

DEPARTMENT OF BIOTECHNOLOGY  
INDIAN INSTITUTE OF TECHNOLOGY MADRAS  
CHENNAI – 600036

---

## **Unsupervised Behavior Discovery and Genetics of Mice**

*A Thesis*

*Submitted by*

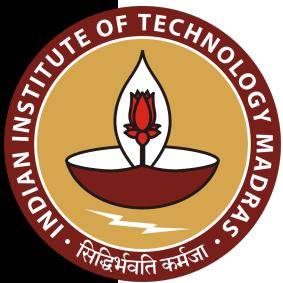
**HARISH MANOHARAN**

*For the award of the degree*

*Of*

**MASTER OF TECHNOLOGY**

June 2024



DEPARTMENT OF BIOTECHNOLOGY  
INDIAN INSTITUTE OF TECHNOLOGY MADRAS  
CHENNAI – 600036

---

## Unsupervised Behavior Discovery and Genetics of Mice

*A Thesis*

*Submitted by*

**HARISH MANOHARAN**

*For the award of the degree*

*Of*

**MASTER OF TECHNOLOGY**

June 2024

# THESIS CERTIFICATE

This is to undertake that the Thesis titled **UNSUPERVISED BEHAVIOR DISCOVERY AND GENETICS OF MICE**, submitted by me to the Indian Institute of Technology Madras, for the award of **Master of Technology**, is a bona fide record of the research work done by me under the supervision of **Dr. Balaraman Ravindran**. The contents of this Thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

**Chennai 600036**

**Date: June 2024**

**Harish Manoharan**

**Dr. Balaraman Ravindran**  
Research Guide  
Professor  
Department of Computer Science  
IIT Madras

**Dr. Vivek Kumar**  
Research co-advisor  
Jackson Laboratories

## **ABSTRACT**

Numerous methods, both supervised and unsupervised, have been suggested to understand behaviors from video data. Supervised methods help users pinpoint specific behaviors but require a lot of manual work to label frames precisely. Unsupervised approaches, while powerful and not needing upfront behavior labeling, produce numerous short behavior patterns, making it tough to figure out which ones are biologically relevant. This is a continuation of previous work on the comparison of four unsupervised methods for temporal behavior segmentation in animal videos. We attempt to further validate one of the unsupervised methods, VAME, on a more diverse set of behaviors and analyze the results.

# CONTENTS

	Page
<b>ABSTRACT</b>	<b>i</b>
<b>LIST OF FIGURES</b>	<b>iii</b>
<b>CHAPTER 1      OVERVIEW</b>	<b>1</b>
<b>CHAPTER 2      RELATED WORKS</b>	<b>3</b>
<b>CHAPTER 3      DATASET DESCRIPTION</b>	<b>5</b>
3.1      Pose Estimation . . . . .	5
<b>CHAPTER 4      METHODS</b>	<b>7</b>
<b>CHAPTER 5      COMPARISON OF METHODS</b>	<b>9</b>
<b>CHAPTER 6      VAME WORKFLOW</b>	<b>12</b>
<b>CHAPTER 7      ANALYSIS</b>	<b>14</b>
7.1      Preprocessing . . . . .	15
7.2      Model Training . . . . .	15
7.3      Parameter Tuning for number of clusters . . . . .	15
7.4      Bout Statistics . . . . .	17
7.5      Cluster overlap between keypoints . . . . .	18
7.6      Validating the results . . . . .	20
7.7      Rare motif usage . . . . .	22
7.8      Genetic Analysis . . . . .	24
<b>CHAPTER 8      CONCLUSION</b>	<b>25</b>
8.1      Further work . . . . .	25
<b>APPENDIX A      EXPLORING POTENTIAL APPLICATIONS OF                         LARGE LANGUAGE MODELS</b>	<b>26</b>
A.1      Literature Survey . . . . .	26
<b>REFERENCES</b>	<b>28</b>

# LIST OF FIGURES

Figure	Caption	Page
3.1	Pose estimation . . . . .	6
6.1	. . . . .	12
6.2	VAME - Complete workflow . . . . .	13
7.1	(A) An example top-down view frame from a video feed for 12 keypoints. Marker positions are estimated for the frame from the HRNet32 architecture. (B) Top-down view frame for 10 keypoints. (C) Top-down view frame from 8 keypoints. . . . .	14
7.2	WCSS vs Number of Cluster . . . . .	16
7.3	Angular change . . . . .	16
7.4	Number of bouts . . . . .	17
7.5	Average length of bouts . . . . .	17
7.6	Total length of bout . . . . .	18
7.7	Size of each cluster across keypoints . . . . .	18
7.8	IoU percent of clusters across keypoints . . . . .	19
7.9	Clusters with high pairwise IoU are grouped together and marked with corresponding behaviour . . . . .	19
7.10	. . . . .	21
7.11	. . . . .	22
7.12	Percent Zero value . . . . .	23
7.13	Correlation of Body weight filtered by sex . . . . .	24

# **CHAPTER 1**

## **OVERVIEW**

Animals exhibit a diverse array of behaviors, which they combine and sequence to execute intricate tasks. Rather than solely being driven by evolutionary pressures, the boundaries of their anatomy play a crucial role in shaping the spectrum of behaviors animals can display. Despite this, most animal repertoires consist of a limited set of repetitive and interconnected actions that are distinctive to their species. Through careful human observation, researchers have identified and categorized these behavior sets, understanding their structure and how they form sequences. These efforts have resulted in the development of simplified descriptions of animal behavior, linking them to underlying motor mechanisms, thus advancing our comprehension of neural circuitry and neural-motor mapping and aiding in the identification and assessment of psychiatric disorders that may manifest as aberrations in behavior.

Precise measurement of behavior is essential when constructing an animal's behavioral repertoire. While older studies often relied on coarse metrics such as average speed or total distance traveled, newer approaches recognize the need for more nuanced measurements that can capture behavior at finer scales. Constructing a behavioral repertoire involves making careful and justified decisions regarding postural representation, experimental conditions, data extraction, and criteria for segmenting unstructured demonstrations into constituent behaviors.

Unsupervised methods offer a promising avenue for accurately measuring behavior, as they don't depend on human annotations and can capture rich dynamical representations of behavior on a sub-second scale. Recent advancements in unsupervised behavioral quantification have been introduced to address this need. Despite the consensus among

computational ethologists that observable behavior can be encoded in a lower-dimensional subspace or manifold, existing methods have limitations in analyzing extensive data.

# **CHAPTER 2**

## **RELATED WORKS**

Behavioral analysis necessitates a sophisticated, high-dimensional, and often hierarchical representation (Gomez-Marin et al., 2014). Recent advancements in behavioral neuroscience exploit technological progress to extract more detailed representations, predominantly from visual data (Calhoun and Murthy, 2017; Egnor and Branson, 2016). The prevailing approach involves quantifying behavior through predefined categories with the involvement of human operators.

An alternative perspective on the behavior discovery problem posits that the requisite structure is inherent in the data, and efforts should focus on surfacing and extracting it. This unsupervised learning philosophy, propelling recent advancements in the field, inspires this study. Seminal research mapping the behavioral space for fruit flies (Berman et al., 2014) attests to the effectiveness of this approach. The proposed Motion Mapper solution employs video data of *Drosophila*, extracting postural representations using Principal Component Analysis (PCA) and Morlet wavelet transform. This representation is embedded in two dimensions using distributed stochastic neighbor embedding (t-SNE) and segmented to generate distinct behavioral regions, revealing a hierarchical structure (Berman et al., 2016).

While unsupervised methods have shown success with animals like roundworms, *Drosophila*, and zebrafish, fewer studies have explored mouse behavior mapping. The intricate morphology of mice, with their flexible movements, presents challenges, but recent investigations suggest that unsupervised methods remain applicable. For instance, MotionMapper has been applied to study mouse autism models, showcasing its adaptability. Additionally, comprehensive work on sub-second behavior in mice,

employing Autoregressive Hidden Markov Models (AR-HMM), reveals inherent structure in mouse pose data (Wiltschko et al., 2015).

Recent developments in unsupervised behavioral segmentation (Hsu and Yttri, 2020) operate on the sub-second timescale, aiming for high throughput with a focus on geometric and displacement features. The VAME model (Variational Animal Motion Embedding) introduces an unsupervised deep learning approach for behavior segmentation, outperforming other methods in a grooming dataset validation (Ke et al., 2021).

Previous efforts at validation of these models have been characterized by several limitations. Firstly, these models have not been systematically compared using a consistent dataset. Secondly, the datasets employed in these validation attempts have been relatively small in scale. In this study, we aim to address these shortcomings by conducting a comprehensive validation of these models using a large, well-labeled dataset. Our hypothesis posits that one of these methods shows higher performance, and our aim is to compare and find which of these methods excel when evaluated on a large dataset.

Our recent comparative study assessed unsupervised methods (VAME, MotionMapper, AR-HMM, and B-SOIID) using an annotated grooming dataset. Results show that VAME outperforms the other three methods, but the experiment was done on a small video subset focusing on grooming behavior. The current study aims to evaluate VAME’s performance on a large dataset encompassing a diverse range of behaviors.

## CHAPTER 3

### DATASET DESCRIPTION

A large strain survey dataset was used to comprehensively examine variations across mouse strains and capture a diverse array of behaviors. This extensive dataset JABS600 comprises nearly 600 mice, encompassing 60 different strains that include classical inbred laboratory strains, wild-derived inbred strains, and F1 hybrid strains. Each animal in the dataset is represented by one hour of video footage.

The data for each mouse is presented in the form of a top-down video feed, with frames measuring  $480 \times 480$  pixels. An illustrative sample image is provided in Fig. 4.1 for reference. The pose data is derived from a sophisticated modified HRNet architecture (Sheppard et al., 2020), which estimates the positions of 12 keypoints. These keypoints include the nose, left ear, right ear, base of the neck, center of the spine, left front paw, right front paw, left rear paw, right rear paw, base of the tail, middle tail, and the tip of the tail. Additionally, the confidence levels for each keypoint estimate are provided. The coordinates of these keypoints collectively constitute the pose information that is fed into the pipeline developed in this study, enabling a detailed analysis of the diverse behaviors exhibited across the extensive range of mouse strains.

The keypoint coordinates are forward-filtered to remove low confidence points. A threshold of 30% is used for differentiating low confidence points. Any animals with more than 10% of data filtered from any marker are left out of the study.

#### 3.1 POSE ESTIMATION

The present study employs pose estimation, which refers to the process of determining the 2D coordinates of predefined keypoints in images or videos, as the foundation

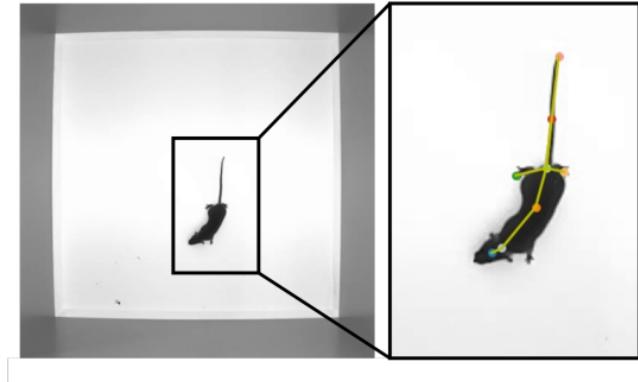


Figure 3.1: Pose estimation

of the method for unsupervised behavior analysis. The chosen keypoints are salient visually, such as the nose or ears. Some are particularly important to understanding pose, such as the limb joints or the paws. In this study, 12 keypoints were selected to context mouse pose, namely nose, right ear, left ear, neck base, left front paw, right front paw, spine center, left-hind paw, right-hind paw, tail base, mid of tail, and tip-tail. To implement pose estimation, we utilized the HRNet neural network architecture(Sun et al., 2019) that was modified to suit our experimental setup. To match the key-point output resolution with the video input resolution catering to our requirements, two  $5 \times 5$  transpose convolutions were added to the head of the network. This involved generating 12 480x480 heatmaps (one for each keypoint) for every 480x480 frame of video. So, the maximum value in each of the heatmaps represented the highest predicted confidence location for each respective keypoint, allowing us to obtain 12 (x, y) coordinates by taking the argmax of each heatmap. The network was trained using the ADAM optimization which is a variant of the stochastic gradient descent. The labels were generated from a diverse range of mouse strains to ensure robustness of the network to wide variety of mice appearances.

# CHAPTER 4

## METHODS

**MotionMapper:** Motion-Mapper employs a Morlet wavelet transform on the time series of keypoints. This process generates a spectrogram for each postural mode, capturing the temporal evolution of key features. After normalization, the data is projected onto a two-dimensional plane using t-distributed Stochastic Neighbor Embedding (t-SNE). Subsequently, a watershed transform is applied to a Gaussian-smoothed density over these points, isolating individual peaks from one another. The regions with high densities are assumed to contain stereotypical behaviors. The reliance on wavelet transformations could limit its ability to capture the full behavior repertoire, especially in animals with prominent low-frequency movements.

**AR-HMM** (Autoregressive Hidden Markov Model) (Wu et al., 2020): AR-HMM is a probabilistic graphical model that integrates autoregressive (AR) and hidden Markov models (HMM) to represent time series data. Specifically, in the context of MoSeq, AR-HMM is employed to identify brief 3D behavioral motifs. By capturing temporal dependencies and incorporating unobserved latent variables, the model can uncover complex patterns in mouse behavior.

**B-SOiD** (Behavioral Segmentation of Open-field in DeepLabCut) (Hsu & Yttri, 2019): B-SOiD constructs Kinematic features from the keypoints and groups data according to the strain. Construction of the spatial embedding through Uniform Manifold Approximation and Projection (UMAP) and consequent clustering with Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) are carried out on a per-animal level and also on a per-strain level. The complete mouse behavior is trained on a boosting classifier to categorize frames by behavior. This method projects framewise

into a UMAP representation mainly from a velocity feature signal, potentially limiting the temporal information utilized.

**VAME** (Variational Animal Motion Embedding)(Luxem et al., 2022): The estimated poses are egocentrically aligned, and trajectory samples are fed into a bidirectional recurrent neural network model (Medsker & Jain, 2001). The variational recurrent neural network autoencoder effectively captures the spatiotemporal features in a latent space. The fully trained model resembles a dynamical system from which motifs are inferred via K-Means clustering.

## CHAPTER 5

# COMPARISON OF METHODS

In order to validate all the models, we used a manually labeled dataset that was annotated by five trained human experts. At each frame in a video, the mouse has been considered in two states: grooming or not grooming. We have trained all the models on the above dataset in an unsupervised fashion and validated their overlap with the annotations on three variations of keypoints. First, we keep all the predefined set of 12 keypoints. In the second set, we remove the mid-tail and tip of the tail to focus on behaviors not biased by the movement of the tail. For the third set, we further exclude the front paws from the pose estimation as the top-down view of the video feed occludes them for a large number of frames, resulting in low confidence estimates.

### Scoring Metrics and Quantification

To investigate the overlap of each model with the annotated dataset, we quantified Purity, Normalized Mutual Information (NMI) and Homogeneity.

Purity is a measure of the proportion of frames within a cluster that belong to the most frequent class (grooming or non-grooming in our context). Purity is defined as

$$\text{Purity}(U, V) = \frac{1}{N} \sum_{u \in U} \max_{v \in V} |u \cup v| \quad (5.1)$$

where  $U$  is the set of annotated labels,  $V$  is the set of labels generated by the model and  $N$  is the number of frames in the behavioral video.

The Normalized Mutual Information measures the agreement between two sets of clusters considering the cluster sizes and the distribution. The score is given by

$$\text{NMI}(U, V) = \frac{\text{MI}(U, V)}{E(H(U), H(V))} \quad (5.2)$$

where  $\text{MI}(U, V)$  is the mutual information between set  $U$  and  $V$  defined as

$$\text{MI}(U, V) = \sum_{u \in U} \sum_{v \in V} \frac{|u \cap v|}{N} \log\left(\frac{N|u \cap v|}{|u||v|}\right) \quad (5.3)$$

Here the  $||$  operator denotes the number of frames that have the corresponding labels assigned.

Homogeneity measures the extent to which each cluster only contains data points from a single true class. Homogeneity is defined as

$$\text{Homogeneity} = 1 - \frac{H(U|V)}{H(U)} \quad (5.4)$$

where,

$$H(U|V) = - \sum_{u=1}^{|U|} \sum_{k=1}^{|K|} \frac{|u \cap v|}{|u \cap v|} \log\left(\frac{|u \cap v|}{|v|}\right) \quad (5.5)$$

All three metrics range from 0 to 1, with a higher score indicating better overlap. Upon analyzing the results obtained from all three measures, we observed that VAME yielded the highest score for each metric. B-SOID, AR-HMM, and Motion-Mapper followed VAME in that order. We also investigated the impact of using different sets of keypoints on the scores obtained for each method. Our results indicate that the number of keypoints does not necessarily correlate with higher scores. Moreover, we found that the exclusion of tail keypoints led to a drastic improvement in the validation scores of VAME.

<b>Models (K ≈ 15)</b>	<b># Keypoints</b>	<b>Abs. Purity</b>	<b>Abs. NMI</b>	<b>Homogeneity %</b>
VAME	12	71.08	6.81	16.52
	10	<b>77.0</b>	11.81	28.40
	8	76.65	<b>11.97</b>	<b>28.70</b>
B-SOiD	12	63.75	3.19	7.72
	10	64.15	4.30	10.65
	8	62.43	2.62	5.96
AR-HMM	12	63.15	2.76	4.80
	10	61.23	1.52	2.54
	8	61.79	2.30	3.58
Motion-Mapper	12	60.84	0.17	0.35
	10	60.84	0.29	0.64
	8	60.69	0.39	0.77

Table 5.1: Quantitative comparison of models based on annotated grooming behaviors

# CHAPTER 6

## VAME WORKFLOW

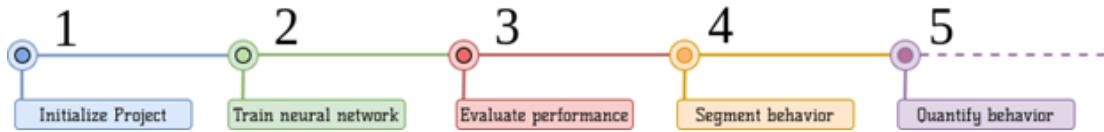


Figure 6.1

The pose data first undergoes egocentric alignment. The goal of egocentric alignment is to transform the recorded data from an allocentric (external reference frame) to an egocentric coordinate system (aligned with the animal's body). This transformation is crucial when studying the behavior of freely moving animals because it helps in analyzing the data from the animal's perspective, making it easier to interpret and quantify their movements.

Once the data is egocentrically aligned, trajectory samples are extracted to create the input for training the VAME model. Trajectory samples capture the kinematic information of the animal's movements over time. These samples are essential for training a model that can learn the underlying patterns and dynamics of the observed behavior. The trajectory samples obtained from the aligned and preprocessed data are used to train the VAME (Variational Animal Motion Embedding) model. The VAME model is a probabilistic deep-learning framework designed to discover underlying latent states in behavioral signals.

The VAME model utilizes a two-layer bidirectional Recurrent Neural Network (biRNN) for encoding and decoding. The encoder captures temporal dependencies and reduces the dimensionality of input sequences. Latent representations are obtained using the reparameterization trick, with the encoder hidden state concatenated to form a global

hidden state. The generative model uses a biRNN decoder to reconstruct input sequences and predict their future dynamics.

The training objective involves minimizing a loss function that combines Mean Squared Error (MSE) reconstruction loss with a regularization term for predicting future dynamics. The regularization term introduces an additional prediction decoder (biRNN) with its own set of parameters.

Either of HMM or k-means clustering can be applied to the latent space to detect behavioral motifs.

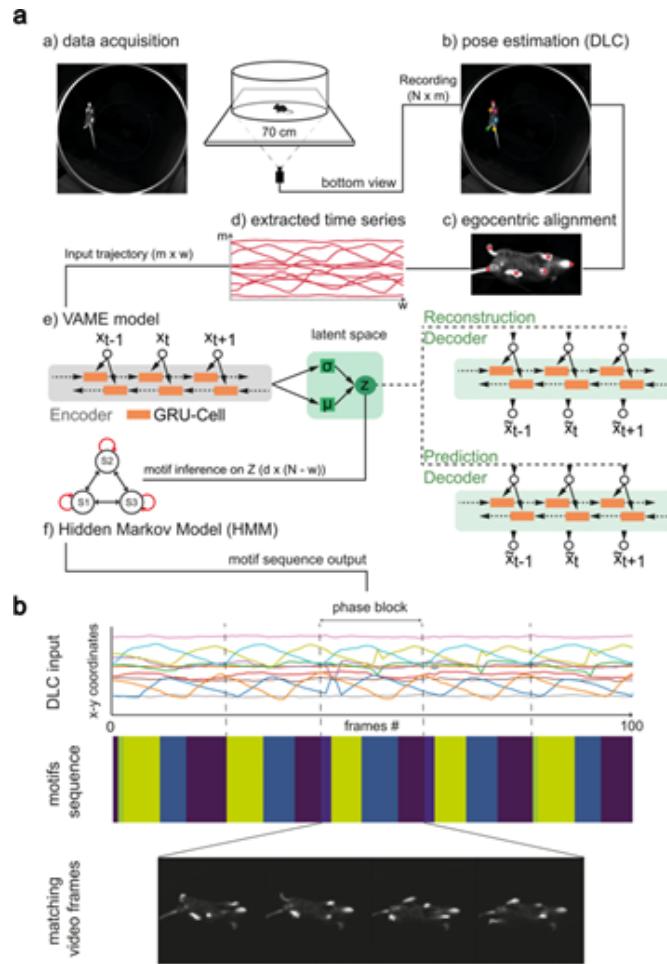


Figure 6.2: VAME - Complete workflow

# CHAPTER 7

## ANALYSIS

In the analysis of the JABS600 dataset using the VAME model, three sets of keypoints are considered

- 12 keypoints – we keep all the predefined set of 12 keypoints
- 10 keypoints – we remove the mid-tail and tip of the tail to focus on behaviors not be biased by the movement of the tail
- 8 keypoints – we further exclude the front paws from pose estimation due to frequent occlusion in the top-down view, leading to low confidence estimates

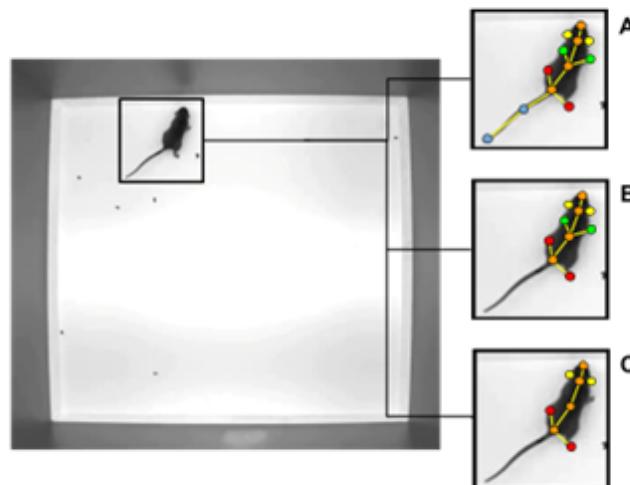


Figure 7.1: (A) An example top-down view frame from a video feed for 12 keypoints. Marker positions are estimated for the frame from the HRNet32 architecture. (B) Top-down view frame for 10 keypoints. (C) Top-down view frame from 8 keypoints.

## 7.1 PREPROCESSING

Since the JABS600 dataset is very large, containing 600 hours of video data, and could take weeks to train, we take subsamples of the dataset to be trained and inferred upon. Three subsamples of 10%, 20%, and 40% are created by extracting continuous snippets from each video. For instance, in the 20% subsample, four random snippets of 5000 frames each are selected from every video. The pose data is subsampled accordingly, forming the training dataset for VAME. Once the dataset is subsampled, the pose data undergoes egocentric alignment and is then fed into the model.

## 7.2 MODEL TRAINING

VAME models are trained for each combination of subsample percentage and set of keypoints (three keypoints and three subsamples in total). All nine models were trained in parallel and took about a week to complete. We chose to use the 20% subsample for all three keypoints, as it had the lowest reconstruction loss, effectively capturing the pose information. Next, we apply pose segmentation to all three keypoints to get the clusters and the motif videos. The motif videos are then combined cluster-wise to make it easier to label them.

## 7.3 PARAMETER TUNING FOR NUMBER OF CLUSTERS

In order to find the optimal number of clusters for VAME, we employ the elbow method. To perform this, we train each of the keypoint VAME models with the number of clusters ( $K$ ) varying from 5 to 100 with a step interval of 5. The elbow method infers that the optimal number of clusters corresponds to the juncture after which the gains are minimal with increasing cluster count. In this context, the gains here can be estimated with the change in within cluster sum of squares (WCSS), a metric used to gauge the dispersion of datapoints within each cluster. WCSS encapsulates the cumulative sum of squared Euclidean distances between data points and their respective cluster centroids. Note that the trajectory of the WCSS is intrinsically anticipated to decrease as the number of

clusters increases.

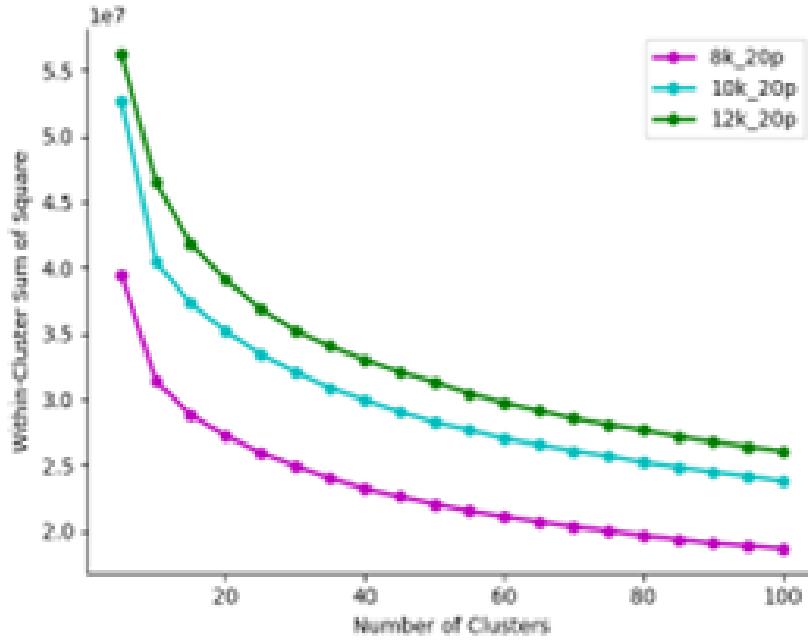


Figure 7.2: WCSS vs Number of Cluster

We notice that a particular elbow point could not be identified with WCSS vs No. clusters plot. So, for a deeper analysis, we plot the angle contained by the point in the plot between the lines formed by the points before and after it. A smaller angle implies the plot is closer to linear, and linear gain is close to the minimum gain we expect. For clusters 8k, we see a steady decrease in the angle until 35 clusters. For 10k and 12k, we chose 40 as after that point, the angle oscillates a lot up and down the value there. Hence, we feel 40 is a more optimum number for our K-parameter.

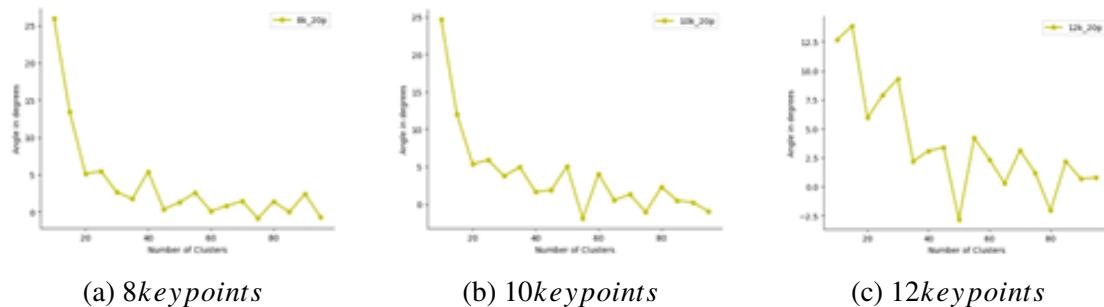


Figure 7.3: Angular change

## 7.4 BOUT STATISTICS

Using the frame-wise cluster label information for each video, we are able to extract statistics on average bout length, number of bouts, and total duration for each video snippet extracted.

For example, for cluster label 0,

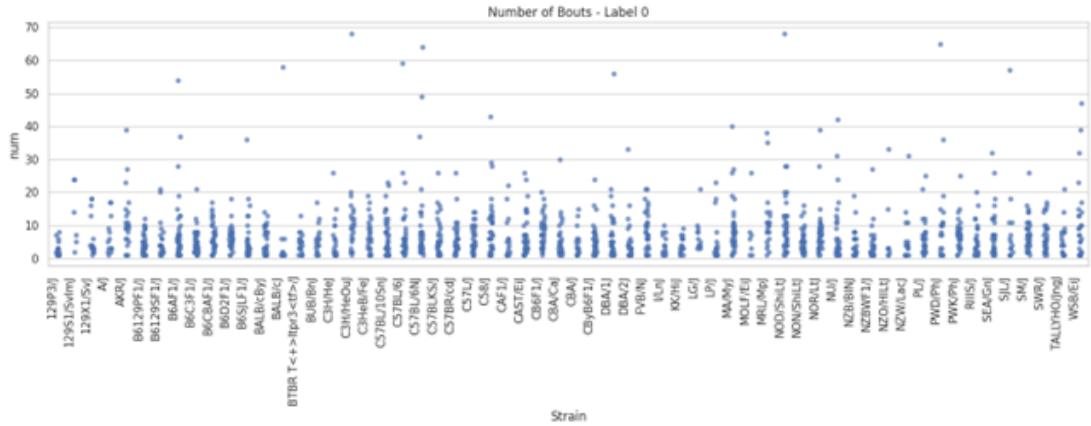


Figure 7.4: Number of bouts

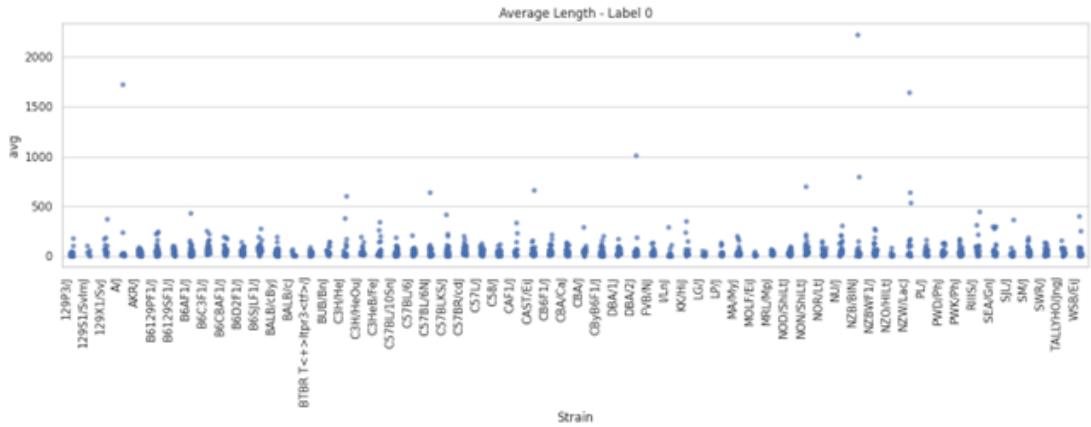


Figure 7.5: Average length of bouts

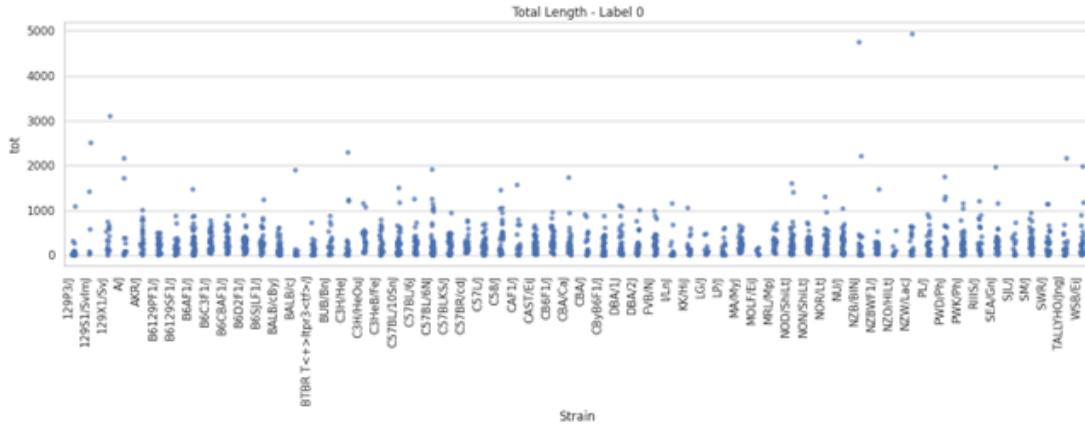


Figure 7.6: Total length of bout

## 7.5 CLUSTER OVERLAP BETWEEN KEYPOINTS

We explore the similarity between cluster labels pairwise between different keypoints to both aid in labelling them as well as a sanity check on the clusters. We find the IoU (intersection over union) values between pairwise labels of the three keypoints and form a network. IoU values serve as a metric for comparing cluster labels and provide insights into the consistency of clustering across different keypoints.

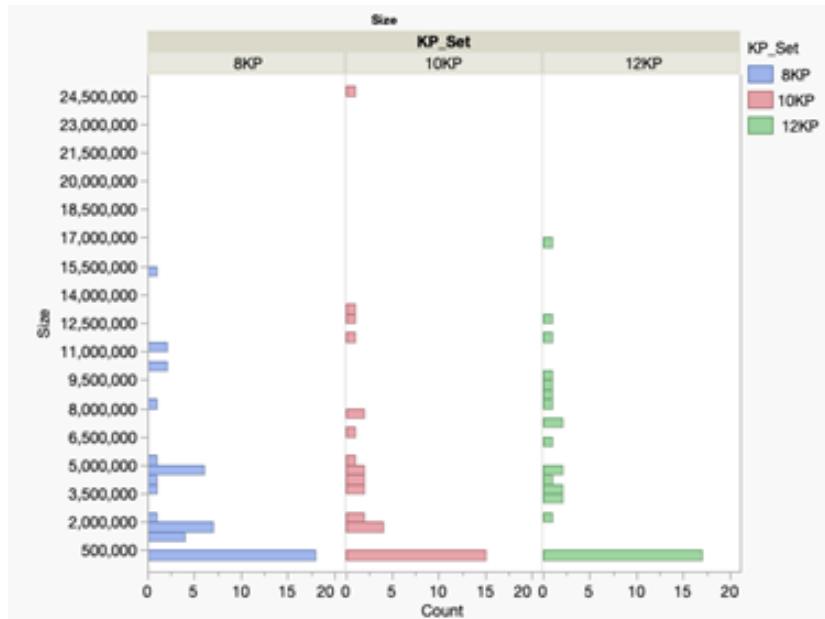


Figure 7.7: Size of each cluster across keypoints

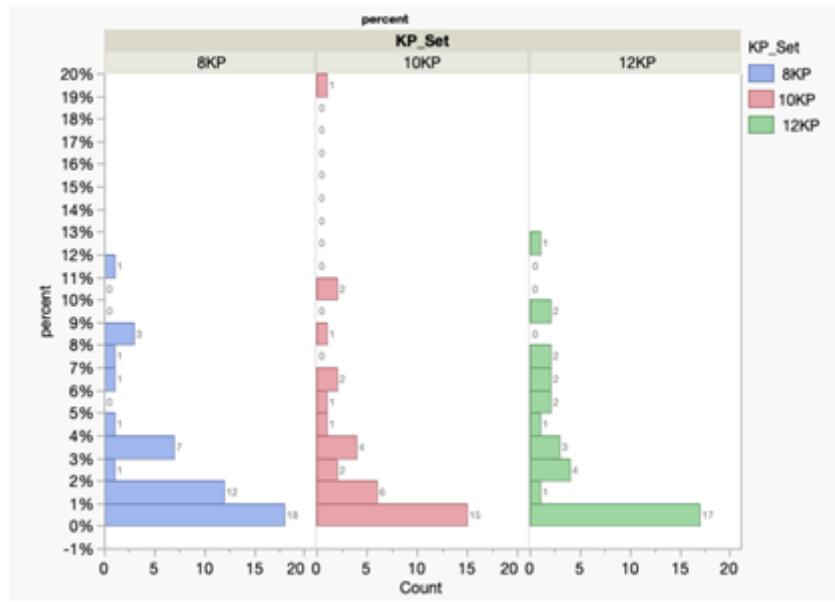


Figure 7.8: IoU percent of clusters across keypoints

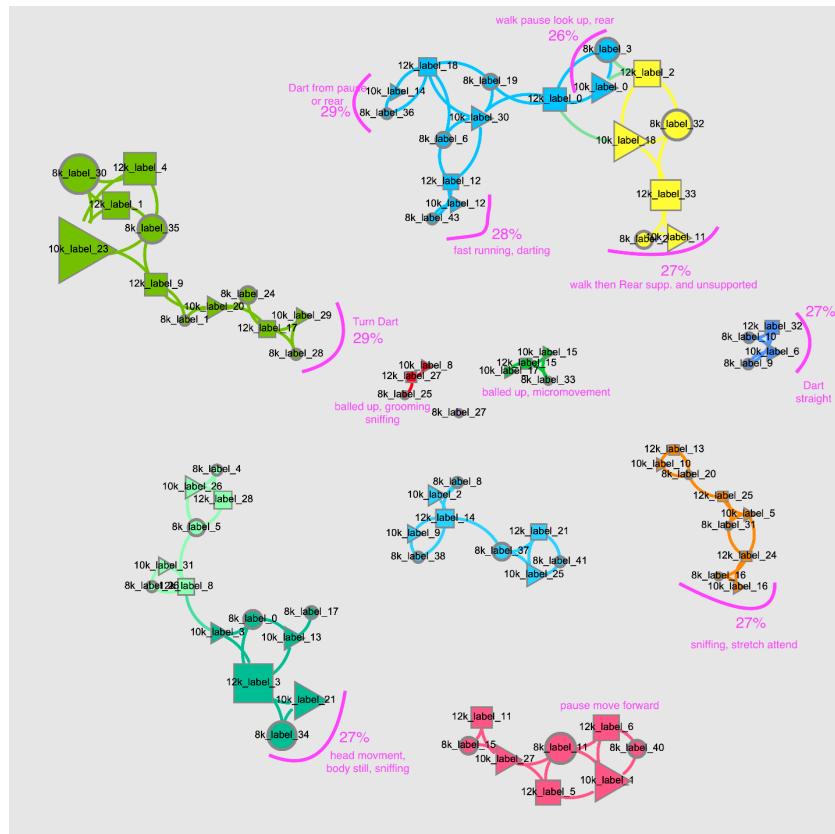


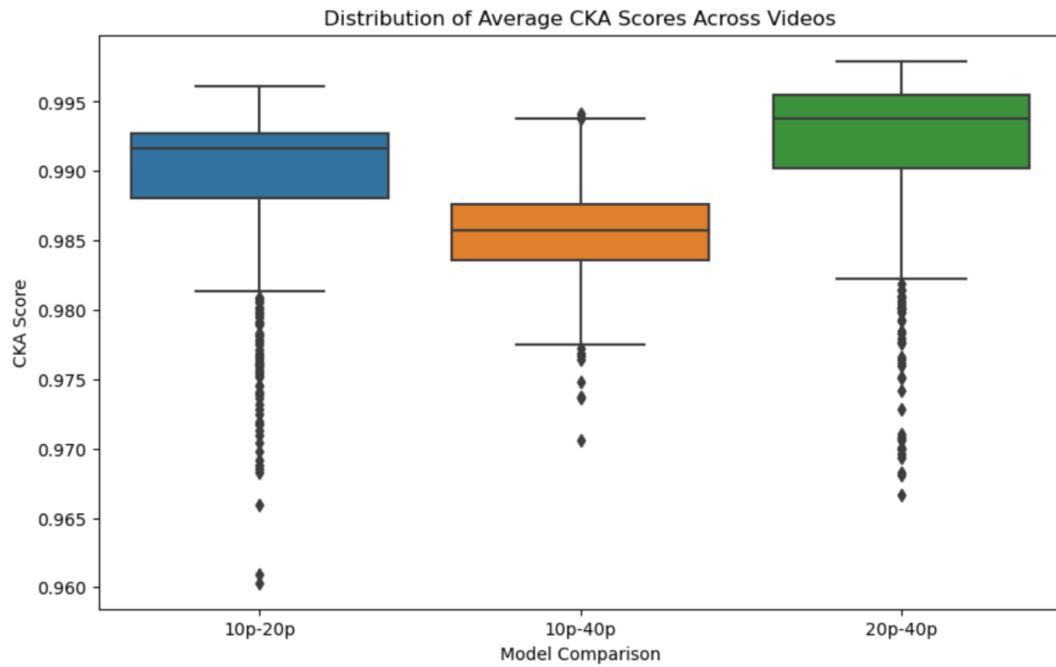
Figure 7.9: Clusters with high pairwise IoU are grouped together and marked with corresponding behaviour

## 7.6 VALIDATING THE RESULTS

We would like to see how “close” the latent vectors are for each comparison (for ex: model-1 vs model-2) in the embedding space of RNN-VAE used in VAME.

Specifically, for our problem, we would like to see if the embeddings produced from the models trained on 10, 20 and 40% of the random subset of the full training data, produce similar embeddings or not.

Linear CKA analysis (X, Y are two representation matrices). Well defined for a list of frames (videos) but is memory hungry (scales nonlinearly with number of examples)



All the models have fairly good similarity scores, which means all the models are usable and have learned a consistent representation of poses. However, within the latent representation space, model trained with 10% and 40% data are farthest apart from each other. Similarly, models trained with 20% and 40% data are closest to each other.

To further validate, we trained the model on 1% of the data to show that learning has taken place in all these three models. It would be expected that the 1% model would be

significantly different than the three models. Here, we experiment with another metric called the Wasserstein metric. The C57 model here refers to the model trained on the C57 strain alone and inference on entire JABS600.

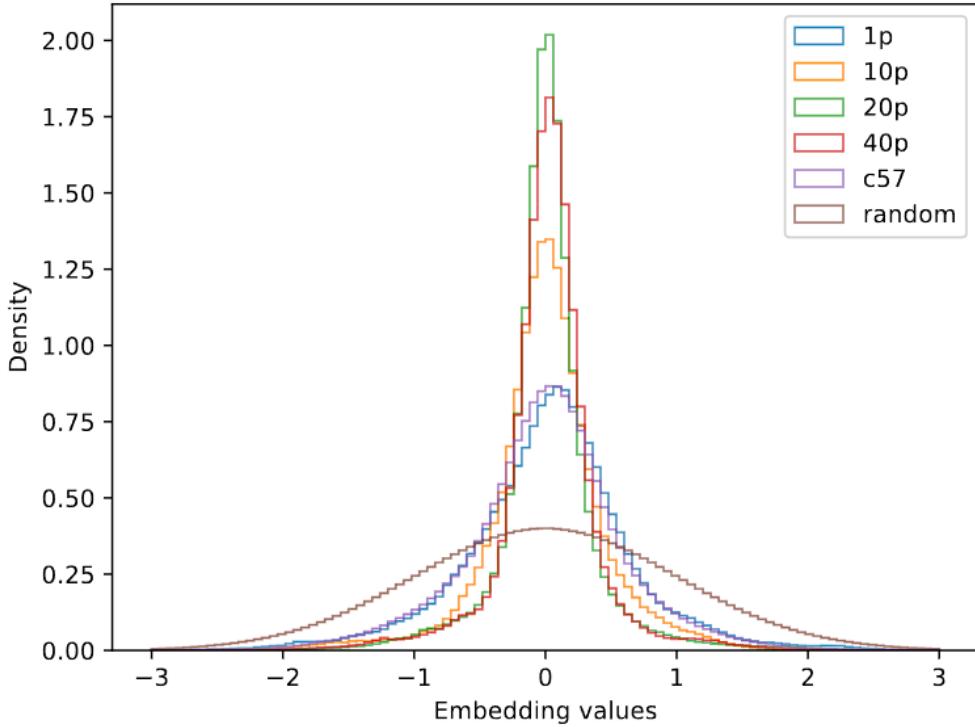


Figure 7.10

Fig 7.12: The latent distribution of models trained with 20% and 40% are closest to each other followed by 10%, 1%, c57 and completely random gaussian latents (untrained VAE). This reinforces our original hypothesis that latents are getting more consistent with each other and there is no significant improvement in latent representation from 20% to 40%.

Interestingly, the model trained on 1% of all the strains vs the model trained just on c57 mice do have very similar latents. Perhaps the model is learning broad features of key-points which may not vary across strains.

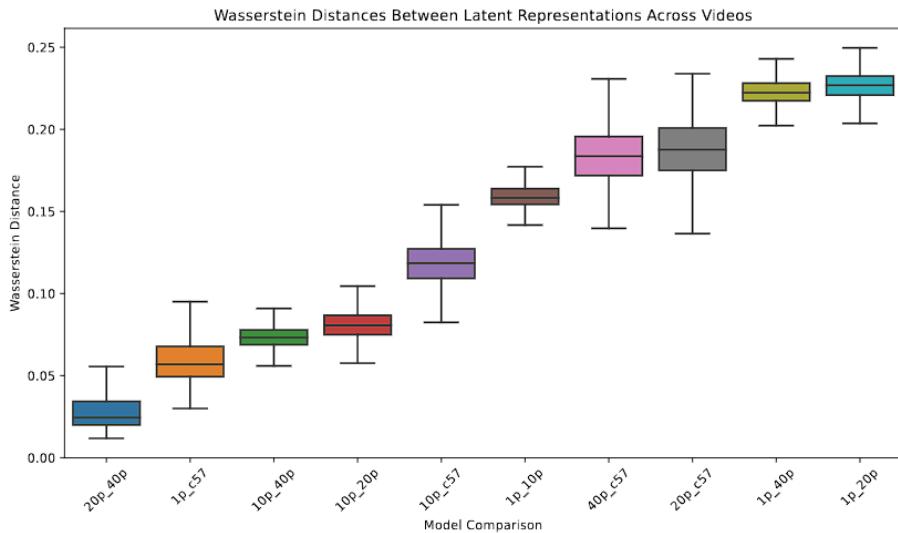


Figure 7.11

As expected, the latents between models trained with 20% & 40% of the data are very similar to each other, followed by 10 & 20% and 10 & 40%.

The most dissimilar ones are (1% and 40%) and (1% and 20%).

## 7.7 RARE MOTIF USAGE

We counted the Percent Zero of each motif in each animal. If a motif is only represented in a couple of animals, it will have a high 0% usage in most animals.

Approximately 50% of the motifs are rare, only seen in a few animals.

We used a cutoff of 0.75 (75%) for further analysis. If a motif is seen not seen (0%) in 75% or more of the animals, it would not be analyzed.

The percent of VAME motifs that are rare  
The higher the "PercentZero", the rarer the motif

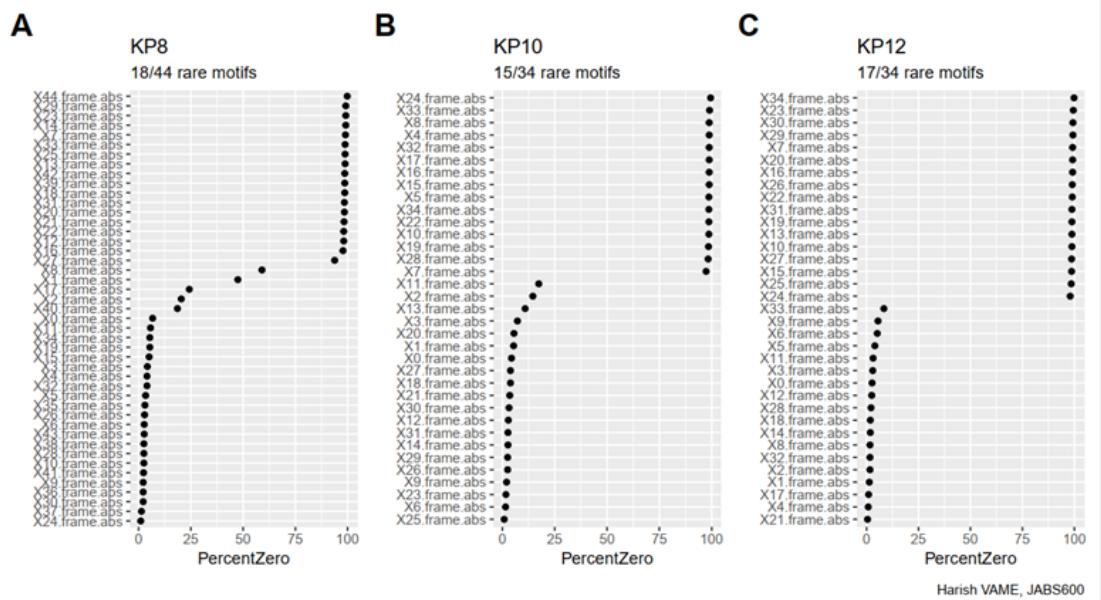


Figure 7.12: Percent Zero value

## 7.8 GENETIC ANALYSIS

We performed genetic analysis of absolute number of frames for each motif with respect to sex and body weight.

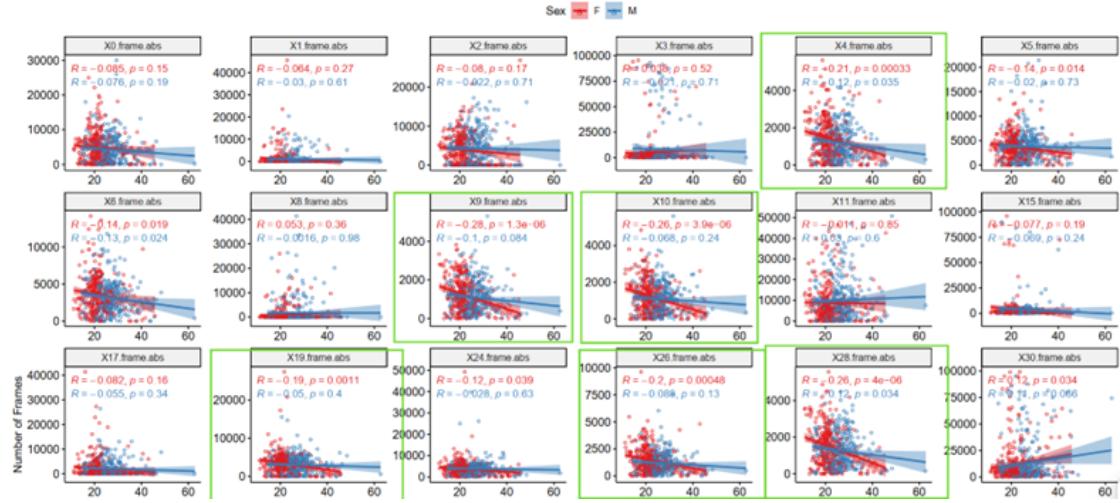


Figure 7.13: Correlation of Body weight filtered by sex

We notice that about a third of the motifs in females correlates with body weight

# **CHAPTER 8**

## **CONCLUSION**

We were successfully able to apply VAME on a subsample of the large JABS600 dataset while covering a diverse set of behaviors. The embeddings from VAME could be used for the analysis of new mice videos. A provision was implemented to allow a new video to be classified into the corresponding labels using the trained VAME model. The cluster labels from the model inference were further analyzed and marked with corresponding behaviors. Genetic analysis was performed based on sex and body weight.

### **8.1 FURTHER WORK**

- Further genetic analysis can be performed on different attributes and traits of mice.
- How behavior changes with different strains could be analyzed further
- Explore different clustering techniques (Ex: TAEC: Unsupervised Action Segmentation with Temporal-Aware Embedding and Clustering)

## APPENDIX A

# EXPLORING POTENTIAL APPLICATIONS OF LARGE LANGUAGE MODELS

Here, we explore how LLMs can be used in understanding videos, using video-text pairs, and then extending it to understanding behaviors and answering questions on videos etc.

### A.1 LITERATURE SURVEY

Hu Xu et. al (2021) proposes a contrastive learning framework that leverages temporally overlapping positive video-text pairs and hard negatives for pre-training a Transformer model. Pre-trains a Transformer model with a contrastive objective using pairs of video-text clips. The resulting pre-trained model can be directly applied to or fine-tuned on various video-text tasks. To address the issue of semantic misalignment between video clips and text transcriptions, the authors pre-train with temporally overlapped pairs of video and text clips. Proposes a retrieval augmented pre-training approach to learn fine-grained video-text similarity from a contrastive loss. By retrieving a cluster of videos that are similar to each other for each training batch, the authors obtain more challenging negative pairs, leading to better learning outcomes. The video features are extracted by a pre-trained frozen video encoder below which MLP layers are trained to obtain video tokens, with same embedding dimensions as text embeddings. For text encoding, vectors for text tokens are obtained via embedding lookup from BERT. Both video and text tokens are passed through separate trainable Transformers to obtain hidden states. A contrastive loss is used to learn the correspondence between video and text. VideoCLIP model allows for zero-shot transfer to various video-text understanding tasks.

Yue Zhao et. al (2022) aims to use LLMs to create automatic video narrators by repurposing pre-trained LLMs to be conditioned on visual input and fine-tuning them.

Video clips are encoded into video tokens capturing the spatial and temporal information. The encoded visual features are then used to condition the LLMs during training. They are incorporated into the LLM architecture through cross-attention modules added before each Transformer decoder layer. These cross-attention modules enable the LLMs to attend to the visual information while generating text, facilitating the generation of visually informed descriptions. The attention mechanism computes attention weights that determine the importance or relevance of each visual feature for generating the current word in the textual description. A rephraser, a text-to-text LLM, is then used to paraphrase the narrations, further augmenting the annotations.

Generates human motion from action descriptions output motion =  $f(\text{text}, \text{task}, \text{input motion})$ . It maps human poses to discrete motion codes using a pre-trained motion VQ-VAE. Generates instructions by combining codes from language prompts and motion prompts. The fine-tuning process incorporates pose sequence information and leverages strong motion priors from the original LLM. Uses pre-trained motion VQ-VAE (Vector Quantized Variational Autoencoder) to learn discrete representations for generative models. It takes a human pose as input and generates a motion code. Here, instructions are designed to combine task prompts and control conditions. Task prompts, text control conditions, and pose control conditions are organized into a unified question template for the LLM. Pose control conditions are generated using the same motion VQ-VAE. It uses the Low-Rank Adaptation (LoRA) technique to fine-tune the LLM using motion instructions. Instruction tuning enables the LLM to handle different generation tasks.

The instruction I and the answer  $P^A$  (generated motion codes) are passed to the LLM, and the model is trained to maximize the similarity between predicted motion codes and the ground-truth motion codes. The LLM takes the instructions as input and generates a sequence of motion codes as the answer. The output motion codes are decoded into sequence of human poses

## REFERENCES

- [1] G. J. Berman, D. M. Choi, W. Bialek, and J. W. Shaevitz. Mapping the stereotyped behaviour of freely moving fruit flies. *Journal of The Royal Society Interface*, 11(99):20140672, 2014.
- [2] S. R. Egnor and K. Branson. Computational analysis of behavior. *Annual review of neuroscience*, 39:217–236, 2016.
- [3] B. Geuther, K. Kikuchi, T. Pohlkamp, B. Rust, and M. Wöhr. Automated video tracking of mouse social behavior with identification by e-nose. *PLoS One*, 14(3):e0212796, 2019.
- [4] W. Hong, A. Kennedy, X. P. Burgos-Artizzu, M. Zelikowsky, S. G. Navonne, P. Perona, and D. J. Anderson. Automated measurement of mouse social behaviors using depth sensing, video tracking, and machine learning. *Proceedings of the National Academy of Sciences*, 112(38):E5351–E5360, 2015.
- [5] A. Hsu and E. A. Yttri. Unsupervised behavioral segmentation using spatial embeddings and geometric displacement features. *eLife*, 9:e59317, 2020.
- [6] A. I. Hsu and E. A. Yttri. B-soid: an open source unsupervised algorithm for discovery of spontaneous behaviors. *BioRxiv*, 770271, 2019.
- [7] X. Ke, G. De Rienzo, S. Martin, T. Dening, P. Domenici, T. Clark, and T. Ellis. Vame: A variational autoencoder for unsupervised animal motion segmentation and analysis. *Frontiers in Neural Circuits*, 15, 2021.
- [8] K. Luxem, P. Mocellin, F. Fuhrmann, J. Kürsch, S. R. Miller, J. J. Palop, S. Remy, and P. Bauer. Identifying behavioral structure from deep variational embeddings of animal motion. *Communications Biology*, 5(1):1267, 2022.

- [9] H. G. Marques, N. von Ellenrieder, S. Grünewälder, and R. M. da Costa. Unsupervised classification of the swimming behaviour of the zebrafish using time derivatives of motion features. *Journal of The Royal Society Interface*, 15(146):20180297, 2018.
- [10] A. Mathis, P. Mamidanna, K. M. Cury, T. Abe, V. N. Murthy, and M. W. Mathis. Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience*, 21(9):1281–1289, 2018.
- [11] A. A. Robie, K. M. Seagraves, S. Egnor, and K. Branson. Motionmapper: a web-based tool for analysis of motion in animals. *Nature Methods*, 14(4):351–352, 2017.
- [12] K. Sheppard, J. Gardin, G. Sabnis, A. Peer, M. Darrell, S. Deats, B. Geuther, C. M. Lutz, and V. Kumar (2020) Gait-level analysis of mouse open field behavior using deep learning-based pose estimation. *bioRxiv*.
- [13] Stirling, D. I., Bakshi, V. R., Mechling, A. E., Schweitzer, J. B., Kelley, A. E., Buchovecky, C. M., Graw, S. L., Abramowitz, J., Zgombic-Knight, M., & Trumbower, R. D. (2002). Heritability of open-field behavior in mice. *Behavior Genetics*, 32(6), 443–452.
- [14] Wiltschko, A. B., Johnson, M. J., Iurilli, G., Peterson, R. E., Katon, J. M., Pashkovski, S. L., Abraira, V. E., Adams, R. P., & Datta, S. R. (2015). Mapping subsecond structure in mouse behavior. *Neuron*, 88(6), 1121–1135.
- [15] Wu, C., Huang, H., Dong, J., Yang, L., & Wang, H. (2020). Ar-hmm: A behavior recognition method based on autoregressive hidden markov model. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(1), 88–100.
- [16] Lin, W., Kukleva, A., Possegger, H., Kuehne, H., & Bischof, H. (2023).

TAEC: Unsupervised Action Segmentation with Temporal-Aware Embedding and Clustering.

- [17] Xu, H., Ghosh, G., Huang, P.-Y., Okhonko, D., Aghajanyan, A., Metze, F., Zettlemoyer, L., & Feichtenhofer, C. (2021). VideoCLIP: Contrastive Pre-training for Zero-shot Video-Text Understanding.
- [18] Ye, S., Lauer, J., Zhou, M., Mathis, A., & Mathis, M. W. (2023). AmadeusGPT: a natural language interface for interactive animal behavioral analysis.
- [19] Zhang, Y., Huang, D., Liu, B., Tang, S., Lu, Y., Chen, L., Bai, L., Chu, Q., Yu, N., & Ouyang, W. (2023). MotionGPT: Finetuned LLMs are General-Purpose Motion Generators.
- [20] Zhao, Y., Misra, I., Krähenbühl, P., & Girdhar, R. (2022). Learning Video Representations from Large Language Models.