Distribution Agreement

Shinyoung Kang April 7, 2021

Mapping Behavioral and Neural Data with Variational Autoencoders

by

Shinyoung Kang

Gordon J. Berman
Adviser

Department of Quantitative Theory and Methods

Gordon J. Berman

Adviser

Robert Liu

Committee Member

Jeremy Jacobson

Committee Member

2021

Mapping Behavioral and Neural Data with Variational Autoencoders

By

Shinyoung Kang

Gordon J. Berman

Adviser

An abstract of
a thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Science with Honors

Department of Quantitative Theory and Methods

2021

Abstract

Mapping Behavioral and Neural Data with Variational Autoencoders
By Shinyoung Kang

Animal behavior exists across many time and length scales, likely requiring commensurate multiscale neural activity to control it. Although there have been improvements in our ability to measure neural circuitry and animal behaviors over long timescales, the biggest challenge in measuring behavior has been finding precise methods of selecting meaningful quantitative representation for an animal's behavior and fixing errors in the recordings. In this paper, we look at three-dimensional rodent's kinematic data obtained through 20 sensors attached to a rodent. Previously, many methods such as principal component analysis attempted to reduce a complex animal's postural data by selecting a select number of points representing the majority of the variance of the actions to create a meaningful representation. Instead of selecting a few points, we utilize variational autoencoders (VAEs), which not only reduce the dimension of the complex data but also can be used to generatively reconstruct the rodent's posture allowing us to fix tracking errors in the recordings. The VAE fixes tracking errors by training on non-error data, thereby reducing noise in the postural data, a clearer identification of the stereotyped behaviors and generation of behavioral maps of those behaviors emerges. We look at behavioral maps created with different dimensional reduction methods, predicting future behavioral transitions from one behavior to another. Lastly, we analyze simultaneous behavioral and neural recording to see if we can identify certain neurons that fire during particular stereotyped behaviors such as grooming. This allows prediction of the behavior of an animal using neural data and opens the possibilities for quantitatively linking brain and behavior.

Mapping Behavioral and Neural Data with Variational Autoencoders

By

Shinyoung Kang

Gordon J. Berman

Adviser

A thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Science with Honors

Department of Quantitative Theory and Methods

2021

Acknowledgements

I would like to thank Dr. Gordon J. Berman and Kanishk Jain for teaching necessary skills for this project and supporting me through the past years to complete this project. I would like to thank Dr. Gordon J. Berman again for helping me through the writing process.

Table of Contents

Mapping Behavioral and Neural Data with Variational Autoencoders

Shinyoung Kang[1]

[1]Emory University Department of Quantitative Sciences

Introduction:

There have been many improvements and attempts in trying to analyze animal behavior. New methods have been introduced for neural circuit recording and continuous appendicular and postural tracking using reflector embedding (CAPTURE) that allows data to be collected over long period of time (Marshall et al., 2020). However, the difficulties in improving our ability to analyze behavior in light of the neural activity driving it are more conceptual rather than technical. How do we quantify animal behavior? In other words, how can we put meaningful numbers that described the animal's movements and make predictions about the underlying neural dynamics.

Previously, studies have discussed computational techniques of measuring behavior (Dell et al., 2014, Egnor et al., 2016) to finding numbers of actions based on tasks (Jones et al., 2018) to simplifying large behavioral data (Stephens et al., 2011, Gomez-Marin et al., 2014) to studying dynamic movements of ground-based flies (Berman et al., 2014). Problems, however, have arisen as recording animal behaviors have wide range of errors for tracking body parts of the animal. Even with newer methods for tracking animal behavior, tracking error can occur from errors in specific animal's behavior when it has body parts together or magnification of certain sensors due to its position in the recording arena. Tracking errors can be largely defined as two types of errors: missing values for sensors and incorrect values for sensors. Missing values are easier to solve as they are defined as non-existing values from the start while it is difficult to find tracking errors occurring from magnification or diminution of sensors. The more noise in the data, the harder it is to analyze animal behavior due to misidentification of stereotyped behaviors.

We apply a novel version of an analysis of whole-body kinematic data that originally developed for images of flies to motion capture data of rats, relying on postural data collected using the CAPTURE method. What makes this approach novel is that we utilize variational autoencoders (VAEs) instead of principal component analysis (PCA). A VAE is a type of deep learning architecture consisting of an encoder and a decoder. They summarize data into a lower dimensional stochastic latent space using the encoder and reconstructing data by fitting it back to the original probability distribution using the decoder (Kingma et al., 2014, Rezende et al., 2014). These properties allow VAEs to provide a low-dimensional quantification for the animal's postural data. More details of our implementation will follow below.

With VAE's ability to reconstruct postural data from VAE's latent space trained from well-measured posture, our results show that we can fix majority of the tracking errors for the rodent's postures both for missing values, and magnification and diminution for values. Previously, missing values for sensors have been replaced with the median of adjacent values. This creates two major error in identifying animal behaviors. First, if they are not in the correct position creating a different posture than the animal was, in reality, behaving. Second, if the missing values for long period of time will create that specific sensor to be labeled as idle. As VAE predicts based on other positional values of sensors as well, VAE fixes missing positional values by predicting the actual posture by inputting median or interpolate filled missing values. VAE can also fix magnification and diminution errors by carefully choosing training data that we input the VAE. Through both VAE defined (later explained in the method) and manual selection, we train the VAE to regularize the encoder with non-scaled data so that it will fix scaled tracking errors.

By fixing errors and reducing dimension, VAE can aid identifying stereotyped behaviors, non-random and repeated multiple times, from the wide range of potential postural movements that an animal can possibly make by removing noises in the data (Davis et al., 1974, Lefebvre et al., 1982). There has been attempts to use VAE on animal motions for 2D postural images for flies (Graving et al., 2020); however, VAE on 3D postural data to reduce dimension has not been attempted and could provide a new analysis method for future works that deals with breaking down 3D kinematic data with meaningful representation for behaviors and fixing 3D kinematic recordings. Stereotyped behaviors can be viewed as an intricate system that is a combination of the structure of the organism, neural signaling, and metabolism. Therefore, interpreting these behavioral sequences in terms of biological mechanisms could potentially suggest how behaviors transition from one to another. This can be seen in the prevailing theory for behavioral analysis of time series: animal's behavioral motifs are hierarchical, going from one behavioral action to another in order (Dawkins et al., 1976, Simon et al., 1962).

Although there have been many theories for behavioral hierarchy, there are limitations as studies often use hierarchical clustering to identify hierarchical structures in data, which impose a hierarchical structure without providing quantitative validation that the representation is hierarchical in the first place. Another problem occurs as analysis of temporal organization of behaviors is limited to a single-time scale whereas certain states such as hunger can be ongoing for a long period of time (Heiligenberg et al., 1973, Miller et al., 2003). Precisely measuring these behaviors for long-time scales in an animal's natural setting through a quantifiable method could provide a way of studying them using a reproducible element.

Neural systems controlling body movements have been hypothesized to mirror the hierarchical structure of animal behavior with different body parts being controlled by different brain regions (Merel et al., 2019). Although task-based behavioral studies of individual body movements have been conducted, the movement of the entire body has not been properly studied (Svoboda and Li et al., 2018). With the CAPTURE method, measuring the reproducible 3D structure of animals could help explain behavior using postural and neural data. However, due to limitations in the CAPTURE method, there are certain points where the tracking of the sensor is missing or has errors such as when leg tracking.

In this thesis, we use 3D CAPTURE data of rodents, using VAE and t-Distributed stochastic neighbor embedding (t-SNE) to characterize temporal behavioral actions performed over a day along with fixing errors in tracking using VAE. Using these results, we find stereotyped behaviors in the behavioral map and apply the information bottleneck method to show hierarchical representation of rodent behavior with partitioning. Lastly, we look at neural recordings from electrodes in the striatum of the rodent to observe relationship between individual neurons and rodent's behavioral actions.

Results:

Continuous Appendicular and Postural Tracking using Reflector Embedding (CAPTURE)

We use data taken via CAPTURE method (Marhsall et al., 2020) which continuously tracks naturalistic rodent behavior as 3D whole-body kinematics. This method uses calibrated camera array that tracks the position of 20 retroreflective markers placed in the rodents through body piercings – which solves the problem of stable attachment over long period recordings. The rodent behaves within a two-foot-diameter plexiglass arena that is surrounded by a motion capture studio composed of motion capture cameras positioned around a two-foot-diameter plexiglass arena.

Variational Autoencoder Correction

While CAPTURE allows for detailed behavioral measurements, there still are difficulties due to occlusions. Approximately 21% frames have a missing tracked point or have tracking errors that are not physical given the potential configuration of a rodent as shown in Figure 1. Most of the errors were in the back and arm/leg tracking. There also were issues with tracking where the lengths of the body parts were magnified (e.g., the leg was magnified about 2 to 3 times the original length), which are not usually included in the 21%. These issues were most likely caused by problems with angles. Using VAE, we aim to predict the missing values for sensors along with identifying and fixing tracking error.

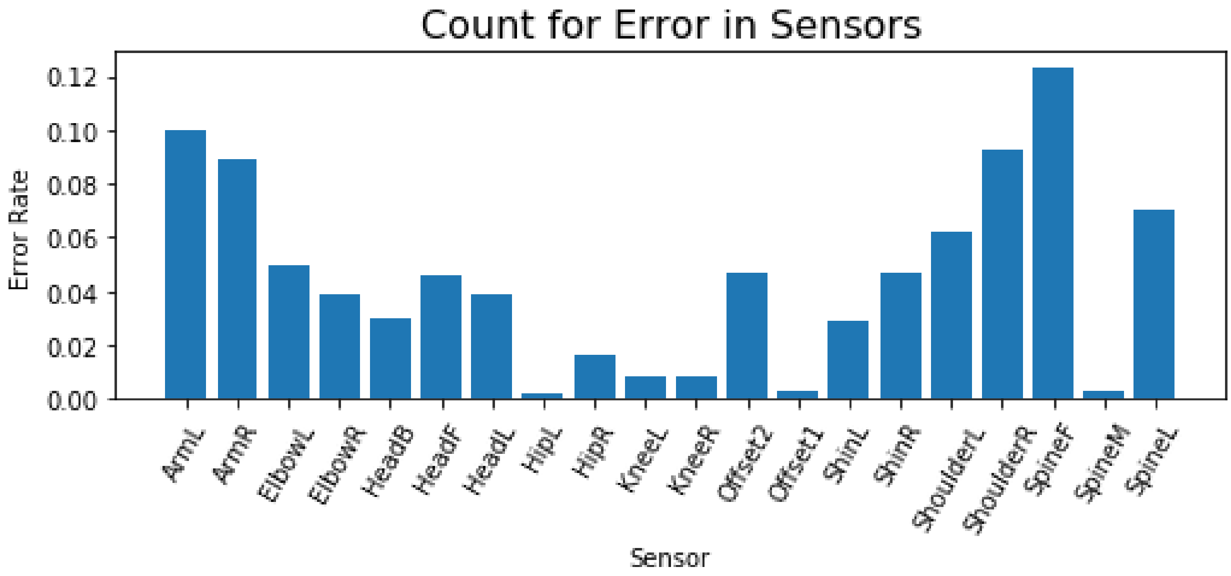

Figure 1: Error Rate of Sensors on the rodent's original postural data

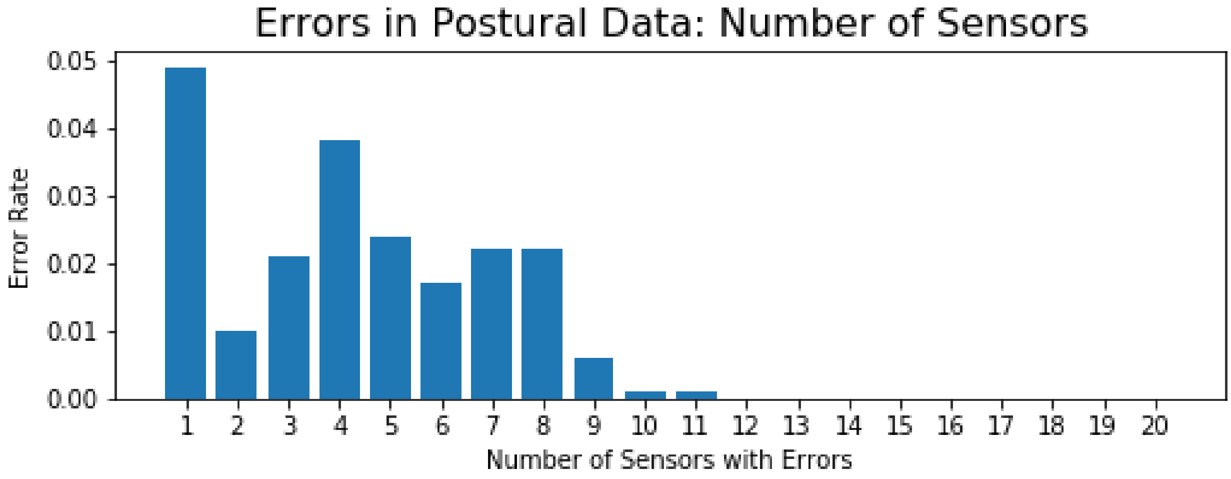

Figure 2: Number of sensors with errors for tracking errors in the rodent's original postural data

Variational autoencoders are generative models that consist of an encoder, a decoder, and a loss function (Kingma et al., 2013, Rezende et al., 2013). The encoder is a neural network that bottlenecks data x into a latent hidden representation space z, that is smaller than the original dimension of the data $x, q_\theta(z|x)$ figure 3. This distribution is a Gaussian probability density function with a mean $\vec{\mu}$ and covariance matrix C. The decoder is another neural network where the input is representation space z as a probability distribution of the data with weights and biases φ, $p_\varphi(x|z)$. Finally, the loss function is a negative log-likelihood with a regularizer.

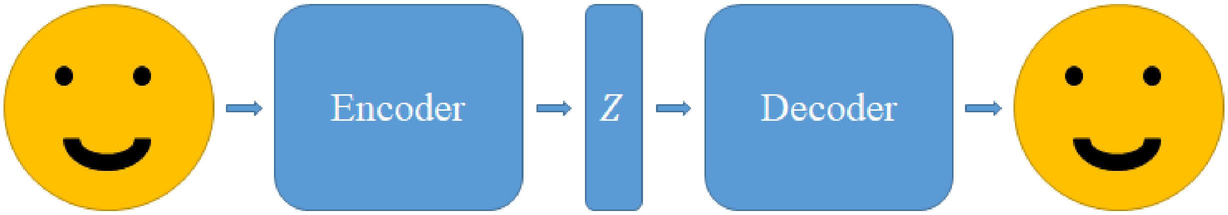

Figure 3: A cartoon of how variational autoencoders work

The motino capture systems identifies frames where a given marker cannot be reliably tracked. Then we transform the data by defining one of the sensors as the origin and re-centering by rotating the back sensor to be always pointing to the right. For training the variational autoencoder, we first train on the data with no major errors by optimizing the hyperparameters to minimize the training loss with Adam (Kingma et al., 2017). Using the variational autoencoder we trained, we exclude high-error data that are usually caused by tracking errors. We specifically remove data with,

$$|[x_{vae}, y_{vae},\ z_{vae}] - [x, y, z]| > 20\ pixels, \tag{1}$$

at least one sensor with a difference above 20 pixels ($\approx 1.3\ cm$). Then we re-train the variational autoencoder with the filtered data with the same steps as the previous model. The final model has an average training loss of 4.24 pixels per sensor ($\approx 2.8\ mm$).

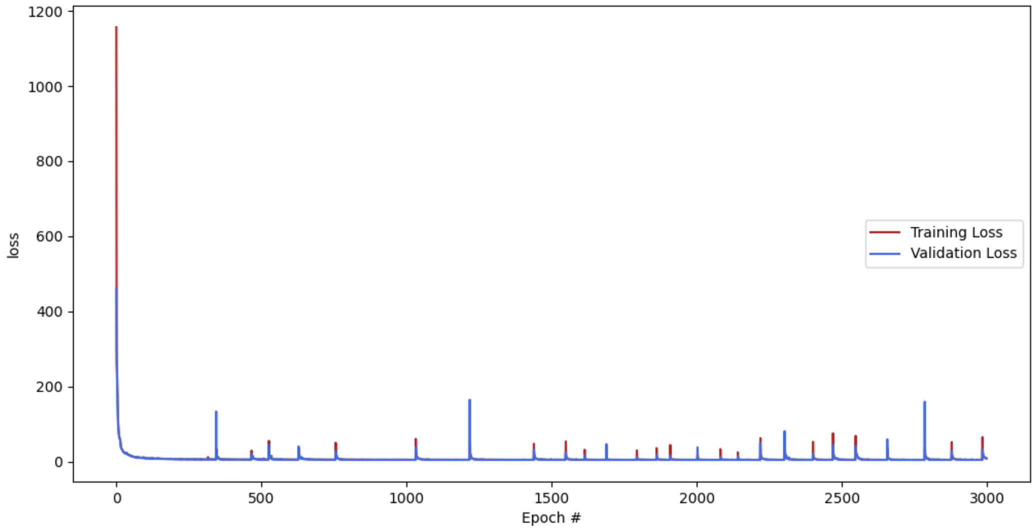

Figure 4: Training Loss for Variational Autoencoder

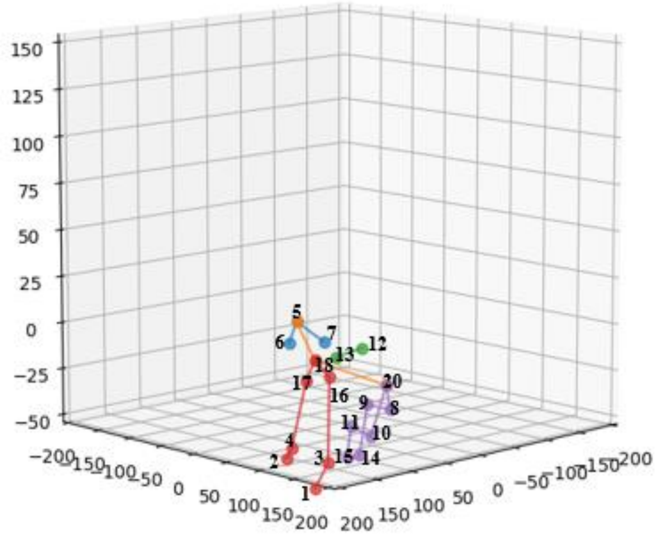

Figure 5: Labeling of Sensors on the rodent's posture

There were also other problems such as changes in certain sensors (e.g., changes in leg and ShinR and KneeR sensors) shown in Figure 6. Results show that VAE successfully fixes changes in position for sensors along with correctly reconstructing postural positions without errors(figure 6-7). Using Equation 1, VAE defines 5009825 errors out of the total 19980000 time series which is about 25.1% (figure 8).

Comparing VAE defined errors to missing value errors, we notice a higher rate of VAE defined errors due to some of the sensors being labeled as off-track by VAE. After VAE fixes the rodent's posture, VAE defines there is still 1490100 about 7.5% error. This suggests VAE fixes about 70.3% of the total VAE defined errors, compared per sensor in figure 9. There were certain configurations that VAE didn't perform well. These configurations are usually composed of missing the center sensor values or time series with tracking errors of 6 or more sensors, 25% of the total missing position errors as seen in figure 2. From the results, VAE has the hardest time predicting the positional values for the head. We expect this is due to the head values being less related to other body parts compared to other sensors.

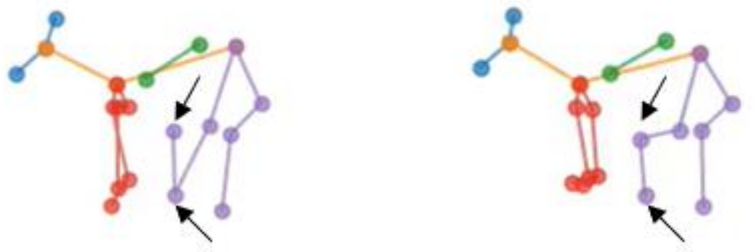

Figure 6: Error for changes in sensors. This example shows specifically the changes in sensor values for knee and shin of the rodent

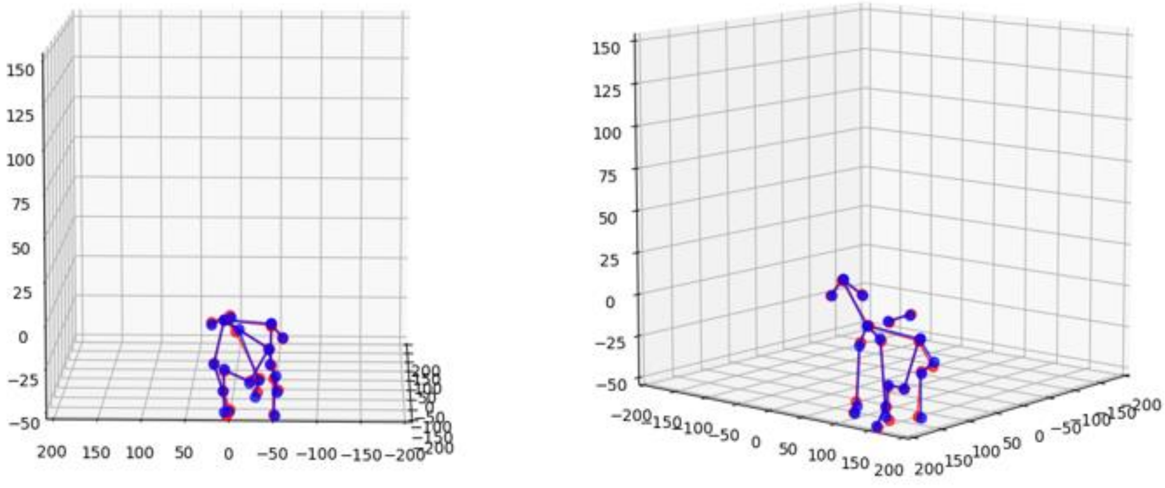

Figure 7: VAE prediction for rodent's posture without errors. Blue shows the original graph of the rodent while the red shows the VAE predicted rodent's posture

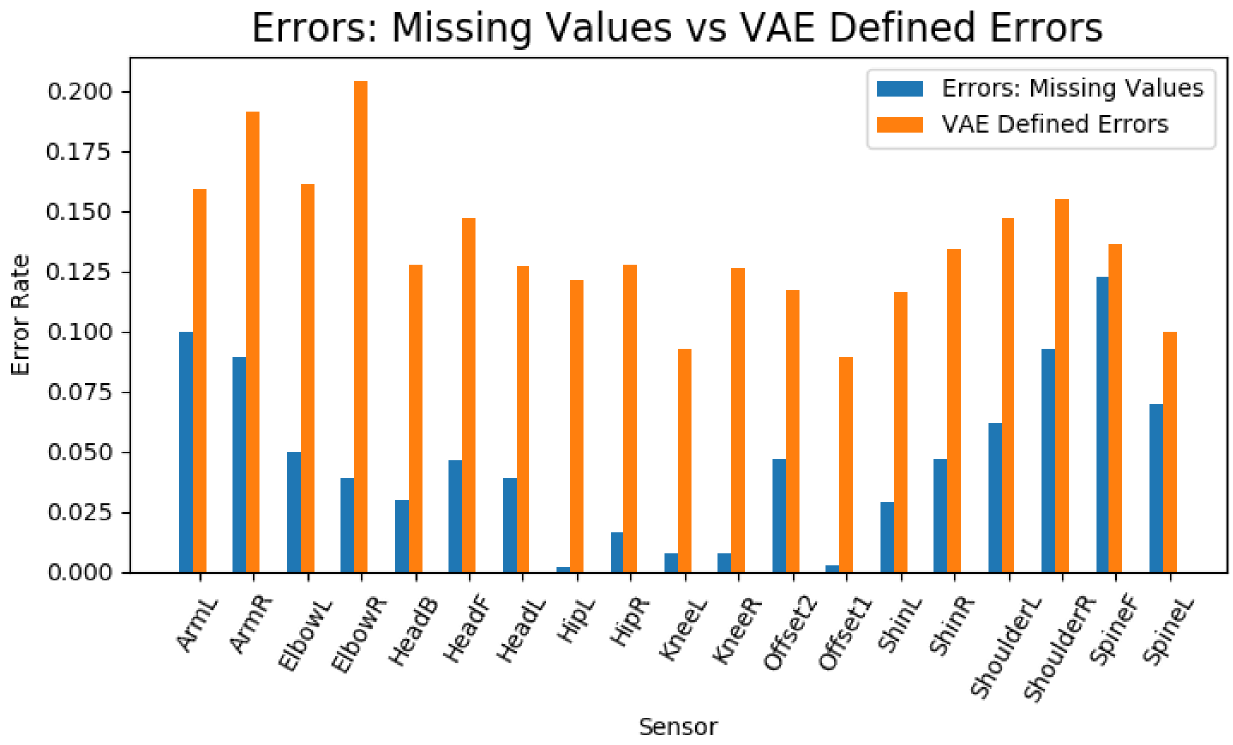

Figure 8: Number of Errors defined by missing values versus VAE. VAE tracks scaled body parts caused by tracking errors. We still see the highest error rate in the Arm position.

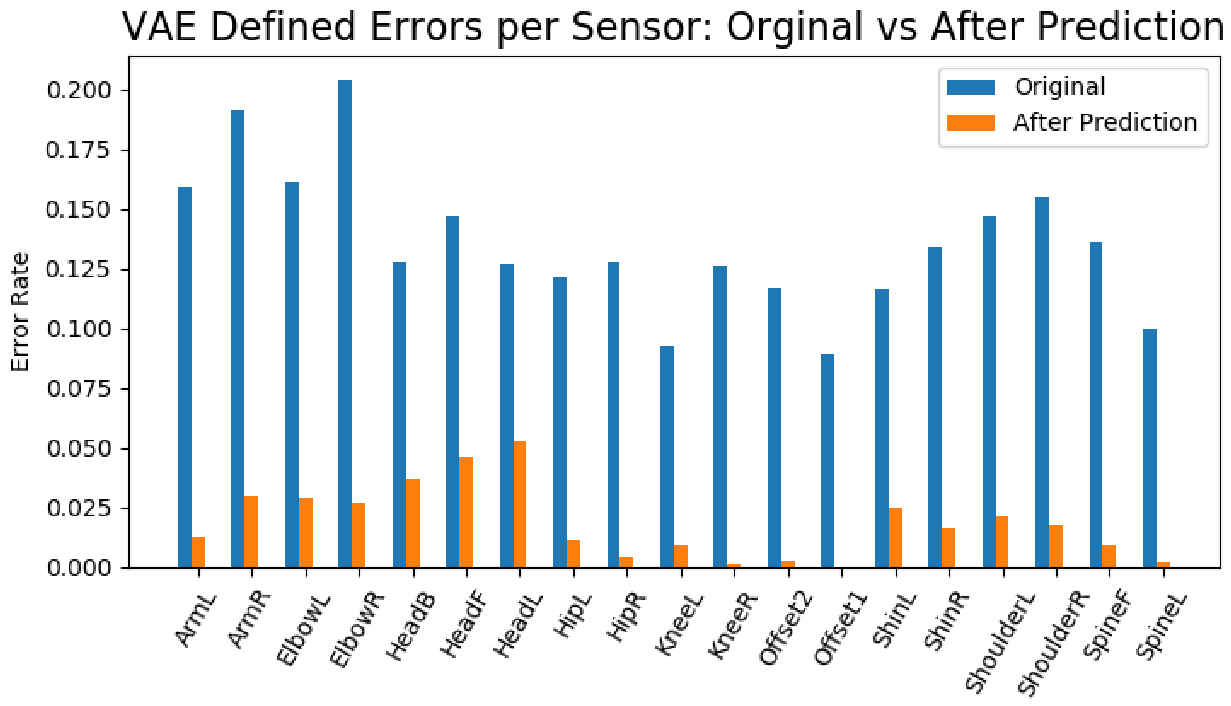

Figure 9: Changes in error rate defined by VAE after the prediction. We notice a significant decrease in the arm, hip, and knee sensors. We also notice that VAE has the hardest time correcting the head positions for the rodent.

Behavioral Map Comparison

By quantifying animal posture in terms of smaller dimension that represents the whole-body kinematics, we can find animals behaving in stereotyped actions as similar latent values will be representing similar animal posture. We, thereby, generate a behavioral map with reduced dimensional values of the postural data to find and map similar behavioral motions of the rodent to find stereotyped behaviors of animals. We view behavior as a trajectory through a high-dimensional space of postural dynamics (Berman et al., 2014). In this map, each epoch contains discrete behaviors where trajectory exhibits pause near repeatable stereotyped behaviors. Similar rodent’s stereotyped behaviors will be clustered in the map.

The general framework for the behavioral analysis is shown in Figure 10. To compare previous PCA method and VAE method, we reduce the three-dimensional postural data of the rodent using either PCA or VAE. Then, they are wavelet transformed to create spatio-temporal representation of the rodent's actions. As behaviors are dynamical, we look at wavelets with multi-resolution time-frequency trade-off using Morlet Wavelet Transform. This is ultimately to be embedded into two dimensions using t-distributed stochastic neighbor embedding (t-SNE) to minimize local distortions to construct a space with trajectories pausing near repeatable position when a stereotyped behavior occurs (Maaten et al., 2008). Finally, we calculate the probability distribution for the embedded two-dimensional space, using density function to locate peaks in the distribution. We observe discrete behavioral actions in the sustained pauses for the peaks such as idle state.

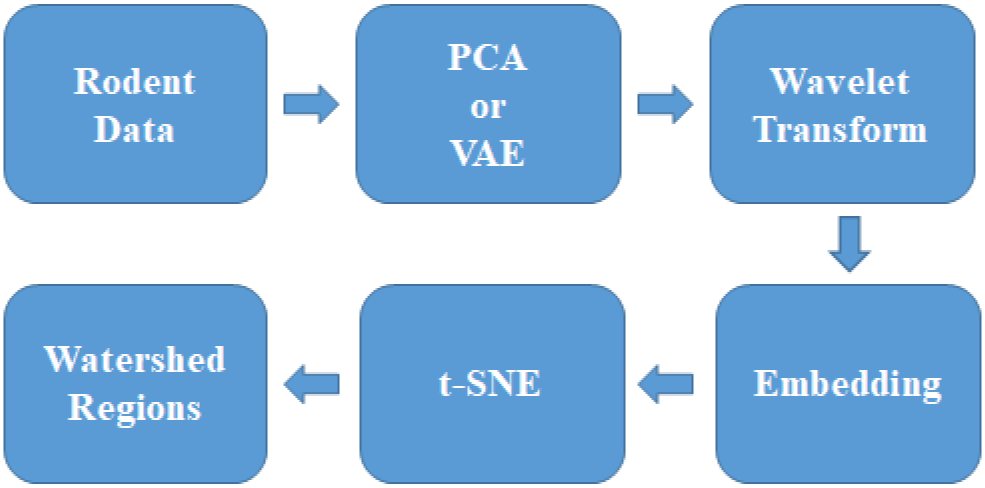

Figure 10: General pipeline for behavioral analysis. We use either the interpolated rodent data or the VAE predicted rodent data to start. Then we either use PCA or VAE to reduce data and then use wavelet transform, embedding, and t-SNE to create a non-linear embedding of the data. Then we use spatial segmentation to create watershed regions.

PCA t-SNE

First, we use a previously performed method that was used to map the stereotyped behavior of flies for comparison of the behavioral map with the VAE method we use (Berman et al., 2014). This involves PCA, which transforms correlated variables into a linearly uncorrelated eigenmodes and then selecting the largest eigenvalues from the total latent dimensions. Using PCA, we select 8 sensors, which explain about 95% of the variation as shown in Figure 11.

Next, we use a Morlet continuous wavelet transform for each of the latent values (Goupillaud et al., 1984). The Morlet wavelet transform is used to isolate short lengths of periodic motion, as it measures amplitudes of the transform occurring at continuous time scales (Daubechies et al., 1992). We use 25 frequency channels dyadically spaced from 1 and $150\ Hz$ with $150\ Hz$ being the Nyquist frequency for the measurements.

Finally, we use t-SNE, a dimensionality reduction algorithm that embeds high-dimensional data into lower dimensional space in a manner that preserves local clustering structure. This enables stereotyped behaviors to be close to one another, pausing near repeatable motions. t-SNE minimizes local distortions but does not preserve longer length behaviors, which is not needed (Maaten et al., 2008).

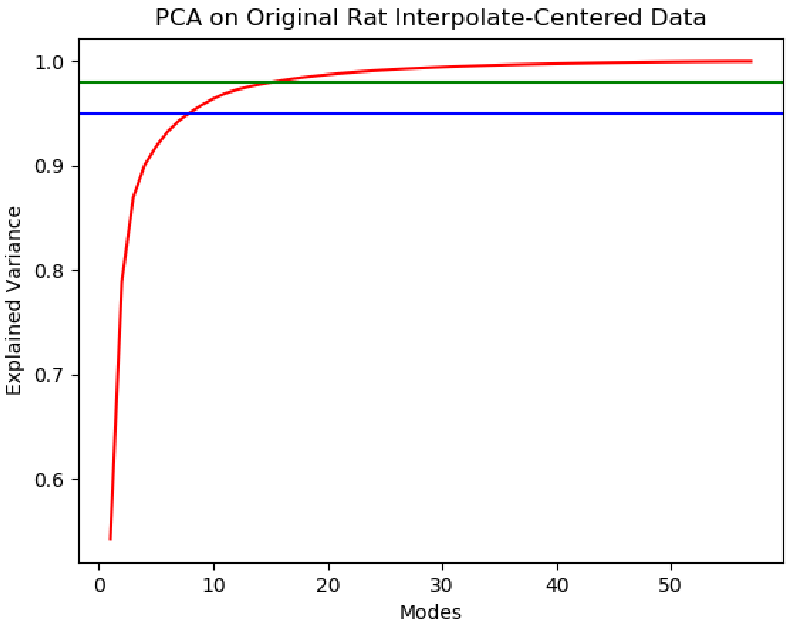

Figure 11: PCA of the interpolate filled Rodent data. 8 sensors explain about 95% of the entire variance of the data.

After the spectral feature vectors are embedded into two-dimensions, we convolve the embedded points with a small Gaussian ($\sigma = 0.6$) to generate a probability density function, as shown in figure 13. This probability density function has multiple local maxima that represent stereotyped behaviors of rodents. Differentiating $z_1(t)$ and $z_2(t)$ in time, we also observe a 'resting-moving' pattern of dynamics that are shown quantitatively by graphing the distribution of speeds within the embedded space (figure 13). The distribution of low-speed and high-speed points are shown in Figure 14.

Using a watershed transform (Meyer et al., 1994) to delineate these peaks and their surrounding regions, we found a total of 144 regions from the embedded space. At peaks on the probability density behavioral map, we find that the epoch shows the rodent's stereotyped behavior ranging up to 15 seconds.

VAE Transformed Data PCA t-SNE

Next, we create a behavioral map doing the same process as PCA t-SNE except that we use the variational autoencoder outputs instead of the original rat data.

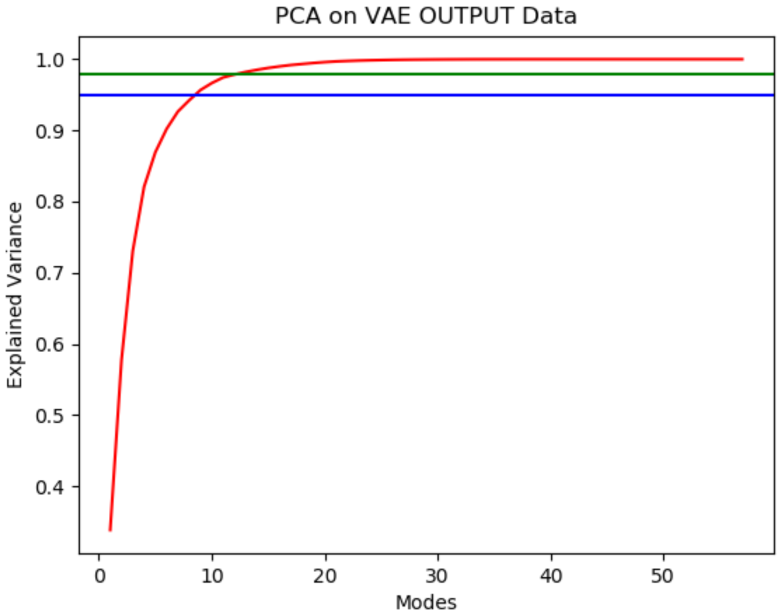

Figure 12: PCA of the VAE predicted rodent data. 9 sensors explain about 95% of the entire variance of the data.

The PCA results show that 9 sensors will explain about 95% of the variation (Figure 12). The graph of the velocities in the embedded spaces in two-component log-Gaussian mixture model shows a rest probability of 33%.

We find 131 regions from the watershed transform with peaks representing possible stereotyped behaviors (figure 13).

VAE t-SNE

Lastly, we use VAE to reduce dimension instead of using PCA at the start of the analysis. Using the reduced dimension latent space time series from the VAE, we perform Morlet wavelet transform and t-SNE to create a map.

The distribution of low versus high movement speeds in the embedded space is shown in Figure 14 with a rest probability of 33%. There is a total of 142 regions from the watershed transform (Figure 13).

Observing the behaviors for each of the pauses specific to the region, we find distinct similar behaviors for regions that are close by (electronic supplementary material, movies a – b). Many of the actions performed by the rodent are behaviorally defined by common definition such as walking, posterior movements, rearing, and grooming.

We find that performing PCA t-SNE on VAE Transformed Data results in a similar behavioral map as the VAE t-SNE while the previous method of using PCA t-SNE results in a different kind of embedded behavioral map. We also find interesting results as the rodent’s behavioral representation map was alike the map of the fly (Berman et al., 2014).

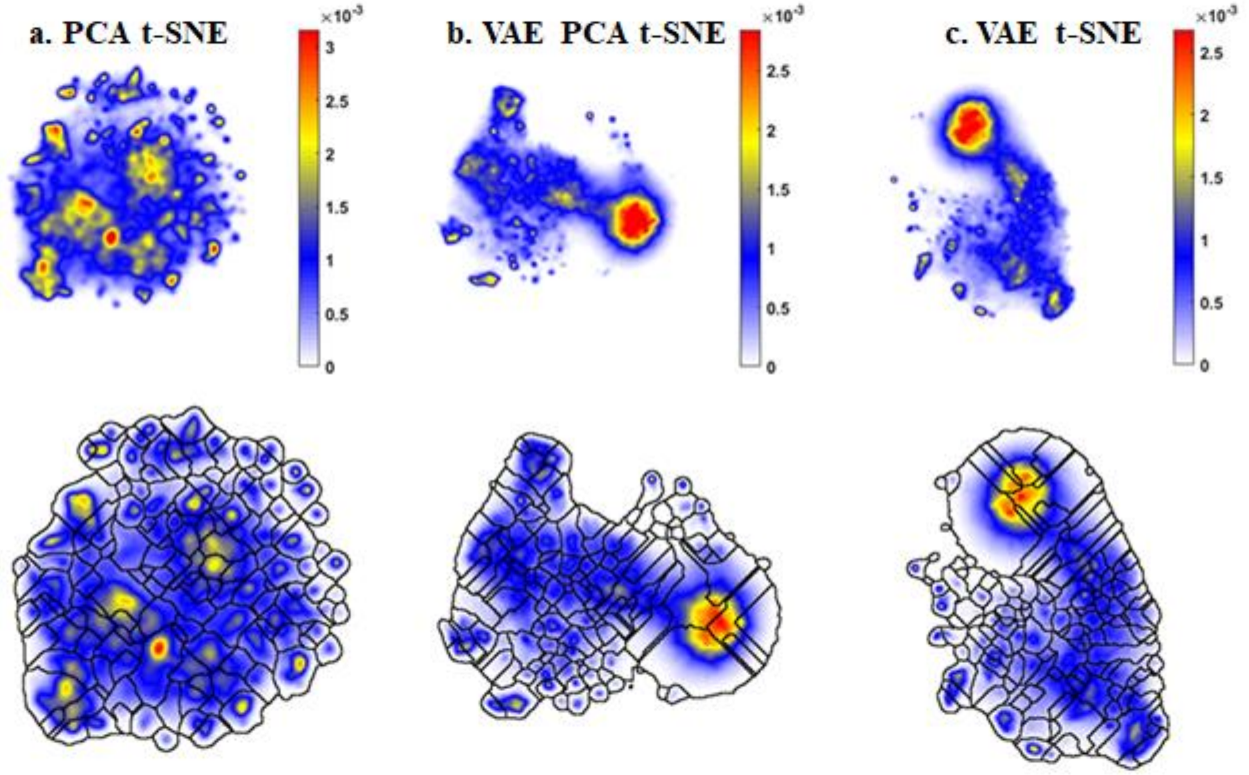

Figure 13: Behavioral maps of each of the behavioral methods.
a. The first behavioral map is created using PCA reduction and t-SNE.
b. The second behavioral map is created using the VAE predicted rodent data. We perform PCA on the predicted data then use the same methods as the first behavioral map to generate this figure.
c. The final behavioral map is using VAE to reduce dimension of the original data instead of the traditional way of using PCA. Same process was followed afterwards.

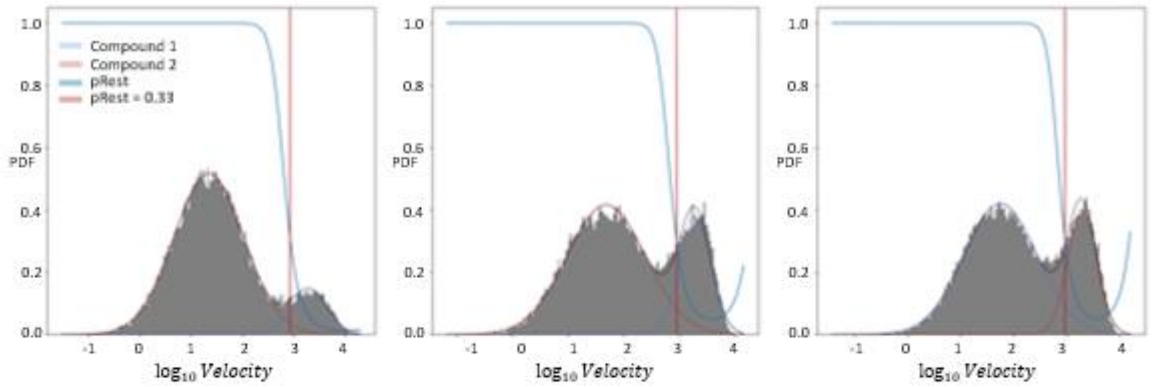

Figure 14: Histogram of velocities in the embedded space fitted into two-component log-Gaussian mixture model for each of the behavioral methods. All three methods show probability of rest of 33%.

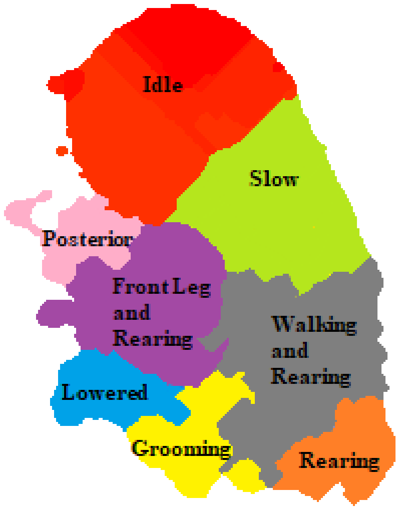

Figure 15: Watershed Regions Labeled from observing the behavioral actions of the rodents corresponding to the region numbers of the VAE t-SNE map. We find 8 major regions of idle, slow, walking and rearing, rearing, grooming, leg locomotion of lowered position, front leg locomotion and rearing, and posterior leg locomotion

Transitional Matrix Analysis

Each time series is divided into either stereotyped movement with a region number or non-stereotyped movement. We eliminate non-stereotyped actions and remove consecutive duplicates in order to look only at behavioral transitions instead of time-to-time transition. To analyze temporal pattern of behaviors, we calculate a transitional matrix over transitional states, which describes the probability from state j to state i after τ number of transition states.

$$[T(\tau)]_{ij} = p(S(n+\tau) = j \mid S(n) = i) \tag{2}$$

As τ increases, we expect the matrix to be less structured as this indicates the probability of predicting a future state based on the current state after a longer period.

Matrix T with $\tau = 1$ shows the probability of going from one transition state to the next transition state. We generate a graph for a Markov transitional matrix for moving from one region to the next, based on region numbers and another matrix for stereotyped behaviors grouped with one another (Figure 16). Results show that there is a high probability of going from one behavior to a similar behavior.

We find that in a transitional matrix plot of $\tau = 1$, behavioral transition is more likely to move to similar behaviors such as leg locomotion proceeding to another leg locomotion, which is consistent with previous studies of transitions occurring among similar actions (Takahata et al., 1981, Bialek et al., 2001).

Then we plot transitional matrix for $\tau = 10$ and $\tau = 100$. We observe that as the number of transitions increases, the probability of going from one region to another is not generally based on similar behaviors other than slow and idle motions (regions 130 to 140).

We then change the order of region numbers to the ones matched in behavioral space (Figure 15), putting regions with similar behaviors together (Figure $16d$). Compared to Figure $16a$, we find more square-based similarities as expected, indicating that behaviors are likely to transition from one action to a similar behavior.

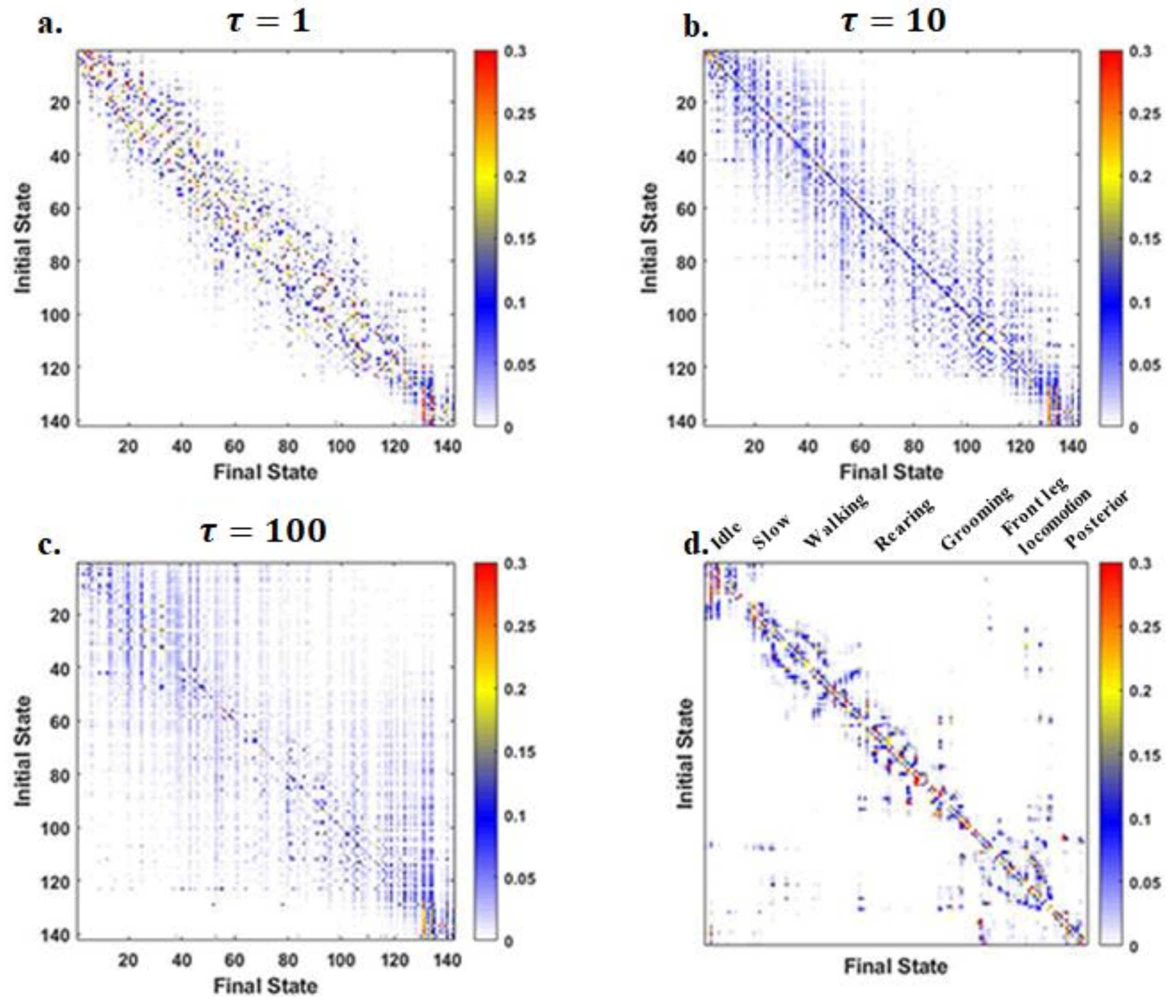

Figure 16: Markov Transitional Matrix
a. Transitional Matrix at $\tau = 1$, next transition state. The transition from one state to another is fairly similar by its region numbers as region numbers tend to be close to each other in the behavioral map.
b. Transitional Matrix at $\tau = 10$, next transition state. Compared to a, we observe more correlated region to region probability.
c. Transitional Matrix at $\tau = 100$, next transition state. Compared to a and b, we observe that this really explains nothing as all the probabilities are spread out.
d. Transitional Matrix at $\tau = 1$, next transition state. This time, the matrix is grouped by similar behaviors. We see that the transitional matrix shows more of a square like structure compared to the matrix in a. This shows the grouped behaviors are more similar to each other.

Predictability and Hierarchy

With the increase in transition numbers (Figure 18), we find that we need to be able to observe behavioral states for longer time scales. We thus aim to find clusters of groups to preserve the current behavioral state in predicting future actions. We expect to find a hierarchy of clusters as the number of clusters increase instead of clusters being created from two different clusters.

To accomplish this, we maximize the information of the future $I(Z; S(n+\tau))$ as we hold fixed information in the past $I(Z; S(n))$ in terms of groups $S(n) \to Z$. We use a Lagrange multiplier to keep past information fixed while maximizing F (Tishby et al., 1999).

$$F = I\big(Z; S(n+\tau)\big) - \beta I\big(Z; S(n)\big) \tag{3}$$

When β is 0, the equation conserves the complete complexity of all the behavioral states. Maximizing F as we increase β, we compact the description which explains why we lose predictive power. Figure 16 shows trade-off curves for different lags. As the number of lags increase, we see a decrease in the predictive power in the behavioral states. We observe relatively larger differences for small τ and smaller differences in bigger τ, indicating long time scales in behaviors exist.

Similarly, the partition plot shows longer time-scale actions still spatially cluster between similar behaviors. As the number of clusters increase, for example, in Figure 15 from 1 to 9 for $\tau = 100$, we see the clusters separate hierarchically instead of new complete clusters being created or emerging from existing multiple clusters.

We then plot transitional probabilities on the behavioral map. The red dots represent the local maxima of the local regions of the behavioral map. The line thickness is proportional to the transition rates from one region to another at $\tau = 1$ times the probability of starting at the initial region (Figure 17).

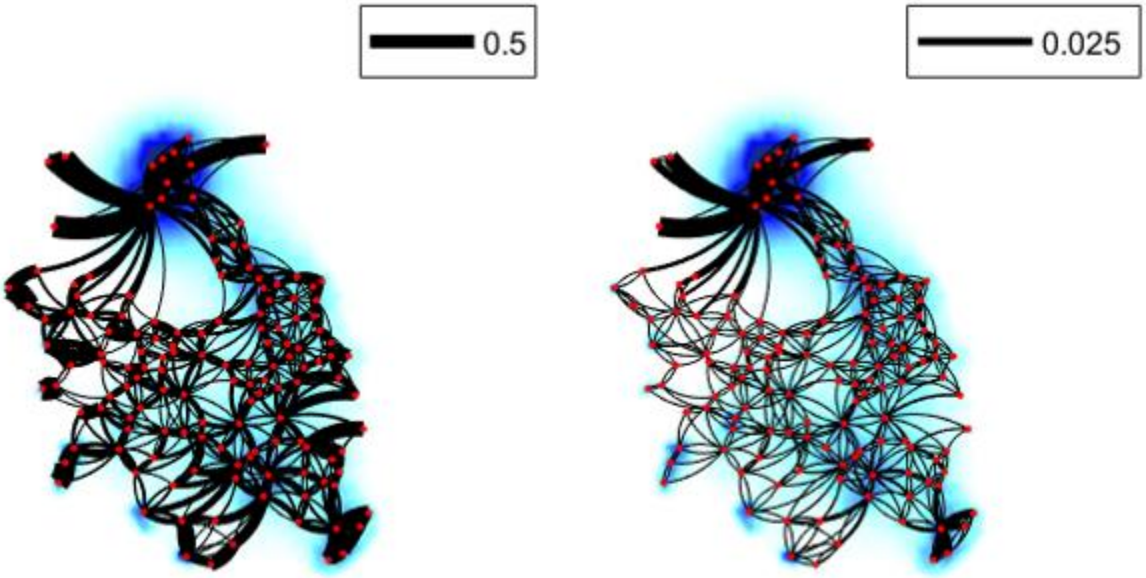

***Figure 17:** a. Transition rates plotted on behavioral map b. Flux plotted on behavioral map. The line thickness is proportional to the transitional rate for a and flux for b. The transitional rate is the probability of going from initial region to another region and the flux is probability of region occurring times transitional rate. In figure 17b. the wide lines for some of the points have been narrower due to having small possibility of rodent behaving in that region.*

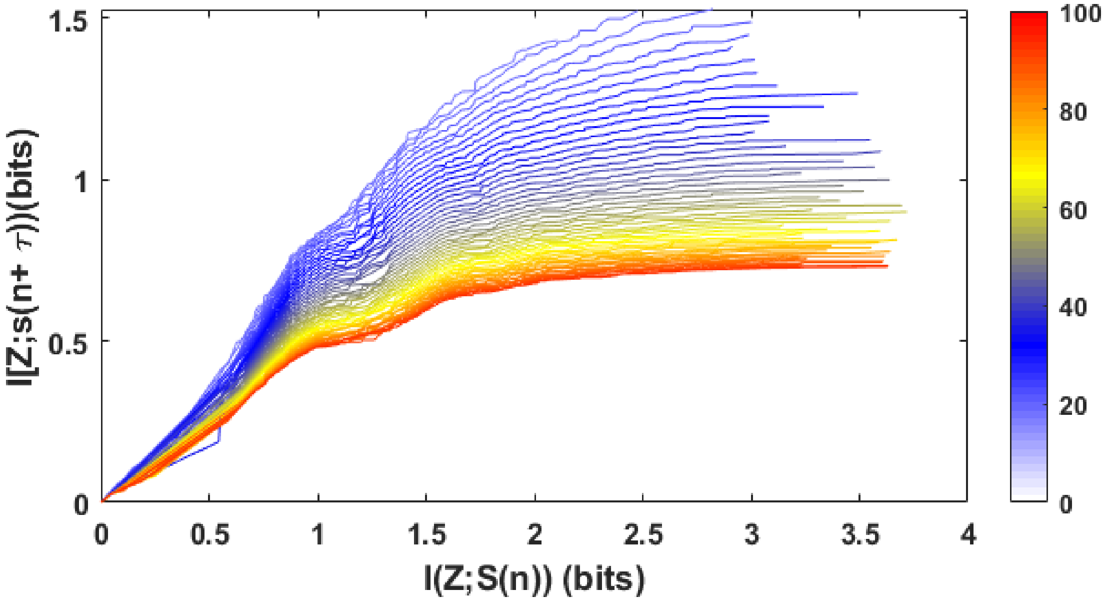

***Figure 18:** Optimal trade-off curves for lags $\tau = 1$ to $\tau = 100$. For each transitional state change τ, we plot the complexity of partitioning.*

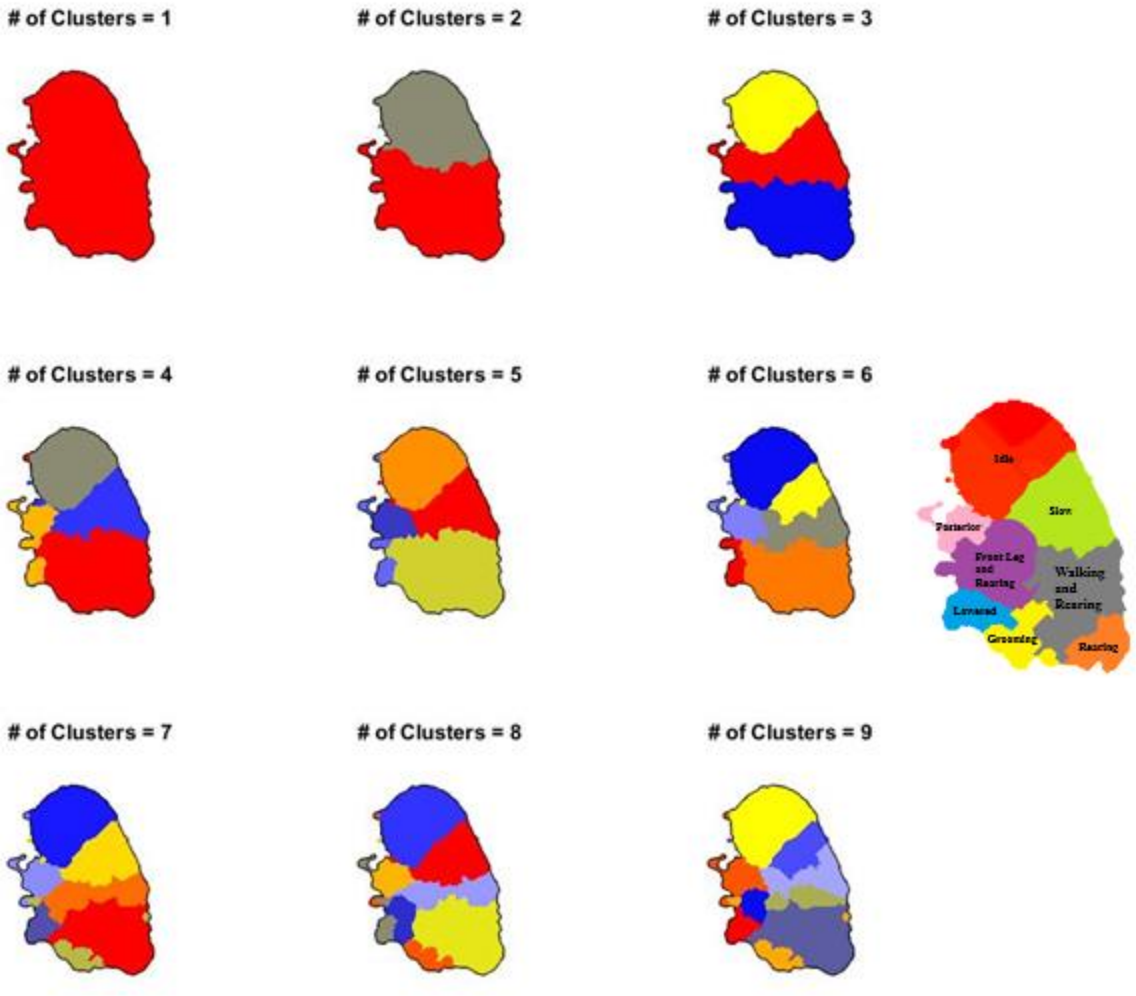

Figure 19: Informational bottleneck partitioning at behavioral space at transition $\tau = 100$. We show up to 9 clusters to help how the clusters are mostly splitting hierarchically instead of combinations of previous clusters. The right map is the watershed VAE t-SNE map of figure 15.

Neural Analysis

Neural data was measured using electrodes implanted into the dorsal striatum of the rodent. There was a total of 74 units that fired during the experiment (spike sorting done by Jesse Marshall, Ph.D., Ölveczky Lab). We first observe when each of the neurons spike and compute a spike-triggered averages in the density map to find which specific stereotyped behaviors (e.g., grooming area) are associated with the unit’s activity (Figure 15).

Observing the probability density map for Neurons 5 and 61 (figure 20), we observe that these neurons spike mostly when front leg rubbing and grooming occurs. Neuron 56 (figure 26) spikes when the rat is in the idle state. Neurons 2 and 16 (figure 22) spike when the rodent is in the lowered position performing leg locomotion and grooming. Neuron 12 and 19 (figure 21) spike when the rodent is rubbing its front leg. Neurons 4, 9 ,14, 30, 38, and 47 (figure 25) fire when the rodent is the majority of the time performing posterior motion with its leg. Neurons 53 and 73 (figure 23) fire most of the time during the rodent's rearing position. Neurons 34 and 74 fire when the rodents perform lowered position leg locomotion (figure 24).

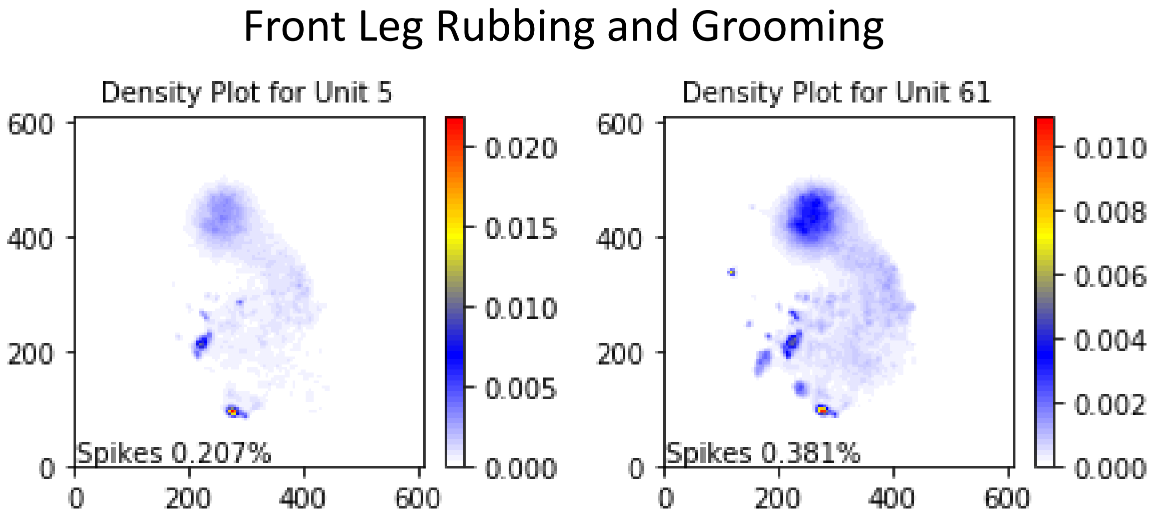

Figure 20: Density Plot for units that fire at front leg rubbing and grooming regions

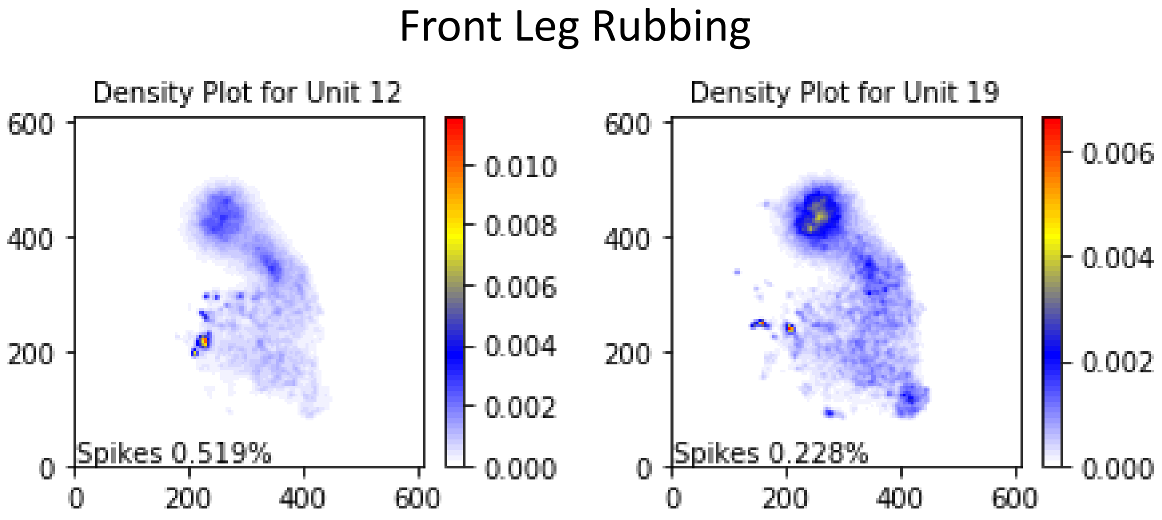

Figure 21: Density Plot for units that fire at front leg rubbing regions

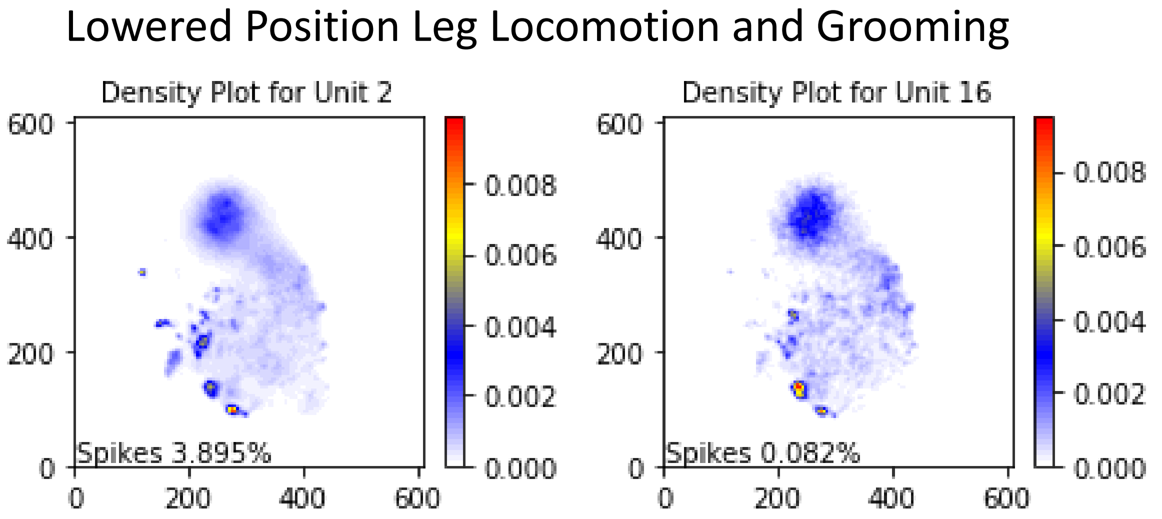

Figure 22: Density Plot for units that fire at lowered position leg locomotion and grooming regions

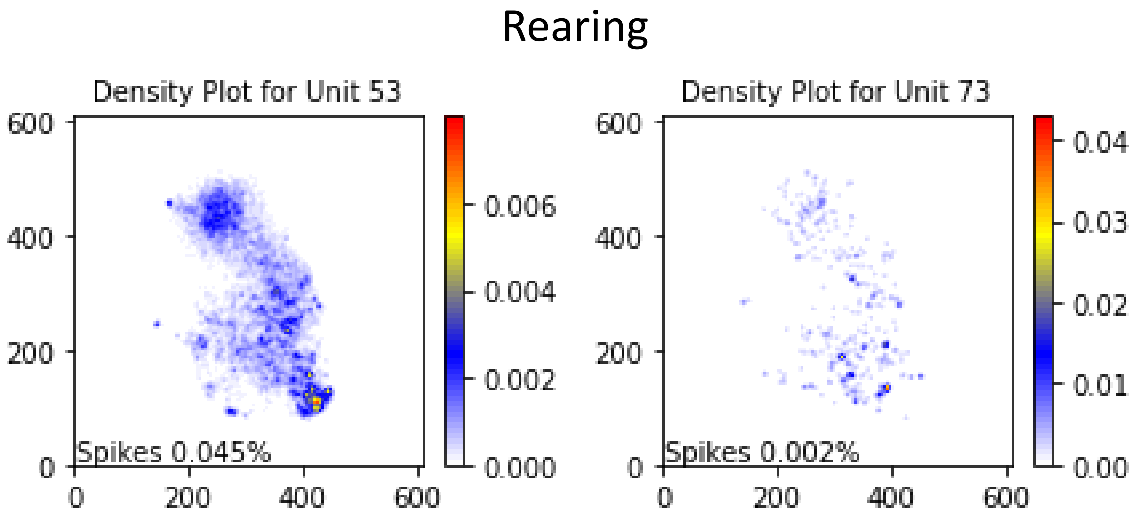

Figure 23: Density Plot for units that fire at rearing regions

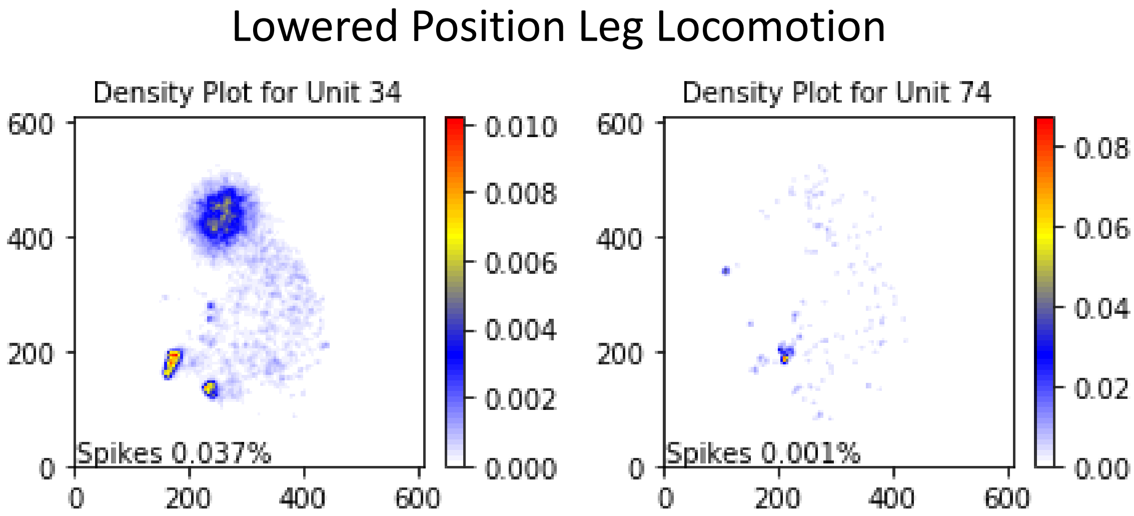

Figure 24: Density Plot for units that fire at lowered position leg locomotion regions

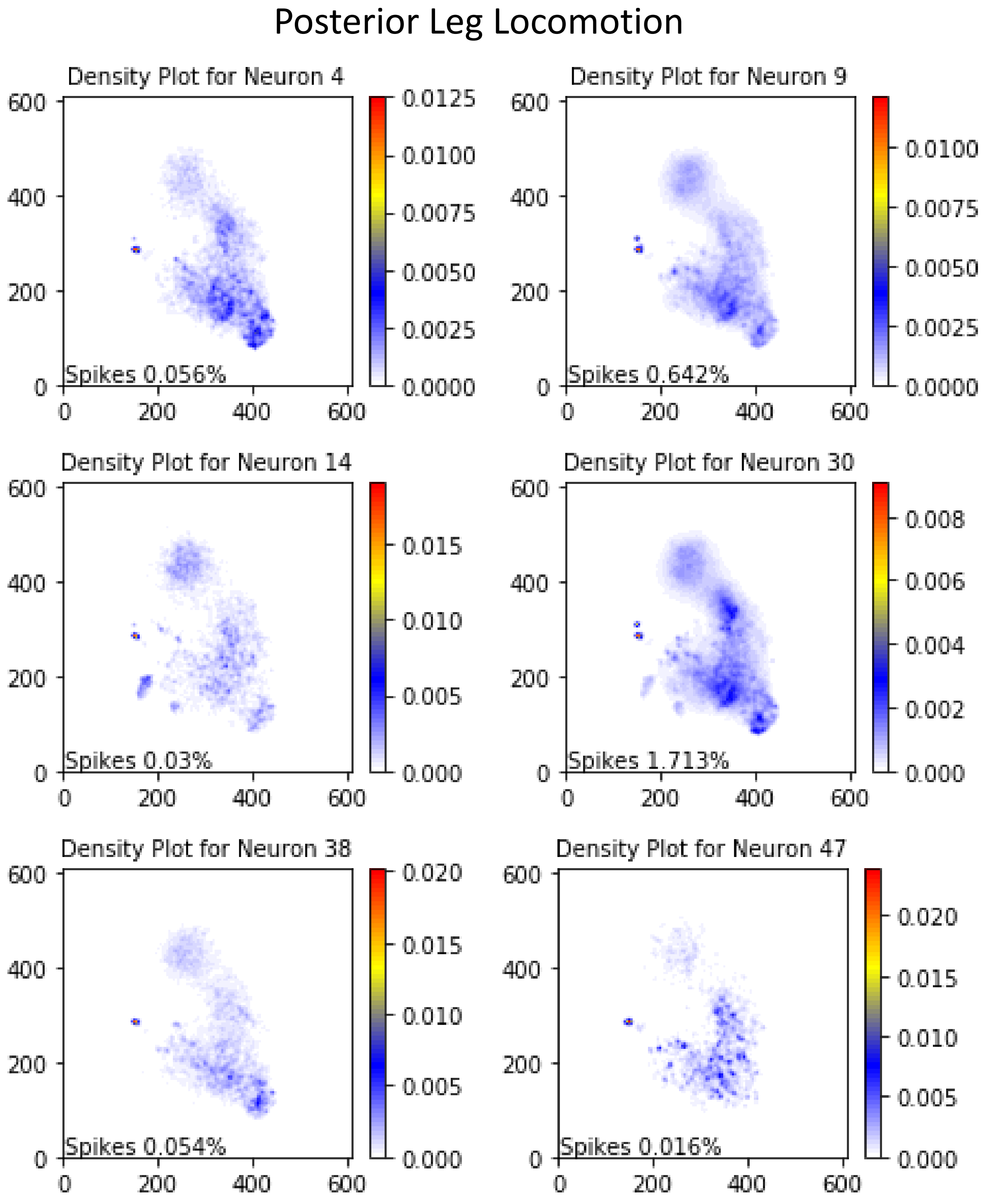

Figure 25: Density Plot for units that fire at posterior leg locomotion regions

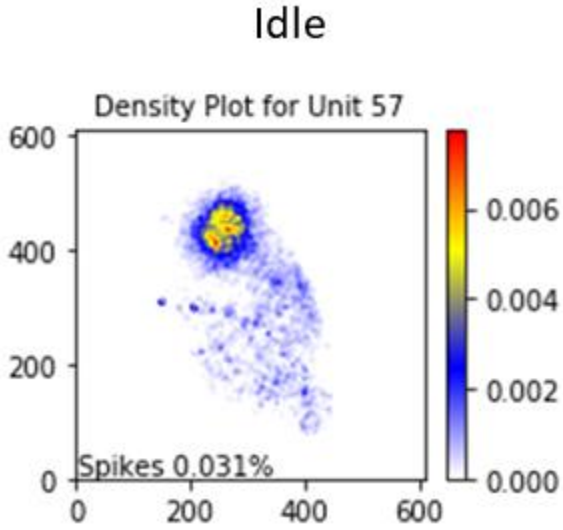

Figure 26: Density Plot for unit that fire at for firing at idle regions

Discussion

Limitations

Variational Autoencoder

There are a few limitations in this analysis, in particular, as the variational autoencoder was not completely successful in fixing the data without the centered sensor. This could possibly be fixed with running a dropout function in the encoder to enable the autoencoder to learn dropped data values to predict the missing value for the postural data (Wang et al., 2018).

Having the error sensor closest to the rodent's actual posture is an important factor in using VAE to fix postural data. It would be beneficial to compare VAE prediction methods between filling the error values with median, interpolation, and other methods such as Kalman Filters (Wan et al., 2000). As the rodent's motion could be simple to complex depending on the situation, there could be better methods of computing missing error points for better input for a better prediction using VAE. We would quantitatively compare these by using VAE defined errors, mentioned more detailed in methods section.

This would have to be done through analyzing how sensors have errors in temporal measurements. For sensors with low consequent measurement errors, we will not have to use complicated methods to fill in the missing values as the filling method for error sensors will be quite similar to the actual position of the rodent. However, as problems mostly occur for certain rodent's posture or position of the rodent in the recording area, errors tend to be consequential for a long period of time (> 5 seconds). The longer the rodent went with a tracking error, the more difficult it is to predict the actual position of the rodent using rest of the positions of the sensors. This also holds true for tracking errors for multiple sensors: as number of tracking errors increases in a given particular time series, the harder it is the predict the correct position of the rodent.

Another problem in using VAE is that optimizing hyperparameters of the VAE layers will need to be different for different set-ups of experiments. While running the VAE itself takes long, optimizing the number of layers, dimension of the hidden layers, and activation function for each of the transitions from one layer to the next will take a longer time.

Lastly, finding a well-trained VAE could be difficult, as over-fitting can occur as one tries to reduce latent or validation loss of the model. Unlike VAE for other experiments, the only way to truly know if the VAE predictive model works is to go through an animation of rodent's predictive and measured posture as we ourselves do not truly know the correct position of the rodent with tracking errors.

Future Challenges

Analyzing Latent Value

One interesting topic to look for in future studies would be a study of latent values of stereotyped behaviors. In the VAE, we have VAE that has 10 latent values in the last layer. As the latent spaces are a reduced dimension of the actual posture, we expect to find similar spiking latent values for each of the similar behavioral stereotyped actions such as grooming or rearing.

Analyzing each of the latent values in terms of rodent's posture could provide interesting results of how the VAE defines animal's posture. This could be done through either looking at high spikes per latent value and matching them to animal's posture or vice versa.

Predicting Behavior

We hope to be able to analyze neural recordings to predict behavioral state of the rodent based on neural recordings in the future. We define $p_{\vec{x}}$ as the probability of spiking at a behavioral region corresponding to the position at time t for neuron n_i. We similarly calculate the probability of behavioral state $p(x, y|N)$ for neurons n_i.

$$p_{\vec{x}}(behavior|n_i) = \frac{p(n_i|behavior) * p(behavior)}{p(n_i)} \tag{4}$$

$$p(x, y|N) = \prod_i p(x, y|n_i) * \prod_i \big(1 - p(x, y|{\sim}n_i)\big) \tag{5}$$

We also suggest a possibility in predicting behaviors using both the latent space of VAE and neural recording combined. VAE sample is a reduced dimensional data of the posture. With the latent value and the neural data linked to each of the time series, it could be possible to create a variational autoencoder that predicts the rodent's next behavior. This would involve a combination to produce simultaneous dynamical posturing and neural activity (Berman et al., 2018).

As discussed before, there are animal behaviors that have neural activity lasting over longer periods such as hunger and mating interactions. Instead of observing neural spikes of that single time point, it would be helpful to use a combination of previous neural spiking data combined with the latent values of VAE of the rat posture to predict the behavior in the future. For example, we should train the VAE with latent value of the rodent’s posture along with 10 seconds of rodent’s neural recording values to predict the next behavior of the rodent.

Methods

Data

Our initial rat postural data has $19980000 \times 20 \times 3, 19980000$ being the time series (measured in 300 fps over a day) with 20 sensors of x, y, z coordinates. To reduce dimension and align the rodent to a certain degree, we center one of the sensors in the back of the rodent and rotate the data so that the other two remaining sensors on the back always point to the right.

There are error points in the data with values below 0.001 for all three dimensions of a specific sensor. We defined those as “bad” data points and filled them with missing values. We use interpolate method to fill in the missing values for later VAE prediction method.

Variational Autoencoder

Training the VAE

Training the hyperparameter to reduce the loss function as much as possible is needed to have a VAE that fixes error points. We use a hyperparameter of three hidden layers for both the encoder and the decoder.

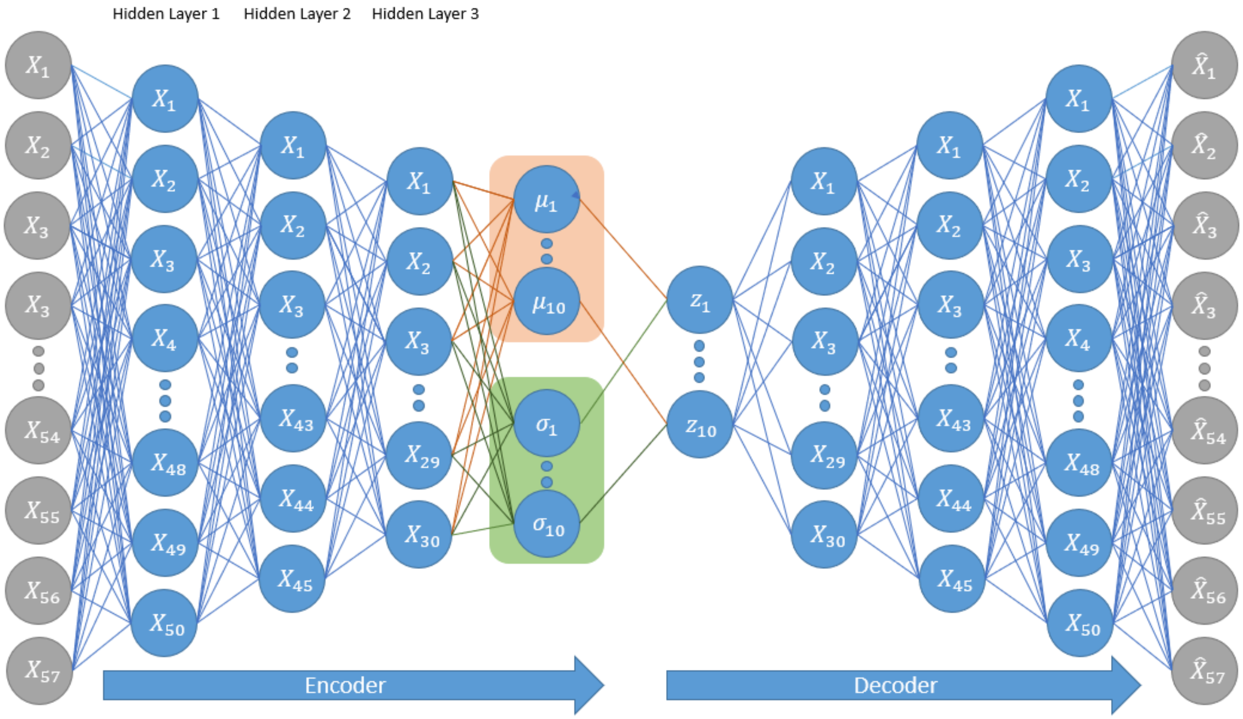

Figure 27: Variational Autoencoder Layers. This shows the reducing dimensions from 57, 50, 45, 30, 10 (mean,and variance), and sampling for the encoder. For the decoder, this shows how latent values z goes from 10, 30, 45, 50, and 57 (original dimension of the rodent posture). At the very last layer of the decoder, the rodent's posture is reconstructed.

Encoder

In this thesis, we use an input shape of 57 (coming from three-dimensional data of 19 sensors), reducing them to 50, reducing again to 45 with a '$tanh$' function, then to 30, and then 10 without any activation functions. The bottleneck layer contains the mean, variance, and sample for the latent space.

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \left(\frac{e^{x} - \mathrm{e}^{-\mathrm{x}}}{\mathrm{e}^{x} + e^{-x}}\right) \tag{6}$$

Decoder

The decoder mainly mirrors the encoder as it goes from input of sample of the latent space to $30, 45, 50$ with 'tanh' function, then to 57. The resulting final layer of the Decoder is what we would define as VAE-fixed rat posture.

Latent Loss

We use KL divergence for the latent function. The KL divergence between two probability distributions measures the divergence from one another. Minimizing the KL divergence value optimizes μ and σ to resemble a similar distribution.

$$D_{KL}[N(\mu(X),\sum(X))||N(0,1) = \frac{1}{2}\sum_{k}(e^{\Sigma(X)} + \mu^2(X) - 1 - log\sum(X)) \tag{7}$$

Training Data Selection

For training the VAE (Figure 29), we only use “good” data points without missing values as we defined in the Data section using Equation 1 to define VAE error data. We use about 16.7% of validation training data count compared to the training data.

Initially, we run the VAE with all the good points, although we do acknowledge there will be error points due to tracking errors as suggested in the introduction and results. We train the VAE about 3000 epochs selecting the one with the lowest validation error.

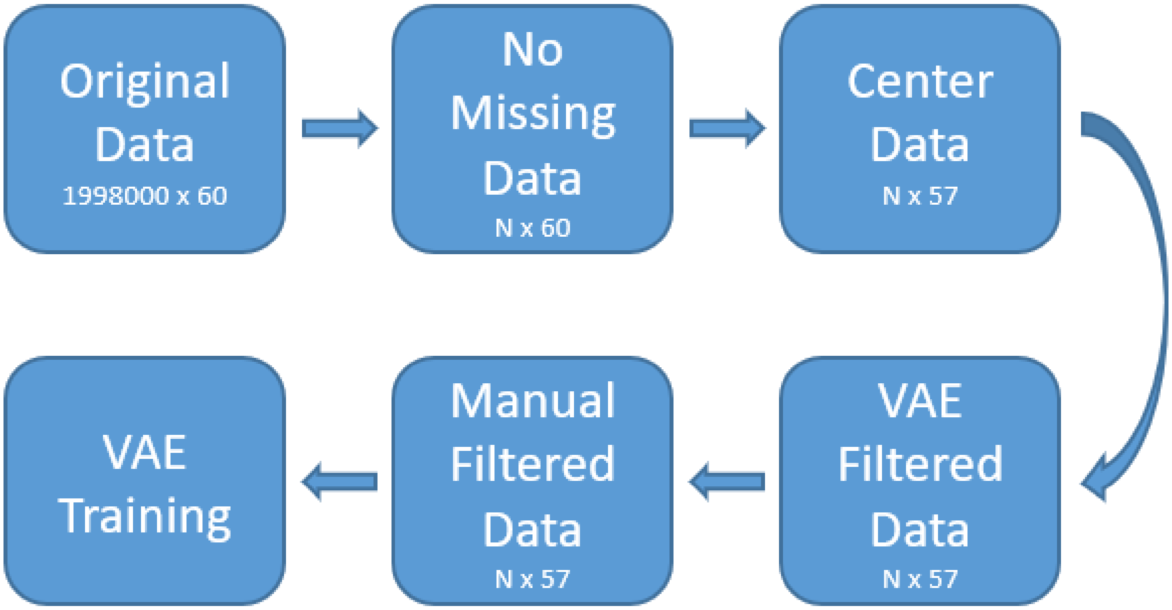

Figure 28: Training VAE: dataset selection pipeline. The data is first done through removing missing values in the data, errors in tacking in the postural dataset. Then we center the rat data to run the initial VAE. Then we remove VAE filtered data along with manual filtering of the data to run the final VAE on them.

As there are more accurate samples in the training data, we observe that the VAE does fix some of the tracking errors to a certain degree. However, the main purpose of running the first VAE with “good” data points is to remove data with multiple tracking errors. After training the VAE, we use the prediction method to compare individual sensors to observe if there are any high differences (over 20 pixels) from the original data.

After removing VAE-defined error time series, we observe the three-dimensional movie of the remaining time series of the posture to de-select any major problems in tracking.

We then use the VAE to train about 5000 epochs, selecting the one with lowest validation error. As a result, we obtain an error of 4.24 pixels, equivalent to about $2\ mm$.

One thing to keep in mind is that repeating these steps multiple times can produce over-fitting data that will not correctly predict “bad” data with error points even though we obtain a smaller loss value by over-fitting.

One thing to keep in mind is that repeating these steps multiple times can get you over-fitting data where it won’t correctly predict “bad” data with error points even though we obtain a smaller loss value, over-fitting.

Another essential point is although the error value for the VAE will be somewhat proportional to how well the VAE will predict the missing values-interpolated rat postural data, the only way to truly look at the accuracy of VAE is to observe the movie for the rodent’s actions.

PCA

PCA by default centers the data and uses singular value decomposition (SVD) algorithm. PCA returns three values: coeff, score, latent. Coeff is the eigenvectors of covariance matrix of raw data arranged in descending order. Score is the input rotated to a new basis of principal components. Latent is the eigenvalues of a covariance matrix of the input arranged in descending order.

We choose the top principal components containing higher variance of the data. It is used essentially as a filtering method to select only the most significant principal components for explaining 95% of the variance of the entire data.

Morlet Wavelet Transform

Morlet is a complex continuous exponential Fourier with a Gaussian that helps preserve localization.

$$W_{s,\tau}[y(t)] = \frac{1}{\sqrt{s}} \int_{-\infty}^{\infty} y(t)\psi^{*}\left(\frac{t-\tau}{s}\right) dt \tag{8}$$

using

$$\psi(\eta) = \pi^{-\frac{1}{4}} e^{i\omega_0 \eta} e^{-\frac{1}{2\eta^2}} \tag{9}$$

Here, $y(t)$ is the time series; s is the time scale of interest; τ is a point in time; and ω_0 is a non-dimensional parameter.

Morlet wavelet relates to time scale, s, to Fourier frequency, f.

$$s(f) = \frac{\omega_0 + \sqrt{2 + \omega_0^2}}{4\pi f} \tag{9}$$

To correct disproportionally large $A(s, \omega)$, we find all scalar function $C(s)$ so that ω^* is 2π times to Fourier frequency. $C(s)A(s, \omega^*) = 1$ for all s

$$C(s) = \frac{\pi^{-\frac{1}{4}}}{\sqrt{2s}} e^{\frac{1}{4\left(\omega_0 - \sqrt{\omega_0^2 + 2}\right)^2}} \tag{10}$$

We can then define the power spectrum, $S(k, f; \tau)$, using the function below.

$$S(k, f; \tau) = \frac{1}{C\big(s(f)\big)} \left| W_{s(f),\tau}[y_k(t)] \right| \tag{11}$$

Finally, we create a wavelet spectrogram resolved for the selected latent values.

$f_i = f_{max} 2^{\frac{i-1}{N_f - 1} \log_2 \frac{f_{max}}{f_{min}}}$ spaced between $f_{min} = 1Hz$ and $f_{max} = 150Hz$.

t-SNE

Initially for the embedding used in t-SNE, we use Kullback-Leibler divergence to minimize the cost function

$$C = D_{KL}(P||Q) = \sum_{ij} p_{ij} \log \frac{p_{ij}}{q_{ij}},\ \#(12)$$

where $p_{ij} = \frac{1}{2}\left(p_{j|i} + p_{i|j}\right),\ q_{ij} = \frac{\left(1+\Delta_{ij}^2\right)^{-1}}{\sum_k \sum_{l\neq k}\left(1+\Delta_{k,l}^2\right)^{-1}}$, and Δ_{ij} is the Euclidean distance between points i and j within the embedded space. As t-SNE has poor memory complexity scaling, we use a sampling technique to choose $50{,}000$ data points for each training set. Then using each of the training sets to build the embedding space, we re-embed the remaining points into the embedded space. We then use this embedded space to estimate a probability density by convolving each point with a two-dimensional Gaussian which has a value of distance from the point to 10 nearest neighbors. Then we segment the space using watershed transform to create regions.

We embed resulting embedded spaces for each of the training sets to create a behavioral space by comparing each one. We define X as the composition of all feature vectors in the training set, X' as the associated embeddings from t-SNE, z as a new feature vector that we will be embedding depending on the mapping between X and X', and ζ as the embedding of z.

We embed z by setting the transitional probability of two spaces as similarly as possible. The transition of full space $p_{j|z}$ is defined by

$$p_{j|z} = \frac{\exp\left(\frac{d(z,j)^2}{2\sigma_z^2}\right)}{\sum_{x\in X} \exp\left(-\frac{d(z,k)^2}{2\sigma_z^2}\right)} \qquad (13)$$

where $d(z,k)$ is the KL divergence between z, k, and σ_z using the same parameters as t-SNE embedding. The transitional probabilities of the embedded space are calculated using

$$q_{i|\zeta} = \frac{\left(1+\Delta^2_{\zeta,x'}\right)^{-1}}{\sum_{x'\in X'} \exp\left(1+\Delta^2_{\zeta,x'}\right)^{-1}}. \tag{14}$$

For each z, we try to find ζ^* that minimizes KL divergence

$$\zeta^* = arg \min_{\zeta} D_{KL}(p_{x|z}||q_{y|\zeta})\,. \tag{15}$$

Instead of starting local optimization from ζ_0, the weighted average of points, using Nelder-Mead simplex algorithm, we start from point $y(x^*)$ where

$$x^* = arg \max_{x} p_{x|z} \tag{16}$$

which gives a better solution for 5% of the time (Berman et al., 2014).

We then use watershed transform to create a number of regions that divides the map depending on the σ value increasing from 0.35 by 0.05 incrementally to create a map with maximum regions below 150 total regions.

Transition Matrix

For analyzing the transitions, we first change the watershed regions, a time series with values corresponding to the region number in the behavioral map, into a list of transition states, eliminating non-region time series and consecutive same region values to be represented only once.

Predictive Information Bottleneck

The informational bottleneck method is designed to find the best tradeoff between accuracy and complexity when clustering a variable given a joint probability distribution.

$$F = I\big(Z; S(n+\tau)\big) - \beta I\big(Z; S(n)\big) \tag{17}$$

This follows self-consistent equations that could be explained using a similar log to Blahut-Arimoto algorithm in rate-distortion theory (Tishby et al., 1999, Blahut et al., 1972). Defining β as inverse temperature, $x \in S(n),\ y \in S(n+\tau), z \in Z, D_{KL}$ as KL divergence between the probability distributions, and $Z(B,x)$ as normalizing function, self-consistent equations are iterated until $[(F_t - F_{t+1}]/F_t \ < 10^{-6}$, convergence criterion, is qualified.

$$p(z|x) = \frac{p(z)}{Z(B,x)} \exp\left[-\beta D_{KL}\big(p(y|x)\big|\big|p(y|z)\big)\right] \tag{18}$$

$$p(z) = \sum_{x} p(z|x)p(x)\,, \tag{19}$$

$$p(y|z) = p(y|x)p(z|x)p(x), \tag{20}$$

We start from $\beta = 0.1$ to $\beta = 500$ with a 40 exponentially spaced values to find the solution. Starting at a random initial state, optimization is performed until the convergence criterion is satisfied and then that solution is used as the next initial condition for the next β. We repeat this 24 times with different random starting initial values for $K\ = 2, \dots, 25$ and for 81 time lag values from $n\ = 1$ to $n\ = 5{,}000$.

After calculating the solutions, we use each clustering($p(z|x) = \delta_{z,arg\,\underset{x}{\max}\, p(z'|x)}$)’s deterministic limit and recalculate $I(Z; S(n))$ and $I(Z; S(n+\tau))$. We define Pareto front, ζ^n, as set of all solution so that no other solutions for that given lag has a larger value for $I(Z; S(n+\tau))$ and smaller value for $I(Z; S(n))$.We choose a maximum of 23 clusters, and when we choose fixed numbers of clusters we choose the representation that has the highest value of $I(Z; S(n+\tau))$.

References

VAE t-SNE supplemental materials10.5281/zenodo.4685124

Berman, G. J., Bialek, W., & Shaevitz, J. W. (2016). Predictability and hierarchy in Drosophila behavior. Proceedings of the National Academy of Sciences, 113(42), 11943-11948. doi:10.1073/pnas.1607601113

Berman, G. J., Choi, D. M., Bialek, W., & Shaevitz, J. W. (2014). Mapping the stereotyped behaviour of freely moving fruit flies. J R Soc Interface, 11(99). doi:10.1098/rsif.2014.0672

Berman, G.J. Measuring behavior across scales. BMC Biol 16, 23 (2018). https://doi.org/10.1186/s12915-018-0494-7

Kingma, Diederik P, and Max Welling. “Auto-Encoding Variational Bayes.” ArXiv.org, 1 May 2014, arxiv.org/abs/1312.6114v10.

Rezende, Danilo, and Mohamed Shakir. “Stochastic Backpropagation and Approximate Inferences in Deep Generative Models.” ArXiv.org, 1 Jan 2014 arxiv.org/abs/1401.4082

van der Maaten L, Hinton G. 2008 Visualizing data using t-SNE. J. Mach. Learn. Res. 9, 85.
Wang XJ, Krystal JH. Computational psychiatry. Neuron. 2014;84(3):
638–54.

Lebedev MA, Nicolelis MAL. Brain-Machine Interfaces: From Basic Science to Neuroprostheses and Neurorehabilitation. Physiol Rev. 2017;97(2):767–837.

Goupillaud P, Grossman A, Morlet J. 1984 Cycleoctave and related transforms in seismic signal analysis. Geoexploration 23, 85–102. (doi:10.1016/0016-7142(84)90025-5)

Daubechies I. 1992 Ten lectures on wavelets. Philadelphia, PA: SIAM.

Meyer F. 1994 Topographic distance and watershed lines. Signal Process. 38, 113–125. (doi:10.1016/0165-1684(94)90060-4)

Takahata M, Yoshino M, Hisada M (1981) The association of uropod steering with postural movement of the abdomen in crayfish. J Exp Biol 92(1):341–345.

Bialek W, Nemenman I, Tishby N (2001) Predictability, complexity, and learning. Neural Comput 13(11):2409–2463.

Tishby N, Pereira FC, Bialek W (1999) The information bottleneck method. Proceedings of the 37th Annual Allerton Conference on Communication, Control and Computing,

Dawkins R (1976) Hierarchical Organization: A Candidate Principle for Ethology in Growing Points in Ethology, eds Bateson P, Hinde R (Cambridge Univ Press, Cambridge, UK), pp 7–54.

Simon HA (1962) The architecture of complexity. Proc Am Philos Soc 106(6):467–482.

Heiligenberg W (1973) Random processes describing the occurrence of behavioural patterns in a cichlid fish. Anim Behav 21(1):169–182.

Miller GA (2003) The cognitive revolution: A historical perspective. Trends Cogn Sci 7(3):141–144.

Blahut RE (1972) Computation of channel capacity and rate-distortion functions. IEEE Trans Inf Theory IT-18(4):460–473.

Gould JL. 1982 Ethology: the mechanisms and evolution of behavior. New York, NY: W. W. Norton and Company.

Stephens GJ, Osborne LC, Bialek W. 2011 Searching for simplicity in the analysis of neurons and behavior. Proc. Natl Acad. Sci. USA 108, 15 565–15 571. (doi:10.1073/pnas.1010868108)

E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," *Proceedings of the IEEE 2000* Adaptive *Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, Lake Louise, AB, Canada, 2000, pp. 153-158, doi: 10.1109/ASSPCC.2000.882463.

Dell AI, Bender JA, Branson K, Couzin ID, De Polavieja GG, Noldus LPJJ,et al. Automated image-based tracking and its application in ecology. Trends Ecol Evol. 2014;29(7):417–28. Egnor SER, Branson K. Computational Analysis of Behavior. Ann Rev Neuro. 2016;39:217–36.

Wang, Dongfang, and Jin Gu. “VASC: Dimension Reduction and Visualization of Single-Cell

RNA-Seq Data by Deep Variational Autoencoder.” Genomics, Proteomics & Bioinformatics, vol. 16, no. 5, 2018, pp. 320–331., doi:10.1016/j.gpb.2018.08.003.

Stephens GJ, Osborne LC, Bialek W. Searching for simplicity in the analysis of neurons and behavior. Proc Nat Acad Sci. 2011;108(Supp 3):
15565–71.

Gomez-Marin A, Paton JJ, Kampff AR, Costa RM, Mainen ZF. Big behavioral data: psychology, ethology and the foundations of neuroscience. Nat Neurosci. 2014;17(11):1455–62.
Jones, S., Neville, V., Higgs, L. et al. Assessing animal affect: an automated and self-initiated judgement bias task based on natural investigative behaviour. Sci Rep 8, 12400 (2018).

Kingma, Diederik P, and Ba, Jimmy. “Adam: A Method for Stochastic Optimization.” ArXiv.org, 1 May 2014, arxiv.org/abs/1412.6980