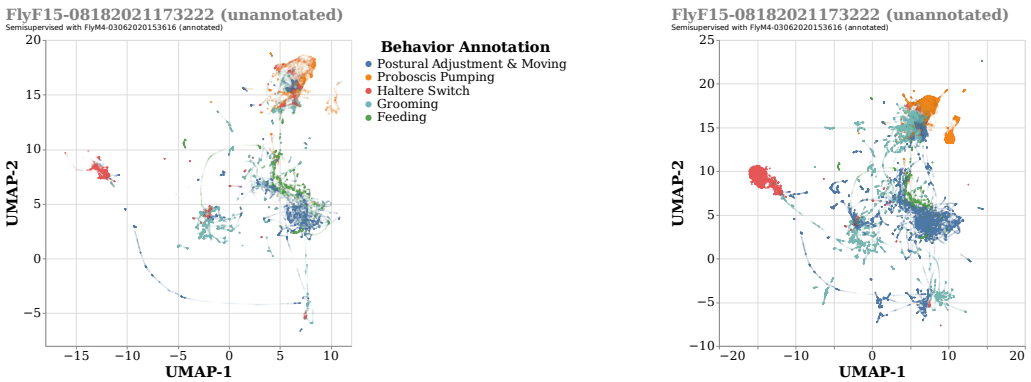
Especially for the joint treatment of annotated experiments, supervised UMAP fails to embed similar behaviors of different experiments closely and homogeneously. There are many factors contributing to this impairment of supervised joint embeddings, such as behavioral variations among flies and experiments, differences in orientation while exhibiting similar behaviors, and broad definitions of annotation categories. Unless the number of jointly embedded experiments does not exceed several, unsupervised UMAP performs relatively well, and enables a fully unsupervised and unbiased analysis of behaviors. We can exploit unsupervised joint behavioral embeddings for visualization purposes to discover how different feature combinations are exhibited, and density-based clustering to extract similar behav-

ioral bouts. However, utilizing unsupervised joint embeddings becomes problematic as the number of experiments increases, and behavioral space gets "too crowded".

5.2.3 Pair Embeddings

Using disparate embeddings does not allow one to embed an annotated and an unannotated experiment into a joint behavioral space, and joint embeddings poorly blend different experiments in a shared space. Instead, we propose a novel alternative approach to benefit from the semi-supervised dimensionality reduction capabilities of UMAP, while avoiding generating a hard-to-interpret embedding. In this approach, namely semi-supervised pair embeddings, we compute a joint behavioral space for each annotated and unannotated pair, using the available annotations. As a result, for R^- unannotated experiments and R^- annotated experiments, a semi-supervised pair embedding will be generated for each $R^- \times R^+$ pair.

For a single unannotated experiment, a semi-supervised behavioral embedding for each annotated experiment provides different "views". Especially when the behavioral repertoire of the annotated and the unannotated experiments are similar, the provided "view" turns out to be an accurate, easy-to-interpret low-dimensional representation of the exhibited behaviors of the unannotated experiment. When the behavioral repertoire and/or feature distribution are dissimilar, the resulting embedding may not provide useful information about the unannotated experiment, but an advantage of this approach is that the other pair embeddings do not get distorted by poor matches. As described in Section 5.3, semi-supervised pair embeddings are utilized to predict behavioral categories of unannotated experiments by combining multiple view acquired from annotated ones.



(a) Only unannotated experiment. (b) Unannotated and annotated experiments.

Figure 5.3: Semi-supervised pair embeddings with an annotated and an unannotated experiment. FlyM4-03062020 (annotated) provides a "view" on the behavioral repertoire of the FlyF15-08182021 (unannotated).

5.3 Nearest Neighbor Analysis and Classification

One of our ultimate goals is to annotate an experiment using already annotated ones. In previous sections and chapters, we described feature extraction, activity detection, and computation of behavioral embeddings. In this section, we describe a novel nearest neighbor based classification approach, which has to tackle following challenges:

- sparsity of behavioral expressions,
- imbalanced distribution of behavioral categories,
- scarcity of annotated experiments, which are laborious to produce,
- variation between experiments.

Our approach consists of two steps, briefly stated as follows:

1. computing behavioral weights of an unannotated experiment, using its semi-supervised pair embeddings for each annotated experiment,
2. combining behavioral weights of the unannotated experiment with an annotated experiment committee by voting.

We compute the behavioral weights with our nearest neighbor based approach by considering sparsity of behavioral expressions and imbalanced distribution of behavioral categories, as described in Section 5.3.1. In the Section 5.3.2, we detail the committee by voting approach, which helps us to deal with variation between experiments by providing multiple "views" on observed behavioral expression that we attempt to annotate. Finally, the post-processing of resulting predictions is described in the Section 5.3.3.

5.3.1 Behavioral Weights

Consider two experiments, an annotated one expt^{+}, and an unannotated one expt^{-}, and their semi-supervised pair embedding, respectively $\mathbf{E}^{+} = \left[\mathbf{e}_1^{+}, \cdots \mathbf{e}_{F^{+}}^{+}\right]$ and $\mathbf{E}^{-} = \left[\mathbf{e}_1^{-}, \cdots \mathbf{e}_{F^{-}}^{-}\right]$. Given the true annotations $\mathbf{y}^{+}$ of the frames in the $\mathsf{Micro\text{-}activity}^{+}$ set of expt^{+}, and K behavioral categories, the goal is to compute $\hat{\mathbf{b}}_f = \left[\hat{b}_{f,1}, \cdots \hat{b}_{f,K}\right]$, representing the weights (in other words, the similarity score) of each behavioral category for expt^{-}, using $\mathbf{E}^{+}, \mathbf{E}^{-}$ and $\mathbf{y}^{+}$.

The procedure start by querying k-nearest neighbors of expt^{-}'s each frame f in the joint behavioral embedding space consisting of $\mathbf{E}^{+}$ and $\mathbf{E}^{-}$, using the k-d trees for efficiency [Bentley, 1975]. $k\text{-NN}\,(f)$ denotes the set of indices of $\mathbf{e}_f^{-}$'s k nearest neighbors, and the k-NN weight $b_{f,i}$ for each query point (i.e., frame) f of expt^{-},

and behavioral category i, is computed by

$$b_{f,i} = \begin{cases} \sum\limits_{f' \in k\text{-NN}_i(f)} \dfrac{1}{\mathrm{d}\left(\mathbf{e}^-_f, \mathbf{e}^+_{f'}\right)^p + \epsilon} & \text{if } |k\text{-NN}_i(f)| \neq 0, \\ 0 & \text{if otherwise,} \end{cases} \qquad \text{where} \quad p \in \{0, 1, 2\}. \tag{5.4}$$

Here, $k\text{-NN}_i(f) = \left\{ f' : y^+_{f'} = i, \text{ and } f' \in k\text{-NN}(f) \right\}$, is the set of indices of data points of expt$^+$ whose annotation is the behavior category i and is one of the k nearest neighbors of $\mathbf{e}^-_f$. $\mathrm{d}\left(\mathbf{e}^-_f, \mathbf{e}^+_{f'}\right)$ is the euclidean distance between $\mathbf{e}^-_f$ and $\mathbf{e}^+_{f'}$, and p parameterizes the relation between distance and weight $b_{f,i}$. We add a small number ϵ (10^{-6}) to the denominator to avoid numerical errors. The resulting vector $\mathbf{b}_f = \left[b_{f,1}, \cdots, b_{f,K}\right]$ weights the similarity of the frame f to each behavioral category in the shared embedding space based on nearest neighbors.

Naturally, the number of occurrences or durations of the behavior bouts are different for each behavioral category, and therefore, $\mathbf{y}^+$ is highly imbalanced. As a result, number of nearest neighbors and $\mathbf{b}_f$ are biased in favor of frequently occurring and long-bout behaviors. For instance, pumping-like movements of the proboscis occur more frequently in longer bouts than switch-like movements of the haltere. Especially when k is large, it becomes crucial to consider the imbalanced distribution of behavior occurrences, since the embedding space will be dominated by frequent behaviors. Thus, incorporating this imbalance into the formulation may help to improve the recall of rarely occurring short-bout behaviors and precision of frequently occurring long-bout behaviors. To achieve this, we normalize the scores previously computed, $b_{f,i}$, as a function of the number of occurrences of the behavioral category i as follows:

$$, b'_{f,i} = \frac{b_{f,i}}{(1 + N^+_i)^p} \quad \text{or} \quad \frac{b_{f,i}}{\log_k(1 + N^+_i)} \quad \text{where} \quad p \in \{0, 1/2, 1\}, k \in \{2, 10\}, \tag{5.5}$$

where $N^+_i = \left|\left\{ f : y^+_f = i \right\}\right|$ is the number of frames annotated as behavioral category i. In the above equation, two different alternatives are given for this normalization step; a polynomial one and a logarithmic one, where p and k parameterize the relation between N^+_i and $b'_{f,i}$. For instance, if one is mostly interested in achieving high recall for frequently occurring behaviors, low p values or using the logarithmic alternative might be more appropriate. It may be even desired to set $p = 0$, and not considering the number of occurrences in some cases, see Section 6.1 for more details.

The resulting vector $\mathbf{b}'_f \in \mathbb{R}^K$ is dependent on the annotated experiment expt$^+$, and

the vectors computed based on different annotated experiments are not comparable with each other. Hence, we map the values of $b'_{f,i}$ to $[0,1]$ using either the softmax function or L_1 normalization as follows:

$$\hat{b}_{f,i} = \frac{\exp\{b'_{f,i}\}}{\sum_{j=1}^{K} \exp\{b'_{f,j}\}} \quad \text{or} \quad \frac{b'_{f,i}}{\sum_{j=1}^{K} b'_{f,j}}. \tag{5.6}$$

The resulting behavioral weight vector $\hat{\mathbf{b}}_f \in [0,1]^K$ can be considered as a probability distribution. Here, the vector $\hat{\mathbf{b}}_f$ represents the behavioral characteristics of the frame f of expt$^-$ based on the behavioral repertoire of expt$^+$. The voting-like scheme, as described in Section 5.3.2, incorporates the behavioral weight vectors of all annotated experiments to finalize the classification for expt$^-$.

5.3.2 Experiment Committee by Voting

Consider all experiments: unannotated experiments $\text{expt}^-_1, \cdots, \text{expt}^-_{R^-}$, and annotated experiments $\text{expt}^+_1, \cdots, \text{expt}^+_{R^+}$, where R^- and R^+ are the number of experiments, respectively. The goal is to combine the behavioral weights of an unannotated experiment expt^-_k, calculated separately for each annotated experiment.

Let $\hat{\mathbf{b}}^{k,l}_f$ denote the behavioral weights of expt^-_k computed with expt^+_l. Each annotated experiment contributes to the overall behavioral score of expt^-_k; in other words, annotated experiments, forming a committee, vote according to $\hat{\mathbf{b}}^{k,l}_f$. Each annotated experiment provides a different view to the behavioral repertoire, as in the case of pair embeddings (see Section 5.2.3). Before describing hard voting and soft voting approaches, there is one more step to discuss.

There exists a significant variation among the exhibited behavioral repertoires in the experiments. An annotated experiment might lack some behavioral expressions or manifest some behaviors excessively. In such cases, if the behavioral weight vector $\hat{\mathbf{b}}^{k,l}_f$ is not confident, i.e., weights of behavioral categories are close to each other, one may want to decrease its contribution to the voting. To achieve this, we propose two optional approaches; penalizing the behavioral weights with the entropy of the "probability distribution" $\hat{\mathbf{b}}^{k,l}_f$, or with the uncertainty. The contribution of votes of expt^+_l to the committee formed for expt^-_k is $\mathbf{vote}^{k,l}_f = \left[\mathsf{vote}^{k,l}_{f,1}, \cdots, \mathsf{vote}^{k,l}_{f,K}\right]$, and is given by

$$\mathsf{vote}^{k,l}_{f,i} = \left(\log_2(K) - \mathrm{H}\left(\hat{\mathbf{b}}^{k,l}_f\right)\right) \hat{b}^{k,l}_{f,i} \quad \text{or} \quad \left(1 - \max_{1\le j\le K} \hat{b}^{k,l}_{f,j}\right) \hat{b}^{k,l}_{f,i} \quad \text{or} \quad \hat{b}^{k,l}_{f,i}. \tag{5.7}$$

If $\max_i \hat{b}_{f,i}$ is close to $1/K$, which means that the computed vector weights the behaviors uniformly, then the factors $\left(\log_2(K) - \mathrm{H}\left(\hat{\mathbf{b}}^{k,l}_f\right)\right)$ or $\left(1 - \max_{1\le j\le K} \hat{b}^{k,l}_{f,j}\right)$

may be used to decrease the "importance" of the vote.

Now, after computing behavioral votes for each annotated and unannotated experiment pair, we can combine those votes for a single unannotated experiment expt^-_k by forming a committee of annotated experiments $\text{expt}^+_1, \cdots, \text{expt}^+_{R^+}$. The predicted behavioral category at frame k of expt^-_k is given by combined votes of the committee. The predicted behavioral category at frame k of expt^-_k, denoted by $\widehat{y}^k_f$, is given by combined votes of the committee. To obtain the overall aggregate view of the committee, we can follow two alternative voting schemes, namely hard voting (i.e., majority rule voting) or soft voting. At this point, we may want to have scores for each behavioral category rather than hard-labels. Such score information might be desired to capture complex and hierarchical structure of the behaviors. Behaviors sometimes occur simultaneously, and sometimes observed behaviors might manifest similarities to more than one behavioral category [Berman et al., 2016]. Hence, one may also use the combined vote scores directly, without computing the $\arg\max$.

Hard Voting

In the hard voting scheme, prediction of a frame f is given by the majority behavioral category of the $\arg\max$ of the each annotated experiment's votes and computed by the following formula:

$$\widehat{y}^k_f = \underset{1\leq i\leq K}{\arg\max} \left\{ \underset{1\leq j\leq K}{\arg\max}\ \mathsf{vote}^{k,l}_{f,j} \ :\ j = i,\ 1 \leq l \leq R^+ \right\}. \tag{5.8}$$

Soft Voting

In contrast to majority voting (hard voting), the soft voting scheme assigns the behavioral category as the $\arg\max$ of the sum of each annotated experiment's vote vector, and $\widehat{y}^k_f$ is given by

$$\widehat{y}^k_f = \underset{1\leq i\leq K}{\arg\max} \sum_{l=1}^{R^+} \mathsf{vote}^{k,l}_{f,i}. \tag{5.9}$$

Behavioral Scores

Definitions of behavioral categories might be broad, and expressed behaviors sometimes are a combination of multiple behavioral categories. For example a switch-like movement of the haltere can simultaneously occur with a postural adjustment. Also, one may also want to examine and interpret top-k predictions, especially when there exists many narrow behavioral categories. Thus, in addition to assigning categories to frames, it is also useful to compute and report scores for behavioral category

based on the votes contributed by each annotated experiments. Moreover, such scores can be utilized as confidence scores, and might be helpful to analyze differences in behavioral repertoire (see Figure 6.2 and Figure 6.5).

For a behavioral category i, such a score can be calculated by combining votes as follows:

$$\frac{\sum_{l=1}^{R^+} \mathsf{vote}_{f,i}^{k,l}}{\sum_{i=1}^{K} \sum_{l=1}^{R^+} \mathsf{vote}_{f,i}^{k,l}}. \tag{5.10}$$

5.3.3 Post-processing

Finally, we post-process predicted annotations to improve our nearest neighbor based algorithm by incorporating a couple of assumptions on the temporal organization of the behaviors and physical constrains. Before applying post-processing procedures, the frames in the sets Macro-activity and Quiescence should be recovered, as we only predicted the annotations of the frames in the set Micro-activity. Without having a temporally continuous vector of annotations, post-processing would lead unintended consequences and erroneous results. Let $\mathbf{pred\text{-}a}^k = (\hat{y}_1^k, \cdots, \hat{y}_T^k)$ be the expt_k's vector of predicted annotations for the entire experiment, defined as follows:

$$\mathsf{pred\text{-}a}_t^k = \begin{cases} 0 & \text{if } t \in \mathsf{Quiescence}_k, \\ \hat{y}_f^k & \text{if } t \in \mathsf{Micro\text{-}activity}_k \text{ and } t = t_f, \\ K+1 & \text{if } t \in \mathsf{Macro\text{-}activity}_k. \end{cases} \tag{5.11}$$

This vector of annotations spans an entire experiment of k, and consists of detailed annotations of behavioral subcategories during dormancy, the general category of macro-activities, and quiescence.

Before discussing the post-processing, we first need to define behavioral bouts. Informally, a behavioral bout is a segment of time, in which the same behavioral category is continuously observed. We can formally define a behavioral bout for a given $\hat{y}_t^k$ as follows:

$$\begin{aligned} \mathsf{bout}_t^0 &= \max \left\{ \arg\max_{t'} \ (\hat{y}_t^k \neq \hat{y}_{t'}^k) \wedge (1 \leq t' \leq t) \vee (t' = 1) \right\}, \\ \mathsf{bout}_t^1 &= \min \left\{ \arg\max_{t'} \ (\hat{y}_t^k \neq \hat{y}_{t'}^k) \wedge (t \leq t' \leq T) \vee (t' = T) \right\}, \end{aligned} \tag{5.12}$$

where bout_t^0 and bout_t^1 are respectively the beginning and the end of the behavioral bout, which $\hat{y}_t^k$ belongs to.

We have sensible expectations about the durations of behavioral bouts of each be-

havioral category acquired by human annotators. For instance, a bout of proboscis's pumping-like movement should not be shorter than 200 millisecond, as bout duration distributions of annotations indicates. In addition to annotations, we may have reason about physical and biological constraints. An example is that grooming behavior, bout duration of grooming can not exceed a couple of minutes, probability of observing a grooming behavior lasted longer than 2 minutes is extremely low [Qiao et al., 2018]. Considering such constraints and temporal expectations, we apply the following post-processing procedures, and parameters should be determined based on the behavioral repertoire of interest. Post-processing procedures are especially helpful for avoiding misleading classification of the time points with erroneous tracking of body parts. Each procedure is optional, but should be applied in the given order.

Pruning Interrupting Bouts

We prune short intervals interrupting possibly long and continuous behavioral bouts. Given a window size τ, the majority behavioral category in the surrounding window around a time point t is assigned that point by:

$$\widehat{\mathsf{pred\text{-}a}}_t^k = \arg\max_{\mathsf{a}} |\{\, t' \,:\, \mathsf{pred\text{-}a}_{t'}^k = \mathsf{a},\ \max\{1, t-\tau\} \le t' \le \min\{t+\tau, T\}\,\}|. \tag{5.13}$$

The window size τ is typically less than a second.

Elimination of Overly Long and Overly Short Bouts

We eliminate behavioral bouts whose durations are extremely short or extremely long, and hence, contradicting with physical constraints and our temporal expectations. For each behavioral category i, we define two thresholds δ_i^{short} and δ_i^{long}, namely upper bound and lower bound. Then, the behavioral bouts whose duration longer or shorter than the corresponding thresholds, are set to quiescence.

$$\widehat{\mathsf{pred\text{-}a}}_t^k = \begin{cases} 0 & \mathsf{bout}_t^0 - \mathsf{bout}_t^1 < \delta^{\text{short}}_{\mathsf{pred\text{-}a}_t^k}, \\ 0 & \mathsf{bout}_t^0 - \mathsf{bout}_t^1 > \delta^{\text{long}}_{\mathsf{pred\text{-}a}_t^k}, \\ \mathsf{pred\text{-}a}_t^k & \text{otherwise}, \end{cases} \tag{5.14}$$

For each behavioral category i, the thresholds δ_i^{short} (δ_i^{long}) should be greater (smaller) than the window size τ used in Section 5.3.3.

Chapter 6

Results

6.1 Employing the Pipeline and Evaluation

Total of 11 experiments (7 wild type sleep and 4 sleep-deprived) and their annotations are split as 10 training experiments and 1 test experiment for all combinations. We report results and evaluate each split and demonstrate varying performance for different splits.

Our pipeline starts with the feature extraction stage, where raw tracking data generated by DeepLabCut is used to generate meaningful behavioral representations. Body parts used to extract features are proboscis, haltere, thorax, head (left and right), leg joints (left and right). We use oriented pose vales for body parts with left and right counterparts, as described in the Section 3.2.1. In order to detect occluded body parts, we set the low confidence score threshold to 0.075. Moving median window size τ is 15 frames, (0.5 seconds) and threshold τ set to 15 microns. If the position of a body part exceeds the median of the window centered around it by τ, it is marked as occluded, as described in Section 3.2.1.

Linear interpolation is used for imputation of marked frames. After that, firstly, a median filter of size 6 frames, and then a boxcar filter of size 6 frames is applied to reduce the tracking noise and smooth the signal, without smoothing behaviors exhibited by rapid movements. Aligning different orientations and transforming the frames to be egocentric did not result in performance improvement in our case, hence, we did not transform frames egocentric.

Then, we computed snapshot features and gradient features as described in Section 3.3, and given in the Table 6.1. In order to generate postural dynamics, we applied wavelet transformation to snapshot features at 20 different frequency channels dyadically spaced between 1 Hz and 20 Hz (see Equation 3.21 for determining spec-

trum frequencies). Different timescales are normalized as given in Equation 3.20. After flattening and L_1 frame normalization, we ended up with a $13 \times 20 = 260$ dimensional behavioral representation matrix $\hat{\mathbf{W}}$. Similarly, for gradient features, moving mean values are computed for a single timescale which is 33 milliseconds, resulting in $9 \times 1 = 9$ dimensional behavioral representation matrix $\hat{\mathbf{M}}^{\mu}$, which is only used for activity detection and constructing Dormancy set.

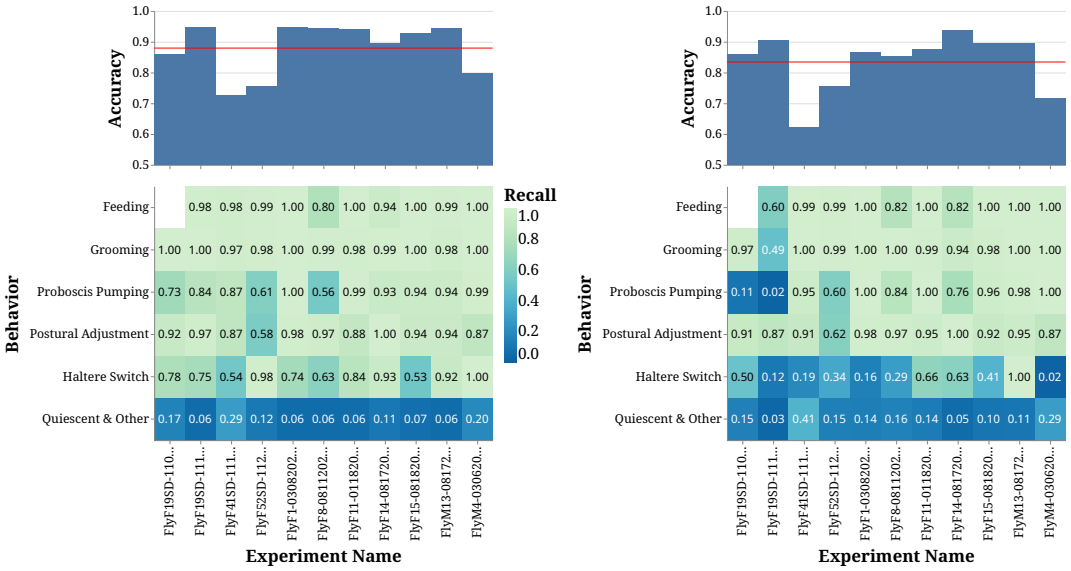
	Snapshot Features	**Gradient Features**
Distance between	haltere and origin proboscis and origin thorax and origin head and proboscis haltere and thorax (average) leg joints and origin (average) leg joints and thorax	head and proboscis thorax and proboscis thorax and origin
Cartesian components of	haltere head thorax	head proboscis thorax

Table 6.1: Computed spatio-temporal features.

After constructing behavioral representation matrices, the activity detection stage of our pipeline takes place. Here reported recall scores are for the frame set Micro-activity, and it is desired to achieve high recall scores, except for the "quiescent and other" category. We evaluate both unsupervised and supervised approaches, recall, and accuracy values are shown in Figure 6.1.

For unsupervised activity detection, the threshold c is set to the decision boundary λ_1 of two Gaussian components for construction of the Dormancy set. Similarly, thresholds c_i for each $\mathbf{u}_i$ value are the first decision boundaries of 3 Gaussian components, sorted by their means. Resulting detection performance is quite poor for switch-like behavior of haltere, since this behavior occurs very subtly, and it is similar to quiescent frames. 8 out of 11 splits, has recall score less than 0.5. Also, recall scores for two of the sleep-deprived experiment splits (FlyF19SDs) are below 0.15, which makes it impossible to proceed with a successful mapping.

In the supervised approach, a random forest of decision trees [Breiman, 2001] is utilized with 10 estimators, where the maximum depth of each tree is 5 and the criterion is Gini impurity. For grooming, recall is always greater than 0.97, which implies that we do not lose an insignificant amount of annotated frames. Similarly, the recall score of feeding drops below 0.95 only once. For proboscis pumping, 6 out of 11 splits have recall score greater than 0.9, and performance is relatively poor for two splits (0.61 and 0.56 recall). Again, haltere switch is the most challenging



(a) Supervised detection.

(b) Unsupervised detection.

Figure 6.1: Performance summary of activity detection and micro-activity detection. The red line indicates the macro average of accuracy scores achieved for each split. High recall scores are desired for behavioral categories, as opposed to quiescent frames. Supervised and unsupervised detections are given respectively in the left subfigure and in the right subfigure.

behavioral category, but recall is often greater than 0.75 as opposed to unsupervised approach. Considering its superior performance over unsupervised approach, we proceed to behavior mapping by employing supervised detection.

After activity detection, the next stage is the behavior mapping, where the frames in the set Micro-activity are mapped to behavioral categories. First step is computation of behavioral embeddings. The semi-supervised pair embeddings described in the Section 5.2.3 computed for each annotated and unannotated experiment pair. We set the embedding space dimension to 2, which 260 dimensional behavioral representation matrix $\hat{\mathbf{W}}$ is reduced to. Number of neighbors parameter of UMAP is set to 75, minimum distance parameter is determined as 0, and the distance metric is Hellinger distance (Equation 5.1).

Then, with the generated behavioral embeddings, nearest neighbors analysis is performed to assign behavioral scores to unannotated frames. In each semi-supervised embedding, we computed 25 nearest annotated neighbors of the frames, and contributions of the neighbors are weighted disproportional to its distance, using the Euclidean distance, as formally given in the Equation 5.4. After that, weights are normalized with the $\log_2$ number of occurrences of behavioral category of contributing neighbors, and then L_1 normalized to make behavioral weights sum up to 1 (see Equation 5.5, 5.6). Finally, we summed each behavioral weight, obtained from a

different "view" (i.e., annotated experiment), and L_1 normalized again to end up with the final behavioral score values.

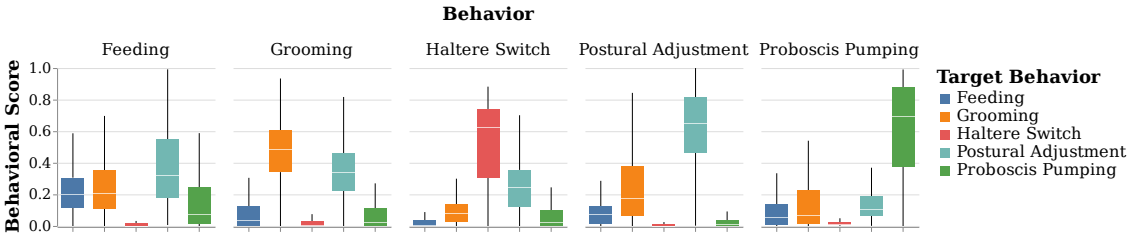
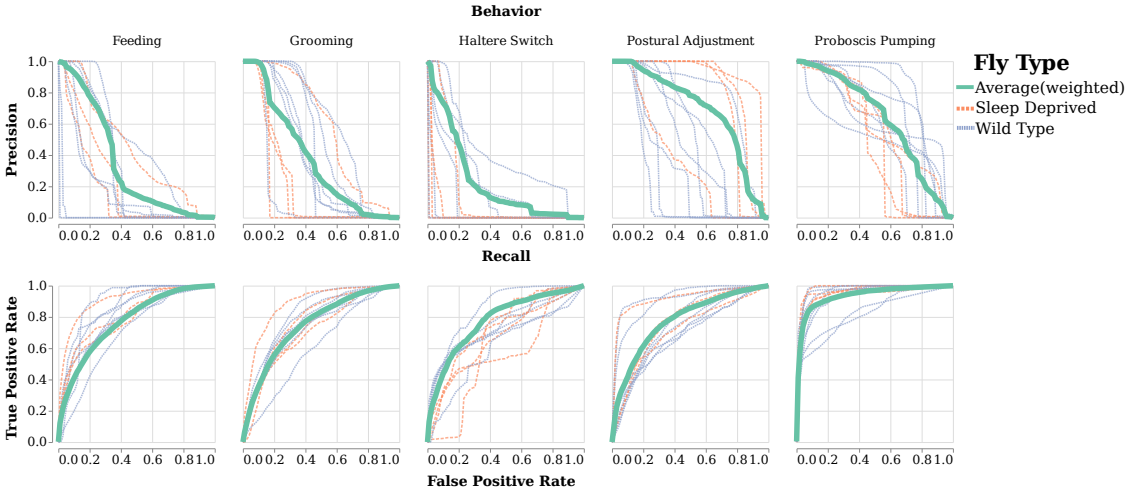


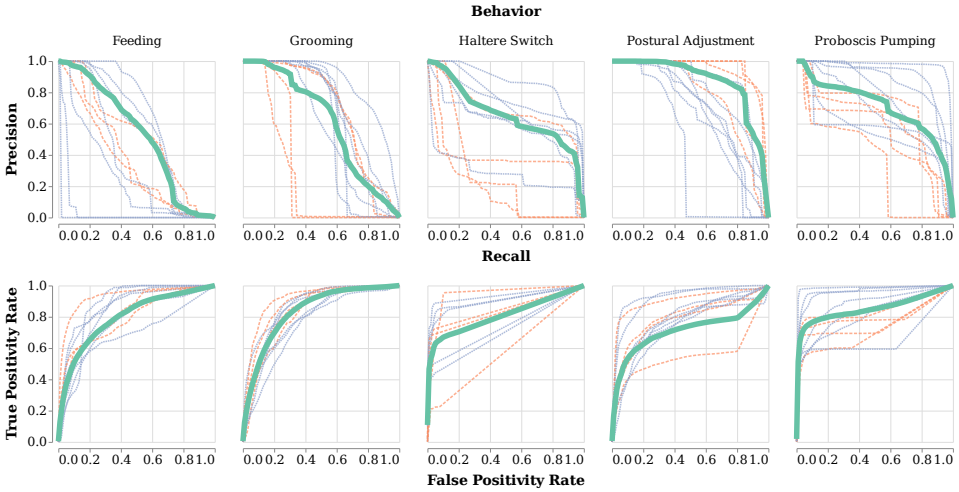
Figure 6.2: Distributions of behavioral score values of each behavioral category for all splits. Each box-plot column demonstrates the behavioral score distributions of target behaviors for the corresponding annotation.

Resulting score vectors are used for classification and also utilized as confidence scores. In the Figure 6.2, behavioral score distribution for each category is given for all splits together. Except for the feeding category, median score of the correct target behavioral category is greater than 0.5. Grooming and postural adjustment are the second most similar target behavior for each other, as expected. When the grooming behavior is short, it becomes more similar to short duration moving and postural adjustment behavior. Behavioral scores also helpful for revealing similarities and dissimilarities between behavioral categories, and can be helpful for determining spatio-temporal feature set. Although we normalized behavioral weights with number of occurrences, which favors the relatively short duration and rare switch-like haltere behavior, we can see that its target behavior score does not exceed 0.1 for wrong categories. Compared to other behaviors, feeding has the poorest performance, with median correct score 0.2. Postural adjustment category wrongly has the highest median score for feeding. Frames with feeding behavior annotation often misidentified as postural adjustment, grooming, and proboscis pumping. As proboscis movement is common both for feeding and proboscis pumping, confusion with the proboscis pumping is expected for feeding.

Using behavioral computed behavioral scores, we computed precision recall and receiver operating characteristic curves for each split, and also computed interpreted weighted averages of curves, see Figure 6.3 and Figure 6.4. Evaluations are done by considering two subsets of frames, the first subset is the frames annotated as one of the behavioral categories and the other one is the Micro-activity set. The latter one contains false positive micro-activity frames, which is quantified as the recall score of quiescent and other category. False positive micro-activity frames affects haltere switch detection performance most, for instance, its weighted mean precision decreases by $\sim$0.5 for fixed recall at $\sim$0.65. This performance difference also shows the importance of achieving low recall for quiescent and other category



(a) ROC and precision-recall curves for frames detected as micro-activity.

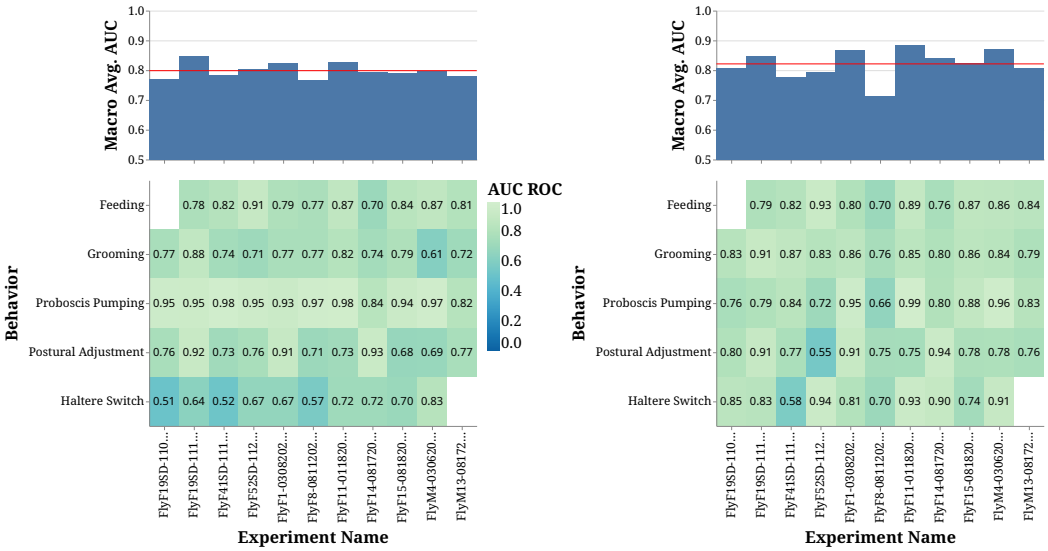


(b) ROC and precision-recall curves for annotated frames.

Figure 6.3: Performance summary of behavior mapping demonstrated using receiver operating characteristic curve and precision-recall curve. Weighted average of ROC and precision-recall curves computed by interpolation. Curves for both frames estimated as micro-activity and annotated frames are given respectively in Figures 6.3a, 6.3b.

in the activity detection stage. Since the evaluation is done with the Micro-activity set is more realistic, the following comments are made on considering it.

The most robust detection is achieved for proboscis pumping and postural adjustment behavioral categories, respectively, achieving F-1 greater than 0.8 and 0.85 scores for some splits. The maximum F-1 score achieved for grooming is 0.60 for FlyF1-03082020164520 split, whereas F-1 score of haltere switch behavioral category do not exceed 0.4. AUC scores for the proboscis pumping often exceeds 0.95, implying robust detection for almost all the splits. We observe that detection performance vary greatly among different experiments. This variation can be due to a number of reasons, but most likely reasons are related to tracking performance and



(a) AUC scores for Micro-activity set.

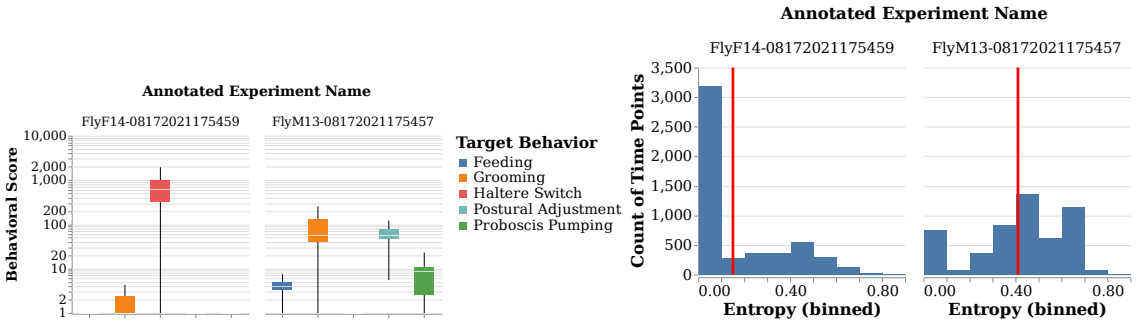
(b) AUC scores for annotated frames.

Figure 6.4: Performance summary of behavior mapping with area under curve scores of ROC. The red line indicates the macro average of AUC scores for each split. Scores for both frames estimated as micro-activity and annotated frames are given respectively in Figures 6.4a, 6.4b.

annotation quality, and as well as behavioral repertoires of flies. Considering all splits together, we achieve a macro average of 0.8 AUC score with a one-versus-rest approach.

As discussed in the Chapter 1, the behavioral repertoire of the fruit fly is much richer than the behavioral categories that we considered. Moreover, there exists a great amount of variation among flies behavioral repertoire and behavioral visits (see Section 6.2). Thus, we also investigate our pipeline's ability of detection of unseen behavioral categories by utilizing behavioral scores. Such an ability is important for avoiding misleading the user of the pipeline. It may also be desired to reconsider defined and annotated behavioral categories, and may further lead to interesting biological observations. To this end, we consider the following experimental setting: behavior mapping of an unannotated experiment (FlyF1-03082020) is done separately for two annotated experiments, namely FlyF14-0817202 and FlyM13-08172021. Then, behavioral scores of the FlyF1-03082020's frames with haltere switch annotation is compared between FlyF14-0817202 guided mapping and FlyM13-08172021 guided mapping. As it can be seen from the Figure 6.8c, FlyF1-03082020 and FlyF14-0817202 spent $\sim$4 minutes by exhibiting switch like haltere movement behavior, whereas FlyM13-08172021 spent no more than 5 seconds. Therefore, compared to FlyF14-0817202, haltere switch is an unseen and lacking behavioral category for FlyM13-08172021. We expect to identify this using behavioral scores. Indeed, behavioral scores reflect the difference in the behavioral

repertoires of the annotated flies. As it can be seen in the Figure 6.5a, behavioral scores obtained based on FlyF14-0817202's behavioral repertoire are much more confident about the correct behavioral category, whereas behavioral scores obtained based on FlyM13-08172021's behavioral repertoire are more uniform-like and total score is often shared among multiple behavioral categories. Using the entropy of the behavioral score, one can quantify existing of unseen and new behavioral categories in a more systematic way. In the Figure 6.5b, we compare entropy distributions of the behavioral scores obtained using annotated experiments FlyM13-08172021 and FlyF14-0817202. We can immediately see that scores obtained based on the FlyM13-08172021's repertoire have higher entropy, with mean 0.41, whereas for the other one mean is 0.15. For instance, if the entropy of a frame's score is greater than 0.2, it may be recommended to visit them and investigate the reason. Accordingly, behavioral annotations can be extended.



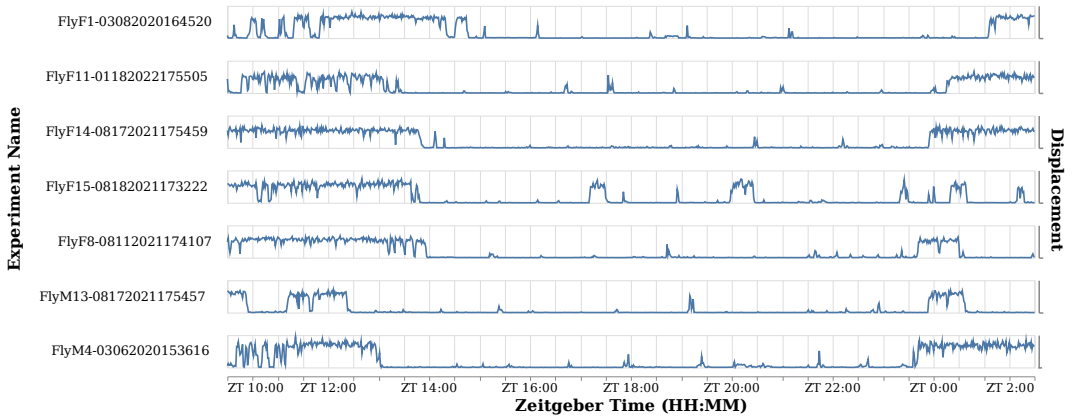
(a) Behavioral score values, before L-1 normalization.

(b) Entropy values. The red line, is the mean.

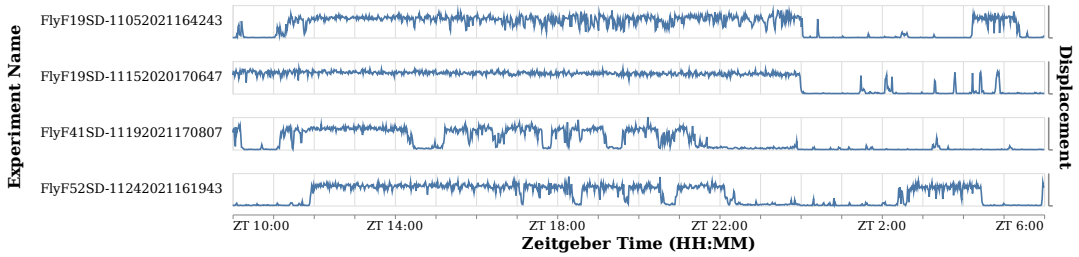
Figure 6.5: Histogram of entropy values and box-plot of behavioral scores computed using one unannotated and two annotated experiments with varying behavioral repertoires. Here, behavioral repertoire of FlyF1-03082020 is predicted separately with two different annotated experiments, namely FlyF14-08172021 and FlyM13-08172021. Behavioral scores and entropy values are computed for the haltere switch behavior. The latter one lacks the haltere switch behavior, and as a result behavioral scores tend to have higher entropy. Results demonstrate the ability of the proposed pipeline to detect and discover unseen unannotated behavioral categories using behavioral scores.

6.2 Analyzing Behavioral Repertoires

In this section, we analyze the collected data of sleep experiments to discover how behaviors that we are interested in exhibited. Starting from a broad perspective, the most brief description of sleep experiments can be done by quantifying the displacement of the flies placed in the chamber. In the Figure 6.6, we plot velocity-based feature vectors $\mathbf{u}$ between ZT10-ZT02 and ZT10-ZT06, respectively for each wild type sleep and sleep-deprived experiments. We observe that the long dormancy



(a) Wild type.



(b) Sleep-deprived.

Figure 6.6: Overall displacement values over entire experiments. Displacement of the body reveals long dormancy and sleep epochs, and macro-activities. Displacement values are computed as described in Equation 4.1, and smoothed with a rolling mean of 1-minute window.

epochs are interrupted by relatively short macro activity bouts during the sleep. In addition to short macro activity bouts, we observe very small displacements all over the night, which indicates there exists micro-activities other than major positional and postural changes. Hence, a more fine-grained analysis and detailed examination is necessary to gain an insight, as this work attempts to achieve.

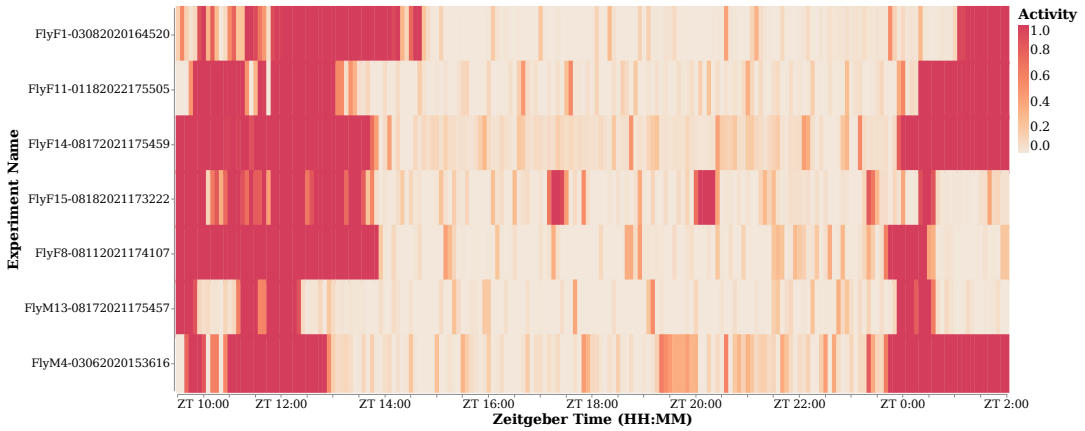
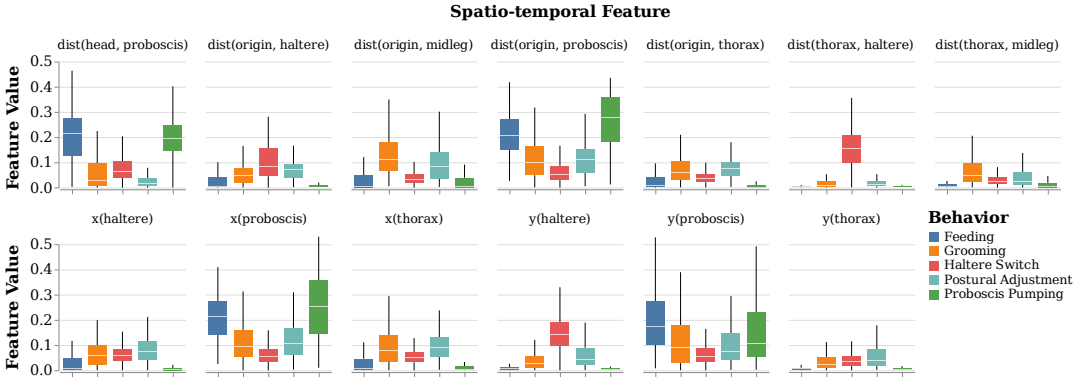
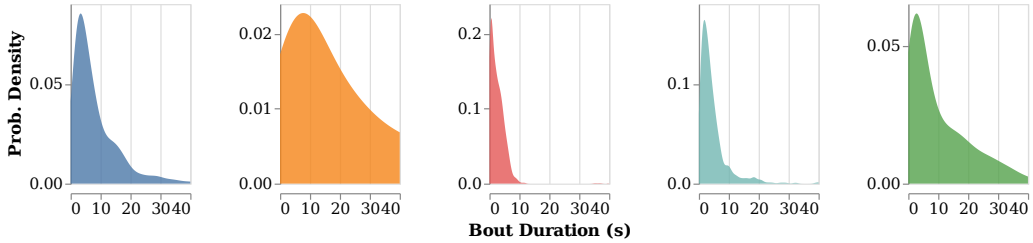


Figure 6.7: Binned temporal heatmap of activities. Each bin is 5 minutes, corresponding to 9000 frames. Activity value is computed as the ratio of the number of annotated frames and total number of frames in that bin.

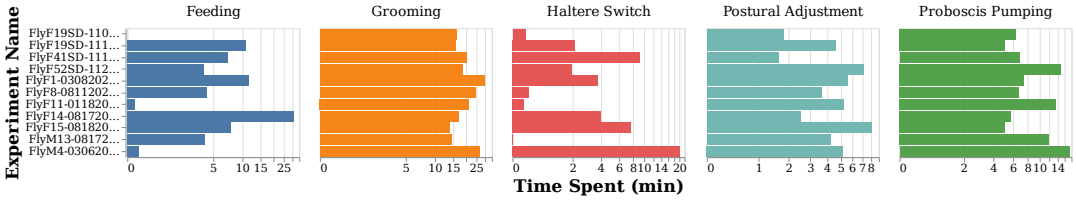
Using annotated behavioral categories, we further examine more closely how activities manifested during the sleep. In the Figure 6.7, experiments are temporally divided into 5 minute bins, and activity quantified as the ration between number of annotated frames to total number of frames (900 frames). Such a quantification provides us a view on the sleep which demonstrates that many activities occur without the displacement of the fly's body.



(a) Summation of all frequency channels of behavioral representation value for each spatio-temporal feature.



(b) Kernel density estimations of bout durations for each behavioral category. Variance of bout durations within and across behavioral categories demonstrates the rich behavioral repertoire.



(c) Time spent while exhibiting each behavioral category for all experiments.

Figure 6.8: Summary of behavioral repertoires, demonstrated using both spatial and temporal characteristics.

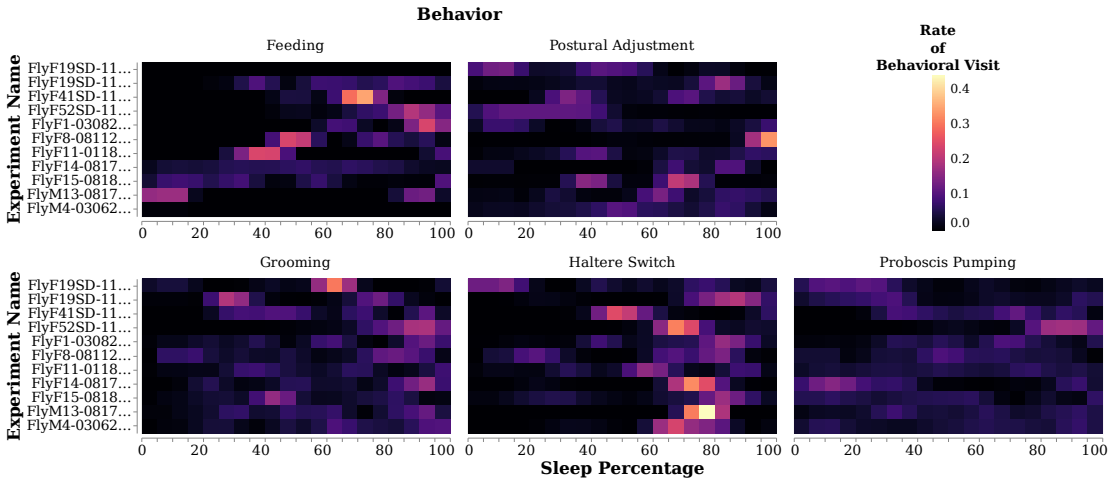
In addition to temporal organization of the activities, we are also interested in the characteristics of each behavioral categories such as total time spent, bout durations, kinematics, and feature value distributions. In order to visualize feature distributions of each behavioral category, we sum behavioral representation values of all frequency channels for each spatio-temporal feature. For a single frame, resulting values sum up to 1 as they are L-1 normalized. In the Figure 6.8a, we see how behavioral categories are defined by spatial characteristics. As expected, proboscis

related features play a significant role in characterizing feeding and proboscis pumping. Similarly, features related to leg and thorax are more apparent for grooming and postural adjustment.

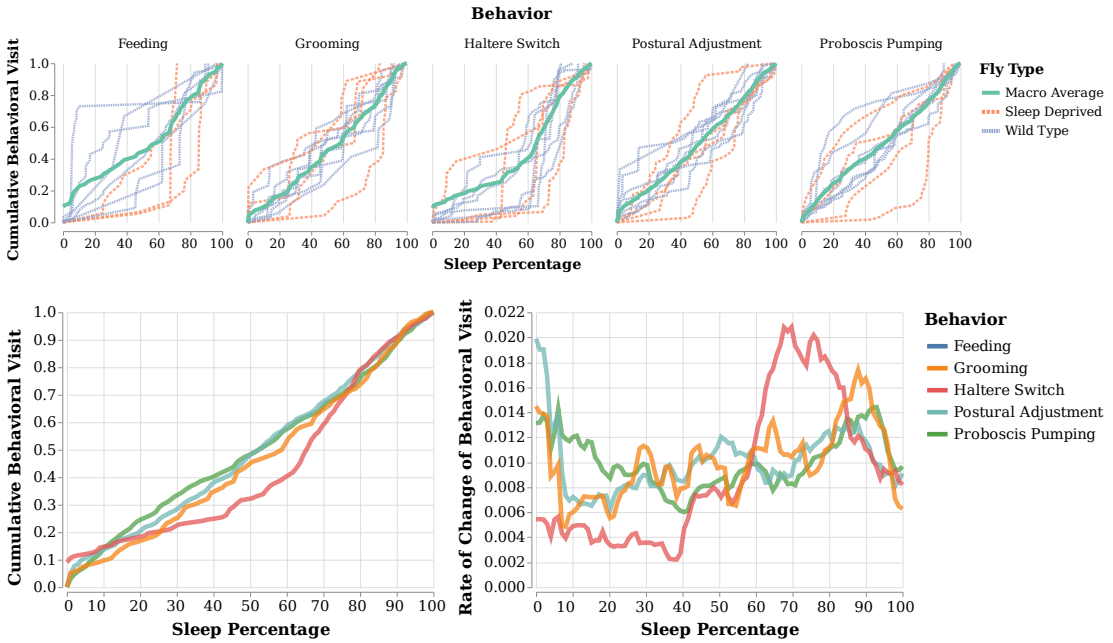
Behaviors differ in terms of bout durations as well. Kernel density estimations of bout duration distributions for each behavioral category is given in the Figure 6.8b. Haltere switch rarely exceeds several of seconds, and postural adjustments usually occur within $\sim$4 seconds, and both have relatively low variance 4.4, and 8.2 seconds, respectively. Feeding, proboscis pumping and grooming tends to have longer bouts with greater variances. For example, bout duration distribution of grooming has variance of 15 seconds, and the mean bout duration is 6.2 seconds.

Another important aspect of fruit flies' behavioral repertoires is the total time spent while exhibiting each behavior (see Figure 6.8c). Grooming is the most exhibited behavior in terms of total time spent, with an average of 19.9 minutes, and a standard deviation of 5.2 minutes. We observe a great variance for time spent exhibiting haltere switch among experiments, the average, and the standard deviation are 4.5 minutes and 5.9 respectively.

Characterization of underlying changes in behaviors is essential for understanding sleep, and thus, we also examine behavioral visits occurred during flies' sleep. To evaluate different experiments jointly, we align sleep bouts by dividing the longest dormancy epoch into bins. In the Figure 6.9, behavioral visits are plotted with a joint x-axis, namely the sleep percentages. The most immediate observation about the temporal organization of the behavioral categories is of haltere switch. Occurrence of haltere switch is not uniform along the time, for 3 sleep-deprived and 3 wild type sleep experiments 90% of the haltere switch occurrences are after 40 sleep percentage. For 8 out of 11 experiments, more than 40% of the switch-like haltere movements occurs between 60 and 80 sleep percentages. Moreover, for 4 out of 11 experiments, 80% of haltere switches are observed between this interval. Another observation is that grooming behaviors are more likely to occur in the last 40 sleep percentage, and very unlikely to occur within the first 20 sleep percentage.



(a) Rate of behavioral visits during the sleep within each 5 sleep percentage bin. Each row is normalized to sum up to 1.



(b) Cumulative behavioral visits for each behavior, behaviors with less than 2 bouts are excluded.

Figure 6.9: Demonstration of behavioral visits during the sleep. Ethograms of flies are aligned by dividing the longest dormant epoch to 100 bin, corresponding to the sleep percentages. Number of behavioral visits are normalized by the total number of bouts for each fly and behavioral category, separately.

Chapter 7

`basty`: A Software Package for Automated Behavioral Analysis of Asleep Fruit Fly

Altair: Interactive Statistical Visualizations for Python: [VanderPlas et al., 2018] PyWavelets: A Python package for wavelet analysis: [Lee et al., 2019] Scikit-learn: Machine Learning in Python: [Pedregosa et al., 2011] UMAP: Uniform Manifold Approximation and Projection: [McInnes et al., 2018]

Chapter 8

Conclusion

Bibliography

S. S. Campbell and I. Tobler. Animal sleep: a review of sleep duration across phylogeny. *Neuroscience and Biobehavioral Reviews*, 8(3):269–300, 1984. ISSN 0149-7634. doi: 10.1016/0149-7634(84)90054-x.

Ravi D. Nath, Claire N. Bedbrook, Michael J. Abrams, Ty Basinger, Justin S. Bois, David A. Prober, Paul W. Sternberg, Viviana Gradinaru, and Lea Goentoro. The Jellyfish Cassiopea Exhibits a Sleep-like State. *Current Biology*, 27(19):2984–2990.e3, October 2017. ISSN 0960-9822. doi: 10.1016/j.cub.2017.08.014. URL https://www.sciencedirect.com/science/article/pii/S0960982217310230.

Michael A. Corner. Sleep and the beginnings of behavior in the Animal Kingdom—Studies of Ultradian motility cycles in early life. *Progress in Neurobiology*, 8:279–295, January 1977. ISSN 0301-0082. doi: 10.1016/0301-0082(77)90008-9. URL https://www.sciencedirect.com/science/article/pii/0301008277900089.

S. Sauer, M. Kinkelin, E. Herrmann, and W. Kaiser. The dynamics of sleep-like behaviour in honey bees. *Journal of Comparative Physiology A*, 189(8):599–607, August 2003. ISSN 1432-1351. doi: 10.1007/s00359-003-0436-9. URL https://doi.org/10.1007/s00359-003-0436-9.

Sandeep Robert Datta, David J. Anderson, Kristin Branson, Pietro Perona, and Andrew Leifer. Computational Neuroethology: A Call to Action. *Neuron*, 104(1):11–24, October 2019. ISSN 0896-6273. doi: 10.1016/j.neuron.2019.09.038. URL https://www.sciencedirect.com/science/article/pii/S0896627319308414.

Talmo D Pereira. Quantifying behavior to understand the brain. *Nature Neuroscience*, 23:13, 2020.

David J. Anderson and Pietro Perona. Toward a Science of Computational Ethology. *Neuron*, 84(1):18–31, October 2014. ISSN 0896-6273. doi: 10.1016/j.neuron.2014.09.005. URL https://www.sciencedirect.com/science/article/pii/S0896627314007934.

Alexander Mathis, Pranav Mamidanna, Kevin M. Cury, Taiga Abe, Venkatesh N. Murthy, Mackenzie Weygandt Mathis, and Matthias Bethge. DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience*, 21(9):1281–1289, September 2018. ISSN 1546-1726. doi: 10.1038/s41593-018-0209-y. URL https://www.nature.com/articles/s41593-018-0209-y. Number: 9 Publisher: Nature Publishing Group.

Talmo D. Pereira, Diego E. Aldarondo, Lindsay Willmore, Mikhail Kislin, Samuel S.-H. Wang, Mala Murthy, and Joshua W. Shaevitz. Fast animal pose estimation using deep neural networks. *Nature Methods*, 16(1):117–125, January 2019. ISSN 1548-7105. doi: 10.1038/s41592-018-0234-5. URL https://www.nature.com/articles/s41592-018-0234-5. Number: 1 Publisher: Nature Publishing Group.

Talmo D. Pereira, Nathaniel Tabris, Arie Matsliah, David M. Turner, Junyu Li, Shruthi Ravindranath, Eleni S. Papadoyannis, Edna Normand, David S. Deutsch, Z. Yan Wang, Grace C. McKenzie-Smith, Catalin C. Mitelut, Marielisa Diez Castro, John D'Uva, Mikhail Kislin, Dan H. Sanes, Sarah D. Kocher, Samuel S.-H. Wang, Annegret L. Falkner, Joshua W. Shaevitz, and Mala Murthy. SLEAP: A deep learning system for multi-animal pose tracking. *Nature Methods*, 19(4):486–495, April 2022. ISSN 1548-7091, 1548-7105. doi: 10.1038/s41592-022-01426-1. URL https://www.nature.com/articles/s41592-022-01426-1.

Bart van Alphen, Evan R. Semenza, Melvyn Yap, Bruno van Swinderen, and Ravi Allada. A deep sleep stage in Drosophila with a functional role in waste clearance. *Science Advances*, 7(4):eabc2999, January 2021. doi: 10.1126/sciadv.abc2999. URL https://www.science.org/doi/10.1126/sciadv.abc2999. Publisher: American Association for the Advancement of Science.

Quentin Geissmann, Esteban J. Beckwith, and Giorgio F. Gilestro. Most sleep does not serve a vital function: Evidence from Drosophila melanogaster. *Science Advances*, 5(2):eaau9253, February 2019. doi: 10.1126/sciadv.aau9253. URL https://www.science.org/doi/10.1126/sciadv.aau9253. Publisher: American Association for the Advancement of Science.

Bing Qiao, Chiyuan Li, Victoria W Allen, Mimi Shirasu-Hiza, and Sheyum Syed. Automated analysis of long-term grooming behavior in Drosophila using a k-nearest neighbors classifier. *eLife*, 7:e34497, February 2018. ISSN 2050-084X. doi: 10.7554/eLife.34497. URL https://elifesciences.org/articles/34497.

Gordon J. Berman, Daniel M. Choi, William Bialek, and Joshua W. Shaevitz. Mapping the stereotyped behaviour of freely moving fruit flies. *Journal of The Royal Society Interface*, 11(99):20140672, October 2014. doi: 10.1098/rsif.2014.0672.

URL https://royalsocietypublishing.org/doi/full/10.1098/rsif.2014.0672. Publisher: Royal Society.

Jeremy G. Todd, Jamey S. Kain, and Benjamin L. de Bivort. Systematic exploration of unsupervised methods for mapping behavior. *Physical Biology*, 14(1):015002, February 2017. ISSN 1478-3975. doi: 10.1088/1478-3975/14/1/015002. URL https://doi.org/10.1088/1478-3975/14/1/015002. Publisher: IOP Publishing.

Gordon J. Berman, William Bialek, and Joshua W. Shaevitz. Predictability and hierarchy in *Drosophila* behavior. *Proceedings of the National Academy of Sciences*, 113(42):11943–11948, October 2016. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1607601113. URL https://pnas.org/doi/full/10.1073/pnas.1607601113.

Brian D DeAngelis, Jacob A Zavatone-Veth, and Damon A Clark. The manifold structure of limb coordination in walking Drosophila. *eLife*, 8:e46409, June 2019. ISSN 2050-084X. doi: 10.7554/eLife.46409. URL https://elifesciences.org/articles/46409.

Machiko Katori, Shoi Shi, Koji L. Ode, Yasuhiro Tomita, and Hiroki R. Ueda. The 103,200-arm acceleration dataset in the UK Biobank revealed a landscape of human sleep phenotypes. *Proceedings of the National Academy of Sciences*, 119(12):e2116729119, March 2022. doi: 10.1073/pnas.2116729119. URL https://www.pnas.org/doi/10.1073/pnas.2116729119. Publisher: Proceedings of the National Academy of Sciences.

André E. X. Brown, Eviatar I. Yemini, Laura J. Grundy, Tadas Jucikas, and William R. Schafer. A dictionary of behavioral motifs reveals clusters of genes affecting Caenorhabditis elegans locomotion. *Proceedings of the National Academy of Sciences*, 110(2):791–796, January 2013. doi: 10.1073/pnas.1211447110. URL https://www.pnas.org/doi/abs/10.1073/pnas.1211447110. Publisher: Proceedings of the National Academy of Sciences.

Duncan S. Mearns, Joseph C. Donovan, António M. Fernandes, Julia L. Semmelhack, and Herwig Baier. Deconstructing Hunting Behavior Reveals a Tightly Coupled Stimulus-Response Loop. *Current biology: CB*, 30(1):54–69.e9, January 2020. ISSN 1879-0445. doi: 10.1016/j.cub.2019.11.022.

Jesse D. Marshall, Diego E. Aldarondo, Timothy W. Dunn, William L. Wang, Gordon J. Berman, and Bence P. Ölveczky. Continuous Whole-Body 3D Kinematic Recordings across the Rodent Behavioral Repertoire - SI. *Neuron*, 109(3):420–

437.e8, February 2021. ISSN 08966273. doi: 10.1016/j.neuron.2020.11.016. URL https://linkinghub.elsevier.com/retrieve/pii/S0896627320308941.

Alexander B. Wiltschko, Matthew J. Johnson, Giuliano Iurilli, Ralph E. Peterson, Jesse M. Katon, Stan L. Pashkovski, Victoria E. Abraira, Ryan P. Adams, and Sandeep Robert Datta. Mapping Sub-Second Structure in Mouse Behavior. *Neuron*, 88(6):1121–1135, December 2015. ISSN 0896-6273. doi: 10.1016/j.neuron.2015.11.031. URL https://www.sciencedirect.com/science/article/pii/S0896627315010375.

Cristina Segalin, Jalani Williams, Tomomi Karigo, May Hui, Moriel Zelikowsky, Jennifer J Sun, Pietro Perona, David J Anderson, and Ann Kennedy. The Mouse Action Recognition System (MARS) software pipeline for automated analysis of social behaviors in mice. *eLife*, 10:e63720, November 2021. ISSN 2050-084X. doi: 10.7554/eLife.63720. URL https://doi.org/10.7554/eLife.63720. Publisher: eLife Sciences Publications, Ltd.

Ye Emily Wu, James Dang, Lyle Kingsbury, Mingmin Zhang, Fangmiao Sun, Rongfeng K. Hu, and Weizhe Hong. Neural control of affiliative touch in prosocial interaction. *Nature*, 599(7884):262–267, November 2021. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-021-03962-w. URL https://www.nature.com/articles/s41586-021-03962-w.

Matthew R. Whiteway, Dan Biderman, Yoni Friedman, Mario Dipoppa, E. Kelly Buchanan, Anqi Wu, John Zhou, Niccolò Bonacchi, Nathaniel J. Miska, Jean-Paul Noel, Erica Rodriguez, Michael Schartner, Karolina Socha, Anne E. Urai, C. Daniel Salzman, The International Brain Laboratory, John P. Cunningham, and Liam Paninski. Partitioning variability in animal behavioral videos using semi-supervised variational autoencoders. *PLOS Computational Biology*, 17(9):e1009439, September 2021. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1009439. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1009439. Publisher: Public Library of Science.

João C. Marques, Simone Lackner, Rita Félix, and Michael B. Orger. Structure of the Zebrafish Locomotor Repertoire Revealed with Unsupervised Behavioral Clustering. *Current biology: CB*, 28(2):181–195.e5, January 2018. ISSN 1879-0445. doi: 10.1016/j.cub.2017.12.002.

James P Bohnslav, Nivanthika K Wimalasena, Kelsey J Clausing, Yu Y Dai, David A Yarmolinsky, Tomás Cruz, Adam D Kashlan, M Eugenia Chiappe, Lauren L Orefice, Clifford J Woolf, and Christopher D Harvey. DeepEthogram, a machine learning pipeline for supervised behavior classification from raw pixels.

eLife, 10:e63377, September 2021. ISSN 2050-084X. doi: 10.7554/eLife.63377. URL https://elifesciences.org/articles/63377.

Alexander I. Hsu and Eric A. Yttri. B-SOiD, an open-source unsupervised algorithm for identification and fast prediction of behaviors. *Nature Communications*, 12(1): 5188, December 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-25420-x. URL https://www.nature.com/articles/s41467-021-25420-x.

Quentin Geissmann, Luis Garcia Rodriguez, Esteban J. Beckwith, Alice S. French, Arian R. Jamasb, and Giorgio F. Gilestro. Ethoscopes: An open platform for high-throughput ethomics. *PLOS Biology*, 15(10):e2003026, October 2017. ISSN 1545-7885. doi: 10.1371/journal.pbio.2003026. URL https://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.2003026. Publisher: Public Library of Science.

Mayank Kabra, Alice A. Robie, Marta Rivera-Alba, Steven Branson, and Kristin Branson. JAABA: interactive machine learning for automatic annotation of animal behavior. *Nature Methods*, 10(1):64–67, January 2013. ISSN 1548-7105. doi: 10.1038/nmeth.2281. URL https://www.nature.com/articles/nmeth.2281. Number: 1 Publisher: Nature Publishing Group.

Simon RO Nilsson, Nastacia L. Goodwin, Jia Jie Choong, Sophia Hwang, Hayden R Wright, Zane C Norville, Xiaoyu Tong, Dayu Lin, Brandon S. Bentzley, Neir Eshel, Ryan J McLaughlin, and Sam A. Golden. Simple Behavioral Analysis (SimBA) – an open source toolkit for computer classification of complex social behaviors in experimental animals. preprint, Animal Behavior and Cognition, April 2020. URL http://biorxiv.org/lookup/doi/10.1101/2020.04.19.049452.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, NV, USA, June 2016. IEEE. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.90. URL http://ieeexplore.ieee.org/document/7780459/.

H. E. Rauch, C. T. Striebel, and F. Tung. Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 3(8):1445–1450, August 1965. ISSN 0001-1452. doi: 10.2514/3.3166. URL https://ui.adsabs.harvard.edu/abs/1965AIAAJ...3.1445R/abstract.

Lexiang Ye and Eamonn Keogh. Time series shapelets: a novel technique that allows accurate, interpretable and fast classification. *Data Mining and Knowledge Discovery*, 22(1):149–182, January 2011. ISSN 1573-756X. doi: 10.1007/s10618-010-0179-5. URL https://doi.org/10.1007/s10618-010-0179-5.

John G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *JOSA A*, 2 (7):1160–1169, July 1985. ISSN 1520-8532. doi: 10.1364/JOSAA.2.001160. URL https://opg.optica.org/josaa/abstract.cfm?uri=josaa-2-7-1160. Publisher: Optica Publishing Group.

Christopher Torrence and Gilbert P. Compo. A Practical Guide to Wavelet Analysis. *Bulletin of the American Meteorological Society*, 79(1):61–78, January 1998. ISSN 0003-0007, 1520-0477. doi: 10.1175/1520-0477(1998)079<0061:APGTWA>2.0.CO;2. URL http://journals.ametsoc.org/doi/10.1175/1520-0477(1998)079<0061:APGTWA>2.0.CO;2.

Yonggang Liu, X. San Liang, and Robert H. Weisberg. Rectification of the Bias in the Wavelet Power Spectrum. *Journal of Atmospheric and Oceanic Technology*, 24(12):2093–2102, December 2007. ISSN 1520-0426, 0739-0572. doi: 10.1175/2007JTECHO511.1. URL http://journals.ametsoc.org/doi/10.1175/2007JTECHO511.1.

Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, October 2001. ISSN 1573-0565. doi: 10.1023/A:1010933404324. URL https://doi.org/10.1023/A:1010933404324.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011. ISSN 1533-7928. URL http://jmlr.org/papers/v12/pedregosa11a.html.

Mohammed Ali, Mark W. Jones, Xianghua Xie, and Mark Williams. TimeCluster: dimension reduction applied to temporal data for visual analytics. *The Visual Computer*, 35(6-8):1013–1026, June 2019. ISSN 0178-2789, 1432-2315. doi: 10.1007/s00371-019-01673-y. URL http://link.springer.com/10.1007/s00371-019-01673-y.

H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441, September 1933. ISSN 0022-0663. doi: 10.1037/h0071325. URL https://search.ebscohost.com/login.aspx?direct=true&db=pdh&AN=1934-00645-001&site=eds-live. Publisher: Warwick & York.

J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, March 1964. ISSN 0033-3123,

1860-0980. doi: 10.1007/BF02289565. URL http://link.springer.com/10.1007/BF02289565.

Leland McInnes, John Healy, and James Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv:1802.03426 [cs, stat]*, September 2020. URL http://arxiv.org/abs/1802.03426. arXiv: 1802.03426.

Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. ISSN 1533-7928. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

Mikhail Belkin and Partha Niyogi. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, 15(6):1373–1396, June 2003. ISSN 0899-7667. doi: 10.1162/089976603321780317. Conference Name: Neural Computation.

Jian Tang, Jingzhou Liu, Ming Zhang, and Qiaozhu Mei. Visualizing Large-scale and High-dimensional Data. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 287–297, Republic and Canton of Geneva, CHE, April 2016. International World Wide Web Conferences Steering Committee. ISBN 978-1-4503-4143-1. doi: 10.1145/2872427.2883041. URL https://doi.org/10.1145/2872427.2883041.

Tim Sainburg, Leland McInnes, and Timothy Q. Gentner. Parametric UMAP Embeddings for Representation and Semisupervised Learning. *Neural Computation*, 33(11):2881–2907, October 2021. ISSN 0899-7667. doi: 10.1162/neco_a_01434. URL https://doi.org/10.1162/neco_a_01434.

Sameer A Nene, Shree K Nayar, and Hiroshi Murase. Columbia Object Image Library (COIL-20). Technical Report CUCS-005-96, Columbia University, February 1996. URL http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms, September 2017. URL http://arxiv.org/abs/1708.07747. Number: arXiv:1708.07747 arXiv:1708.07747 [cs, stat].

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, Red Hook, NY, USA, December 2013. Curran Associates Inc.

E. Hellinger. Neue Begründung der Theorie quadratischer Formen von unendlichvielen Veränderlichen. *Journal für die reine und angewandte Mathematik*, 1909(136):210–271, July 1909. ISSN 1435-5345. doi: 10.1515/crll.1909.136.210. URL https://www.degruyter.com/document/doi/10.1515/crll.1909.136.210/html?lang=en. Publisher: De Gruyter.

Wei Dong, Charikar Moses, and Kai Li. Efficient k-nearest neighbor graph construction for generic similarity measures. In *Proceedings of the 20th international conference on World wide web - WWW '11*, page 577, Hyderabad, India, 2011. ACM Press. ISBN 978-1-4503-0632-4. doi: 10.1145/1963405.1963487. URL http://portal.acm.org/citation.cfm?doid=1963405.1963487.

Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, September 1975. ISSN 0001-0782. doi: 10.1145/361002.361007. URL https://doi.org/10.1145/361002.361007.

Jacob VanderPlas, Brian E. Granger, Jeffrey Heer, Dominik Moritz, Kanit Wongsuphasawat, Arvind Satyanarayan, Eitan Lees, Ilia Timofeev, Ben Welsh, and Scott Sievert. Altair: Interactive Statistical Visualizations for Python. *Journal of Open Source Software*, 3(32):1057, December 2018. ISSN 2475-9066. doi: 10.21105/joss.01057. URL https://joss.theoj.org/papers/10.21105/joss.01057.

Gregory R. Lee, Ralf Gommers, Filip Waselewski, Kai Wohlfahrt, and Aaron O'Leary. PyWavelets: A Python package for wavelet analysis. *Journal of Open Source Software*, 4(36):1237, April 2019. ISSN 2475-9066. doi: 10.21105/joss.01237. URL https://joss.theoj.org/papers/10.21105/joss.01237.

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. UMAP: Uniform Manifold Approximation and Projection. *Journal of Open Source Software*, 3(29):861, September 2018. ISSN 2475-9066. doi: 10.21105/joss.00861. URL http://joss.theoj.org/papers/10.21105/joss.00861.