Detection, Tracking and Quantification of Grooming vs Non-Grooming Behaviour in Mice

Devrat Singh devrat@kth.se
Yuqi Shao yshao@kth.se
Vittorio Pellegrini vpel@kth.se
Rebwar Ali Omer Bajallan raob@kth.se

Abstract

The study of self-grooming behavior of mice on a neural level is important as the mice serve as models for us humans, and this therefore gives us a better understanding of the neuroscience of human behaviors. Using computer vision based frameworks to study behavior has become increasingly popular, as it allows for efficient analysis of videos displaying certain behavioral patterns. However, many of the approaches using these frameworks lack generalizability, and only works accurately given the same experimental settings. In this work we explore the ability to model mice self-grooming behavior using the computer vision based framework DeepLabCut for trajectory extraction in combination with machine learning methods for trajectory classification. In particular, the trajectories extracted from using DeepLabCut are annotated and are modeled in a supervised manner using both an LSTM network and a TDA approach. We show that we are able to accurately distinguish the grooming trajectory sequences from the non-grooming ones which is evident given the evaluation of the LSTM network.

1. Introduction

The study of behavioral patterns is essential for the understanding of the brain [13]. A goal of neuroscience is therefore to understand the relationship between behavior and neural activity [6].

Self-grooming is a complex innate behavior represented by a sequence of action patterns and is the most frequently performed behavioral activity in rodents. The self-grooming behaviour is involved in the rodent hygiene maintenance and physiological processes such as thermoregulation, social communication and de-arousal. The study of self-grooming in rodents is potentially useful for translation neuroscience research given that humans also involve in self-grooming behavior during stressful situations or in the case of certain neuropsychiatric disorders. Being able to understand the self-grooming behavior in rodents can therefore serve as a baseline that can be used to understand the neural basis of the self-grooming behavior in humans, both in normal conditions and in the case of neural disorders. The understanding of the neural circuits involved in the self-grooming activity is therefore of great importance, and is a way for us to get better knowledge of behavior, neural disorders and their potential treatments [2].

Given the recent advances in computer vision behaviour can be studied with the help of markerless pose estimation tools, which notably simplifies the analysis of behavioural patterns and allows for high accuracy pose estimates [13]. A pose estimation tool that has become increasingly popular is DeepLabCut, which by the help of transfer learning principles and their convenient interface allows for easy and efficient pose estimation tracking.

Although automated systems such as DeepLabCut provides quick and efficient pose estimation tracking services, most of them have proven to only be able to recognize behaviors displayed in the exact same setting as the training setting. Which in practice is good if you only focus on standardized testing scenarios, where the mice and the cages are representative of the training setting. In practice however the this is not the case, and both the mice acts very spontaneously and the cage or video settings might be different [15].

An important practice when developing algorithms for rodent or mice behavior detection is therefore to analyse the ability for the pose estimation tool and the resulting behavior detection models to generalize over multiple testing settings.

2. Related Work

The self-grooming behavior of mice has previously been studied by [14], where the authors analysed self-grooming as sequential execution of 5 discrete action phases in order to draw conclusions about the neural circuits involved in the self-grooming phases. The authors modelled the transition probabilities between the grooming phases using Markov models, along with modelling the termination probabilities of the grooming sequences. Additionally the authors showed that the neurons in cortical and striatal circuits together are involved in the encoding of the action sequences

displayed in the grooming phases.

The authors of [10] proposed of a method of using a convolution recurrent neural network (CRNN) in order to detect scratching behavior of mice. The scratching behavior was induced by injecting a chemical acid into the back of the mouse. Videos of the mice were then recorded, and each image frame was manually labelled as scratching or non-scratching. The CRNN was then trained in a supervised manner using the labeled data and evaluated using a hold-out dataset. The CRNN model could accurately distinguish between scratching and non-scratching, and obtained a sensitive (recall) score of 81.6% along with a positive predictive rate (precision) of 87.9%. The major distinguishing part of the proposed method compared to ours, is that they trained the CRNN model directly on the images while we trained our models on the trajectories obtained by DeepLabCut.

The authors of [5] proposed using Topological Data Analysis (TDA) given the objective of activity recognition using pose estimates of human body-parts. The authors examined the recognition of five activities, walking, waving, sitting bicycling and golfing. They showed that all activities accuracy could be segmented and recognized achieving an accuracy of $>$0.97% for all activities.

3. Problem Description

The goal of this study is to analyse, quantify and classify the self-grooming behaviour of mice given their movement patterns using pose estimates obtained by mice video recordings. The pose estimates will be provided by the pose estimation tool DeepLabCut, in which multiple configurations will be experimented with in order to obtain accurate pose estimate trajectories from the mice videos. The trajectories will then be used to train different statistical models with the objective of being able to distinguish between self-grooming and normal behavior.

4. Methodology

4.1. DeepLabCut

An integral component of this project is the DeepLabCut (DLC) toolbox [12][13], but before we discuss that, it is crucial to first understand the idea that motivates its use. As the title of the report implies, we are interested in detecting and tracking the Mice behaviours. The behaviors of interest such as grooming, involve complex set of movements among the body parts. Therefore, tracking just the position of the mouse as a whole, is not enough. What we need is a method that could track each and every body part of interest. Generally, when tracking distinct body parts, marker-based methods are utilized, whose tracking is dependent on the placement of reflective markers on each body part. Moreover, two main reasons discouraged the use of marker-based methods in our application.

First, the team at Karolinska Institute (KI), aims at capturing situations in which the mouse is unencumbered by any external parameters that could influence behavior. As one could imagine, placement of external markers on the body would certainly influence some change, especially if the mouse is not given the time to accommodate. Second, there is the matter of ethics, we need to take into consideration if the placement of markers would affect a mouse's well-being. For example, the nose is an important region of interest in this project, but it's also a highly sensitive area for the mouse, therefore, a marker on the nose can be harmful. Taking the above mentioned points into account, our only option was to use marker-less tracking and this is where DLC comes in.

DeepLabCut is a toolbox for non-invasive, marker-less pose tracking of animals performing different tasks. At the base of it, DLC is a modification of DeeperCut [8], which is another neural network based pose tracking method. However, DeeperCut requires thousands of frames to be labelled with body part positions. In comparison, DLC only needs a minimal training dataset (50-300 frames). DLC is able to achieve this through transfer learning, where it relies on the pre-trained weights from ImageNet [4]. By using the pre-trained ImageNet as base, DLC automatically gains the ability to detect and distinguish general objects such as dog, mouse etc. Facilitated by this, it becomes much easier for the network to learn to detect further specific body parts like mouse's feet or nose.

DeepLabcut comes equipped with several features and going into each of their details is outside the scope of this project. Regardless, the features which entailed significant use, will of-course be specified.

4.2. Data Collection and Preparation

A considerable part of this project consisted of going through cycles of data analysis and changing parameters to obtain reliable data. At the rawest form, the mouse behaviours are captured in video recordings, where a single mouse is recorded while it goes though its normal motions. It is worth noting that throughout the recording process, care is taken towards keeping the human intervention to a minimum, such that the mouse remains undisturbed.

4.2.1 General Outline

Now, prior to discussing the individual components, it would be helpful to briefly go over some crucial steps in the data collection process. Starting with the video recordings, the goal is to extract the body part locations in each frame of the video. As a result, the combined output from a video is in the form of pose trajectories. Considering that the behaviours are captured onto a 2D medium (i.e video

or frame), the trajectories are thus formed by x and y pixel co-ordinates of the tracked appendage. The reader might recognize that this is where DeepLabCut (DLC) is utilized. The next step involves establishing a DLC model that could predict these trajectories. At the very least, this requires an image dataset consisting of labelled body part locations. In our case, every image in this dataset is manually labelled. The DLC-model, after training, is then utilized to make predictions on the raw videos and finally we end up with the desired time series trajectories. Figure 1 describes this general outline with some additional components from further sections. After this, the data is modified or augmented accordingly to suit the detection and classification methods.

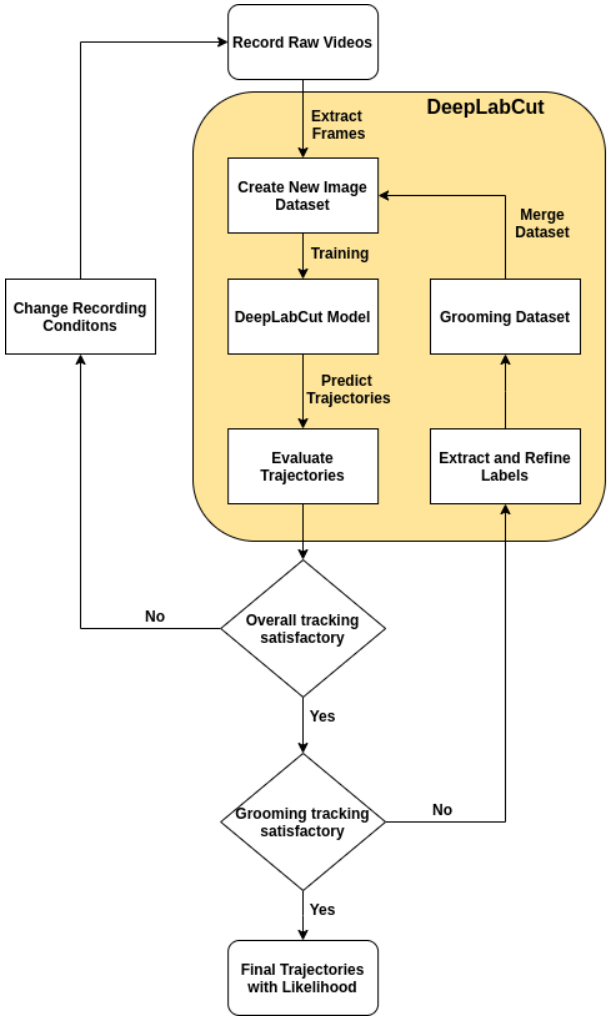


Figure 1: Data Collection Pipeline

4.2.2 Recording Raw Data

The videos for this project were recorded at the research facility in Karolinska Institute. For the record session, a single mouse is selected and then placed in a transparent glass cage. In the videos, we observed that at the beginning, the mouse is mostly moving around the cage and the grooming periods are short and far in-between. This, as we were told, could be because of the mouse's curiousness about the new environment (the cage). Once this acclimation period is over, the mouse grooms more often and these periods of grooming are much longer in duration. Another interesting fact is that most of the grooming occurrences were observed in the same corner of the cage. We are not really sure why this happens, or if this trait will persist in a different mouse. Moreover, with regard to the current project, due to above mentioned traits, you will observe that points in the grooming plots are concentrated in a single corner of the image.

Recording Conditions Going back to raw videos, the conditions in which the mice are recorded hold significant impact on the tracking results. Hence, there is merit to a discussion about the following conditions:

1. Frames per second (FPS) for recorded videos
2. Light conditions and Background
3. Camera Placement

Frame Rate Grooming has been mentioned quite often until this point, but what exactly happens during it? Although this question motivates the whole premise of this project, to put it simply, grooming involves rapid paw and head movements with the mouse repeatedly rubbing its nose using the paws (among other traits). This is where the FPS rate becomes a decisive factor. As we know, a low frame rate can result in motion blur and due to the fast paced motion of grooming, we often end up with stills in which the appearance of the paws is changed (they look blurry and stretched). This often leads to the predictions from DLC being incorrect for such frames.

Lighting The lighting serves the purpose of facilitating the visibility of appendages. Howbeit, it was outside the realm of possibility to have brightly lit recording conditions. According to our contact researcher at Karolinska Institute (KI), bright lights can stimulate stress in the mice and thus, apart from the ethical standpoint, it would defeat the purpose of this endeavour.

Camera Placement The project began with us receiving some video clips from KI. In our assessment of those videos, we found that there were instances where the mouse was grooming with its back towards the camera. Noting that the number of times a mouse grooms is much smaller as compared to its other idiosyncrasies. Therefore, every occasion of grooming is valuable data which later affects the

classification process. Such unbalancedness of behaviours and difficulties associated with it, will entail its own discussion in the later sections. Moreover, the point here is that the camera placement should be chosen such that it captures grooming as best as possible. Also, since we are mapping 3D motion (in real life) to 2D (in video), it is impossible to get completely occlusion free data. Adding to that, we cannot control where the mouse faces w.r.t the camera when grooming.

Keeping all these points in mind, we recorded the mouse from two different camera angles. Please note that even if there were two cameras recording at the same time, their videos are treated as separate entities. The idea behind the use of multiple cameras is to compare the obtained data and use the one in which the visible grooming ratio is higher. The video from the remaining angle can be used to test the generalization of our models.

Final Parameters The above insights have been realized through iterations of parameter variation and evaluation of their effects on DLC trajectories. The final classification was done on videos of the same mouse, recorded at 50 FPS (maximum for the camera provided to us at KI), with a resolution of 1920 x 1080 and close to no external light. In total, we obtained 3 hours long video data (approximately 540,000 frames) from two different camera angles. So, for each angle, we have around 1.5 hours of recording.

4.2.3 Obtaining Trajectories using DeepLabCut

Tracked Bodyparts There has been a significant mention of tracking, but which parts to track? A curious reader might remember that grooming involves nose and paws. Apart from this, the mouse also brushes the ears with its paws during the grooming phase. Hence, primarily, we are interested in tracking the nose, left ear, right ear, left paw and right paw. All these parts are generally in motion in all behaviours, but something that distinguishes grooming is that the mouse as a whole is stationary with respect to the cage. This is also reflected in the tail-base and the hind legs, which are relatively still during grooming. As a result, tracking these parts could be beneficial in detection and classification of behaviours.

The following 8 parts are tracked: Nose, Left Ear, Right Ear, Left Paw, Right Paw, Tail Base, Left Foot, Right Foot. Figure 2, illustrates these parts.

Image Dataset DeepLabCut provides the functionality to extract frames from videos and to manually label the regions of interest in all these frames. In our case, we initially extracted 320 frames from a 1 hour long video, consisting of all kinds of behaviours. These frames are randomly sampled from a uniform distribution by k-means clustering

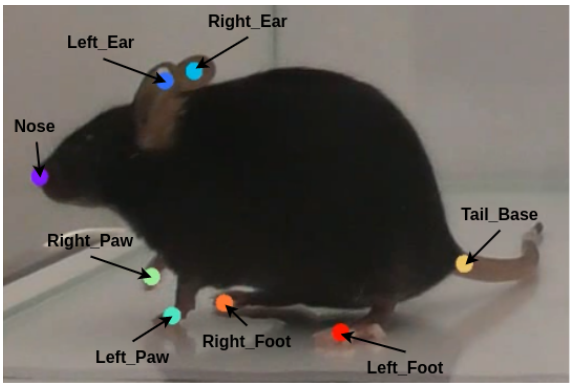


Figure 2: Tracked Bodyparts

based on visual appearance. This is done to obtain a dataset which represents the whole range of movements and not just for a single behavioral aspect. There is also the possibility to extract frames manually. Nonetheless, due to varied nature of movements throughout the video, it was fitting to use the clustering method. When it came to labelling the extracted frames, only those bodyparts which were clearly visible in a particular frame were marked and it was made sure that the labels are consistent along the whole dataset.

DLC training Naturally, the next step in the pipeline is training using the labelled dataset. For that, an ImageNet pre-trained network (for example: ResNet_50, ResNet_101 or MobileNet) should be selected. In our trials with these networks, ResNet_101 offered the best tracking results. Consequently, the final DLC model was trained using ResNet_101. The model was trained for 117500 iterations and achieved the lowest cross-entropy loss of 0.00071. The loss curve over the training iterations is shown in Figure 3. When considering the pixel error, i.e. the difference between location of predicted position and the labelled position, the model peaked at 6.37 pixel error on the train set and 20.26 on the test set. This model is then used to predict trajectories for the videos.

Evaluation and Refinement Since, we are lacking any type of ground truth about the actual location of markers, the tracking accuracy is evaluated based on human assessment. The predicted labels are plotted on top the original video, such that a video frame looks similar to the depiction in Figure 2 (Note: The label names are not displayed in the video, solely the colored dots). From our manual assessment, it was observed that the prediction were satisfactory for the general movement of the mouse in the cage but impermissible for grooming phases. The reason for this is again the grooming ratio being much smaller than other movements.

Accordingly, for better predictions during grooming, we

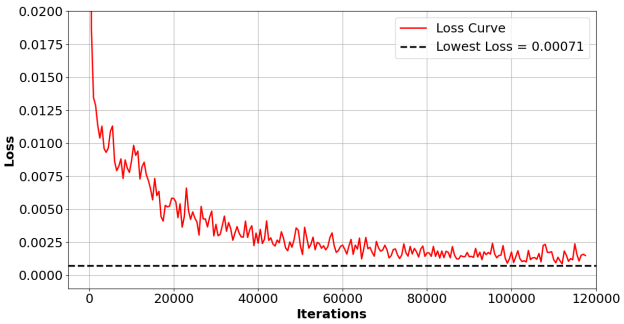


Figure 3: DeepLabCut training loss curve (Note: The loss for initial iterations (Loss at iteration_0: 0.3) have been cropped out to better illustrate the relevant details)

extracted 80 more frames from grooming periods, labelled them and added it to the previous dataset (in total, 400 labelled frames). The model is trained again for 5000 iterations, with initial weights taken from the previous training session. The above process of evaluation is repeated until satisfactory tracking is achieved during grooming.

4.2.4 Data

We have talked in lengths about the pipeline for data collection. Therefore, we will finally discuss the data we obtain after the whole process. Apart from the (x, y) pixel coordinates, there is also a third entity in DLC predictions, it's the likelihood of the marker being predicted correctly. Thus, each marker holds the following information: *[bodypart_id ,frame_id, x, y, likelihood]*. Correspondingly, if a video has N number of frames and we are tracking 8 bodyparts, then we have $(N \times 3) \times 8$ unique marker points. An example of such time series data is shown in Figure 4 for a video with approx. 68000 frames. A better glimpse of the individual trajectories from the same video is found in Figure 5.

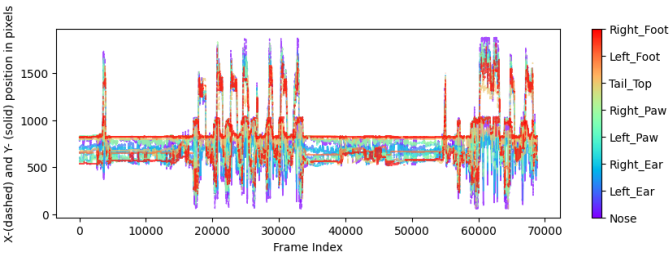


Figure 4: Predicted (x,y) for all body parts. The color map on the right hand side of this Figure shows which color represents which bodypart.

Trends in trajectories The generated trajectories when analyzed, display some noticeable trends in mouse's behaviour and given that most of our discussion has surrounded grooming, its natural that the patterns we found

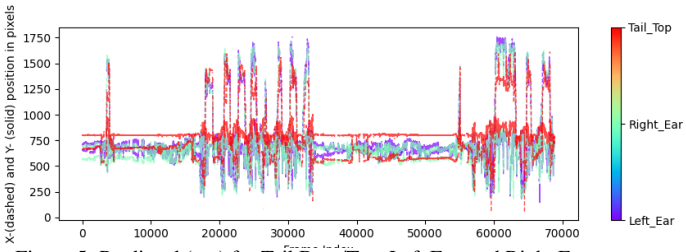


Figure 5: Predicted (x,y) for Tail Base/Top, Left Ear, and Right Ear

are primarily related to it. In addition, for this section, in order to have a balanced representation of grooming versus non-grooming, the ensuing figures are based on a video clip which fulfils the said criteria.

Throughout the video data, it was noticeable that the mouse almost always grooms in the same corner of the cage. This attribute is evident when the mouse locations are plotted simultaneously for all frames (Figure 6) and we observe a concentration of red dots in one corner of the cage.

When the trajectories are plotted in 3D (Figure 7), we can notice two distinct looking clusters of points. The planar-like point clusters associate with the duration in which the mouse is moving around the cage and as a result the points are spread all across the xy-plane. On the other hand, there are also these distinct tube-like clusters in-between the before-mentioned planner ones. These tube like regions subscribe to the periods in which the mouse is stationary (in particular, positioned at that cage corner mentioned in the previous paragraph). When the mouse is relatively stationary, that's when the grooming happens. However, one should be careful and take into account that not all stationary behaviours are grooming. So, within these tube clusters, there are points associated to grooming, but not all of it is grooming.

For each of the two mentioned clusters, 1000 consecutive frames (20 seconds long period) are extracted. These specifically associate to grooming (Figure 8) and moving around or non-grooming (Figure 9).

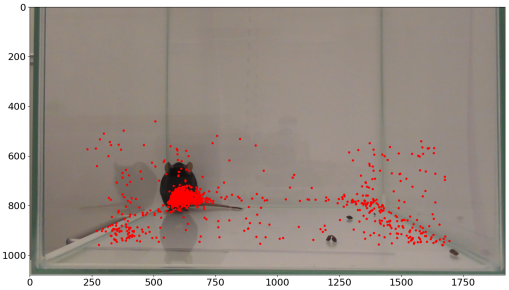


Figure 6: Plotted Centers over the entire duration of the video. The red dots represent the location of mouse's center of mass, calculated by taking mean of the predicted body part positions for each frame. The background image is a still from the video and has been added to provide some context

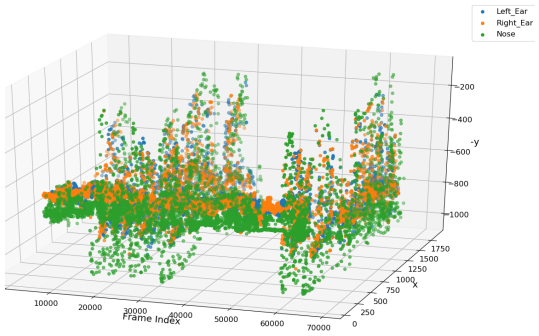


Figure 7: Left Ear, Right Ear, and Nose trajectories plotted across the whole video duration. The remaining body parts and some intermediate points have been omitted to avoid crowding and to maintain a readable plot

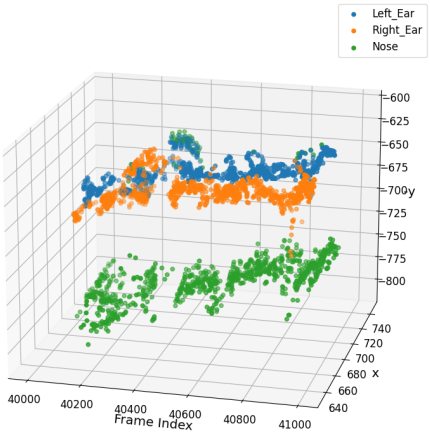


Figure 8: Grooming segment: The idea here it to show that the answer for quantification of grooming may not lie in the analysis of the full trajectories but within these short segments from this seemingly chaotic movements. This segment covers 1000 consecutive frames, which is equal to 20 seconds of video recording

4.3. Data Analysis using TDA

The field of topological data analysis (TDA) in recent years has gained increasing attention. It has the ability to discover geometrical invariants in high-dimensional, unstructured and noisy data sets. Comparing to deep learning models, the biggest advantage of TDA is associated with its non-black-box property. The representations from features are more explainable. Specifically, we applied the method persistent homology with Vietoris-Rips filtration in TDA to analyse grooming segments through their topological signatures.

Point-cloud data are needed to apply methods from TDA. In our method, the distance space (x_i, d) is transformed into simplicial complexes parameterised by $t \in [0, \infty)$ by applying Vietoris-Rips filtration, $VR_t(d)$. From this geometrical representation of (x_i, d) one can identify the number of partitions, cycles etc. at different scales t, by applying the homology functions H_n inferred from algebraic topology. Homology H_n at an increasing level of filtration t can be represented as bar codes $\{(a_i < b_i)|i = 1, ..., r\}$, characterizing the birth and death time of topological features. Finally, the topological signatures are computed by stable ranks as follows:

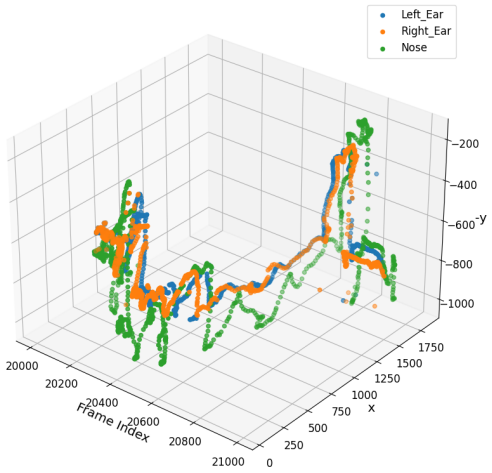


Figure 9: Non-grooming segment, where the mouse is moving around the cage. This segment is equal in length to the grooming segment from Figure 8, yet, here a clear trajectory of motion can be detected.

$$\widehat{\text{rank}}(t) = \{\text{number of bars s.t. } b_i - a_i \geq t\} \quad (1)$$

It can be proved that stable ranks are piece-wise constant non-increasing functions, so their similarities between each other can be assessed by interleaving distances:

$$d_{\bowtie}(f, g) = \inf\{v | f(x) \geq g(x+v) \text{ and } g(x) \geq f(x+v) \text{ for any } x\} \quad (2)$$

The grooming trajectory is segmented by a sliding window of size 2 seconds (100 frames) with 1 second overlap. For each grooming window, we formulated a point cloud of 16 points corresponding to x, y coordinates of the 8 body-part markers, in 100 dimensional space. To capture the movement representation in every grooming segment, we consider correlation metric as pair-wise distance between points in the point cloud,

$$d(x, y) = 1 - \frac{x_c \cdot y_c}{\|x_c\| \|y_c\|} \quad (3)$$

To enrich the effects of negative correlations, a modified correlation metric is also experimented.

$$d(x, y) = 1 + \frac{x_c \cdot y_c}{\|x_c\| \|y_c\|} \quad (4)$$

In our approach only homology H_0 is considered since many grooming segments do not have H_1 or higher homologies. The assumption is that similar grooming patterns have similar pair-wise correlations between body parts, thus resulting in similar topologies. For example, the persistence bar codes of two similar grooming segments and two rearing segments are shown as follows.

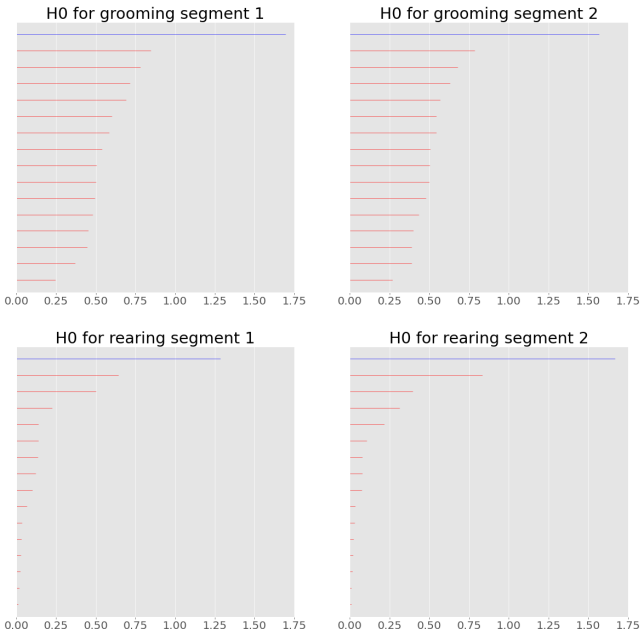


Figure 10: Persistence bar codes for mouse behaviours

The resulting persistence bar codes reveal similarities and dissimilarities for same behaviours and different behaviours respectively. This implies that analyzing their stable ranks encoding of these bar codes is a promising research direction.

We studied the topological signatures by adapting a Support Vector Machine (SVM) classifier with homology stable rank kernel, as well as K-nearest-neighbours taking interleaving distances between stable ranks. The homology stable rank kernel function for SVM is introduced in a newly published paper [1]:

$$K_d(X,Y) = \int_0^\infty \widehat{\text{rank}}_d(X)\widehat{\text{rank}}_d(Y)dt \qquad (5)$$

Both raw trajectories and ARIMA filtered trajectories with AR degree 1 and MA degree 2 are tested with the models. To take account of imbalanced classes, we used SVM with class weighting and KNN with subsampling of equal amount of non-grooming data as grooming.

4.4. LSTM-based approach

We decided to tackle the problem by exploiting a deep learning technique, suitable for time-series. Long Short Term Memory [7] networks have shown astonishing results in several fields. Few works ([15], [11]), related to behavior recognition, have made use of LSTM, proving its effective capability to model such task.

Our approach consists in a many-to-many architecture. Each input sequence represents a fixed number of frames. A single frame represents a temporal input and it is represented by 24 numerical data, where each marker (8 in total) is represented by a triplet: 2-dimensional coordinates (*x, y*) and *likelihood* value. A single sequence can be intended as the concatenation of such information from contiguous frames. The objective of the network is to predict the behavior associated to each frame.

The architecture consists of several blocks, here reported:

1. **a bidirectional LSTM encoder** which takes a sequence of fixed length as input and gives a sequence of hidden states as output;

2. **an attention mechanism [3]**: it takes the hidden states from the encoder and gives a weighted sum of the hidden states as output: the so-called **context vector**;

3. **a LSTM decoder**: it takes the current **context vector** as input and produce a hidden state as output;

4. **a Fully Connected Neural Network**: this takes each distinct hidden state (from the decoder) and produces a vector of length 2. Bear in mind that the latter vector is computed for each hidden state, separately;

5. **a Logarithmic Softmax block**: this is the last block and it is intended to produce the two logarithmic probabilities for each frame in the input sequence.

The number of input frames inside the sequence is equal to the number of logarithmic probability vectors (each of length 2) produced by the network.

In figure 11 is reported the schema of our architecture.

Inside the figure is also reported a **pre-process block**: this represents a relevant aspect of our development process, since we tested several pre-processing techniques and evaluated which one was the most effective for our task. The evaluation results will be treated in Section 5. Here, we introduce the different pre-processing strategies considered for our task:

1. **centering with respect to frame center**: all the markers in a frame are centered around the center of that frame. This means that the centering is done without considering statistics from other frames. As one may be think, this procedure may cause loosing of temporal relationships, leading to worse results;

2. **centering with respect to sequence center**: in this case, the markers are centered around the center of the

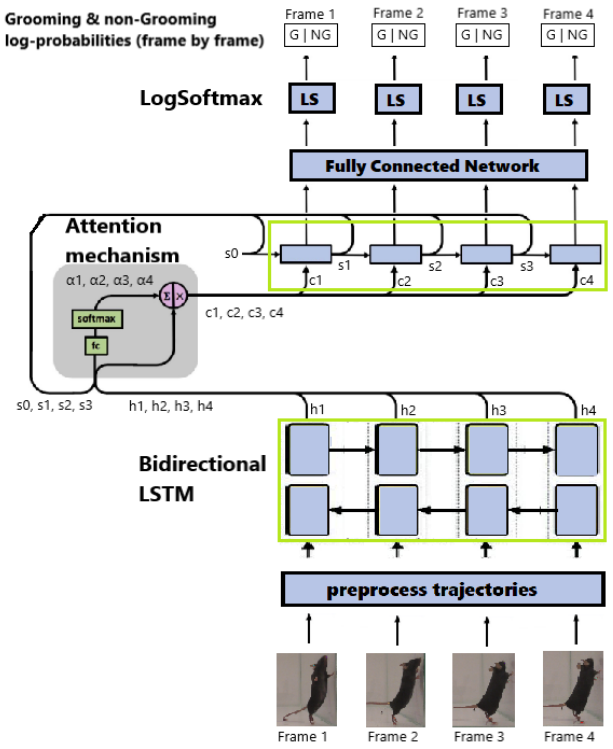


Figure 11: Schematic representation of the LSTM-based architecture. Each **context vector** c_i produced by the attention mechanism is the result of a weighted sum of the encoder hidden states. The weights $(\alpha_1, \alpha_2, ..., \alpha_{sequenceLength})$ are automatically learnt by the network. Furthermore, they depend by the encoder hidden states and the hidden state generated by the decoder in the previous temporal instant. See [3, Figure 1] for a clear understanding

whole sequence. This procedure relies on sequence-based statistics and it is then related to the length of the sequence;

3. **standard normalization**: applied separately to each distinct marker coordinate and likelihood, using the training set statistics for each feature.

After this initial phase, we further investigated the impact of the sequence length. We evaluated several sequence lengths for this purpose.

The network has been trained with Adam optimizer, with a learning rate of 10^{-4} and a weight decay of 10^{-4}. The latter, along with a dropout of 0.3, has been inserted in order to prevent overfitting.
The loss function defined for our network is the **negative log likelihood**, which takes as input the log probabilities produced by the network and the ground truth label associated to each frame.

5. Experimental Evaluation

[1] For all the experiments we made use of the same train-evaluation-test split. In particular, we used recordings from the same angle (right-side camera). We used 2 videos:

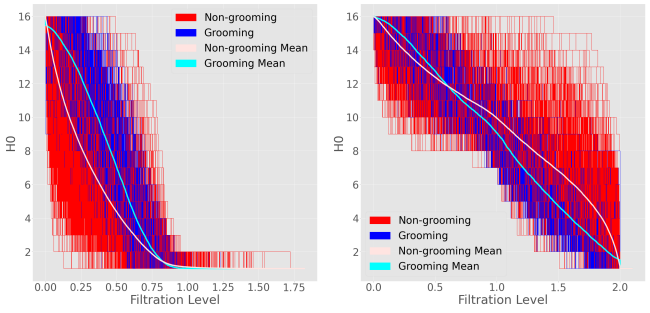
- one of length 1 hour, 6 minutes and 13 seconds
- and one another of length 22 minutes and 57 seconds.

In particular, these are our sets:

1. **train set**: trajectories extracted with DLC from the first video, considering the first 46 minutes and 40 seconds;
2. **evaluation set**: trajectories extracted with DLC from the first video, considering the remaining 19 minutes and 33 seconds;
3. **test set**: trajectories extracted with DLC from the second video.

5.1. Result Analysis for TDA approach

It is found that for default correlation metric H_0 with single linkage has highest classification performance, and for modified correlation metric H_0 with complete linkage has highest classification performance. This is because both of these settings use significantly the large amount of positive pair-wise correlations between body parts for running and rearing behaviours. Since the pair-wise distances for points in the point cloud depends on pair-wise correlations directly, it results in features for running and rearing segments merge quickly in filtration process of default correlation metric setting, and vise versa for modified correlation metric setting. This makes the topological signatures more distinct between bahaviours containing rearing and running and behaviours not containing them.



(a) H_0 with default correlation metric, single linkage

(b) H_0 with modified correlation metric, complete linkage

Figure 12: H_0 Stable Ranks of training data with ARIMA filtering

The above figures show clearly the concentration areas of grooming and non-grooming stable ranks, as a result of

[1] https://github.com/vittoriop17/mouse_behavior

non-grooming behaviours contain lots of rearing and running. The topological differences for different behaviour categories induced by negative pair-wise correlations are not visually distinguishable.

In the table below, 'f1(G)' and 'f1(NG)' denote the f1 score for grooming and non-grooming respectively. 'weighted avg' denotes the weighted accuracy. After some experiments it is found that there is no significant differences on the number of neighbours chosen in k-NN algorithm, whereas k=9 neighbours has slightly better performance overall.

Table 1: H_0 with default correlation metric, unfiltered trajectory

	SVM		9-NN	
	Train	Validation	Train	Validation
f1(G)	0.59	0.47	0.83	0.47
f1(NG)	0.84	0.78	0.81	0.79
weighted avg	0.78	0.71	0.82	0.72

Table 2: H_0 with default correlation metric, filtered trajectory

	SVM		9-NN	
	Train	Validation	Train	Validation
f1(G)	0.57	0.49	0.83	0.50
f1(NG)	0.83	0.79	0.83	0.79
weighted avg	0.77	0.72	0.83	0.73

Table 3: H_0 with modified correlation metric, filtered trajectory

	SVM		9-NN	
	Train	Validation	Train	Validation
f1(G)	0.45	0.47	0.79	0.46
f1(NG)	0.75	0.72	0.78	0.76
weighted avg	0.69	0.67	0.78	0.69

The classification did not significantly improve before and after ARIMA filtering. Slight improvements on generalization of grooming have been observed. Best model is 9-NN applied on ARIMA filtered trajectories, with default correlation metric. We applied this model to the test data, and the f1 score for grooming reached a similar score as validation data. Both of them are around 50%.

Table 4: H_0 with default correlation metric, filtered trajectory

	9-NN
	Test
f1(G)	0.48
f1(NG)	0.90
weighted avg	0.86

There is only a small portion of grooming signatures which is linearly sparable in the homology stable rank kernel transformed space. The large improvement in f1 score of grooming in training data with kNN is due to that only a subsample of non-grooming data is considered, which lowers down the number of false positives significantly hence increases precision a lot. Classification using kNN achieved better performance, revealing that similar movements from grooming and non-grooming form clusters.

A drawback of this method is that pair-wise correlations are variant to rotations, so it leads same grooming behaviours facing different directions have quite different pair-wise correlations between some points. Further research should look into this issue.

5.2. Result Analysis for LSTM approach

First of all, we evaluated the best pre-processing technique, using the test-evaluation split. All the hyperparameters have been kept fixed for all the 3 executions. In particular, we kept a sequence length of 200 frames (having 50 fps, this means that a single sequence represents an excerpt of 2 seconds).

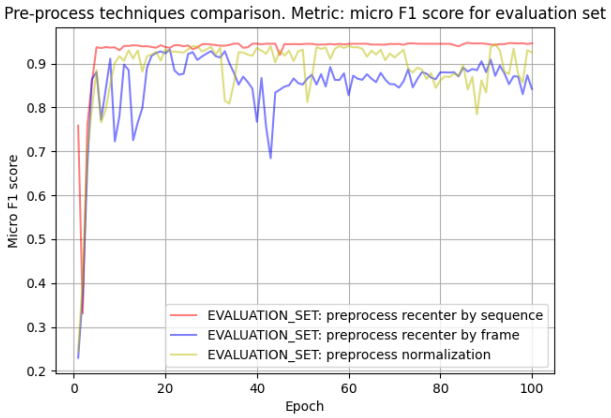


Figure 13: The plot shows the evolution of the evaluation scores during training, epoch by epoch. The trained model is evaluated in terms of f1 score.

Looking the plot 13, it is evident that the **centering with respect to the sequence center** gave the best results. After that, we moved our attention to the tuning of the sequence length: this hyper-parameter has an obvious impact to the training, since it affects the centering technique.
The evaluation results for the sequence length are reported in figures 14 and 15. The latter reports the results from the 50th to the 100th epoch. From the second figure it is possible to spot the differences between the different sequence lengths. The best value is 200 (5 seconds sequences). It is a reasonable result: a sequence length too short (like 50) may lead to loose relevant temporal information. Instead a sequence length too large may cause the inclusion of noisy information.

Once defined all the value of interest, we tested our model. This phase has been done in the following way and

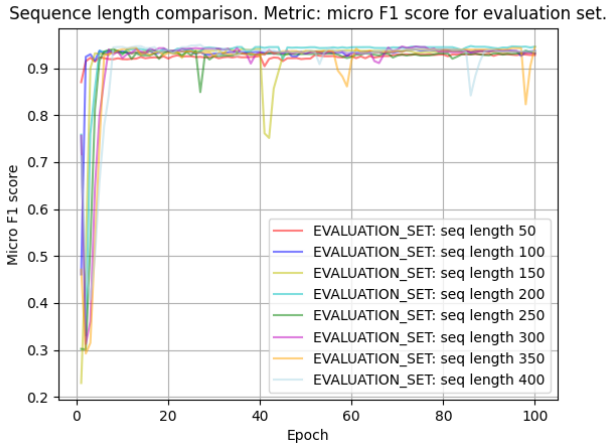


Figure 14: Evaluation scores in terms of F1 score. Each line is associated to a distinct sequence length.

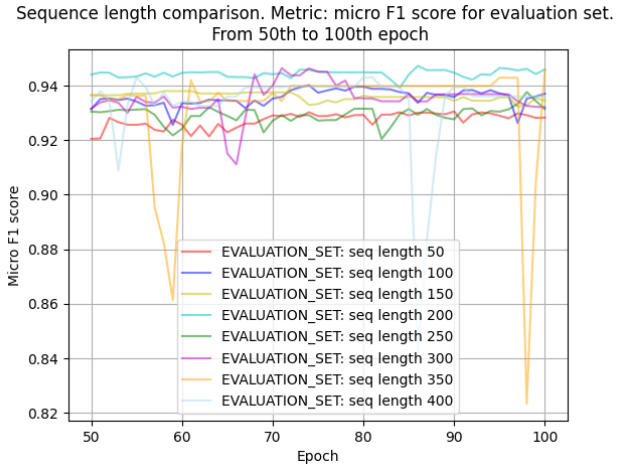


Figure 15: Evaluation scores in terms of F1 score. Each line is associated to a distinct sequence length. Only the scores from the 50th to the 100th epoch are reported, in order to simplify the visualization

with the following setting:

- we used the recentering with respect to the sequence and a sequence length of 200;
- then, we trained the network using both: training and evaluation sets. We trained the network for 100 epochs;
- we picked the model that reached the best (highest) grooming F1 score for the set used during training;
- eventually, we tested the best model, using the unseen test set.

The results are reported in figure 16. The results are also reported in terms of F1 scores for each distinct set: these can be found in table 5

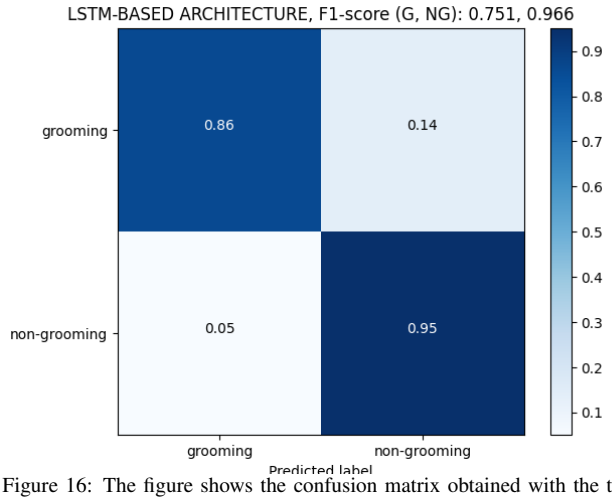


Figure 16: The figure shows the confusion matrix obtained with the the test set. The confusion matrix is reported in relative terms, due to the high imbalance between the two classes

Table 5: Results obtained with the best LSTM model, for each distinct set

	LSTM		
	Train	Validation	Test
f1(G)	0.9	0.87	0.75
f1(NG)	0.97	0.96	0.97
weighted avg	0.95	0.93	0.94

6. Summary and Discussion

Followed by behaviour trajectories extraction from DeepLabCut, we implemented two approaches for trajectories modelling. The approach with LSTM achieved better prediction accuracy for grooming vs non-grooming behaviours, benefiting from the high approximation strength of deep learning models. However, the approach with persistent homology in TDA has the advantage of being more explainable and non-blackbox-ish. It produces more direct representations of behaviours following solid mathematical proofs.
In order to easily assess the comparison between the aforementioned techniques, the results for the test set have been reported in table 6.

Table 6: Test results obtained with TDA-based and LSTM-based approaches

	9-NN	LSTM
	Test	
f1(G)	0.48	0.75
f1(NG)	0.90	0.97
weighted avg	0.86	0.94

6.1. Lessons learned

A very important step in the project was the collection of high-quality mice videos to which we could apply

DeepLabCut and obtain accurate trajectories representing the movements of the mice body parts. This also became a drawback, since the videos we first obtained to work with was of lower quality and we iteratively had to reach out to Fisone-lab at KI in order to adjust the videos to increase their quality. The final videos we worked with were videos we ourselves recorded at site. This resulted in a delay of modeling part of the project, and a lesson learned for further projects is to focus on getting high-quality data fast in the beginning on the project since this will determine how fast you will be able to start modeling the data and also determine the quality of your final models.

6.2. Future Work

In the future, it would be good to test different experimental settings for video data collection, such as using a different mouse, combining videos from different angles to obtain a 3D trajectory etc. We were limited by 50fps, which has motion blur problems. Therefore a higher frame rate can be tested in the future.

The future work with the approach of TDA involves designing a new coordinate system for mouse which captures angle invariance and reflexivity. The designation of this new coordinate system should aim at ensuring the pair-wise correlations between body parts are the same irrespective to which direction the mouse is facing when performing a motion.

The future work with the approach of LSTM involves testing the generalization ability on more data. Furthermore, assessing the representations obtained by the LSTM model is also valuable for neural science research.

An aspect that could be further investigated regards the evaluation of each model using a subset of the body parts. In both our experiments (TDA and LSTM) we considered all the body parts for the training of the models (for the LSTM case, we also made use of *likelihood* features). It may be interesting to evaluate if the models reach comparable prediction results considering a reduced number of input features (e.g. only considering paws, ears and nose, maybe)

7. Ethical considerations

When working with mice or other animals there are ethical aspects to consider [9]. In the case of mice their living standards are of importance, which covers things like cage size and breeding. Animal research in Sweden is strictly regulated under both Swedish and EU legislation. In order to carry out animal experiments a licence has to be applied for which undertakes the ethical review of the experiment. All our experiments were carried out under the supervision of the Fisone Laboratory at KI. We provided the guidelines of the experiments, but the handling of the mice was done by Fisone Lab researchers.

8. Opposing groups

The opponent groups for our project are groups 8 and 17. Group 17 has the project of imaging colorization. The main challenge they faced was the evaluation metric of the colored images, since same object in the world typically has multiple realistic colors. They decided to tackle this by assessing the robustness of models. Group 8 has the project of medical images classification for detection of skin lesions. They firstly wasted some time on finding a good GPU, later on had difficulty of achieving a desired accuracy with multiple instance learning approach they experimented. In our case, we had difficulty of poor quality video at the beginning, which is eliminated by recording new videos on KI site with desired conditions by ourselves. We also had the difficulty in generalizing LSTM to unseen data. The algorithm started to generalize better after we decided to normalize the mouse to the center of each sequence rather than center of each frame.

References

[1] Jens Agerberg, Ryan Ramanujam, Martina Scolamiero, and Wojciech Chachólski. Supervised learning using homology stable rank kernels. *Frontiers in Applied Mathematics and Statistics*, 7:39, 2021.

[2] Cai Song Kent C. Berridge Ann M. Graybiel Allan V. Kalueff, Adam Michael Stewart and John C. Fentress. Neurobiology of rodent self-grooming and its value for translational neuroscience. *Nature Reviews Neuroscience*, 2016.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *ArXiv*, 1409, 09 2014.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database.

[5] Alireza Dirafzoon, Namita Lokare, and Edgar Lobaton. Action classification from motion capture data using topological data analysis. In *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1260–1264, 2016.

[6] Batty E, Whiteway M, Saxena S, Biderman D, Abe T Murthy, Musall S, Gillis W, Markowitz J, Churchland A, Cunningham JP, and Datta SR. Behavenet: nonlinear embedding and bayesian neural decoding of behavioral videos. 2019.

[7] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[8] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model, 2016.

[9] Karolinska Institutet. Animal research at karolinska institutet, `https://education.ki.se/ethics`, 2021. Last accessed 22 December 2021.

[10] Koji Kobayashi, Seiji Matsushita, Naoyuki Shimizu, Sakura Masuko, Masahito Yamamoto, and Takahisa Murata. Auto-

mated detection of mouse scratching behaviour using convolutional recurrent neural network. *Nature research*, 2021.

[11] Gregory Kramida, Yiannis Aloimonos, Nikolas Francis, Patrick Kanold, Cornelia Fermüller, and Chethan M. Parameshwara. Automated mouse behavior recognition using vgg features and lstm networks. 12 2016.

[12] Mathis Laboratory. Official implementation of deeplabcut - github. Last accessed 22 November 2021.

[13] Alexander Mathis, Pranav Mamidanna, Kevin M. Cury, Taiga Abe, Venkatesh N. Murthy, M. Mathis, and Matthias Bethge. Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience*, 21:1281–1289, 2018.

[14] Joel Sjöbom, Martin Tamtè, Pär Halje, Ivani Brys, and Per Petersson. Cortical and striatal circuits together encode transitions in natural behavior. *Science Advances*, 6(41):eabc1173, 2020.

[15] Elsbeth A. van Dam, Lucas P.J.J. Noldus, and Marcel A.J. van Gerven. Deep learning improves automated rodent behavior recognition within a specific experimental setup. *Journal of Neuroscience Methods*, 332:108536, 2020.

A. Implementation details

Here follows more specifications in order to easily reproduce our experiments

A.1. TDA

The main details are described in the section 4.3. Here is a summary of parameters for the best model.

Table 7: Hyperparameter setting for the TDA-based approach

Hyperparameter	Value
Sliding Window Size	100
Overlapping Size	50
Distance Metric	Correlation Metric
Linkage	Single Linkage
Homological Dimension	0
Nearest Neighbour Number	9

A.2. LSTM

The main details of the network are reported in the section 4.4. In particular, the architecture is clearly described in figure 11. Nevertheless, some secondary details were omitted. For the sake of completeness, in table 8 is reported the list of all the the hyperparameters of our network.

Table 8: Hyperparameter setting for the LSTM-based approach

Hyperparameter	Value
Hidden size	150
Batch size	64
Dropout	0.3
Sequence length	200
Stride	0.75

The stride is reported in relative terms and it represents a parameter used for the construction of the input sequences: for example two consecutive sequences share a portion of 25% of information, since the stride is equal to 75%. We did not tune all the aforementioned hyperparameters for lack of time. It may be interesting to fine tune them, for example exploiting a random search approach.

Another important aspect that has not been mentioned in the LSTM section regards the dense network used for the prediction of the behavior classes (the one that takes in input the hidden state produced by each LSTM cell of the decoder). The configuration of such network is quite simple, indeed it contains only a hidden layer with 50 neurons.

Another detail that could be further investigated regards the omission of the likelihood features for the training of the model. Furthermore, it may be interesting to evaluate the effectiveness of different combination of markers: in our architecture we made use of all the 8 markers. Nevertheless, it would be interesting to see if the same results (or even better results) may be reached with few of these markers (for example only ears, nose and paws).