Honor's Thesis

Ishan Saran

01.29.20

1 Introduction

1.1 Potential section: Why study behavior?

→ treat behavior as evolutionary trait, better understand evolutionary mechanisms
→ pharmocologists would like to better understand drugs
→ developmental biologists want to understand developmental growth
→ pilots care about it to increase vigilance & alertness during flights

1.2 Defining the problem

With the advent of better technology, there has been a lot of work in recent years in the field of neuroscience to assemble and analyze large swaths of neural data [1] [20] [14] [13] [11]. The connectome of *Caenorhabditis elegans* has been fully realized [2] and that of *Drosophila melanogaster* is soon on its way [19], with minimally mesoscale models being built for other organisms [16] [9]. However, a rich description of the anatomy and physiology of the brain does not immediately translate to an understanding of the output of the brain. Namely, if we consider the input to be the set of neuronal firing patterns and environment an organism is placed in, the output would be something along the lines of behavior.

That a well-defined, precise, mathematically rich description of behavior is a necessary precursor to understanding a mapping from neural circuitry to behavior has been articulated by previous authors [12] [3] [7]. The problem lies in the fact that there is no single precise, meaningful description of behavior. This may be necessarily so, behavior is rich and complex. It operates on a variety of both time and length scales, it is context-dependent and the set external and internal stimuli which modulate behavior are not given equal weightage. They are hierarchically ordered and that, too, is changing over time. What numbers can capture the essence of behavior across these scales while still being comprehensible to us? This is an open question, one which will require a bit of work on the theoretical level. As a result of this, the field as a whole has seen a continuum of approaches.

1.3 Traditional approaches to quantifying behavior

Traditionally, ethologists have studied animals in the lab setting. The classic picture that comes to mind is the rat-in-a-maze experiment, where an animal is put in an environment outside of its natural habitat and observed under a set of contrived conditions. While this approach promises reproducibility and interpretable results, the scope of the behavior is incredibly limited and there is no guarantee at all that the behavior an animal is displaying is a part of its natural repertoire of actions. What if an animal "evolves" a set of actions just to please the researcher? Moreover, even if an animal is studied in its natural habitat, what numbers of the animal would provide a meaningful representation of its behavior? If the goal is to study a specific behavior of an animal, such as bird singing, then sometimes a specialized set of numbers can describe the narrow repertoire of behavior, such as song frequency or HVC brain volume, and this has certainly been done [18] [17]. However, with other behaviors, such as aggression, the definition any researcher chooses, or the set of actions which they count as aggressive, will have naturally imposed researcher biases. This is true even if they wish to describe the entire repertoire of behavior. If a researcher wants to minimize bias, they must take themselves out of the analysis process. In other words, rather than themselves defining the behavior, it should fall out naturally from the behavioral analysis.

1.4 Stereotyped behaviors & modern approaches

While the set of all possible actions an animal might possess is quite large, theoretically limited only by the biomechanical limits of their morphology, animals tend to occupy only a fraction of the total postural space. The movements which they spend a lot of time performing are known as "stereotyped behaviors" and are well-documented in the scientific literature. For example, head-bobbing in pigeons [8], cribbing in horses [15], and wing grooming in bees [10] are all well studied phenomena. Stereotyped behaviors offer a convenient set of actions reproducible in a lab setting. However it is not certain whether stereotyped behaviors are merely a convenient way of representing behavior in an interpretable manner or whether they are a fundamental manner in which behaviors decompose. Berman (2014) was the first to conceptualize behavior as a trajectory through posture space, that is: in the space of all possible body orientations of an animal, as an animal moves from one posture to another, it exhibits a behavior. Those trajectories which it traverses through more times relative to other trajectories can then be thought of as stereotyped behaviors [4]. Thinking of behavior in this way is beneficial because behavior is intrinsically a dynamical variable and has led to significant discoveries in the nuances of animal behavior [5]. Yet this is not the end of the story. The underlying states which explain the behavior is developed independently of the environment the animal is placed in, whereas sensory information based on environmental changes drive behavior. Others have identified the internal states which drive social behavior, for example, with a mix of hidden Markov Models (HMMs) and generalized linear models (GLMs) [6]. In my thesis I present yet another way of quantifying behavior: using RNNs to build a representation of the underlying neural dynamics.

1.5 RNN to represent behavior

We develop a recurrent neural network (RNN) model to build a dynamical representation of the underlying forces driving the movement of *Drosophila melanogaster*. The idea is that by building an RNN with many more parameters than necessary to capture the fly's movements, what will result is a chiefly low dimensional representation embedded in a high dimensional space. Then, the structure of the resulting manifold can be visualized with dimensionality reduction techniques to analyze the structure. Our hypothesis is that the representation that the RNN builds will be cleaner than the representation the raw postural data itself yields.

There are many reasons to believe an RNN will be able to capture the underlying dynamical system governing the fly's actions. For one, the underlying hidden states of the RNN are inherently dynamical - they vary as a function of time. Secondly, the "memory" associated with RNNs lend naturally to the fact that behaviors operate across scales - the RNN can learn connections on both short and long timescales as its architecture naturally lends to that. RNNs also act as denoising filters, if they Finally, and this is much down the line, there is a sense in which behaviors operate on many scales - the kinematics, the behaviors, and the "mood" of the animal. RNNs can be linked to each other in a sequential manner to form an "encoder-decoder architecture" where a much lower dimensional representation must be learned in order to translate the input to the output. A similar "encoder-decoder architecture" exists in the brain - there must be data compression of some sort from the brain to the set of movements, since the number of motor neurons and neurons in the neck are orders of magnitude fewer than those in the brain. Thus, one could hope an RNN could learn the method in which the brain encodes information as it is being sent down the bottleneck that is the neck.

2 Data Used

The data was taken from a data repository which contains the outputs of LEAP, LEAP Estimates Animal Pose, a tracking software to track selected points on an animal. There are 32 tracked points in total, described in Figure 1 below. There are a total of 30 prediction files, each with $N \sim 3.6 \times 10^5$ frames. For each of the 32 joints both x and y positions were tracked, leading to 30 (N x 2 x 32) time series arrays, on the order of 10^8 frames in total. The data is stored in self-describing HDF5 files, a file format which supports hierarchical, large datasets.

2.1 Fly recording experiments

The original videos of the freely moving fruit fly consisted of 59 male and 51 female *D. melanogaster* (Oregon-R strain) in a circular arena. The flies were restricted to a 2-D plane of movement and the arena was coated in a repellant silane compound to minimize long bouts of upside-down walking. The camera recorded at 100 Hz and the size of the frames were a 200 x 200 pixel square containing the fly. Each fly was recorded for periods of 1 hour, yielding 3.6×10^5 frames for each individual and 4×10^7 frames total. The flies were all of similar age (time after eclosion was controlled for) and the time (to control for circadian rhythm specific behaviors) and temperature (to control for temperature specific behaviors) at which the recording was conducted was held constant. For more information on the original fruit fly videos, including the arena and camera specification, see the imaging apparatus in Berman et. Al 2014.

2.2 Pose estimation with LEAP

LEAP takes videos of animals as input, videos of the aforementioned freely moving fruit flies in my case, along with user annotated frames and predicts the animal pose for every frame given. LEAP uses a convolutional neural network (CNN) to generate an estimate and a confidence map for each of the joint position time series. The network architecture is simple, generalizable, and requires only a few labeled frames (order 10) to generate estimations. LEAP consists of a 15-layer CNN, with a set of convolution-max pooling layers whose weights are updated during training. The strength of LEAP lies in the fact that only a relatively few number of iterations are required to obtain a set of initial values which are fairly good. Then, one can iteratively correct the generated labels with a far reduced labeling time per frame. The authors note the network is able to achieve ¡2.5 pixel error (2-3% of the flys body length) in 74% of the data with only ten labeled images, in some cases. The data set I used had 1,500 labeled images to train the network and ¡3 pixel error in 87% of the test set (N = 168 test frames, from 7 held-out flies).

For more information on the network architecture and training/benchmark methods, see Pereira 2019.

3 Methods to fit data

3.1 Converting to joint angle time series

First, the position time series was converted to a joint angle time series. For each body point tracked except for points 4 and 5, the angle between two vectors: the mesothoracic phragma-neck vector (points 4 & 5 in Figure 1) and the mesthoracic-point in question vector was calculated. Converting from positions to joint angles was beneficial because it reduced the dimensionality of the problem from tracking 64 coordinates (x & y positions of 32 points tracked) to 30 coordinates (the angles except for the mesothoracic phragma and neck, which

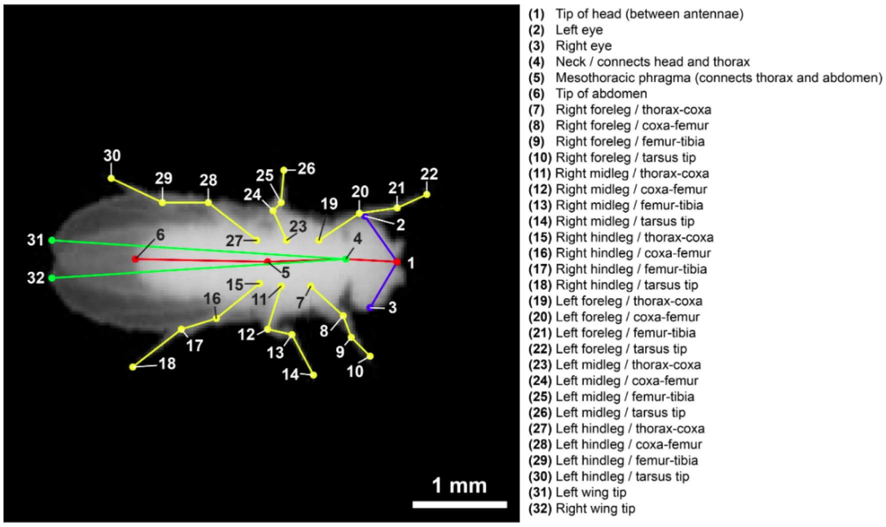
30
29
28
25
26
24
19
20
21
22
23
2
31
6
27
4
1
32
15
5
16
11
7
3
12
8
17
9
13
18
14
10
1 mm
(1) Tip of head (between antennae)
(2) Left eye
(3) Right eye
(4) Neck / connects head and thorax
(5) Mesothoracic phragma (connects thorax and abdomen)
(6) Tip of abdomen
(7) Right foreleg / thorax-coxa
(8) Right foreleg / coxa-femur
(9) Right foreleg / femur-tibia
(10) Right foreleg / tarsus tip
(11) Right midleg / thorax-coxa
(12) Right midleg / coxa-femur
(13) Right midleg / femur-tibia
(14) Right midleg / tarsus tip
(15) Right hindleg / thorax-coxa
(16) Right hindleg / coxa-femur
(17) Right hindleg / femur-tibia
(18) Right hindleg / tarsus tip
(19) Left foreleg / thorax-coxa
(20) Left foreleg / coxa-femur
(21) Left foreleg / femur-tibia
(22) Left foreleg / tarsus tip
(23) Left midleg / thorax-coxa
(24) Left midleg / coxa-femur
(25) Left midleg / femur-tibia
(26) Left midleg / tarsus tip
(27) Left hindleg / thorax-coxa
(28) Left hindleg / coxa-femur
(29) Left hindleg / femur-tibia
(30) Left hindleg / tarsus tip
(31) Left wing tip
(32) Right wing tip

Figure 1: Skeleton of fly with points tracked

redefined the x-axis), which led to quicker computations. Joint angles are also useful because they are independent of body orientation and body size and although the dataset from Pereira 2019 came with aligned data, in principle this is not required for angle calculation.

The angle calculation was done for each point (take, point 7, the right foreleg, for example) by defining two vectors: $\vec{v}_4$, the vector from the mesothoracic phragma (point 5) to the neck (point 4) and $\vec{v}_i$, the vector from the mesothoracic phragma to the point of interest ($\vec{v}_7$ for the right foreleg). Each of the angles with respect to the original coordinate plane's x-axis was calculated by taking the arctangent of the $\frac{y}{x}$ values for each of the vectors to get θ_1 and θ_2, as shown in Figure 2 below. Then, the angle α between the two vectors is

$$\alpha = \begin{cases} \theta_2 - \theta_1, & \text{if } \theta_2 - \theta_1 \geq 0 \\ \theta_2 - \theta_1 + 2\pi, & \text{if } \theta_2 - \theta_1 < 0 \text{ and } i \in \{8, 9, 10, 17, 18, 31\} \\ \theta_2 - \theta_1, & \text{if } \theta_2 - \theta_1 \leq 0 \\ \theta_2 - \theta_1 - 2\pi, & \text{if } \theta_2 - \theta_1 > 0 \text{ and } i \in \{6, 20, 21, 22, 29, 30, 32\} \end{cases}$$

where i corresponds to the *i-th* index of the vector $\vec{v}_i$, for each of the points tracked, going from 1 to 32, excluding 4 and 5 (since it wouldn't make sense to draw the vector over itself, the angle would just be 0). Arctangent has a range of $[-\pi, \pi]$ and for any vector above the x-axis would return a positive value from $[0, \pi)$ and any vector below x-axis would return a negative value from $(-\pi, 0]$. The indices in the second conditional correspond to the points on the right side

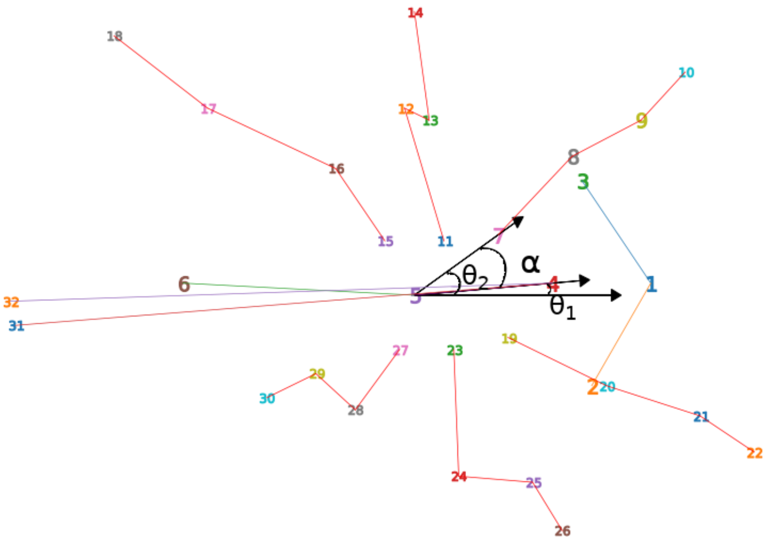

Figure 2: Defined angles for angle conversion calculation

of the fly and the indices in the fourth conditional correspond to the points on the left side of the fly. As seen in figure 2, though, sometimes the fly's wings would cross the x-axis defined in the original coordinate system, which would shift the value from π to $-\pi$ or vice versa rapidly, causing a discontinuity in the time series plots. This occurred for the wings, femur-tibia and tarsus tip connections (the points more lateral/further away from the body) and during grooming behaviors especially. In order to make the joint angle time series continuous, the angles which belonged to the right side of the fly were redefined to be 2π plus (or minus, if the points were on the left side of the fly) their initial value if they crossed the axis, which is what the second and fourth conditionals check for.

3.2 Median-filtering the data

Next, the joint angle time series was passed through a median filter of window length 3 time points. A median filter is a common data pre-processing technique whereby a moving window is run through a time series and at each point, the median of the set of points in the window is calculated. The median filter is a great way to reduce noise, or tracking errors from LEAP. This is especially important because even if the network makes substantial tracking errors 0.5% of the time, this still corresponds to $\sim 5 \times 10^5$ problematic frames. Median-filtering allows us to greatly reduce the frequency of these erroneous frames. Figure 3 below demonstrates some of the peaks (in blue) which get glossed over with a median filter.

With median-filtering, there is a trade-off between denoising the data, and getting a representative chunk of the data. The larger the kernel window, the more data is getting glossed over, which means there are less points to work with, but a smaller kernel size means more errors. Given that the RNN is hypothesized to build an internal representation of the underlying neural dynamics, it should also denoise the input time series. Since this is the case, a smaller kernel window can be used in hopes of finding a denoised output time series (which is confirmed later) but still helping alleviate tracking errors.

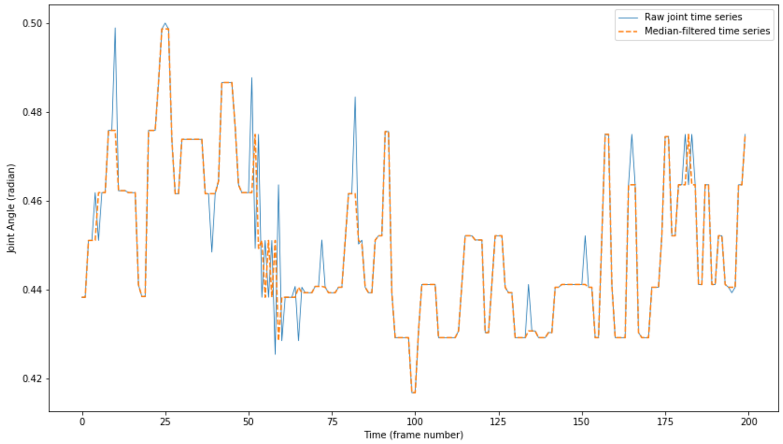

Figure 3: Median-filtering gets rid of some peaks, which corresponds to tracking errors

The median filter was performed with SciPy's signal Median Filter function.

3.3 Splitting the data into train-validation set

The data were concatenated into a training set and validation set. Because the network architecture is stateful (which will be defined later), the angle time series was shaped in such a way that the batch-size b and lookback l parameters were taken into account. These variables will be delved into later, but for now it is important to note that $b = 32$ and $l = 100$. The data were concatenated as diagrammed below, where $\{\alpha, \beta, ..., \omega\}$ are the set of angles measured; α corresponding to the first point, the tip of the head, β corresponding to second point, the left eye, and so forth.

The 32 individual time series of size $N_l \times n_a$ - where $n_a = 30$ is the number of joint angles and $N_l = N(\frac{l}{b})$; lookback and batch-size are RNN parameters and $N = 10^5$ - were shaped into a multidimensional array of size $N \times l \times n_a$ which was split into a final training and validation set, of sizes $0.8N \times l \times n_a$

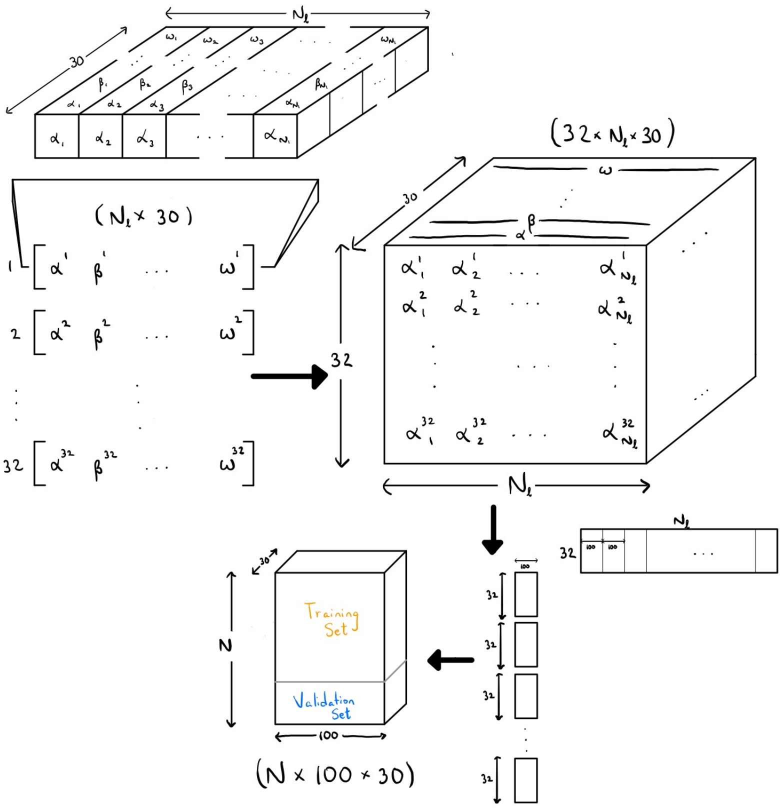
N_l
30
ω_1 ω_2 ω_3 . . . ω_{N_l}
β_1 β_2 β_3 . . . β_{N_l}
α_1 α_2 α_3 . . . α_{N_l}
$(N_l \times 30)$
1 $[\alpha^1 \; \beta^1 \; \dots \; \omega^1]$
2 $[\alpha^2 \; \beta^2 \; \dots \; \omega^2]$
32 $[\alpha^{32} \; \beta^{32} \; \dots \; \omega^{32}]$
$(32 \times N_l \times 30)$
ω
β
α
30
32
α^1_1 α^1_2 . . . $\alpha^1_{N_l}$
α^2_1 α^2_2 . . . $\alpha^2_{N_l}$
α^{32}_1 α^{32}_2 . . . $\alpha^{32}_{N_l}$
N_l
N_l
32
100
100
32
32
32
32
N
30
Training Set
Validation Set
100
$(N \times 100 \times 30)$

Figure 4: Concatenation of time series into training data set. $N_l = Nl$, where $l = 100$ is the *lookback* parameter for our RNN, a measure of how many points previous in the time series it looks to generate its future prediction

and $0.2N \times l \times n_a$, respectively. The data was shaped in a way such that every 32nd value corresponded to the same angle, or $\{0, 32, 64, ..., 99968\}$ correspond to the α angles, $\{1, 33, 65, ...99969\}$ correspond to the β angles, and so on. More concisely, the indices which correspond to α can be defined as $i_\alpha = \{n \mid n \mod 32 = 0\}$ and so on.

3.4 Standardizing the data

In order to train the network more efficiently, the joint angle time series was standardized. This was done by subtracting the mean and dividing by the standard deviation of every point for each individual time series, or z-scoring the joint angle series. In order to ensure no information from the validation set seeped into the model (and more for good practice), the mean and the standard deviation of the training set was applied to both the training and validation set.

3.5 Building a dynamical model: feeding the data into an RNN

3.5.1 RNN Hyperparameters

The hyperparameters listed here are not exhaustive, but they are representative of the tuning factors and if there were hyperparameters not mentioned here, they were kept the same across all models. The first parameter in the RNN is the batch-size. The batch-size is the number of samples which get fed into the RNN to estimate the error gradient before the model's weights get updated. I use a batch-size of 32, primarily because the RNN model built for the rat model used a batch-size of 32.

I use the Adam learning rate optimizer, which is the default optimizer for the learning rate. The Adam optimizer is an adaptive learning parameter, which means it has a running learning rate for each of the parameters. The Adam optimizer also modifies the learning rate according to the first and second moments of the gradient, where the moments are the same as in probability and statistics:

$$n\text{-th moment: } m_n = E[X^n]$$

In terms of network architecture, I use a Long Short-Term Memory (LSTM) network, a popular RNN used to capture dynamics of the input vector both on the long- and short-term scale. LSTM networks have an architecture similar to traditional feed-forward networks with two important specifications: (1) they contain feedback connections, such that the output of one cell is fed into the input of the next cell (there is nt necessarily a sequential order of cells, the output can be fed into many cells) and (2) they have the ability to "remember" and "forget" the previous state depending on a set of rules. Below is an diagram of an individual LSTM cell.

There are three key components of an LSTM cell: the "forget" gate, the "input" gate, and the "output" gate.

With regards to the network architecture, another hyperparameter is the number of layers in the RNN. I analyze both the 1-, 2-, and 3-layer RNN models, the motivation for which is further discussed in the discussion section.

3.5.2 1-layer RNN

Batch-size: 32 Number of LSTM cells/neurons in network: 128 Number of traning epochs: 500

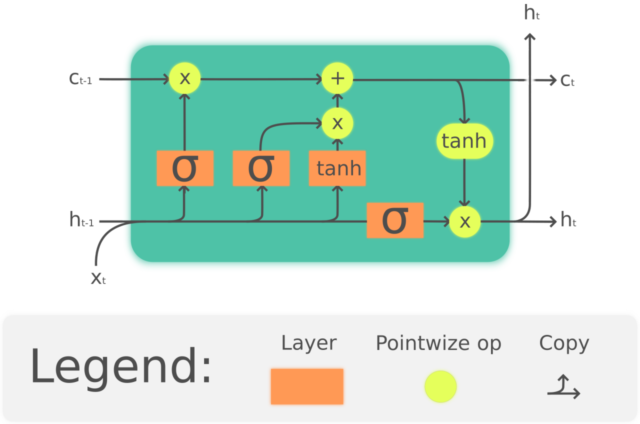

Figure 5: Single LSTM cell (figure courtesy of Guillaume Chevalier via Wikipedia, Creative Commons license)

3.5.3 2-layer RNN

3.5.4 3-layer RNN

3.6 Building a behavioral representation

3.6.1 PCA on hidden states of RNN

3.6.2 Wavelet transform of principle components

3.6.3 Creating a behavior map with tSNE

4 Results

4.1 1-layer RNN

4.2 2-layer RNN

4.3 3-layer RNN

4.4 LEAP is unable to track well on low quality data

5 Discussions

The behavior map for each of the maps, particularly the second layer bear striking resemblance to the behavior map generated by Berman et. Al in 2014. For the rat model, the behavior map generated from the postural data was different from the behavior map generated from the RNN hidden states. Because of this, we would have expected a similar difference to be noted in the fly behavior map. However, the maps created are nearly identical, both in terms of coarse visual analysis, and when looking into the specific regions and seeing what

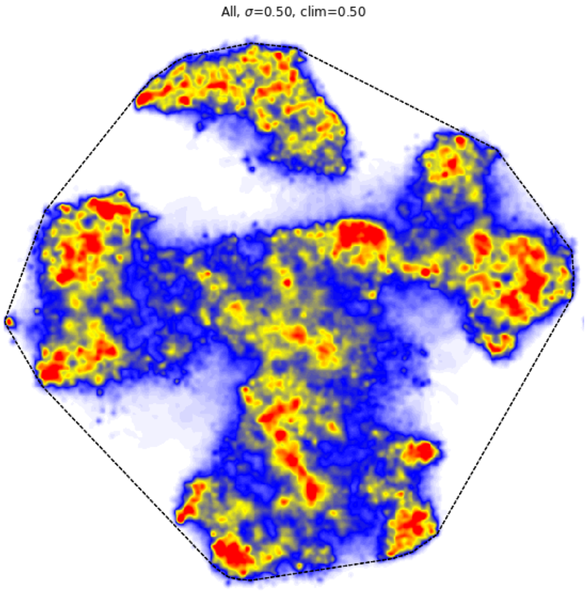

Figure 6: Behavior map for 1-layer RNN

the fly is doing in each region. This says something deep about the relationships in the fly data versus the rat data, something about the correlations...

6 Conclusions

References

[1] Thomas Baumgartner, Urs Fischbacher, Anja Feierabend, Kai Lutz, and Ernst Fehr. The neural circuitry of a broken promise. *Neuron*, 64(5):756–770, 2009.

[2] Barry Bentley, Robyn Branicky, Christopher L Barnes, Yee Lian Chew, Eviatar Yemini, Edward T Bullmore, Petra E Vértes, and William R Schafer. The multilayer connectome of caenorhabditis elegans. *PLoS computational biology*, 12(12):e1005283, 2016.

[3] Gordon J Berman. Measuring behavior across scales. *BMC biology*, 16(1):23, 2018.

[4] 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.

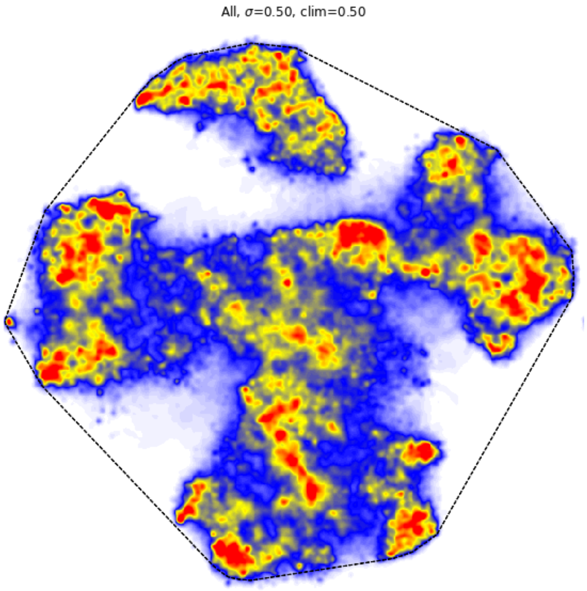

Figure 7: Behavior map for 1-layer RNN

[5] Onno D Broekmans, Jarlath B Rodgers, William S Ryu, and Greg J Stephens. Resolving coiled shapes reveals new reorientation behaviors in c. elegans. *Elife*, 5:e17227, 2016.

[6] 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.

[7] Matteo Carandini. From circuits to behavior: a bridge too far? *Nature neuroscience*, 15(4):507–509, 2012.

[8] BJ Frost. The optokinetic basis of head-bobbing in the pigeon. *Journal of Experimental Biology*, 74(1):187–195, 1978.

[9] David Grant Colburn Hildebrand, Marcelo Cicconet, Russel Miguel Torres, Woohyuk Choi, Tran Minh Quan, Jungmin Moon, Arthur Willis Wetzel, Andrew Scott Champion, Brett Jesse Graham, Owen Randlett, et al. Whole-brain serial-section electron microscopy in larval zebrafish. *Nature*, 545(7654):345–349, 2017.

[10] Rudolf Jander and Ursula Jander. Wing grooming in bees (apoidea) and the evolution of wing grooming in insects. *Journal of the Kansas Entomological Society*, pages 653–665, 1978.

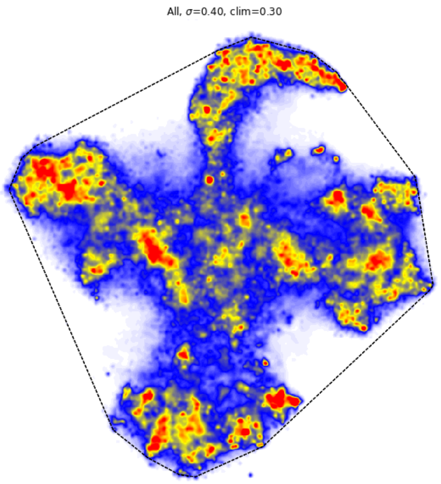

Figure 8: Behavior map for 1st layer of 2-layer RNN

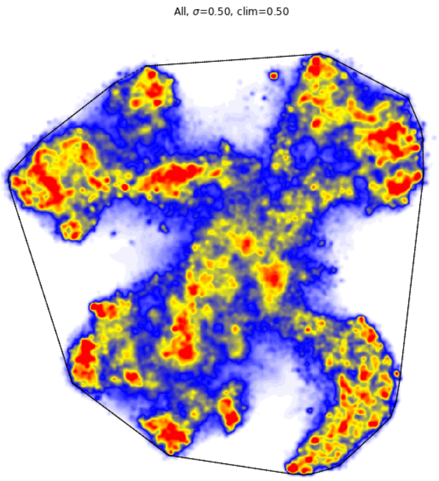

Figure 9: Behavior map for 2nd layer of 2-layer RNN

[11] Reza Kalhor, Kian Kalhor, Leo Mejia, Kathleen Leeper, Amanda Graveline, Prashant Mali, and George M Church. Developmental barcoding of whole mouse via homing crispr. *Science*, 361(6405):eaat9804, 2018.

[12] John W Krakauer, Asif A Ghazanfar, Alex Gomez-Marin, Malcolm A MacIver, and David Poeppel. Neuroscience needs behavior: correcting a reductionist bias. *Neuron*, 93(3):480–490, 2017.

[13] Talia N Lerner, Li Ye, and Karl Deisseroth. Communication in neural circuits: tools, opportunities, and challenges. *Cell*, 164(6):1136–1150, 2016.

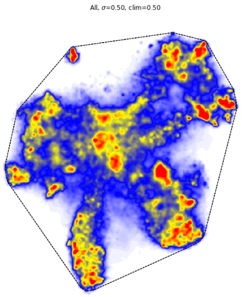

Figure 10: Behavior map for 1st layer of 3-layer RNN

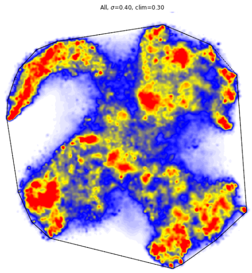

Figure 11: Behavior map for 2nd layer of 3-layer RNN

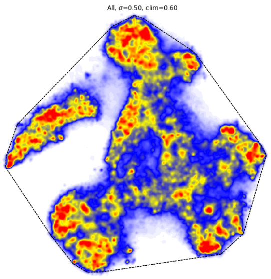

Figure 12: Behavior map for 3rd layer of 3-layer RNN

[14] Aaron McKenna, Gregory M Findlay, James A Gagnon, Marshall S Horwitz, Alexander F Schier, and Jay Shendure. Whole-organism lineage tracing by combinatorial and cumulative genome editing. *Science*, 353(6298):aaf7907, 2016.

[15] C Nicol. Understanding equine stereotypies. *Equine veterinary journal*, 31(S28):20–25, 1999.

[16] Seung Wook Oh, Julie A Harris, Lydia Ng, Brent Winslow, Nicholas Cain, Stefan Mihalas, Quanxin Wang, Chris Lau, Leonard Kuan, Alex M Henry, et al. A mesoscale connectome of the mouse brain. *Nature*, 508(7495):207–214, 2014.

[17] Jeremy A Pfaff, Liana Zanette, Scott A MacDougall-Shackleton, and Elizabeth A MacDougall-Shackleton. Song repertoire size varies with hvc volume and is indicative of male quality in song sparrows (melospiza melodia). *Proceedings of the Royal Society B: Biological Sciences*, 274(1621):2035–2040, 2007.

[18] Hans Slabbekoorn and Thomas B Smith. Bird song, ecology and speciation. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 357(1420):493–503, 2002.

[19] C Shan Xu, Michal Januszewski, Zhiyuan Lu, Shin-ya Takemura, Kenneth Hayworth, Gary Huang, Kazunori Shinomiya, Jeremy Maitin-Shepard, David Ackerman, Stuart Berg, et al. A connectome of the adult drosophila central brain. *BioRxiv*, 2020.

[20] Zhihao Zheng, J Scott Lauritzen, Eric Perlman, Camenzind G Robinson, Matthew Nichols, Daniel Milkie, Omar Torrens, John Price, Corey B Fisher, Nadiya Sharifi, et al. A complete electron microscopy volume of the brain of adult drosophila melanogaster. *Cell*, 174(3):730–743, 2018.