Radboud University Nijmegen

Faculty of Social Science

Detecting Behavioural Motifs of Mosquitoes Using Deep Variational Embedding of Posture Dynamics

Thesis MSc Intelligent Technology

Author:
Ali Shahbaaz Khan
s1081233

Supervisor:
Dr. Felix Hol
Dr Luca Ambrogioni

Second reader:
Dr. Johan Kwisthout

October 2023

Abstract

Mosquitoes are considered to be one of the most dangerous animals as they have been responsible for countless number of deaths in the past. Even to this day, mosquito-borne diseases such as malaria and dengue continue to claim millions of lives in humid regions around the world. Recent advancements in the study of mosquito behavior have shed light on alterations in the behaviour of infected mosquitoes which facilitates the easier spread of diseases. The quantification of these behavioral alterations offers a promising avenue for the reduction in infection transmission. Investigating the genetic influence on behaviour could help establish correlations between specific genes and their impact on mosquito biting behaviors, potentially leading to the development of disease-resistant "super mosquitoes" that cannot transmit infections to other animals.

In the field of biology and medicine, deep learning approaches have helped scientists explore new avenues. From tumor detection to wildlife tracking, biologists can now explore novel theories and approaches. The combination of different machine learning methodologies to address complex problems has consistently yielded reliable results. Drawing inspiration from the success of deep learning, our work introduces a comprehensive pipeline using techniques such as pose estimation, tracking, dimensionality reduction, and clustering to quantify observed behavioural alterations in mosquitoes.

This project introduces a methodology, wherein we utilize mosquito videos to recognise and analyze their behaviours. We compare the behaviors exhibited by a population of dengue-infected mosquitoes with those of an uninfected control population. Our primary focus lies in the identification of behaviours most crucial for disease transmission and ascertaining whether the distinctions between the control and dengue-infected mosquito populations align with recent findings in the field.

Contents

1 Introduction **3**

2 Background **5**
2.1 Approximate Inference 5
2.1.1 Variational Inference 5
2.1.2 Variational Autoencoder 6
2.2 Hidden Markov Model 8

3 Related Works **10**
3.1 BiteOscope 10
3.2 DeepLabCut 12
3.2.1 Multi-Task Convolutional Networks 12
3.2.2 Assembly of Animals 13
3.3 Variational Animal Motion Embedding 13

4 Methodology **15**
4.1 Data Collection and Preparation 15
4.1.1 Pre-processing 15
4.1.2 Egocentric Alignment 16
4.1.3 Discritization of space 16
4.2 Multi-Decoder β-Variational Autoencoder 16
4.2.1 Variational Lower Bound of our model 19
4.3 Latent Space clustering and Motif Identification 20
4.4 Evaluation 20
4.4.1 Precision, Recall and F1-Score 20
4.4.2 Purity, Normalised Mutual Information and Homogeneity 21

5 Results **22**

6 Discussion and Conclusion **26**

1 Introduction

One of the essential behaviours that a mosquito needs to perform is blood feeding. This behaviour is only exhibited by female mosquitoes as they need the haemoglobin in human blood as an iron source to develop their eggs [1]. This behaviour is also of particular interest to humans as it is the point where pathogens are transferred from the gut of the mosquito into the human blood-stream and causes an infection. Vector-borne disease represents around 17% of infectious diseases around the world[2]. It is estimated that vector-borne diseases(VBD) infect over a billion people and kill approximately a million people every year[3]. Out of these VBDs, Mosquito-borne diseases dominate the spectrum of number of infections and mortality rate [4]. The most common of these diseases are malaria and dengue which causes more than half of the fatalities mentioned before, most of the victims being children under the age of 5[2]. Most of the mosquito-borne diseases are mainly caused by three species of mosquitoes, namely *Aedes Aegypti*, *Anopheles* and *Culex* [5][6][7][8].

Before a mosquito engages in blood-feeding, it exhibits a set of precursory behaviours. Firstly, a mosquito will land on a feeding substrate, which could be human or animal skin and also artificial substrates created for studying mosquito behaviour in a experimental setting. Then, the mosquito will explore the substrate, which would include walking on the substrate till it finds an area of interest. It will then exhibit a behaviour called probing where a mosquito uses its olfactory sensors present in the fore legs to taste the surface[9]. Finally, it will penetrate its proboscis, which is its feeding organ, to the bite substrate and feed until full engorgement. It is hypothesised that viruses like malaria and dengue change this cycle of behaviours [10][11]. Behaviours related to host seeking and feeding are the most effected as it is the point of contact where a mosquito can further spread the virus in other animals or humans. An indicator to determine the infectivity of a disease is its force of infection which is denoted as λ. The force of infection is a fundamental epidemiological parameter that quantifies the instantaneous per capita rate at which susceptible individuals in a population acquire a specific infectious disease. It encompasses various factors, including the current prevalence of the disease in the population, the frequency of contacts between susceptible and infected individuals, and the probability of transmission of the disease during such interactions. The force of infection serves as a key metric for assessing the ongoing transmission dynamics of an infectious disease and plays an important role in epidemiological modeling and disease control strategies. [10] developed an empirical mathematical model to determine the force of infection. They concluded that malaria leads to behavioural altercation in the bite cycle of a mosquito which in turn leads to a higher force of infection. If we can quantify this behavioural altercation, it will give us a deeper insight in determining how dangerous or infectious a virus can be.

One of the major challenges in behavioural neuroscience is the detection and quantification of behaviours. Advances in markerless pose estimation have made the tracking of spatio-temporal posture dynamics possible for animal motion. Packages like DeepLabCut[12] and OpenPose[13] are pose estimation tools which are commonly used to extract posture dynamics. An assumption in performing behaviour quantification using posture dynamics is that information is inherently encoded in these posture dynamics. Therefore, with the help of some dimension reduction, we can extract a representation of the postures which explains the behaviours. Simple approaches would include using basic reduction techniques like Uniform Manifold Approximation Projection(UMAP), t-Stochastic Neighbourhood embedding(t-SNE) or Principle component analysis(PCA) directly on the posture data. However, this may not yield any meaning-

ful behaviours as the raw coordinates do not hold enough information to be clustered without being processed(We will prove this in our results section). To deal with this, there have been many supervised [14][15] and unsupervised [16][17][18] approaches developed to yield a better representation of the posture coordinates which we can cluster to get the expected behaviours. An Unsupervised approach is more suitable to solving the problem primarily because a large number of frames(approx. 1.5 million) will need to be annotated by a human expert which is a tedious and time consuming task. Another advantage is the detection of behaviours which cannot be recognised by a human based on simple observation. Furthermore, we will not have to deal with conflicts in annotation by different experts and hence avoid inducing an annotator bias as different experts interpret behaviours differently.

In this project, we propose an unsupervised approach to cluster posture time-series of mosquitoes. We use a Variational Autoencoder to yield a rich representation of the postures which we then cluster using a Hidden Markov Model. The layout of the document will be as follows: In background, we will introduce all the technique being used to perform unsupervised behaviour classification. In the related works, we will introduce our experimental setup developed by Hol et al[19] which is used to record videos of mosquitoes. We will then talk about DeepLabCut(DLC) which is the package used to extract posture coordinates. Furthermore, we will discuss about a similar approach called Variational Animal Motion Embedding which uses an unsupervised approach to detect behaviours in mice.

2 Background

Within this section, our primary focus centers on Variational Autoencoders (VAEs), as they represent our core methodology for discerning behaviors based on observed data. In the course of our discussion, we will not only elaborate on VAEs but also touch upon the broader concept of variational inference, which constitutes the underlying mechanism driving the functioning of VAEs. Additionally, we shall briefly delve into the realm of Hidden Markov Models (HMMs) and their relevance to our research. Specifically, we will elucidate how HMMs are used for the purpose of inferring latent or hidden states from observable data.

2.1 Approximate Inference

In this section we introduce the concept of bayesian inference and how it is used to estimate an approximate posterior for a model with variational parameters ϕ. Lets define the inference problem mathematically. Suppose we have a dataset $\mathbf{D} = \{x_1, x_2, ..., x_n\}$ representing some recorded observation of coordinates of a moving animal. Lets consider a variable $\mathbf{H}$ to represent the hypothesis which can explain the data. So we can say a distribution $P(D|H)$ represents a density which tells us about what the data looks like given a hypothesis. This distribution is also called the likelihood. Now, we want to know the opposite of this, i.e., the probability density representing observations given data which can also be written as $P(H|D)$, which we also call the posterior. We can simply use Bayes' Theorem to calculate this as :

$$P(H|D) = \frac{P(D|H)P(H)}{P(D)} \tag{1}$$

.

Over here $P(H)$ is also called the prior distribution. We can calculate $P(D)$ as follows :

$$P(D) = \int P(D, H)dH \tag{2}$$

However, this integral is hard to compute due to its high dimensional nature. Therefore, using a naive approach cannot be used to find a solution to this integral. This implies that we need to approximate the solution of this integral. One of the ways to achieve this is through a technique called variational inference.

2.1.1 Variational Inference

Variational Inference which is also called Variational Bayes, is a method to estimate an approximate posterior distribution [20]. This involves replacing the sampling from the posterior distribution $P(H|D)$ with trying to find an approximation of the posterior. This approximate posterior is represented in the form of a variational distribution $q_\phi(H)$ which belongs to a family of densities Q and ϕ are the parameters for this distribution. Variational inference converts the problem of finding the posterior distribution into an optimisation task[21] which can be stated as

$$q^*_\phi(H) = \arg\min_{q \epsilon Q} D_{KL}(q_\phi(H)||p(H|D)) \tag{3}$$

where $D_{KL}(.||.)$ represents the KL-divergence between two distributions which measure the difference in information between two distributions. The aim of the above equation is to find an optimal value of the variational parameters ϕ which minimizes the measure of the approximate posterior distribution $q_\phi(H)$ with the true posterior $p(H|D)$. The formula for the KL-divergence can be written as follows :

$$\begin{aligned} D_{KL}(q_\phi(H)||p(H|D)) &= \int q(H) \log \frac{q(H)}{p(H|D)} dH \\ &= \mathbb{E}_{q(H)}[\log \frac{q(H)}{p(H|D)}] \end{aligned} \tag{4}$$

However, this value still cannot be computed due to the presence of the distribution $p(D)$ because $p(D) = \int p(D|H)p(H)dH$ which is intractable. This can be verified by expanding eq. 4

$$\begin{aligned} D_{KL}(q_\phi(H)||p(H|D)) &= \mathbb{E}_{q(H)}[\log \frac{q(H)}{p(H|D)}] \\ &= \mathbb{E}_{q(H)}[\log q(H)] - \mathbb{E}_{q(H)}[\log p(H|D)] \\ &= \mathbb{E}_{q(H)}[\log q(H)] - \mathbb{E}_{q(H)}[\log \frac{p(H,D)}{p(D)}] \\ &= \mathbb{E}_{q(H)}[\log q(H)] - \mathbb{E}_{q(H)}[\log p(H,D)] + \mathbb{E}_{q(H)}[\log p(D)] \\ &= \mathbb{E}_{q(H)}[\log q(H)] - \mathbb{E}_{q(H)}[\log p(H,D)] + \int q(H) \log p(D) dH \\ &= \underbrace{\mathbb{E}_{q(H)}[\log q(H)] - \mathbb{E}_{q(H)}[\log p(H,D)]}_{\text{-ELBO(q)}} + \underbrace{\log p(D)}_{\text{intractable}} \end{aligned} \tag{5}$$

We can see in the expansion above how the presence of $\log p(D)$ makes the KL-divergence intractable. Moreover, we also see that the value of $\log p(D)$ remains independent of q(H), therefore, the expectation value of $\log p(D)$ is itself. We also see a new term called ELBO or the *Evidence Lower Bound*. This term is the objection function in variational inference.

It should be noted that since the KL-divergence for two different distributions has to be a positive value, we can rewrite equation 5 as

$$\log p(D) = \text{ELBO(q)} + D_{KL}(q_\phi(H)||p(H|D)) \geq \text{ELBO(q)} \tag{6}$$

Equation 6 implies that in order to minimise the KL-divergence, we would need to maximise the ELBO, therefore making it an objective function. Since maximising a function is equivalent to minimising its negative, we can restate the optimisation function ELBO as

$$\text{ELBO(q)} = \mathbb{E}_{q(H)}[\log p(H,D)] - \mathbb{E}_{q(H)}[\log q(H)] \tag{7}$$

2.1.2 Variational Autoencoder

Variational Autoencoders(VAE) [22] are deep generative networks that learn to compress an input signal into a lower dimensional representation, called the latent space, and then regenerate the original signal from this latent space. VAEs have a variety of use cases ranging from computer vision and image processing tasks to bioinformatics and computational biology. There are two ways in which we can view a VAE :

1. VAEs are probabilistic models that use variational inference to fit data(as discussed in section 2.1.1)

2. VAEs are neural networks that are used to perform autoencoding tasks.

Both these views can be considered to complement each other too. In this section we are going to lay out the mathematical framework on which a VAE is trained. The architecture of a basic VAE can be seen in figure 1

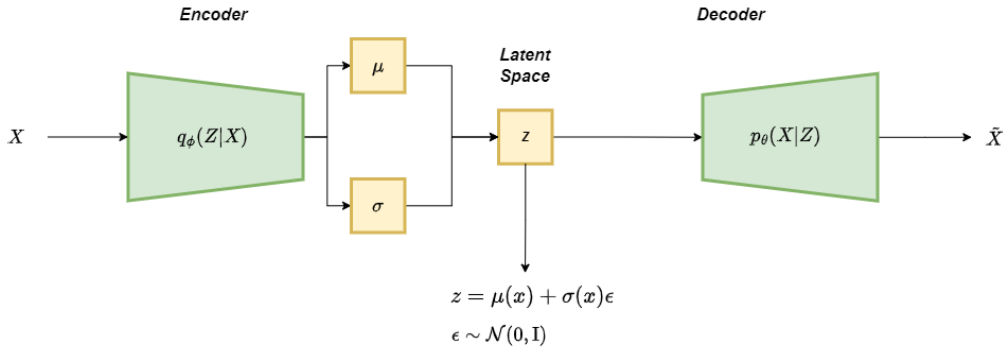

Figure 1: The architecture of a Vanilla Variational Autoencoder

Suppose that there is exists some hidden variable Z that explains the generation of the observation X(see figure 2). Since only X can be observed, we need to find a way to infer the hidden variable Z. VAEs are a way to solve this problem. As the name suggests, a VAE is a type of probabilistic autoencoders that takes an input vector X, compresses it to a lower dimensional space Z, and then regenerates this input from Z which we call $\tilde{X}$. The compression and decompression are performed by an encoder q_ϕ and a decoder p_θ. These functions are represented by a Multi layered Perceptron(MLP) and θ and ϕ are the parameters of these MLPs. The goal is to estimate a variational posterior $q_\phi(Z|X)$. This can be done using Bayes theorem, however, it is still plagued with the problem of intractibility discussed in section 2.1.1. Therefore, we use variational inference to solve this task by approximating the real posterior as discussed in section 2.1.1. We want to optimise a lower bound of our marginal likelihood. We rewrite equation 6 as

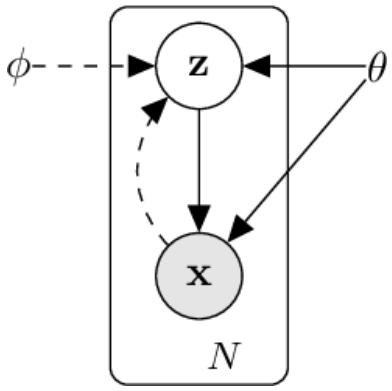

Figure 2: Hidden variable $\mathbf{z}$ explaining observations $\mathbf{x}$ [22]

$$
\begin{aligned}
ELBO(\theta,\phi) &= \sum_{i=1}^{N} \mathbb{E}_{z_i\sim q}[\log p_\theta(x_i,z_i) - \log q_\phi(z_i|x_i)] \\
&= \sum_{i=1}^{N} \int q_\phi(z_i|x_i)[\log p_\theta(x_i|z_i) + \log p_\theta(z_i) - \log q_\phi(z_i|x)_i]dz_i \\
&= \sum_{i=1}^{N} \mathbb{E}_{z_i\sim q}[\log p_\theta(x_i|z_i)] + \sum \mathbb{E}_{z_i\sim q}[\log \frac{p_\theta(z_i)}{q_\phi(z_i|x_i)}] \\
&= \sum_{i=1}^{N} \mathbb{E}_{z_i\sim q}[\log p_\theta(x_i|z_i)] - D_{KL}(q_\phi(z_i|x_i)||p_\theta(z_i))
\end{aligned}
\tag{8}
$$

Willing and Kingma(2014)[22] provided a closed form solution for D_{KL} term in the $ELBO$

$$
D_{KL}(q_\phi(z|x)||p_\theta(z)) = -\frac{1}{2}\sum_{j=1}^{J}(1 + \log(\sigma_j^2) - \mu_j^2 - \sigma_j^2) \tag{9}
$$

Therefore, plugging the value of D_{KL} from equation 9 in the final step of equation 8, we get the loss function on which we need to optimise for training our VAE.

Another issue which arises is calculating gradients for backpropogation during training. To backpropogate, we would need to calculate $\nabla_\phi D_{KL}(q_\phi(z|x)||p_\theta(z))$ which can be problematic because of z being a stochastic variable. The problem is solved by converting it into a deterministic form using the reparameterisation trick introduced by [22]. We reformulate z with a differentiable transformation $g(\epsilon, x)$.

$$
z = g(\epsilon, x) \text{ where } \epsilon \sim \mathcal{N}(0,1) \tag{10}
$$

For a gaussian $z \sim p(z|x) = \mathcal{N}(\mu, \sigma^2)$, we can simply sample from ϵ and shift it by μ and and scale it by σ. This can be formulated as

$$
z = \mu + \sigma \odot \epsilon \text{ where } \epsilon \sim \mathcal{N}(0,1) \tag{11}
$$

where $\odot$ represents the element wise product

2.2 Hidden Markov Model

Hidden Markov Models is a technique based on the concept of Markov Chains. Markov chain is a method for modelling states by telling us the probability of the state being observed. A Markov chain assumes that the probability of a future state being observed depends only on the current state and not the states in the past. To formalise this statement, if we have a set of states $s_1, s_2, ..., s_N$, then we can say

$$
P(state = s_i|s_1, ..., s_{i-1}) = P(state = s_i|s_{i-1}) \tag{12}
$$

Markov models are very useful when determining the probability of observations that occur in a sequence. Hidden Markov Models are an extention to markov chains. They are a probabilistic graphical model which uses a combination of hidden and observed state to capture the underlying structure of sequence. An HMM model is defined by a tuple (H, O, π, A, B). Each of the variables are explained as follows

- $H = \{h_1, h_2, ..., h_N\}$ are a finite set of N hidden states. These states are not observable directly but explain the observations

- $O = \{o_1, o_2, ..., o_T\}$ is a sequence of T observations. These observations are dependent on the hidden states
- $\pi = \{\pi_1, \pi_2, ..., \pi_N\}$ is the distribution of the initial state, therefore, representing a probability of starting in each of the hidden states h_i
- $A = \{a_{ij}\}_{N \times N}$ is state transition probability matrix. Each entry of this matrix represents the probability of transition between states h_i and h_j. This matrix defines the temporal dynamics of the hidden states
- $B = \{b_i(o_i)\}_{N \times T}$ is the probability distribution of the observations. It represents the probability of an observation o_i being generated by a hidden state h_i

Using the Expectation-Maximization(EM) algorithm, the HMM model is trained to find an optimal set of parameter. In the expectation step of E-step, the log-likelihood value of the data is determined using an approximate posterior. In the maximisation or the M-step, the maximisation of the log-likelihood takes place by updating the model parameters. In this step, A, B and π are variables that are updated. These steps are repeated till the model reaches convergence. In this project, we use the Baum-Welch algorithm to perform EM steps and to determine the most likely sequence of hidden state that generate the observations, we use the Viterbi algorithm.

3 Related Works

In this section, we will discuss the previous work that has been utilised and influenced our project. We will begin by introducing BiteOscope, a setup used to record mosquito videos. Following that, we will delve into the functioning of DeepLabCut (DLC), a tool that plays a crucial role in our project for keypoint tracking. Finally, we will provide a brief introduction to VAME, an unsupervised model to detect behaviours of mice.

3.1 BiteOscope

One of the significant challenges encountered by mosquito biologists pertains to the observation of mosquitoes during their feeding process. Traditionally, researchers have resorted to using their own arms as a substrate for mosquitoes to feed upon, allowing for the study of mosquito behavior within a controlled laboratory environment. This observation of mosquitoes can provide valuable insights for researchers aiming to develop strategies to prevent the transmission of infections by virus-infected mosquitoes during biting. Nevertheless, this approach to behavioral research is plagued by many limitations. Firstly, conducting such experiments can prove uncomfortable for the volunteers who have to willingly subject themselves to mosquito bites. Secondly, a major limitation arises from the inability to study infected mosquitoes, primarily due to the risk of infection transmission to the volunteers. In 2020, Hol et al. introduced an economical apparatus known as the "BiteOscope" [19], which addressed these issues.

The BiteOscope was designed to create a favourable environment for stimulating mosquito feeding behavior, eliminating the need for researchers to sacrifice their own skin as a blood source. Figure 3 shows a schematic of the experimental setup of BiteOscope. To encourage mosquitoes to engage in blood feeding, the apparatus must establish an environment that attracts them and enables surface exploration before they penetrate it with their probocis to commence feeding on an artificial blood meal. One of the primary factors influencing mosquito mobility and activity is the ambient temperature[23][24][25][26]. Hence, by maintaining an optimal temperature that encourages exploration, surface probing, and eventual feeding, mosquitoes can be drawn to the surface of the setup. Since the recording occurs through the bite substrate, and real blood is non-transparent, an artificial blood meal was created. This artificial blood meal comprises adenosine triphosphate (ATP), a strong phagostimulant, and sodium ions. The solution is adjusted to match the osmotic pressure of human blood. These factors ensure that the mosquito feeds till full engorgement[27][28]. This artificial blood meal is applied to the exterior of an optically clear flask, which contains water and is sealed with Parafilm, as depicted in Figure 3. Parafilm, a commonly used laboratory membrane, is employed for this purpose. Water within the flask serves as a medium for regulating the surface temperature.

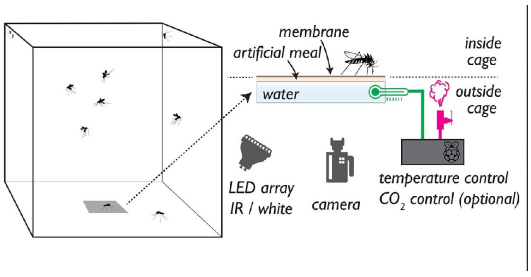

Figure 3: The experimental setup of BiteOscope [19]

To keep the mosquitoes confined to a limited area where they can be recorded, an acrylic cage is used with an opening where the bite substrate is placed. Since the substrate is transparent, imaging can be facilitated by placing a camera outside the cage. The field of view of the camera was set to 8 x 8 as it allowed for the tracking of at least 15 individual mosquitoes(see Fig. 4) simultaneously without any compromise in the resolution of small parts of the mosquito like the stylet or the tip of the legs. Using these videos, Hol et al. created a computational pipeline for pose estimation on a frame-by-frame basis. They trained a convolutional neural network(CNN) based on a deep learning framework called DeepLabCut[12](See section 3.2). Using this framework they tracked the proboscis, head, abdomen, all six legs, the edges of the abdomen and the thorax of the mosquito *Aedes aegypti* and *Aedes albopictus*. Since both these species of mosquitoes are morphologically similar, the same CNN network can be used for tracking the bodyparts. Using this tracking data, they empirically recognised various behaviours. For example, grooming was recognised by observing a characteristic circular motion of the forelegs while all other legs remained stationary. Similarly, walking was detected by observing an oscillatory movement of all the legs relative to the axis if the body of the mosquito. Therefore, this implies that the outputs of the computational pipeline developed by Hol et al. can be used as features for a deep learning behaviour classification model. [15] and [14] proposed a supervised method and [17][16] and [29] proposed an unsupervised method to use trajectory data of *drosophila*, also commonly known as fruitfly, to recognise and classify different behaviours , therefore, validating the method of extracting behaviours from trajectories. In the next few sections, we will explain how we used unsupervised learning to classify behaviours of *Aedes aegypti* mosquitoes.

Figure 4: The field of view of the camera with DLC output plotted over mosquitoes

3.2 DeepLabCut

In the age of rapid advancements in the field of computer imaging, vision and computational analysis, animal tracking and monitoring has significantly improved and become a data-driven task [30]. One of the crucial computation analysis to detect fine-grained behaviours is pose estimation. There exists many deep learning approaches that rely on large quantities of data and have significantly affected the fields of ethology and neuroscience[31][32][33][34]. However, the task of multi-animal pose estimation still remains a challenging task and relies heavily on advances in computer vision research. Generally, the process comprises of three essential steps which we shall now discuss :-

- **Pose estimation** - This step is localizing and detecting different bodypoints, also called keypoints, of the animal being tracked. This is generally done by training a deep learning model with bodypart or keypoint(we shall use these terms interchangeably throughout this section) annotated frames.
- **Assembly** - This step involves the integration of different detected keypoints into an animal. Ideally, there should not be any overlap between keypoints of different animals that are in close proximity.
- **Tracking** - As the name suggests, this step involves tracking the detected animal over a period of time.

However, all the steps mentioned above pose different challenges. Firstly, to make sure a deep learning model can deal with frames in which the animal is obstructed in some way, one has to make sure that there are enough annotated frames where there is close interaction between the animals, which causes the obstructed view in most cases. Secondly, when assembling or grouping different keypoints to form an animal, tracking it between frames can be a difficult task due to similarity in appearance, possible obstruction of view or the animal exhibiting non-stationary behaviours. There have been several proposed methods to solve the problems we discussed before like the use of transformers [35] and the use of associated embeddings [36]. Using these methods and building on the developments in human pose estimation, tools like *SLEAP* [37] and *AlphaTracker* [38] emerged. In 2018, Mathis,A et al. introduced a markerless pose estimation framework called DeepLabCut(DLC) [12]. Later on Lauer, J et al. built on the DLC model to develop a network architecture(see fig. 5) with superior performance, animal assembly method driven by data [39] and they compared the performance of their network on COCO(common objects in context)[40] on four different animal datasets. Lets discuss the individual componenents of DLC which makes it suitable at solving pose estimation and tracking problems

3.2.1 Multi-Task Convolutional Networks

DeepLabCut uses a pre-trained deep Convolutional Neural Network(CNN) to perform keypoint detection. The architecture also consists of multiple de-convolutional layers that perform different tasks. The architectures of these networks are inspired from Residual Nets(ResNets) and EfficientNets. These networks predict a score map and also local refinement fields. The score maps predicted by each of the de-convolution network is combined, where each of the layers learns a representation of the input on a different scale. The output of the DeepLabCut framework is the intensity and the vector field for each keypoint which is also upsampled by the de-convolution layers. The authors use Part Affinity Fields(PAFs) to combine keypoints belonging to the same animal. PAFs calculates a directional vector from one keypoint to another indicating a link. The model

is trained by minimising the L1-loss for the predicted and the ground truth PAFs. For the optimisation, the authors utilise the Adam optimiser and incorporate augmentation techniques as rotation, blurring, and random box covering. The authors also introduce a technique to crop the center of a frame using the keypoint density.

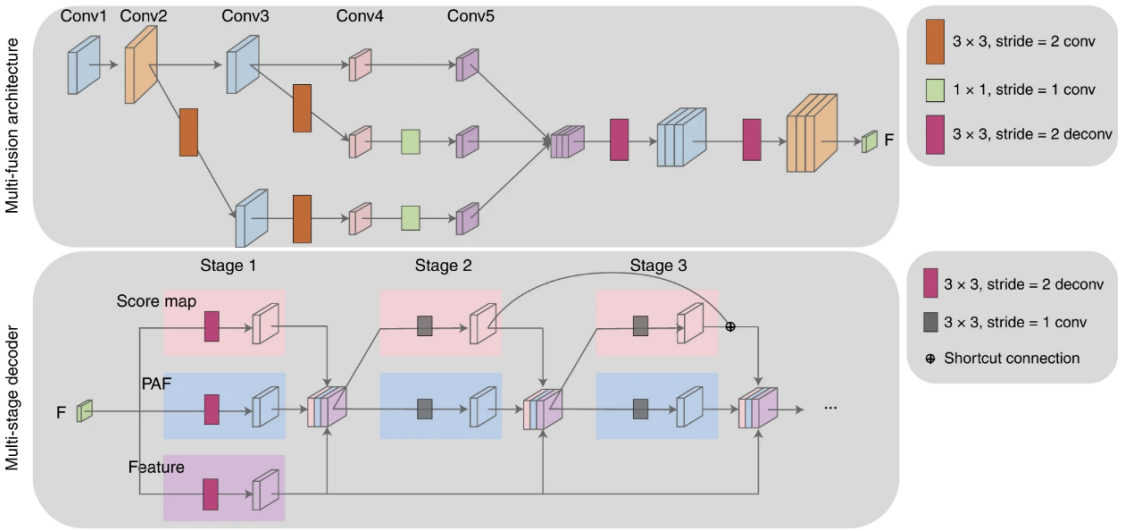

Figure 5: Network arhitecture used for pose estimation and tracking in DeepLabCut [39]

3.2.2 Assembly of Animals

To establish a connection between keypoints belonging to the same animal, a data-driven approach is used by the authors. The CNN models are trained using the complete body representation of an animal in the form of a connected graph. This was proceeded by pruning of certain nodes in the graph based on edge discriminability. Finally, a maximum spanning tree was derived which covered all keypoints with minimum number of edges. The graph selected for animal assembly had the maximum purity and connected keypoints.

Having discussed the animal assemply protocol in the previous paragraph, it is important to note that k-dimensional matching problems in graph theory are considered to be NP-hard. To find a way around this, the authors of DeepLabCut divide the problem into a set of sub-problems. In the graph, the edges representing high affinity costs are identified and based on these edges a individual animal was defined. Using a greedy approach, the remaining edges are sorted based on the affinity cost. To deal with connections that are ambiguous, a calibration can be performed on the entire assembly process by modelling the animals pose as a multivariate gaussian distribution with its parameters determined by the labelled data. All these processes are parallelised to improve efficiency in yeilding a complete set of keypoints for individual animals.

3.3 Variational Animal Motion Embedding

In 2022, researchers at the Leibniz Institute of Neurobiology introduced an unsupervised probabilistic deep learning framework which they called Variational Animal Motion Embedding(VAME)[18]. Since we are dealing with a large quantity of data, an

unsupervised method is suitable as it takes away the need of laborious annotations which can be very time consuming. Moreover, unsupervised behaviour detection can help us obtain a rich and dynamical representation of animal behaviours and can also give us insight into behaviours not noticeable to the human eye. As discussed in section 3.2, tools like DeepLabCut have enabled researchers to perfom efficient multi-body part tracking using supervised methods and when combined with a tool like VAME, we have a powerful pipeline which can help us to extract behaviours from videos. VAME uses a variational autoencoder parameterized with a recurrent neural network to embed the input trajectory data into a lower dimension manifold. This lower dimension representation is then clustered using a Hidden Markov Model(see section 2.2), where each cluster is a hidden state which represents behavioral motifs of the animal.

The authors used VAME to compare two groups of mice: a normal control population of mice and a population of mice with a mutation in the presenelin 1 gene. The VAME pipeline was used to differentiate between the underlying structure of behaviour within the two groups of mice. VAME uses a variational autoencoder to get a latent space representation of the keypoint detection results from DeepLabCut. Then, the authors used a Hidden Markov Model (HMM) to cluster the latent space and extracted 50 different behavioural motifs. These motifs were used to construct a hierarchical tree representation, denoted as $\mathbb{C}$. They also compared the performance of KMeans clustering with HMM and determined HMM is superior and yields more coherent motifs and better performance metrics. Furthermore, this tree structure enabled the comparison of branches with corresponding motif videos which facilitated the grouping of each motif in a general behaviour community. In their analysis, Luxem et al. discovered communities representing a coherent cluster of movements, which were associated with specific actions such as rearing, turning, or walking. Within each community, the motifs can be interpreted as subsets, denoted as $c \in C_i$ (with i represent all detected behaviours), representing variations in these actions. Therefore, the communities that were discovered from the clustering of the latent space provided the authors with a hierarchical understanding of animal behaviour.

4 Methodology

4.1 Data Collection and Preparation

Our dataset was taken from the recordings made by Hol et al. [19]. This dataset consisted video recordings of mosquitoes with dengue and healthy mosquitoes. These videos were passed through the pipeline of DeepLabCut [12](see section 3.2) to get the coordinates of all body parts of the mosquito. A total of 19 body parts were tracked. This subset of body parts was chosen based on observations by experts. We had a total of 46 recordings and after passing it through the DLC pipeline, we obtained the same number of files, each containing 38 sets of (x,y)-coordinates and the likelihood value for each coordinate, in *.pickle* format. In the next few sections, we will talk about how we transformed our dataset to make it suitable for training.

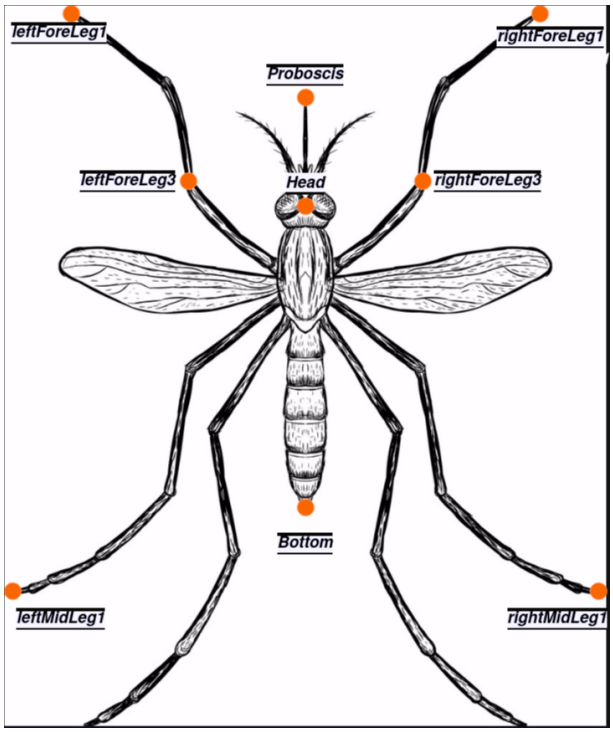

Figure 6: Body parts of Mosquito being tracked

4.1.1 Pre-processing

As discussed before, we have a total of 19 body parts that are being tracked by DLC. Out of these, we select a subset of 9 body parts that can be seen in 6. The decision to select this subset was based on observations made by an expert in mosquito behaviour. Each of the pickle files for each experiment contained the compressed byte-stream data for each mosquito that were detected and tracked in that particular experiment. Then, we extract all the trajectories for each experiment and removed files of individual mosquitoes where more that 60 percent of the tracking data had a low confidence value(≤ 0.5). The trajectories across all individuals and experiments were concatenated into a single vector. This final vector contained the data for 1.046.287 frames.

4.1.2 Egocentric Alignment

Most computer vision libraries calculate coordinates of a frame from the top left corner. Since we use *openCV* for all our computer vision tasks, we have an allocentric version of the coordinates. This increases unnecessary variability in our data as we only care about the movement of mosquito body parts with respect to its own body. For this purpose, we converted out coordinate system into an egocentric form. We define an axis with respect to which we want to calculate our new coordinates. We chose the axis formed by the head and the bottom of the mosquito(refer to figure 6) and every other keypoint is recalculated about this central axis of the mosquito.

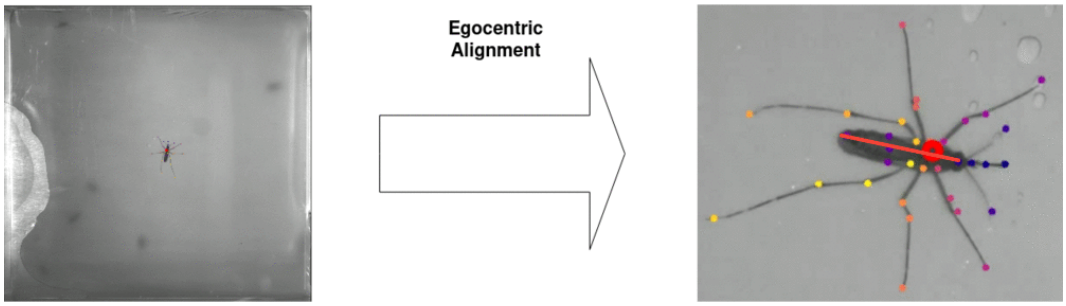

Figure 7: Allocentric to Egocentric conversion. The axis of can be seen as a red line on the mosquito body

4.1.3 Discritization of space

One major limiting factor faced by us during the course of the project was the inherent jitter in the keypoint detection. There are many factors that contributed to this, namely, occlusions or obstruction in the field of view of the camera, movement of the camera while recording and the rapid and unpredictable behaviour exhibited by mosquitoes. This lead to a noticeable increase in inconsistency and variation in the entire dataset. However, quantifying the jitter was a complex problem as there was no way to differentiate between actual movement and jitter since we were only dealing with the posture coordinate data of the mosquito. To overcome this, we came up with a solution which involved the discritization of the entire egocentric coordinate space. To elaborate, we converted our video frames as a structure grid with each grid being 16 by 16 pixels in size. The block size was a parameter determined by observing the amplitude of the jitter. The original coordinates were remapped to the position of the grid in which they fell. This lead to a significant reduction on the impact of the jitter on our model performance. Keypoints exhibiting jitter stayed within the blocks which lead to a stable and consistant representation of out data. However, it should be noted that this approach reduced the resolution of our original data but the overall performance was not impacted. On the contrary, performance appeared to improve after the incorporation of this step as the clusters derived were coherent and uniform.

4.2 Multi-Decoder β-Variational Autoencoder

As previously discussed, our objective is to use an unsupervised approach for the purpose of clustering trajectories into distinct behaviors. To achieve this, we have adopted a variant of Variational Autoencoders (VAE) [22], which consists of a multi-decoder architecture. We shall represent it with the abbreviation MDBVAE from this point on.

This approach of using multiple decoders, referred to as *task programming*, is inspired from the work of Sun JJ. et al [41]. We also add a decoder which predicts the next time-step x_{i+1} as suggested by Srivastava et al. [42]. They reason that to predict the next time-step, the encoder will have to extrapolate information about the position of objects present in the frame and how they are moving. Therefore the encoder will be compelled to encode this information in the latent space. By making our decoders reconstruct features from the latent space, the encoder is compelled to produce a structured depiction within the latent space, thereby yielding a more information-rich representation of the input postures. We hypothesise that the latent space structures along different behaviours. Furthermore, we apply a weight of β to the KL-divergence term, thus promoting a more spread out and disentangled latent space [43]. The utilized features in the decoders and the corresponding behaviors influencing them is detailed in table below

Decoder Features	
Feature	Associated behaviours
Velocity	walking and exploratory behaviours
Foreleg Velocity	probing, grooming
Belly Width	engorging or feeding

To elaborate a more formal version of our approach, we proceed as follows: Let $\mathcal{T} = \{t_1, t_2, ..., t_N\}$ be a set of N trajectories where each trajectory is a set of egocentric discretized time-series of bodypart coordinates $t_i = \{x_1, x_2, ..., x_m\}$ containing $m \times T$ values, where m are the number of (x, y)-coordinates pairs and T is a temporal window over which we define a single data point. To elaborate, we use T consecutive frames as a single data point which will have the shape $m \times T$. We want to learn a D-dimensional latent space. Since $m \times T$ data points are being converted to a vector D, we obtain a compressed form of the trajectories since $D < m \times T$ .For this, we need to estimate a function that learns a non-linear mapping $t_i \rightarrow z_i$. We also want to recover the original input from the latent space, therefore, we would also require a function to learn a non-linear mapping $z_i \rightarrow t_i$. We use a bi-directional Recurrent neural network(bi-RNN) for modelling these functions due to its capability to preserve both forward and backward dependencies of a time-series. The basic building blocks of the bi-RNN were designed using a Gated Recurrent Unit(GRU)[44].

We model our encoder and decoder as a bi-RNN with variational parameters θ and ϕ respectively. We represent the encoder as q_ϕ and our decoder as p_θ. Since we also have decoders predicting features and the next time-step, we represent them as $p_\alpha, p_\gamma, p_\tau$ and p_δ and also model them as a bi-RNN. As, mentioned before a bi-RNN can incorporate information from both the past and the present to make a prediction. This is achieved by adding a hidden layer that is propagating backwards in addition to a hidden layer propagating forward(like a unidirectional RNN). The forward and backward hidden states , $\overrightarrow{H_t}$ and $\overleftarrow{H_t}$ respectively, are determined at each time-step and their recursive update is performed as

$$\begin{aligned} \overrightarrow{H_t} &= tanh(GRU(t_i, \overrightarrow{H_{i-1}})) \\ \overleftarrow{H_t} &= tanh(GRU(t_i, \overleftarrow{H_{i+1}})) \end{aligned} \tag{13}$$

Here GRU represents the gated recurrent unit block. These hidden states are then concatenated to obtain the final hidden state $\mathbf{H}_t$ which is then fed into the output as

$$\mathbf{O}_i = \mathbf{W}_i \mathbf{H}_t + \mathbf{b}_i \tag{14}$$

where $\mathbf{W}_i$ and $\mathbf{b}_i$ are the weights and bias respectively of the output layer $\mathbf{i}$

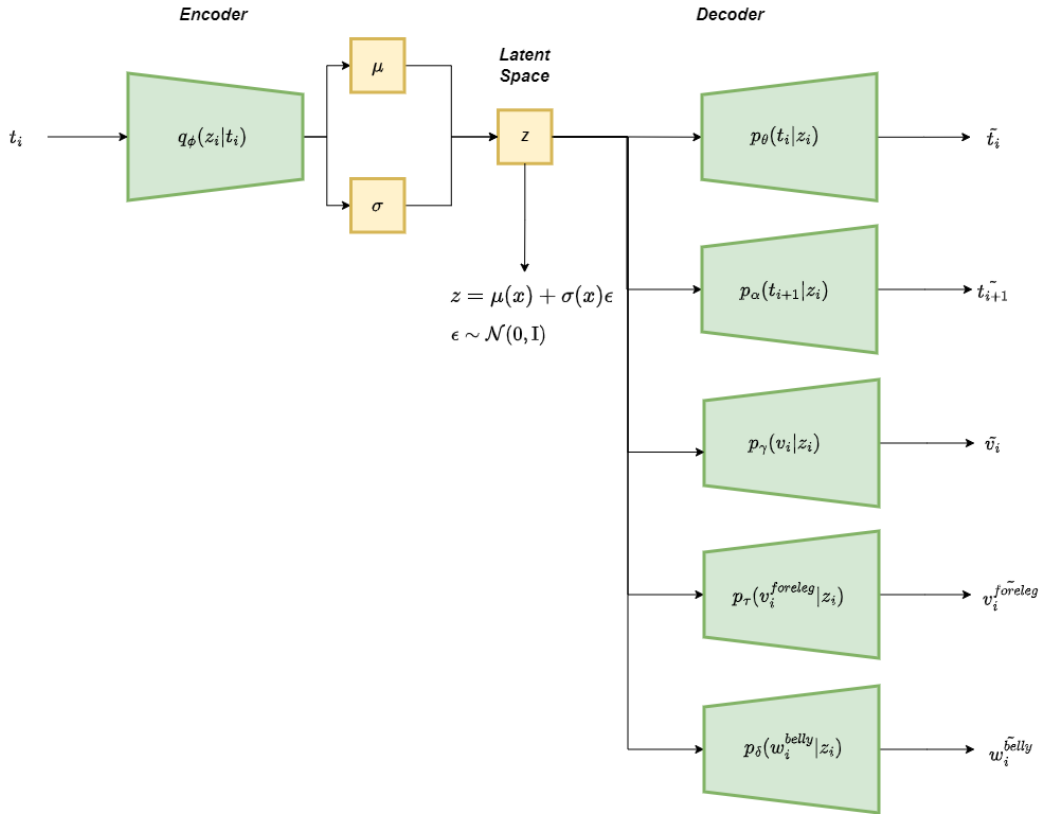

Figure 8: Network architecture of our model

4.2.1 Variational Lower Bound of our model

We want to estimate a hidden variable $\mathbf{Z}$ that encodes the pattern and variations of our trajectories $\mathcal{T}$. Therefore, we can state our probabilistic model as

$$p_\theta(\mathcal{T}, \mathbf{Z}) = p_\theta(\mathcal{T}|\mathbf{Z})p_\theta(\mathbf{Z}) \tag{15}$$

We use Variational Autoencoders [22] to learn the latent space $\mathbf{Z}$. As discussed in section 2.1.1, the real posterior $p_\theta(\mathbf{Z}|\mathcal{T})$ cannot be determined, therefore, we use an approximate posterior $q_\phi(\mathbf{Z}|\mathcal{T})$. We also know a VAE optimises the $ELBO$ written in 8. With our additional decoders, we adjust the $ELBO$ as follows

$$\begin{aligned}\mathcal{L}(\theta, \phi, \alpha, \gamma, \tau, \delta; \mathbf{t}_i) = \mathbb{E}_{z\sim q_\phi(\mathbf{z}|\mathbf{t}_i)}[\log p_\theta(\mathbf{t}_i|\mathbf{z}_i)] + \mathbb{E}_{z\sim q_\phi(\mathbf{z}|\mathbf{t}_i)}[\log p_\alpha(\mathbf{t}_{i+1}|\mathbf{z}_i)] \\ + \mathbb{E}_{z\sim q_\phi(\mathbf{z}|\mathbf{t}_i)}[\log p_\gamma(\mathbf{v}_i|\mathbf{z}_i)] + \mathbb{E}_{z\sim q_\phi(\mathbf{z}|\mathbf{t}_i)}[\log p_\tau(\mathbf{v}_i^{foreleg}|\mathbf{z}_i)] \\ + \mathbb{E}_{z\sim q_\phi(\mathbf{z}|\mathbf{t}_i)}[\log p_\delta(\mathbf{w}_i^{belly}|\mathbf{z}_i)] - \beta D_{KL}(q_\phi(\mathbf{z}_i|\mathbf{t}_i)||p_\theta(\mathbf{z}_i))\end{aligned} \tag{16}$$

where $\mathbf{v}_i, \mathbf{v}_i^{foreleg}$ and $\mathbf{w}_i^{belly}$ represent the velocity, egocentric foreleg velocity and belly width of the mosquito at timepoint $\mathbf{i}$, with γ, τ and δ being their variational parameters. The D_{KL} term is weighted by β to disentangle and spread out the latent space.
We take the prior of our model $p_\theta(\mathbf{z}_i)$ to be a centered isotropic gaussian $\mathcal{N}(0, I)$. In order to sample the latent space, we use the reparameterisation trick[22], therefore yielding, $z_i = \mu_i + \sigma_i \odot \epsilon$, where μ_i and σ_i are mean and variance of the approximate posterior $q_\phi(\mathbf{z}_i|\mathbf{t}_i)$ and ϵ is a unit gaussian. Now, we have $\mathbf{z}_i$ which is then fed to the 4 different decoders. Since, the generative decoders $p_\alpha, p_\gamma, p_\tau$ and p_δ need to reconstruct the original input, we use the mean square error to represent their log likelihood

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - \tilde{x}_i)^2 \tag{17}$$

where x_i is the general representation of an input and $\tilde{x}_i$ is the general representation of an output
We now have a training objective which we can state as

$$\min_{\theta,\phi,\alpha,\gamma,\tau,\delta} \mathcal{L}(\theta, \phi, \alpha, \gamma, \tau, \delta; \mathbf{t}_i) \tag{18}$$

We train our model on 1.47 million frames. For optimisation, we use the Adam optimiser [45] and the hyperparameters used are stated in table below

List of Hyperparameters	
Hyperparameter	Value
batch size	256
β value	5
latent dimensions	30
learning rate	0.0005
time window	15
Encoder layer size	512
Decoder layer size	512

4.3 Latent Space clustering and Motif Identification

After the completion of the training, a latent space denoted as $\mathbf{Z} \in \mathbb{R}^{(D\times N)-T}$ is obtained. The objective is to identify a collection of $\mathbf{k}$ states, denoted as $\mathcal{M} = \{m_1, m_2, m_3, \ldots, m_k\}$ within this latent space, which effectively represents various behavioural motifs. For this, we use the Hidden Markov Model(HMM). We set the number of states to be detected in the latent space as 10 empirically. To convert the labels generated by the HMM model to a human readable form, each cluster is manually annotated by a human expert. The label assignment corresponds to the dominant behavior observed within that cluster. Following this manual annotation process, our analysis revealed the presence of 5 distinct resting clusters, 2 clusters associated with walking behavior, 1 cluster associated with probing behavior, and 2 clusters associated with either incoherent or undefined behaviors. With these manual annotations of our clusters, we proceed to perform the evaluation of our model

4.4 Evaluation

For evaluation, we created a evaluation set of labelled frames. An expert in mosquito behaviour labelled 15.595 frames with behaviours observed. This is done by using a tool with which the annotator could scroll both forward and backward on a frame-by-frame basis and label each frame based on the behaviour observed. The posture data of this set was then passed through our pipeline which yielded a latent space which was then clustered using HMM with the same parameters learned during training. After this, we get 2 vectors, one representing predictions and the other representing the ground truth, other representing the predictions. Since using deep learning to analyse mosquito behaviour is a novel task, we do not have a baseline to compare our model performance with. Therefore, we define our baseline by randomly shuffling the predictions. This would give us an insight if our model performs better than randomly labelled predictions. To perform our evaluation we used the following metrics.

4.4.1 Precision, Recall and F1-Score

Precision is simply defined as the ratio of the number of true positives and the sum of a true and false positives. Recall is the ratio of the number of true positive and the sum of true positives and false negatives

$$Precision(P) = \frac{tp}{tp + fp} \tag{19}$$

$$Recall(R) = \frac{tp}{tp + fn} \tag{20}$$

where tp=True positives, fp=False Positive and fn=False Negatives. F1-score can be defined as the harmonic mean of precision and recall

$$\begin{aligned} \text{F1 score} &= \frac{2}{\frac{1}{P} + \frac{1}{R}} \\ &= \frac{2PR}{P + R} \end{aligned} \tag{21}$$

The performance of our model on these metric are as follows

Overall Performance			
Method	Precision	Recall	F1-Score
Random Shuffle	0.186	0.248	0.212
MDBVAE	0.419	0.341	0.376

4.4.2 Purity, Normalised Mutual Information and Homogeneity

Purity is a metric that measures uniformity of clusters,i.e., measure of a cluster containing a single class. Normalised Mutual Information(NMI) measures how much information can be extracted from a cluster about a different cluster. Homogeneity can be considered similar to purity but more comprehensive. If A are the ground truth labels and B are the model predictions with N being the total number of frames, then we can calculate these metrics as

$$Purity(A, B) = \frac{1}{N} \sum_{a \in A} \max_{b \in B} |a \cap b| \tag{22}$$

Normalised mutual information can be calculated as

$$NMI(A, B) = \frac{MI(A, B)}{\sqrt{E(A)E(B)}} \tag{23}$$

where MI represents the mutual information and E represents the entropy. These values can be calculated as

$$MI(A, B) = \sum_{a \in A} \sum_{b \in B} \frac{|a \cap b|}{N} \log(\frac{N|a \cap b|}{|a||b|}) \tag{24}$$

$$E(A) = -\sum_{i=1}^{|A|} \frac{|a \cap b|}{N} \log(\frac{|a \cap b|}{N}) \tag{25}$$

Over here, the operator $||$ represents the number of frames that have the same prediction and ground truth. Next, we can define homogeneity as

$$Homogeneity = 1 - \frac{E(A|B)}{E(A)} \tag{26}$$

where $E(.|.)$ represents a conditional entropy that can be defined as

$$E(A|B) = -\sum_{i=1}^{|A|} \sum_{j=1}^{|B|} \frac{a \cap b}{|a \cap b|} \log(\frac{a \cap b}{|b|}) \tag{27}$$

The results of the clustering performance of our model is as follows

Clustering Performance			
Method	Purity	NMI	Homogeneity
Random Shuffle	0.849	5.514×10^{-5}	3.841×10^{-5}
MDBVAE	0.852	0.096	0.067

5 Results

We considered the possibility of the posture dynamics already holding the information regarding behaviours and could be segregated without any further processing. To verify if this was the case, we made the use of our labelled data which we created for evaluation. We calculated a Uniform Manifold Approximation Projection(UMAP) of the labelled data along 2 dimensions. If our hypothesis is true, then we should see each labelled behaviour segregated in the calculated projections. However, we observed that this was not the case as we can see in figure 9. Furthermore, a vanilla architecture of VAE was also explored. After creating video sample from the clustered latent space, we found none of the clusters were consistent. Since evaluation of our model is dependant on manual labelling of clusters, it could not be performed due to incoherent clusters. Therefore, we could not perform a comparative analysis of a vanilla VAE with our β-VAE with a multi-decoder architecture. But a UMAP projection of the latent space of the MDBVAE revealed a segregated latent space which can be seen in figure 10

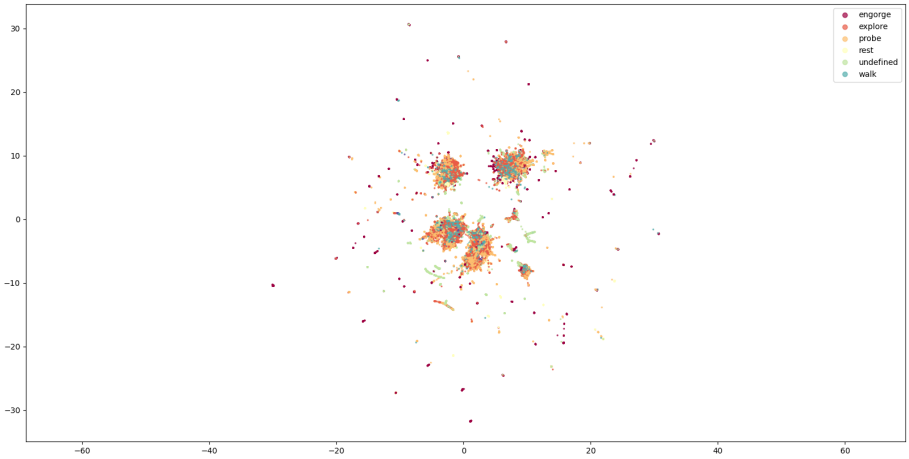

Figure 9: 2 dimensional UMAP projection of posture dynamics

The aim of this project was to quantify the difference behaviours of dengue infected and control mosquitoes. With the labels generated by our trained model, we can now determine behavioural altercations. The variables of interest calculated for this purpose are: the percentage representation of a behaviour in the entire data and the transition probability matrix between behaviours. These values are calculated separately for the dengue and control dataset. The percentage representation of a behaviour is calculated as

$$ratio_m = \frac{N_m}{N_{total}} \tag{28}$$

where N_m is the total number of frames labelled as behaviour m and N_{total} is the total number of frames. We perform our analysis on two different populations of mosquitoes: dengue-infected mosquitoes and control uninfected mosquitoes. The control dataset comprised of 342,091 frames whereas the dengue-infected dataset comprised of 676,813 frames. We analysed the occurance of both walking and probing behaviours within each of these two categories of mosquitoes. For probing, we observed

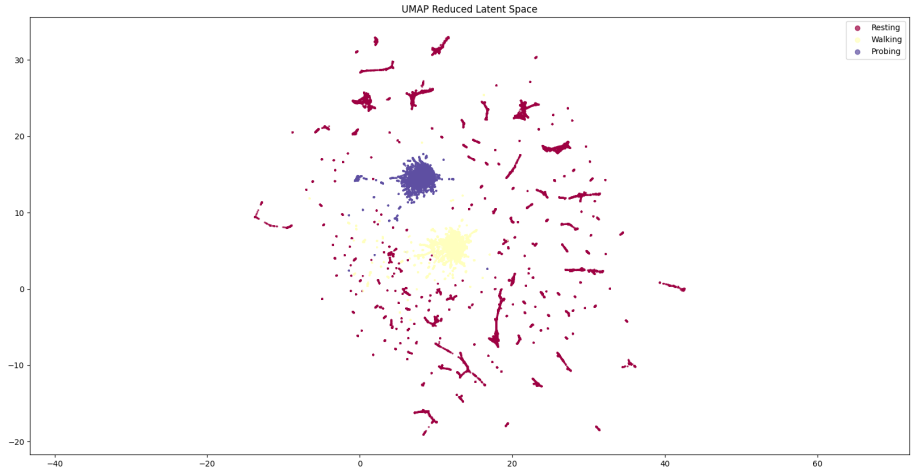

Figure 10: 2 dimensional UMAP projection of MDBVAE latent space

that the control mosquitoes occupied 17% of the total frames, while in the dengue-infected mosquitoes occupied 18.8%. This implies that dengue-infected mosquitoes have a 1.11-fold increase in engaging in probing behaviours(p-value $= 7.026 \times 10^{-47}$ using Fischer's Exact Test) when compared to control mosquitoes. It should be noted that this observed rise in probing aligns with prior findings stated by [46], which indicated that infected mosquitoes tend to exhibit a greater inclination towards probing behavior than non-infected mosquitoes. Conversely, in the context of walking, we noted that control mosquitoes represented 22.79% of the total number of frames, while dengue-infected mosquitoes represented 20.73%. In contrast to probing behaviour, we observed a 1.09-fold reduction in walking behavior among dengue-infected mosquitoes when compared to control mosquitoes(p-vlaue $= 1.938 \times 10^{-38}$ using Fischer's Exact Test).

The hypothesis being considered is that infected mosquitoes tend to engage more in behaviours that could be the point of transmission of the infection. We primarily focus on probing and biting behaviours as they are the most likely point of infection transmission. It should be noted that biting behaviour can be more accurately detected analytically by observing changes in the mosquitoes belly size, as an increase in belly size typically indicates active feeding. Therefore, our analysis primarily centers around probing behaviour. Although we observe an increase in probing behaviour for our dengue-infected population, we cannot draw definitive conclusions from this. This is due to a specific factor: the relatively shorter lifespan of dengue-infected mosquitoes. It has been observed that a significant number of these infected mosquitoes die during the recording of the videos. As a result, there is an imbalance in the number of control and dengue mosquitoes which could lead to biased results. However, it is important to note that since the number of dengue mosquitoes were less in our dataset, we still observe a rise in probing behaviours which seems to be contrary to what we expect. This observation leads us to believe that the hypothesis is valid as a smaller population of dengue mosquitoes tend to probe more than a larger population of control mosquitoes.

In the course of our research, we also analyse the transitions between different behaviours, namely: walking, probing and resting. We create a transition probability

matrix for each of the two population of the mosquitoes. These matrices can be seen in 12 and 11. The objective of creating these transition matrix is to recognise any alterations in the flow of behaviours in the two populations. From the matrices, we analyse the transition between resting and probing. We observe that there is an increase in probability that a mosquito transitions to probing from resting in the dengue population. This further validates our hypothesis mentioned in the previous paragraph. Another interesting finding was observed in transitions from resting to walking and also probing to walking. There was a reduction in the transition probability between these behaviours for the dengue population which. This further entails that mosquitoes infected with dengue tend to engage less in exploratory behaviours and would start feeding at the point where they land on a bite substrate. Therefore from this analysis of our dataset, we can say that dengue infected mosquitoes tend to engage more in probing behaviour which, in turn, increases the risk of disease transmission. Furthermore, dengue mosquitoes do not explore a surface to find an ideal spot for feeding and prefer to start feeding at the point of contact. Our analysis reveals a clear altercation in the behaviours between dengue and control mosquitoes which warrants further research in this field to reach a conclusive finding.

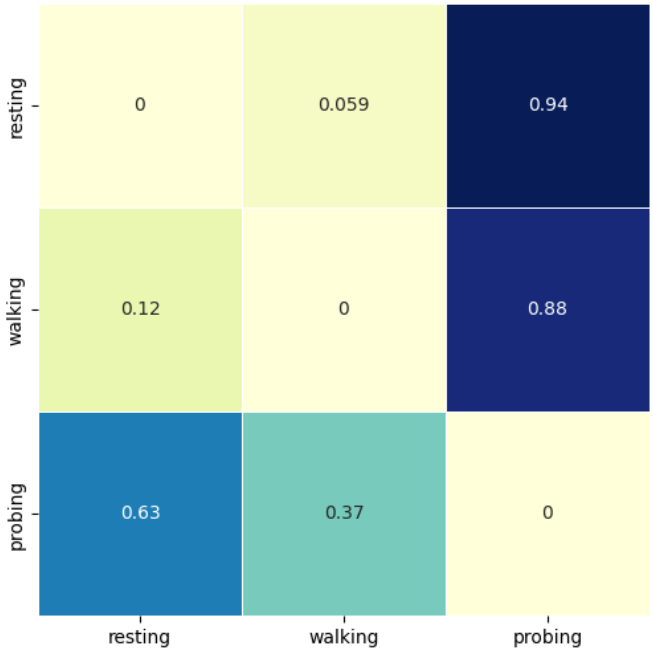

Figure 11: Behaviour transition matrix of dengue mosquitoes

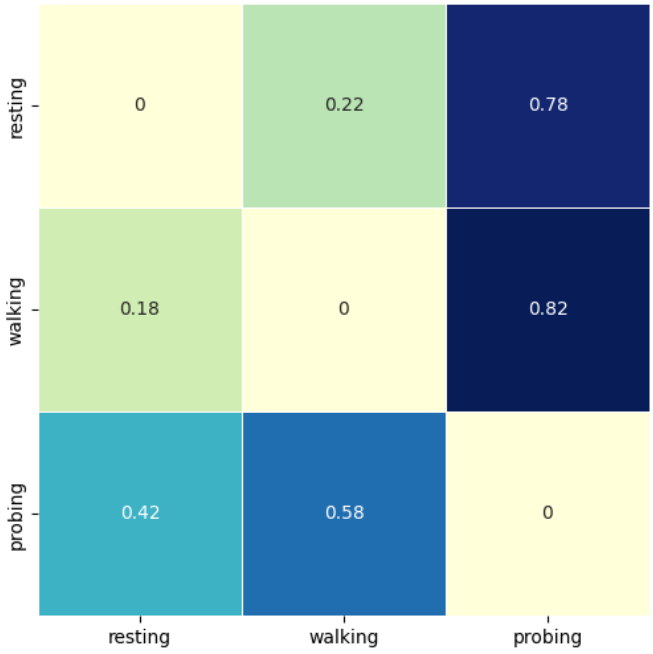

Figure 12: Behaviour transition matrix of control mosquitoes

6 Discussion and Conclusion

In this project, we have created a novel methodology to quantify altercations in behavioural patterns exhibited by infected and uninfected mosquitoes by observing their posture dynamics. Our approach is built upon advances in pose estimation methods and tools like DeepLabCut. Furthermore, we make use of a multi-decoder architecture for our VAE to get a more consistent and distinct representation of behaviours in the latent space. We further use the hidden markov model to segregate and cluster these behaviours in the latent space. This yields a label for each frame in a video which we can further use to create validation videos of each cluster which we can use to analyse the performance of our model.

Our pipeline yielded consistent clusters representing walking, probing and resting behaviours. Using these clusters we could distinguish between how these behaviours look for dengue and control populations of mosquitoes. We were plagued with a challenge of not having a baseline to compare the performance of our model as most of the current research focuses on mice. Since the physiology of mice is very different from mosquitoes, we could not mice-based performance metrics as a baseline. Therefore we resorted to randomly shuffling the predictions to create our own baseline.

The performance of our model may appear subpar when assessed in isolation based on our defined metrics. Nevertheless, when compared against randomly assigned labels, we observe a significant improvement. The subpar isolated performance can be attributed to our evaluation approach. Each cluster is labeled by human annotators with the behaviour which appears to be dominant in the cluster. Clusters showing a diverse range of behaviours, undefined behaviors, or instances where there were occlusions leading to inaccurate keypoint detection are designated as "undefined" and are excluded from the evaluation. This exclusion introduces a degree of uncertainty regarding the true number of false negatives that were produced by our model. However, using human annotations to perform evaluation is a standard approach used by most researchers in animal behaviour. This could be stated as a limitation of using an unsupervised approach.

Another limitation of our approach is determining the optimal number of states that our hidden markov model needs to identify in the latent space. As we determine this value empirically, we do not know what the optimal value would be. [18] showed in their research that an increased number of states leads to better metrics. However, we observed the contrary. The cluster videos seem to become less coherent with an increase in the number of states. In particular, we lost the walking cluster as a distinct cluster as it was spread out across multiple clusters. Additionally, we encountered challenges associated with jitter in keypoint detection due to the unpredictable and rapid movements of mosquitoes. Our solution to this problem involved discretizing the space, which led to a reduction in the resolution of the data. This reduction could lead to loss in the detection of fine-grained behaviours that cannot be captured by the human eye.

In summary, the problem we are addressing presents several inherent limitations. To reduce the effects of these limitations, we have adapted existing methodologies designed for mice data. Since our model does successfully yield distinct clusters for certain behaviors, we perform further analysis to find differences in the behaviour of dengue-infected and control mosquitoes. Our results seem to be in accordance with existing research on mosquito behaviour altercations when infected which leads us to believe that our methodology does help us answer our research questions. The findings of our research can have a significant impact on the field of mosquito biology. By adapting our pipeline

to their experimental setups, mosquito biologists can gain insights into various factors affecting mosquito behaviour. One notable application is the examination of how specific mosquito repellent substances, such as DEET, influence mosquito behaviour. This could lead researchers to develop a more potent mosquito-repellent cream which could reduce the transmission of infection. Moreover, this research could pave the way for an exploration of the genetic factors that affect mosquito behaviour. Using gene-editing techniques such as CRISPR-Cas, researchers can selectively deactivate certain genes and assess their impact on mosquito biting behaviour. The objective of this genetic analysis would be to discover strategies for reducing infection transmission. We conclude that our model can serve as a valuable tool in assisting researchers in their quest to find innovative solutions to this critical issue.

References

[1] Guoli Zhou, Pete Kohlhepp, Dawn Geiser, Maria Frasquillo, Luz Vazquez-Moreno, and Joy Winzerling. Fate of blood meal iron in mosquitos. *Journal of insect physiology*, 53:1169–78, 11 2007.

[2] Geneva World Health Organization. Fate of blood meal iron in mosquitos. *World Health Organization*, March 2020.

[3] World Health Organization et al. A global brief on vector-borne diseases. Technical report, World Health Organization, 2014.

[4] Chapter 2 - mosquito-borne diseases. In Adnan I. Qureshi, editor, *Zika Virus Disease*, pages 27–45. Academic Press, 2018.

[5] Giovanni Benelli and John C Beier. Current vector control challenges in the fight against malaria. *Acta Tropica*, 174:91–96, 2017.

[6] Nicole L Achee, John P Grieco, Hassan Vatandoost, Gonçalo Seixas, Joao Pinto, Lee Ching-Ng, Ademir J Martins, Waraporn Juntarajumnong, Vincent Corbel, Clement Gouagna, et al. Alternative strategies for mosquito-borne arbovirus control. *PLoS neglected tropical diseases*, 13(1):e0006822, 2019.

[7] Nanzaburo Omori. A review of the role of mosquitos in the transmission of malayan and bancroftian filariasis in japan. *Bulletin of the World Health Organization*, 27(4-5):585, 1962.

[8] André Barretto Bruno Wilke and Mauro Toledo Marrelli. Paratransgenesis: a promising new strategy for mosquito vector control. *Parasites & vectors*, 8(1):1–9, 2015.

[9] Emily J Dennis, Olivia V Goldman, and Leslie B Vosshall. Aedes aegypti mosquitoes use their legs to sense deet on contact. *Current Biology*, 29(9):1551–1556, 2019.

[10] Lauren J Cator, Penelope A Lynch, Matthew B Thomas, and Andrew F Read. Alterations in mosquito behaviour by malaria parasites: potential impact on force of infection. *Malaria journal*, 13(1):1–11, 2014.

[11] Benjamin Wong Wei Xiang, Wilfried AA Saron, James C Stewart, Arthur Hain, Varsha Walvekar, Dorothée Missé, Fréderic Thomas, R Manjunatha Kini, Benjamin Roche, Adam Claridge-Chang, et al. Dengue virus infection modifies mosquito blood-feeding behavior to increase transmission to the host. *Proceedings of the National Academy of Sciences*, 119(3):e2117589119, 2022.

[12] 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, 2018.

[13] Nobuyasu Nakano, Tetsuro Sakura, Kazuhiro Ueda, Leon Omura, Arata Kimura, Yoichi Iino, Senshi Fukashiro, and Shinsuke Yoshioka. Evaluation of 3d markerless motion capture accuracy using openpose with multiple video cameras. *Frontiers in sports and active living*, 2:50, 2020.

[14] 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, 2013.

[15] Jamey Kain, Chris Stokes, Quentin Gaudry, Xiangzhi Song, James Foley, Rachel Wilson, and Benjamin De Bivort. Leg-tracking and automated behavioural classification in drosophila. *Nature communications*, 4(1):1910, 2013.

[16] 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, 2014.

[17] Adam J Calhoun, Jonathan W Pillow, and Mala Murthy. Unsupervised identification of the internal states that shape natural behavior. *Nature neuroscience*, 22(12):2040–2049, 2019.

[18] Kevin Luxem, Petra Mocellin, Falko Fuhrmann, Johannes Kürsch, Stephanie R Miller, Jorge J Palop, Stefan Remy, and Pavol Bauer. Identifying behavioral structure from deep variational embeddings of animal motion. *Communications Biology*, 5(1):1267, 2022.

[19] Felix JH Hol, Louis Lambrechts, and Manu Prakash. Biteoscope, an open platform to study mosquito biting behavior. *Elife*, 9:e56829, 2020.

[20] Geoffrey E Hinton and Drew Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on Computational learning theory*, pages 5–13, 1993.

[21] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

[22] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[23] TP Healy, MJW Copland, A Cork, A Przyborowska, and JM Halket. Landing responses of anopheles gambiae elicited by oxocarboxylic acids. *Medical and Veterinary Entomology*, 16(2):126–132, 2002.

[24] Chloe Greppi, Willem J Laursen, Gonzalo Budelli, Elaine C Chang, Abigail M Daniels, Lena Van Giesen, Andrea L Smidler, Flaminia Catteruccia, and Paul A Garrity. Mosquito heat seeking is driven by an ancestral cooling receptor. *Science*, 367(6478):681–684, 2020.

[25] Roman A Corfas and Leslie B Vosshall. The cation channel trpa1 tunes mosquito thermotaxis to host temperatures. *elife*, 4:e11750, 2015.

[26] Paula F Zermoglio, Eddy Robuchon, María Soledad Leonardi, Fabrice Chandre, and Claudio R Lazzari. What does heat tell a mosquito? characterization of the orientation behaviour of aedes aegypti towards heat sources. *Journal of insect physiology*, 100:9–14, 2017.

[27] Laura B Duvall, Lavoisier Ramos-Espiritu, Kyrollos E Barsoum, J Fraser Glickman, and Leslie B Vosshall. Small-molecule agonists of ae. aegypti neuropeptide y receptor block mosquito biting. *Cell*, 176(4):687–701, 2019.

[28] Rachel Galun, Y Avi-Dor, and M Bar-Zeev. Feeding response in aedes aegypti: stimulation by adenosine triphosphate. *Science*, 142(3600):1674–1675, 1963.

[29] Liangyu Tao, Siddhi Ozarkar, Jeffrey M Beck, and Vikas Bhandawat. Statistical structure of locomotion and its modulation by odors. *Elife*, 8:e41235, 2019.

[30] Roland Kays, Margaret C Crofoot, Walter Jetz, and Martin Wikelski. Terrestrial animal tracking as an eye on life and planet. *Science*, 348(6240):aaa2478, 2015.

[31] Mackenzie Weygandt Mathis and Alexander Mathis. Deep learning tools for the measurement of animal behavior in neuroscience. *Current opinion in neurobiology*, 60:1–11, 2020.

[32] Daniel Schofield, Arsha Nagrani, Andrew Zisserman, Misato Hayashi, Tetsuro Matsuzawa, Dora Biro, and Susana Carvalho. Chimpanzee face recognition from videos in the wild using deep learning. *Science advances*, 5(9):eaaw0736, 2019.

[33] Maxime Vidal, Nathan Wolf, Beth Rosenberg, Bradley P Harris, and Alexander Mathis. Perspectives on individual animal identification from biology and computer vision. *Integrative and comparative biology*, 61(3):900–916, 2021.

[34] Alexander Mathis, Steffen Schneider, Jessy Lauer, and Mackenzie Weygandt Mathis. A primer on motion capture with deep learning: principles, pitfalls, and perspectives. *Neuron*, 108(1):44–65, 2020.

[35] Lucas Stoffl, Maxime Vidal, and Alexander Mathis. End-to-end trainable multi-instance pose estimation with transformers. *arXiv preprint arXiv:2103.12115*, 2021.

[36] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. *Advances in neural information processing systems*, 30, 2017.

[37] TD Pereira, N Tabris, J Li, S Ravindranath, ES Papadoyannis, ZY Wang, DM Turner, G McKenzie-Smith, SD Kocher, AL Falkner, et al. Sleap: Multi-animal pose tracking. biorxiv 2020. *Google Scholar*.

[38] Zexin Chen, Ruihan Zhang, Hao-Shu Fang, Yu E Zhang, Aneesh Bal, Haowen Zhou, Rachel R Rock, Nancy Padilla-Coreano, Laurel R Keyes, Haoyi Zhu, et al. Alphatracker: a multi-animal tracking and behavioral analysis tool. *Frontiers in Behavioral Neuroscience*, 17:1111908, 2023.

[39] Jessy Lauer, Mu Zhou, Shaokai Ye, William Menegas, Steffen Schneider, Tanmay Nath, Mohammed Mostafizur Rahman, Valentina Di Santo, Daniel Soberanes, Guoping Feng, et al. Multi-animal pose estimation, identification and tracking with deeplabcut. *Nature Methods*, 19(4):496–504, 2022.

[40] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[41] Jennifer J Sun, Ann Kennedy, Eric Zhan, David J Anderson, Yisong Yue, and Pietro Perona. Task programming: Learning data efficient behavior representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2876–2885, 2021.

[42] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852. PMLR, 2015.

[43] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2016.

[44] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[45] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[46] PA Rossignol, JM Ribeiro, and A Spielman. Increased intradermal probing time in sporozoite-infected mosquitoes. *The American journal of tropical medicine and hygiene*, 33(1):17–20, 1984.