Encoding the Structure of Animal Motion Dynamics using Variational Auto-Encoding

DISSERTATION

zur Erlangung des akademischen Grades

Doktoringenieur (Dr.-Ing.)

angenommen durch die Fakultät für Informatik
der Otto-von-Guericke-Universität Magdeburg

von Master of Science Kevin Luxem

geb. am 19.11.1989 in Andernach

Gutachterinnen/Gutachter

Prof. Dr. Sebastian Stober
Prof. Dr. Stefan Remy
Prof. Dr. Jürgen Gall

Magdeburg, den 23.05.2024

"We may regard the present state of the universe as the effect of its past and the cause of its future. An intellect which at any given moment knew all of the forces that animate nature and the mutual positions of the beings that compose it, if this intellect were vast enough to submit the data to analysis, could condense into a single formula the movement of the greatest bodies of the universe and that of the lightest atom; for such an intellect nothing could be uncertain and the future just like the past would be present before its eyes."

Marquis Pierre Simon de Laplace, 1814

Abstract

Quantifying and detecting the hierarchical organization of behavior presents a significant challenge in neuroscience. Recent advancements in markerless pose estimation have enabled the spatiotemporal tracking of behavioral dynamics. However, there is a pressing need for robust and reliable technical approaches that can unveil the underlying structure within these data and segment behaviour into hierarchically organized motifs. In this thesis, I propose an unsupervised probabilistic deep learning framework called Variational Embeddings of Animal Motion (VAME) that addresses these challenges. By leveraging VAME, I can identify the behavioural structure from deep variational embeddings of animal motion, providing a powerful tool for behavioural analysis. To demonstrate the framework's effectiveness, I utilize a mouse model of beta amyloidosis as a use case. The results demonstrate that VAME not only identifies discrete behavioral motifs, but it also captures a hierarchical representation of how these motifs are utilized. This hierarchical representation allows for the grouping of motifs into communities, revealing intricate behavioral patterns that were previously overlooked by human visual observation. Remarkably, VAME detects differences in community-specific motif usage among individual mouse cohorts that were previously undetectable without the framework's aid. Importantly, the proposed approach offers robust segmentation of animal motion, making it applicable to a wide range of experimental setups, models, and conditions. It eliminates the need for supervised or a-priori human interference, which greatly enhances its versatility and efficiency. VAME's unsupervised nature also alleviates the burden of manual annotation and subjective biases associated with traditional methods of behavioral analysis. In summary, this work presents a significant advancement in the field of behavioral neuroscience by providing a powerful and unsupervised framework for uncovering the hierarchical organization of behavior. VAME's ability to identify discrete motifs, capture their hierarchical representation, and detect differences in motif usage among communities paves the way for deeper insights into the complex dynamics of animal behavior.

Contents

List of Abbreviations

AD	Alzheimer's Disease
Adam	Adaptive moment estimation
AE	Autoencoder
ANN	Artificial Neural Network
AR-HMM	Autoregressive Hidden-Markov Model
BiRNN	Bi-directional Recurrent Neural Network
CMOS	Complementary Metal-Oxide Semiconductor
CNN	Convolutional Neural Network
DLC	DeepLabCut
ELBO	Evidence Lower Bound
GAN	Generative Adversarial Network
GPU	Graphical Processing Unit
GRU	Gated Recurrent Unit
HMM	Hidden-Markov Model
LSTM	Long Short Memory Networks
MCMC	Markov Chain Monte Carlo
MLP	Multilayer Perceptron
MSE	Mean Squared Error
NMI	Normalized Mutual Information
MoSeq	MotionSequencing
PCA	Principal Component Analysis
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
SSL	Self-Supervised Learning
TED	Tree Edit Distance
TG	Transgenic

t-SNE	t-distributed Stochastic Neighbor Embedding
UMAP	Uniform Manifold Approximation and Projection
VAE	Variational Autoencoder
VAME	Variational Animal Motion Embedding
VAE-RNN	Variational Recurrent Neural Network Autoencoder
VI	Variational Inference
WT	Wildtype

Chapter 1

A Science of Quantitative Behavior

This thesis represents the culmination of my work as a Doctoral candidate at the German Center for Neurodegenerative Diseases in Bonn and the Leibniz Institute for Neurobiology in Magdeburg. Within the duration of my doctoral candidacy, this work has yielded the following significant contributions:

- Kevin Luxem, Falko Fuhrmann, Stefan Remy and Pavol Bauer. *Hierarchical network analysis of behavior and neuronal population activity*. 2019. Conference of Cognitive Computational Neuroscience

- Kevin Luxem, Jennifer J. Sun, Bradley S. Peterson, Keerthi Krishnan, Talmo D. Pereira, Eric A. Yttri, Jan Zimmermann and Mark Laubach. *Open-Source Tools for Behavioral Video Analysis: Setup, Methods, and Development* 2022. eLife

- Kevin Luxem and Petra Mocellin. *Self-supervised learning as a gateway to reveal underlying dynamics in animal behavior*. 2022. 12th International Conference on Methods and Techniques in Behavioral Research

- 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*. 2022. Nature Communications Biology

- Petra Mocellin, Oliver Barnstedt, Kevin Luxem, Hiroshi Kaneko, Anna Karpova, Julia Henschke, Dennis Dalügge, Falko Fuhrmann, Janelle Pakan, Michael R. Kreutz, Sanja Mikulovic and Stefan Remy. *A septal-VTA circuit drives exploratory behavior*. 2023. Accepted in Neuron

- Stephanie R. Miller, Kevin Luxem, Kelli Lauderdale, Nick Kaliss, Katie K. Ly, Pranav Nambiar, Yuechen Qiu, Catherine Cai, Kevin Shen, Zhaoqi Yan, Andrew Mendiona, Takashi Saito, Takaomi C. Saido, Alexander R. Pico, Reuben Thomas, Katerina Akassoglou, Pavol Bauer, Stefan Remy and Jorge J. Palop. *Machine Learning Reveals Prominent Alterations in Spontaneous Behavior of Preclinical and Clinical Stage Mouse Models of Alzheimer's Disease*. 2023. In Review at Neuron

1.1 Classical ethology

The movement of body parts, such as limbs, muscles, and joints, and the resulting behavior is essential for an animal's survival and reproduction. They allow animals to perform tasks such as hunting for food, escaping from predators, and finding a mate. The ability to move efficiently and effectively also plays a role in the evolution of species, as animals that can control their body parts more successfully are more likely to survive and pass on their genetic traits to future generations. Within an animals nervous system, the motor cortex plays a critical role in controlling and coordinating these movements. The motor cortex is the region of the brain responsible for planning, initiating and controlling voluntary movements; it sends signals to the muscles via the spinal cord, which then generate movement. This allows animals to perform complex movements Kandel, Schwartz, and Jessell [1]. Additionally, it plays a key role in the learning and adaptation of movement patterns, allowing animals to improve their movement skills over time.

Before researchers were able to study both animal movements and brain activity simultaneously, ethologists were the pioneers in studying the fundamental elements of behavior by observing the execution of movement patterns and their functions. Ethology, the study of animal behavior in its natural environment, emerged as a distinct discipline in the 1920s. The founders of this field are Austrian biologists Konrad Lorenz and Nikolaas Tinbergen, who were awarded the Nobel Prize in *Physiology or Medicine* in 1973 (together with Karl von Frisch) for their pioneering work in the study of behavior. They focused on understanding the evolutionary and adaptive functions of animal behavior. Their work has had a lasting impact on the way we interpret and study animal behavior.

1.1.1 Tinbergen's four questions

Tinbergen's key contribution was his work on the four questions of behavior, which include the (1) proximate mechanisms, (2) development, (3) function, and (4) evolution of behavior (Table 1.1) [2]. In his view, behavior is built from simpler actions, which he called *elementary responses*. He proposed that these elementary responses are combined in various ways to form more complex behaviors, so called fixed action patterns. His concept of fixed action patterns was based on the idea that certain behaviors are innate and triggered by specific stimuli. In his work, he observed that many animals exhibit complex behaviors that are highly stereotyped and consistent across individuals and populations. Another contribution of Lorenz's and Tinbergen's work is the concept of *instinct*, which is defined as a fixed, innate, and unlearned action pattern. Hence, instincts are inherited behaviors that are present at birth and are not learned through experience.

In modern ethology Tinbergen's questions are still used as a framework for studying animal behavior ranging from kinship, conflict, sexual selection, foraging, exploration, and aggression across a variety of species. Answering these questions is challenging and requires the use of a variety of scientific methodologies. For example, addressing the question about (1) *Mechanism*, scientists study the physiological and neurological processes that underlie behavior. They examine brain activity using non-invasive techniques like electroencephalography, functional magnetic resonance imaging, or invasive techniques like two-photon imaging, mutli-electrode-arrays, or optogenetic manipulations [3, 4, 5, 6]. Answering (2) requires investigating the animals at different ages or stages of development. Addressing question (3) can only be answered through a variety of different species in the context of their environment and evolutionary history. The last question (4) can be answered through an animal's ecology and assess its impact on the animal's survival and reproductive success. For instance, this can be done by manipulating its behavior and measuring the effects on the animal's fitness.

1.1.2 Modern definition of ethology

Tinbergen definition of ethology is "the total movements made by the intact animal" [7, 2]. A recent study updated this definition to better reflect the current understanding of behavior among scientists. The updated definition states that "behavior is the internally

Table 1.1: Tinbergen's four questions regarding the explanation of animal behavior.

Tinbergen's Questions	Definition
Mechanism	What are the underlying mechanisms of behavior? Understanding the underlying mechanisms of behavior involves studying the complex interplay of physiological and neurological processes that give rise to observable behavior. This can include investigating the neural circuits and pathways involved in generating behavior, as well as the hormones, neurotransmitters, and other physiological factors that modulate those neural circuits.
Development	How does behavior change throughout an animal's lifespan? Understanding how behavior changes throughout an animal's lifespan is a crucial aspect of studying animal behavior. This can involve investigating how behavior patterns, motivations, and abilities change as the animal goes through different stages of development. For example, young animals may exhibit different behaviors than adults, and these behaviors may change as the animal matures. Additionally, it can provide insights into how different behaviors are acquired and how they are integrated into the animal's overall behavior repertoire.
Evolution	How has behavior evolved over time across different species? Investigating how behavior has evolved over time across different species is a crucial aspect of understanding animal behavior. By studying behavior in different species and examining its adaptive value in the context of the animal's environment and evolutionary history, researchers can gain insights into how behavior has changed over time and how it is shaped by natural selection.
Function	What is the function or significance of behavior in the animal's life? This involve examining the behavior in the context of the animal's ecology and evaluating its effects on the animal's survival and reproductive success. With this, researchers can gain insights into how behavior contributes to the animal's overall fitness.

coordinated responses (actions or inactions) of whole living organisms (individuals or groups) to internal and/or external stimuli, excluding responses more easily understood as developmental changes" [8]. This definition emphasizes that behavior is a response to stimuli, whether internal or external and highlights the fact that behavior is an "internally coordinated response" that reflects the dynamics of the brain and its actions. Finding a common language between behavioral structure and brain activity is one of the most compelling goals of modern neuroscience. Modern day researchers aim to uncover behavioral structure and correlate it with brain activity to gain insights how the brain generates behavior [9, 10].

1.2 Computational ethology

Observable motion, or the ability to detect and observe changes in movement, is a crucial aspect for understanding the workings of the brain and its dynamic processes [11, 12]. By studying animal motion, researchers are able to gain insights into the neural mechanisms that underlie sensory perception, motor control, and cognitive processes such as attention and decision making. Acquiring this knowledge not only enhances our comprehension of neurological disorders and facilitates the creation of innovative treatments but also provides valuable insights for refining technologies, including the development of advanced prosthetics and robots designed for seamless interaction with humans. To analyze movement patterns in animals with precision and effectiveness, the application of computational techniques is indispensable. These methods empower the identification of intricate patterns and facilitate the streamlined processing of extensive experimental data. Notably, the field of *Computational Ethology* has emerged in recent years, dedicated to addressing these specific research inquiries.

1.2.1 Definition

The term computational ethology was introduced and defined in 2014 as an interdisciplinary field that uses mathematics, engineering, and computer science to measure and model animal behavior [9]. Ethology has traditionally involved qualitatively observing animal behavior, quantifying relevant variables by note-taking, and deriving a description based on the observations. However, in recent years, there has been a shift towards

data-driven quantitative methods to measure and analyze behavior. This includes using specific criteria to evaluate and assign numerical values to observations. For example, with the advancement of technology, it has become easier to use computer-assisted video analysis tools to label and measure behavior frame-by-frame. This allows for the calculation of statistics such as the frequency, latency, duration, and relative proportion of different behaviors in large datasets.

Human observation is, however, still the most prevalent approach for scoring behavior, which can introduce several limitations on the analysis and data acquisition of experiments. One major limitation is that manual scoring or labelling of video frames is often time-consuming and can be subject to human error, especially for complex experiments. This can lead to inconsistent or inaccurate data, which can hinder progress in understanding neural circuit functions of relevant behaviors [13, 14]. A further shortcoming is that it is often not feasible to perform large-scale experiments due to the time and resources required for manual scoring. This can limit the scope of studies and the number of behaviors that can be analyzed. Additionally, depending on the complexity of the experiment, it may not be possible to assign scores to certain behaviors trivially, which can further limit the accuracy and reliability of the data (see Table 1.2).

Within the field of neuroscience, recent advancements in technology have enabled researchers to measure and manipulate brain activity with increasing precision and accuracy. This includes techniques such as calcium imaging for recording brain activity and optogenetics for manipulation of brain activity [15, 16, 17]. However, while there have been significant advancements in the technology used to measure brain activity, the technology to measure behavior has not progressed at the same rate [11, 10]. This means that researchers have a more detailed understanding of what is happening in the brain, but do not have the same level of insight into how this activity manifests in an individual's behavior. This can make it difficult to fully understand the relationship between brain activity and behavior. Computational ethology aims to close the gap between the advancements in neural recording and behavior quantification. Moreover, new analysis tools to quantify behavior can be used for high-throughput experiments and allow for the analysis of large amounts of data, which was previously not possible without many hours of human labour.

Table 1.2: Limitations of human annotation for behavioral analysis and data acquisition (adapted from [9]).

Limitations of human annotation	
Slow	Labelling frames by hand is very time consuming and labour intensive. This severely limits the amount of experiments that could be done, the sample size (which limits statistical power), and also the reliability of the results.
Imprecise and Subjective	There is no standardization since different human observer score behavior slightly differently. Hence, it is subjective and inconsistent between observers [8]. In our benchmark dataset presented in chapter four, there is only a 70% overlap between three human observer. This makes it hard to reproduce the same behavioral episodes over multiple datasets or between different laboratories.
Low dimensionality	A simple behavior like walking can be deconstructed into smaller components like specific phases of the fore- and hind-limb movements. Its granularity is ultimately limited by the animals motor system. To measure a single behavior at multiple spatial and temporal resolutions is highly challenging. Lastly, the number of different behavior that can be possibly scored is limited.
Interpretation	Machine learning based algorithm have the opportunity to identify patterns in the data which are not visible or detectable by a human observer due to inattention, ascertainment bias, or timescales.
Language	How an observer classifies some behavior cannot always be described in perfect verbal terms. Hence, training new observers is difficult and it is not guaranteed that they replicate the scoring in the exact same manner.
Repetitive	Scoring videos for hours over multiple days or even weeks will eventually lead to a mind fatigue in a very short time and the attention of the observer will drift. This increases the likelihood of errors but more crucially the chance to discover something interesting within the data.

1.2.2 Pose tracking and segmentation

One of the driving forces to develop modern computational approaches for behavior is the collection of various data. Sensors, tracking devices, cameras, and other technologies have made it possible to collect large amounts of data from the behaving animal over multiple time scales. This includes information on the location, movement, or behavior of the animal. With this data, it is possible to create computational methods that allow scientists to study patterns of behavior or predict their next movement. In this work, as well as in other studies, the focus will be on video data captured from stationary cameras.

To identify patterns in the data, statistical analysis method like regression analysis were used to understand factors that influence behavior. Such techniques are especially useful to identify trends or relationships in the data. Recently, more advanced machine learning algorithms have become increasingly important in the field of neuroscience and other fields due to the vast amount of data that is now available or easily collectable. These algorithms can be trained on large datasets, allowing them to recognize patterns that may not be apparent to humans. Identifying these patterns in animal motion can help scientists understand the underlying mechanisms that drive behavior, when correlated with or combined with neural activity data.

A significant advancement has been made in the field of animal tracking and pose estimation due to artificial neural networks. Historically, this has been done by simply identifying the body mass center of an animal and tracking its position over time. In recent years, there has been the development of numerous open-source tools that enable efficient tracking of animal-body parts via supervised deep learning [18, 19, 20]. One of their most important aspects is the ability to track the body parts of an animal with minimal efforts in labelling data. Moreover, advancements have been made in the area of data-driven segmentation of patterns from behavioral signals, pioneered by several approaches, two of which are highly influential for this work, namely MotionMapper and MotionSequencing (MoSeq) [21, 22]. Both methods use unsupervised techniques that aim to uncover patterns in time series data. All these methods have greatly expanded the scope and depth of measuring animal behavior and provided new insights into the mechanisms and development of behavior.

1.2.3 Aims of computational ethology

A major goal of computational ethology is to quantitatively and accurately measure behavior in all its complexity. It provides an immense opportunity to understand how the brain works as the dynamics of the brain are directly (or indirectly) reflected by the movements of an animal. By building methods that can express behavior in terms of numerical values, scientists are able to correlate brain activity to behavior or derive principles from it. Such methods provide the possibility to uncover behavior patterns that are not easily measurable by other means. By developing methods that have the potential to not only score or label behavior but to also quantify and project the dynamic of the full behavioral repertoire of an animal we can gain insights into the sensory mechanisms of behavior. Moreover, by quantifying the full range of an animals behavior it can be possible to investigate neurological diseases based on pure behavior observations and transfer knowledge into human medicine. This new science of quantitative behavior has the potential to quantify every detail of an animal's movement in near completeness while it performs its ethological relevant task, in an experiment or naturalistic environment. The work presented aims to take a further step in this direction.

1.3 Challenges and central hypothesis

The quantification of behavior within neuroscience has traditionally been simplified to measure simple output variables such as the frequency of a lever press or an animal's location. This was achieved by using operant conditioning chambers to train the animal to perform a specific task. Such an approach has the advantage of providing easily quantifiable behavioral data that can be correlated with neural measurements. However, in recent years, there has been a growing recognition of the importance of behavior that mimics the animal's natural movements [9, 11, 10, 12]. As behavior is the primary output of the brain, treating it as a complex process similar to neural activity, can lead to a new understanding of the relationships between brain activity, behavior, and internal states. This shift in thinking has caused a paradigm shift in neuroscience, with researchers now moving away from using behavior as a mere readout variable and recognizing it as a

complex phenomenon in its own right. And with the advent of better data storage, computational power, and improved graphical processing capabilities, there is now the ability to track and quantify behavior in great detail. Here, I want to build up on recent methodological advancements to measure the dynamical and hierarchical nature of behavior using deep learning, latent variable modelling, information theory, and graph theory to develop an approach for a more complete understanding of behavior. Since an animal's behavior inherently evolves over time, capturing this time-varying structure is the central goal of this work and requires measuring features of the animal's body and pose, tracking those features over time, and then identifying patterns on different hierarchical levels that correspond to different movement patterns, behavioral categories or behavioral states.

1.3.1 Challenges of measuring animal motion

Measuring animal motion presents three main challenges [10]. Firstly, to gain a thorough understanding of naturalistic behaviors, it is vital to take into account the coordinated movements of various body parts such as limbs, facial features, and the animal's three-dimensional pose dynamics, which necessitates the simultaneous measurement of multiple body part positions over time. Secondly, labelling naturalistic behaviors on a moment-to-moment basis is a complex task due to the variability in behavior execution, both in space and time, despite these behaviors often being constructed from stereotyped components [7, 9, 21, 22]. This variability, combined with the continuous evolution of many spontaneous behaviors over time, makes it challenging to accurately assign labels and establish clear start and stop times for each action. Finally, the different levels of granularity at which naturalistic behaviors can be described can lead to multiple valid ways of describing an animal's behavior at any given time point [23]. For instance, an animal can be described as "walking" or at a more detailed level as "moving its left hind leg forward while keeping the right hind leg stationary". Both of these descriptions are accurate, but one is more general while the other is more specific. It is important to note that the choice of description often depends on the specific research question or purpose of the study.

In recent years, numerous approaches to studying animal behavior have emphasized the identification of stereotyped movements or actions, categorizing and labeling these

patterns. While these methods can offer utility, they may fall short of capturing the complete complexity and dynamics of an animal's behavior in every experiment. Many approaches that seek to analyze behavioral patterns or actions concentrate exclusively on identifying clusters through predetermined features or by categorizing behaviors using human labels. However, these approaches fail to consider the dynamic nature of behavior, which can be complex and multifaceted. Here, I argue that it is necessary to identify and understand behavioral patterns or actions in relation to the dynamics and multi-hierarchy of it and maximize the information content from the raw behavioral data itself.

In this thesis, I introduce a methodology aimed at capturing the intricacies of animal motion. Referred to as dynamical embedding, this process allows for a more thorough depiction of the behaviors exhibited by animals. The method facilitates the exploration of behavioral patterns and actions, taking into account the dynamic nature of behavior. By adopting this approach, a more nuanced and comprehensive understanding of behavior can be attained, surpassing the limitations of previous methods that overlook the explicit consideration of behavioral dynamics. This advancement represents a significant improvement in the realms of computational ethology and neuroscience.

1.3.2 Dynamical embedding of animal motion

This thesis introduces a methodology that uses dynamical embedding, a novel approach designed to capture the intricacies of animal motion comprehensively. By employing dynamical embedding, the method delves into a detailed representation of animal behaviors, allowing for a nuanced exploration of behavioral patterns and actions. This approach considers the dynamic nature of behavior, providing a more thorough understanding compared to previous methods that neglect explicit considerations of behavioral dynamics. Consequently, this advancement stands as a significant improvement within the realms of computational ethology and neuroscience.

The central focus of this thesis is to develop an innovative method for capturing latent variables underlying animal movements, using Variational Autoencoder (VAE) models. The proposed approach, named Variational Animal Motion Embedding (VAME), utilizes recurrent neural networks (RNNs) to model multivariate time series data obtained from pose estimation techniques. The primary objective is to quantify the underlying motion patterns within the data. By parameterizing the VAE with an RNN, the model learns

the underlying probability distribution of the dataset, effectively capturing the complex dynamics inherent in animal motion, including subtle movements and hierarchically organized behavioral categories.

The latent variables derived from the VAE offer a unique opportunity to explore the relationship between these variables and neural activity. This exploration provides valuable insights into the neurological processes that underlie animal behavior. To showcase the efficacy of the VAME approach, the thesis demonstrates its application in investigating the correlation between quantitative descriptions of behavioral patterns and measurements of neural activity in the hippocampal CA1 region. This approach, therefore, holds the potential to shed light on how various neural circuits contribute to behavior and elucidate the connections between behavior and cognition.

1.3.3 Behavioral phenotyping in rodents

VAME is designed to detect behavioral patterns in laboratory mice. A primary research focus of the group and institute where this work has been conducted are neurodegenerative diseases, especially Alzheimer's disease (AD). To introduce this disease briefly, AD is a progressive brain disorder that affects memory, thinking, and behavior. It is the most common cause of dementia among older adults. The pathology of AD is characterized by the presence of two types of brain lesions: amyloid plaques and neurofibrillary tangles. Amyloid plaques are deposits of a protein called beta-amyloid that build up between nerve cells in the brain. Neurofibrillary tangles are made up of a protein called tau that forms inside the nerve cells. These changes in the brain lead to the death of nerve cells and the loss of connections between them, causing problems with memory, thinking, and behavior. As the disease progresses, it leads to a decline in cognitive and physical abilities, ultimately leading to death. By building methods that have the potential to detect early subtle changes in mice behavior with this disease phenotype could also have translational impact on human Alzheimer research.

By using AD animals to extract behavioral patterns it is possible to test if the method can distinguish the phenotype of a transgenic and non-transgenic animal's based on subtle differences in behavior. A behavior quantification method that can detect subtle changes in the behavior of such mice would be highly useful in AD research. It would

allow for early detection of the disease, provide a better understanding of the progression of the disease, and help to improve the diagnosis, treatment, and management of the disease in both mice and humans. For example, it could be tested whether a drug can reduce the effects of AD in transgenic mice compared to their wildtype. The method would also allow for a more accurate characterization of the disease in transgenic mice models, which can help to provide insights into the behavioral changes seen in human AD patients, and allow for the development of new therapeutic strategies to treat the disease in humans.

1.3.4 Structure

The structure of this thesis unfolds as follows. In Chapter 2, an exhaustive exploration of computational ethology is presented, delving into key concepts and methodologies, with a specific emphasis on the application of pose estimation and behavioral segmentation techniques. This chapter not only reviews but also extends the lexicon associated with measuring animal motion. Chapter 3 introduces the foundational principles of deep latent variable modeling, elucidating the core concepts of VAE's. The subsequent chapter, Chapter 4, provides a detailed account of the methodology employed in developing the proposed approach. This involves presenting the experimental setup for data collection, applying mathematical and computer science concepts, RNN and latent variable modeling, and deriving the objective function of the VAME framework. The chapter further explores the application of graph theory to identify macro classes of behavior and outlines the creation of a benchmark dataset for method evaluation and comparison against existing approaches. Moving to Chapter 5, the principal findings are presented. The initial segment scrutinizes the efficacy of VAME in capturing nuanced distinctions between two groups of mice exhibiting Alzheimer's phenotype during free movement. The subsequent part applies the method to data from a head-fixed animal, integrating two-photon calcium imaging to fuse behavioral actions with neural activity. Here, an approach founded on information and graph theory is developed to harmonize both modalities. Chapter 6 critically examines and evaluates the results of the previous chapter. Finally, Chapter 7 concludes the research and offers insights into the future trajectory of computational ethology.

Chapter 2

Computational Measurements of Behavior

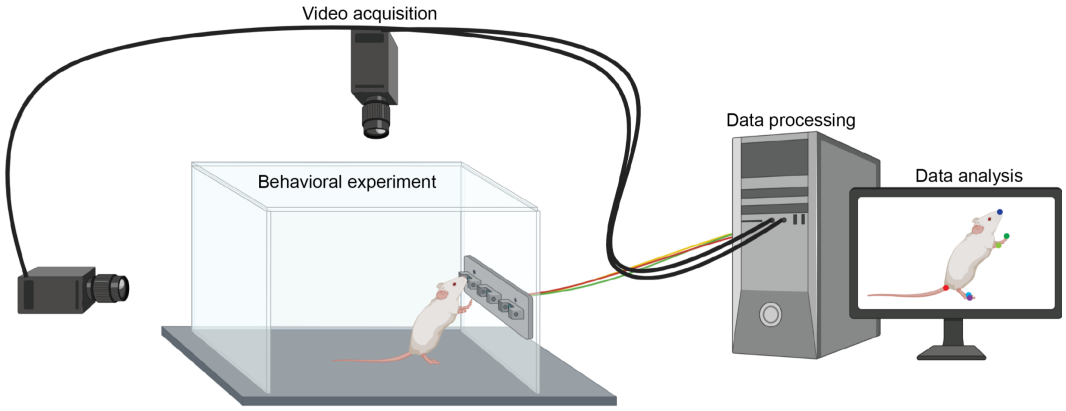

Figure 2.1: **Illustration of a behavioral experiment arrangement and its measurable results.** Above and beside a behavioral arena, cameras are installed to capture image sequences of an animal engaged in a behavioral task. The recorded footage is stored on a computer and subsequently analyzed using techniques like pose estimation and behavior classification. (modified from [24].)

A core principle of behavioral measurements is the identification of a representation of the behaving animal to study and analyze its behavior during an experiment. The usefulness of a particular representation is dependent on the ability to measure the relevant variables in an experiment, ideally in large amounts and at a low cost. The development of better cameras and computer vision algorithms has greatly increased the range of spatial and temporal scales that can be used to analyze behavior, as well as the range of environments in which recording is possible [25]. In this chapter, I start by introducing a general approach of a behavioral experimental setup and defining the characteristics of animal motion measurements by reviewing the language and vocabulary terms used in

the field and expanding this vocabulary. I continue to describe how current experimental techniques enable the precise measurement of behavioral kinematics through pose estimation methods and how recent behavioral quantification methods segment behavior into different categories. Finally, I identify the current limitations and state how this work aims to transcend these. This chapter mainly builds upon comprehensive articles that review the field of computational ethology from different perspectives, which offer in-depth information. [25, 26, 10, 12].

2.1 A modern behavioral recording setup

2.1.1 Measuring and tracking pose

As introduced above, Tinbergen and Lorenz thought of behavior as being constructed from simple actions that underlie fixed patterns, learned or innate. Through the further development of these concepts over the past decades modern computational ethology views behavior as stereotyped movements that make up the building blocks of more complex behavioral actions. This gives rise to the complete behavioral repertoire of an animal, encompassing dynamic changes in body posture over time [25]. To quantify the posture of an organism, it is common to start with an imaging experiment that captures a visual representation of the animal, such as a video (Figure 2.1). From this input, more abstract representations of the animal can be calculated using various methods. In the past, this often required manually annotating each frame of the video, which made it difficult or impossible to quantify posture in long videos with high frame rates. As a solution, techniques were developed that were inspired by human motion capture and involved attaching small, visible tracking markers to the animal's limbs [27, 28, 29]. However, this approach introduces potential issues, such as the continuous and potentially unnatural sensory stimulation of the animal. As we will see below, with improved computational power and algorithms, it has become increasingly feasible to track the position of animals, or even the coordinates of their limbs, with great accuracy and without the use of physical markers. Markerless methods have emerged in recent years and became the gold standard in neuroscience to track the pose of an animal [18, 19, 20, 30]. These methods utilize the power of deep neural networks to identify keypoints on the animal body

in the tracking video, which are defined by the researcher. A keypoint refers to a virtual marker, which is annotated by hand on the animal body in a video frame (Figure 2.1, right; colored marker on the mouse body). By annotating multiple body parts like the snout, limbs, and spine as well as different postures of the animal, a pose estimation model can generalize these keypoints and can label the full video automatically once trained. I introduce the specifics of this method further below. The virtual marker signal forms a time series of the postural changes of the animals body and is a lower representation of the video frames (Figure 2.4).

2.1.2 Extracting behavioral information

To extract behavioral information from the virtual marker signal, there are two common approaches (Figure 2.4). The first and straightforward approach is to apply a supervised model to the virtual marker time series [31, 14] or raw video signal [32] to identify behavioral classes. By predefining behavioral categories like Rearing, Walking and Grooming, a supervised model is in principle able to recover these actions throughout the video. Once properly trained, their advantage is that the model can extract behavioral categories from all the video data fast and with high accuracy. Their disadvantage, though, is that these models do not capture the full dynamical spectrum of behavior, do not recognize new or unseen behavior and have no latent embedding, which would compress behavioral information and make it easier to combine with neural activity for correlation or joint representations.

Reaching the goal of maximizing information content from the raw behavioral signal requires a complete capture of observable motion and its objective interpretation rather than pre-construct behavioral classes. Unsupervised methods provide a gateway for this purpose as they do not rely on human annotations. They are able to uncover complex dynamics of behavior at a very fine-grained level, which may not be discernible using other methods [33, 9, 22, 26, 25, 10]. By applying unsupervised methods to the virtual marker coordinates, the algorithm is able to cluster the information into behavioral patterns and find a suitable embedding for the motion (Figure 2.4). Depending on the type of unsupervised algorithm, this can be based on hand crafting features from the virtual marker coordinates like velocities, distances, or frequencies within and between virtual markers to uncover patterns [34]. Such methods usually do not explicitly model the dynamics of

the behavior. By explicitly modeling the temporal dynamics, it is possible to gain a more profound comprehension of how the behavior is structured and uncover crucial information regarding the underlying mechanisms that might influence the behavior, which may not be evident from static measures of behavior or frame-by-frame differences. Extracting the underlying patterns and dynamics from a multivariate time series of virtual marker positions remains a key challenge. This work addresses this challenge by developing a methodological framework that not only identifies behavioral patterns on different hierarchical scales but also embeds the information into a dynamical embedding space which can be used to analyze the behavior in various ways.

2.2 Characteristics of animal motion

It is essential to define a language for the quantification of behavioral measurements. Establishing a standardized vocabulary to articulate the size or quantity of physical characteristics in animals enables a more accurate and precise description and comparison of these measurements. This precision is crucial for ensuring reliable and consistent results. In laboratory settings, animals are often subjected to a restricted set of behaviors in a controlled environment as a way of measuring brain activity [35, 36]. An example of a restricted approach could be confining an animal to a maze where it can only turn left or right, or by limiting its head movement through head-fixing and providing a reward only when the animal licks in a particular direction or location. However, while these methods may take into consideration the animal's internal state or motivation for performing a behavior, they are rather unnatural and ignore much of the behavioral repertoire the brain was designed for or any other spontaneous movements that may occur. A recent study found that a significant portion of brain activity can be linked to spontaneous movements made by an animal under restricted conditions [37]. It highlights the importance of considering the full repertoire of an animal's movements in order to understand brain functions. The desire for repeatable measurements and high-throughput data collection is an essential part for statistical significance, and it often drives the use of methods such as mazes or head-fixing in order to record neural activity. However, this approach can result in measuring behavior that is overly restricted and not representative of the animal's typical range of actions. Previous research that focused solely on the correlation

between a few restricted actions may not have provided a complete or accurate picture. By introducing a vocabulary tailored to the new era of computational measurements of behavior, along with implementing innovative methods that leverage advancements in machine learning, computer vision, and pattern recognition, coupled with novel recording techniques for neural activity in freely behaving animals, we have the potential to address and overcome these challenges.

2.2.1 Phenomenological behavioral modeling

2.2.1.1 Behavioral representation and naturalistic behavior

We first need to define the term *behavioral representation* as it is commonly used but may carry varying meanings depending on the specific research context. A representation is "the way someone or something is shown or described" according to the Cambridge dictionary. Here, I define a behavioral representation (or description) as the way of quantitatively describing the actions or movements of an animal during an experiment. It can be used to distil or summarize any aspect of the animal's behavior, whether it is a simple action or a more complex movement. These representations can take many forms, from a classical ethogram, which is a detailed catalogue of all the different behaviors an animal can exhibit, to a low-dimensional plot that captures the animal's movement through space over time. The goal of such a representation is to provide a clear, concise, and quantitative summary of an animal's behavior that can be used to make predictions or draw conclusions about the animal's behavior or future motions.

The next term that shows up frequently is *naturalistic*. In its first approximation, naturalistic refers to behaviors that are similar to those observed in the animal's natural habitat. These behaviors include activities such as exploring new environments, obtaining food, finding shelter, and identifying mates. The term "naturalistic" is often used to distinguish these behaviors from those that are artificially induced by researchers through training the animal to be an expert at a given task, or those that are constrained by e.g. head-fixation. In neuroscience, naturalistic behaviors are often studied to understand how animals behave in an open-field arena environment, sometimes with simultaneous neural activity recording in form of local field potentials or multi electrode arrays, which

can help researchers understand how the animal's brain processes and responds to different stimuli. Those behaviors are thought of as self-motivated and expressed freely, which makes them more representative of the animal's natural behavioral repertoire.

2.2.1.2 Behavioral motif and sequences

To describe the actions or sub-movements an animal can express, the term *behavioral motif* is often applied. A behavioral motif is a unit of movement that is stereotyped and repeated [10]. The term "motif" does not have an exact definition that specifies the spatial or temporal scale at which the unit of behavior is organized. The terms "action" and "behavior" are also used to refer to collections of units of behavior, but there is no clear distinction between these terms. Some researcher have proposed a taxonomy that differentiates between different levels of movement, where a "moveme" is the simplest movement associated with a behavior, an "action" is a sequence of movemes, and an "activity" is a set of movemes and actions that is characteristic of a particular species [9]. Throughout this work I will refer to these types of actions generally as motifs.

Having defined behavioral motifs and actions, we can now specify another term linked to their temporal arrangement: a *behavioral sequences*. A behavioral sequence is a period during which an animal or organism expresses multiple behavioral motifs. The motifs that make up a sequence can be analyzed and their specific order can be determined. For example, motif A always follows motif B, which would represent a deterministic sequence. A second example would be motif A follows motif B in 50% of the time, which would represent a more random order and be called a probabilistic sequence. Behavioral sequences can be used to study the organization and coordination of different behaviors in an animal or organism and can provide insights into the underlying neural mechanisms that control these behaviors when paired up with neural activity measurements.

2.2.1.3 Behavioral communities

In this thesis and the accompanying publications, I provide an additional definition for behavioral sequences that are composed of sub movements of macro behaviors such as walking, rearing or grooming. I refer to these sequences as *communities*, which are conceptually borrowed from network theory [13, 38]. The communities are identified by

constructing a directed graph from the complete behavioral motif sequence and then identifying subgraphs on this graph. Subgraphs are groups of nodes on the graph that are densely connected to each other but have only a few connections to other nodes. A community is represented by such a subgraph. With these communities, we can identify hierarchies of behavior. The communities are identified by merging densely connected nodes together. I will introduce this in more detail in the subchapter 4.3. An advantage of representing behavioral motifs as communities is that it allows for the dynamical embedding to be converted into a discrete network representation, which can be used to study the transition and usage of motifs within the network as well as the intra- and inter-community transitions. This can provide valuable insights into the building blocks of behavior and be easily visualized. Additionally, by comparing networks between groups of animals, it can be easier to identify differences or to focus only on the relevant communities for a given experiment.

2.2.2 Computational behavioral modeling

2.2.2.1 Dimensionality reduction

In the context of this thesis and similar research, a crucial concept for identifying a robust representation of animal motion involves considering the notions of *dimensionality* and *dimensionality reduction*. These concepts play a critical role in many forms of machine learning and neuroscience. Dimensionality refers to the number of variables that describe a phenomenon or dataset. For example, in the case of animal motion, a large number of variables would be needed to consider all aspects of why a limb moves, such as the biomechanics of the movement, the different muscles and nerve cells involved, the environment, and the animal's internal state. However, it is not feasible to measure all these variables, so researchers often make constraints on their measurements and methods. One way to reduce the dimensionality of a dataset is to capture the behavior using a camera. This reduces the dimensionality to the dimensions of the video frame, which usually includes two spatial dimensions (height and width) and one temporal dimension (time). However, this is still a large amount of data to process, and it can be challenging to extract meaningful information from it. To further reduce the dimensionality, we can apply keypoint extraction methods. These methods allow to identify specific landmarks

on an animal's body, such as joints, that are relevant to the behavior being studied. By extracting these keypoints, the dimensionality of a single limb can be reduced to just two variables: the x and y coordinates of the keypoint. This makes it easier to track the movement of a limb over time. By reducing the dimensionality of the dataset, it also becomes easier to extract meaningful information and to compare the behavior of different animals. Dimensionality reduction also helps to visualize the data in a more intuitive way.

2.2.2.2 Behavioral state space

By introducing the dimensionality of an animal's representation it is now essential to define a *behavioral State Space* $\mathbb{B}$, as this is the mathematical framework in which the dimensionality reduced behavioral data is represented. A point in this space corresponds to a specific animal posture or motif. The concept of a state space is borrowed from mathematical state spaces, which are used to represent systems that change over time. It allows for the representation of the dimensionality reduced behavioral data in a mathematical framework, making it possible to study how an animal's behavior changes over time and to compare the behavior of different animals. Within the literature in the realm of neuroscience and computational ethology, different names are used to refer to $\mathbb{B}$. It is sometimes referred to as the embedding space or behavioral map. The term "embedding space" usually refers to the process of reducing the dimensionality of a dataset, which is assumed to exist on a high-dimensional manifold and collapsing it into a lower-dimensional space where it is more easily visualized or analyzed. This can be done using mathematical techniques such as principal component analysis (PCA) for linear embeddings or t-distributed stochastic neighbor embedding (t-SNE) or uniform manifold approximation and projection (UMAP) for non-linear embeddings [39, 40]. The term "behavioral map" is usually used to refer to a two-dimensional visualization of the dimensionality-reduced dataset. This visualization can be useful for identifying patterns or clusters in the data, which can provide insights into the underlying behavior of the animal. It is worth noting that the terms "embedding space" and "behavioral map" are used interchangeably in the literature, and the choice of terminology often depends on the specific context or application. Both terms refer to the process of reducing the dimensionality of a dataset in order to make it more amenable to analysis and visualization, but

the "embedding space" term tends to focus more on the mathematical and computational aspect, while "behavioral map" term is more focused on the visualization aspect.

2.2.2.3 Supervised and unsupervised learning

Two terms that are commonly encountered in computer science but also recently in the fields of neuroscience and computational ethology are *supervised* and *unsupervised* learning. Supervised learning is a type of machine learning algorithm that is trained on a labeled dataset, where the desired output or label is provided for each input data point. The algorithm learns to make predictions or classifications based on the patterns it discovers from the labeled data. Some examples of classical supervised learning algorithms are linear regression, decision trees, random forests, support vector machines, or linear neural networks. These algorithms are trained by minimizing a cost function, which measures the difference between the predicted output and the true label. Once the algorithm is fully trained, it can be used to make predictions on new, unseen data. In the field of neuroscience and computational ethology, supervised learning algorithms are used for a variety of tasks, including classifying behavioral labels, such as grooming or epileptic episodes, identifying specific features of an animal's behavior, such as movement patterns, and tracking the movement of animals over time. These algorithms can also be used to classify different types of behaviors, such as social interactions, foraging, or predator-prey interactions. Supervised learning algorithms are particularly useful when only a limited number of behaviors are of interest, as they can be trained on a dataset that includes labeled examples of those specific behaviors. Unsupervised learning, on the other hand, is a type of machine learning algorithm that learns the structure of the data without the use of labeled examples. These algorithms discover patterns and features in the data that are not explicitly provided. The most common unsupervised learning algorithms are k-Means, hierarchical clustering, and PCA. These algorithms are trained by minimizing a cost function, which measures the similarity or dissimilarity between the data points. In the field of neuroscience and computational ethology, unsupervised learning algorithms are used to identify patterns and features in animal behavior that may not be immediately obvious. These algorithms can be used to identify patterns of movement in animals, such as changes in speed or direction, or to identify clusters

of similar behaviors. Unsupervised learning algorithms can also be used to extract features from high-dimensional datasets, such as videos of animal behavior, that can be used to train supervised learning algorithms. Another class of unsupervised algorithm is self-supervised learning (SSL), which, in principle, is interchangeable with unsupervised learning, but the main difference is that commonly SSL algorithms uses artificial neural networks (ANN) as architecture (Note, however, that these are only modern naming conventions and can change from literature to literature). SSL algorithms learn by using the data itself as supervision, without requiring external labels. These algorithms are trained by providing an ANN with input data and a corresponding task, such as predicting the next frame of a video or reconstructing an image. The network learns to perform the task by discovering useful representations of the data, similar to how unsupervised learning algorithms discover patterns and features in the data.

In the context of machine learning, we need define the concept of a *feature*. Features are generally used to train a machine learning model. They represent certain characteristics of the data and are either selected by hand or inferred from the data via dimensional compression. In the light of behavioral analysis, a feature refers to a specific aspect or characteristic of an animal's behavior that can be quantified and used as a building block for further analysis. Examples of features could include the position and movement of an animal's body parts, the duration of certain behaviors, or the frequency of reoccurring movements. The selection and use of appropriate features is crucial for accurately and effectively quantifying and analysing behavior. Different behaviors may require different sets of features for accurate analysis, and the ability to identify and extract relevant features is a key component of any behavioral analysis method.

2.3 Tracking and pose estimation

2.3.1 Ellipse tracking and background subtraction

To quantitatively describe behavior, we need to observe and track the movements of an animal. This involves using computational tools to extract details of the animal motion from video recordings. A general method is to track the position of the animal's center of mass (or centroid) over time. This can be done by measuring the centroid location as

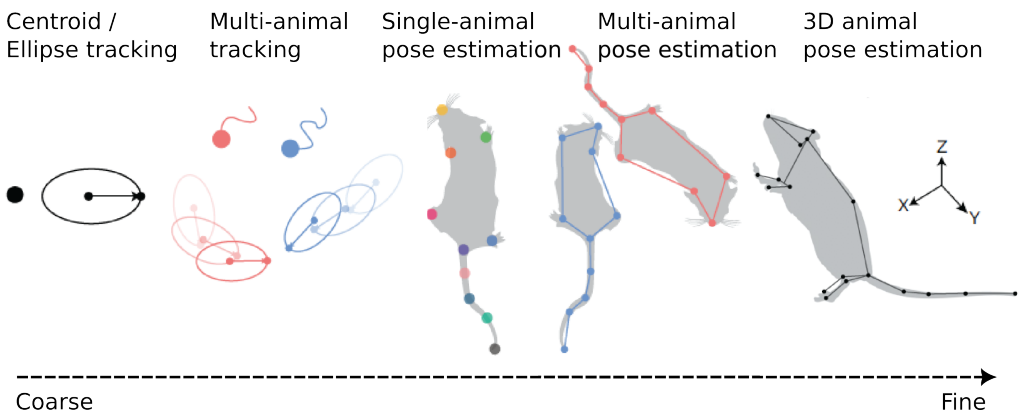

Figure 2.2: **Different forms of tracking methods, from coarse to fine (modified from [12])**.

a series of image coordinates, which gives information about the animal's motion direction and spatial navigation. However, this approach only provides a rough estimate of the animal's behavior, as it does not consider its orientation. To capture this information, we can augment the centroid measurements by finding the major and minor axes of an ellipse that encloses the animal (4.1, left). Traditionally, centroids and ellipses have been calculated using background subtraction, a process where the pixels in the image that belong to the animal (foreground) are identified and the centroid is calculated by finding the midpoint of these coordinates. When the background has high contrast with the animal, like in experimental chambers with a background illumination, background subtraction can be done by simply thresholding the image intensity. If the background is static, it can be modeled using the median image frame and subtracting the median image frame from the other frames. However, this approach can be problematic when the animal is stationary for a long time. While classical methods use robust algorithms to model the background, newer methods have started using deep learning to better handle more complex backgrounds and allow for tracking animals in more realistic conditions [41, 42]. Background subtraction is a straightforward technique for identifying animals in an arena, provided the setup is uncomplicated. However, this method can be prone to failure due to differences in lighting or camera angles, and may require adjustments to the threshold values to work effectively.

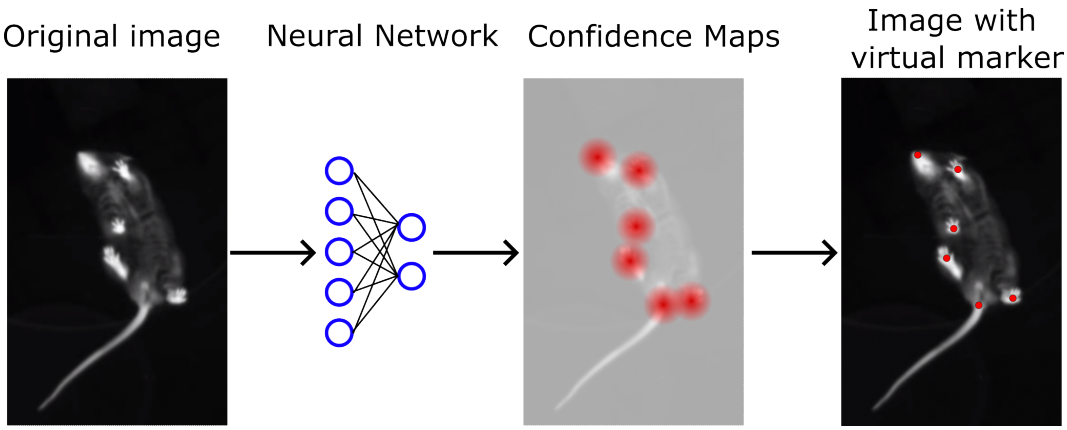

Figure 2.3: **Pose estimation principles.** In the context of single-animal pose estimation, a convolutional neural network is employed to forecast a confidence map for each body part type using an input image (left). The network is trained to generate confidence maps exhibiting a single peak per channel (middle), and the decoding process involves identifying the overall peak in each channel of the confidence map (right).

2.3.2 Animal pose estimation

While centroid and ellipse tracking provide information about the navigation of an animal and locomotion variables like velocity, acceleration and angular velocities, they are unable to capture the movements of specific body parts or its posture and therefore cannot be used to detect specific behavioral patterns like grooming, rearing, and coordination during a walking movement. Animal pose, on the other hand, involves identifying the location of landmarks on the animal's body (usually easily identifiable landmarks like limbs, paws, snout or body center) (4.1, middle). Pose estimation captures almost all of the ways in which an animal's body can move, and with that, the movements that the brain can control through the motor system [12]. In its essence, pose estimation is the process of determining the positions of an animals (or persons) body parts and has been a topic of research for a long time. It has been studied from a biological perspective, which involves understanding how humans and animals perceive the movement of other organisms, as well as from an engineering perspective, which involves creating algorithms to accurately determine pose from video. The biological perspective helps to understand why pose is an important representation of behavior, particularly in social situations, while the engineering perspective has made it possible to automatically estimate pose from video.

2.3.2.1 General principles

The recent success of deep learning and computer vision has sparked the development of a variety of pose estimation tools in humans and paved the way to further develop them for the purpose of animal pose tracking. These methods drastically improved the accuracy of previous methods. As more video data of animals in experimental setup's becomes available and graphical processor unit (GPU) technology becomes more widespread, researchers have the opportunity to utilize modern pose estimation techniques in their own labs with ease. Owing to the effectiveness of deep neural networks combined with GPUs, there has been a recent spark of open-source tools for pose estimation [18, 20, 19]. In general, all these methods are using convolutional neural networks (CNNs) [43] to generate heatmaps for the representation of a landmarks location i.e. annotated keypoint (Figure 2.3) [44, 45, 46]. With this method, the location of each landmark is encoded as a two-dimensional Gaussian distribution, or heatmap, centered on the true coordinates of the landmark in the image. The heatmap is a single image in which the pixel intensity is highest at the location of the landmark. This representation is well-suited for use with CNNs because they are able to learn complex transformations of image patches. To train a pose estimation CNN, labeled examples with known ground truth landmark coordinates are used to generate the correct heatmaps, which are then compared to the CNN's predictions. Once trained, the CNN can predict heatmaps for unlabeled images, and the landmark coordinates can be decoded from the predicted heatmaps by identifying the peak intensity in each heatmap.

The most significant obstacle to using this approach for estimating animal poses is the large amount of training data required to learn the model. While the computer vision community has collected millions of labeled examples of human poses to train and refine CNN models for human pose estimation [47, 48, 49], these models cannot simply be transferred to animal pose estimation. Instead, a new model would need to be trained specifically on animal pose data from scratch, which may be more difficult to obtain in sufficient quantities. One way to address this problem is the use of transfer learning.

2.3.2.2 Transfer learning versus efficient neural network design

Transfer learning is a technique in deep learning that allows a model trained on one task to be used as a starting point for a model trained on a related task. For example, if a model is trained to recognize certain objects in images, that model might be able to serve as a good starting point for a model that is trained to recognize other objects. The same goes for pose estimation from humans to animals. The idea behind transfer learning is that many of the features learned by the original model will be useful for the new task, and so the new model will be able to learn the task more quickly and with fewer training examples, which is key to its success. A common approach is to use the weights of a pre-trained model as the initial weights for a new model. In a CNN this would be the backbone of its visual feature detectors, typically trained on the ImageNet dataset [50]. This approach is based on the idea that if we can decrease the requirement for learning general-purpose visual characteristics, like textured patches and oriented edges, it will be easier to fine-tune the network's parameters with less training data. By freezing the backbone and keeping the weights fixed, it is only necessary to fine-tune the projection head which learns to identify the keypoints based on the heatmap localization in an animal dataset. A main contribution and widely used approach is the open-source method DeepLabCut (DLC), which applies transfer learning to identify keypoint based animal posture [18].

Although being successful in studying a wide range of animal species and behavior, such as mice, zebrafish and flies [30], it is worth mentioning that transfer learning comes with the drawback of utilizing a heavy network with many parameters, which might be unnecessary and only increases computational costs. By designing an efficient neural network and keeping the architecture of the CNN small, it is possible to train a model from scratch in a shorter amount of time. This is because the model will have fewer parameters to train compared to a general-purpose architecture used in transfer learning. This reduction in network size assumes that the variability of imaging conditions in animal behavioral data is relatively low, which is a characteristic of reproducible laboratory experiments. As a result, a lower level of representational capacity is needed. The main contributions in this direction have been made by the open source methods SLEAP and DeepPoseKit [19, 20].

The success and ease of use of these methods have led to a new standard in fields that study animal movement, including neuroscience [51] and ecology [52]. However, it is important to note that, while transfer learning and efficient neural network design-based tracking methods are commonly used in computer vision and neuroscience to track animal behavior, they rely on human-provided annotations and concepts about which features or body parts of the animal are important. Other methods, such as principal component analysis of raw images, can be used to identify the most significant parts of the animal that change during behavior. Newer research has attempted to identify key-points in an unsupervised manner, which eliminates human preconceptions in selecting areas of interest [53].

2.4 Behavior classification and learning from data

Behavior is a dynamic and complex phenomenon that encompasses a wide range of actions that an animal exhibits over time. While tracking the parts of an animal's body can be relatively straightforward, quantifying the temporal structure of behavior is often challenging as it can lack clear standards or ground truths. To better understand behavior, it is often divided into a sequence of discrete behavioral states, such as "walking", "rearing", "eating" etc. These states are defined based on observable characteristics, such as the position and movement of body parts. The advancements of machine learning techniques has made it possible to classify these states from video or tracking data, making it possible to automate the process of behavioral quantification. This can facilitate comparisons between different instances of a behavior, and allows for the generation of hypotheses about the neural circuitry that underlies them [54, 55, 56].

2.4.1 Rule and label based classification

Defining a behavior can be done in a straightforward manner by creating a set of clear and definite rules that detail the conditions necessary for the behavior to be considered present or occurring at a specific moment. An initial approach to defining a behavior could be as simple as categorizing instances of locomotion when the animal's central point is moving at a speed greater than a pre-determined minimum threshold. However,

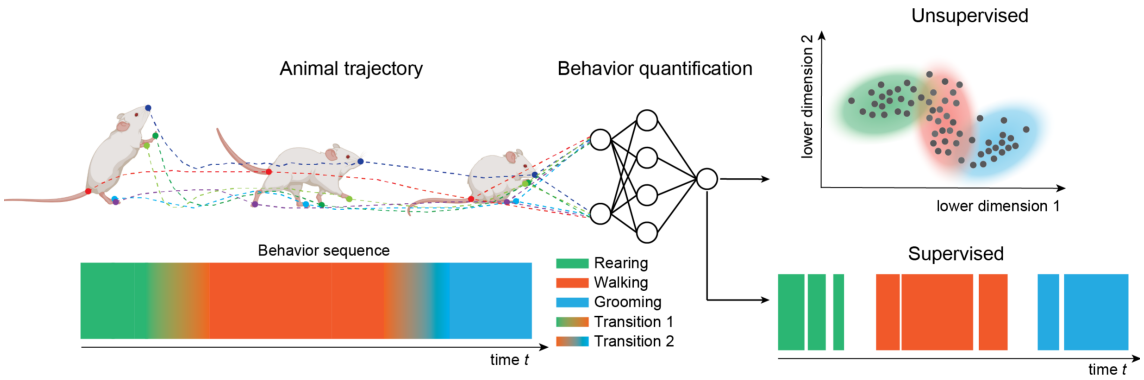

Figure 2.4: **Illustration of a contemporary setup for a behavioral experiment along with its measurable results.** The trajectory of animal poses captures crucial kinematic information and serves as input for quantifying behavior. Below the trajectory, the corresponding behavioral sequence is displayed, featuring three exemplary behaviors (Rearing, Walking, and Grooming) along with two transitional states between behavior classes. The behavior quantification method can be either unsupervised, involving the learning of an embedding for state or cluster identification, or supervised, aiming to classify trajectories based on human-annotated labels (modified from [24].)

defining a behavior can become more complex when requiring precise inclusion and exclusion criteria based on intricate postural feature descriptions. While fixed rules for defining behaviors can provide ease for evaluation and interpretation, they do not capture the entire range of expressions an animal can exhibit, especially when it is subject to experimental manipulations that can alter the statistical features used for classification.

An approach that balances the human definition of behaviors with computer-aided classification is the use of supervised machine learning (Figure 2.4). In this method, users provide examples of instances when certain behaviors are present or absent, and the machine learning algorithm derives classification criteria based on specific features such as body-part positions or speeds extracted from raw data. Popular toolkits use decision trees or random forest ensembles to learn complex or abstract classifiers from animal tracking features [31]. This approach leverages the data to avoid the laborious and potentially error-prone manual design of classification criteria, while also providing measures of accuracy.

Supervised machine learning methods have significantly enhanced the consistency and reduced the manual labor needed for the analysis of behavioral data through user-specified categorization of behavior. Despite these advancements, the underlying challenges persist - the characterization of behavior is either broad or based on the subjective

perception of a human observer, which includes underlying assumptions about the animal's behavior that are encoded explicitly. Moreover, studies have shown that there is a significant degree of disagreement among experts, even when clear guidelines for annotating a behavior are provided. This highlights the limitations of relying solely on human definitions of a behavior. The subjective nature of human definitions can lead to inconsistent interpretations and annotations, making it challenging to accurately and consistently identify and classify behaviors [57, 58, 59].

2.4.2 Learning patterns from data

The remainder of this work focuses on the approach of directly learning behavioral patterns from the data without using human definitions or rules. As previously mentioned, these types of algorithms are generally referred to as unsupervised learning techniques. The underlying assumption across these methods is that data belonging to the same state show similar, stereotyped dynamics based on some measure of similarity. This assumption forms the basis for these techniques, which use the statistics of the behavioral time series to identify distinct clusters or states [25, 26].

Assuming that a behavioral sequence can be represented as a series of short, non-overlapping trajectories through the space of behavioral observations, one approach for mapping these units is to extract the trajectories through temporal segmentation and then cluster them into discrete groups based on similarity. However, clustering in high-dimensional space can be difficult, so the behavioral dimensionality is often reduced using techniques such as PCA or nonlinear embedding algorithms such as t-SNE, UMAP, or Isomap [60]. Clustering algorithms aim to group time points into discrete sets, based on their similarity, such that each time point is more alike to the members within the set than to those outside of it (Figure 2.4). This approach to mapping behavioral patterns may seem straightforward, however, the complexity and variability of natural behavior can pose a challenge. This is because different instances of the same stereotypical behavior are unlikely to have identical trajectories in terms of shape and timing, and there may not always be a clear boundary between one movement pattern and the next. To effectively segment and cluster the fundamental units of behavior, a good model should be able to handle variations between realizations of the same behavior, while also being sensitive enough to identify differences between behaviors that serve distinct functions.

2.4.2.1 Behavioral mapping based on t-SNE

The MotionMapper approach [21] is a popular method for unsupervised behavioral mapping, which was initially developed to study the free walking behaviors of Drosophila melanogaster from high-speed video recordings. Unlike other methods that aim to directly segment a postural time series, MotionMapper uses a continuous wavelet transform to encode short-time trajectories, obtained from PCA projections of the recordings in the frequency domain by retaining only amplitude information and ignoring phase [61]. This encoding makes the approach robust against small temporal misalignments. To cluster the data into stereotyped behavioral patterns, the MotionMapper approach smooths a two-dimensional t-SNE embedding and then uses a watershed algorithm. To tackle computational challenges posed by large data sets, the authors used a variant of t-SNE that learns a low-dimensional embedding for a representative subset of training data and maps the remaining points onto the embedding in an additional step.

2.4.2.2 Clustering versus representation learning

The interpretation of behavioral clusters will vary depending on the specific application. Here, clusters refer to groups of points that are self-similar, but do not necessarily specify what distinguishes one cluster from another. One way to understand such clusters is through a qualitative observation of raw data examples from each group, such as video footage. This visual inspection may show that one cluster corresponds to locomotor behavior and another to grooming, but it may not provide a clear reason for the differences between two clusters of locomotor behavior. To interpret clusters, the empirical feature distribution of the data per cluster can be used. This can reveal differences in forms of locomotion based on factors like peak frequency of limb oscillations, but it may become more challenging to interpret when the differences are small or when the input data is high-dimensional. While clustering can be effective when there is limited knowledge about the structure of the behavioral dynamics, more advanced approaches allow for explicitly modeling the characteristics that define the representation of behavioral dynamics. Such methods can be more interpretable through direct examination of the model parameters, or by generating new examples from the model. One direction is to use state-space models which build on probabilistic graphical models. Instead of

trying to divide the data based on its similarity, these models posit the existence of unobservable (hidden) discrete states that parametrize the processes underlying the data. The method developed in this work utilizes state-space modeling through the implementation of Hidden Markov Models (HMM) to parametrize a continuous embedding distribution. Although these models need more amounts of data for robust fitting of their parameters and careful approaches for model selection and regularization, they enable simultaneous global fitting of the data to automatically identify behavioral structure and label the time series accordingly.

2.4.2.3 Embedding depth image dynamics

A method with these properties was introduced in 2015 to identify stereotyped behavior in mice, called MotionSequencing (MoSeq) [22]. They use an autoregressive Hidden-Markov Model (AR-HMM) to study inherent structure in mouse pose dynamics of freely moving mice from depth images (KinectV2). By reducing the dimensionality of egocentrically aligned depth images with the use of PCA, a time series of principal components is extracted. The output is then fed into an unsupervised machine learning model called AR-HMM to identify the underlying structure of mouse behavior. In this work, the authors presented a two-tier data description, where behavioral patterns represent short-term postures recorded through a series of continuous autoregressive processes. An HMM then outlines the sequence of transitions between these patterns. The AR-HMM has the ability to depict behaviors of varying durations, but a "sticky" timescale parameter focuses the model on behaviors of a specific time frame. In their study, MoSeq identified around 60 unique behavioral units from mouse movement recordings, accounting for over 95% of the data. Additionally, using the probability distribution of the units as a summary statistic was found to be effective in distinguishing between neuroactive and psychoactive drugs in drug discovery experiments.

2.5 Transcending limitations

It is evident that advancements in machine learning and computer vision have greatly impacted our capacity to uncover increasingly precise descriptions of behavior, from

tracking to dynamics. Despite recent advancements in unsupervised computational methods for analysing naturalistic behavior, there are still many technical and theoretical difficulties to overcome [62]. This work focuses explicitly on bringing new advances to the realm of behavior segmentation and embedding with a focus on mice as a model animal. The extraction of behavioral motifs and the embedding of dynamics from pose estimation methods continues to present a significant challenge in the field.

2.5.1 Uncertainty of current methodologies

The spectral energy of a signal is the primary input feature for MotionMapper, but its effectiveness in capturing the full behavioral repertoire is limited by low frequency movements, which are more prominent in mice than in flies. It is particularly effective in detecting the movement of orthogonal limbs, such as fly appendages. MoSeq was initially used to study freely moving rodents from depth camera images, enabling the detection of sub-second behavioral structure. However, the underlying AR-HMM model can lead to many fast-switching and short motifs, causing uncertainty in animal action classification. This highlights a broader issue concerning the scale of behavioral extraction, which is critical for understanding the action and kinematics, especially in the context of studying various disease states. This also raises concerns about the generalizability of these methods. The specificity of the applications of MotionMapper and MoSeq, which rely on the spectral energy of a signal or the AR-HMM model respectively, can limit their effectiveness in capturing the full behavioral repertoire, leading to uncertainty in animal action classification and sub-optimal results when applied to virtual marker time series. Therefore, while these methods have shown effectiveness in their specific applications, their lack of generalizability may be a disadvantage when studying complex behaviors from pose estimation signals in contexts where a more comprehensive understanding of the action and kinematics is needed. In light of these limitations, I argue that new and innovative approaches are required to provide a reliable and robust solution for uncovering the underlying latent states and behaviors encoded in a lower-dimensional subspace or manifold. The current unsupervised methods, however, are not adequate in capturing the complete spatiotemporal dynamics of behavior.

2.5.2 Surpassing conventional depictions

I aim to challenge the standard representation of behavior, which is the behavioral map, and commonly thought of as consisting of clear-cut, separate behaviors. However, the reality is that the execution of movements is a result of a continuous flow of motion. For instance, while distinct motor commands are needed for activities like walking and running, the difference between slow and fast walking could only be the pace of the stride cycle, with a gradual transition to a moderate walking speed. A comprehensive representation of behavior should therefore reflect both the discrete separations between behaviors and the smooth variations within them. Current methodologies have not yet fully achieved this or have only been able to accomplish it partially.

2.5.3 Projecting animal motion into a dynamical embedding space

A major aspect I want to address in this thesis is the utilisation of recent advancements in deep learning and latent variable modeling to develop a method that can project animal motion patterns into a dynamical and continuous embedding space. This method must enforce spatiotemporal similarity and have the ability to uncover discrete behavioral motifs within the space. Such a model would have the potential to overcome the limitations of the previously discussed methods, which lack both these capabilities. By applying constraints on the distribution of representations, the aim is to foster the capture of more meaningful and understandable quantities within the dynamical landscape of behavior. These constraints are referred to as variational constraints and serve to encourage the inclusion of interpretable information in the representations. I intend to evaluate the performance of this new method (VAME) against the other two methods using a benchmark dataset. Moreover, I compare the mapping of MotionMapper with the mapping of VAME latent vectors. To understand and analyze the behavioral structure learned by VAME, I aim to utilize methods from information and network theory. This will involve discretizing the continuous latent space embedded by VAME into a network structure to analyze patterns and identify subgraphs that form communities of behavioral patterns. Finally, by applying information theory, I aim to connect neural activity and behavioral patterns, and to explore their relationship and interactions.

Chapter 3

Deep Latent Variable Modeling

3.1 Latent variable modeling

Computational ethology is an active area of research that aims to understand the complex patterns and structures present in behavioral data. A challenge arises from the fact that conventional approaches to modeling this data often face limitations in capturing the comprehensive complexity of the information [10]. Recent advancements in the field of machine learning have led to the development of deep generative models, which combine elements of probabilistic modeling and deep learning (see Appendix A.1 for a introduction to deep learning). These models are designed to learn the underlying structure of complex real-world data and can be used for a variety of tasks such as finding patterns, clustering data, and identifying statistical correlations. Additionally, deep generative models can generate new data that is similar to the original data based on the learned distribution, which can be used to validate the model.

Two important examples of deep generative models are Generative Adversarial Networks (GANs) and VAEs [63, 64]. GANs are trained to generate new data from a noise distribution, while VAEs are optimized to learn latent embeddings from the input data. VAEs have several advantages over GANs that make them a more powerful option for modeling complex real-world data. One of the main advantages of VAEs is that they explicitly aim to learn latent embeddings from the input data [64]. This means that VAEs are able to identify and extract the underlying structure of the data, which is crucial for understanding complex patterns and relationships. Another key advantage of VAEs over GANs is their ability to model a probabilistic distribution. VAEs are, in principle, able

to model the full data distribution, which allows them to generate new data that is similar to the original data based on the learned distribution. This makes VAEs a powerful tool for tasks such as data generation and anomaly detection. However, GANs are also making progress in this area with the introduction of new techniques such as Adversarial Feature Learning and Adversarial Latent Inference [65, 66]. Albeit GANs are known to be difficult to train, and can be unstable and prone to generating low-quality or unrealistic samples. VAEs, on the other hand, have a more stable training process, which makes them more robust and reliable for modeling complex data. Therefore, this work focuses exclusively on the VAE model.

3.1.1 Modeling the posterior distribution

A major challenge in machine learning is to accurately estimate a complex probability distribution $p(x)$ using only a limited set of data points. This is particularly difficult when the data points are high-dimensional. In the case of virtual markers from the animal pose estimation, the distribution would need to model the intricate relationships between all markers over time that make up the full motion and behavior of an animal. Attempting to model this distribution directly is a difficult task and may be impossible to achieve within a reasonable amount of time. This is because the complexity of the distribution increases rapidly as the dimensionality of the data increases, making it challenging to accurately estimate $p(x)$.

3.1.1.1 Bayesian likelihood and prior distribution

Latent variable models are a type of machine learning model that aims to learn $p(x)$ of a given data set. The main idea behind these models is to introduce a latent variable z, which can be thought of as a hidden feature or characteristic of the data. This latent variable is used to define a conditional distribution, $p(x|z)$, which is known as the likelihood in Bayesian terms. This formalism makes the latent variable z a random variable, hence a measurable function $z : \Omega \rightarrow E$ that maps a set of possible outcomes Ω to a space E. In the context of animal movement, the latent variable z can contain the hidden dynamics of the animal's movements, as measured from the pose estimation signal. As we will see here, a VAE is an instantiation of such a latent variable model, that uses an encoder-decoder architecture to model the underlying distribution $p(x)$ of the data.

With introducing z, we can specify a prior distribution $p(z)$ over the latent variables. This prior distribution encodes our prior knowledge or assumptions about the data $\mathcal{X}$. With $p(z)$, we can compute the joint distribution $p(x, z)$ over both the observed and latent variables. The joint distribution is the product of the prior distribution $p(z)$ and the likelihood $p(x|z)$. The joint distribution represents a complete probabilistic model of the data.

$$p(x, z) = p(x|z)p(z) \tag{3.1}$$

The joint distribution $p(x, z)$ allows us to express the $p(x)$ in a more manageable way (Equation 3.2). This is because the components of the joint distribution, $p(x|z)$ and $p(z)$, are typically much simpler to define than the original probability distribution $p(x)$. For example, these distributions can be defined by using distributions from the exponential family, which include commonly used distributions such as Gaussian, Poisson and Exponential distributions. These distributions are chosen because they have closed-form expressions for the probability density function and have easy-to-compute moments, which makes them more tractable than other distributions. Additionally, the joint distribution can also be factorized into the product of $p(z)$ and $p(x|z)$ making it computationally more efficient to work with. This factorization allows for efficient sampling and inference algorithms to be applied to the model, making it more tractable to work with. To obtain the data distribution $p(x)$ we need to marginalize over the latent variables:

$$p(x) = \int_z p(x, z)dz = \int_z p(x|z)p(z)dz \tag{3.2}$$

Furthermore, using Bayes theorem, we can compute the posterior distribution $p(z|x)$, which represents the probability of the latent variable z given the observed data x. Bayes theorem states that the posterior distribution is proportional to the product of the likelihood and the prior distribution, which is mathematically represented as:

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} \tag{3.3}$$

This means that the posterior distribution is a function of the likelihood and the prior distribution, and it encodes our updated knowledge of the latent variable z given the observed data x. Hence, the posterior distribution allows us to infer the latent variables

given the observations.

3.1.1.2 Properties of latent variables

Latent variables are typically lower-dimensional than the observed input vectors, and this compression of the data is one of the key features of latent variable models. The intuition behind this is that the data has a lot of redundancy and noise, and the latent variables are used to extract the most informative and relevant features of the data. One way to think about the role of the latent variables is as an information bottleneck [67]. The information bottleneck theory is a framework that explains how a model learns to compress the data by reducing the dimensionality of the latent variables. The idea is that the model has to balance between preserving the information that is needed to generate the data and discarding the information that is irrelevant or redundant. The latent variables represent the compressed representation of the data that contains only the most relevant information. The manifold hypothesis is another important concept that is closely related to the information bottleneck theory. This hypothesis states that high-dimensional data, such as real-world data, lies on lower-dimensional manifolds embedded in the high-dimensional space [68]. This effectively means that the data can be represented by a lower-dimensional model. This justifies the use of lower-dimensional latent spaces in latent variable models, as they can capture the underlying structure of the data more effectively.

3.1.2 Variational lower bound

The posterior distribution $p(z|x)$ is a crucial component in probabilistic reasoning, as it updates our beliefs about the latent variables after observing a new data point. In practice, the posterior distribution for real-world data is often intractable, as there is no analytical solution to the integral in equation (3.2), which also appears in the denominator of equation (3.3). To approximate the posterior distribution, there are two main methods: Markov Chain Monte Carlo (MCMC) and Variational Inference (VI). MCMC methods such as Metropolis-Hastings, Gibbs sampling, and Hamiltonian Monte Carlo generate samples from the posterior distribution using a Markov Chain that has the target distribution as its equilibrium distribution. The samples from the Markov Chain can be used to estimate the posterior distribution. These methods are exact in the sense

that they converge to the true posterior distribution as the number of samples increases. However, they are computationally expensive and do not scale well to large data sets. Furthermore, they are also sensitive to the choice of starting points, and they can get stuck in local modes. On the other hand, VI is a deterministic approximation technique that seeks to find the best approximation to the true posterior distribution by minimizing the Kullback-Leibler divergence between the approximation and the true posterior distribution. VI methods are more efficient than MCMC, and they scale well to large data sets. Additionally, VI methods can handle multimodal distributions and allow for efficient online learning. However, VI methods are not exact, and they provide only an approximation to the true posterior distribution.

3.1.2.1 Variational Inference and Kullback-Leibler divergence

The mathematical formulation behind VI is to approximate the intractable true posterior distribution $p(z|x)$ with a tractable family of distributions, such as a multivariate Gaussian, represented by $q(z)$. The goal is to find the best approximation to the true posterior by minimizing the Kullback-Leibler divergence between the two distributions. The Kullback-Leibler divergence is a measure of the difference between two probability distributions and it is defined as:

$$KL[q(z)||p(z|x)] = \int q(z) \log \frac{q(z)}{p(z|x)} dz = -\int q(z) \frac{p(z|x)}{q(z)} dz. \tag{3.4}$$

The Kullback-Leibler divergence between the true posterior $p(z|x)$ and the approximation $q(z)$ is used as a loss function in VI. The idea is to minimize this divergence by adjusting the parameters of the approximation $q(z)$ so that it becomes as close as possible to the true posterior. By minimizing the Kullback-Leibler divergence, we ensure that the approximation $q(z)$ captures the most important features of the true posterior $p(z|x)$. The Kullback-Leibler divergence is not symmetric and it is non-negative. So in practice, we minimize the negative of the Kullback-Leibler divergence.

3.1.2.2 Approximating the intractable posterior distribution

The problem with equation (3.4) is that it still contains the intractable true posterior distribution $p(z|x)$. By decomposing the equation we will find the following:

$$
\begin{aligned}
KL[q(z)||p(z|x)] &= \int q(z) \log \frac{q(z)}{p(z|x)} dz && (3.5)\\
&= \int q(z)(\log q(z) - \log p(z|x))dz && (3.6)\\
&= \int q(z) \log q(z) - q(z) \log p(z|x)dz && (3.7)\\
&= \int q(z) \log q(z)dz - \int q(z) \log p(z|x)dz && (3.8)\\
&= \mathbb{E}_q [\log q(z)] - \mathbb{E}_q [\log p(z|x)] && (3.9)\\
&= \mathbb{E}_q [\log q(z)] - \mathbb{E}_q \left[\log \frac{p(x,z)}{p(x)}\right] && (3.10)\\
&= \mathbb{E}_q [\log q(z)] - \mathbb{E}_q [\log p(x,z) - \log p(x)] && (3.11)\\
&= \mathbb{E}_q [\log q(z) - \log p(x,z)] + \mathbb{E}_q [\log p(x)] && (3.12)\\
&= \mathbb{E}_q [\log q(z) - \log p(x,z)] + \log p(x) && (3.13)
\end{aligned}
$$

The intractable term $p(x)$ is also still present in equation (3.13). However, we can apply a trick here by rearranging the quantities in equation (3.13), where the intractable terms are on the same side:

$$\mathbb{E}_q [\log p(x,z) - \log q(z)] = \log p(x) - KL(q(z)||p(z|x)) \tag{3.14}$$

The idea is to maximize the left term since it contains only tractable solutions. By doing so we maximize the evidence $p(x)$ and minimize the Kullback-Leibler divergence between the approximated distribution $q(z)$ and the true posterior distribution $p(z|x)$. Since the Kullback-Leibler divergence is non-negative ($Kullback - Leibler \geq 0$) the left term becomes a lower bound over the log-evidence $p(x)$, which is also called the *Evidence Lower BOund* or short ELBO:

$$ELBO(q) = \mathbb{E}_q [\log p(x,z) - logq(z)] = \mathbb{E}_q \left[\log \frac{p(x,z)}{q(z)}\right] \tag{3.15}$$

To avoid minimizing the Kullback-Leibler divergence directly we maximize another

term with the ELBO, which is equivalent up to an added constant. The ELBO is also called a "variational lower bound" or "negative free energy". The next equations will unpack the ELBO further to achieve a composition that is easy to manage:

$$ELBO(q) = \mathbb{E}_q\left[\log p(x,z) - \log q(z)\right] \tag{3.16}$$

$$= \mathbb{E}_q\left[\log p(x,z)\right] - \mathbb{E}_q\left[\log q(z)\right] \tag{3.17}$$

$$= \mathbb{E}_q\left[\log(p(x|z)p(z))\right] - \mathbb{E}_q\left[\log q(z)\right] \tag{3.18}$$

$$= \mathbb{E}_q\left[\log p(x|z)\right] + \mathbb{E}_q\left[\log p(z)\right] - \mathbb{E}_q\left[\log q(z)\right] \tag{3.19}$$

$$= \mathbb{E}_q\left[\log p(x|z)\right] + \mathbb{E}_q\left[\log p(z) - \log q(z)\right] \tag{3.20}$$

$$= \mathbb{E}_q\left[\log p(x|z)\right] + \int q(z) \log \frac{p(z)}{q(z)} dz \tag{3.21}$$

$$= \mathbb{E}_q\left[\log p(x|z)\right] - KL(q(z)||p(z)) \tag{3.22}$$

The first term in equation (3.22) represents the likelihood of the data given the latent variable, and maximizing the ELBO maximizes this likelihood by selecting the best-predicting models in the variational family for the data. The second term is the negative Kullback-Leibler divergence between the variational model $q(z)$ and the prior distribution $p(z)$ for the latent variables. Maximizing the ELBO pushes this term towards zero, meaning the two distributions are made similar, with the variational distribution matching the prior.

3.2 Variational auto-encoding

3.2.1 Minimizing the objective function

3.2.1.1 Monte Carlo integration

The ELBO is a common choice for the objective function to be minimized in VI [70]. In practice, computing the ELBO is usually done via Monte Carlo integration. The expectation of the log-likelihood term is approximated by taking the average of the log-likelihood over a set of samples from the approximate posterior. Monte Carlo integration is a method for approximating the value of an integral by averaging the function over a large number of randomly sampled points [71]. The basic idea is to generate a large number of random samples from the target distribution and use the average of the function

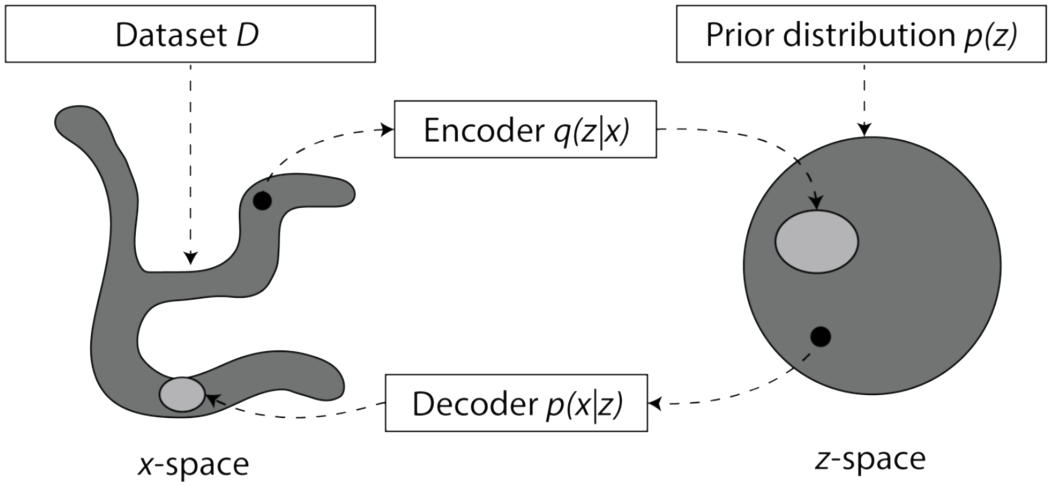

Figure 3.1: **Schematic illustration of the mechanism of a VAE.** The VAE learns a stochastic mapping from the original dataset space D, whose empirical distribution $q_D(x)$ is usually complicated, to a latent space z, whose distribution can be very simple. The generative model learns a joint distribution $p_\theta(x, z)$, which is factorized as $p_\theta(x, z)$. Here, $p_\theta(z)$ is a prior distribution over the latent space and $p_\theta(x|z)$ a stochastic decoder. The stochastic encoder $q_\phi(z|x)$ approximates the true (but intractable) posterior $p_\theta(z|x)$ of the generative model (modified from [69]).

evaluated at these samples as an estimate of the integral. The general equation for Monte Carlo integration is the Monte Carlo estimator:

$$\int f(x)dx = (b-a)\frac{1}{N-1}\sum_{i=0}^{N} f(x_i), \tag{3.23}$$

where x_i are N random samples from the distribution, $f(x)$ is the function to be integrated, and a and b are the limits of the integral. The more samples are used, the more accurate the approximation becomes. However, for some complex distributions, the Monte Carlo integration can be computationally expensive and converge slowly. While Monte Carlo integration can be used to perfectly match the target distribution, it has several drawbacks that limit its applicability to the problem in this work. First, it has a high variance, which means that the estimate can be highly variable and may require a large number of samples to converge to the true value. Second, convergence can be slow for complex distributions and high-dimensional integrals. Third, for some distributions, sampling can be difficult, for example, distributions with multiple modes or other complex structures. And lastly, Monte Carlo integration is not always applicable, for example, if the function is not computationally tractable or if the integral has infinite limits.

3.2.1.2 Gradient based optimization

Another solution for finding the ELBO is using optimization methods such as gradient-based optimization. This approach is known as Variational Autoencoder, which has been introduced in 2014 [72] and will be the main conceptual driving force for identifying behavioral structure in the animal motion data presented here. In this method, the approximate posterior $q(z|x)$ is parameterized by a neural network and the ELBO is used as the objective function to be maximized. This is done through the reparameterization of the variational lower bound, which yields a simple differentiable unbiased estimator of the lower bound. This Stochastic-Gradient-Variational-Bayes estimator can be used for efficient approximation of the posterior inference. The parameters of the neural network are updated using gradient-based optimization algorithms such as Stochastic Gradient Descent (SGD) [73] or Adaptive Moment Estimation (Adam) [64]. The main advantage of this method is that it can handle complex distributions and high-dimensional latent spaces. Additionally, by using neural networks, the model can be easily adapted to different types of data and can be trained with large amounts of data.

3.2.2 Implementing a VAE

3.2.2.1 Recognition model and probabilistic decoder

Lets consider a dataset $\mathcal{X}$ made up of N independent samples of a continuous or discrete variables x. The data is generated by a random process involving an unseen continuous random variable z. This process has two steps: first, a value of z_i is generated from a prior distribution $p^*_\theta(z)$, then a value of x_i is generated from a conditional distribution $p_{\theta^*}(x|z)$ based on the generated z_i. Here we assume that both the prior $p_{\theta^*}(z)$ and the likelihood $p_{\theta^*}(x|z)$ come from parametric families of distributions, and their probability density functions can be differentiated with respect to both θ and z. However, much of this process is not visible to us, as the true parameters θ^* and the values of the latent variables z_i are unknown. To solve this, the VAE framework introduces a recognition model $q_\phi(z|x)$, which represents an approximation to the intractable posterior $p_\theta(z|x)$. Since from a coding theory view the unobserved latent variable z can be interpreted as a latent representation or code the recognition model is also referred to as encoder. It produces a distribution (e.g. Gaussian) over the possible values of the code from which

a data point x could have been generated. In a similar fashion, the likelihood $p_\theta(x|z)$ is referred to as probabilistic decoder as it produces a distribution over the possible values of x given z. The parameter ϕ and θ are learned jointly by parameterizing the VAE with neural networks.

3.2.2.2 Deriving the objective function

The goal of the VAE is to infer $p_\theta(z|x)$ from $q_\phi(z|x)$. By inserting both distribution into Kullback-Leibler divergence equation we can formulate it as follows:

$$KL[q_\phi(z|x)||p_\theta(z|x)] = \sum_z q_\phi(z|x) \log \frac{q_\phi(z|x)}{p_\theta(z|x)} \tag{3.24}$$

$$= \mathbb{E}_{q_\phi(z|x)} \left[\log \frac{q_\phi(z|x)}{p_\theta(z|x)} \right] \tag{3.25}$$

$$= \mathbb{E}_{q_\phi(z|x)} \left[\log q_\phi(z|x) - \log p_\theta(z|x) \right] \tag{3.26}$$

Applying Bayes rule we can integrate the prior distribution $p(z)$, the data distribution $p(x)$ and likelihood $p(x|z)$:

$$KL[q_\phi(z|x)||p_\theta(z|x)] = \mathbb{E}_{q_\phi(z|x)} \left[\log q_\phi(z|x) - \log \frac{p(x|z)p(z)}{p(x)} \right] \tag{3.27}$$

$$= \mathbb{E}_{q_\phi(z|x)} \left[\log q_\phi(z|x) - (\log p(x|z) + \log p(z) - \log p(x)) \right] \tag{3.28}$$

$$= \mathbb{E}_{q_\phi(z|x)} \left[\log q_\phi(z|x) - \log p(x|z) - \log p(z) + \log p(x) \right] \tag{3.29}$$

$$= \mathbb{E}_{q_\phi(z|x)} \left[\log q_\phi(z|x) - \log p(x|z) - \log p(z) \right] + \log p(x) \tag{3.30}$$

$$KL[q_\phi(z|x)||p_\theta(z|x)] - \log p(x) = \mathbb{E}_{q_\phi(z|x)} \left[\log q_\phi(z|x) - \log p(x|z) - \log p(z) \right] \tag{3.31}$$

The right hand side of equation (3.31) can be rewritten as another Kullback-Leibler divergence:

$$\log p(x) - KL[q_\phi(z|x)||p_\theta(z|x)] = \mathbb{E}_{q_\phi(z|x)} \left[\log p(x|z) - (\log q(z|x) - \log p(z)) \right] \tag{3.32}$$

$$= \mathbb{E}_{q_\phi(z|x)} \left[\log p(x|z) \right] - \mathbb{E}_{q_\phi(z|x)} \left[\log q(z|x) - \log p(z) \right] \tag{3.33}$$

$$= \mathbb{E}_{q_\phi(z|x)} \left[\log p(x|z) \right] - KL[q(z|x)||p(z)] \tag{3.34}$$

With this we have derived the objective function of a VAE:

$$ELBO_{VAE} = \mathbb{E}_{q_\phi(z|x)}\left[\log p(x|z)\right] - KL[q(z|x)||p(z)] \tag{3.35}$$

Here, $q(z|x)$ projects a data point x_i into the latent variable space $\mathcal{Z}$ and $p(x|z)$ generates a data point from this space (Figure 3.1). The connection to a classical autoencoder becomes clear from this view. The Kullback-Leibler term acts as a regularization while the other term is an expected negative reconstruction error.

3.2.2.3 Modeling the prior distribution

The VAE objective function has a clear interpretation, where the goal is to model the data by finding the lower bound of the data's true distribution. In practice, this is a useful approach as finding the exact distribution can be infeasible. A question that I still have left out is how to implement and define the prior distribution $p(z)$ for the latent variables? The most common choice is to choose a Normal distribution $\mathcal{N}(0,1)$. This means the VAE tries to make $q(z|x)$ as close as possible to this. A benefit of modeling $p(z)$ is that it is easy to sample from this distribution. But the more important benefit is that the Kullback-Leibler divergence between both distributions can be computed in closed form. Given the Gaussian parameter $\mu(X)$ and $\Sigma(X)$, the mean and variance respectively, the closed form solution becomes:

$$KL[\mathcal{N}(\mu(X),\Sigma(X))||\mathcal{N}(0,1)] = \frac{1}{2}(tr(\Sigma(X)) + \mu(X)^T\mu(X) - k - \log det(\Sigma(X))), \tag{3.36}$$

where k is the dimension of the Gaussian and $tr(x)$ the trace function i.e. sum of the diagonal of matrix X. Equation (3.36) can be further simplified to be numerically stable (3.37):

$$KL[\mathcal{N}(\mu(X),\Sigma(X))||\mathcal{N}(0,1)] = \frac{1}{2}\sum_k(\exp(\Sigma(X)) + \mu^2(X) - 1 - \Sigma(X)) \tag{3.37}$$

3.2.2.4 Reparameterization trick in VAE

Now, by parametrizing the VAE with neural networks we want to jointly optimize the parameter ϕ and θ as mentioned before. This presents a problem as we cannot simply

sample a latent variable z from the encoder network. Sampling is a non-differentiable operation as the gradients cannot be propagated through it, which in turn makes the VAE not differentiable. Here, the reparameterization trick is used to circumvent this issue by expressing the random variable as a deterministic variable and a random noise, so that the gradient of the deterministic variable can be propagated through the random variable. This allows the VAE to be trained using gradient-based optimization methods, such as backpropagation. By taking a univariate Gaussian and let $z \sim p(z|x) = \mathcal{N}(\mu, \sigma^2)$, a valid reparameterization is

$$z = \mu + \sigma\epsilon, \tag{3.38}$$

where ϵ is an auxiliary noise variable ϵ $\mathcal{N}(0, 1)$. During backpropagation, the reparameterization trick allows the model to bypass the non-differentiable sampling process by expressing it as a deterministic variable and a random noise. In this way, the sampling process is outside of the network and does not depend on anything within it, so the gradients won't flow through it and the model remains differentiable with respect to its parameters.

Chapter 4

Methodology

In this section, a comprehensive elucidation of the Variational Animal Motion Embedding (VAME) framework, a key algorithmic contribution of this study, will be presented. This framework combines principles from latent variable modeling, deep learning, information theory, and network theory. Given the necessity for a substantial dataset to train such a model, the initial part of this chapter is dedicated to experimental design for acquiring mouse motion data through a bottom-up camera in an open-field arena. The subsequent segment introduces the VAME model and outlines its objective function. Following that, the discussion delves into the concept of communities, presenting their mathematical formulation and illustrating their application to the dynamic embedding space of VAME. Emphasis is placed on leveraging network theory concepts to unveil the hierarchical structure inherent in the data. Lastly, detailed information is provided on a benchmark dataset formulated for assessing the performance of VAME.

4.1 Experimental design, animal model and data processing

4.1.1 Experimental setup and data collection

4.1.1.1 Side and top-down view designs

An experimental design that can effectively capture the motion of a behaving animal is essential in order to study its behavioral structure. The design is a crucial factor as it will greatly impact the quality and quantity of the obtained data. The principal goal for such a design is to have a comprehensive view of the entire animal's body and track changes in body posture while the animal is engaging in various behaviors. There are different setup approaches that can be used to achieve such an objective. Many researchers are using a

home cage setup where the animal is placed in a small cage and recorded by a top-down and/or side view camera [74]. The side-view approach is not optimal as it captures the animal from a perspective that can be challenging, as there are many occlusions when the animal is not facing the side-view camera. In addition, the home cage approach is limited in that the animal's behavior is confined to a small space, which may not accurately reflect more naturalistic behaviors. Another approach is to record the animal from a top-view angle, which is often done in an open-field arena or behavioral box. This approach allows for the full body of the animal to be visible, and, combined with a bigger open-field arena, provides a more naturalistic setting for the animal to engage in various behaviors. However, when using a top-down facing camera, it is difficult to detect limb movements in small animals like mice, which limits the behavioral output that can be measured, as there is no information about the frequency of these limb movements. Others are using multi-camera setups for open-field recordings, which can then be triangulated to get a three dimensional body posture. While this has many advantages once set up, the biggest disadvantage is that the cameras used need to be constant in position after triangulation and it can easily happen that the camera position drifts due to experimental procedures or other effects.

4.1.1.2 Bottom-up view design

For this work, I have decided to construct a simple open-field arena approach that utilizes a Plexiglas plate to allow for a camera to be placed beneath the behaving animal (Figure 4.1, left). The design has several advantages for studying animal motion. First, the bottom-up view provided by this setup allows for the capture of the full body and limb movements of the animal. This is crucial for studying animal motion, as it allows for the detection of subtle changes in posture and movement that may be missed by other designs. Additionally, the open-field arena setup allows for the animal to move freely and engage in a variety of behaviors, providing a more naturalistic setting for the study. Another advantage of this design is that it makes it easy to track keypoint positions of the animal and to align it into an egocentric position in a later preprocessing step. This is important, as it allows for the extraction of behavioral patterns, as the animal's movements can be tracked relative to its own body rather than relative to the environment. Also, the use of a Plexiglas plate allows for the integration of neural recordings during

the experiment since the cables used for these recordings will not occlude the camera view. With this integration it can be possible to investigate how neural activity is related to the animal's movements and to study neural pathways that control behavior. The disadvantage to such a setup is that mice like to hide in their natural environments, which is not possible for them in an open-field. This could be resolved by adding objects or chambers into the arena but was not of interest for this work.

4.1.1.3 Treadmill for neural activity recording

To explore the correlation between neural activity and behavioral patterns in mice, I will adopt a well-established experimental design employed in our laboratory, as outlined in the work by Fuhrmann et al. (2015) [3]. This experimental approach integrates two-photon microscopy and kinematic analysis. Initially, a mouse will be head-fixed in a position under a two-photon microscope, enabling the acquisition of high-resolution images of the brain while the animal is running on a treadmill. This configuration ensures a robust capture of neural activity in specific brain regions, such as the hippocampus, during the execution of behaviorally relevant tasks. In conjunction with the two-photon microscope, a camera will be strategically positioned near the mouse to record its entire body movements. This setup facilitates the extraction of kinematic data by incorporating virtual markers onto the mouse's body. The virtual marker signals are subsequently fed into the model, enabling the identification of the structural pose patterns inherent in mouse behavior. The collected neural activity and behvioral data will serve to investigate the relationship between both modalities. For that, I will employ techniques derived from information and network theory to analyze the data. The goal is to identify patterns of neural activity that correlate with specific behaviors.

4.1.2 Animal model

Mice are a widely used model for AD research due to several reasons [75]. Firstly, mice are small and easy to handle, which makes them convenient to use in laboratory experiments. They also have a relatively short lifespan, which allows for rapid study of aging-related changes in behavior and disease progression. Secondly, mice have well-characterized genetics, which allows for the development of genetically engineered models to study specific aspects of the disease. For example, transgenic mice models that

express mutant forms of human genes associated with AD can be used to study the development and progression of the disease. Thirdly, mice have a similar nervous system and brain organization as humans, which makes them a good model to study neural mechanisms underlying behavior. Lastly, the availability of a wide range of behavioral tests and techniques for measuring cognitive function in mice, such as the Morris water maze, open field test, and fear conditioning, make it possible to study a wide range of behavioral changes associated with AD. These tests can be also used to study the relationship between cognitive decline and behavioral changes in AD.

4.1.2.1 Experimental conditions

After assembling the setup for the open field I conducted the experiment with eight 12 month old male transgenic and non-transgenic APPSwe/PS1dE9 (APP/PS1) mice [76] on a C57BL/6J background (Jackson Laboratory) (four animals per group). Prior to the experiment, the mice were group housed under standard laboratory conditions with a 12-h light-dark cycle with food and water ad libitum. The experiment itself consisted of the mice freely moving through the Plexiglas arena. For this, mice were placed in the center of the arena and were recorded for a duration of 50 minutes each from a bottom-up perspective. During the time, the mice were left unperturbed to not influence their behavior. All experimental procedures were performed in accordance with institutional animal welfare guidelines and were approved by the state government of North Rhine-Westphalia, Germany.

4.1.3 Data acquisition and preprocessing

4.1.3.1 Experimental room setup

The behavior of the mice in the open-field arena was captured with a temporal resolution of 60 frames per second by a complementary metal-oxide semiconductor (CMOS) camera (Basler acA2000-165umNIR) that was centrally located 35 cm below the arena. The camera was equipped with a wide-angle lens (CVO GM24514MCN, Stemmer Imaging) to ensure that the entire arena was captured in the recordings (see Figure 4.1). In order to provide homogeneous illumination of the recording arena from below, three infrared light sources (LIU780A, Thorlabs) were placed around 70 cm away from the center of the

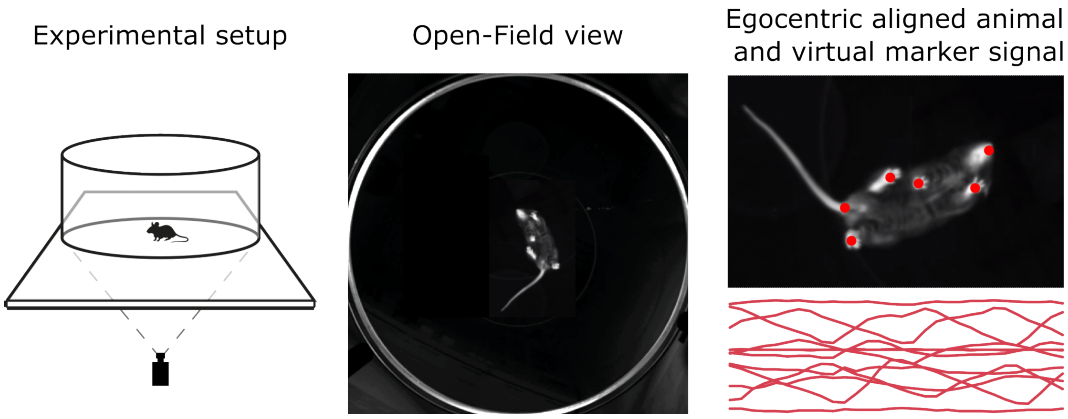

Figure 4.1: **Transforming the animal's position from open-field allocentric coordinates to its egocentric coordinates.** (Left) Depiction of the experimental arrangement captured from a bottom-up camera perspective. (Middle) Exemplary image featuring a mouse in the open field as viewed from below. (Right) Placement of a virtual marker on the mouse's body with a representation of the resulting time series.

arena. All recordings were performed under dim room light conditions to mimic light conditions in which mice are most active. The use of infrared lights in conjunction with the CMOS camera ensured that the behavior of the mice was captured in high quality, even in the low light conditions in which the experiment was conducted. This allows for a good visibility of the animal's body and makes the downstream analysis of identifying keypoints of the mice in the open field exploration experiment easier (Figure 4.1).

4.1.3.2 Extracting keypoints using pose estimation

After capturing videos of all mice, the processing of the videos begins with the goal of identifying behavioral patterns. Specifically, the previously described method of keypoint identification are applied to extract behavioral kinematics. To achieve this, I place six virtual marker on the animal's body: the four paws, the nose, and the tailroot. These keypoints are chosen because they represent the main kinematic movement points of the animal and allow for a comprehensive analysis of the animals behavior. The DeepLabCut algorithm [18] is used to assign virtual markers to every video frame. This algorithm is a widely used and well-established method for tracking body parts of animals in videos. It uses a residual neural network (ResNet-50) that is pre-trained to detect image features and then fine-tuned to recognize the location of the virtual markers in the video frames by providing the algorithm with a set of labeled frames where the position of every virtual marker is manually annotated. In this experiment, from the resulting 16 videos, 650

frames were uniformly sampled and the position of every virtual marker is hand-labeled prior to training the algorithm. This is done to ensure that the algorithm can generalize to new data and can accurately detect the virtual markers in the remaining frames of the videos. Once the algorithm is trained, performance is evaluated by comparing the predicted position of the virtual markers with the manually annotated ones. The resulting training error was 2.14 pixels and the test error 2.51 pixels. This indicates a high confidence of the DLC algorithm to detect the selected animal body keypoints in all video frames reliably. This process allows for the extraction of kinematic data, such as the virtual markers position and frequency.

4.1.3.3 Egocentric alignment of the keypoints

The data must undergo a reorientation process to align it with the animal's frame of reference before it becomes suitable for use in the VAME algorithm. This is known as egocentric alignment. The goal of this process is to align the animal's body from left to right, i.e. tail-root to nose, in every video frame (Figure 4.1, right). To achieve this, a rotational matrix $\mathcal{R}$ is computed. This matrix is used to rotate the frame around the center point ($c = x_c, y_c$) between the nose and tail of the animal. The angle of rotation, θ, is calculated as the angle between an assumed horizontal line at the center point and the line connecting the nose and tail-root. By rotating the frame by this angle, the body of the animal is aligned from tail-root to nose in each frame. Depending on the initial mouse orientation this can also lead to a flipped mouse position where the mouse is aligned from nose to tail. This can be countered by simply rotating the frame by 180 degrees around c. This process results in frames and marker coordinates that are aligned with the animal's perspective, represented as $\mathbf{X} \in \mathbb{R}^{N \times m}$. where N is the length of the recording (90000 frames = 25 minutes recording with 60 Hz) and m is the number of marker coordinates (x and y) of the animal. Implementing this algorithm requires no human intervention and operates entirely autonomously. The crucial prerequisite is a proficiently trained DLC model to ensure accurate pose estimation, preventing any confusion between the nose and tail-root during the process.

4.2 Variational Animal Motion Embedding

4.2.1 Introduction

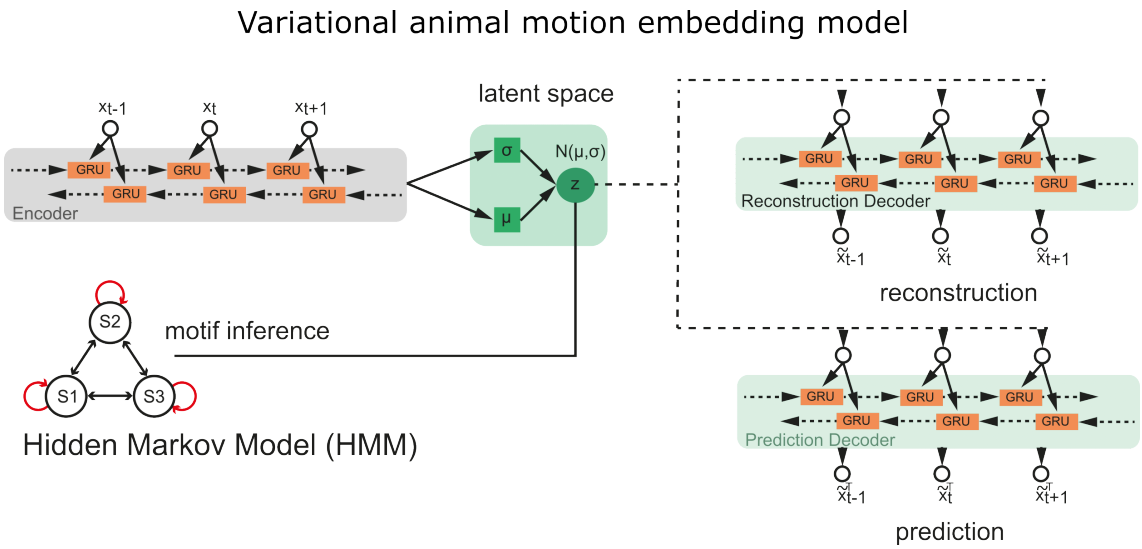

Figure 4.2: **Schematic representation of the VAME model** The encoder and both decoders of the model are parameterized with bidirectional recurrent neural networks. The encoder is trained to project the data into a latent space, and subsequently, an Hidden Markov Model (HMM) is employed to infer motifs from this space (modified from [13]).

This section will provide a comprehensive explanation of the key model of this thesis, VAME. This method utilizes all concepts introduced in chapter 3 to overcome a number of limitations that have been identified in existing models. VAME aims to address these limitations and to provide a more robust and accurate way to analyze animal motion. It is a significant step forward in the field of computational ethology and has the potential to improve our understanding of animal behavior, movement patterns and their correlation to brain activity.

VAME is a deep learning framework for time series embedding that applies unsupervised probabilistic techniques to identify hidden states or clusters, here for behavior signals obtained from pose estimation tools or dimensionally reduced video data. It utilizes RNNs combined with the VAE framework, which are used to learn and compress the input signal into a lower dimensional space, also referred to as dynamical embedding (Figure 4.2). This allows for more efficient and accurate representation of the behavior signals, making it a powerful tool for analysing and understanding animal behavior.

4.2.2 Model design

4.2.2.1 Data representation and latent projection

Given a set of n multivariate time series $\mathbb{X} = \{\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^n\}$ where each time series $\mathbf{X}^i = (x^1, x^2 \ldots, x^N)$ contains $N \times m$ ordered real values, the objective of VAME is to learn a lower dimensional latent space $\mathbf{Z}$ that captures the dynamics of the time series data and embeds them based on their spatiotemporal similarities. To achieve this goal the multivariate time series $\mathbf{X}^i$ are sampled into defined subsequences $\mathbf{x}_i \in \mathbb{R}^{m \times w}$, where m representing the features of the time series data (which can be the x, y virtual marker coordinates) and w representing the sampled time window. Now, for every $\mathbf{x}_i$ we learn a vector representation $\mathbf{z}_i \in \mathbb{R}^d$, which effectively reduces its dimension ($d < m \times w$). More specifically, this vector representation $\mathbf{z}_i$ is learned via the non-linear mappings $f_{enc} : \mathbf{x}_i \rightarrow \mathbf{z}_i$ and $f_{dec} : \mathbf{z}_i \rightarrow \tilde{\mathbf{x}}_i$, where f_{enc}, f_{dec} denotes the encoding and decoding process, respectively and is defined by,

$$\mathbf{z}_i = f_{enc}(\mathbf{x}_i)\,. \tag{4.1}$$

4.2.2.2 Encoder and decoder formulation

VAME utilizes a bi-directional RNN (biRNN) with two layers as the encoder to encode the spatiotemporal latent representation. This encoder has parameters denoted by ϕ. The model also has two decoders, each of which is also a biRNN, with parameters θ for the reconstruction decoder and η for the prediction decoder. The biRNN encoder and decoders work together to capture and reconstruct the spatiotemporal information present in the data being processed. I will expand the notion of the prediction decoder in subchapter 4.2.3.

The input data is temporally dependent, meaning that the current state of the data is influenced by past and future events. In order to effectively capture this temporal dependency, biRNNs are used as the preferred method in the model. A biRNN extends the traditional unidirectional RNN by adding a second hidden layer that runs in the opposite direction of the first layer. This allows the model to gather information about the temporal dependencies of the input data from both the past and the future. For example, if the first layer processes the data in the forward direction, the second layer processes

the data in the reverse direction. The result is that the biRNN can understand the temporal dependencies of the data from multiple perspectives, enabling it to capture the full temporal dynamics of the input data. Its hidden representation is determined by recursively processing each input and updating their internal state $\mathbf{h}_t$ at each timestep for the forward and backward path via,

$$\mathbf{h}_t^f = tanh(f_\phi(\mathbf{x}_i^t, \mathbf{h}_{t-1}), \quad \mathbf{h}_t^b = tanh(f_\phi(\mathbf{x}_i^t, \mathbf{h}_{t+1}), \quad \mathbf{h}_c = \mathbf{h}_t^f + \mathbf{h}_t^b \tag{4.2}$$

where $\mathbf{h}_t^f$ is the hidden information of the forward pass and $\mathbf{h}_t^b$ is the hidden information of the backward pass, $\mathbf{x}_i^t$ is the current time step of the input sequence $\mathbf{x}_i$, f_ϕ is a non-linear transition function, and ϕ is the parameter set of f_ϕ. The transition function f_ϕ is an important component of an RNN that determines how information is passed from one step to the next in the processing of sequential data. Typically, the transition function is modeled as either a long short-term memory (LSTM) [77] or a gated recurrent unit (GRU) [78]. In this model, GRUs have been chosen as the transition function for both the encoder and decoder since they have fewer parameter to train which makes them computationally more efficient.

4.2.2.3 Modeling the data distribution

The joint probability $p_\phi(\mathbf{x}_i)$ for each subsequence $\mathbf{x}_i$ is represented in an RNN as the factorized product of conditionals,

$$p_\phi(\mathbf{x}_i) = \prod_{t=1}^{T} p_\phi(x_t | x_{1:t-1}). \tag{4.3}$$

In the context of this model, VAEs are utilized to learn the joint distribution $p(\mathbb{X})$ over all subsequences of the input data and to uncover the underlying generative process of that data. VAEs have been shown to effectively model complex multivariate distributions. Additionally, they have been shown to generalize well across different datasets, making them ideal for this framework.

To recap the VAE method in the context of this model, a set of latent random variables $\mathbb{Z}$ are embedded by the VAE model, which represents the underlying variations in the observed data $p(\mathbb{X})$ and from which the model is able to generate new data sequences

$\mathbf{x}_i^{new}$ through conditioning $p(\mathbb{X})$ on $\mathbb{Z}$. Hence, the joint probability distribution is defined as,

$$p_\theta(\mathbb{X}, \mathbb{Z}) = p_\theta(\mathbb{X}|\mathbb{Z})p_\theta(\mathbb{Z}) \tag{4.4}$$

and parameterized by θ.

Determining the data distribution $p(\mathbb{X})$ by marginalization is intractable due to the non-linear mappings between $\mathbb{X}$ and $\mathbb{Z}$ and the integration of $\mathbb{Z}$. In order to overcome the problem of intractable posteriors the VAE framework introduces an approximation of the posterior $q_\phi(\mathbb{Z}|\mathbb{X})$ and optimizes a lower-bound on the marginal likelihood,

$$\log p_\theta(\mathbb{X}) \geq \mathbb{E}_{q_\phi(\mathbb{Z}|\mathbb{X})}[\log p_\theta(\mathbb{X}|\mathbb{Z})] - KL(q_\phi(\mathbb{Z}|\mathbb{X})||p_\theta(\mathbb{Z})), \tag{4.5}$$

where $KL(Q||P)$ denotes the Kullback-Leibler divergence between two probability distributions Q and P. The prior $p_\theta(\mathbb{Z})$ and the approximate posterior $q_\phi(\mathbb{Z}|\mathbb{X})$ are typically chosen to be in a simple parametric form, such as a Gaussian distribution with diagonal covariance so that there exists a closed-form solution. The generative model $p_\theta(\mathbb{X}|\mathbb{Z})$ and the inference model $q_\phi(\mathbb{Z}|\mathbb{X})$ are trained jointly by optimizing equation (4.5) w.r.t their parameters. Using the *reparameterization trick* (equation (4.6)) the full model can be trained through standard backpropagation techniques with stochastic gradient descent.

4.2.3 Variational lower bound of VAME

4.2.3.1 Deriving the objective function

The inference model (or encoder) $q_\phi(\mathbf{z}_i|\mathbf{x}_i)$ of VAME is parameterized by a biRNN. By concatenating the last hidden states of the forward and backward steps of the biRNN a global hidden state $\mathbf{h}_i$ is obtained, which is a fixed-length vector representation of the entire sequence $\mathbf{x}_i$. To get the probabilistic latent representation $\mathbf{z}_i$ a prior distribution over the latent variables $p_\theta(\mathbf{z}_i)$ is defined as an isotropic multivariate Normal distribution $\mathcal{N}(\mathbf{z}_i; \mathbf{0}, \mathbf{I})$. Its parameter μ_z and Σ_z of the approximate posterior distribution $q_\phi(\mathbf{z}_i|\mathbf{x}_i)$ are generated from the final encoder hidden state by using two fully connected linear layers. The latent representation $\mathbf{z}_i$ is then sampled from the approximate posterior and

computed via the reparameterization trick,

$$\mathbf{z}_i = \mu_z + \sigma_z \odot \epsilon \quad , \tag{4.6}$$

where ϵ is an auxiliary noise variable and $\odot$ denotes the Hadamard product.

The generative model $p_\theta(\mathbf{x}_i|\mathbf{z}_i)$ (or decoder) receives $\mathbf{z}_i$ as input at each timestep t and aims to reconstruct $\mathbf{x}_i$. The mean squared error (MSE) is used as a reconstruction loss, defined by,

$$\mathcal{L}_{MSE} = \frac{1}{n}\sum_{i=1}^{n} ||\mathbf{x}_i - \tilde{\mathbf{x}}_i||_2^2 \,. \tag{4.7}$$

The log-likelihood of $\mathbf{x}_i$ can be expressed as in equation (4.5). Since the KL divergence is non-negative the log-likelihood can be written as

$$\mathcal{L}(\theta, \phi; \mathbf{x}_i) = \mathbb{E}_{q_\phi(\mathbf{z}_i|\mathbf{x}_i)}[\log p_\theta(\mathbf{x}_i|\mathbf{z}_i)] - KL(q_\phi(\mathbf{z}_i|\mathbf{x}_i)||p_\theta(\mathbf{z}_i)) \,. \tag{4.8}$$

Here, $\mathcal{L}(\theta, \phi; \mathbf{x}_i)$ is a lower bound on the log-likelihood (ELBO) and represents an objective function to train the model.

4.2.3.2 Extending the objective function

Since the VAME model is using an additional decoder, the the ELBO needs to be extended by an additional biRNN decoder $p_\eta(\tilde{\mathbf{x}}_i|\mathbf{z}_i)$ to predict the evolution $\tilde{\mathbf{x}}_i$ of $\mathbf{x}_i$, parameterized by η. This composite model is able to jointly learn important features for reconstruction and predicting subsequent virtual marker signal. In this way, the model is able to capture the temporal relationships between subsequent subsequences $\mathbf{x}_i$ and produce better results compared to traditional approaches [79]. Moreover, $p_\eta(\tilde{\mathbf{x}}_i|\mathbf{z}_i)$ serves as a regularization for learning $\mathbf{z}_i$ so that the latent representation not only memorizes an input time series but also estimates its future direction. Equation (4.8) is then extended by an additional term and parameter,

$$\mathcal{L}(\theta, \phi, \eta; \mathbf{x}_i) = \mathbb{E}_{q_\phi(\mathbf{z}_i|\mathbf{x}_i)}[\log p_\theta(\mathbf{x}_i|\mathbf{z}_i)] + \mathbb{E}_{q_\phi(\mathbf{z}_i|\mathbf{x}_i)}[\log p_\eta(\tilde{\mathbf{x}}_i|\mathbf{z}_i)] - KL(q_\phi(\mathbf{z}_i|\mathbf{x}_i)||p_\theta(\mathbf{z}_i)) \,. \tag{4.9}$$

Equation (4.9) represents the derived variational lower bound of VAME by incorporating an additional decoder. The objective of VAME is to minimize

$$\min_{\theta,\phi,\eta} \mathcal{L}(\theta, \phi, \eta; \mathbf{x}_i). \tag{4.10}$$

Finally, the loss function for this model can be written as

$$\mathcal{L}_{total} = \mathcal{L}_{reconstruction} + \mathcal{L}_{prediction} + \mathcal{L}_{KL} \quad , \tag{4.11}$$

where $\mathcal{L}_{prediction}$ is the MSE loss of the additional decoder.

4.3 Motion structure and hierarchy

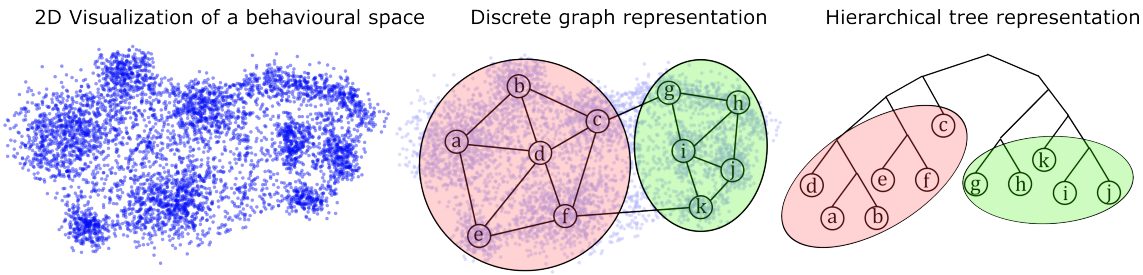

Figure 4.3: **Illustration showcasing a two dimensional visualization of an embedded space.** Utilizing an HMM leads to the discretization of the space. The resultant time series can be converted into a graph structure. Through the establishment of a hierarchical representation of this graph structure, we can identify subgraphs or communities within the embedding space.

In this section, I introduce the mathematical framework for the concept of communities, which are conceptually borrowed from network theory. I illustrate their application ot the dynamic embedding space and emphasize how they can be used to unveil the hierarchical structure present in the mouse pose data.

4.3.1 Behavioral state space

After convergence, the latent space variable set $\mathbb{Z}$ can be utilized to examine the structure of the time series signal and its hierarchical order. The first goal is to identify a set of underlying states $\mathbb{B} = \{b_1, \ldots, b_k\}$ with k being the number of states that might exist in the data. Typically, there is a lack of a priori knowledge regarding the number of states within a continuous time series, unless it is precisely defined. In this context, I will

now present the method employed to identify states within a dynamical embedding, and subsequently, I will delve into strategies for determining the appropriate number of

4.3.1.1 Identifying states in dynamical embedding space

K-Means is one of the primary methods for identifying clusters in a lower dimensional representation of data, but it is most effective when the clusters in the representation are well-separated. This is often not the case in spatiotemporal or dynamical representations, as the input data is usually a continuous signal. VAME groups similar time series signals and creates a spatiotemporal structure within the embedding, resulting in a continuous representation of the data. Although k-Means and other density clustering techniques can still be used, they may not be the optimal choice for identifying clusters in this type of space.

Another approach is to unroll the latent space $\mathbb{Z}$ along its time axis and treat it as a new time series $\mathbb{X}_{emb}$. Now, we can apply an HMM, which is a suitable method for separating time series data into distinct states. Moreover, the Markovian property is particularly fitting for animal motion data, as animal behavior often exhibits predictable transitions between different states. For example, an animal may alternate between resting and moving, or between different types of movement. By modeling these transitions as a Markov process, HMMs can effectively identify and label the different states of animal motion in the time series data.

Here, $\mathbb{X}_{emb} \in \mathbb{R}^{d \times \mathbb{N}-w}$ represents the feature space from which the state structure is to be identified, with the embedding dimension d, the number of datapoints $\mathbb{N}$ and the subsequence length w. By conceptualizing the underlying dynamical system as a discrete-state continuous-time Markov chain, an HMM with Gaussian distribution emission probability was employed in this space for the purpose of detecting states or motifs. The HMM was implemented using the *hmmlearn* python package, utilizing the default settings for the Gaussian emission model as provided by the package.

4.3.1.2 Transition probability matrix as graph structure

The resulting state time-series S_B can now be interpreted as a discretized version of the continuous latent representation time series $\mathbb{X}_{emb}$. More specifically, it can be now treated

as a discrete-time Markov chain where the transition probability of one state is only dependent on its past state. Using this formalism, a $K \times K$ transition probability matrix $\mathbb{T}$ can be created with the elements

$$\mathcal{T} = P(b_k|b_l), \tag{4.12}$$

being the transition probabilities from one state $b_l \in \mathbb{B}$ to another state $b_k \in \mathbb{B}$.

The discrete state time series $\mathbb{S}_B$ can be transformed into a graph structure, generally termed as $\mathbb{G}$ (Figure 4.3). Representing a matrix or transition matrix as a graph or directed graph is a concept from linear algebra and network theory. In the case of a discrete time series $\mathbb{S}_B$, it can be modeled as a directed graph $\mathbb{G}_B$. modeling $\mathbb{S}_B$ as a directed graph allows us to utilize its graph structure, which requires introducing some general terms from graph theory. This forms the basis for creating a hierarchical order of $\mathbb{G}_B$ and defining its state communities.

4.3.2 Concepts from network theory

4.3.2.1 Definition of a graph

A graph $\mathbb{G} = (V, E)$ is defined by a finite, non-empty set of vertices V with vertex v referred to as $v \in V(\mathbb{G})$ and a finite set of edges E that connect the vertices, where an edge is denoted as $e \in E(\mathbb{G})$. I will refer to v as nodes in the following. An edge e_{ij} connects two nodes (v_i, v_j). In the case of an undirected graph the edge e_{ij} as well as the edge e_{ji} refer to the same edge and have the same edge weight to it. In a directed graph, the weight of edges e_{ij} and e_{ji} can be different or unequal. Additionally, either one of the edges or both may exist as a directed graph allows for unidirectional edges.

4.3.2.2 Degree of a graph

The degree of a vertex is an important metric for characterizing the structure of a graph. It is defined as the number of edges incident to a particular vertex, meaning the number of edges that are connected to that vertex. The degree of a vertex provides valuable information about the connectivity of the graph and can be used to calculate other graph parameters, such as the average degree, the minimum and maximum degree, and the degree distribution. There are two types of degree in a graph: the in-degree and the out-degree. The in-degree of a vertex is the number of incoming edges, while the out-degree

is the number of outgoing edges. In undirected graphs, the degree of a vertex is simply the sum of its in-degree and out-degree. The degree of a vertex is a useful metric for identifying special vertices, such as isolated vertices (degree 0), leaf vertices (degree 1), and hub vertices (high degree), which can have significant impact on the structure and behavior of the graph.

4.3.2.3 Formalism of subgraphs and cliques

Another idea I need to introduce is the formalism of a *subgraph* $\mathbb{Q}$. The concept of a subgraph is useful in various areas of graph theory and computer science. It allows to focus on a smaller portion of a larger graph and analyze its properties, patterns, or relationships. A subgraph $\mathbb{Q}$ can be obtained by selecting a specific subset of vertices and edges from the supergraph $\mathbb{G}$, or it can be derived from the supergraph by a specific rule or criterion. The subgraph retains all the characteristics of the supergraph, such as its adjacency relationships and weights, and it also inherits any additional properties, such as connectivity or cycle formation, that the supergraph may have. The subgraph, however, has its own unique properties, such as its own degree distribution and graph density.

With this formalism, I can introduce the concept of a *clique*. In graph theory, a clique is a subgraph $\mathbb{Q}_c$ in which every pair of vertices is directly connected by an edge. This means that all vertices in a clique are completely connected, forming a densely connected subset within the larger graph. Cliques are useful for characterizing the structure of a graph and identifying densely connected regions within a graph. A clique can be of any size, ranging from a single vertex clique to a complete graph, where every vertex is connected to every other vertex. Cliques are commonly used in various areas of computer science and mathematics, such as social network analysis, community detection, and graph clustering. Here, community detection, also known as graph clustering or network clustering, is the task of identifying groups of vertices in a graph that are densely connected internally, but sparsely connected with other groups. In other words, it involves partitioning the graph into distinct, non-overlapping subgraphs, such that the vertices within each subgraph are highly connected, while the connections between subgraphs are sparse. The goal of community detection is to uncover the underlying structure of a graph and gain insight into the relationships and patterns within the data. There are

various methods for community detection, including heuristic methods, such as modularity optimization, and probabilistic methods, such as generative models. The choice of method depends on the specific problem and requirements of the data.

4.3.3 Data-Driven Communities

In this thesis, my objective was to employ a data-driven method inspired by both the data itself and the inherent structure of behavioral motion. A community is by definition a subset of vertices that forms a densely connected cluster within the larger graph, and its members are more likely to have strong connections with each other compared to vertices outside the community. Here, we loosen the definition of a clique and are interested in densely connected vertices where clusters do not need to be completely connected to each other.

To detect these communities within the directed graph $\mathbb{G}_B$, I used the idea of hierarchical graph clustering. Here, the graph $\mathbb{G}_B$ is transformed into a binary tree representation $\mathbb{T}_B$ (Figure 4.3). This can be achieved by iteratively merging two nodes (v_i, v_j) into a new node $v_i j$ based on a cost criterion until only the root node v_R is left. Every leaf of this tree represents an original node b_k from the graph $\mathbb{G}_B$. In this work, I tested four different cost function, all with the principle of being motivated by the data. In general, I use the probability of occurrences of states U_i, which refers to the state distribution and their probability of appearance in the discrete time series $\mathbb{S}_B$ as well as the transition probability $\mathcal{T}_{ij}$ between to states. The cost function $\mathbb{C}_R$ used in the VAME open-field experiment is

$$\mathbb{C}_R = \min_{i,j} \left(\sum_{i,j} \frac{U_i + U_j}{T_{ij} + T_{ji}} \right). \tag{4.13}$$

It is important to note that after each reduction step in the process, the matrix $\mathbb{T}$ must be recalculated to consider the merging of nodes. The result of the process is the identification of communities. One approach is by cutting $\mathbb{T}$ at a specified depth of the tree, similar to the hierarchical clustering method used for dendrograms. An alternative method is to visually examine the tree and identify branches that contain a high number of interconnected nodes. In the context of behavioral motif discovery, videos of the identified behavior can be created to allow for closer inspection of the tree and determine which

branches contain specific behavioral types. This leads to the discovery of macro behaviors, such as "walking," "rearing," or "grooming," as communities that consist of smaller motifs.

4.4 Benchmark dataset

4.4.1 Expert labeling of the data

VAME is a self-supervised approach focused on uncovering the structural information contained in spatiotemporal signals. As the data used is from real-world sources, there are no labels assigned to identify what each timestep represents. This creates a challenge in determining its performance since there is no direct metric to apply. Additionally, for comparison with other methods, a consensus on what constitutes accuracy and what does not must be established. To address this, I created a benchmark dataset with two experts in behavioral neuroscience. This dataset includes a six minute recording of a mouse freely moving in the experimental open-field setup. The experts possessed extensive experience in in-vivo experiments and quantifying behavior using an ethogram-based method. They reviewed the video in slow-motion, both forwards and backwards in time, and labeled the behavior by breaking it down into smaller atomic motifs and combining these motifs. For instance, they could label a behavioral sequence as either "walk" or "exploration," or both. The experts' annotations were then condensed into five coarse behavioral labels based on the atomic motifs as shown in Table 4.1. The coarse labels were established based on the behavior descriptions from the Mouse Ethogram database (www.mousebehavior.org), which aggregates several previously published ethograms.

Table 4.1: Assignment of motifs into coarse behavior labels (adapted from [13]).

Coarse label	Assigned motif
Walk	Walk, walk and bend, walk and sniff
Pause	No locomotion, Bending, looking up or down while standing still
Groom	Groom
Rear	Rear, low-rear, wall-rear
Exploratory	Undirected sniffing while standing still, bending, looking up or down

4.4.2 Clustering evaluation metrics

To assess the performance of a model using the benchmark dataset, metrics must be established. I employed three clustering evaluation metrics: Purity, Normalized Mutual Information (NMI), and Homogeneity. Purity evaluates the degree to which each cluster consists of data from a single class. NMI normalizes the Mutual Information score and ranges from 0 (no mutual information) to 1 (perfect correlation). Homogeneity is a stricter version of Purity, requiring all clusters to contain data from a single class. Purity is defined as

$$\text{Purity}(U, V) = \frac{1}{N} \sum_{u \in U} \max_{v \in V} |u \cap v|, \tag{4.14}$$

where U it the set of manually assigned labels u, V is the set of labels generated by VAME v and N is the number of frames in the behavioral video. The NMI score is written as

$$\text{NMI}(U, V) = \frac{\text{MI}(U, V)}{E(H(U), H(V))}, \tag{4.15}$$

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

$$\text{MI}(U, V) = \sum_{u \in U} \sum_{v \in V} \frac{|u \cap v|}{N} \log \left(\frac{N|u \cap v|}{|u||v|} \right), \tag{4.16}$$

and $H(U)$ is the entropy of set U defined as

$$H(U) = -\sum_{i=1}^{|U|} \frac{|u \cap v|}{N} log \left(\frac{|u \cap v|}{N} \right), \tag{4.17}$$

where the $||$ operator denotes the amount of frames that have the corresponding labels assigned. Homogeneity is defined as

$$\text{Homogeneity} = 1 - \frac{H(U|V)}{H(U)}, \tag{4.18}$$

where the conditional entropy of manually assigned labels given the cluster assignments from VAME is given by

$$H(U|V) = -\sum_{u=1}^{|U|} \sum_{k=1}^{|K|} \frac{u \cap v}{|u \cap v|} log \left(\frac{u \cap v}{|v|} \right), \tag{4.19}$$

It is important to note that the Purity score (4.14) tends to be higher when the set V is larger than set U, and the NMI score (4.15) is usually higher when the sizes of sets U and V are similar, meaning that the number of labels in the human-assigned set is comparable to the number generated using VAME.

Chapter 5

Results

In this chapter, I present the main results of this thesis. The first part is a modified and extended version of the publication "Identifying behavioral structure from deep variational embeddings of animal motion" in *Nature Communication Biology* [13], while the second part follows the conference paper "Hierarchical network analysis of behavior and neuronal population activity" published at the *Conference on Cognitive Computational Neuroscience* [38] and adds some further details on the correlation between behavioral motifs and the hippocampal CA1 brain activity.

5.1 VAME

5.1.1 Introduction

Tools such as DeepLabCut [18], SLEAP [19], and DeepPoseKit [20] are using supervised deep learning to efficiently track animal body parts in videos or images. This is achieved by training the models on labeled data, where the position of body parts is manually annotated. The robustness of deep neural networks allows for a high degree of generalization between datasets, meaning the models can be applied to different animals or different conditions with good performance [18]. However, while such tools provide a continuous representation of the animal body motion, the extraction of underlying discrete states as a basis for quantification remains a key challenge. Recent methods provided a way of segmenting the motion into specific, meaningful behaviors or postures for further analysis. However, they each come with their own limitations as previously discussed.

5.1.2 Overview of the method

VAME has been developed to provide an effective and accurate way to extract underlying latent states from behavioral signals obtained from pose estimation tools or dimensionality-reduced video information. It is an unsupervised, probabilistic deep learning framework that uses a variational recurrent neural network autoencoder (VAE-RNN) to learn and embed a time series signal into a lower-dimensional space. An HMM is then used to infer hidden states, which here represent behavioral motifs. HMMs are a powerful tool for modeling temporal dependencies in the data and are able to identify the underlying structure in the latent embedding of the behavioral motion time series. The VAE-RNN in VAME is designed to learn a disentangled representation of latent factors. This is done by embedding the input signal into a lower-dimensional space, where segments of the behavioral signal are grouped by their spatiotemporal similarity. The VAE framework allows the model to effectively simplify the data distribution by mapping it to a simpler prior distribution. This enables VAME to discover underlying latent states that are not immediately obvious in the raw data. VAME is inspired by recent advances in the field of temporal action segmentation [80], representation learning [81, 82, 83, 84], and unsupervised learning of multivariate time series [85, 86]. This allows to leverage recent techniques and methodologies in these fields to develop a powerful and effective tool for extracting robustly and reliably these underlying latent states from the behavioral virtual marker time series. Moreover, it allows for a high degree of generalization between datasets, and the use of the VAE-RNN and HMM components enable the extraction of underlying discrete states as a basis for quantification, which I will show in more detail in the following sections.

5.1.3 Experiment and model

5.1.3.1 Extracting and aligning the virtual marker

I conducted a behavioral experiment in an open-field arena where the mice were allowed to move freely (Fig. 4.1)). The movement of the mice was continuously monitored using a bottom-up camera for a duration of 50 minutes. This camera view allows to capture most of the animal's movements with just one camera angle, which could then be tracked efficiently using pose estimation. The objective of the experiment was to build a model

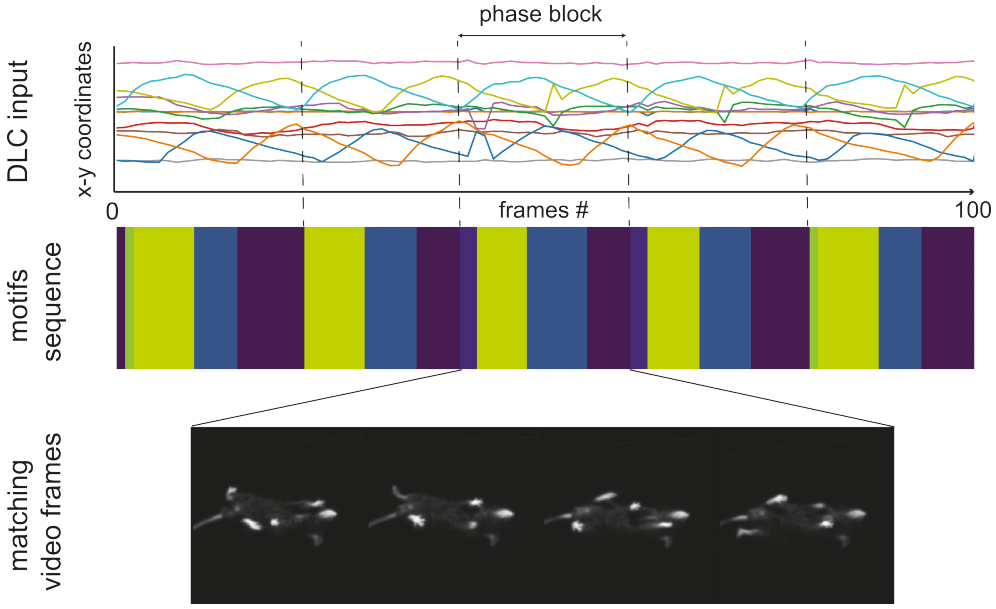

Figure 5.1: **Example of an egocentrically aligned DLC signal and its inferred motif sequence.** Top: Example of an egocentrically aligned DLC sequence, depicting a complete walking cycle (phase block). Middle: The motif sequence identified by VAME, revealing the underlying phase block structure. Bottom: Video frames from the walking cycle that align with the identified sequence (modified from [13]).

that could learn the behavioral structure of the mice solely from the kinematic pose tracking data. To identify the postural dynamics of the animals from the video recordings, I utilized DLC [18]. It was used to track the movement of the mice by placing six virtual markers on the animal's four paws, nose, and tailbase. The animal was in a preprocessing step aligned from its allocentric arena coordinates to its egocentric coordinates by rotating each frame around the center between its nose and tail. This alignment ensured that the animal was oriented from left to right, with the tailbase on the left and the nose on the right. This process resulted in a time-dependent series of data $\mathbf{X} \in \mathbb{R}^{N \times m}$ for each animal, where N represents the number of frames and $m = 10$ represents the number of (x, y)-marker positions that captured the kinematic of the specified body parts. This data captured the movements of the mice in a series of (x, y) positions, providing insight into the kinematics of the mice over time. For further information on the experimental setup, preprocessing and alignment functionality used in the experiment, please refer to the Methodology subchapter 4.1.3.1.

5.1.3.2 Learning an embedding from trajectory samples

The aim of the study was to extract meaningful information from the virtual marker time series data of mouse kinematics, with the goal of effectively quantifying behavior based on the spatial and temporal information of the body dynamics of the mice. To do this, the developed algorithm picked randomly trajectory samples $\mathbf{x}_i \in \mathbb{R}^{m \times w}$ from the time series data $\mathbf{X}$. These trajectory samples consisted of pre-defined time windows of length $w = 30$ and served as the input for training the VAME model. The first objective was to identify behavioral motifs, which were defined as "stereotyped and re-used units of movements" [10]. The second objective was to identify the hierarchical and transition structure of these behavioral motifs. This involves understanding how the different motifs are organized and how they transition from one to another over time. This information can be used to identify patterns and regularities in the mouse behavior and can provide a deeper understanding of the behavioral structure.

To briefly recap, the VAME model is made up of three biRNNs with GRUs as transition function. The encoder biRNN takes a trajectory sample (500 ms of behavior) and converts it into a lower dimensional latent space called $\mathbf{Z}$. This is achieved by mapping the trajectory sample to a fixed vector representation ($\mathbf{z}_i$) with a lower dimensionality than the input ($d < m \times w$) and passing it on to the biRNN decoder. The biRNN decoder then reconstructs the lower dimensional vector back into an approximation of the original input trajectory ($\tilde{\mathbf{x}}_i$). Another biRNN decoder is used to predict the structure of the subsequent time series trajectory ($\tilde{\mathbf{x}}_{i+1}$) from $\mathbf{z}_i$, which helps to regulate $\mathbf{Z}$ and enhances the encoder's ability to learn important dynamical features from the behavioral time series. The two decoder model (reconstruction and prediction) was found to perform better than a single decoder model (reconstruction only) in terms of the tested metrics. See appendix section A.3 for the model selection process.

5.1.3.3 Cyclic phase block

The model is trained as VAE [87] with a standard normal prior. Within the VAE framework, it is possible to investigate if the model has learned a useful representation of the input data by drawing random samples from the latent space and comparing them to

a subset of reconstructions (see Chapter 5.3). After the model is trained on the experimental data (1.3×10^6 data points), the encoder embeds the data during inference onto a learned latent space. The algorithm then segment the continuous latent space into discrete behavioral motifs using an HMM [88], thereby treating the underlying dynamical system as a discrete-state continuous-time Markov chain. Comparing the HMM to a k-Means clustering, I found that HMM is consistently 5% more accurate in the metrics used (see appendix table A.1).

Figure 5.1 shows an example of a time series of egocentrically aligned DLC virtual marker with 100 data points. The orange line represents a full walking cycle, which represents a phase block. More specifically, I define a phase block as a full sinusoidal phase from 0 to 2π for a walking movement. The walking pattern is aligned with the inferred motif sequence, where each motif starts at a specific phase of the input signal. The video frames illustrate the corresponding walking cycle within the phase block. I will investigate this structure further in the next section to reveal that this cyclic form is also captured within the dynamical embedding.

5.1.4 Learning motion structure

5.1.4.1 APP/PS1 mice

To showcase the effectiveness of the VAME method in uncovering motif structure, I used four transgenic (tg) mice with beta-amyloid deposits in the cortex and hippocampus that carried human mutations in the APP and presenilin 1 gene (APP/PS1) [89]. These mice were compared to four wildtype (wt) mice housed under identical conditions. The APP/PS1 mouse line has been reported to exhibit several behavioral differences [90], such as motor and coordination impairments [91], changes in anxiety levels [92], and deficits in spatial reference memory [93]. This dataset was well-suited for the application of unsupervised behavior quantification as the differences could only be detected in specific repetitive tasks, not in open field tests [4].

5.1.4.2 General locomotor variables

I analyzed general locomotor variables to determine if there were any noticeable differences between the animals, focusing on speed, distance travelled, and time spent in the

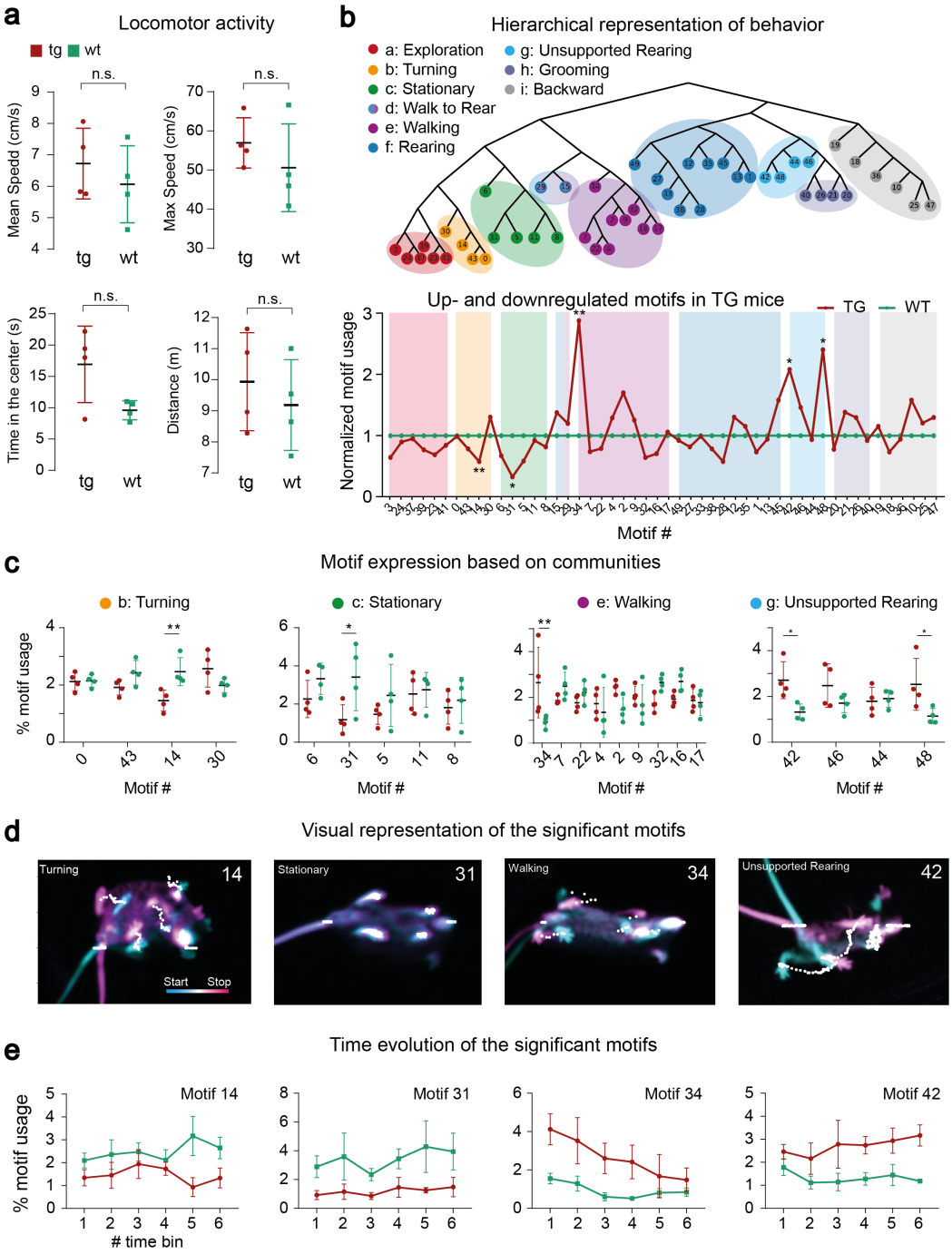

Figure 5.2: **Quantification of behavior using VAME and hierarchical community clustering.** (A) Locomotor activity of tg (N=4) and wt (N=4) animals. (B) Representation of behavioral motifs in a hierarchical structure. The color grouping on the tree illustrates clusters of motifs belonging to the same observable behavior category. Below a representation of the up- and down regulated motifs for the tg animals are shown. (C) Quantification of motif usage organized by communities with the highest differences between the tg and wt phenotypes in communities b, c, e, and g. (D) Examples for the communities b, c, e, and g. (E) Binned motif usage evolution over time for the full experiment within the most significant motifs (modified from [13]).

center (Figure 5.2 A). The average speed during the trial was 6.12 ± 1.36 cm/s for wildtype animals and 6.84 ± 1.57 cm/s for transgenic animals, with a maximum velocity of 50.61 ± 12.47 cm/s for wildtype and 57.14 ± 8.91 cm/s for transgenic. The average time spent in the center, calculated from center crossings, was 9.92 ± 1.81 seconds for wildtype and 17.14 ± 7.79 seconds for transgenic animals. The average distance travelled was 9187.44 ± 1266.4 cm and 9937.07 ± 1367.08 cm for transgenic and wildtype animals, respectively. No statistically significant differences were found between the groups for all measures, but a trend was observed for transgenic animals to move at a higher speed and spend more time in the center, as previously reported [94, 95, 4].

5.1.4.3 Human expert categorization of the behavior

To determine if behavioral differences were noticeable through human observation in our experiment, I conducted a survey where trained experts in behavioral neuroethology classified the phenotype based on video recordings. Eleven experts participated in the survey, where I created a blind online questionnaire for them to watch all videos and make a decision (refer to Appendix A.4). Experts with prior knowledge of APP/PS-1 had slightly higher classification accuracy ($50.98\% \pm 11.04\%$ for experts, $42.5\% \pm 15.61\%$ for non-experts), but overall classification accuracy was at chance level ($46.61\% \pm 8.41\%$) for all participants. This result aligns with previous findings of behavioral homogeneity between the two animal groups [94].

5.1.4.4 Inferring the latent representation

In the experiment, the first 25 minutes were allocated for the animals to get accustomed to the experimental environment. During the next 25 minutes, the behavioral structure was identified by applying VAME to the entire cohort of animals. The VAME approach aimed to infer the latent representation for each animal. The size of the latent dimension was determined by comparing the difference between the input and reconstructed signals. This parameter played an important role in controlling the amount of information flow between the encoder and decoder networks. By keeping the bottleneck small, the encoder learned to extract the most important features from the input signal. Afterwards, an HMM was applied to the latent representation and 50 motifs were inferred for each animal. The number of motifs in the dataset was determined using a similar method as

described in [22]. The HMM was used to infer 100 motifs and any motif with less than 1% usage was considered noise. This resulted in a total of 50 motifs. The motif usage is visualized in Figure 5.3 and showed that the first 10 motifs had a high usage, gradually declining until reaching the 1% threshold (48 motifs) and continuing to drop until reaching almost 0% usage (100 motifs).

Sup. Fig. 2: Sorted motif usage for all animals.

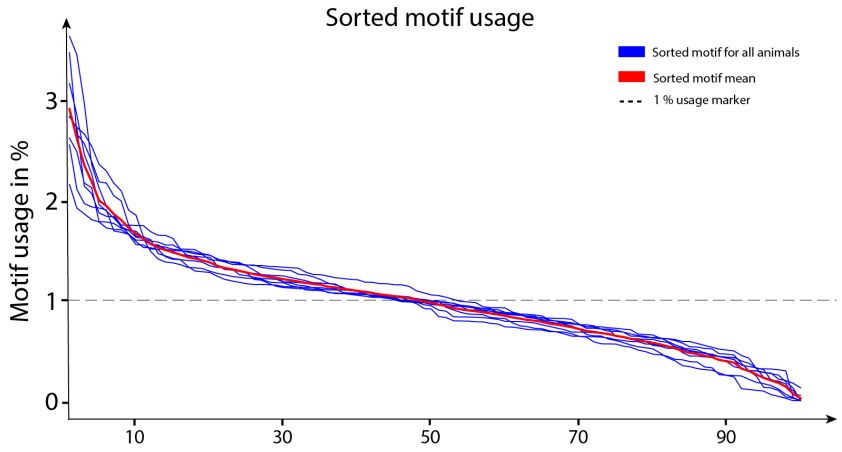

Figure 5.3: **Sorted motif usage to determine the optimal number of behavioral motifs in the dataset.** The blue lines depict individual data for all animals (n = 8), with the red line representing the mean usage. The dashed line signifies the 1%′ usage threshold, which is attained at approximately 50 motifs.(modified from [13]).

5.1.4.5 Inferring the hierarchical representation

A hierarchical tree representation $\mathcal{C}$ was created to identify communities within the discrete motif time series (Figure 5.2, B (top)). By comparing the branches of the tree with the corresponding motif videos, similar behavioral motifs could be identified in the resulting nine communities. They are denoted from a to i and each represented a cluster of movements that can be simplified into macro actions such as rearing, turning, and walking. Motifs within each community can be considered as a subset of these actions ($c \in \mathcal{C}_i$, with $i = a, ..., i$). The communities detected by VAME demonstrate a multi-scale behavioral representation. To better understand each community, they were visualized along with their respective DLC trace and further described in the Appendix section A.5.

5.1.4.6 Identifying differences between mice

I attempted to identify differences between tg and wt mice by looking for motifs and communities that were either up- or down-regulated (Figure 5.2, B (bottom)). The usage of each motif was calculated as a ratio and normalized against the wt group. I found that the communities for *Exploration*, *Turning*, and *Stationary* were down-regulated, while the communities for *Walk to Rear* and *Unsupported Rearing* were up-regulated. Some communities had motifs with differing usage, but there was no significant group difference that could be detected. To check for differences between the tg and wt mice, I examined the usage of up- or downregulated motifs/communities. A multiple t-Test was performed, and statistical significance was determined using the Holm-Sidak method with $alpha = 0.05$ (* $= P \leq 0.05$, ** $= P \leq 0.01$). The results showed significant differences in five motifs. The *Turning* and *Stationary* motifs were more prominent in wt mice, while tg mice showed more of the *Unsupported Rearing* and *Walking* motifs (referenced by arrows in Table 5.1 and visualized in Figure 5.2 C). The visual representation of these motifs can be seen in Figure 5.2 (D), where the start and end frames of a random motif episode are colored cyan and magenta respectively, with white dots representing the positions of the DLC virtual markers over time. The motifs are described in detail in appendix section 2.

To study the stability of differences throughout the experiment, I divided the experiment into six equal sections (as shown in Figure 5.2 E). I analyzed the consistency of the usage of the five most notable motifs over time. My results revealed that the utilization of these motifs remained either consistently increased or decreased throughout the experiment. This finding suggests that the differences in motif usage between the two groups (tg and wt mice) were stable over the duration of the experiment, providing evidence for the validity of the results obtained.

Table 5.1: Distinctive motifs that show significant differences between tg and wt. (from [13]).

Motif	Community	Mean Usage tg (%)	Mean Usage wt (%)	p-Value
14	Turning	1.4	2.4 ↑	0.005
31	Stationary	2.6	3.4 ↑	0.008
34	Walking	1.4 ↑	0.9	0.0003
48	Unsupported Rearing	2.5 ↑	1.1	0.006
42	Unsupported Rearing	2.7 ↑	1.3	0.008

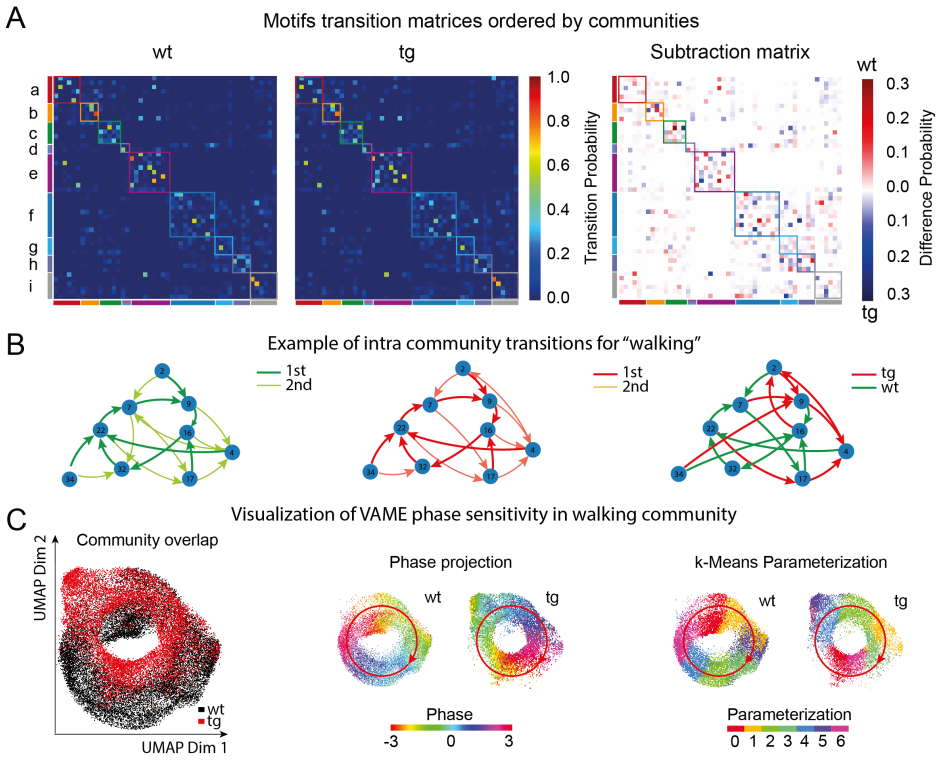

Figure 5.4: **Identification of transition structure and locomotion patterns.** (A) Left: Transition probability matrices arranged by communities for the wt and tg groups. Right: Difference plot comparing both matrices. Squares along the diagonal indicate community grouping. (B) Illustration of an intra-community transition graph for the walking community. The first and second highest transitions for both groups (left, middle) and the greatest difference in transition (right) are presented. (C, left) Joint UMAP embedding of points belonging to the walking community in a wt (19.783 points, black) and a tg (13.264 points, red) mouse reveals a circular structure. (C, middle) Projection of the mean phase angle of the horizontal hind paw movement onto the embedding displays the cyclic phase space of the walking movement in both animals. (C, right) Parametrization of both point clouds with k-Means shows blocks organized around the cyclic structure. The red arrow indicates the phase direction. (modified from [13]).

5.1.4.7 Analyzing the temporal structure of behavior

As I have shown above, the VAME framework provides a method for categorizing animal behavior into discrete representations, which are organized at multiple levels of resolution, from individual movements to larger-scale community behavior patterns. The temporal structure of behavior can be studied by analyzing the probability of transitions. This analysis can be conducted either at the community level or the level of individual behavioral motifs. Here, I created the transition matrices for both wt and tg animals, which were sorted based on the community structure, as shown in Figure 5.4 (A). The results showed that both wt and tg animals exhibited a similar pattern of transitions, which was

in agreement with prior observations of their behavior in an open field setting [4, 94, 95]. To further investigate any differences in transition probabilities between the two groups, a subtraction matrix $\mathcal{T}_{sub} = \mathcal{T}_{lk}^{WT} - \mathcal{T}_{lk}^{TG}$ was created to illustrate which transitions were more prominent in wt (Red) or tg animals (Blue). The analysis revealed significant differences in the usage of transitions within communities, with the most notable differences appearing in the "Stationary" and "Walking" communities, as shown in the Appendix A.6.

5.1.4.8 Investigating the cyclic nature of locomotion

I conducted a detailed investigation into the Walking community to better understand the differences in transitions. When analyzing the highest transitions on the Markov graph, I found a cyclic structure, where different walking motifs are more heavily used by both experimental groups (Figure 5.4 B). To understand this structure, I embedded the latent vectors of the Walking community onto a 2D plane using UMAP (Figure 5.4 C). Next, I visualized the UMAP for two example animals from each group. To confirm the cyclic nature of the structure, I decoded all points back to the original marker movement traces and computed the mean phase for the hind paw movement using Fourier transformation. The results showed that the phase angle follows the curve of the cyclic embedding (as indicated by the red arrow). To quantify the structure in both animals, I applied k-Means clustering, resulting in discrete clusters organized along the cyclic embedding. This type of pattern is known to emerge from oscillatory dynamics modeled by RNNs [96, 97]. A recent study of Drosophila locomotion also described cyclic representation of walking behavior, and my findings confirm the existence of this representation in rodent locomotion as well. The results from Figure 5.4 (C) were utilized to identify specific locomotion patterns within the "Walking" community using the representation learned by VAME. I designed an algorithm to identify reoccurring motif patterns and count their occurrences. This led to the detection of 22 sub-patterns that were common among all animals. Further analysis showed that four sub-second sequences were more frequently used in either wt or tg mice (as determined by an unpaired t-Test). The most prominent sub-pattern for wt animals was the sequence $\{16, 32\}$ and $\{32, 22, 7\}$, which can be observed as a strong transition in the difference graph. Conversely, the strongest pattern for tg mice involved the sequence $\{2, 9\}$, which is also evident as a strong transition in the difference graph for the tg group (Figure 5.4 B). These findings highlight the capability of

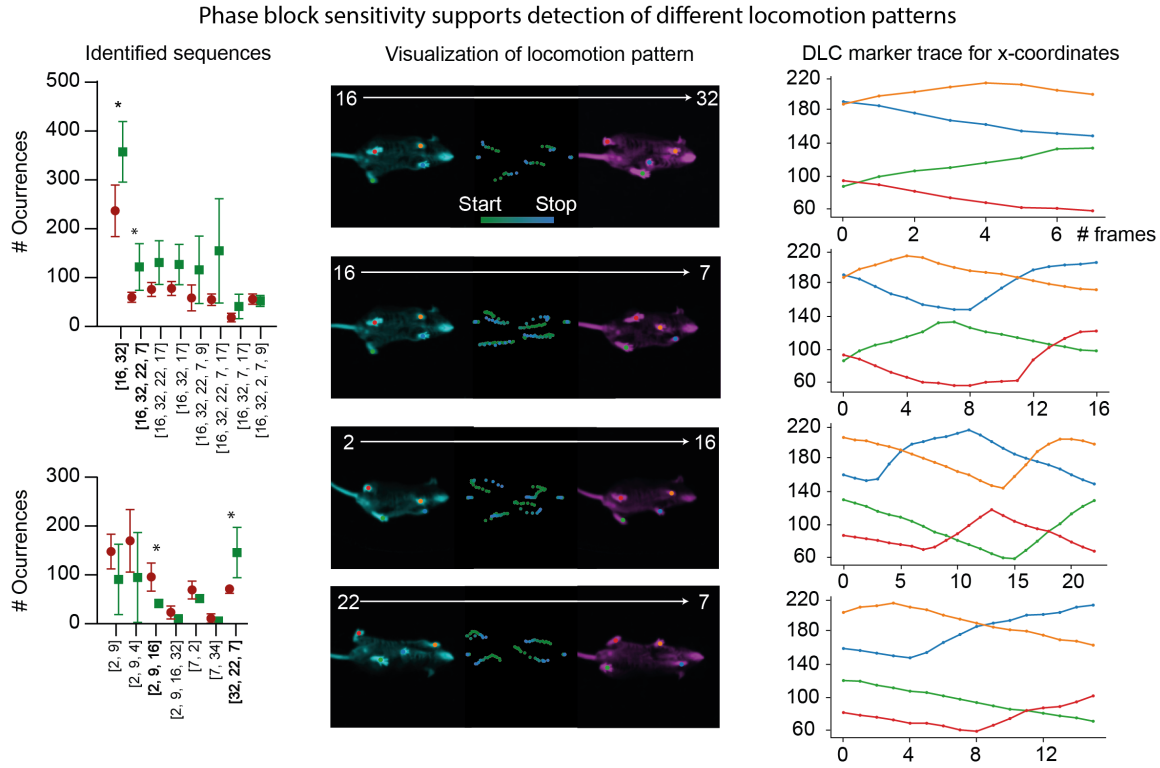

Figure 5.5: **Identification of the locomotion structure in both groups.** On the left, the discerned locomotion sequences are illustrated for two specific combinations of motifs from the walking community. In the middle, four locomotion patterns that exhibit significant differences are visualized. On the right, the corresponding DLC traces are displayed. Error bars represent standard deviation (modified from [13]).

VAME to effectively capture patterns related to transitions between motifs, particularly in the context of locomotion behavior due to its phase sensitivity (as illustrated in Figure 5.5).

5.2 Latent dimension analysis

The VAE employed in the VAME model allows for latent interpolations, a technique that enables dissecting behavioral differences on the sub-motif level. For this, I paired two animals according to the similarity of their phase angle within the same walking motif (phase angle wt: - 2.95, phase angle tg: -2.81). When observing the latent vectors underlying these motifs, I found a high agreement in all latent dimensions. However, I identified several dimensions with a deviation between both latent vectors that encode for either frequency shifts or other transformations (Figure 5.6 A). This was confirmed using latent interpolation, a technique that allows varying coordinates continuously in the latent

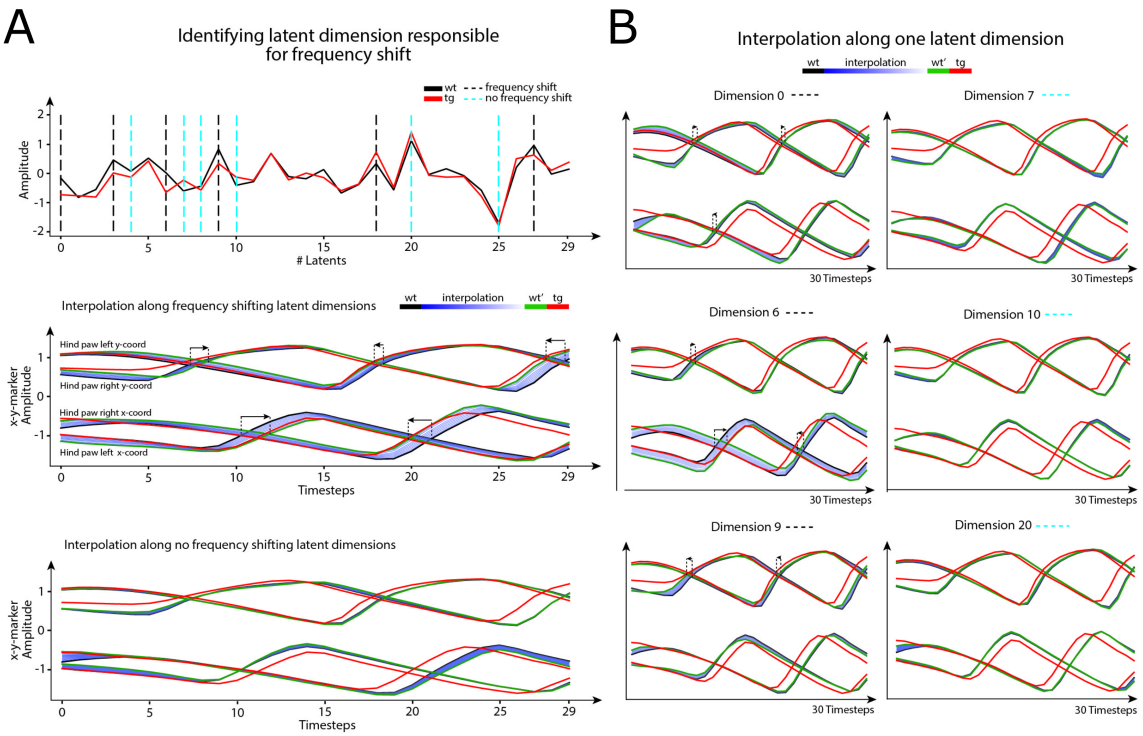

Figure 5.6: **Examination of the latent dynamics within the walking community.** (A, Top) Latent vectors of two cluster centers synchronized in phase for both animals. (A, Middle, Bottom) Interpolation of latent representations between paired cluster centers for both animals, along dimensions encoding frequency shifts (A, Middle, indicated by arrows) or dimensions encoding transformations that do not impact walking frequency (A, Bottom). (B) Latent interpolation along individual dimensions.

space and to decode realistic input traces from each step along the interpolation path. This is possible due to an advantageous property of the VAE, allowing the model to learn a smooth latent space from which previously unseen input traces can be generated (see section 5.3). When interpolating over all dimensions that encode for transformations involving a frequency shift, I was able to transform the movement traces from one animal to the target frequency of the other animal (Figure 5.6 A, Middle). Note, that other signal properties were unchanged, as I only interpolated over the dimensions indicated with black dashed lines. When interpolating over the dimensions that are not involved in the frequency shift altogether, I found specific transformations of the input signal. For example, I see an increase of the amplitude at characteristic phases of the movement, for a specific set of pose tracking markers involved (Figure 5.6 A, Bottom). This approach allows understanding specific and subtle differences between movement traces and allows researchers to pinpoint towards nuances of behavior that are non-trivial to uncover otherwise. Note that this can be also done with more complex behavioral time series, like

latent interpolation of, for example, walking to rearing, stationary to grooming, or others.

5.3 Generative aspects of the model

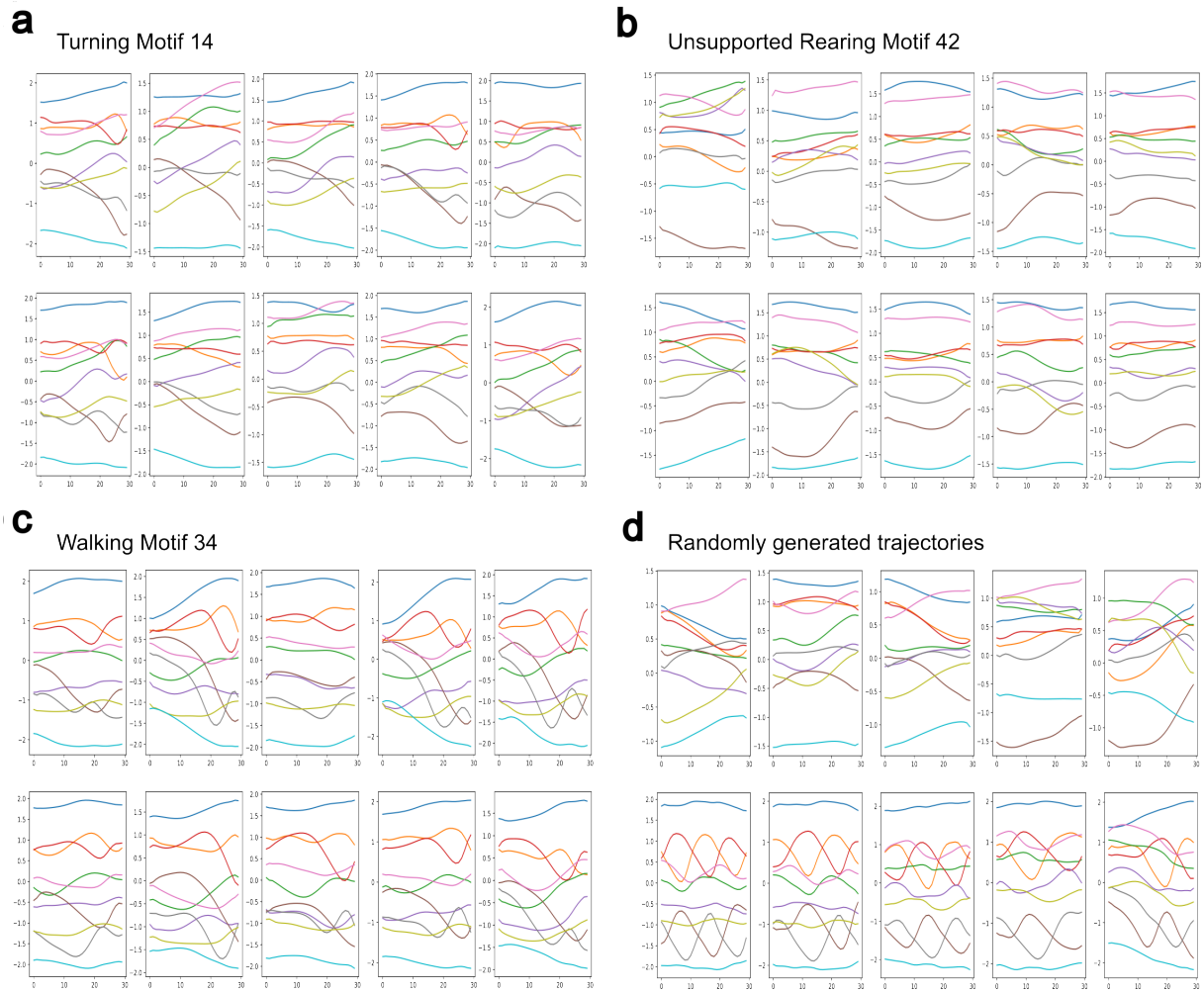

Figure 5.7: **Generative capabilities of VAME by sampling from the latent distribution.** (a) Random generative sample of the *Turning* motif 14. (b) Random generative samples of the *Unsupported Rearing* motif 42. (c) Random generative samples of the *Walking* motif 34. (d) Generated samples obtained by fitting a Gaussian mixture model on the latent space (modified from [13]).

Another advantage of VAME is the generative capability of the VAE. The strength of generative models is their ability to learn the distribution $p(x)$ of the data, as I have introduced before, and to generate new, unseen samples from this distribution. This capacity can be used to demonstrate the model's capability to represent the data distribution accurately. In Figure 5.7, I showcase VAME's capabilities to generate specific samples from the motif distribution and random trajectories. To verify that the model

has successfully learned the data distribution, I performed ex-post density estimation using a 10-dimensional Gaussian mixture model [98]. From the fitted density, I was able to sample data points that can be transformed into input trajectories. With this capability, VAME can internally verify that it can generate realistic synthetic samples from its learned distribution. This approach can also be used to validate individual clusters by fitting the density on regions belonging to specific motifs only. In Figure 5.7 (A-C), I use three exemplar motifs and sample from their latent distributions. By using the reconstruction decoder of VAME as a generative model, I can demonstrate that the distributions are well-defined as the decoder reconstructs similar trajectory samples. It is important to note that these samples were not decoded from any input sample and are entirely generated from the learned latent distribution. Figure 5.7 (D) shows trajectory samples generated from randomly sampled data points of the latent distribution.

5.4 Quantitative comparison with other methods

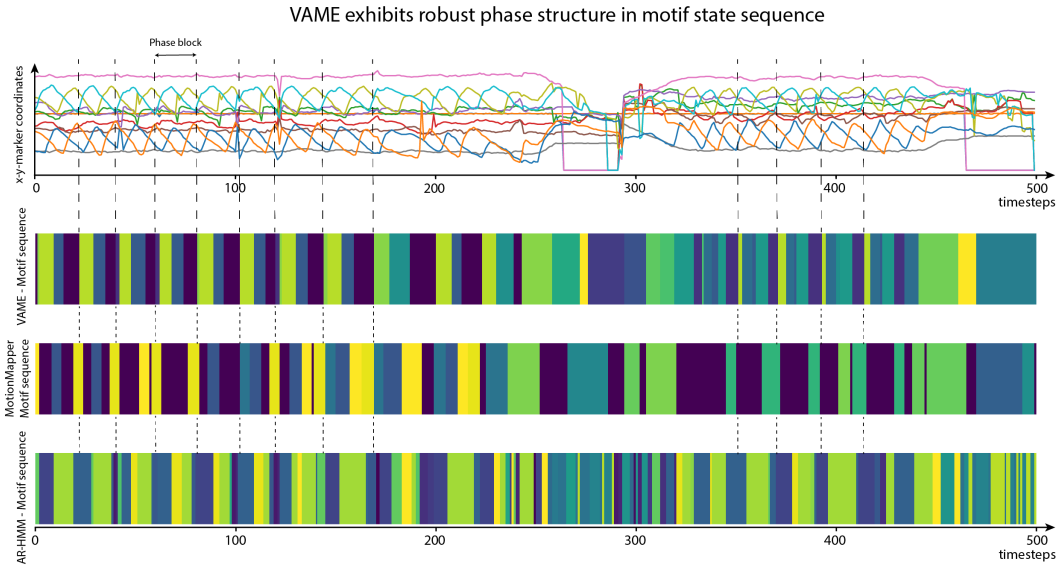

Figure 5.8: **Qualitative comparison with MotionMapper and AR-HMM.** The illustration showcases a sample trace of the input time series along with the motif segmentations obtained from the VAME, MotionMapper, and AR-HMM methods (modified from [13]).

There are numerous techniques available for measuring and quantifying behavior in various model organisms, which all contribute to the collection of insightful neuroethological data. These methods play a critical role in advancing our understanding of the

relationship between behavior and the underlying neural activity, providing valuable insights into the mechanisms of behavior. However, the approach taken by VAME is distinct from other methods, as it utilizes the VAE framework combined with powerful autoregressive models from deep learning. To assess the effectiveness of VAME in comparison to other methods, a qualitative and quantitative comparison was conducted with two established and commonly used approaches, namely AR-HMM (MoSeq) and MotionMapper [22, 21]. The results of this comparison provide a comprehensive understanding of the strengths and limitations of each method and highlight the potential benefits of using VAME for future behavioral quantification studies.

5.4.1 Setup and parameter settings

First, I will outline the configuration of the remaining two techniques. To compare VAME with the AR-HMM, the original codebase provided by the authors under a non-disclosure agreement was utilized. The default parameter values were used for all parameters (γ = 999, N_{lags} = 3, ν = 4), with the maximum number of states set to the corresponding cluster size $k = 50$. The sticky parameter setting was employed, with the value of κ set to the number of data points, as recommended by the authors in the usage documentation within the MoSeq repository. To compare my model with the MotionMapper framework, I utilized the original codebase supplied by the authors available on GitHub. The input signal was first transformed into the time-frequency domain using the Wavelet transform, resulting in 15 frequency bins in the range between 0 and 30 Hz. A two-dimensional t-SNE embedding was then derived from the stacked spectrogram (with a perplexity of 32 and a learning rate of 200, after 3000 iterations). The watershed segmentation of the embedding space was adjusted to match different numbers of k clusters.

5.4.2 Qualitative and quantitative analysis

5.4.2.1 Visual contrast between methods

Figure 5.8 provides a qualitative comparison of the results obtained from VAME with those of the other two methods. I trained all three models on the data and segmented

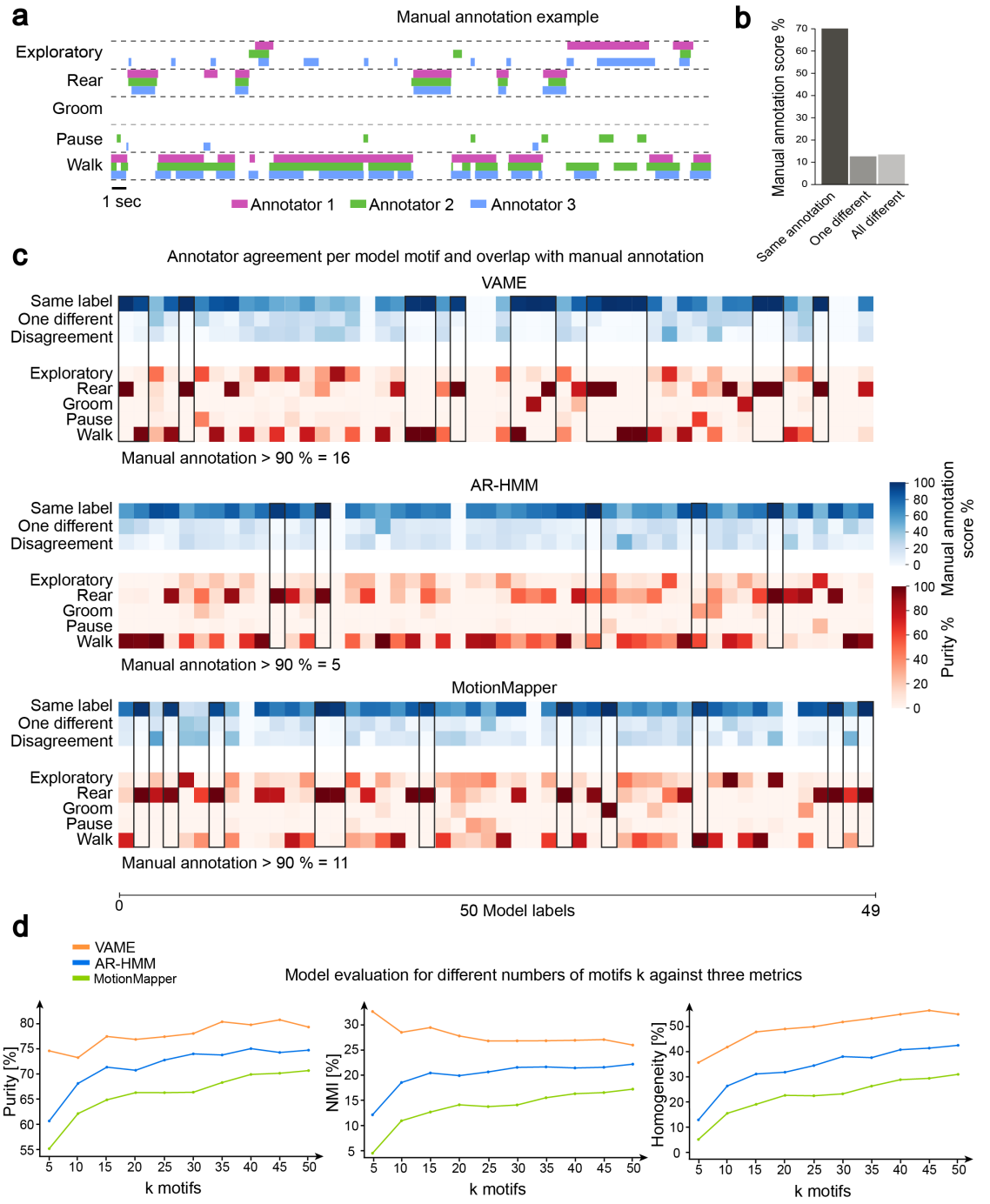

Figure 5.9: **Annotated dataset and model comparison based on annotator agreement.** (a) Intersection of labels manually assigned by three experts. (b) Discrepancies in manual annotation. (c) Confusion matrices illustrating annotator variability (blue) and agreement between 50 model motifs (VAME, AR-HMM, MotionMapper) and 5 manually annotated labels (red). Empty columns indicate motifs absent in the annotated benchmark data. (d) Evaluation of the model using three metrics: Purity, NMI, and Homogeneity (modified from [13]).

them into a similar number of behavioral motifs (VAME: 50, AR-HMM: 50, MotionMapper: 51). The figure depicts a trace that has been aligned with the motif sequences, highlighting two instances of walking and rearing. As a reference, the x-coordinate of the

left hind paw (indicated by an orange marker) was used to define a phase block, as indicated by the dashed lines between time step 0 and 200. A visual inspection of the results showed that VAME motifs match the phase of the signal more accurately compared to the other methods. The AR-HMM motif sequence exhibits more frequent state switches, whereas the MotionMapper motifs tend to last longer.

5.4.2.2 Benchmark dataset for quantitative evaluation

The validity of VAME, AR-HMM, and MotionMapper was evaluated through the creation of a manually labeled dataset (see subchapter 4.4). The dataset was generated from a video of a wt animal moving freely and consisted of 20,000 frames (approx. 6 minutes in length). The video was annotated by three human experts with training in behavioral neuroscience, using five behavioral labels (Walk, Pause, Groom, Rear, and Exploratory behavior) (Figure 5.9 (A)). The agreement between the individual experts was quantified, revealing that 71.93% of the frames were labeled consistently by all three experts. However, the remaining 13.61% of frames were labeled similarly by only two experts, and 14.47% were labeled differently by all three experts (Figure 5.9 (B)). This highlights the considerable observer variability in behavior and the difficulty in assigning it to discrete labels [9, 10].

5.4.2.3 Evaluation of method performance

I evaluated the performance of the three models by training them on the full dataset and comparing their results with the manual annotations (Figure 5.9 (C)). The blue columns show the degree of agreement between the model's predictions and the manual annotations, while the red columns indicate the accuracy of the model's predictions based on the expert labels. If the agreement between the model and the manual annotations was over 90%, I marked it with a black box in both blue and red columns. VAME had 16 motifs with high agreement with the manual annotations, while MotionMapper and AR-HMM had 11 and 5, respectively. This suggests that VAME is more effective in identifying human-readable labels compared to the other two models. However, it should be noted that some columns in Figure 5.9 (C) are empty. This is because the models were trained on the full dataset, but the evaluation was only done on a smaller annotated dataset (0.8% of the full dataset). VAME had 5 empty columns, while MotionMapper and AR-HMM

had 2 and 3, respectively, which may indicate that VAME is more selective in detecting motifs and not all motifs are present in the smaller benchmark dataset.

5.4.2.4 Quantification using clustering evaluation metrics

To further investigate the overlap of each model with the benchmark dataset I quantified Purity, NMI and Homogeneity (see subchapter 4.4). Applying all three measures I found that VAME had the highest score for each measure (Purity: 80.65%, NMI: 28.61%, Homogeneity: 54.89%), when applied to a motif number of $k = 50$. In Figure 5.9 (D), I further showed that VAME achieves the best scores on all three metrics when measured as a function of motif number k. Interestingly, the performance of VAME stays stable even for small motif numbers compared to the AR-HMM and MotionMapper. Additionaly, I passed the original pose data also to a standard gaussian emission HMM and applied all three metrics to the outcome to rule out that the performance of VAME is only determined by the downstream HMM. Here, I found that the HMM performance is similar to MotionMapper and significantly lower than our approach.

Table 5.2: Quantitative model comparison based on an annotated benchmark dataset (from [13]).

k = 50	Abs. Purity	Abs. NMI	Homogeneity %
HMM	71.42	16.51	33.91
MotionMapper	70.67	17.35	30.82
AR-HMM	74.72	22.42	42.5
VAME	**80.65**	**28.61**	**54.89**

5.5 Latent projections and trajectories

The latent embedding produced by VAME can be visualized and analyzed by projecting it onto a two-dimensional plane. This visualization technique enables a more in-depth investigation of the underlying structure and allows for an assessment of the separability of the latent structure. Additionally, by projecting a sample trajectory, which is a trace with a temporal order, onto the two-dimensional plane, it is possible to determine if the projection exhibits spatiotemporal smoothness. This further enhances the interpretability of the latent embedding produced by VAME.

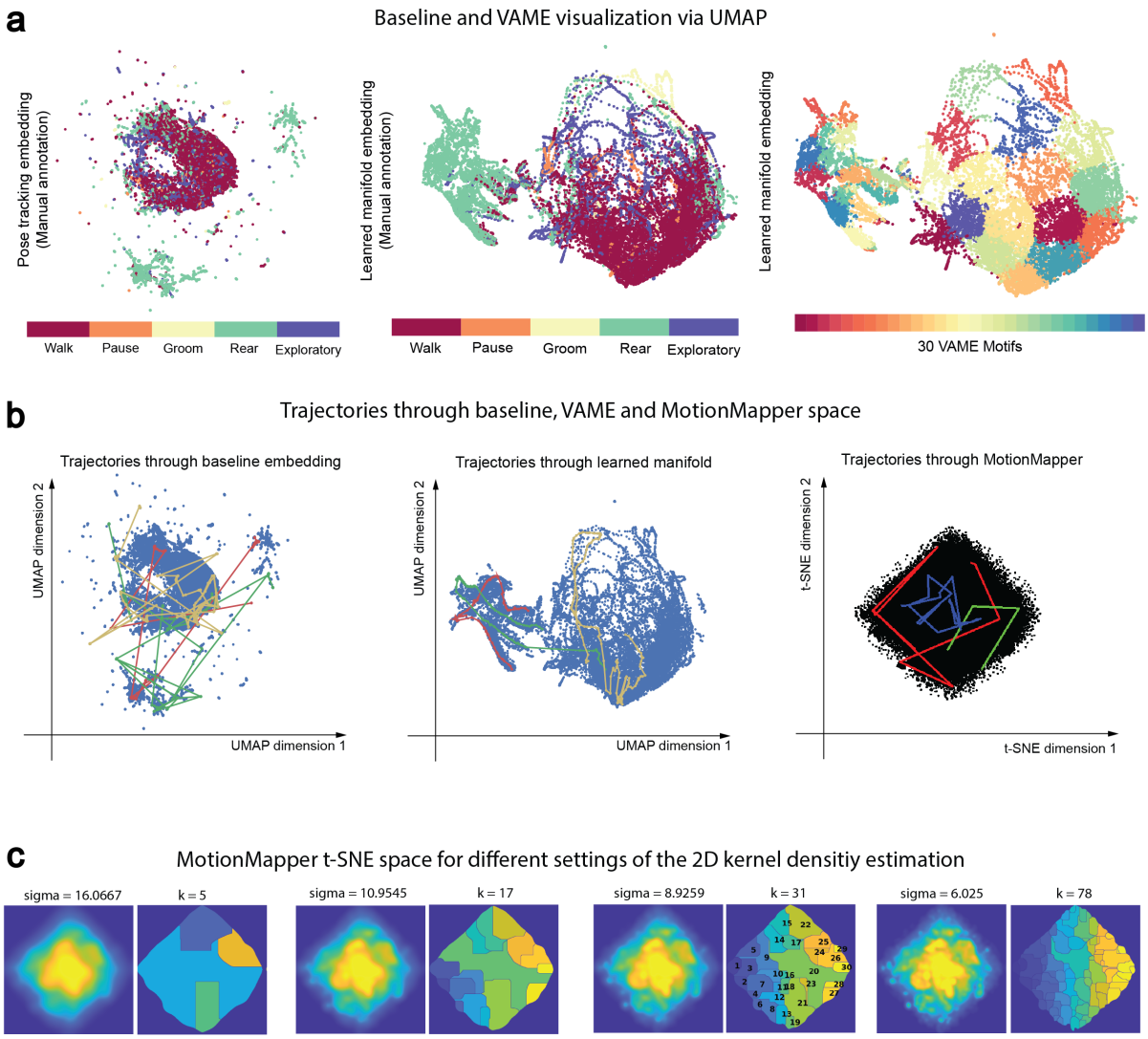

Figure 5.10: **Projection and trajectories of the VAME latent space in 2D, along with a comparison to the t-SNE projection.** (a) UMAP embedding of the marker position input series, with the latent representation encoded from the RNN color-coded for 5 manually labeled behavioral classes and further color-coded based on assignment into 30 VAME motifs. (b) Three illustrative paths of consecutive video frames traversing the UMAP embedding space of the spatial input series (DLC time series), the VAME embedding space demonstrating a smooth spatiotemporal representation, and t-SNE embedding obtained by MotionMapper with three exemplary paths. (c) Embeddings and segmentations acquired from MotionMapper for various settings of the 2D kernel density estimation parameter (sigma). (modified from [13]).

5.5.1 Two-dimensional latent embedding projections

In Figure 5.10 (A), the UMAP projection of the original egocentrically aligned virtual marker signal (baseline) is displayed for the manual annotated dataset. The UMAP projection of the latent vectors obtained by VAME for the same dataset is shown in the middle (human label) and right (VAME motifs) panels. The results suggest that the VAME

embedding is more compact and densely represented compared to the original signal, with a reduced scatter in the representation. This is indicated by the more clustered and less dispersed points in the UMAP projection of the VAME embedding.

5.5.1.1 Trajectories through UMAP and t-SNE space

I performed a test to evaluate the spatial and temporal coherence of the latent representation produced by VAME. I randomly selected three behavioral sequences, each with a duration of 1.5 seconds and plotted their trajectories on the baseline and VAME visualizations. I also compared the VAME trajectories to the ones produced by the MotionMapper by plotting them on top of a t-SNE embedding of the MotionMapper. The results showed that the trajectories in the VAME embedding followed a coherent path through the projected space, while the trajectories in the baseline signal appeared scattered. For MotionMapper, I observed even more scatter in the trajectories (as shown in Figure 5.10 (B)).

5.5.1.2 Center collapsing of t-SNE embeddings

With these results, I delved deeper into the embedding produced by MotionMapper to better understand its behavior on our dataset (Figure 5.10 (C)). Specifically, I evaluated the impact of changing the standard deviation of the smoothing Gaussian used in the two dimensional kernel density estimation, which is a tunable parameter in MotionMapper, on the t-SNE embedding and on the resulting watershed segmentations. The results showed that the choice of this parameter has a significant effect on the ability to form clusters; a high standard deviation results in fewer clusters that are detectable by the watershed segmentation and vice versa. In my analysis, I observed that the density in the middle of the embedding was overrepresented for segmentations with 5, 17, and 31 clusters and was only resolved when using a large cluster size (k=78). This finding suggests that the majority of the behavioral space is contained in a single central motif. This result is consistent with previous observations for pose estimation data collected from mice [34], as the sinusoidal signals have higher frequency-space similarity compared to behavioral signals measured in other organisms, such as fruit flies.

5.6 Neural activity and behavioral structure

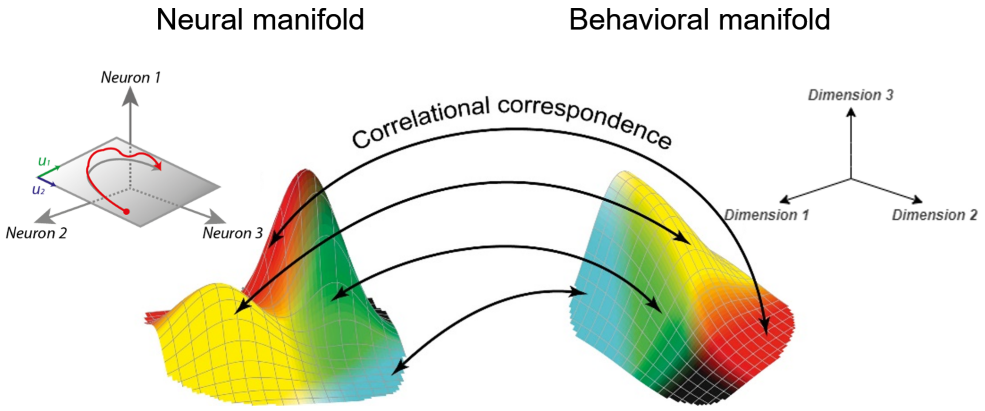

Figure 5.11: **Schematic view between a lower dimensional neuronal and behavior space.** A primary objective in modern neuroscience is to establish a connection between behavioral and neuronal spaces. (modified from [99]).

5.6.1 Connection between neural and behavior space

Establishing a clear connection between neuronal activity and behavior is a major challenge within neuroscience. A precise correlation analysis demands a comprehensive examination of both behavioral features and neuronal activity patterns. Figure 5.11 illustrates a simplified illustration of projections of the neuronal and behavioral space. In order to establish a causal link between neural activity and behavior, a well-structured analysis framework is necessary. This is especially critical in experiments that involve both spaces as various variables can be simultaneously measured. Without such a framework, these experiments can be inadequate. To analyze behavior, researchers can select a set of specific variables, such as position, speed, choice, or reaction time, which are recorded during the experiment. However, it is also possible to learn behavior characteristics through a lower dimensional behavioral manifold that incorporates latent variables of behavior. Similarly, neural activity can also be embedded into a lower dimensional manifold.

5.6.1.1 Manifolds description of low dimensional structure

Let's revisit the concept of a manifold, which has not been formally introduced in this work yet. It has appeared in other forms such as dimensional reduction, embedding or

lower dimensional space. In mathematics, a manifold is a topological space that closely resembles a Euclidean space [68]. The concept of a manifold is an important aspect in mathematics and is often used to describe topological spaces. A manifold locally resembles a Euclidean space, which is a mathematical structure with a regular grid-like pattern. This concept has been applied in the field of neuroscience to describe the underlying geometric structures in neural population activity. The term "neural manifold" refers to these geometric structures that are associated with various cognitive tasks. Despite its use, real-world neural data is not always a perfect representation of a mathematical manifold. This is primarily because of the presence of noise in neural activity and due to the fact that input sampling is often sparse. Despite these limitations, the term "neural manifold" has been used broadly to describe low-dimensional subspaces that underlie population activities embedded in high-dimensional neural state spaces. In neuroscience, the concept of neural manifolds has been applied in various brain regions, including sensory, motor, and cognitive regions. By understanding the low-dimensional structures underlying population activities, researchers can gain insights into the relationships between neuronal activity and behavior. This is an important step towards understanding the complex connections between the brain and behavior [100].

5.6.1.2 Techniques for manifold embedding

To better understand the high-dimensional nature of neural activity, researchers have focused on the observation that the activity can be represented on lower-dimensional subspaces, also known as neural manifolds. To unveil the structure of these neural manifolds, different dimensionality reduction techniques are applied to the analysis of neural data. One popular linear method used for this purpose is PCA, which provides a Cartesian coordinate system that describes the subspaces where the data resides. However, this method only provides a limited understanding of the intrinsic space defined by the data and its geometric properties. To gain a deeper insight into these properties, non-linear dimensionality reduction techniques are often required. There are many different non-linear dimensionality reduction techniques available, each with its own advantages and limitations. Some of the commonly used techniques include multidimensional scaling, t-SNE, and Isomap. It is important to note that the choice of technique will depend on the specific data being analyzed and the research question being addressed. In this

work, I have chosen to employ an approach that has been used successfully in previous studies with similar data, namely spectral co-clustering [101].

5.6.2 Aim of this study

By constructing a lower-dimensional representation of both neural activity and behavior, researchers can study the relationship between these two factors. In this section of this thesis, I utilize a dataset that was collected from a head-fixed mouse running on a linear treadmill while simultaneously imaging the hippocampal CA1 region of the mouse's brain. The continuous signals obtained from behavior tracking (using DLC) is grouped into discrete states through clustering of the latent vector obtained from VAME. This allows me to examine the correlation between the resulting behavioral states and neuronal activity at different hierarchical levels. The correlation is evaluated by assessing the similarity between the behavior, neuronal activity, or both. This approach provides a powerful tool for understanding how changes in neural activity correspond to changes in behavior, and how behavior affects the underlying neural activity.

5.6.3 Experiment and model

5.6.3.1 Treadmill experiment design

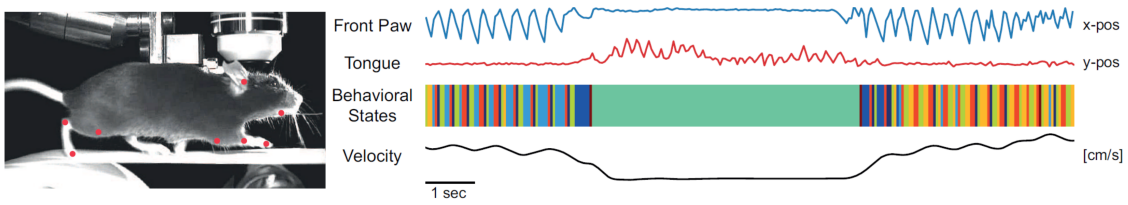

Figure 5.12: **Example data of a mouse video frame on the treadmill along with its corresponding motion signals.** An illustrative video frame displays an active animal with eight virtual markers. Two sample input sequences depict tracked joint movements. Behavioral states are derived from the VAME. The measured velocity, not included in the VAME input, is presented below (modified from [38]).

The Hippocampus is a brain area known to be involved in learning and episodic memory [1]. The mouse used in the study was food-deprived and trained to run head-fixed on a textured linear treadmill, which was 3.6 meters in length. The objective of the mouse was to learn a specific location on the treadmill, where it would receive a reward of liquid water upon licking once per lap. This led to a behavior pattern where

the mouse would repeatedly run approximately three rounds per minute. During this time, the population activity of the CA1 pyramidal neurons in the hippocampus was imaged at 15 Hz using a two-photon resonant scanning system [3]. The behavioral video recording was synchronized with the two-photon imaging, and a side view of the animal was captured at a frame rate of 25 Hz using an IR camera. An example video frame can be seen in Figure 5.12.

5.6.3.2 Preprocessing of the neural and behavioral data

The neural information underwent data preprocessing to ensure accuracy. This involved down-sampling the imaging stack to 5 Hz and correcting for any motion artifacts. To extract active temporal components, I used constrained non-negative matrix factorization [102]. This process resulted in a time series for each detected component, which represented the $\Delta F/F$ of the data. $\Delta F/F$ is a commonly used measurement in calcium imaging, which is a technique used to monitor the activity of neurons in the brain. It represents the fractional change in fluorescence intensity (ΔF) relative to the baseline fluorescence intensity (F), often expressed as a percentage. This measurement allows researchers to quantify changes in calcium ion concentration, which is a good indicator of neuronal activity over time. The onset of each peak in the time series was then identified using a threshold-crossing method. Each peak was assigned a weight based on its maximum value. In cases where multiple peaks occurred within a single transient, the weight of each onset was determined by the difference between the peak and the decay of the preceding peak, which was estimated using an exponential function. For behavioral pose extraction, I placed virtual markers on eight different body parts in 150 randomly selected video frames and trained a residual neural network (DLC) to assign the virtual markers to the entire video sequence [18].

5.6.3.3 Formalism of the used VAE-RNN

I constructed a recurrent neural network variational autoencoder to understand the structure of the temporal representation of a behaving animal. This framework later evolved into VAME and served as the foundation for the main publication. To better comprehend the ideas presented in this section, I will revisit the steps for embedding the behavioral

dynamics with a slight modification in the notation. The objective of the variational autoencoder was to learn a latent vector $\Lambda_t \in \mathbb{R}^d$, a d-dimensional representation of the behavioral dynamics. The input sequence $\mathcal{X}_t \in \mathbb{R}^{2n \times T}$ comprised of the $(x; y)$ coordinates of n marker positions, captured over a video sequence spanning time t to $t + T$. The sequential variational autoencoder then learns the mapping,

$$f_{enc} = \mathcal{X}_t \rightarrow \Lambda_t. \tag{5.1}$$

The design of this approach was inspired by [79], which proposed an unsupervised video representation learning method using a composite encoder-decoder model with LSTM units. However, to make the training process more efficient, GRUs were utilized in every layer of the autoencoder model instead of LSTMs [81]. The encoder f_{enc} was trained to generate a latent vector Λ_t that was fed into two one-layer GRU decoders. The first decoder aimed to reconstruct the sequence $\mathcal{X}_t$, while the second predicted the future evolution of the sequence $\mathcal{X}_{t+T}$. The model was trained using the Adam optimizer [64] with a fixed learning rate of 0.001 and with the mean squared error as the objective function for both reconstruction and prediction as well as the Kullback-Leibler-Loss for the VAE.

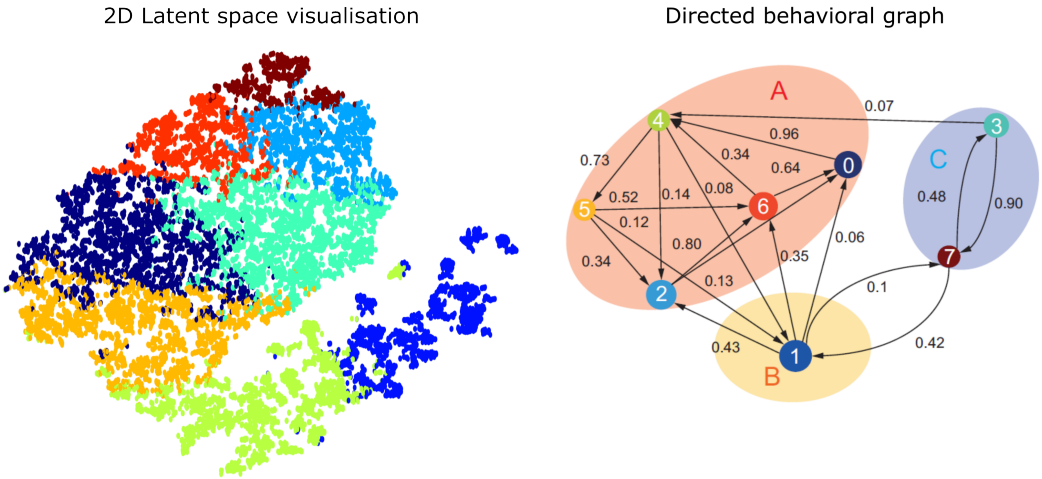

Figure 5.13: **Visualization of the two dimensional latent space and the corresponding transitions and communities.** Left: Display of the continuous latent embedding Λ obtained from VAME. Right: Construction of the directed behavioral graph $\mathbb{G}$ based on the state space B. For visualization purposes, only the edges with a transition probability greater than 0.05 are shown. (modified from [38]).

5.6.3.4 Behavioral state space and transition probability matrix

To determine the behavioral state space $B = b_1, \ldots, b_k$ in the dataset, the latent vector Λ_t was calculated for each data point t. With N frames in the full experiment, the resulting feature matrix $\mathcal{F}$ was $d \times (N - T)$ in dimensionality. K-Means clustering was performed on $\mathcal{F}$ to identify K behavioral states, and an example of a state sequence is shown in Figure 5.12. The transitions between the behavioral states were modeled as a discrete-time Markov chain, where the transition probability to a future state was solely dependent on the present state. This results in a $K \times K$ transition probability matrix $\mathcal{T}$, with elements

$$\mathcal{T}_{lk} = P(b_k|b_l) \tag{5.2}$$

being the transition probabilities from one state $b_l \in B$ to another state $b_k \in B$. The Markov chain, represented by equation (5.2), can be depicted as a directed graph $\mathbb{G}$ consisting of nodes $v_1, \times, v_k$ that are connected by edges with transition probabilities $\mathcal{T}_{lk}$. The size of each node represents the total number of times the corresponding behavioral state occurred throughout all N video frames. The latent space Λ as well as the graph are visualized in Figure 5.13. As conceptually described in chapter 4.3, the Figure clearly illustrates the discretization of the continuous latent space into a graph representation based on real data.

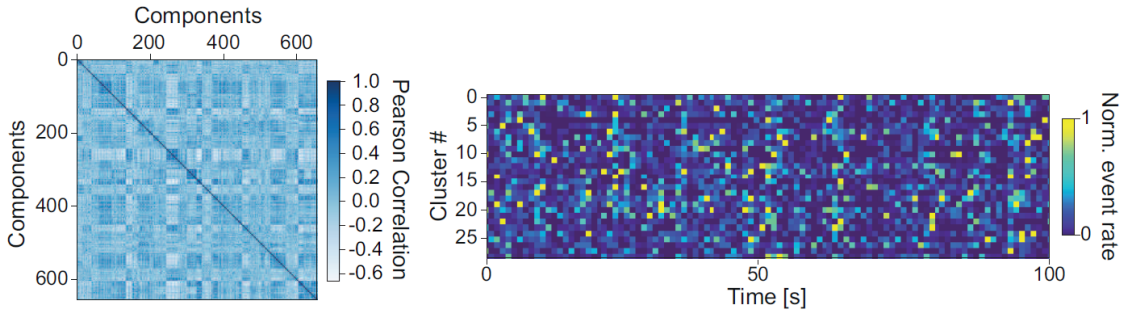

Figure 5.14: **Transformation of neural activity data to a correlation matrix** *R*.Left: Correlation matrix for 640 components. Right: An exemplary sequence of activity for each cluster (modified from [38]).

5.6.3.5 Dimensionality reduction of the neuronal recordings

To reduce the dimensionality of the neuronal data, I computed the pairwise correlations between activity traces of all 640 cells. This resulted in a correlation matrix R (Figure 5.14

(left)). To group components with similar values in corresponding rows and columns of R, I used spectral co-clustering [103] to cluster the matrix into a block-diagonal matrix Z. The number of clusters (M) was determined based on the specific structure of the neuronal recording. I then reduced the dimensionality of each cluster using factor analysis, which revealed a shared component that has been shown to play a significant role in behavior [101]. Figure 5.14 (right) shows an example of activity sequences for each cluster.

5.6.4 Neuronal-behavior Structure

The experiment involved imaging 640 active components from the hippocampal CA1 region, which were grouped into $M = 30$ neuronal clusters. The sliding window size was set to $T = 25$ (1 second of video data) and the dimension of the latent vector Λ_t was set to 20, resulting in a compression ratio of 20. To determine the behavioral states, a k-Means clustering assignment was performed with $k = 8$ based on the Elbow method. To validate the results, both the original video frames and velocity signal from the treadmill experiment (not used to train the autoencoder) were analyzed. The results showed that certain behavioral states were only active during running, while others were active during resting or reward taking periods (as seen in Figure 5.12).

5.6.4.1 Neuronal distance matrix

In order to merge the behavioral and neuronal information, I applied an information-theoretical approach to determine the relationship between the neuronal clusters and the behavioral states. I started by aligning the behavioral state sequence with the temporal resolution of the neuronal recording, creating a $M \times K$ matrix S. This matrix holds the average normalized event rate for each of the $M = 30$ neuronal clusters and each of the $K = 8$ behavioral states b_k (Figure 5.15 (left)). To calculate the dissimilarities between the rows of matrix S, I employed the Kullback-Leibler divergence. The Kullback-Leibler divergence measures the difference between two probability distributions, and I used it to determine the dissimilarity between the averaged normalized event rate of each

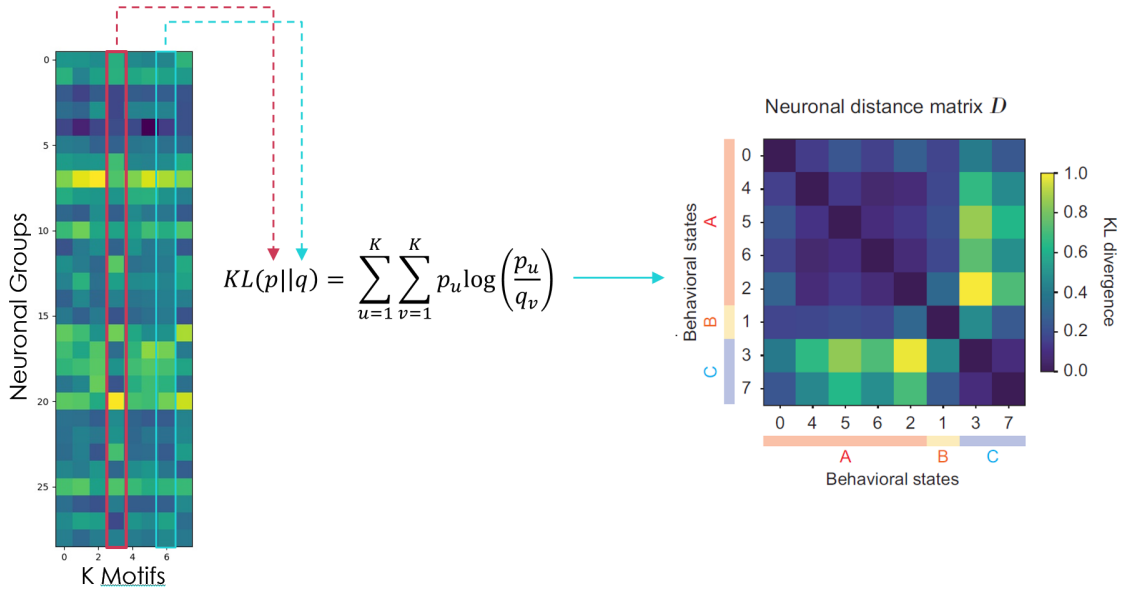

Figure 5.15: **Leveraging neuronal and behavioral groupings to establish a neuronal distance matrix denoted as D.** Left: $M \times K$ matrix S containing the averaged normalized event rate for each neuronal cluster and every behavioral state b_k. Right: Neuronal distance matrix D, which shows the Kullback-Leibler divergence for all combinations of behavioral states. Communities A, B, C are obtained from clustering of the hierarchical representation of $\mathbb{G}$ (modified from [38]).

neuronal cluster and behavioral state. The Kullback-Leibler divergence is defined as

$$KL(p||q) = \sum_k p_k \log \frac{p_k}{q_k} \tag{5.3}$$

where $p, q \in \{1 \dots K\}$ are modeled as the probability distribution of the neuronal clusters from two behavioral states. The calculation of the Kullback-Leibler divergence for all possible combinations of behavioral states result in a neuronal distance matrix D of size $K \times K$ (Figure 5.15 (right)).

5.6.4.2 Comparing trees for community detection

To gain a better understanding of the relationship between behavior and neuronal activity at different levels of hierarchy, I transformed the directed graph $\mathbb{G}$ into a binary tree $\mathbb{T}$. The transformation process involved iteratively merging two nodes (v_i, v_j) until only the root node v_R was left. In each reduction step, I had to select nodes i and j. To do this, I evaluated different combinations of remaining nodes using a cost function. I propose

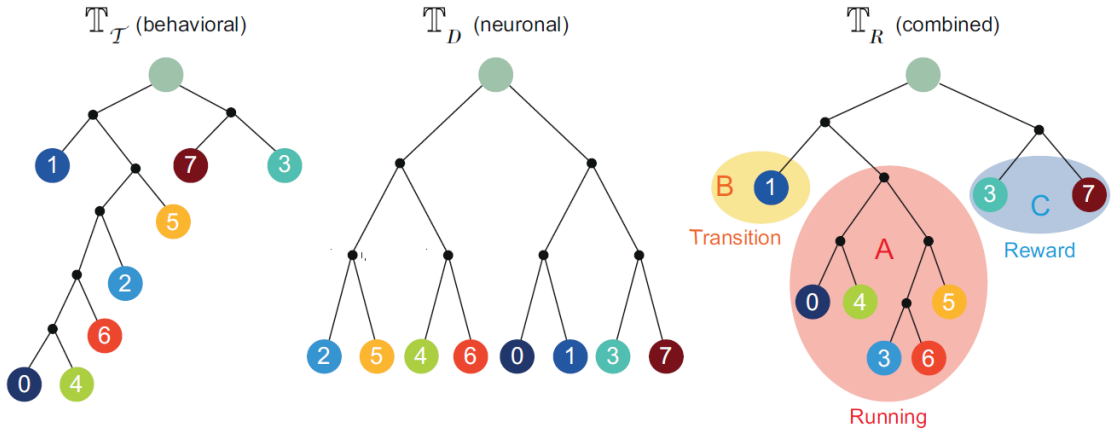

Figure 5.16: **A hierarchical depiction of the behavioral graph $\mathbb{G}$ using three distinct cost functions.** For the tree $\mathbb{T}_R$, communities A, B, C are allocated, each manually labeled as *running*, *reward*, and *transition* phases (modified from [38]).

three different cost functions, each with a different trade-off::

$$C_{\mathcal{T}} = \max_{i,j} \mathcal{T}_{ij} \tag{5.4}$$

$$C_{D} = \min_{i,j} D_{ij} \tag{5.5}$$

$$C_{ratio} = \max_{i,j} \left(\sum_{i,j} \frac{\mathcal{T}_{ij}}{D_{ij}} \right) \tag{5.6}$$

The first cost function (5.4) considers the behavioral similarity of nodes by merging the two nodes with the highest transition probability in $\mathbb{G}$. The second cost function (5.5) merges two nodes with the smallest Kullback-Leibler dissimilarity of the neuronal representation. The third cost function (5.6) is a combination of the first two cost functions, taking into account both the transition probability and the Kullback-Leibler dissimilarity for each node pair. After each reduction step, the matrices $\mathcal{T}$ and D were updated to reflect the changes in the merged nodes.

I have created three different trees (denoted as $\mathbb{T}_{\mathcal{T}}$, $\mathbb{T}_D$ and $\mathbb{T}_R$) using the three different cost functions (5.4), (5.5) and (5.6). The results of these trees are shown in Figure 5.16. To identify communities within the tree $\mathbb{T}_R$, I used the method of community structure detection suggested in [104]. By making a cut at the second hierarchical level, three communities were identified, each representing a stereotyped behavior such as running, reward, and transition phases.

Chapter 6

Discussion

6.1 VAME

6.1.1 Central contributions

The field of neuroscience and computational ethology is faced with a pressing need due to the limitations of current methods in capturing the full spatiotemporal dynamics of behavior and understanding its causal relationship with brain activity [10]. To address this, the field is rapidly utilizing a range of experimental approaches, including imaging, electrophysiology, and cell-specific interrogation methods to monitor neural activity in freely behaving animals [6, 105, 5]. This, along with the use of both new and traditional transgenic animal models, allows for a deeper investigation of molecular pathways in health and disease. However, all these approaches require a thorough and reliable dissection of behavior.

6.1.1.1 Understanding spatiotemporal patterns

In this thesis, I presented Variational Animal Motion Embedding, an unsupervised probabilistic deep learning method for discovering the spatiotemporal structure of behavioral from pose estimation signals. This method merges concepts from latent variable modeling, deep learning, information theory, and graph theory to create a probabilistic mapping between the data distribution and a dynamical embedding space. The VAME framework combines VAEs with an autoregressive model (biRNN), allowing the method to approximate the distribution and identify distinct behavioral motifs and dynamics through encoding relevant information. The approach sets itself apart from other models

by using a biRNN decoder that learns to predict the structure of the subsequent behavioral trajectory and regularizes the latent space. This results in the encoder learning more meaningful dynamical features from the behavioral time series. In this work, the term "behavioral dynamics" is used synonymously with time-dependent analysis of body part movements. The biRNN model within VAME performs a fit of the gated recurrent unit equations to the pose estimation signal, which describes the motion of the DLC markers via difference equations in a data-driven manner. Furthermore, I demonstrated the discriminatory power of VAME using a traditional transgenic model of Alzheimer's disease (the APP/PS-1 model) in a mouse model system that shows clear behavioral deficits in specific tasks but no differences in open-field observation [4, 95]. VAME can be applied to any species or behavioral task if continuous video monitoring can be provided [106].

6.1.1.2 Differentiation of phenotypes

My results indicate that VAME can successfully differentiate between a transgenic and wildtype group of mice, while no differences were detected by human observation. The focus was not on investigating the behavioral deficits related to Alzheimer's disease in the realm of learning and memory. However, the analysis still revealed significant findings in the small sample size, such as the higher usage of the "Unsupported Rearing" behavior and lower usage of the "Stationary" behavior, potentially linked to deficits in spatial orientation and habituation to the environment [92, 93]. Additionally, the low variance in the motif usage within each group highlights the robustness of the VAME method in detecting a consistent and stereotypical behavioral structure.

6.1.1.3 Enhanced sensitivity to signal phase

I discovered that VAME is particularly effective in identifying motif sequences from pose estimation signals due to its high sensitivity to the signal phase, achieved through its biRNN encoder and decoder. To demonstrate this, I plotted the phase angles onto a two-dimensional UMAP projection for the walking behavior, revealing a circularly organized pattern that precisely captured the natural limb movement cycle [107]. This makes VAME potentially valuable in detecting recurring locomotion patterns. Although this advantage has only been demonstrated in one behavioral community, it could be applied to identify differences in other movement types.

6.1.1.4 Hierarchical structure from motif sequences

I also explored in my work the hierarchical structure of the motif sequences learned from pose estimation signals using a tree representation within the VAME framework. In VAME, motifs are sub-patterns of macro behaviors organized in a Markovian graph, which can be analyzed based on transition and usage properties to identify different types of behaviors, such as locomotion. The tree representation converts the motif sequence into more comprehensible categories, such as "Walking" or "Rearing", allowing for easier interpretation. The approach I developed in this thesis for converting behavior sub-patterns into a tree structure can be utilized in other supervised or unsupervised techniques, thereby making the analysis a highly versatile tool.

6.1.2 Evaluation against current approaches

To understand how VAME performs against established and regularly applied approaches, I compared it qualitatively and quantitatively with MoSeq and MotionMapper. VAME is similar to both in terms of behavior segmentation. MoSeq uses an AR-HMM to segment behavior from a series of transformed depth images, incorporating spatiotemporal information. MotionMapper, initially developed for fruit flies, relies on t-SNE embeddings of wavelet transformations from high-speed camera images to identify high-density regions that are assumed to contain stereotypical behaviors. However, the reliance on spectral energy as a key input feature can limit the detection of low-frequency movements, which are more prominent in mice, and result in the loss of the full behavioral repertoire. MoSeq was first applied in freely moving rodents and was able to detect sub-second behavioral structures, but the AR-HMM resulted in many short and fast switching motifs, leading to uncertainty in animal action classification. Both are two widely used methods for capturing the behavioral dynamics of animals in different experimental settings. I compared the performance of these methods against VAME by training each of them on the dataset from this work and evaluating their motif sequence distribution using a benchmark dataset. Although each of these models learned a consistent motif structure, VAME outperformed the other methods by obtaining higher scores in all three metrics (Purity, NMI, Homogeneity). This might be due to the better embedding of spatiotemporal information and the higher phase sensitivity of VAME, which is not as strongly present in the

other methods. Recently, an independent group of scientists has published comparative results between VAME and other methods, including AR-HMM and Behavenet [22, 108], on a benchmark dataset for a hand-reaching task. Their findings fully support my own observation of higher performance levels of VAME compared to the other methods in terms of Accuracy, NMI, and Adjusted Rand Index. In particular, the combination of the video representation model and VAME achieved the best results [106].

6.1.3 Hyperparameter considerations

6.1.3.1 Size of latent dimensions

The choice of hyperparameters in VAME is a critical aspect that can impact the results of the model. Among the hyperparameters, the number of latent dimensions plays a central role in determining the amount of information the model can exploit. According to the information bottleneck theory [67], this number should be as small as possible to extract the most relevant information from the data. However, this decision is also influenced by other factors, such as the choice of the time window w and the number of marker coordinates m, which are used to condense the data into a vector representation z_i. An increase in w or m (or both) may require expanding the latent dimensions or preprocessing the data through a top layer neural network or other techniques like principal component analysis. When using VAME, it is crucial to adjust this number to meet specific requirements, as it significantly impacts the outcome of the model. To determine the appropriate number of latent dimensions, it is recommended to use a benchmark dataset and evaluate the reconstruction score.

6.1.3.2 Number of motifs

Determining the appropriate number of motifs is a challenging task that varies depending on the unique set of behaviors present in each experiment and animal. In this work, only motifs with a usage rate higher than 1% were considered after sampling 100 motifs from the embedding space and re-running the motif segmentation. However, it would be of great interest to identify motifs that are present in one group/animal but not in the other, as this would highlight significant differences in behavior between them. In

the current study, the data was homogeneous in terms of behavior, so that such motifs were not likely to be found. However, the VAME model is capable of finding these "out-of-distribution" motifs when they exist due to its latent variable modeling and variational autoencoding framework. When it comes to behavioral quantification methods, there is always a trade-off between the generalization of the motif distribution and the precision of individual behavior measurement. If the goal is to identify highly specialized individual behavior, it would be possible to parameterize both populations or all animals individually. However, the challenge lies in relating the motifs between populations/animals, as the motif mapping would change with each parameterization.

6.1.4 Constraints and opportunities

While VAME has demonstrated better performance compared to other unsupervised methods, it may not be the best option for all experimental settings. For cases where a complete understanding of the full behavioral repertoire is not necessary, supervised approaches such as SimBa, MARS, or DeepEthogram may be more appropriate [31, 14, 32]. These methods allow for the labeling of specific episodes of interest or rapid identification of similar frames in new data points, respectively. Another recently developed unsupervised approach is B-SOiD [34]. However, it operates differently from VAME in that it does not utilize a deep learning model and projects framewise into a UMAP representation, relying mainly on velocity feature signals for temporal information. When deciding on a behavioral quantification method, the specific goals and requirements of the experiment and species should be considered. VAME may be particularly useful for uncovering behavioral dynamics in a lower-dimensional latent space due to its ability to learn spatiotemporal information. Additionally, it has the potential to train a classifier on the latent vector information for quick assignment of VAME motifs to new data points or for closed-loop experimentations.

VAME has the potential for even greater resolution in the quantification of behavior when combined with three-dimensional pose information, as most behaviors are expressed in three dimensions [109, 110, 111]. The integration of 3D pose information is easily achievable within the VAME model. Moreover, when higher dimensional information is desired, such as cellular calcium responses or neurotransmitter dynamics, the VAME model provides a straightforward way to integrate these additional parameters.

6.2 Neural-behavior Representation

6.2.1 Central contributions

6.2.1.1 Correlation of behavioral states and neural activity

In the second part of this thesis, I introduced an innovative approach for the combined analysis of both behavioral and neuronal population data. This approach involves the conversion of continuous signals obtained from behavior tracking tools (DLC) into discrete behavioral states through clustering of latent vectors obtained from a sequential variational autoencoder (VAME). The resulting behavioral states can then be correlated with clustered neuronal population activity using a hierarchical approach which is driven by principles from information theory. This method has the potential to reveal the organization of behavioral states, as well as the structure of the underlying neuronal correlates. The cost function used for aggregating states plays a critical role in determining the insights obtained from the analysis. The approach provides a new perspective on how to understand the relationship between behavior and neuronal activity and has the potential to advance the field of computational ethology.

6.2.2 Tree transformation and community embedding

To demonstrate the efficacy of this approach, the analysis was performed on a dataset obtained from a mouse running on a linear treadmill while receiving liquid reward at a fixed location. The behavioral clustering resulted in a total of 8 distinct behavioral states, which were then grouped into three communities. These communities consisted of five states that were active during running phases, two states that were active during resting or reward-taking phases, and one state that was active during transitions between the aforementioned phases. Further investigation of these communities has the potential to uncover other sub-communities and behaviors, such as different running patterns.

6.2.2.1 Tree-Edit-Distance

The explicit mapping of a graph to a tree structure can provide additional benefits in behavior comparison and quantification between different experimental conditions, trials,

and animals. By transforming the graph representation of behavior into a tree representation, we can apply the Tree Edit Distance (TED) metric [112] to calculate the dissimilarity between two trees. For instance, if we have two graphs, $\mathbb{G}_1$ and $\mathbb{G}_2$), with their respective tree representations $\mathbb{T}_1$ and $\mathbb{T}_2$, we can use the TED to measure the differences between $\mathbb{T}_1$ and $\mathbb{T}_2$. This approach can provide valuable insights into the differences between behavior patterns in various scenarios and help in the systematic analysis of behavior across different conditions.

6.2.3 Hyperparameter settings and future enhancements

The proposed approach for behavioral clustering is dependent on two key factors: the choice of time window (T) and the number of clusters (k). These parameters can significantly impact the results of the clustering and therefore, care should be taken when selecting them. To further enhance the correlation between behavioral states and neuronal activity, the implementation of dynamic time-warping is suggested. This approach has been found to improve the correlation between behavioral states and neuronal activity [113].

Chapter 7

Conclusion

The field of computational ethology is rapidly expanding, driven by the increasing capacity to gather extensive behavioral data from laboratory animals. Machine learning plays a pivotal role in deciphering the complex structures within this data and understanding their correlations with neural activity. This work contributes to this growing field by introducing a pragmatic approach to identify patterns in behavioral motion structures. The method presented here leverages concepts from deep learning and variational auto-encoding, providing a reliable framework for extracting meaningful patterns from virtual markers recorded over time. The inclusion of recurrent neural networks (RNNs) adds flexibility to the methodology, allowing for robust feature extraction and simultaneous inference of discrete clusters and continuous representations. This dual capability enhances our understanding of the dynamic nature of animal behavioral. By imposing variational constraints on the distribution of these representations, the method enhances interpretability and encourages the extraction of more insightful quantities. The generative capabilities of the method enable the investigation of learned representations through the synthesis of synthetic data and the transformation of representations along their latent axes, offering valuable insights into behavioral dynamics and facilitating comparisons across different animals.

7.1 VAME: A framework for measuring animal motion

7.1.1 Dynamical embedding and behavior segmentation

This thesis delved into the complexities of measuring animal motion, specifically focusing on the challenges associated with capturing naturalistic behaviors. The examination

of three primary challenges — coordinated movements of diverse body parts, moment-to-moment labeling of behaviors, and the granularity of behavioral descriptions — serves as a foundational aspect of this research (1.3). The method developed in response to these challenges, known as Variational Animal Motion Embedding (VAME), introduced a new perspective that goes beyond predefined features, action categories and human-labeled categories. Specifically, VAME learns to represent a behavioral signal within its latent space and to reliably segment behavioral patterns on multiple hierarchical level. Compared to other methods discussed, VAME stands out for its proficiency in discerning subtle movement actions that converge at a higher hierarchical scale, forming macro actions recognizable to human observers. Consequently, the resulting behavioral patterns can be scrutinized at various scales and effectively compared across different animal subjects.

7.1.1.1 Dynamical embedding

The significance of dynamical embedding in unravelling the intricate and multifaceted nature of behavior cannot be emphasized enough. Within this context, VAME is a highly innovative approach. By harnessing the power of a Variational Autoencoder (VAE) combined with RNNs, VAME not only captures the underlying factors influencing animal movements but also embraces the dynamic nature inherent in behavior. This distinctive feature of VAME holds the key to a more precise understanding of behavior, surpassing the capabilities of other methods. More specifically, one of the superior capabilities of VAME lies in its ability to project behavior into a dynamical embedding, an attribute that sets it apart from conventional approaches and allows for a more refined measurement of behavioral patterns. This projection enables a level of precision not met by other methods, such as MoSeq or Motionmapper. Motionmapper projects information into a two-dimensional latent space; however, it falls short in capturing the structure of behavior. The majority of the information is projected into a two-dimensional Gaussian structure, hindering reliable reconstruction or segmentation of behavior (5.4). The latent space of VAME not only enhances the granularity of behavioral analysis but also facilitates a more comprehensive examination of the dynamic interplay between behavioral patterns and neural activity.

7.1.1.2 VAME as open-source tool

VAME emerges as an invaluable asset for behavior segmentation without the need for prior supervision. This tool stands at the forefront, facilitating an in-depth exploration of animal behavior as well as the intricate interplay between brain activity and naturalistic behavior. Its capacity to discern behavioral patterns from pose estimation signals, coupled with its adeptness at generalizing across diverse animals and experimental setups, positions it as an exceptionally versatile tool applicable to scientists across various fields, prominently within neurobiology. Furthermore, VAME's open-source accessibility and user-friendly design further amplify its impact, transcending traditional boundaries. By lowering entry barriers and fostering a collaborative environment, VAME has the potential to catalyse significant advancements in machine learning models tailored for computational ethology and neuroscience. The ripple effect of its widespread applicability not only enhances the tool's utility but also propels the broader scientific community towards more sophisticated and nuanced insights in the realms of behavioral analysis and neurobiological research.

7.1.1.3 Impact on translational research

Moreover, VAME's relevance extends to the investigation of neurodegenerative diseases, here with a particular focus on Alzheimer's disease (AD). The capacity to identify subtle changes in behavior in transgenic mouse models of AD not only introduces novel possibilities for early detection but also enhances our comprehension of disease progression. Moreover, it provides a valuable platform for exploring potential therapeutic interventions. The potential impact of VAME on translational research becomes apparent as it yields insights into the intricate behavioral changes observed in both mice and human AD patients. This not only enriches our understanding of the disease but also holds promise for the development of more effective diagnostic tools, treatment strategies, and management approaches. By offering a nuanced and comprehensive analysis of behavioral changes associated with AD, VAME contributes significantly to the broader field of neurodegenerative disease research, with potential implications for advancements in clinical practices and interventions.

7.2 Behavioral information beyond motifs

7.2.0.1 Essential need for strong metrics

Going beyond, it is crucial to underscore the imperative for more robust metrics and benchmark datasets in the realm of computational ethology. While assessing methods on a singular dataset offers valuable insights into their performance, it is essential to acknowledge that various methods may exhibit different performances across diverse types of data. As the arsenal of tools for computational ethology expands, there arises an escalating need to develop benchmarks and metrics that not only consider the structural aspects of behavioral patterns but also account for the representation of behavior within lower-dimensional spaces. The outcomes obtained through the application of VAME in my study highlight its potential in addressing these fundamental aspects. Nevertheless, it is imperative to note that these conclusions are derived from specific experimental setups and datasets. To conduct a thorough assessment of behavioral tools and identify optimal scenarios for their application, it is critical that future benchmarks undergo testing across a diverse range of scenarios. Presently, benchmarks predominantly concentrate on action categories or motif retrieval, constituting only one facet of computational ethology. It is noteworthy that different metrics need to be either incorporated or developed to provide a more comprehensive evaluation. Notably, my findings demonstrate that VAME excels not only in extracting human-readable behavioral patterns but also in generating meaningful embeddings of behavioral information. This capability is significant in its own right and proves valuable when comparing behavior to neural activity. To advance the field, it is crucial to construct benchmarks and datasets that encompass the multifaceted nature of modern approaches. Such an inclusive strategy will enable the development of even more advanced tools in the future. Ultimately, this comprehensive approach holds the promise of significantly enhancing researchers' ability to delve into and comprehend the intricate links between brain activity and behavior.

7.2.0.2 Integrating neural and behavioral dat

The use of computational ethology techniques, whether studying simple trained behaviors or complex spontaneous behaviors, in single animals or at a larger scale, will provide further insight into the structure and nature of behavior. However, to make sense of these

experiments and understand the relationship between behavior and brain activity, a comprehensive understanding of the animal's goals in generating a behavior is required. As a result of continuous advancements in brain recording and behavioral analysis technology, researchers are soon going to face the challenge of connecting complex and dynamic neural data with complex and dynamic behavioral data. The primary aim of future research will be to uncover behavioral representations that give us an understanding of how neural circuits generate behavior. To achieve this, it is important to determine what it means to "understand" the relationship between natural behavior and the brain. From a psychological perspective, understanding this relationship would involve a comprehensive explanation of the brain circuits that control a specific behavior, including testable predictions on how alterations in these circuits will impact behavior. On the other hand, ethologists aim to comprehend how behavior contributes to a species' survival in its environment, including how behavior evolves through natural selection and how it develops through the interaction of genetics and learning in each individual [10]. These various levels of explanation are interconnected and hold equal significance (Barlow, 1961).

7.3 Limitations

While VAME demonstrates considerable promise in advancing our understanding of animal behavior, it is critical to reflect, recognize and address certain limitations inherent in the framework, paving the way for future enhancements. One notable limitation lies in the variability of behavioral data. As animal behaviors can exhibit significant diversity, VAME's ability to capture and generalize patterns across different species or experimental setups may be constrained. However, this is true for every behavioral embedding framework to date. Hence, future approaches needs to overcome this limitation in order to really raise the threshold for modern tools. Interpreting the latent space generated by VAME poses another challenge. While it provides a valuable representation of behavior, the interpretability of these representations may be complex. In the future, efforts could involve incorporating explainable artificial intelligence techniques or developing additional visualization tools to facilitate a more intuitive understanding of the learned representations. The applicability of VAME to clinical settings, especially in the context of neurodegenerative diseases needs careful consideration. Validating the framework

with a more extensive range of clinical data and collaborating with domain experts to ensure its reliability and relevance in clinical research is critical. Additionally, VAME primarily focuses on behavior captured through pose estimation, and there is room for improvement in integrating data from other modalities such as neuroimaging or physiological signals. A more comprehensive understanding of the relationship between behavior and underlying neural processes could be achieved through future iterations that explore multimodal approaches (see section 7.4.2). Ensuring the ethological relevance of the behavioral patterns identified by VAME is also a crucial aspect. Collaborations with ethologists and domain experts could validate the identified patterns and relate them to ecologically meaningful behaviors, enhancing the overall reliability and ecological validity of the framework. Furthermore, another consideration is the computational resources required by VAME, which may pose challenges, particularly when dealing with large datasets or real-time applications. Exploring optimization techniques or parallel computing strategies to enhance the efficiency of the framework could make VAME more accessible to researchers with varying computational resources. Finally, by addressing these limitations and exploring avenues for improvement, the VAME framework can evolve into a more robust and versatile tool, expanding its applicability across diverse research contexts and contributing to a deeper understanding of animal behavior.

7.4 Future work

As I conclude the presentation of the current state of research and the findings in this thesis, it is essential to give an outline forward toward the horizon of possibilities for further investigation and development. In this section, I will delve into the realm of future work, outlining potential research directions, that may overcome some of the limitations discussed above, and could extend or even re-imaging the work presented in preceding chapters. The journey of scientific inquiry is characterized by its continuous evolution, with each discovery and insight paving the way for new questions and challenges. In this spirit, I recognize that the work presented here is a stepping stone in the larger landscape of representation learning and behavior embedding. The in this chapter proposed research directions represent a call to action and are not only avenues for expansion but

also a testament to the dynamic nature of science in general. I will explore three possible avenues researcher can take to explore new ways of behavior representation.

7.4.1 Learning from video data

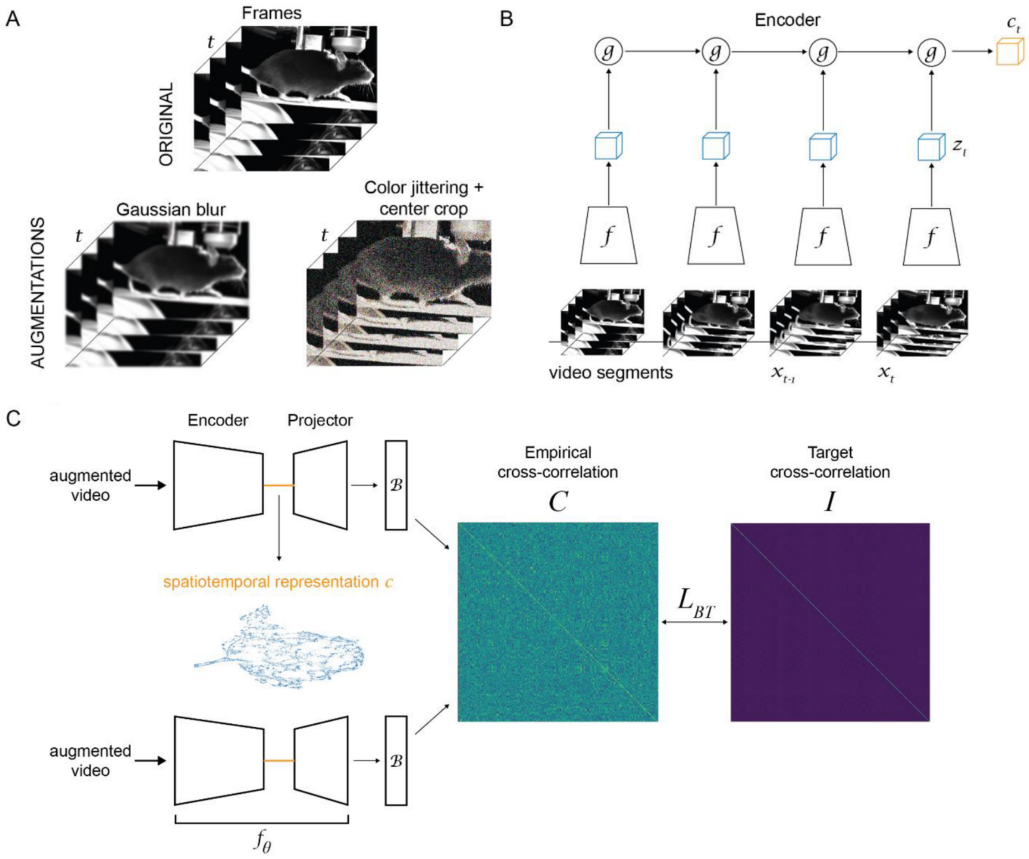

Figure 7.1: **Leveraging Barlow Twins for Unveiling Animal Dynamics from Video Data.** (A) Consecutive occurrences of the initial video frames (top section) juxtaposed with two depictions of applied augmentations (bottom section). (B) Structure incorporating ResNet3D and Convolutional Gated Recurrent Unit, designed for capturing spatiotemporal embeddings. (C) The complete Barlow Twin network. In each iteration, the encoder is presented with two augmented renditions of the original video snippet, with the objective of optimizing the Barlow Twin criterion and acquiring a spatiotemporal embedding (Figure altered from [114]).

To recap, self-supervised learning strives to estimate an approximate distribution $\hat{p}(x)$ that closely matches the original data distribution $p(x)$. Instead of external supervisory signals, it harnesses inherent data structures for guidance. Many of these methods introduce a lower-dimensional latent variable, denoted as z, which encodes the essential aspects of the input signal. This approach is commonly referred to as generative modeling. Once the model has been trained on the data, it gains the capability to generate novel data points from $p(x)$ that resemble the original dataset. Prominent variations of

generative modeling include Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs). As we have seen, VAME utilizes the VAE framework to uncover dynamics and patterns from the behavioral input signal. Nevertheless, VAEs may struggle to consistently distinguish the true signal from its noise component.

7.4.1.1 Utilizing redundancy reduction networks

One promising direction for future research involves utilizing redundancy reduction networks [115]. These networks operate on the fundamental principle of transforming highly repetitive sensory inputs into a factorial code, aligning with the information bottleneck principle [67]. In essence, they consist of two identical networks, each receiving a distinct, distorted version of a signal. Their primary objective is to acquire a representation of the original signal that encapsulates its most crucial information. One notable advantage of this network architecture is its independence from the concept of contrastive learning [116], which necessitates a large number of negative samples and can be computationally intensive. Additionally, in contrast to (variational) autoencoders that attempt to faithfully reproduce every detail of an input signal, often leading to inaccuracies in the presence of diverse lighting conditions or missing data [106], these models are designed to directly reduce such redundancies.

7.4.1.2 Challenges for video data

This also opens the possibility to build a model that operates on the acquired video data rather than user defined pose marker points. The primary approach for studying animal behavior typically involves recording the animal's actions on video. Various factors influence the signal's quality, dependability, and level of noise. Fluctuations in factors like lighting conditions, camera angles/distance, and field-of-view obstructions can introduce disparities in the embedding space. Consequently, there is a demand for models capable of learning to disregard these variations and concentrate on the essential information within the behavioral signal. This makes this kind of model a promising approach since much more of the subtle behavioral changes are embedded from the video.

7.4.1.3 Barlow Twins as instantiation

One instantiation of redundancy networks are Barlow Twins [115]. Barlow Twins have not previously been applied in the context of behavior analysis. For this perspective chapter, I introduce a straightforward implementation of a Barlow Twins model specifically designed for video data obtained from open-field or head-fixed animal recordings (refer to Figure 7.1). To create a behavioral embedding from the input video, I use a temporal duration of 500 ms, corresponding to 30 frames. Each input video undergoes two augmentations, which may involve random cropping, image flipping, Gaussian blurring, or color jittering (see Figure 7.1, A). These augmented videos are then fed into a 3D-ResNet encoder, similar to the one in [117]. I utilize a one-layer Convolutional Gated Recurrent Unit (ConvGRU) with a (1, 1) kernel size as an aggregation function, in line with [116]. The encoder (denoted as f) and aggregation function (referred to as g) share their weights for both distorted video input streams. This design facilitates the transmission of features along the temporal dimension, resulting in a contextual representation c_t (as shown in Figure 7.1, B). The model generates an embedding B through a projector layer, which aims to capture the empirical cross-correlation between the two augmented input videos (as depicted in Figure 7.1, C). I initialize the projector layer in accordance with [115]. The primary objective (Equation 7.1) of this model is to minimize redundancy between the two video inputs, ultimately learning the most probable spatiotemporal embedding.

$$L_{BT} = \Sigma_t (1 - C_{ii})^2 + \lambda \Sigma_i \Sigma_{j \neq i} C_{ij}^2 \tag{7.1}$$

Here, λ represents a positive constant that balances the significance of the first and second loss terms. The matrix C is a cross-correlation matrix computed from the output of two identical networks across the batch dimension. It is a square matrix with values ranging from -1 (indicating perfect anti-correlation) to 1 (indicating perfect correlation). The first term within this framework is referred to as the "invariance term", which aims to equalize the diagonal elements of the cross-correlation matrix to 1. This adjustment promotes embedding invariance with respect to applied distortions. The second term is noted as the "redundancy term", as it seeks to equalize the off-diagonal elements of the cross-correlation matrix to 0, effectively decorrelating the different vector components of the embedding. This process empowers the model to capture non-redundant information

from the video sample. For more comprehensive insights into the operation of the Barlow Twins model, please consult the original paper [115].

7.4.1.4 Summary

I attempted some initial early iterations on such a model which have already yielded promising results, although these results are not presented here and need further exploration and confirmation. However, a significant benefit of this model is its capacity to directly analyze the video signal from a behaving animal without the need for supervision or predefined points of interest. This advantage could also extend to a multi-view system, where different camera angles are presented to the model.

To summarize, I have discussed recent advancements in the realm of self-supervised learning and applied them to the assessment of animal behavior. This perspective serves as an illustration of how the field of (computational) ethology can harness recent machine learning advances to investigate and understand animal behavior.

7.4.2 Multimodal learning of neural-behavior representation

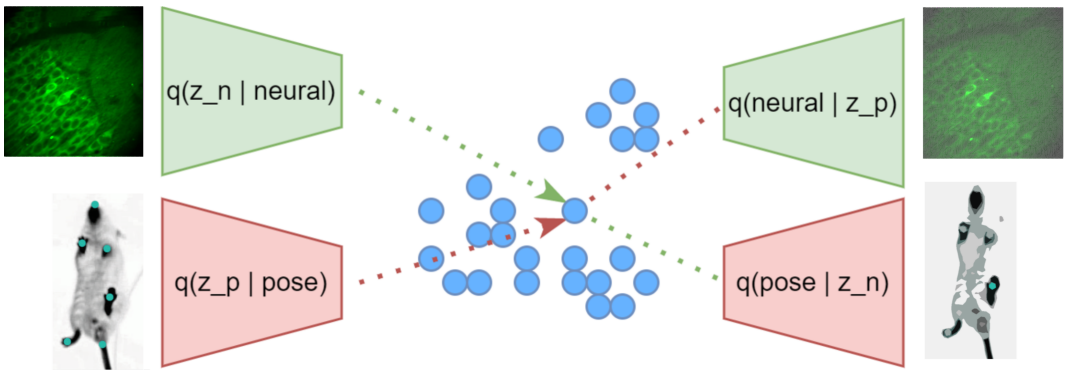

Figure 7.2: **Multimodal embedding architecture of behavior and neural activity.** Left: Neural activity and animal frame. Middle: Joint embedding of both modalities. Right: Reconstructed neural activity and animal frame from shared joint embedding vector.

Another compelling future direction for the field of computational ethology, in my view, will be the implementation of joint representation models, which can encode behavioral signals and neural activity jointly. Such models have a wide range of applications and benefits, including: generating predictions about which neural signals are associated with specific behaviors, determining if observed neural signals are capable of performing a specific task, identifying which behavioral aspects are important and which are trivial,

and categorizing behaviors based on their goals at varying temporal and semantic levels. Such models could be implemented using the variational auto-encoder approach. By adapting joint modality methods from the field of computer vision, behavioral and neural signals could be jointly embedded into the same space [118].

7.4.2.1 Conceptualization of the model architecture

In Figure 7.2, I illustrate the fundamental concept. This model's architecture draws inspiration from [118]. On the left side, we encompass both modalities, namely neural activity and the animal's behavioral output. Specific encoding networks, denoted as $q(z_n|x_{neural})$ and $q(z_p|x_{pose})$, are employed to extract pertinent features from both modalities. These networks are designed to map the resulting lower-dimensional information z_t to the same latent coordinate for corresponding time points t The model subsequently endeavours to reconstruct both modalities from z_t using dedicated decoding networks, $q(x_{neural}|z_n)$ and $q(x_{pose}|z_p)$. By implementing this neural network architecture, researchers can create latent embeddings that combine information from both neural activity and animal behavior simultaneously.

7.4.2.2 Hypothesis for exploration

This framework allows researchers to investigate different hypotheses. The forthcoming hypotheses represent the focus of my future research:

1. Hypothesis 1: By comparing the decoded reconstructions of neural activity and behavior from the shared latent space, researchers can assess the extent to which the two modalities are interrelated and identify potential causal relationships.

2. Hypothesis 2: If we successfully learn a joint embedding space that combines neural activity and animal behavior, it may be possible to accurately reconstruct the behavior of an animal from pure neural input and vice versa. This implies that the shared latent space captures essential information and dependencies between neural activity and behavior, allowing for bidirectional translation between the two modalities. Achieving such bidirectional reconstruction would demonstrate the

model's effectiveness in bridging the gap between neural and behavioral representations, potentially offering a valuable tool for understanding the complex relationship between these two aspects of animal physiology and behavior.

7.4.2.3 Summary

In conclusion, the development of a shared latent space, allowing for bidirectional reconstruction between neural activity and animal behavior, not only promises to elucidate the intricate connections between these two modalities but also offers a valuable tool for researchers to investigate the underlying dynamics of animal physiology and behavior. This innovative approach opens up new horizons in the field of (computational) ethology, providing a practical means to explore the interplay between neural and behavioral data, with broad implications for scientific and practical applications.

7.4.3 Improving the objective of VAME

Finally, I want to explore a direction to improve the model architecture of VAME by extending its objective function. Like with any unsupervised frameworks and cluster assignments, there are imperfections. By accessing the latent space of VAME and encouraging clusterability of the latent vectors, it can be possible to have a better resolution of the underlying animal behavior by enforcing better boundaries. Therefore, we need to design an objective function to encourage similar latent vectors to be assigned to the same cluster label. The idea is to use the mutual information $I(X;Z)$ between the latent vectors within a batch to guide the clustering process. The mutual information is defined as

$$I(Z_u; Z_v) = H(Z_u) - H(Z_u|Z_v), \tag{7.2}$$

where $H(Z_u)$ is the entropy of the the latent vector Z_u and $H(Z_u|Z_v)$ is the conditional entropy between the latent vector Z_u and Z_v. Given a batch of n data points, we can compute the pairwise mutual information $I(Z_u; Z_v)$ for each pair of latent vectors (Z_u, Z_v).

7.4.3.1 Noise contrastive estimation

In practice, maximizing $I(Z_u; Z_v)$ directly can be challenging, so a common approach is to maximize a lower bound on the mutual information. The InfoNCE (Noise Contrastive

Estimation) loss presents a contrastive learning objective that is widely employed in representation learning and self-supervised learning [119] and acts as a lower bound on the mutual information which can be defined as

$$\mathcal{L}_{InfoNCE} = -\log\left(\frac{\exp(sim(z_u, z'_u))}{\sum_{v=1}^{N}\exp(sim(z_u, z'_v))}\right), \tag{7.3}$$

where z_u and z'_u are positive pairs (representations of the same input), z'_v is a negative pair (representation of a different input), and $sim(.,.)$ is a similarity function, such as cosine similarity. At its core, InfoNCE aims to estimate the mutual information between pairs of data points, particularly the relationship between input samples and their corresponding latent representations in a neural network. By formulating a contrastive loss that maximizes the mutual information between positive pairs (similar instances) and minimizes it between negative pairs (dissimilar instances), InfoNCE guides the model to organize data in the latent space, ensuring that similar samples are close together. This approach not only facilitates effective representation learning but also lends itself to tasks like clustering, where the network learns to discriminate between different groups within the data. We can use the computed mutual information values to form an affinity matrix A, where A_{uv} represents the mutual information between Z_u and Z_v. With this affinity matrix, we can now apply a clustering objective like k-Means to assign a cluster label c to each latent vector in the batch.

7.4.3.2 Formulating a mutual information based objective

Lastly, we have to define an objective function that encourages latent vectors with higher mutual information to be assigned to the same cluster. Here, I will provide a simple loss term $\mathcal{L}_{clustering}$ that can be extended to specific needs:

$$\mathcal{L}_{clustering} = \sum_{u=1}^{N}\sum_{v=1}^{N} A_{uv} \times \delta(c_u, c_v) \times d(Z_u, Z_v), \tag{7.4}$$

where N is the batch size, A_{uv} is the mutual information between the latent vectors, c_u and c_v are the assigned cluster labels, $\delta(c_u, c_v)$ is the Kronecker delta function, which is equal to 1 if $c_u = c_v$ (same cluster) and 0 otherwise, and $d(Z_u, Z_v)$ is the distance metric in the latent space (e.g., Euclidean distance) between the latent vectors. This clustering

loss term provides a way to guide the learning of a clustering structure directly within the training of the neural network, without the need for a separate clustering algorithm. This approach is sometimes referred to as "end-to-end clustering" because the clustering structure is learned as part of the neural network training process. Here, the neural network is responsible for both encoding the input data into a latent space and learning a clustering structure within that latent space. Hence, this loss function can be added to the overall loss function of VAME and extends its objective.

7.4.3.3 Summary

Finally, I want to close this discussion with some thoughts about cold starting VAME with such an additional objective function versus pretraining VAME. Pretraining an autoencoder and then fine-tuning it with a clustering objective is a learning form also known as transfer learning that leverages the knowledge gained during the initial autoencoder training. The idea behind this two-step process is that the autoencoder, having learned a good representation during pretraining, provides a more advantageous starting point for the network to learn the clustering structure. However, the effectiveness of this approach depends on the specific characteristics of the data and the nature of the task, so it is recommended to experiment and validate the approach for the specific problem.

7.5 Final thoughts

The field of behavioral measurement is on the verge of transformative advancements in the coming decade, driven by innovative approaches in self-supervised learning. Notably, contrastive learning methods and the integration of transformer architectures, as witnessed in large language models, are poised to redefine the landscape. A particularly promising application involves leveraging the transformer architecture to decipher the "language" of behavior, especially when preceded by a framework such as VAME. In this setting, the role of a framework like VAME is crucial in initially identifying subtle actions, and subsequently, employing a transformer to categorize and understand the broader behavioral context.

The presented work transcends the confines of conventional behavioral analysis by not only identifying behavioral patterns but also elucidating the intricate interplay of

subtle behavioral actions across different hierarchical levels and using the information stored in its dynamical embedding space. The methodology represents a pragmatic integration of state-of-the-art machine learning techniques with the rich dynamics found in behavioral pose data, establishing a new standard for computational ethology. As the field propels forward, the groundwork laid by this research is strategically positioned to steer future investigations toward a more nuanced comprehension of the dynamic facets inherent in animal behavior. The synergy between advanced machine learning methodologies and the intricacies of behavioral data combined with the vast recordings of neural activity is assured to shape the next wave of breakthroughs in behavioral measurements.

Bibliography

[1] E. R. Kandel, J. H. Schwartz, and T. M. Jessell. *Principles of Neural Science*. Elsevier, 1991.

[2] N. Tinbergen. "On aims and methods of Ethology". In: *Zeitschrift für Tierpsychologie* 20.4 (1963), pp. 410–433. DOI: https://doi.org/10.1111/j.1439-0310.1963.tb01161.x.

[3] F. Fuhrmann, D. Justus, L. Sosulina, H. Kaneko, T. Beutel, D. Friedrichs, S. Schoch, M. K. Schwarz, M. Fuhrmann, and S. Remy. "Theta Oscillations, and the Speed-Correlated Firing of Hippocampal Neurons Are Controlled by a Medial Septal Glutamatergic Circuit". In: *Neuron* 86.5 (2015), pp. 1253–64. DOI: 10.1016/j.neuron.2015.05.001.

[4] E. A. Giovannetti, S. Poll, D. Justus, H. Kaneko, F. Fuhrmann, J. Steffen, S. Remy, and M. Fuhrmann. "Restoring memory by optogenetic synchronization of hippocampal oscillations in an Alzheimer's disease mouse model". In: *bioRxiv* (2018). DOI: 10.1101/363820.

[5] N. A. Steinmetz, C. Aydin, A. Lebedeva, M. Okun, M. Pachitariu, M. Bauza, M. Beau, J. Bhagat, C. Böhm, M. Broux, S. Chen, J. Colonell, R. J. Gardner, B. Karsh, F. Kloosterman, D. Kostadinov, C. Mora-Lopez, J. O'Callaghan, J. Park, J. Putzeys, B. Sauerbrei, R. J. J. van Daal, A. Z. Vollan, S. Wang, M. Welkenhuysen, Z. Ye, J. T. Dudman, B. Dutta, A. W. Hantman, K. D. Harris, A. K. Lee, E. I. Moser, J. O'Keefe, A. Renart, K. Svoboda, M. Häusser, S. Haesler, M. Carandini, and T. D. Harris. "Neuropixels 2.0: A miniaturized high-density probe for stable, long-term brain recordings". In: *Science* 372.6539 (2021), eabf4588. DOI: 10.1126/science.abf4588.

[6] W. Zong, H. A. Obenhaus, E. R. Skytøen, H. Eneqvist, N. L. de Jong, R. Vale, M. R. Jorge, M. Moser, and E. I. Moser. "Large-scale two-photon calcium imaging in

freely moving mice". In: *Cell* 185.7 (2022), 1240–1256.e30. DOI: 10.1016/j.cell.2022.02.017.

[7] N. Tinbergen. "The Study of Instinct". In: *Clarendon Press* (1955).

[8] D. A. Levitis, W. Z. Lidicker, and G. Freund. "Behavioural biologists don't agree on what constitutes behaviour." In: *Animal behaviour* 78.1 (2009), pp. 103 –110.

[9] D. J. Anderson and P. Perona. "Toward a Science of Computational Ethology". In: *Neuron* 84.1 (2014), pp. 18–31. ISSN: 0896-6273. DOI: https://doi.org/10.1016/j.neuron.2014.09.005.

[10] S. R. Datta, D. J. Anderson, K. Branson, P. Perona, and A. Leifer. "Computational Neuroethology: A Call to Action". In: *Neuron* 104.1 (2019), pp. 11–24. ISSN: 0896-6273. DOI: https://doi.org/10.1016/j.neuron.2019.09.038.

[11] J. W. Krakauer, A. A. Ghazanfar, A. Gomez-Marin, M. A. MacIver, and D. Poeppel. "Neuroscience Needs Behavior: Correcting a Reductionist Bias". In: *Neuron* 93.3 (2017), pp. 480–490. ISSN: 0896-6273. DOI: 10.1016/j.neuron.2016.12.041.

[12] T. D. Pereira, Joshua W. Shaevitz, and Mala Murthy. "Quantifying behavior to understand the brain". In: *Nature Neuroscience* 23.12 (2020), pp. 1537–1549. ISSN: 1546-1726. DOI: 10.1038/s41593-020-00734-z.

[13] K. Luxem, P. Mocellin, F. Fuhrmann, J. Kürsch, S. R. Miller, J. J. Palop, S. Remy, and P. Bauer. "Identifying behavioral structure from deep variational embeddings of animal motion". In: *Communications Biology* 5 (1 2022), p. 1267. ISSN: 2399-3642. DOI: 10.1038/s42003-022-04080-7.

[14] C. Segalin, J. Williams, T. Karigo, M. Hui, M. Zelikowsky, J. J. Sun, P. Perona, D. J. Anderson, and A. Kennedy. "The Mouse Action Recognition System (MARS) software pipeline for automated analysis of social behaviors in mice". In: *eLife* 10 (2021), e63720. DOI: 10.7554/eLife.63720.

[15] F. Helmchen and W. Denk. "Deep tissue two-photon microscopy". In: *Nature Methods* 2 (2005), pp. 932–940.

[16] E. S. Boyden, F. Zhang, E. Bamberg, G. Nagel, and K. Deisseroth. "Millisecond-timescale, genetically targeted optical control of neural activity". In: *Nature Neuroscience* 8 (2005), pp. 1263 –1268.

[17] D. A. Dombeck, C. D. Harvey, L. Tian, L. L. Looger, and D. W. Tank. "Functional imaging of hippocampal place cells at cellular resolution during virtual navigation". In: *Nature Neuroscience* 13 (2010), pp. 1433 –1440.

[18] A. Mathis, P. Mamidanna, K. M. Cury, T. Abe, V. N. Murthy, M. W. Mathis, and M. Bethge. "DeepLabCut: markerless pose estimation of user-defined body parts with deep learning". In: *Nature Neuroscience* 21.9 (2018), pp. 1281–1289. ISSN: 1546-1726. DOI: 10.1038/s41593-018-0209-y.

[19] T. D. Pereira, N. Tabris, A. Matsliah, D. M. Turner, J. Li, S. Ravindranath, E. S. Papadoyannis, E. Normand, D. S. Deutsch, Z. Y. Wang, G. C. McKenzie-Smith, C. C. Mitelut, M. D. Castro, J. D'Uva, M. Kislin, D. H. Sanes, S. D. Kocher, S. S.-H. Wang, A. L. Falkner, J. W. Shaevitz, and M. Murthy. "SLEAP: A deep learning system for multi-animal pose tracking". In: *Nature Methods* 19.4 (2022), pp. 486–495. ISSN: 1548-7105. DOI: 10.1038/s41592-022-01426-1.

[20] J. M. Graving, D. Chae, H. Naik, L. Li, B. Koger, B. R. Costelloe, and I. D. Couzin. "DeepPoseKit, a software toolkit for fast and robust animal pose estimation using deep learning". In: *eLife* 8 (2019), e47994. DOI: https://doi.org/10.7554/eLife.47994.

[21] G. J. Berman, D. M. Choi, W. Bialek, and J. W. Shaevitz. "Mapping the stereotyped behaviour of freely moving fruit flies". In: *Journal of the Royal Society Interface* 11 (2014), p. 20140672. DOI: 10.1098/rsif.2014.0672.

[22] A. B. Wiltschko, M. J. Johnson, G. Iurilli, R. E. Peterson, J. M. Katon, S. L. Pashkovski, V. E. Abraira, R. P. Adams, and S. R. Datta. "Mapping Sub-Second Structure in Mouse Behavior". In: *Neuron* 88.6 (2015), pp. 1121–1135. DOI: 10.1016/j.neuron.2015.11.031.

[23] R. Dawkins. "Hierarchical organisation: A candidate principle for ethology". In: Oxford, England: Cambridge University Press, 1976.

[24] K. Luxem, J. J. Sun, S. P. Bradley, K. Krishnan, E. Yttri, J. Zimmermann, T. D. Pereira, and M. Laubach. "Open-source tools for behavioral video analysis: Setup, methods, and best practices". In: *eLife* 12 (2023), e79305. DOI: https://doi.org/.

[25] A. E. X. Brown and B. de Bivort. "Ethology as a physical science". In: *Nature Physics* 14.7 (2018), pp. 653–657. ISSN: 1745-2481. DOI: 10.1038/s41567-018-0093-0.

[26] G. J. Berman. "Measuring behavior across scales". In: *BMC Biology* 16.1 (2018), p. 23. ISSN: 1741-7007. DOI: 10.1186/s12915-018-0494-7.

[27] J. A. Bender, E. M. Simpson, and R. E. Ritzmann. "Computer-assisted 3D kinematic analysis of all leg joints in walking insects". In: *PLoS One* 5.10 (2010), e13617. DOI: 10.1371/journal.pone.0013617.

[28] C. Schütz and V. Dürr. "Active tactile exploration for adaptive locomotion in the stick insect". In: *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences* 366.1581 (2011), 2996–3005. DOI: 10.1098/rstb.2011.0126.

[29] J. Kain, C. Stokes, Q. Gaudry, X. Song, J. Foley, R. Wilson, and B. de Bivort. "Leg-tracking and automated behavioural classification in Drosophila". In: *Nature Communications* 4.1 (2013), p. 1910. ISSN: 2041-1723. DOI: 10.1038/ncomms2908.

[30] S. B. Hausmann, A. M. Vargas, A. Mathis, and M. W. Mathis. "Measuring and modeling the motor system with machine learning". In: *Current Opinion in Neurobiology* 70 (2021). Computational Neuroscience, pp. 11–23. ISSN: 0959-4388. DOI: https://doi.org/10.1016/j.conb.2021.04.004.

[31] S. R. O. Nilsson, N. L. Goodwin, J. J. Choong, S. Hwang, H. R. Wright, Z. C. Norville, X. Tong, D. Lin, B. S. Bentzley, N. Eshel, R. J. McLaughlin, and S. A. Golden. "Simple Behavioral Analysis (SimBA) – an open source toolkit for computer classification of complex social behaviors in experimental animals". In: *bioRxiv* (2020). DOI: 10.1101/2020.04.19.049452.

[32] J. P. Bohnslav, N. K. Wimalasena, K. J. Clausing, Y. Y. Dai, D. A. Yarmolinsky, T. Cruz, A. D. Kashlan, M. E. Chiappe, L. L. Orefice, C. J. Woolf, and C. D. Harvey. "DeepEthogram, a machine learning pipeline for supervised behavior classification from raw pixels". In: *eLife* 10 (2021), e63377. DOI: 10.7554/eLife.63377.

[33] A. Gomez-Marin, J. J. Paton, A. R. Kampff, R. M. Costa, and Z. F. Mainen. "Big behavioral data: psychology, ethology and the foundations of neuroscience". In: *Nature Neuroscience* 17.11 (2014), pp. 1455–1462. DOI: 10.1038/nn.3812.

[34] A. I. Hsu and E. A. Yttri. "B-SOiD, an open-source unsupervised algorithm for identification and fast prediction of behaviors". In: *Nature Communications* 12.1 (2021), p. 5188. ISSN: 2041-1723. DOI: 10.1038/s41467-021-25420-x.

[35] J. Altmann. "Observational study of behavior: sampling methods". In: *Behaviour* 49.3 (1974), pp. 227–267. DOI: 10.1163/156853974X00534.

[36] P. Martin and P. Bateson. *Measuring behaviour: an introductory guide*. 2021.

[37] S. Musall, M. T. Kaufman, A. L. Juavinett, S. Gluf, and A. K. Churchland. "Single-trial neural dynamics are dominated by richly varied movements". In: *Nature Neuroscience* 22.10 (2019), pp. 1677–1686.

[38] K. Luxem, F. Fuhrmann, S. Remy, and P. Bauer. "Hierarchical network analysis of behavior and neuronal population activity". In: *Conference of Cognitive Computational Neuroscience*. 2019. DOI: 10.32470/CCN.2019.1261-0.

[39] L. van der Maaten and G. E. Hinton. "Visualizing Data using t-SNE". In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605.

[40] L. McInnes, J. Healy, and J. Melville. "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction". In: *arXiv* (2018).

[41] K. Branson, A. A. Robie, J. Bender, P. Perona, and M. H. Dickinson. "High-throughput ethomics in large groups of *Drosophila*". In: *Nature Methods* 6.6 (2009), pp. 451–457. DOI: 10.1038/nmeth.1328.

[42] B. Q. Geuther, S. P. Deats, K. J. Fox, S. A. Murray, R. E. Braun, J. K. White, E. J. Chesler, C. M. Lutz, and V. Kumar. "Robust mouse tracking in complex environments using neural networks". In: *Communications Biology* 2 (2019), p. 124. DOI: 10.1038/s42003-019-0362-1.

[43] Y. Lecun and Y. Bengio. "The Handbook of Brain Theory and Neural Networks". In: (1995), pp. 255–258.

[44] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik. "Human Pose Estimation with Iterative Error Feedback". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 4733–4742.

[45] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. "Convolutional Pose Machines". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 4724–4732. DOI: 10.1109/CVPR.2016.511.

[46] A. Newell, K. Yang, and J. Deng. "Stacked Hourglass Networks for Human Pose Estimation". In: *Computer Vision – ECCV 2016*. Vol. 9912. Lecture Notes in Computer Science. 2016. DOI: 10.1007/978-3-319-46484-8_29.

[47] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. "Microsoft COCO: Common Objects in Context". In: *Computer Vision – ECCV 2014*. Springer International Publishing, 2014, pp. 740–755. ISBN: 978-3-319-10602-1.

[48] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. "2D Human Pose Estimation: New Benchmark and State of the Art Analysis". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 3686–3693. DOI: 10.1109/CVPR.2014.471.

[49] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. "Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.7 (2014), pp. 1325–1339. DOI: 10.1109/TPAMI.2013.248.

[50] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. "ImageNet Large Scale Visual Recognition Challenge". In: *International Journal of Computer Vision (IJCV)* 115.3 (2015), pp. 211–252. DOI: 10.1007/s11263-015-0816-y.

[51] M. W. Mathis and A. Mathis. "Deep learning tools for the measurement of animal behavior in neuroscience". In: *Current Opinion in Neurobiology* 60 (2020), pp. 1–11. ISSN: 0959-4388. DOI: 10.1016/j.conb.2019.10.008.

[52] S. Christin, É. Hervet, and N. Lecomte. "Applications for deep learning in ecology". In: *Methods in Ecology and Evolution* 10 (2019), pp. 1632–1644. DOI: 10.1111/2041-210X.13256.

[53] J. J. Sun, S. Ryou, R. H. Goldshmid, B. Weissbourd, J. O. Dabiri, D. J. Anderson, A. Kennedy, Y. Yue, and P. Perona. "Self-Supervised Keypoint Discovery in Behavioral Videos". In: *Proceedings. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2022, 2161–2170. DOI: 10.1109/cvpr52688.2022.00221.

[54] J. E. Markowitz, W. Gillis, C. C. Beron, S. Q. Neufeld, K. Robertson, N. D. Bhagat, R. E. Peterson, E. Peterson, M. Hyun, S. W. Linderman, i B. L. Sabatin, and S. R. Datta. "The Striatum Organizes 3D Behavior via Moment-to-Moment Action Selection". In: *Cell* 174.1 (2018), pp. 44 –58.

[55] J. Cande, S. Namiki, J. Qiu, W. Korff, G. M. Card, J. W. Shaevitz, D. L. Stern, and G. J. Berman. "Optogenetic dissection of descending behavioral control in *Drosophila*". In: *eLife* 7 (2018), e34275.

[56] A. J. Calhoun, J. W. Pillow, and M. Murthy. "Unsupervised identification of the internal states that shape natural behavior". In: *Nature Neuroscience* 22 (2019), pp. 1546–1726.

[57] D. A. Levitis, W. Z. Lidicker, and G. Freund. "Behavioural biologists don't agree on what constitutes behaviour". In: *Animal behaviour* 78.1 (2009), pp. 103–110.

[58] B. Szigeti, T. Stone, and Webb B. "Inconsistencies in C. elegans behavioural annotation". In: *bioRxiv* (2016).

[59] X. Leng, M. Wohl, K. Ishii, P. Nayak, and K Asahina. "Quantitative comparison of Drosophila behavior annotations by human observers and a machine learning algorithm". In: *bioRxiv* (2020).

[60] J. B. Tenenbaum, V. de Silva, and J. C. Langford. "A Global Geometric Framework for Nonlinear Dimensionality Reduction". In: *Science* 290.5500 (2000), pp. 2319–2323.

[61] P. L. Goupillaud, A. Grossmann, and J. Morlet. "Cycle-octave and related transforms in seismic signal analysis". In: *Geoexploration* 23 (1984), pp. 85–102.

[62] M. H. McCullough and G. J. Goodhill. "Unsupervised quantification of naturalistic animal behaviors for gaining insight into the brain". In: *Current Opinion in Neurobiology* 70 (2021). Computational Neuroscience, pp. 89–100. ISSN: 0959-4388. DOI: https://doi.org/10.1016/j.conb.2021.07.014.

[63] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems* 27 (2014).

[64] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: *3rd International Conference on Learning Representations, ICLR*. Vol. 3. 2015.

[65] J. Donahue, P. Krähenbühl, and T. Darrell. "Adversarial Feature Learning". In: *arXiv* (2016).

[66] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville. "Adversarially Learned Inference". In: *arXiv* (2017).

[67] N. Tishby and N. Zaslavsky. "Deep learning and the information bottleneck principle". In: Jerusalem, Israel, 2015, pp. 1–5. DOI: 10.1109/ITW.2015.7133169.

[68] A. Ivanov, G. V. Nosovskiy, A. Y. Chekunov, D. A. Fedoseev, V. A. Kibkalo, M. Nikulin, F. Popelenskiy, S. A. Komkov, I. L. Mazurenko, and A. Petiushko. "Manifold Hypothesis in Data Analysis: Double Geometrically-Probabilistic Approach to Manifold Dimension Estimation". In: *ArXiv* (2021).

[69] D. P. Kingma and M. Welling. "An Introduction to Variational Autoencoders". In: *arxiv* (2019).

[70] C. M. Bishop. *Pattern Recognition and Machine Learning*. 1st ed. Information Science and Statistics. Published: 17 August 2006. Springer New York, NY, 2006, pp. XX, 778. ISBN: 978-0-387-31073-2.

[71] A. Kong, P. McCullagh, X. L. Meng, D. Nicolae, and Z. Tan. "A Theory of Statistical Models for Monte Carlo Integration". In: *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 65.3 (2003), pp. 585–618.

[72] D. P. Kingma and M. Welling. "Auto-Encoding Variational Bayes". In: *2nd International Conference on Learning Representations, ICLR* (2014).

[73] S. Ruder. "An overview of gradient descent optimization algorithms". In: *arxiv* (2017).

[74] X. P. Burgos-Artizzu, P. Dollár, D. Lin, D. J. Anderson, and P. Perona. "Social behavior recognition in continuous video". In: *IEEE Conference on Computer Vision and Pattern Recognition* (2012), pp. 1322–1329.

[75] M. Yokoyama, H. Kobayashi, L. Tatsumi, and T. Tomita. "Mouse Models of Alzheimer's Disease". In: *Frontiers in Molecular Neuroscience* 15 (2022), p. 912995. DOI: 10.3389/fnmol.2022.912995.

[76] J. L. Jankowsky, H. H. Slunt, T. Ratovitski, N. A. Jenkins, N. G. Copeland, and D. R. Borchelt. "Co-expression of multiple transgenes in mouse CNS: A comparison of strategies". In: *Biomolecular Engineering* 17.6 (2001), pp. 157–165. DOI: 10.1016/s1389-0344(01)00067-3.

[77] S. Hochreiter and J. Schmidhuber. "Long Short-Term Memory". In: *Neural Comput.* 9.8 (1997), 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735.

[78] K. Cho, C. van Merriënboer B.and Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation". In: 2014, pp. 1724–1734. DOI: 10.3115/v1/D14-1179.

[79] N. Srivastava, E. Mansimov, and R. Salakhutdinov. "Unsupervised Learning of Video Representations Using LSTMs". In: 2015, 843–852. DOI: 10.5555/3045118.3045209.

[80] H. Kuehne, A. Richard, and J. Gall. "A Hybrid RNN-HMM Approach for Weakly Supervised Temporal Action Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42.4 (2020), pp. 765–779. DOI: 10.1109/TPAMI.2018.2884469.

[81] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio. "A Recurrent Latent Variable Model for Sequential Data". In: *Advances in Neural Information Processing Systems*. Vol. 28. 2015.

[82] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. "InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems*. Vol. 29. 2016.

[83] I. Higgins, L. Matthey, A. Pal, C.P. Burgess, X. Glorot, M.M. Botvinick, S. Mohamed, and A. Lerchner. "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework". In: *International Conference on Learning Representations*. 2016.

[84] Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou. "Variational Deep Embedding: An Unsupervised and Generative Approach to Clustering". In: *International Joint Conference on Artificial Intelligence*. 2016.

[85] J. Pereira and M. Silveira. "Learning Representations from Healthcare Time Series Data for Unsupervised Anomaly Detection". In: *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*. 2019, pp. 1–7. DOI: 10.1109/BIGCOMP.2019.8679157.

[86] Q. Ma, J. Zheng, S. Li, and G. W. Cottrell. "Learning Representations for Time Series Clustering". In: *Advances in Neural Information Processing Systems*. Vol. 32. 2019.

[87] D. P. Kingma and M. Welling. "Auto-Encoding Variational Bayes". In: *2nd International Conference on Learning Representations, ICLR* (2014).

[88] L. Rabiner and B. Juang. "An Introduction to Hidden Markov Models". In: *IEEE ASSP Magazine* 3.1 (1986), pp. 4–16. DOI: 10.1109/MASSP.1986.1165342.

[89] J. L. Jankowsky, D. J. Fadale, J. Anderson, G. M. Xu, V. Gonzales, N. A. Jenkins, N. G. Copeland, M. K. Lee, L. H. Younkin, S. L. Wagner, S. G. Younkin, and D. R. Borchelt. "Mutant presenilins specifically elevate the levels of the 42 residue beta-amyloid peptide in vivo: evidence for augmentation of a 42-specific gamma secretase". In: *Human Molecular Genetics* 13.2 (2004), pp. 159–170. DOI: 10.1093/hmg/ddh019.

[90] H. Huang, S. Nie, M. Cao, C. Marshall, J. Gao, N. Xiao, G. Hu, and M. Xiao. "Characterization of AD-like phenotype in aged APPSwe/PS1dE9 mice". In: *Age (Dordrecht, Netherlands)* 38.4 (2016), pp. 303–322. DOI: 10.1007/s11357-016-9929-7.

[91] K. D. Onos, A. Uyar, K. J. Keezer, H. M. Jackson, C. Preuss, C. J. Acklin, R. O'Rourke, R. Buchanan, T. L. Cossette, S. J. Sukoff Rizzo, I. Soto, G. W. Carter, and G. R. Howell. "Enhancing face validity of mouse models of Alzheimer's disease with natural

genetic variation". In: *PLoS Genetics* 15.5 (2019), e1008155. DOI: 10.1371/journal.pgen.1008155.

[92] R. Lalonde, H. D. Kim, and K. Fukuchi. "Exploratory activity, anxiety, and motor coordination in bigenic *APPswe + PS1/DeltaE9* mice". In: *Neuroscience Letters* 369.2 (2004), pp. 156–161. DOI: 10.1016/j.neulet.2004.07.069.

[93] C. Janus, A. Y. Flores, G. Xu, and D. R. Borchelt. "Behavioral abnormalities in APPSwe/PS1dE9 mouse model of AD-like pathology: comparative analysis across multiple behavioral domains". In: *Neurobiology of Aging* 36.9 (2015), pp. 2519–2532. DOI: 10.1016/j.neurobiolaging.2015.05.010.

[94] S. J. Webster, A. D. Bachstetter, and L. J. Van Eldik. "Comprehensive behavioral characterization of an APP/PS-1 double knock-in mouse model of Alzheimer's disease". In: *Alzheimer's research and therapy* 5.3 (2013), p. 28. DOI: 10.1186/alzrt182.

[95] B. T. Biallosterski, J. Prickaerts, M. S. Rahnama'i, S. de Wachter, G. A. van Koeveringe, and C. Meriaux. "Changes in voiding behavior in a mouse model of Alzheimer's disease". In: *Frontiers in aging neuroscience* 7 (2015), p. 160. DOI: 10.3389/fnagi.2015.00160.

[96] R. Chaudhuri, B. Gerçek, B. Pandey, A. Peyrache, and I. Fiete. "The intrinsic attractor manifold and population dynamics of a canonical cognitive circuit across waking and sleep". In: *Nature Neuroscience* 22.9 (2019), pp. 1512–1520. DOI: 10.1038/s41593-019-0460-x.

[97] A. Rubin, L. Sheintuch, N. Brande-Eilat, O. Pinchasof, Y. Rechavi, N. Geva, and Y. Ziv. "Revealing neural correlates of behavior without behavioral measurements". In: *Nature Communications* 10.1 (2019), p. 4745. DOI: 10.1038/s41467-019-12724-2.

[98] P. Ghosh, S. M .M. Sajjadi, A. Vergari, M. Black, and B. Scholkopf. "From Variational to Deterministic Autoencoders". In: *International Conference on Learning Representations*. 2020.

[99] M. Jazayeri and A. Afraz. "Navigating the Neural Space in Search of the Neural Code". In: *Neuron* 93 (2017), pp. 1003–1014.

[100] S. Y. Chung and L. F. Abbott. "Neural population geometry: An approach for understanding biological and artificial neural networks". In: *Current Opinion in Neurobiology* 70 (2021), pp. 137–144.

[101] A. Kohn, R. Coen-Cagli, I. Kanitscheider, and A. Pouget. "Correlations and Neuronal Population Information". In: *Annual Review of Neuroscience* 39.1 (2016), pp. 237–256. DOI: 10.1146/annurev-neuro-070815-013851.

[102] E. A. Pnevmatikakis, D. Soudry, Y. Gao, T. A. Machado, J. Merel, D. Pfau, T. Reardon, Y. Mu, C. Lacefield, W. Yang, M. Ahrens, R. Bruno, T. M. Jessell, D. S. Peterka, R. Yuste, and L. Paninski. "Simultaneous Denoising, Deconvolution, and Demixing of Calcium Imaging Data". In: *Neuron* 89.2 (2016), 285–299.

[103] I. S. Dhillon. "Co-Clustering Documents and Words Using Bipartite Spectral Graph Partitioning". In: 2001, 269–274. DOI: 10.1145/502512.502550.

[104] M. Newman. *Networks: An Introduction*. 2010.

[105] S. Xu, H. Yang, V. Menon, A. L. Lemire, L. Wang, F. E. Henry, S. C. Turaga, and S. M. Sternson. "Behavioral state coding by molecularly defined paraventricular hypothalamic cell type ensembles". In: *Science* 370.6514 (2020), eabb2494. DOI: 10.1126/science.abb2494.

[106] C. Shi, S. Schwartz, S. Levy, S. Achvat, M. Abboud, A. Ghanayim, J. Schiller, and G. Mishne. "Learning Disentangled Behavior Embeddings". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 22562–22573.

[107] B. D. DeAngelis, J. A. Zavatone-Veth, and D. A. Clark. "The manifold structure of limb coordination in walking *Drosophila*". In: *eLife* 8 (2019), e46409. DOI: 10.7554/eLife.46409.

[108] E. Batty, M. Whiteway, S. Saxena, D. Biderman, T. Abe, S. Musall, W. Gillis, J. Markowitz, A. Churchland, J. P. Cunningham, S. R. Datta, S. Linderman, and L. Paninski. "BehaveNet: Nonlinear embedding and Bayesian neural decoding of behavioral videos". In: vol. 32. 2019.

[109] S. Günel, H. Rhodin, D. Morales, J. Campagnolo, P. Ramdya, and P. Fua. "DeepFly3D, a deep learning-based approach for 3D limb and appendage tracking in tethered, adult *Drosophila*". In: *eLife* 8 (2019), e48571. DOI: 10.7554/eLife.48571.

[110] I. Sarkar, I. Maji, C. Omprakash, S. Stober, S. Mikulovic, and P. Bauer. "Evaluation of deep lift pose models for 3D rodent pose estimation based on geometrically triangulated data". In: *CVPR 2021 Workshop CV4animals*. 2021.

[111] T. W. Dunn, J. D. Marshall, K. S. Severson, D. E. Aldarondo, D. G. C. Hildebrand, S. N. Chettih, W. L. Wang, A. J. Gellis, D. E. Carlson, D. Aronov, W. A. Freiwald, F. Wang, and B. P. Ölveczky. "Geometric deep learning enables 3D kinematic profiling across species and environments". In: *Nature Methods* 18.5 (2021), pp. 564–573. DOI: 10.1038/s41592-021-01106-6.

[112] B. Paaßen. "Revisiting the tree edit distance and its backtracing: A tutorial". In: *arxiv* (2022).

[113] P. N. Lawlor, M. G. Perich, L. E. Miller, and K. P. Kording. "Linear-Nonlinear-Time-Warp-Poisson models of neural activity". In: *bioRxiv* (2018).

[114] K. Luxem and P. Mocellin. "Self-supervised learning as a gateway to reveal underlying dynamics in animal behavior". In: *12th International Conference on Methods and Techniques in Behavioral Research*. Vol. 2. 2022, pp. 167–169.

[115] J. Zbontar, L. Jing, I. Misra, and LeCunn Y. "Barlow Twins: Self-Supervised Learning via Redundancy Reduction". In: *Proceedings of the 38 th International Conference on Machine Learning 2021*. 2021.

[116] T. Han, W. Xie, and A. Zisserman. "Video Representation Learning by Dense Predictive Coding". In: *arxiv*. 2019.

[117] K. Hara, H. Kataoka, and Y. Satoh. "Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?" In: *CVPR*. 2018.

[118] A. Spurr, J. Song, S. Park, and O. Hilliges. "Cross-modal Deep Variational Hand Pose Estimation". In: *CVPR*. Salt Lake City, USA, 2018.

[119] A. van den Oord, Y. Li, and O. Vinyals. "Representation Learning with Contrastive Predictive Coding". In: *arxiv*. 2019.

[120] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems* 25 (2012).

[121] F. Rosenblatt. "The perceptron: A probabilistic model for information storage and organization in the brain". In: *Psychological Review* 65.6 (1958), pp. 386–408.

[122] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola. "Dive into Deep Learning". In: *arXiv* (2021).

[123] M. Schuster and K. K. Paliwal. "Bidirectional recurrent neural networks". In: *IEEE Transactions on Signal Processing* 45.11 (1997), pp. 2673–2681. DOI: 10.1109/78.650093.

Appendix A

Appendix

A.1 Deep learning

The field of artificial intelligence has seen significant growth since the widespread availability of graphical processing units (GPUs). In 2012, an artificial neural network (ANN) called AlexNet competed in the ImageNet Large Scale Visual Recognition Challenge, where it achieved a top-5[1] error of 15.3%, more than 10.8% better than previous models [120]. The success of AlexNet was not just due to the depth of the model but also the utilization of the computational power of GPUs, which made the computationally expensive process of training and modeling ANNs feasible. The field of deep learning, which involves the use of multiple layers of artificial neurons interconnected to approximate a desired output function, has seen massive progress since then. These layers are organized in a hierarchical structure, with the input layer receiving the raw input data or features of the data, and the output layer providing the desired output. The intermediate layers, called hidden layers, are used to extract features from the input data. The number of layers and the number of neurons in each layer are called the architecture of the network. Artificial neurons are modeled after biological neurons and are used to simulate the computation of the human brain. The input and output of each neuron are related by a set of weights and a bias. These parameters are learned during the training process of the ANN. The process of training an ANN is done by adjusting the weights and biases to minimize the error between the predicted output and the desired output. The most

[1]The top-5 error is a method of benchmarking a machine learning model in the *ImageNet Large Scale Visual Recognition Challenge*. Is the target label in one of the top five predictions of the model, its considered to be correct.

popular method for training ANNs is called backpropagation, which is an iterative algorithm used to adjust the weights and biases of the network. The algorithm works by propagating the error from the output layer to the input layer through the hidden layers, calculating the gradient of the error with respect to the weights and biases, and adjusting them in the opposite direction of the gradient. The process is repeated multiple times until the error is minimized. In this section, I will provide an overview of the concepts of artificial neurons, layers, and the backpropagation method used in deep learning. However, it is important to note that there are many variations and advancements that have been made in the field of deep learning and this overview is specific to provide only the necessary information for understanding the developed VAME model.

A.1.1 General principles

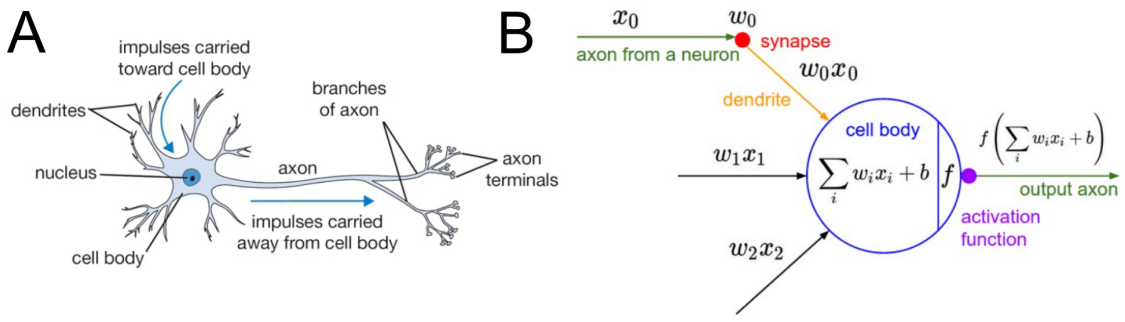

Figure A.1: **Simple sketch of a biological neuron and the mathematical model of an artificial neuron.** (A) Representations of a biological neuron. (B) Illustrations of an artificial neuron, serving as a simplified mathematical counterpart to a biological neuron (modified from *https://cs231n.github.io/neural-networks-1/*).

As this work is rooted in both computer science and neuroscience, I start by describing the foundation of artificial neurons, which is the classical perceptron model. The perceptron model was first proposed by Frank Rosenblatt in 1958 [121], with the goal of creating a machine that could mimic the function of a biological neuron. In a simplistic view, a basic neuron is made of a cell body, synapses, dendrites and axons (Figure A.1, A). Axons eventually branch out and connect to other neurons via synapses and dendrites. In the view of a computational model, the signal, here x_0, travels along the axon and interacts with the dendrites of other neurons based on their synaptic strength w_0. Hence, it performs a multiplicative computation $x_0 w_0$. In the perceptron model, the inputs are multiplied by a corresponding weight and the products are summed up, a bias term b

is then added to the sum, and the result is passed through an activation function. The activation function is a mathematical function that determines the output of the neuron based on the input, bias and weights. This results into the following simple equation:

$$\sigma(\sum_i w_i x_i + b). \tag{A.1}$$

Here, σ represents the activation function. A common choice is to use a *sigmoid function* σ. This function takes in real-valued inputs and clamps it between a range of $0 - 1$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{A.2}$$

The perceptron model was originally developed to solve simple linear classification problems, meaning that it could only classify data into two classes. However, with the development of multi-layer perceptrons, which consist of multiple layers of perceptrons, it became possible to solve more complex non-linear problems.

A crucial idea in artificial neurons is that the synaptic weights w are learnable parameter. These parameter control the strength of influence. Like in a biological neuron, this can lead to excitatory (positive weight) and inhibitory (negative weight) connections. From the dendrite, the signal is carried to the cell body, where it gets summed with signal parts coming from other dendrites. If a threshold value is crossed, the neuron depolarizes, which means it *fires*, sending a signal along its axon. The computational model of an artificial neuron assumes that the precise timing of this firing does not matter but only the frequency of the firing communicates information. The *firing rate* of the artificial neuron is modeled with the activation function σ. It represents the frequency of the spikes along the axon. In summary, an artificial neuron performs a dot product of the inputs and weights, adds a bias and applies an non-linearity function (Figure A.1, B).

A.1.2 Neural network organisation

Artificial neurons in multilayer perceptrons (MLP) are organized in multiple layers to approximate complex functions. These layers, also known as hidden layers, allow for a greater depth in the model, enabling it to handle a more general class of functions. MLPs are the first class of deep learning models and consists of an input, hidden and output

layer. The first $L-1$ layers can be thought of as the representation, while the last layer is a linear predictor.

The MLP model is formally defined as follows: The input to the model is a matrix $X \in \mathbb{R}^{n \times d}$ representing a minibatch of n examples, each with d features. The hidden layer, denoted as $H \in \mathbb{R}^{n \times h}$, has h hidden units and is also known as the hidden representation and hidden variable. The hidden and output layers are fully connected, with weights $W^{(1)} \in \mathbb{R}^{d \times h}$ and $W^{(2)} \in \mathbb{R}^{h \times q}$, and biases $b^{(1)} \in \mathbb{R}^{1 \times h}$ and $b^{(2)} \in \mathbb{R}^{1 \times q}$, respectively. The output $O \in \mathbb{R}^{n \times q}$ is calculated by:

$$H = XW^{(1)} + b^{(1)}, \;\; O = HW^{(2)} + b^{(2)}. \tag{A.3}$$

In equation A.3, the hidden units are given by an affine transformation of the input units and the output units are just an affine transformation of the hidden units. This equation represents a linear transformation, but to fully utilize the potential of the MLP network, a non-linear activation function σ is added and applied to each hidden unit after the affine transformation. This output is called *activations*. With activation functions in place, the MLP cannot be collapsed into a linear model anymore.

$$H = \sigma(XW^{(1)} + b^{(1)}), \;\; O = HW^{(2)} + b^{(2)}. \tag{A.4}$$

More complex MLPs can be created by stacking multiple hidden layers on top of each other, which increases the model's expressiveness.

A.1.3 Optimization and learning

The technique of backpropagation, also known as auto-differentiation, is used to calculate the gradient of a loss function $\nabla J(\theta)$. In this process, a loss function is used to compare the predicted output y to the true output y_i of a training sample (x_i, y_i) among n training examples. The input x can represent various data such as images, audio signals or hand crafted features, and the output y are the corresponding class labels in a classification task. We can define a least-square loss function for the i-th case (x_i, y_i) as

$$J(\theta) = \frac{1}{2}(h_\theta(x_i) - (y_i))^2, \tag{A.5}$$

where h_θ is the hypothesis or neural network model. The least-square loss function is one example of a loss function. Other examples will be discussed later when I introduce the concept of the variational autoencoder. It is important to note that the terms "loss" and "cost" are often used interchangeably in literature.

To optimize a loss function, a common approach is to use a gradient descent algorithm. The most widely used algorithm for this purpose is the stochastic gradient descent (SGD):

$$\theta := \theta - \alpha \nabla_\theta J(\theta), \tag{A.6}$$

where $\alpha > 0$ is the learning step size or rate. This algorithm updates the model parameters in the opposite direction of the gradient of the loss function w.r.t to the parameters. The learning rate controls the step size of the update. This process is repeated for multiple iterations or until a stopping criterion is met. With each iteration, the model parameters are updated, and it is expected that the value of the loss function will decrease, resulting in a better model. One of the advantages of SGD is that it can handle large datasets, as it only requires the gradient of a single example at a time rather than the whole dataset. The gradients can be computed quickly due to the hardware parallelization offered by modern GPUs. This is known as mini-batch SGD, where the model parameters are updated simultaneously for a small subset of the training examples. This can lead to faster convergence and better generalization compared to using the whole dataset at once. Another optimization algorithm which is widely used is the Adam optimizer, which is a variant of gradient descent optimization. It generally requires less memory and computation, and it is less sensitive to the choice of learning rate compared to traditional SGD [64]. It is important to keep in mind that there is a lot more to optimizing neural networks than what can be covered in this work. Specific topics such as numerical stability and initialization of neural networks are critical for achieving state-of-the-art performance. For a worked out example of backpropagation see Appendix A.

A.2 Recurrent neural networks

In this section, I introduce a specific type of neural network known as the recurrent neural network (RNN). The primary purpose of using this model class is to analyse and extract useful information and patterns from a series of data collected over time, and in the

case of this work from the virtual marker signal of a behaving animal. Since standard RNNs can suffer from issues related to gradient problems such as exploding or vanishing gradients [**vanishing**], I will introduce the gated recurrent unit (GRU) transfer function, a recently developed solution, that will be utilized here. Then, I show how the RNN model can be extended to a bi-directional recurrent neural network (BiRNN), which allows the network to process information from both the past and future simultaneously.

A.2.1 Computational principles

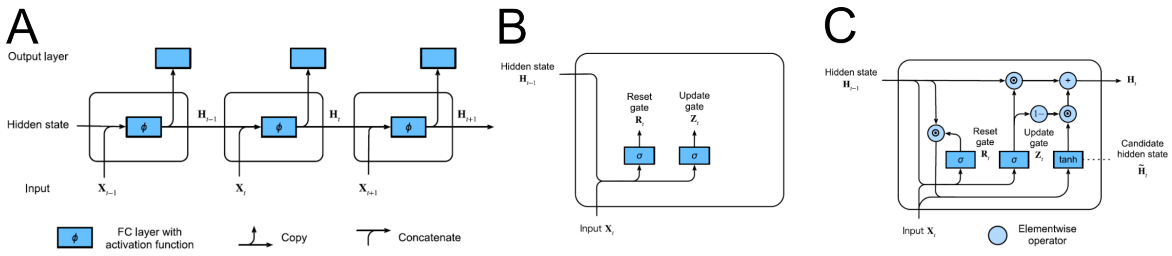

Figure A.2: **Illustration of the computational principles of RNNs and GRUs.** (A) A simple RNN with three recurrent layers. (B) Visualisation for computing the update and reset gate in a GRU model. (C) Visualisation for computing candidate hidden state in a GRU model (modified from [122]).

RNNs are a type of neural network architecture that are specifically designed to handle sequential data such as time series data. Their main purpose is to detect patterns in time series data which can be anything from handwriting, language, genomes, numerical time series data such as stock markets or sensor readings etc. RNNs differ from other neural network architectures such as MLPs in the way information is processed through the network. In traditional feedforward neural networks, there are no cycles present, while RNNs have cycles that allow them to feed information back to itself. This unique functionality allows RNNs to take into account not only the current time step of the data x_t, but also the previous time steps $x_{0:t-1}$ when making predictions or decisions. Mathematically, we can express this as

$$P(x_t|x_{t-1},\ldots,x_1) \approx P(x_t|h_{t-1}), \tag{A.7}$$

where h_{t-1} is a *hidden state* or hidden variable, which stores sequential information up to time step $t-1$. Note that equation (A.7) represents a latent variable model, which we

will discuss in further detail in section 3.1. A hidden state can be computed at any time step t with only the current input x_t and the previous state h_{t-1}:

$$h_t = f(x_t, h_{t-1}). \tag{A.8}$$

It is important to note that the concepts of hidden states and hidden layers in RNNs are different from those in traditional feedforward neural networks (discussed in the previous section). In traditional feedforward neural networks, hidden layers form a direct path from the input to the output, whereas hidden states in RNNs are computed based on previous time steps and are used as input for the next step in the model. This means that hidden states in RNNs are dependent on the sequential nature of the data and allow the network to maintain a "memory" of previous information, while hidden layers in traditional feedforward neural networks do not have this capability.

Figure A.2 (A) displays the computations for a RNN with three recurrent layers. By following the computational graph we see that at any time step t the computation of the hidden state can be dissected into the following two stages: First, the input X_t is concatenated at the current time step t with the hidden state H_{t-1} from the previous time step $t-1$. In a second step, the concatenation result is fed into a fully connected layer with the activation function ϕ. The output results into the hidden state H_t of the current time step t. H_t will now be fed into the next layer to support the computation of H_{t+1} as well as to compute the output O_t at the current time step t. The final hidden state H_f represents the aggregated historical information of the time series and contains all the necessary information about it.

A.2.2 Gated recurrent units

The standard RNN architecture can suffer from numerical instability issues in practice. While techniques such as gradient clipping can help mitigate these issues, there are more advanced methods to overcome these problems. One such method is the use of GRUs, which have become increasingly popular in practice [78]. GRUs are a variation of the RNN architecture that include a gating mechanism for hidden states. This mechanism allows the network to decide when to update or reset the hidden states, rather than relying on pre-defined rules. This adaptability allows the network to focus on the most

important parts of the sequence, and ignore irrelevant or temporary observations. By learning when to update or reset the hidden states, GRUs can improve the stability and performance of RNNs.

Figure A.2 (B) illustrates the computation of the update and reset gate given the input of the current time step t. The output of these two gates are fully connected layers with a sigmoid activation function. To write their mathematical formulation, we consider an input as minibatch $X_t \in \mathcal{R}^{n\times d}$ at time step t and the previous time step $t-1$ hidden state $H_{t-1} \in \mathcal{R}^{n\times h}$. The reset gate notation is $R_t \in \mathcal{R}^{n\times h}$ and the update gate notation is $Z_t \in \mathcal{R}^{n\times h}$. Their computation is:

$$R_t = \sigma(X_t W_{xr} + H_{t-1} W_{hr} + b_r), \;\; Z_t = \sigma(X_t W_{xz} + H_{t-1} W_{hz} + b_z), \tag{A.9}$$

where $W_{xr}, W_{xz} \in \mathcal{R}^{d\times h}$ and $W_{hr}, W_{hz} \in \mathcal{R}^{h\times h}$ are the weight parameters of R_t, Z_t and b_r, b_z are biases. The sigmoid function assures that the output stays in the range $(0, 1)$ to be numerical stable and convex.

The network needs now to identify a *candidate hidden state* $\tilde{H}_t \in \mathcal{R}^{n\times h}$ at time step t. It is called a *candidate* as we need to include the action of the update gate in a next step. Integrating R_t into the equation of a regular latent state update results into:

$$\tilde{H}_t = \tanh(X_t W_{xh} + R_t \odot H_{t-1}) W_{hh} + b_h), \tag{A.10}$$

where $W_{xh} \in \mathcal{R}^{d\times h}$ and $W_{hh} \in \mathcal{R}^{h\times h}$ are weight parameters, and $b_h \in \mathcal{R}^{1\times h}$ is the bias. The symbol $\odot$ represents the Hadamard (elementwise) product operator. The nonlinearity tanh is used to ensure that the values of the candidate hidden state are in the interval range of $(-1, 1)$. In equation (A.10) we can observe that whenever the entries of R_t are close to 1, we recover the standard RNN equation. If the entries are close to 0, preexisting hidden states are reset to their defaults. Figure A.2 (C) visualizes this process. The last step is to integrate the update gate Z_t. This gate determines to which extent the new hidden state $H_t \in \mathcal{R}^{n\times h}$ is either the old state H_{t-1} or how much the new state $\tilde{H}_t$ influences the new hidden state H_t. By simply taking the elementwise convex combination between H_{t-1} and $\tilde{H}_t$, we reach the new hidden state H_t:

$$H_t = Z_t \odot H_{t-1} + (1 - Z_t) \odot \tilde{H}_t. \tag{A.11}$$

Now, whenever the update gate Z_t is close to 1, the old state will be kept. That essentially means that the information of X_t is ignored, which in turns means we effectively skip time step t. On the other hand, if Z_t is close to 0, the updated latent state H_t matches the candidate state $\tilde{H}_t$. The design of the GRU helps with preventing vanishing gradient problems in RNNs and captures better the dependencies for long sequences, as important states from the beginning can be carried to the end, if the update gate stays close to 1 for most of the time steps.

It is worth noting that prior to the introduction of the GRU architecture, one of the earliest methods for addressing the numerical instability issues in RNNs was the long short-term memory (LSTM) network [77]. However, the recent GRU architecture offers similar performance but has the advantage of being significantly faster to compute [81]. Therefore, in this work, the GRU architecture was extensively used for identifying a latent structure in animal motion due to its computational efficiency and performance.

A.2.3 Bi-directional model

Within this work, I will extend the notion of RNNs to bi-directional RNNs (biRNNs) [123]. The main idea behind biRNNs is that they run two RNNs simultaneously, one from the beginning to the end of the sequence and the other from the end to the beginning. This allows the network to process information from both the past and future at the same time, leading to a more comprehensive understanding of the sequence. The addition of another hidden layer with information from the backward pass provides a more flexible way of processing the information. The motivation to use biRNNs in thiw works is that they have access to information from both ends of the input sequence, which is beneficial when trying to identify patterns in animal motion. By taking into account both past and future motion, a biRNN can provide a more comprehensive understanding of the animal's behavior, which can help identify patterns that might be missed by a standard RNN.

I start by defining an input sample as $X_t \in \mathcal{R}^{n\times d}$ at time step t with the hidden layer activation function ϕ. As we have two hidden states in the bi-directional setting, we denote the one in the forward pass as $\overrightarrow{H}_t \in \mathcal{R}^{n\times h}$ and the one in the backward pass as $\overleftarrow{H}_t \in \mathcal{R}^{m\times h}$. Here, h is the number of hidden units. The update rule for the forward and

backward pass is the following:

$$\overrightarrow{H}_t = \phi(X_t W_{xh}^{(f)} + \overrightarrow{H}_{t-1} W_{hh}^{(f)} + b_h^{(f)}), \tag{A.12}$$

$$\overleftarrow{H}_t = \phi(X_t W_{xh}^{(b)} + \overleftarrow{H}_{t-1} W_{hh}^{(b)} + b_h^{(b)}), \tag{A.13}$$

with the weights parameter of the model $W_{xh}^{(f)} \in \mathcal{R}^{d\times h}$, $W_{hh}^{(f)} \in \mathcal{R}^{h\times h}$, $W_{xh}^{(b)} \in \mathcal{R}^{d\times h}$, and $W_{hh}^{(b)} \in \mathcal{R}^{h\times h}$, and bias parameter $b_h^{(f)} \in \mathcal{R}^{1\times h}$, and $b_h^{(b)} \in \mathcal{R}^{1\times h}$.

To obtain the overall hidden state $H_t \in \mathcal{R}^{n\times 2h}$ for the biRNN, we concatenate the forward and backward states $\overrightarrow{H}_t$ and $\overleftarrow{H}_t$ into H_t^c. The output layer $O_t \in \mathcal{R}^{n\times q}$ is then simply computed by

$$O_t = H_t^c W_{hq} + b_q. \tag{A.14}$$

The weight parameter $W_{hq} \in \mathcal{R}^{2h\times q}$ and the bias $b_q \in \mathcal{R}^{1\times q}$ are now the final model parameter. Note that the forward and backward pass can have different numbers of hidden units.

A.3 VAME model selection

The VAME model consists of one biRNN encoder and two biRNN decoder. A HMM is used to identify hidden states (motifs) within our embedding space, as described in the main article (also see Methods). The model was chosen after we tested four different variations of the architecture and compared the HMM against a k-Means algorithm. In Table A.1, we show the different choices and validated their outcome based on our benchmark dataset.

Our architectural choices were either a standard variational autoencoder consisting of a biRNN encoder and a biRNN decoder or with an additional biRNN prediction decoder. Furthermore, we applied to both variants spectral regularization of the latent space [86] to see if this could lead to improved clusterability. We applied three metrics (Purity, NMI, Homogeneity, see Methods) to identify the best model. In both cases (k-Means or HMM), the variational autoencoder model without spectral regularization and an additional decoder had the highest scores. The model with HMM led to the best scores and hence, we chose this as the primary model in our manuscript.

Table A.1: VAME model selection with two different segmentation algorithm (k-Means and HMM) for $k = 50$. Reported is the mean of five repeated training and inference runs.

Model (segmentation)	Purity	NMI	Homogeneity %
VAME single decoder (k-Means)	74.66	22.26	43.02
VAME single decoder + spectral regularization (k-Means)	76.17	22.13	43.20
VAME two decoder (k-Means)	**76.23**	**23.58**	**45.55**
VAME two decoder + spectral regularization (k-Means)	75.56	23.12	44.69
VAME single decoder (HMM)	78.44	26.70	49.82
VAME single decoder + spectral regularization (HMM)	79.81	26.98	51.98
VAME two Decoder (HMM)	**80.66**	**28.61**	**54.85**
VAME two Decoder + spectral regularization (HMM)	79.15	27.64	52.84

A.4 Human phenotyping

For the classification of phenotypes using human experts we have created an online form, where experts could watch all eight videos and make their choice about which phenotype is shown in each video. There was no time limit and the average time to complete the questionnaire was 30 minutes. The participants have not been told how many animals of each group are in the set. For every video, the following five decision could be made: APP/PS1 (Very sure), APP/PS1 (Likely), Unsure, Wildtype (Likely), Wildtype (Very Sure). We have counted a right answers (*Very sure* and *Likely*) as a correct classification (1 point), and wrong answers as well as the choice for the *Unsure* option as wrong classification (0 points). Eleven experts were participating in this classification task. All of them had previous experience with behavioral video recordings in an open field and/or treadmill setting. In addition, six of the participants had previous experience with the APP/PS1 phenotype.

A.5 Community visualization and description

In Figure A.3 we visualized all nine communities by taking the start (cyan color) and end (magenta color) frame for a random community episode. White dots are representing DeepLabCut marker. Next to the visual representation, the DeepLabCut trace for this

episode is shown. Community *a* contains motifs with exploration characteristics such as slow walking and a lot of nose movement which could be interpreted as sniffing. Community *b* shows mainly events in which motifs express rotational behavior. In *c*, the motifs display almost no movement of any body part. Community *d* consists of two motifs which depict transitional behavior from walk to rear or vice versa. In community *e*, we found that all motifs express a specific part of the walking behavior. Community *f* contains motifs which are mainly showing rears along the wall of the arena while *g* contains motifs depicting rears within the arena. Community *h* belongs to the same branch as *g* but portrays mainly motifs with grooming activity. Lastly, community *i* shows motifs in which the animal performs a backward motion e.g after rearing.

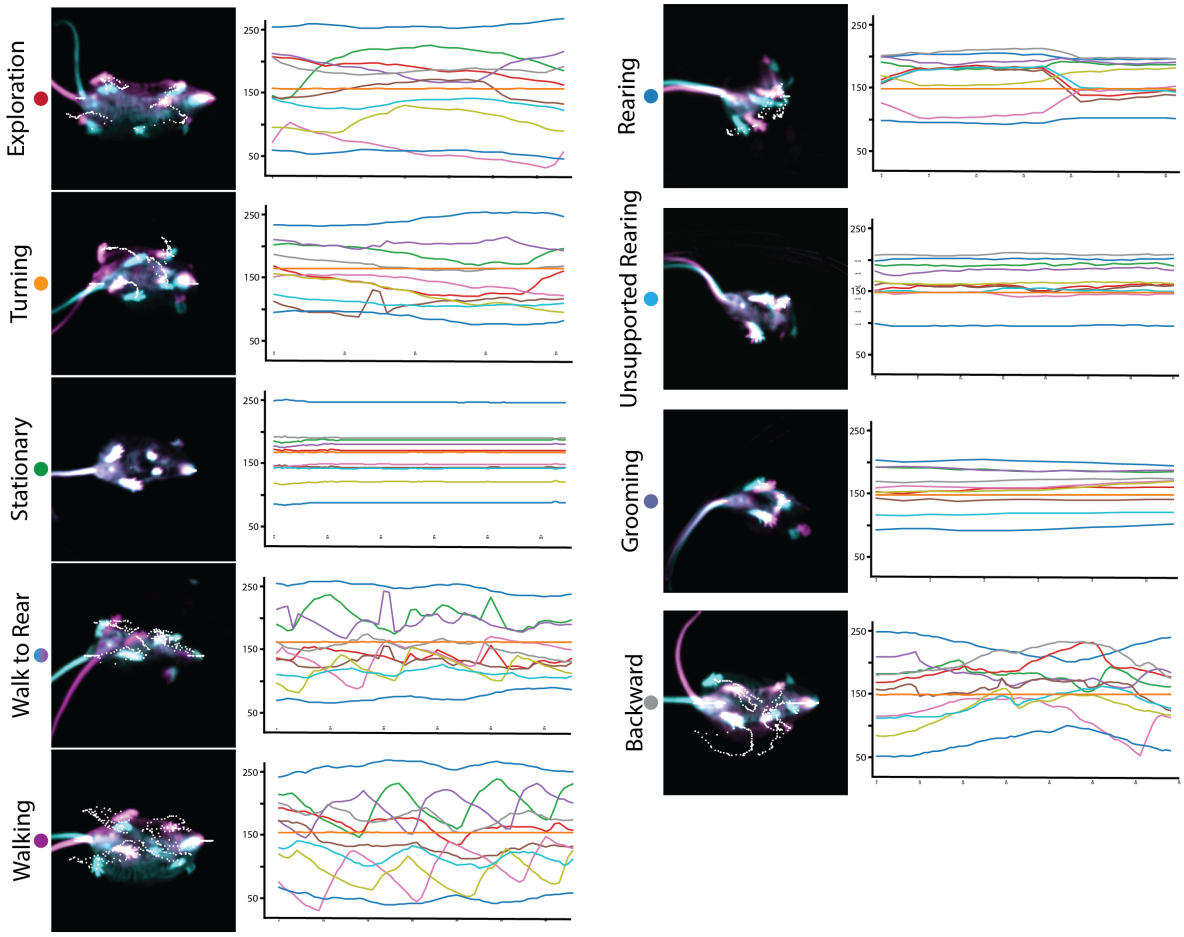

Figure A.3: **Visualization of Communities with their respective DLC trace.** Figure has been modified from [13].

A.6 Community transitions

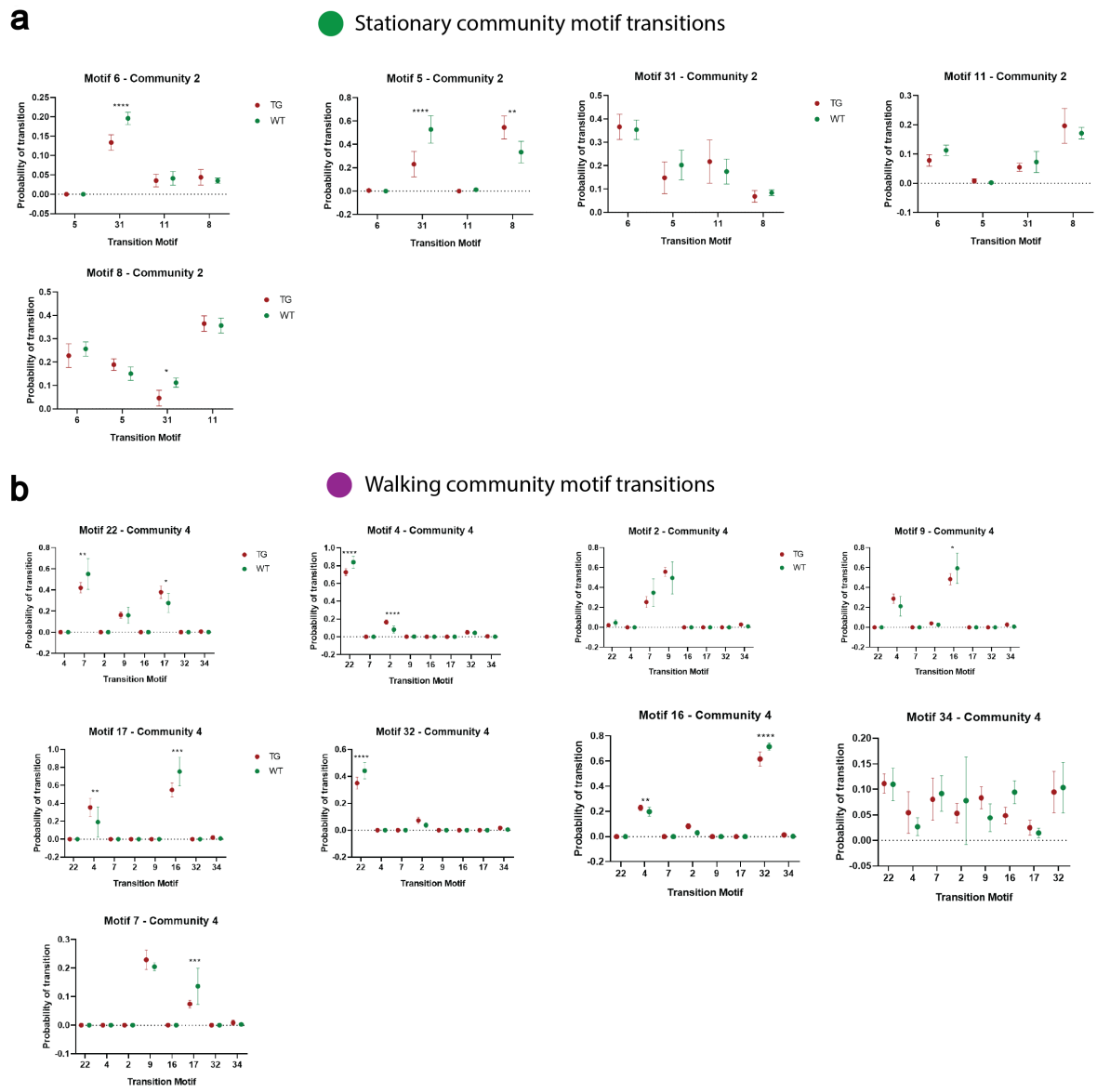

Figure A.4: **Transitions of motif within the *Stationary* and *Walking* community.** Error bars represent standard deviation. Figure has been modified from [13].