MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

„Learning representations from time-series data with multi-step predictions"

verfasst von / submitted by

Dott. Vittorio Boarini

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien, 2025 / Vienna, 2025

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	UA 066 921
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Masterstudium Informatik
Betreut von / Supervisor:	Univ.-Prof. Dipl.-Ing. Dr. -Ing. Moritz Grosse-Wentrup

Abstract

Understanding how neural activity drives behavior requires analyzing high-dimensional, noisy neuronal time-series data. Current representation learning approaches like BunDLe-Net compress neuronal signals into low-dimensional embeddings that preserve behavioral information, but rely on single-step prediction that limits their ability to capture long-term temporal dependencies and behavioral motifs unfolding over extended periods.

This thesis investigates whether integrating multi-step prediction into neuronal representation learning frameworks can improve the modeling of long-term neural dynamics by extending the BunDLe-Net architecture. Using neuronal recordings from C. elegans during oxygen stimulation experiments, the research compares qualitatively and quantitatively several multi-step prediction approaches including multiple-output, recursive, and direct prediction strategies.

The multiple-output multi-step BunDLe-Net, which predicts all future states simultaneously in a single forward pass, substantially outperformed single-step approaches by learning rich, interpretable neural representations that clearly separated distinct behavioral responses. These results show that multi-step prediction significantly improves the modeling of long-term neural dynamics, producing more interpretable and behaviorally informative representations.

Kurzfassung

Das Verständnis dafür, wie neuronale Aktivität Verhalten steuert, erfordert die Analyse hochdimensionaler, verrauschter neuronaler Zeitreihendaten. Aktuelle Ansätze des Repräsentationslernens wie BunDLe-Net komprimieren neuronale Signale in niedrigdimensionale Einbettungen, die Verhaltensinformationen bewahren, aber auf einstufige Vorhersagen angewiesen sind, was ihre Fähigkeit einschränkt, langfristige zeitliche Abhängigkeiten und Verhaltensmotive zu erfassen, die sich über längere Zeiträume entfalten.

Diese Arbeit untersucht, ob die Integration mehrstufiger Vorhersagen in neuronale Repräsentationslernframeworks die Modellierung langfristiger neuraler Dynamiken verbessern kann, indem die BunDLe-Net-Architektur erweitert wird. Unter Verwendung neuronaler Aufzeichnungen von C. elegans während Sauerstoffstimulationsexperimenten vergleicht die Forschung qualitativ und quantitativ verschiedene mehrstufige Vorhersageansätze, einschließlich multiple-output, recursive und direct Vorhersagestrategien.

Das mehrstufige BunDLe-Net mit multiple-output, das alle zukünftigen Zustände gleichzeitig in einem einzigen Vorwärtsdurchlauf vorhersagt, übertraf einstufige Ansätze erheblich, indem es reichhaltige, interpretierbare neurale Repräsentationen lernte, die verschiedene Verhaltensreaktionen klar voneinander trennten. Diese Ergebnisse zeigen, dass mehrstufige Vorhersagen die Modellierung langfristiger neuraler Dynamiken erheblich verbessern und interpretierbarere sowie verhaltensrelevantere Repräsentationen erzeugen.

Contents

List of Figures

1. Introduction

The human brain processes information through networks of neurons that continuously generate vast amounts of complex, high-dimensional signals. Understanding the relationship between this neural activity and the resulting behavior is fundamental to neuroscience (Saxena and Cunningham, 2019; Vyas et al., 2020). Recent advances in recording technologies now allow researchers to increase by orders of magnitude the number of neurons that can be monitored simultaneously (Stevenson and Kording, 2011; Ahrens et al., 2013; Schrödel et al., 2013; Prevedel et al., 2014). However, extracting meaningful insights from these signals remains challenging due to the sheer volume and complexity of the data, as neuronal time-series recordings are inherently high-dimensional, noisy, and computationally demanding to interpret (Paninski and Cunningham, 2018).

Therefore, the complexity of these datasets often requires extensive manual analysis and substantial domain expertise to extract meaningful insights about neural-behavioral relationships. Representation learning, a subfield of machine learning, offers a powerful paradigm for addressing these challenges. These techniques are designed to automatically learn the most informative features in high-dimensional data. A common approach is dimensionality reduction, which maps data into a lower-dimensional space that preserves the essential information while dramatically reducing computational complexity (Cunningham and Yu, 2014). By learning meaningful abstractions of neural dynamics, dimensionality reduction methods allow researchers to visualize, analyze, and interpret neural data to decode the latent structures that link neural dynamics to behavioral outputs. The key insight underlying these methods is that while neural activity may appear high-dimensional and complex, the underlying behavioral and dynamical patterns often reside in much lower-dimensional spaces that can be discovered through appropriate computational techniques (Meilă and Zhang, 2023).

Current approaches for neuronal manifold learning, such as BunDLe-Net (Kumar et al., 2023), have demonstrated success in capturing the essential features of neural dynamics that relate to behavior. This method compresses high-dimensional neuronal signals into low-dimensional embeddings that preserve both behavioral and dynamical information, enabling researchers to decode behavioral states from neural activity while maintaining interpretable representations of the underlying neural processes. The core mechanism of BunDLe-Net, however, is rooted in a single-step prediction strategy: given the current neural state, it models only the very next time point. While effective for modeling immediate state transitions, this approach limits the ability of the model to capture behavioral motifs unfolding over extended timescales.

This single-step limitation represents a significant constraint that prevents current methods from capturing the full richness of neural dynamics. Neural activity patterns often unfold over extended time periods, with behavioral motifs emerging through complex temporal sequences that span multiple time steps (Vyas et al., 2020). The inability of current single-step approaches to model these extended temporal dependencies means that they may miss critical behavioral motifs that emerge over longer time scales. This limitation is particularly problematic in the context of noisy neural data, where single-step predictions may be overly sensitive to momentary fluctuations while missing more consistent long-term patterns. The focus on immediate predictions also means that learned representations are optimized primarily for very short-term dynamics, potentially overlooking slower, longer-scale patterns that may be equally or more important for understanding neural-behavioral relationships.

A significant limitation is that single-step forecasting models may produce embeddings that are insufficient for understanding the long-term causal pathways that underlie and shape how neural activity drives behaviors. Therefore, the integration of multi-step prediction capabilities into neuronal representation learning frameworks represents a natural evolution of these methods. Multi-step prediction approaches could capture richer information about the longer-term evolution of neuronal dynamics, enabling models to more effectively distinguish between initially similar neuronal states that eventually diverge into distinct behavioral motifs over time. By forecasting multiple future states simultaneously, these methods can learn embeddings that encode temporal dependencies across extended time horizons, potentially revealing patterns that may otherwise not be visible with single-step approaches (Oord et al., 2018).

To bridge the gap between single-step forecasting models and the need for more informative, longer-horizon predictions, this thesis investigates the integration of multi-step prediction into existing neuronal representation learning frameworks, specifically focusing on extending the BunDLe-Net architecture.

1.1. Research Questions

The central goal of this thesis is to address the limitations identified in single-step forecasting models, the challenge to capture long-term temporal dependencies in neural data. Therefore, the study focuses on investigating whether multi-step prediction techniques can improve neuronal representation learning to capture longer-term temporal dependencies in neural dynamics.

To investigate this, the thesis is guided by a central research question:

Can the integration of multi-step prediction into neuronal representation learning frameworks improve the modeling of long-term neural dynamics?

To address this central question, this research will investigate the following specific sub-questions:

1. *To what extent does the recorded neural activity contain predictive information across extended time horizons?*

   Do neuronal time-series recordings contain sufficient predictive information to support multi-step forecasting across extended time horizons? Understanding the temporal extent of predictable information in neural data is necessary to determine the feasibility of multi-step prediction methods.

2. *Can the multi-step predictive structure in the neural data be preserved under compression?*

   Can the temporal dependencies identified in neural data be effectively preserved when compressed into low-dimensional representations? This question is necessary to validate the compatibility of multi-step prediction with the dimensionality reduction approach that underlies neuronal representation learning methods.

3. *How can multi-step prediction be integrated in neuronal representation learning frameworks?*

   What architectural modifications and approaches are well-suited for implementing multi-step prediction into the BunDLe-Net representation learning framework? This question investigates different potential strategies for multi-step forecasting, examining multiple-output, recursive, and direct methods.

4. *Do embeddings learned with multi-step forecasting diverge qualitatively or quantitatively from those learned with single-step objectives in their ability to separate behaviorally meaningful motifs?*

   Does the integration of multi-step prediction capabilities improve the ability to learn behaviorally meaningful representations compared to existing single-step approaches? This question evaluates whether the additional temporal information captured by multi-step prediction translates into more interpretable models.

1.2. Structure of the Thesis

This thesis is organized in six chapters that sequentially address the research questions outlined above and provide an investigation on multi-step prediction techniques for neuronal representation learning.

Chapter 1: Introduction presents the context and motivation for the research. It introduces the fundamental challenge of understanding neural-behavioral relationships from complex, high-dimensional time-series data. It outlines the limitations of current single-step prediction models, defines the central problem statement, and presents the research questions that guided this investigation. This chapter concludes with an overview of the thesis structure.

Chapter 2: Background provides the necessary theoretical and technical background for the topics explored in the thesis. It covers the core principles of representation learning, as well as an overview of existing methods for dimensionality reduction in neuroscience, with a particular focus on the BunDLe-Net architecture that serves as the foundation for this work. The chapter concludes by reviewing relevant literature on multi-step time-series forecasting, examining multiple-output, recursive, and direct methods.

Chapter 3: Methodology details the empirical approach taken to answer the research questions. It begins with a description of the neuronal dataset used for the experiments. This is followed by three progressive experimental phases:

1. **Data Examination** addresses Research Question 1 by investigating whether neuronal time-series recordings contain sufficient predictive information to support multi-step forecasting across extended time horizons. This step uses feed-forward architectures to validate the fundamental assumption that future neural states can be predicted from current observations beyond immediate transitions.
2. **Compression Feasibility** addresses Research Question 2 by determining whether the temporal dependencies identified in Step 1 can be effectively preserved when compressed into low-dimensional representations. This step introduces bottleneck architectures to assess the compatibility of multi-step prediction with the dimensionality reduction requirements of neuronal representation learning.
3. **Integration in BunDLe-Net** addresses Research Question 3 by implementing the integration of multi-step prediction into the BunDLe-Net architecture. This step explores the architectural approaches considered for incorporating multi-step forecasting.

Chapter 4: Results presents the empirical findings of the multi-step prediction integration in BunDLe-Net which is described in Chapter 3. This chapter provides the quantitative and qualitative analysis of the integration.

Chapter 5: Discussion discusses the findings from the experiments described in Chapter 4. It addresses Research Question 4 by analyzing whether multi-step embeddings qualitatively and quantitatively improve the separation of behaviorally meaningful motifs compared to the original single-step model.

Chapter 6: Conclusion summarizes the research, synthesizes the key findings, and reflects on the contributions of the thesis.

2. Background

This chapter provides the theoretical and methodological foundation necessary for understanding the contributions of this thesis, which combines principles from representation learning and multi-step time-series forecasting to capture longer-term dependencies in neural dynamics. It first introduces representation learning as a framework to automatically discover meaningful features from raw data, explaining why these techniques are valuable for analyzing high-dimensional neuronal recordings. Then, it reviews methods for neuronal manifold learning to discover low-dimensional structures in neural activity data and provides an overview on the BunDLe-Net architecture which serves as the foundation for this work. Finally, it summarizes multi-step time-series prediction strategies to forecast longer-horizon dynamics.

2.1. Representation learning

Representation learning is a broad class of machine learning techniques that aim to automatically discover meaningful and useful features from raw data. Rather than relying on hand-crafted features designed by domain experts, representation learning algorithms seek to automatically identify the underlying structure and patterns in the data that are relevant for a given task. Therefore, their primary aim is ultimately to transform the raw data into a new format that makes it easier to extract valuable information. By converting data into these more meaningful representations, machine learning methods can use it more effectively for a variety of subsequent tasks, such as classification, retrieval, or clustering (Zhong et al., 2016). This transformation process is critically important as the choice of data representation has a significant impact on the performance and success of machine learning algorithms (Bengio et al., 2013).

The principles of representation learning are particularly valuable in neuroscience, where researchers face the challenge of extracting meaningful information from high-dimensional, noisy neural recordings. Research suggests that neural activity often lies on lower-dimensional manifolds, meaning that despite the high-dimensional nature of the recorded data, the underlying neural computations may be governed by a much smaller number of latent variables (Cunningham and Yu, 2014; Gallego et al., 2017; Meilă and Zhang, 2023). For this reason, representation learning can offer a suitable approach for dimensionality reduction by automatically discovering the most relevant low-dimensional features from neural time-series data. In the context of neuroscience, this means learning compact embeddings of neural activity that preserve both the temporal dynamics of neural activity and their relationship to observed behavior. Ultimately, an effective representation can reveal latent structures in neural dynamics, support the visualization of complex activity patterns, and help to understand how brain activity drives behaviors.

2.1.1. Neuronal manifold learning

A classical approach to neuronal manifold learning is Principal Components Analysis (PCA) (Ahrens et al., 2012; Kato et al., 2015). It's a widely used linear dimensionality reduction technique that works by identifying the directions of maximum variance in the data, reducing dimensionality while preserving as much information as possible. However, its linearity limits its capacity to capture non-linear neural dynamics. For exploratory visualization beyond linear subspaces, a couple of dimensionality reduction methods have gained popularity. t-Distributed Stochastic Neighbor Embedding (t-SNE) (van der Maaten and Hinton, 2008) preserves local neighborhood structure while revealing non-linear relationships in neural data. Specifically, it embeds high-dimensional data in a two or three dimensional space by matching neighborhood probabilities. However, t-SNE is primarily a local neighborhood method, with limited preservation of the global structure of the data. Uniform Manifold Approximation and Projection (UMAP) (McInnes et al., 2018) addresses some of t-SNE's limitations by preserving both local and global manifold structure while typically providing faster computation. Although both methods excel at revealing hidden structures in neural data, they lack the ability to model temporal dependencies. Autoencoders (AEs) (Hinton and Salakhutdinov, 2006; Pandarinath et al., 2018) have emerged as a deep learning alternative for learning non-linear, low-dimensional representations of neural activity using neural networks. They use a non-linear encoder–decoder pair to compress data to a low-dimensional space and then reconstruct the input. An AE's encoder transforms input data into a low-dimensional latent space, while its decoder reconstructs the original input from this embedding. Despite their flexibility, standard AEs often prioritize reconstruction fidelity over behavioral relevance.

However, all these methods primarily optimize for reconstruction quality of neuronal data, preserving all recorded information regardless of behavioral relevance. As a result, while they can accurately reconstruct or clearly visualize neural activity, they often retain noise, ignore behavioral context, and create embeddings that don't capture the causal structure of neural dynamics. BunDLe-Net (Kumar et al., 2023) addresses this by learning low-dimensional embeddings specifically to predict the future behaviors: it jointly optimizes both encoding neuronal dynamics to a lower dimensional space and forecasting the next state of behavioral activity. This aligns the representations with behaviorally relevant dynamics rather than mere reconstruction.

2.1.2. BunDLe-Net

BunDLe-Net (Kumar et al., 2023) is a neural network architecture for neuronal manifold learning. It was designed for dimensionality reduction of high-dimensional neuronal time-series data, while preserving behaviorally relevant dynamics. Developed to address the challenges of interpreting complex neuronal recordings, BunDLe-Net learns meaningful low-dimensional Markovian embeddings that capture the essential structure of neuronal dynamics while filtering out noise and irrelevant information.

To achieve this, BunDLe-Net uses an architecture composed of two main branches and is optimized using two loss functions: dynamic and behavioral. Ultimately, at each time t a window of the previous n time steps of neural activity $\{X_{t-n}, \ldots, X_{t-1}, X_t\}$ is encoded into a single latent state Y^L_{t+1}, using a single-step prediction approach. Specifically, in the lower branch, an encoder *tau* transforms the sliding window of neuronal trajectories into a low-dimensional latent state Y_t. Next, a transition model T_y forecasts the change ΔY_t from the current state to the subsequent one, generating an estimate of the latent state Y^L_{t+1} at time $t+1$. This forecasted latent state is then compared with the actual latent state obtained by the upper branch. Through the joint optimization of the encoder τ and the transition model T_y to reduce the Markovian loss $\mathcal{L}_{Markov}$, tfhe result is a low-dimensional latent time-series Y_{t+1} that inherently satisfies Markovian properties. Nevertheless, this alone does not guarantee a meaningful representation, since a simple projection to a constant value could satisfy Markovian conditions without capturing meaningful information. In order to achieve an informative latent space, the architecture needs to ensure that the behavioral context can be decoded from the latent representation Y_{t+1}. This is accomplished by incorporating in the model a behavioral loss $\mathcal{L}_{Behavior}$, which quantifies the discrepancy between the actual behavioral labels B_{t+1} and the predictions $\tilde{B}_{t+1}$ derived from the latent states. By jointly training the encoder *tau* and transition model T_y to minimize both $\mathcal{L}_{Markov}$ and $\mathcal{L}_{Behavior}$, BunDLe-Net generates low-dimensional, Markovian embeddings that encode the components of the original high-dimensional neuronal trajectories most relevant to the behavioral context.

In practice, *tau* has been implemented as a sequence of ReLU layers followed by a normalization layer. To prevent overfitting, the *tau* layer also adds Gaussian white noise into the latent space. The *predictor* and T_y layers each consist of a single dense layer. Finally, the embedding dimension is set to three.

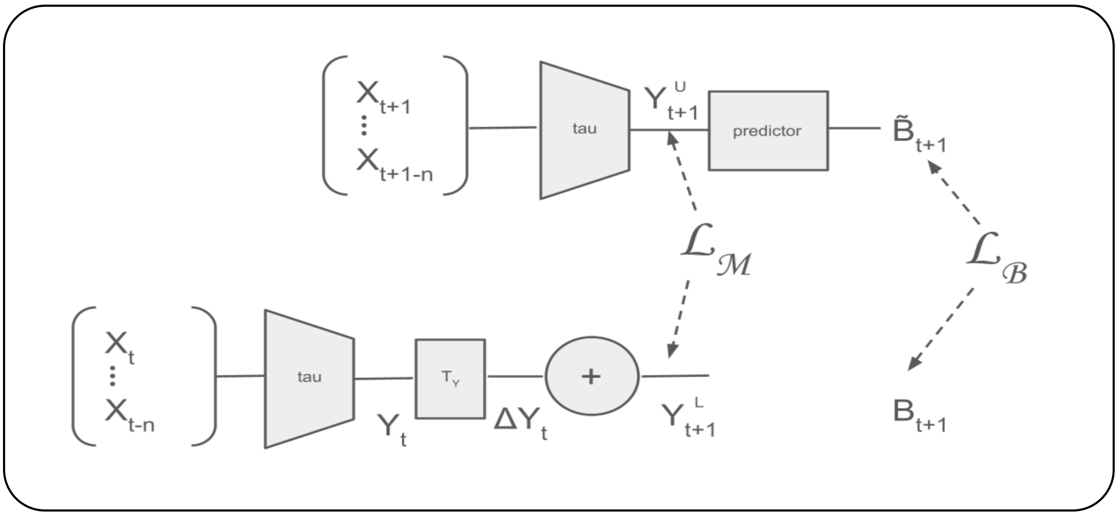


Figure 2.1.: BunDLe-Net Architecture

However, its current single-step prediction horizon optimizes the embeddings only for the immediate next state, which can miss motifs unfolding over longer horizons, may amplify noise sensitivity, and ultimately may generate less meaningful low-dimensional abstractions of the underlying dynamics. This work addresses these limitations by extending BunDLe-Net to forecast an entire sequence of future states, forcing the embeddings to encode the longer-term evolution of neuronal dynamics.

2.2. Time-Series Forecasting

The primary objective of time-series forecasting is to develop a model based on historical observations to predict future values of a time-series. A time-series can be defined as a sequence of observations $\{x_{t0}, x_{t1}, \ldots, x_{tn}\}$ indexed by time, where each observation x_t represents the values of a variable at time t. In this context, multi-step forecasting refers to forecasting an entire sequence of future values over an extended horizon, unlike single-step forecasting, which forecasts only the immediate next value in a sequence. Therefore, in multi-step forecasting, the prediction task involves estimating future values $\{x_{t+1}, x_{t+2}, \ldots, x_{t+h}\}$ given the historical sequence up to time t, where h represents the prediction horizon.

2.2.1. Multi-Step Forecasting

Classical multi-step prediction approaches can be grouped into single-output and multiple-output strategies. The two major strategies for single-output multi-step prediction are the recursive approach and the direct approach. In the recursive strategy, a single one-step model is applied iteratively to forecast future values. Specifically, there is a single-step *predictor* which is trained to forecast one step ahead, and multi-step predictions are obtained by iteratively applying this *predictor*, using the previous prediction value as input for the subsequent prediction (Kline, 2004; Cheng et al., 2006; Hamzaçebi et al., 2009). On the other hand, the direct strategy uses multiple models, one per time horizon. A separate, independent model is trained for every step in the prediction horizon. Each model will directly return the prediction for its respective time step (Kline, 2004; Cheng et al., 2006; Hamzaçebi et al., 2009). In contrast to these single-output approaches, multiple-output strategies use a single model which is trained to predict all future time steps simultaneously. In a single forward pass, the model returns a vector of all values of the prediction horizon (Kline, 2004; Bontempi, 2008; Bontempi and Ben Taieb, 2011).

In the context of extending BunDLe-Net with multi-step forecasting, this study will investigate and compare all three multi-step prediction strategies.

3. Methodology

This chapter presents the empirical approach developed to investigate whether multi-step prediction techniques can improve the interpretability and structure of low-dimensional representations of neural dynamics. The methodology is structured as a multiple-phase investigation designed to incrementally validate the feasibility and effectiveness of extending BunDLe-Net from single-step to multi-step prediction.

It begins by describing the curated dataset used in the study: recordings of neuronal activity under controlled stimuli, with expert-annotated labels. Then it presents the individual phases, which systematically address: (1) whether the neuronal dataset contains sufficient predictive information for multi-step forecasting, (2) whether this temporal information can be preserved through dimensionality reduction, and (3) how multi-step prediction can be effectively integrated into the BunDLe-Net architecture. Finally, it outlines the quantitative and qualitative evaluation methodology designed to compare the performance of multi-step approaches against the original BunDLe-Net architecture.

3.1. Dataset

The dataset used in this study consists of neuronal time-series recordings from *Caenorhabditis elegans* (C. elegans) worms. Neuroscientists obtained these recordings using calcium imaging techniques, which allowed them to capture the real-time activity of neurons during behavioral experiments. These experiments were specifically designed to trigger responses while simultaneously monitoring neural activity as worms engaged in various different behaviors.

The selection of this specific dataset was based on several criteria that align with the objectives of our study, investigating how neural activity drives behavior. The dataset offers high-dimensional neural data of nine worms, capturing the activity of numerous neurons throughout experimental sessions. The number of neurons varies from 61 to 133, depending on the individual worm. Additionally, it includes a variety of distinct behaviors, along with multiple repetitions of these behavioral states over time. Notably, each time step is manually annotated by neuroscientists based on manual analysis and domain expertise. Specifically, the observed behaviors were: "forward", "reverse", "sustained reversal", and "turn".

Most importantly, the dataset features experiments specifically designed to trigger different behavioral responses through oxygen manipulation. This included a baseline environment with 11% oxygen (no stimulus) and an elevated 21% oxygen condition (stimulus). Each experiment spanned 1890 seconds. The initial 450 seconds were consistently at the baseline of 11% oxygen. This was followed by multiple 90-second trials: each trial began with 30 seconds of the 21% oxygen stimulus, followed by 60 seconds at 11% oxygen. Neuroscientists manually reviewed and classified the worms' responses to the oxygen stimulus into two categories: "hit" responses for observable behavioral reactions and "miss" responses if no reaction occurred. However, some trials were excluded.

This curated dataset therefore provides an ideal foundation for understanding how neural activity drives behavior. The expert-annotated outcomes serve as the ground truth for evaluating whether the low-dimensional representations learned by our model can naturally separate different behavioral outcomes, even without explicit supervision.

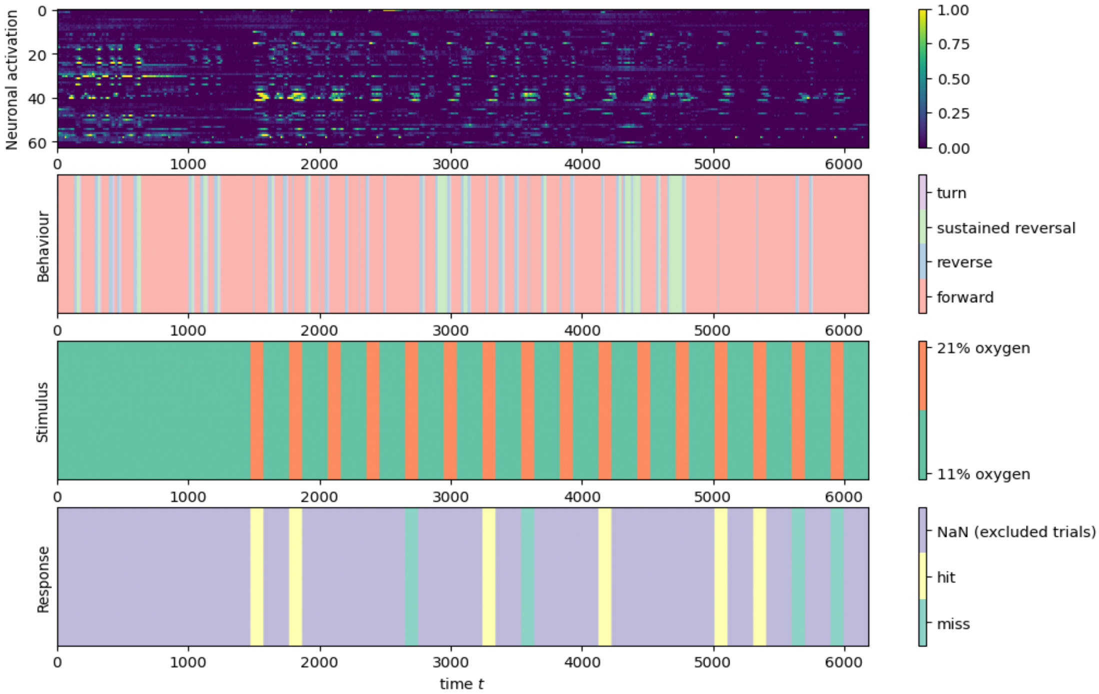


Figure 3.1.: Visualization of the dataset for one of the worms

3.2. Step 1: Data Examination

The initial phase of this work focuses on examining neuronal recordings to determine whether they are both suitable and sufficient for developing a multi-step prediction model. This preliminary analysis aims to validate the fundamental assumption that future neuronal states can be predicted from current observations across extended time horizons. Therefore, the primary objective of this step is to verify that the dataset contains the temporal dependencies and contextual information necessary to forecast future behavioral outcomes from previous neural activity. This is necessary because the dataset can be used to evaluate the extension of the BunDLe-Net architecture from single-step to multi-step forecasting, only if the data contains predictive information beyond immediate transitions. Ultimately, the inherent predictability of the dataset is established by quantifying how well past neuronal states can anticipate future behavioral labels across a range of time horizons.

3.2.1. Approaches

The examination focused on evaluating the predictive capacity of the neural recordings using feed-forward neural networks architectures. In particular, two distinct approaches were implemented to assess the suitability of the dataset for multi-step prediction. The first approach used a single neural network that predicts all future time steps simultaneously in one forward pass (multiple-output), while the second approach employed multiple individual neural networks, each dedicated to predicting a specific future time step (direct). These architectures were designed to isolate the core question of temporal predictability without the complexity of dimensionality reduction or integration with the existing BunDLe-Net framework.

The multiple-output architecture consisted of one feed-forward neural network that takes the current neuronal state X_t as input and produces predictions for all n future behavioral states (B_{t+1}, B_{t+2}, ..., B_{t+n}) simultaneously. In contrast, the direct approach employed separate individual neural networks, each trained to predict a specific future time step from the current neuronal state. Therefore, in this design, each network is optimized exclusively for its target horizon, learning a direct mapping from current neural activity to a specific future behavioral state. So, the first model predicts B_{t+1} from X_t, the second model predicts B_{t+2} from X_t, and so forth.

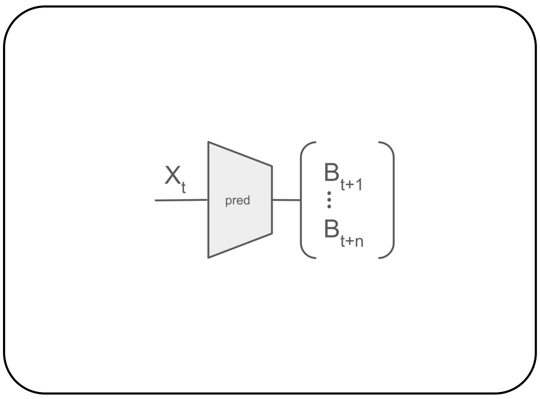


Figure 3.2.: Multiple-Output Feed-Forward Architecture

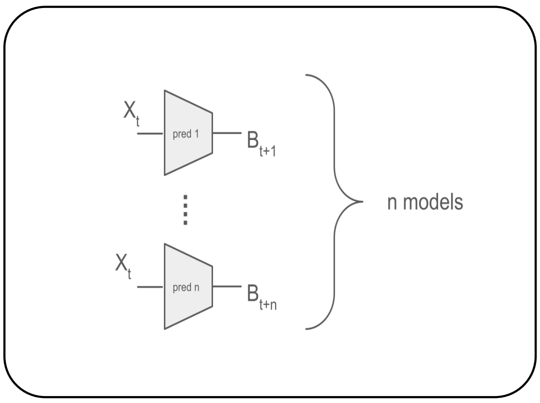


Figure 3.3.: Direct Feed-Forward Architecture

3.2.2. Evaluation

The examination process consisted in training both the multiple-output and the direct architectures on the neural dataset, in order to evaluate their ability to predict future behavioral states. The dataset was partitioned into training and testing sets using an 80-20 split. Given the temporal nature of the data, the initial 80% of the recordings was used as the training set, while the subsequent 20% was reserved for testing to ensure predictions were made on unseen future data and prevent any data leakage.

The performance of the models was assessed by quantifying the prediction accuracy across extended time horizons. This allows to determine the temporal extent of predictable information intrinsic in the neural data. At each time step, the model was tasked with predicting the behaviors over the following forty time steps. For each predicted time horizon, from one to forty, the accuracy of the model was defined as the percentage of correctly predicted behavioral states at that specific future time step.

In order to establish a baseline for comparison, a chance level accuracy was also calculated. It was obtained by shuffling the behavioral labels across the dataset prior to training. This shuffling process deliberately disrupts any genuine temporal alignment between neural activity and behavior, providing an accuracy benchmark representing scenarios where no predictive information is present. An additional baseline ("Prediction previous state") was included using a naive predictor, which assumes the persistence of the previous state for all future predictions. The naive baseline represents the true behavior at time t forecasted for all future time steps of the forecasting horizon, while the models never receive this state as input, they must infer the state and predict the future ones using only the high-dimensional, noisy neural activity X_t. Ultimately, the presence of meaningful predictive information can be determined by comparing the performance of the models against these baselines.

3.2.3. Discussion

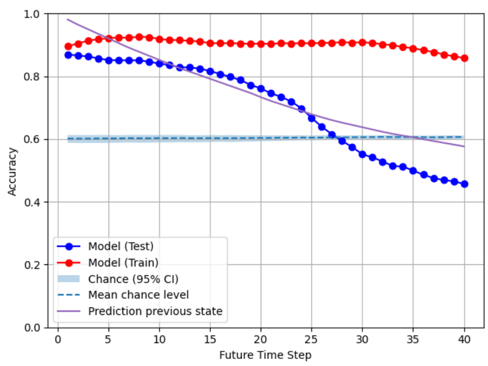


Figure 3.4.: Feed-Forward Accuracies, Multiple-Output

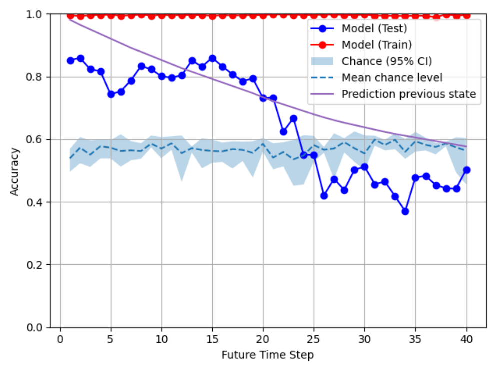


Figure 3.5.: Feed-Forward Accuracies, Direct

The results of this examination show that the neuronal dataset contains predictive information for future behavioral states. Both architectural approaches demonstrated the ability to forecast behavioral outcomes beyond the immediate next time step, with accuracies above the chance level for over twenty-five future time steps. This confirms that the neural recordings contain sufficient temporal information extending across multiple time points. However, the models failure to outperform the naive predictor baseline indicates that, without any architectural constraints, the raw, high-dimensional neural data is too noisy or complex to extract good predictive signals. Therefore, this data examination phase confirmed that the neuronal dataset possesses enough information to support multi-step prediction, while showing the need for architectural modifications. Consequently, this examination both validates the feasibility of multi-step prediction and motivates the next phase, which will test whether the temporal information can be effectively compressed into low-dimensional representations that retain above-chance predictive capability and ideally surpass the persistence baseline.

3.3. Step 2: Compression Feasibility

Having established that the neuronal dataset contains sufficient predictive information for multi-step forecasting, the second phase of this work focuses on determining whether this temporal information can be effectively compressed into low-dimensional representations while preserving predictive capability. This step is essential for validating the feasibility of integrating multi-step prediction with the dimensionality reduction approach of BunDLe-Net, which relies on learning low-dimensional embeddings that capture the essential dynamics of high-dimensional neuronal data.

The primary objective of this compression feasibility analysis is to assess whether the temporal dependencies identified in Step 1 can be preserved, and ideally made more predictive, through a dimensionality reduction bottleneck layer. This is essential because the BunDLe-Net architecture fundamentally depends on the ability to compress neuronal signals into low-dimensional embeddings that retain behaviorally relevant information. Therefore, demonstrating that multi-step predictive information can be preserved through compression process is necessary to justify the subsequent integration of multi-step prediction into the BunDLe-Net architecture.

3.3.1. Approaches

The feasibility of information compression was assessed using neural network architectures with bottleneck layers. Two distinct approaches were implemented, mirroring the design patterns established in Step 1 while adding dimensionality reduction capabilities. The first approach used a single neural network with a bottleneck layer that compresses the current neuronal state into a low-dimensional embedding, from which all future behavioral states are predicted simultaneously in one forward pass. The second approach employed multiple separate neural networks, each containing its own bottleneck layer and focused on predicting a specific future time step.

The multiple-output bottleneck architecture consisted of an encoder that transforms the high-dimensional current neuronal state X_t into a three dimensional compressed representation, followed by a predictor that generates forecasts for all n future behavioral states (B_{t+1}, B_{t+2}, ..., B_{t+n}) from this low-dimensional embedding. In contrast, the direct bottleneck approach used separate individual neural networks, each incorporating its own encoder-decoder structure dedicated to a specific prediction horizon. Each network independently compresses the current neuronal state X_t into a three dimensional embedding for its particular prediction target, then generates a forecast for its assigned future time step.

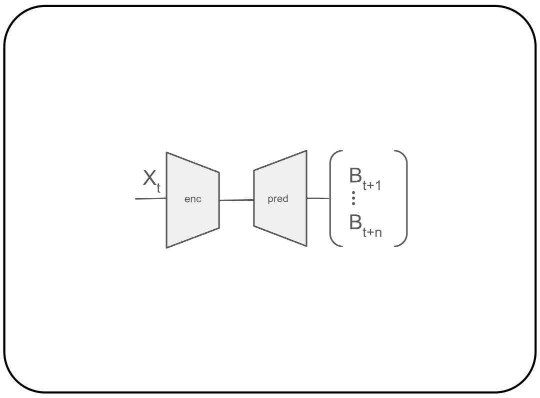


Figure 3.6.: Multiple-Output Bottleneck Architecture

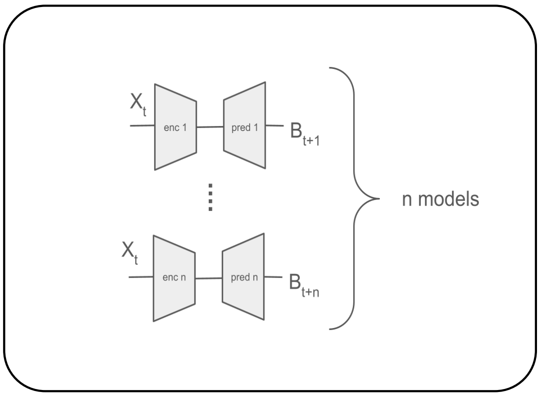


Figure 3.7.: Direct Bottleneck Architecture

3.3.2. Evaluation

The evaluation followed the same approach described in the previous step, with the dataset partitioned into training and testing sets using an 80-20 temporal split to prevent data leakage. Models were trained to predict behavioral states over the following forty time steps, with performance evaluated through prediction accuracy at each predicted future time horizon. Similarly, the comparison with the persistence and the chance-level baselines was still used for determining whether meaningful predictive information was preserved through the compression process.

3.3.3. Discussion

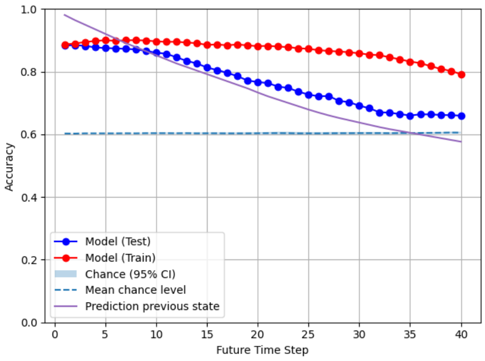


Figure 3.8.: Bottleneck Accuracies, Multiple-Output

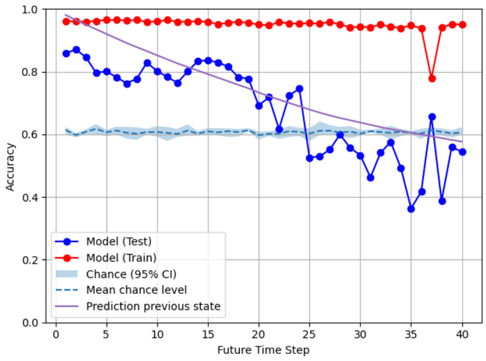


Figure 3.9.: Bottleneck Accuracies, Direct

The results of this compression feasibility analysis show that temporal predictive information can indeed be effectively preserved through dimensionality reduction. Both bottleneck architectures successfully maintained prediction accuracies above chance levels across extended time horizons, for over twenty-five future time steps. This confirms that temporal predictive information can be effectively preserved in low-dimensional representations, validating the core assumption necessary for extending BunDLe-Net to multi-step prediction. Additionally, the ability of the multiple-output model to now outperform the naive baseline over longer horizons demonstrates the value of dimensionality reduction itself. The compression forces the model to learn a denoised, essential representation of the neural state, that is able to effectively capture meaningful predictive signals beyond simple temporal persistence.

Based on the consistent demonstration across both Step 1 and Step 2 that above chance-level predictive information can be preserved for over twenty-five future time steps, the prediction horizon for the subsequent BunDLe-Net integration was set to twenty-five steps. Furthermore, the comparison between multiple-output and direct approaches showed that the multiple-output approach is more accurate and consistent, while also requiring significantly less computational overhead and training time. This finding led to the decision to pursue the multiple-output approach for subsequent integration with BunDLe-Net, as it provides a more efficient solution for multi-step prediction.

These findings provided the foundation for the final implementation phase, which would integrate the multiple-output multi-step prediction approach in BunDLe-Net to evaluate this new extended architecture.

3.4. Step 3: Integration in BunDLe-Net

Having validated the presence of multi-step predictive information in the neuronal dataset (Step 1) and confirmed that this information can be effectively preserved through dimensionality reduction (Step 2), the final phase of this work focuses on integrating multi-step prediction capabilities into the BunDLe-Net architecture. This step addresses Research Question 3 by exploring suitable architectural modifications required to extend the original single-step forecasting model to multi-step prediction.

The primary objective of this integration phase is to extend the BunDLe-Net architecture to incorporate multi-step forecasting while preserving its core functionality of learning meaningful low-dimensional embeddings that capture behaviorally relevant neural dynamics. Furthermore, the findings from Step 2 showed that the multiple-output approach provides computational efficiency without performance degradation compared to the direct approach. Therefore, this step will focus on the integration, evaluation, and comparison of the multiple-output and the recursive methods within the BunDLe-Net architecture.

3.4.1. Multi-Step Approach 1: Multiple-Output

The fundamental modification involved replacing the single-step predictor in the original BunDLe-Net architecture with a multi-step predictor capable of forecasting multiple future behavioral states simultaneously. By extending the behavior forecasting horizon from the immediate next state to a sequence of future states, the model is forced to learn embeddings that capture longer-term temporal dependencies rather than optimizing solely for immediate transitions. This extension was implemented using the multiple-output approach selected from Step 2, where a single neural network generates predictions for all future time steps in one single forward pass. Ultimately, the extended architecture maintains BunDLe-Net's dual-branch structure.

The implementation can be found in the appendix section A.1.

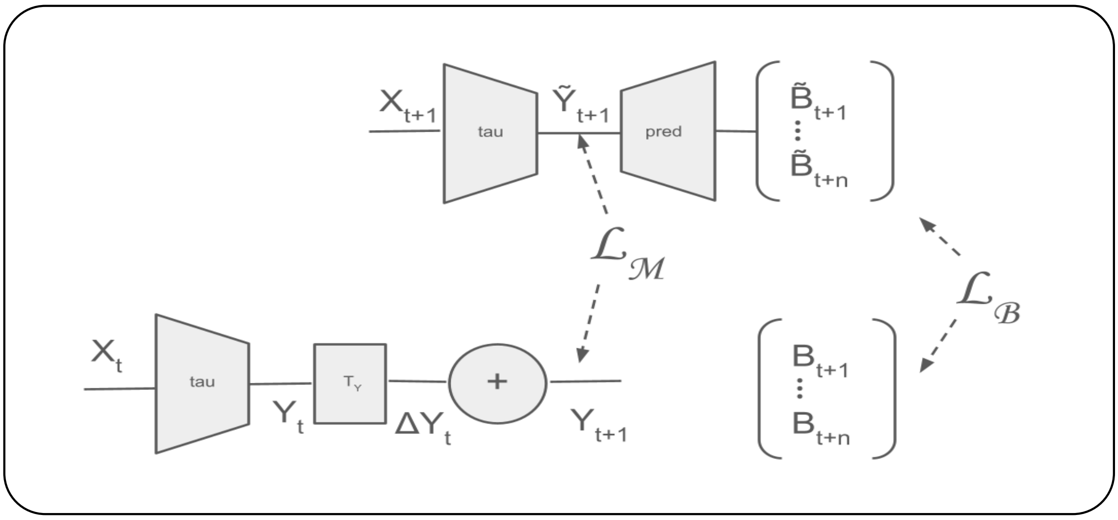


Figure 3.10.: Multiple-Output BunDLe-Net Architecture

Encoder *tau* The encoder has been modified from the original BunDLe-Net implementation to process only a single time step of neuronal activity, rather than a sliding window of previous time steps. Now, this component transforms a single current neuronal state X_t into a latent state Y_t. The encoder maintains the original architecture of a series of ReLU layers, but now includes batch normalization after each layer to improve stability and convergence. Finally, Gaussian white noise is still added in the latent space to reduce overfitting.

Predictor *pred* Another significant architectural modification is implemented in the predictor, which is extended to predict the behaviors for the entire forecast window. Rather than producing a single behavioral prediction $\tilde{B}_{t+1}$, the updated predictor generates a sequence of behavioral predictions $\{\tilde{B}_{t+1}, \tilde{B}_{t+2}, \ldots, \tilde{B}_{t+n}\}$. This is to ensure that the embeddings are relevant and optimized for the entire prediction horizon, not just for the immediate next state. Additionally, to handle the increased complexity of multi-step behavioral prediction, the predictor component has been updated with a deeper architecture. For this reason, the single dense layer used in the original BunDLe-Net has now been replaced with two layers, using the ReLU activation function.

Transition T_y The transition model remains unchanged from the original BunDLe-Net implementation. It continues to predict the future latent state change ΔY_t.

3.4.2. Multi-Step Approach 2: Recursive

The recursive approach represents an alternative strategy for implementing multi-step prediction within the BunDLe-Net architecture. Unlike the multiple-output approach that generates all future predictions simultaneously, the recursive method iteratively applies the model to forecast future states step by step. This approach leverages the single-step prediction capability of the original BunDLe-Net architecture while extending its temporal horizon through sequential application of the prediction operation.

The fundamental modification involved extending the forecasting mechanism to iteratively predict future using the transition model and behavioral predictor in sequence. At each iteration, the model uses the transition model to predict the next latent state and applies the behavioral predictor to generate the behavioral prediction. Ultimately, like the multiple-output approach, even this recursive implementation maintains BunDLe-Net's dual-branch structure.

The implementation can be found in the appendix section A.2.

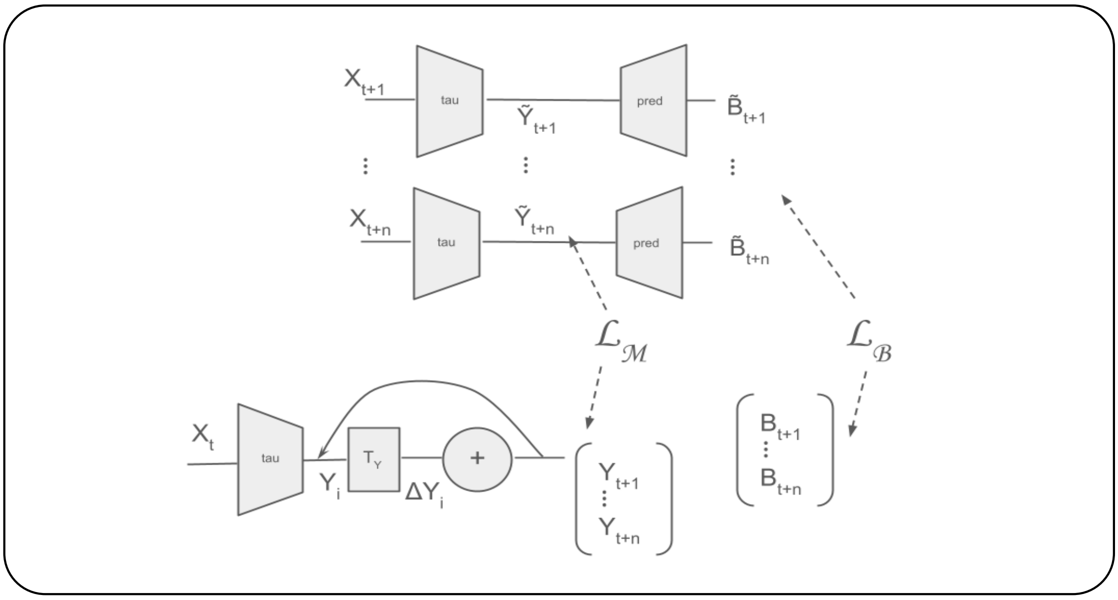


Figure 3.11.: Recursive BunDLe-Net Architecture

Encoder *tau* The encoder used in this approach uses the same modification implemented in the multiple-output approach. Therefore, it processes only a single time step of neuronal activity X_t, encoding it into the latent state Y_t. It uses a sequence of ReLU layers, while including batch normalization after each layer. Gaussian white noise is still added in the latent space to reduce overfitting.

Predictor *pred* In this approach, the predictor uses the same architecture as in the original BunDLe-Net, a single dense layer. So, it continues to produce a single behavioral prediction B_t from a latent state Y_t. However, in this case, the predictor is applied iteratively at each step of the forecasting sequence. At each iteration, after the transition model generates the latent state, the predictor takes this latent state as input and generates the behavioral prediction for the corresponding time step. Through this iterative application, the predictor generates the complete sequence of behavioral predictions.

Transition T_y In this approach, the transition also uses the same architecture as in the original BunDLe-Net, a single dense layer. However, now, like the predictor, it is used iteratively. Starting from a latent state, the transition model is repeatedly applied to predict successive latent state changes. At each iteration, the transition model takes the latent state from the previous step and predicts the latent state difference in order to estimate the next latent state. As a result, this iterative application of the transition model creates the sequence of future latent states.

3.4.3. Evaluation

This section outlines the evaluation framework to assess the performance of the multi-step extended BunDLe-Net, focusing on its ability to generate more meaningful embeddings that are able to capture long-term neural dynamics. The primary goal is to determine whether the embeddings learned by the new model provide a quantitatively superior and qualitatively more interpretable representation of neural dynamics compared to the original single-step approach. The evaluation combines objective performance metrics with visual, qualitative analysis to provide a comprehensive view of the performance of the new extended model.

Baseline for Comparison The performance of the new extended multi-step BunDLe-Net is benchmarked directly against the original, single-step BunDLe-Net architecture. An additional version of the original model trained without a past window of neural activity is also used as a secondary baseline. This ablation variant is to isolate the contribution of the past window.

Embeddings Interpretability The interpretability assessment of the learned embeddings evaluates whether the multi-step approaches generate more interpretable and structured representations compared to the original single-step methods. Three-dimensional plots of the learned embeddings were generated for all model configurations: the two multi-step approaches (multiple-output and recursive) and the two original baselines (with and without past window). Specifically, the visual analysis focused on assessing the clustering structure and separability of hit versus miss responses within the three-dimensional embedding space, as well as the overall organization of neural trajectories and the presence of meaningful structures that reflect the underlying behavioral patterns. Well-performing models should demonstrate clear spatial separation between the two behavioral classes, with hit responses clustering in distinct regions of the latent space compared to miss responses, while also displaying meaningful trajectory structures that capture the temporal evolution of neural states. This qualitative evaluation assesses whether the multi-step prediction objectives successfully encourage the learning of embeddings that naturally separate and model behaviorally relevant motifs.

Embeddings Consistency To assess whether the learned embeddings are consistent across individuals, all nine worms were projected into a single, shared latent space. The shared space is defined by fixing the transition model T_y and the behavior prediction head *pred* from a single reference animal (worm 6), while learning a worm-specific encoder *tau* for each remaining animal. This design allows to align neuronal recordings that contain different neuron subsets by mapping them through their own encoder into the common space defined by the reference transition model and behavioral predictor. Then, all the generated three-dimensional embeddings are visually compared, and consistency between them is judged qualitatively: comparable manifolds across worms and similarly shaped trajectories.

Quantitative Analysis Given the specific experimental design of the dataset with oxygen stimulus manipulation, a primary evaluation metric is the ability of the model to distinguish between "hit" and "miss" responses to the stimulus. This binary classification task serves as a direct test of whether the learned embeddings are able to capture the neural dynamics that drive behavioral responses to environmental stimuli. To quantitatively assess this capability, Linear Discriminant Analysis (LDA) was trained on the low-dimensional embeddings generated by the models. The LDA classifier was trained to predict whether a given embedding corresponds to a hit or a miss response, in order to evaluate whether the learned three-dimensional representations can capture the neural dynamics that drive behavioral responses to environmental stimuli. Its performance was evaluated using 5-fold cross-validation. Statistical significance was established with permutation tests, in which labels were randomly permuted to obtain a chance-level accuracy distribution.

Training Details Following the original BunDLe-Net, the training uses the Adam optimizer with a learning rate of 0.001 and a batch size of 100. The loss weighting parameter γ was maintained at 0.9 to balance the two losses $\mathcal{L}_{Markov}$ and $\mathcal{L}_{Behavior}$. Based on the findings from Step 1 and Step 2, the multi-step prediction horizon was set to 25 steps.

4. Results

This chapter presents the findings from the integration and evaluation of multi-step prediction in the BunDLe-Net framework. The results are organized into several sections: two qualitative analysis to evaluate the interpretability and the consistency of the learned embeddings and a quantitative analysis assessing the models' ability to distinguish between "hit" and "miss" behavioral responses.

4.1. Embeddings Interpretability

This qualitative analysis evaluates the structure and interpretability of the three-dimensional embedding space learned by each model. Each point corresponds to a compressed neural state at a specific time in the recordings, and the trajectories represent the temporal evolution of those states. The evaluation examines how well these embeddings preserve the underlying dynamical structure and behaviorally relevant information while reducing high-dimensional neural activity to an interpretable three-dimensional representation.

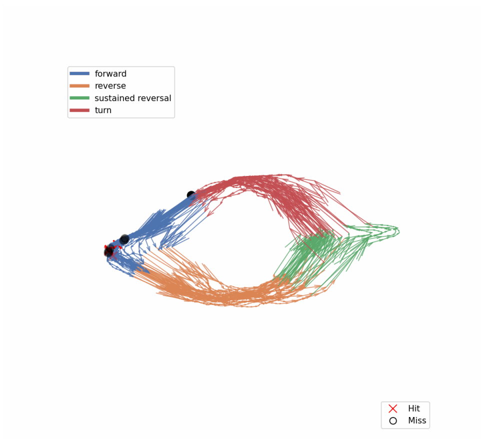


Figure 4.1.: Original, with past window

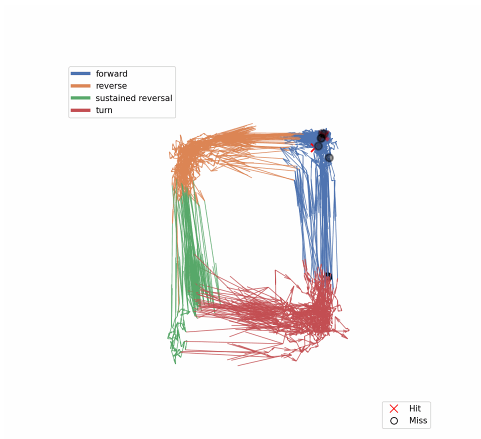


Figure 4.2.: Original, without past window

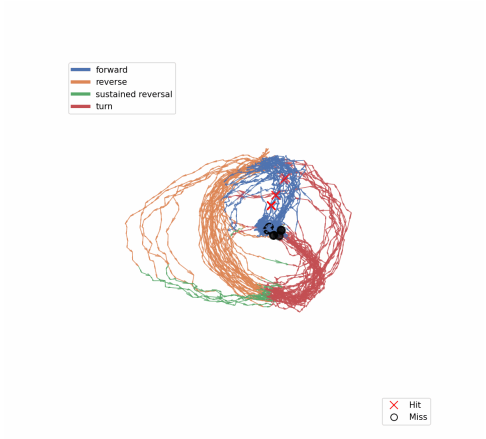


Figure 4.3.: Multi-Step, Multiple-Output

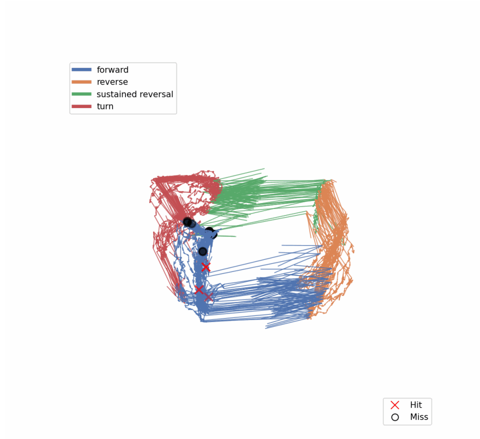


Figure 4.4.: Multi-Step, Recursive

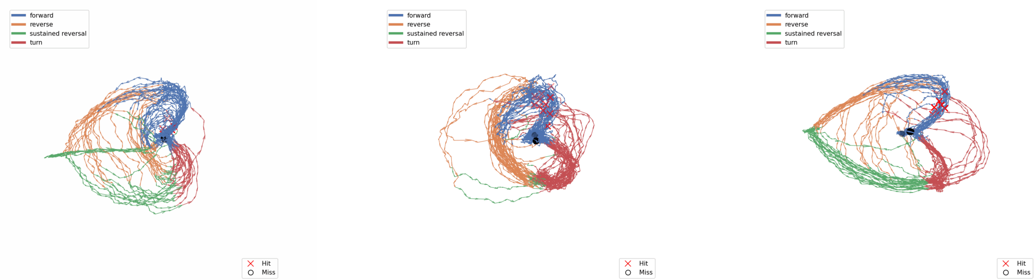
The analysis of these three-dimensional embeddings shows clear differences in how multi-step versus single-step prediction models organized the learned representations in relation to the stimulus response. Specifically, the spatial arrangement of hit and miss responses within the embedding space. For the multi-step approaches, in both the multiple-output and the recursive implementations, hit and miss responses to oxygen stimulation occupy distinct, compact regions that remain in roughly the same portion of the manifold. Hit responses tended to cluster within one compact region of the three-dimensional embedding, whereas miss responses occupied another distinct region. In contrast, the original BunDLe-Net models did not show a clear separation between hit and miss responses. The resulting embeddings show poor separability between the two behavioral classes, with hits and misses overlapping in the same region of the latent space. This result suggests that the single-step prediction objective was not sufficient to learn representations that capture the temporal structure necessary to distinguish behavioral outcomes. Consequently, the model appeared to encode neural activity without reliably preserving information relevant for behavioral discrimination.

Beyond behavioral outcome separability, the embeddings visualizations show very significant differences in the topological organization of the neural trajectories generated by the models. The multiple-output approach generated the most sophisticated topological structure, displaying a complex branching topology. The embedding clearly reveals two distinct turn branches and two distinct reverse branches. This suggests that the model appears to have learned behaviorally relevant sub-patterns within these broader behavioral categories, which extend beyond the four initially established behaviors of forward, reverse, turn, and sustained reversal. This branching structure indicates that the multiple-output model learned to distinguish between different types of turns and different types of reversals, potentially capturing subtleties in the neural dynamics that drive variations within the same behavioral class. On the other hand, despite achieving behavioral separation between hits and misses, the recursive approach produced a simple topological structure consisting primarily of a basic loop. While the model successfully learned to separate behaviorally relevant patterns, it failed to capture the complexity of behavioral motifs that are visible in the multiple-output approach. The original BunDLe-Net models also exhibited a simple loop topology, reinforcing the conclusion that single-step prediction objectives limit the model's capacity to discover complex behavioral motifs. The loop structure suggests that the model learned a basic cyclical pattern but failed to capture the branching dynamics that reflect the true complexity of neural-behavioral relationships.

Additionally, the multiple-output multi-step model reveals a structure that can provide some insights into the underlying neural-behavioral relationship. In particular, the clear separation of hit and miss responses and the more complex topology of the embedding allow for an interpretation of the underlying mechanism of stimulus response. The embedding reveals that hit responses are consistently located in the region of the manifold corresponding to a well-established forward movement, while the misses cluster at the beginning. Therefore, these trajectories suggest that the timing of stimulus application relative to the worm's behavioral state is an important factor for determining the response outcome, suggesting the existence of a behavioral threshold during the forward movement after which the worm is more likely to react to the oxygen stimulus. Specifically, it appears that when the oxygen stimulus is applied while the worm is well into the forward movement, it's more likely to react to stimulus. Conversely, a miss is more likely to occur.

4.2. Embeddings Consistency

This section presents the evaluation of the embeddings consistency across worms only for the multi-output multi-step BunDLe-Net, as it is the only model that was able to learn a rich, branched topology. The embeddings across all worms generated by the other models can be found in the appendix section A.5, section A.6, and section A.7.



Worm 0 Worm 1 Worm 2

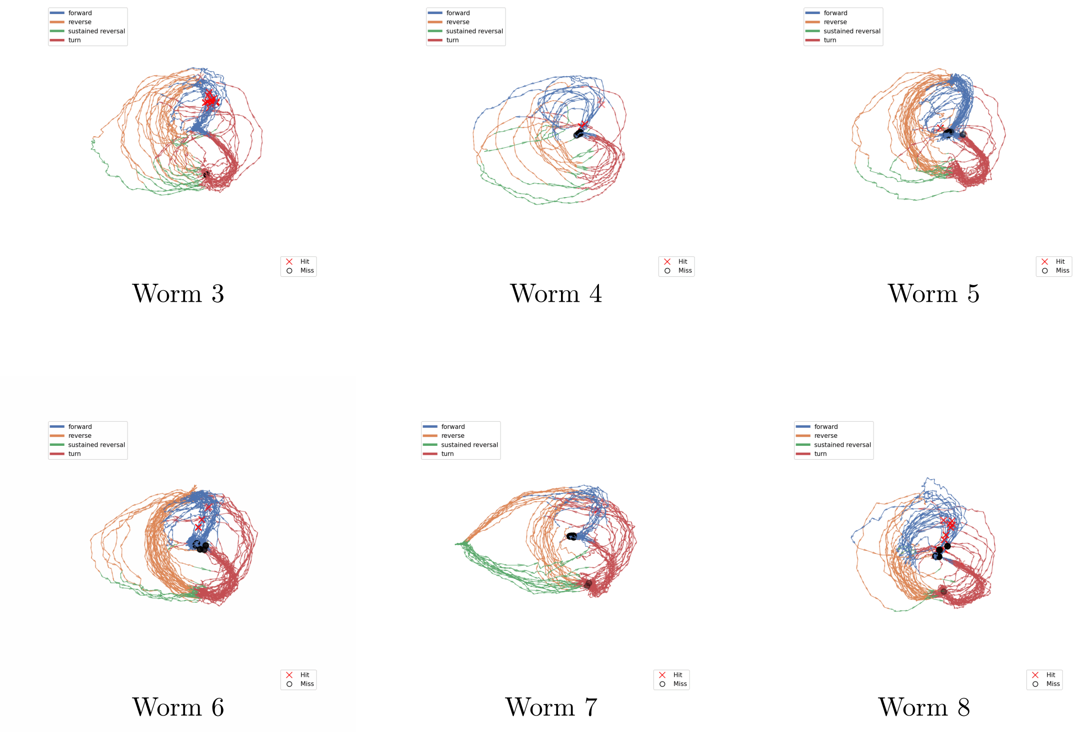


Figure 4.6.: Embeddings of all nine worms (multi-step, multiple-output)

Qualitatively, the embeddings generated by this model reveal a consistent topological structure. The three-dimensional embeddings from all worms show a very similar global manifold: a prominent loop that tracks the behavioral cycle of reverse, turn, and forward. Beyond this loop, the model reveals additional branching structure across all worms. In every animal two additional branches can be observed. A branch corresponding to a long sustained reversal between the reverse and the turn behaviors, and a branch that connects the turn behavior directly with the reverse, with only a very short forward in between. Ultimately, these observations show that the multiple-output multi-step objective yields embeddings that are topologically stable across individuals.

4.3. Quantitative Analysis

This evaluation assesses quantitatively each model's ability to distinguish between hit and miss responses to oxygen stimulation using Linear Discriminant Analysis (LDA) trained on the generated three-dimensional embeddings. This analysis provides an objective measure of how well the different prediction approaches encode behaviorally relevant information in their low-dimensional representations. The following figures show confusion matrices for the LDA classification results, displaying the true versus predicted behavioral response classes (hit or miss) for each model. They provide a clear visualization of classification accuracy, showing how effectively each model's three-dimensional embeddings can distinguish between the two behavioral outcomes.

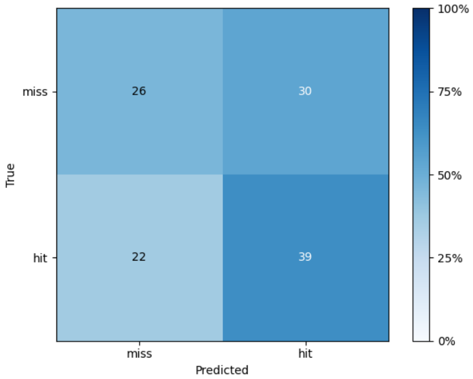


Figure 4.7.: Original, with past window

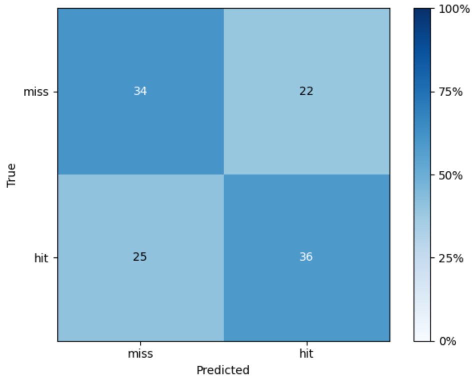


Figure 4.8.: Original, without past window

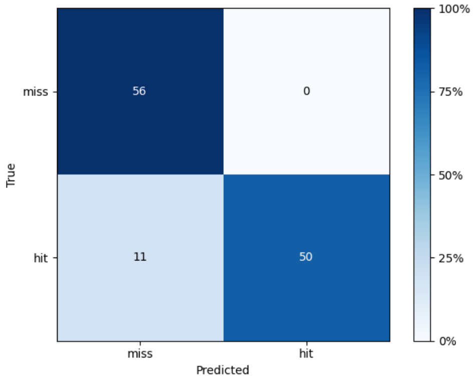


Figure 4.9.: Multi-Step, Multiple-Output

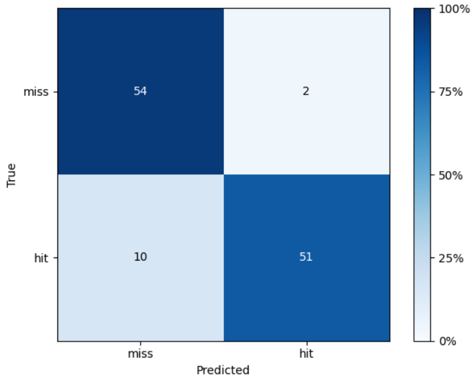


Figure 4.10.: Multi-Step, Recursive

	Original, with past window	Original, without past window	Multi-Step, Multiple-Output	Multi-Step, Recursive
Accuracy	55.6%	59.8%	90.6%	89.7%
Chance	50.9% ± 5.1%	50.3% ± 5.8%	50.9% ± 5.0%	51.0% ± 5.3%

The quantitative results strongly support the qualitative observations regarding behavioral separability. Both multi-step approaches demonstrated substantially better performance in distinguishing hit and miss responses. These accuracies show well above chance performance, indicating that the multi-step prediction objectives successfully encourage the learning of embeddings that preserve behaviorally meaningful information. On the other hand, the original BunDLe-Net, both with and without past window, performed close to chance levels. These results provide clear quantitative validation for the multi-step approach, confirming that extending the prediction horizon is essential for learning embeddings that can distinguish between distinct behavioral outcomes. The far superior performance of the multi-step models highlights the limitations of single-step approaches in capturing longer-term, behaviorally relevant neural dynamics.

5. Discussion

This thesis investigated whether the integration of multi-step prediction into neuronal representation learning frameworks can improve the modeling of long-term neural dynamics. The study addressed this question through a multi-phase experimental approach, ultimately demonstrating that extending BunDLe-Net from single-step to multi-step prediction significantly improves the interpretability and behavioral relevance of learned neural embeddings. However, the effectiveness of multi-step prediction depends heavily on the specific architectural approach. This investigation revealed that not all multi-step strategies are equally effective; the multiple-output approach was found to be the only method capable of learning a rich, structured, and behaviorally discriminative latent space, while the direct approach was less accurate and more computationally demanding, and the recursive approach failed to capture complex dynamics.

In particular, this investigation, structured across three progressive phases, identified several key findings that address the research questions and collectively demonstrate the value of multi-step prediction. For Research Question 1, the data examination phase confirmed neuronal time-series recordings contain predictive information across extended horizons. Feed-forward models predicted future behavioral states with above-chance accuracy over horizons of at least twenty-five steps, indicating feasibility for multi-step forecasting. Second, the compression feasibility analysis to address Research Questions 2 demonstrated that these long-term temporal dependencies can be effectively preserved even when the data is compressed into a low-dimensional bottleneck. This finding was essential for justifying the integration with BunDLe-Net. The success of the bottleneck architectures confirmed the predictive signal is not lost in compression but can be encoded into compact, informative embeddings. Additionally, the direct strategy underperformed compared to multiple-output here, so it was ruled out early. Addressing Research Question 3, integration into BunDLe-Net involved modifying the predictor *pred* for the multiple-output approach, iteratively applying the predictor *pred* and transition model T_Y for the recursive implementation, and modifying the encoder *tau* for both approaches. Finally, Research Question 4 revealed qualitative and quantitative differences, identifying the multiple-output model as the only method that overcame the limitations of single-step approaches to yield richer, more interpretable representations of neural dynamics.

The multiple-output multi-step BunDLe-Net not only achieved a clear separation of hit and miss behavioral responses but, more importantly, was the only approach to uncover more sophisticated and meaningful topological structure in the neural data. The resulting embeddings displayed a complex branching topology, revealing distinct sub-patterns. This complex structure, was also found to be topologically consistent across all nine worms, validating the generalizability of the multi-step approach. This qualitative success was also confirmed quantitatively. The embeddings from the multiple-output model enabled an LDA classifier to distinguish between hit and miss outcomes with very high accuracy, performing substantially above chance levels. In contrast, the direct approach was eliminated early in the experimental process due to consistently inferior performance compared to the multiple-output method across both the initial data examination and compression feasibility phases. The recursive approach, while achieving some behavioral separability, failed to learn the sophisticated topological structures discovered by the multiple-output method.

Beyond these empirical findings, this study can offer contributions across multiple scientific domains. In the field of machine learning, this work provides insights into the architectural designs for dimensionality reduction and manifold learning. The findings suggest that simultaneous prediction of entire future sequences can allow models to encode more causally relevant dynamics, generate more interpretable latent representations, and discover more complex topological structures compared to single-step approaches that optimize only for the single next state. From a neuroscience perspective, the extended BunDLe-Net functions as a tool for both visualization and partial automation of insights into neural-behavioral processes. The findings of this work show that the learned low-dimensional trajectories revealed behaviorally meaningful structures that were not captured by single-step approaches. The ability to uncover these richer topologies and distinct sub-patterns in embeddings reduces reliance on manual analysis, allowing researchers to identify behaviorally relevant motifs more efficiently and better understand the relationship between neuronal dynamics and observable behaviors. More broadly, the principles demonstrated in this study extend to other time-series domains where dimensionality reduction and representation learning are used to improve interpretability and understand causal relationships between high-dimensional variables. Many fields face similar challenges of extracting meaningful patterns from complex, noisy temporal data while preserving interpretability. By applying multi-step prediction to compress these datasets while preserving predictive information, this framework can help identify causal links between variables.

Despite the ability of the multiple-output multi-step approach to learn richer, more interpretable representations compared to single-step approaches, there are limitations. A primary limitation is the fixed time horizon, as the model is currently constrained to predict a specific, pre-determined number of future steps. Although this number has been empirically validated for this dataset, it is not going to be optimal across different datasets and experiments. Additionally, the research also found that not all multi-step approaches performed well. Recursive models, in particular, can be prone to error propagation, which diminishes their effectiveness over longer horizons. To address these limitations and further advance the research, several developments could be explored. Among these, developing a method that allows the model to automatically determine the optimal number of future steps from the data could create embeddings that are better tuned to the intrinsic patterns of the neural dynamics being studied. Alternative loss functions could also be investigated, such as contrastive loss, which could potentially encourage the model to learn representations where behaviorally similar states are close in the embedding space while behaviorally distinct states are pushed apart.

6. Conclusion

This thesis investigated the implementation of multi-step prediction techniques to improve representation learning of noisy, high-dimensional neuronal time-series data. It successfully extended BunDLe-Net from single-step to multi-step forecasting to capture longer-term temporal dynamics and behavioral motifs, showing significant improvements in both interpretability and behavioral relevance of learned neural embeddings. By forcing the model to anticipate a sequence of future states, it is able to learn low-dimensional embeddings that are better at separating behaviorally distinct outcomes and revealing a rich topological structure of neural dynamics. The multiple-output approach, which predicts all future states simultaneously in a single forward pass, produced interpretable embeddings that clearly separated distinct behavioral responses and revealed a complex, branching topology in neural dynamics, substantially outperforming the single-step baselines. These findings shows that by extending temporal horizons beyond immediate state transitions, multi-step approaches can reveal complex structures underlying neural-behavioral relationships that remain hidden to single-step methods.

Bibliography

Misha B. Ahrens, Jennifer M. Li, Michael B. Orger, Drew N. Robson, Alexander F. Schier, Florian Engert, and Ruben Portugues. Brain-wide neuronal dynamics during motor adaptation in zebrafish. *Nature*, 7399(485):471–477, 2012.

Misha B Ahrens, Michael B Orger, Drew N Robson, Jennifer M Li, and Philipp J Keller. Whole-brain functional imaging at cellular resolution using light-sheet microscopy. *Nature Methods*, 10(5):413–420, 2013.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35:1798–1828, 08 2013.

Gianluca Bontempi. Long term time series prediction with multi-input multi-output local learning. *Proceedings of the 2nd European Symposium on Time Series Prediction (TSP), ESTSP08*, page 145–154, 01 2008.

Gianluca Bontempi and Souhaib Ben Taieb. Conditionally dependent strategies for multiple-step-ahead prediction in local learning. *International Journal of Forecasting*, 27:689–699, 07 2011.

Haibin Cheng, Pang-Ning Tan, Jing Gao, and Jerry Scripps. Multistep-ahead time series prediction. *Advances in Knowledge Discovery and Data Mining*, 3918:765–774, 04 2006.

John P Cunningham and Byron M Yu. Dimensionality reduction for large-scale neural recordings. *Nature Neuroscience*, 17:1500–1509, 2014.

Juan A Gallego, Matthew G Perich, Lee E Miller, and Sara A Solla. Neural manifolds for the control of movement. *Neuron*, 94(5):978–984, 2017.

Coşkun Hamzaçebi, Diyar Akay, and Fevzi Kutay. Comparison of direct and iterative artificial neural network forecast approaches in multi-periodic time series forecasting. *Expert Systems with Applications*, 36:3839–3844, 03 2009.

Geoffrey E. Hinton and Ruslan R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 5786(313):504–507, 2006.

Saul Kato, Harris S Kaplan, Tina Schrödel, Susanne Skora, Theodore H Lindsay, Eviatar Yemini, Shawn Lockery, and Manuel Zimmer. Global brain dynamics embed the motor command sequence of caenorhabditis elegans. *Cell*, 163(3):656–669, 2015.

Douglas Kline. *Methods for Multi-Step Time Series Forecasting with Neural Networks*, pages 226–250. 01 2004.

Akshey Kumar, Aditya Gilra, Mauricio Gonzalez-Soto, Anja Meunier, and Moritz Grosse-Wentrup. Bundle-net: Neuronal manifold learning meets behaviour. *bioRxiv*, 2023. doi: 10.1101/2023.08.08.551978.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. 02 2018. doi: 10.48550/arXiv.1802.03426.

Marina Meilă and Hanyu Zhang. Manifold learning: What, how, and why. *Annual Review of Statistics and Its Application*, 11:393–417, 2023.

Aaron Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 07 2018.

Chethan Pandarinath, Daniel J. O'Shea, Jasmine Collins, Rafal Jozefowicz, Sergey D. Stavisky, Jonathan C. Kao, Eric M. Trautmann, Matthew T. Kaufman, Stephen I. Ryu, Leigh R. Hochberg, Jaimie M. Henderson, Krishna V. Shenoy, L. F. Abbott, and David Sussillo. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature Methods*, (15):805–815, 2018.

Liam M. Paninski and John P Cunningham. Neural data science: accelerating the experiment-analysis-theory cycle in large-scale neuroscience. *Current Opinion in Neurobiology*, 50:232–241, 2018.

Robert Prevedel, Young-Gyu Yoon, Maximilian Hoffmann, Nikita Pak, Gordon Wetzstein, Saul Kato, Tina Schrödel, Ramesh Raskar, Manuel Zimmer, Edward S Boyden, and Alipasha Vaziri. Simultaneous whole-animal 3d imaging of neuronal activity using light-field microscopy. *Nature Methods*, 11:727–730, 2014.

Shreya Saxena and John P Cunningham. Towards the neural population doctrine. *Current Opinion in Neurobiology*, 55:103–111, 2019.

Tina Schrödel, Robert Prevedel, Karin Aumayr, Manuel Zimmer, and Alipasha Vaziri. Brain-wide 3d imaging of neuronal activity in caenorhabditis elegans with sculpted light. *Nature Methods*, 10:1013–1020, 2013.

Ian H Stevenson and Konrad P Kording. How advances in neural recording affect data analysis. *Nature Neuroscience*, 14(2):139–142, 2011.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

Saurabh Vyas, Matthew D. Golub, David Sussillo, and Krishna V. Shenoy. Computation through neural population dynamics. *Annual Review of Neuroscience*, 43:249–275, 2020.

Guoqiang Zhong, Li-Na Wang, and Junyu Dong. An overview on data representation learning: From traditional feature learning to recent deep learning. *The Journal of Finance and Data Science*, 2:265–278, 2016.

A. Appendix

A.1. Multiple-Output BunDLe-Net

```
class BunDLeNet_MultipleOutput(nn.Module):
    def __init__(self, input_size, latent_size, output_size, n_steps):
        super(BunDLeNet_MultipleOutput, self).__init__()
        self.n_steps = n_steps
        self.output_size = output_size

        self.tau = nn.Sequential(
            nn.Linear(input_size, 50),
            nn.BatchNorm1d(50),
            nn.ReLU(),
            nn.Linear(50, 40),
            nn.BatchNorm1d(40),
            nn.ReLU(),
            nn.Linear(40, 30),
            nn.BatchNorm1d(30),
            nn.ReLU(),
            nn.Linear(30, 20),
            nn.BatchNorm1d(20),
            nn.ReLU(),
            nn.Linear(20, 10),
            nn.BatchNorm1d(10),
            nn.ReLU(),
            nn.Linear(10, latent_size),
            GaussianNoise()
        )
        self.predictor = nn.Sequential(
            nn.Linear(latent_size, 50),
            nn.ReLU(),
            nn.Linear(50, n_steps * output_size)
        )
        self.T_y = nn.Sequential(
            nn.Linear(latent_size, latent_size)
        )

    def forward(self, x):
        xt0, xt1 = x

        yt0 = self.tau(xt0)
        yt1_l = yt0 + self.T_y(yt0)

        yt1_u = self.tau(xt1)
        bt1 = self.predictor(yt1)

        return yt1_l, yt1_u, bt1.view(-1, self.n_steps, self.output_size)
```

Listing A.1: Multiple-Output BunDLe-Net

A.2. Recursive BunDLe-Net

```python
class BunDLeNet_Recursive(nn.Module):
    def __init__(self, input_size, latent_size, output_size, n_steps):
        super(BunDLeNet_Recursive, self).__init__()
        self.n_steps = n_steps
        self.output_size = output_size

        self.tau = nn.Sequential(
            nn.Linear(input_size, 50),
            nn.BatchNorm1d(50),
            nn.ReLU(),
            nn.Linear(50, 40),
            nn.BatchNorm1d(40),
            nn.ReLU(),
            nn.Linear(40, 30),
            nn.BatchNorm1d(30),
            nn.ReLU(),
            nn.Linear(30, 20),
            nn.BatchNorm1d(20),
            nn.ReLU(),
            nn.Linear(20, 10),
            nn.BatchNorm1d(10),
            nn.ReLU(),
            nn.Linear(10, latent_size),
            GaussianNoise()
        )
        self.predictor = nn.Sequential(
            nn.Linear(latent_size, output_size),
        )
        self.T_y = nn.Sequential(
            nn.Linear(latent_size, latent_size)
        )

    def forward(self, x):
        xt0, xt1 = x

        yt1_l = self.tau(xt0)
        yt1_ls, yt1_us, behavs = [], [], []
        for i in range(self.n_steps):
            yt1_l = yt1_l + self.T_y(yt1_l)
            yt1_ls.append(yt1_l)

            yt1 = self.tau(xt1[:, i])
            bt1 = self.predictor(yt1)
            yt1_us.append(yt1)
            behavs.append(bt1)

        behavs = torch.stack(behavs, dim=1)
        yt1_us = torch.stack(yt1_us, dim=1)
        yt1_ls = torch.stack(yt1_ls, dim=1)

        return yt1_ls, yt1_us, behavs
```

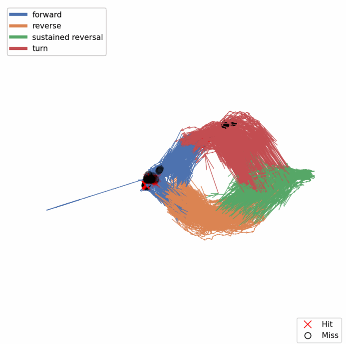
Listing A.2: Recursive BunDLe-Net

A.3. BunDLe-Net Loss Function

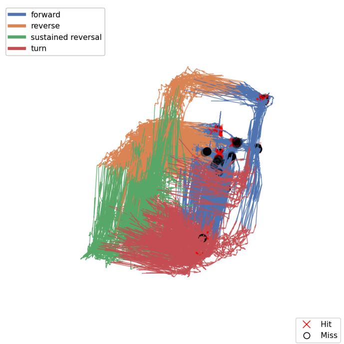
```python
def loss(yt1_ls, yt1_us, behavs, labels, gamma):
    mse = nn.MSELoss()
    cross_entropy = nn.CrossEntropyLoss()

    behav_loss = cross_entropy(behavs, labels.long())
    neural_loss = mse(yt1_ls, yt1_us)
    loss = (1 - gamma) * behav_loss + gamma * neural_loss

    tot_loss += loss.item()
    weighted_neural_loss += (gamma * neural_loss).item()
    weighted_behav_loss += ((1 - gamma) * behav_loss).item()

    return weighted_neural_loss, weighted_behav_loss, tot_loss
```

Listing A.3: BunDLe-Net Loss Function

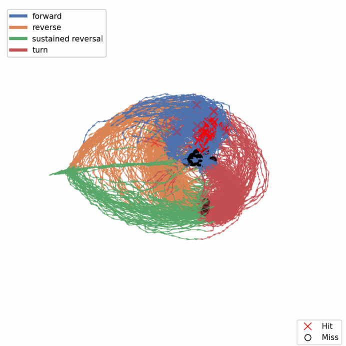
A.4. Cumulative embeddings of all worms



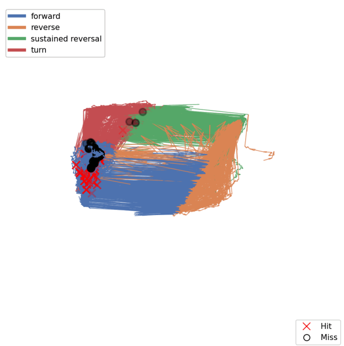
Original, with past window



Original, without past window



Multi-Step, Multiple-Output



Multi-Step, Recursive

Figure A.2.: Cumulative embeddings of all worms

A.5. Embeddings across all worms - original, past window

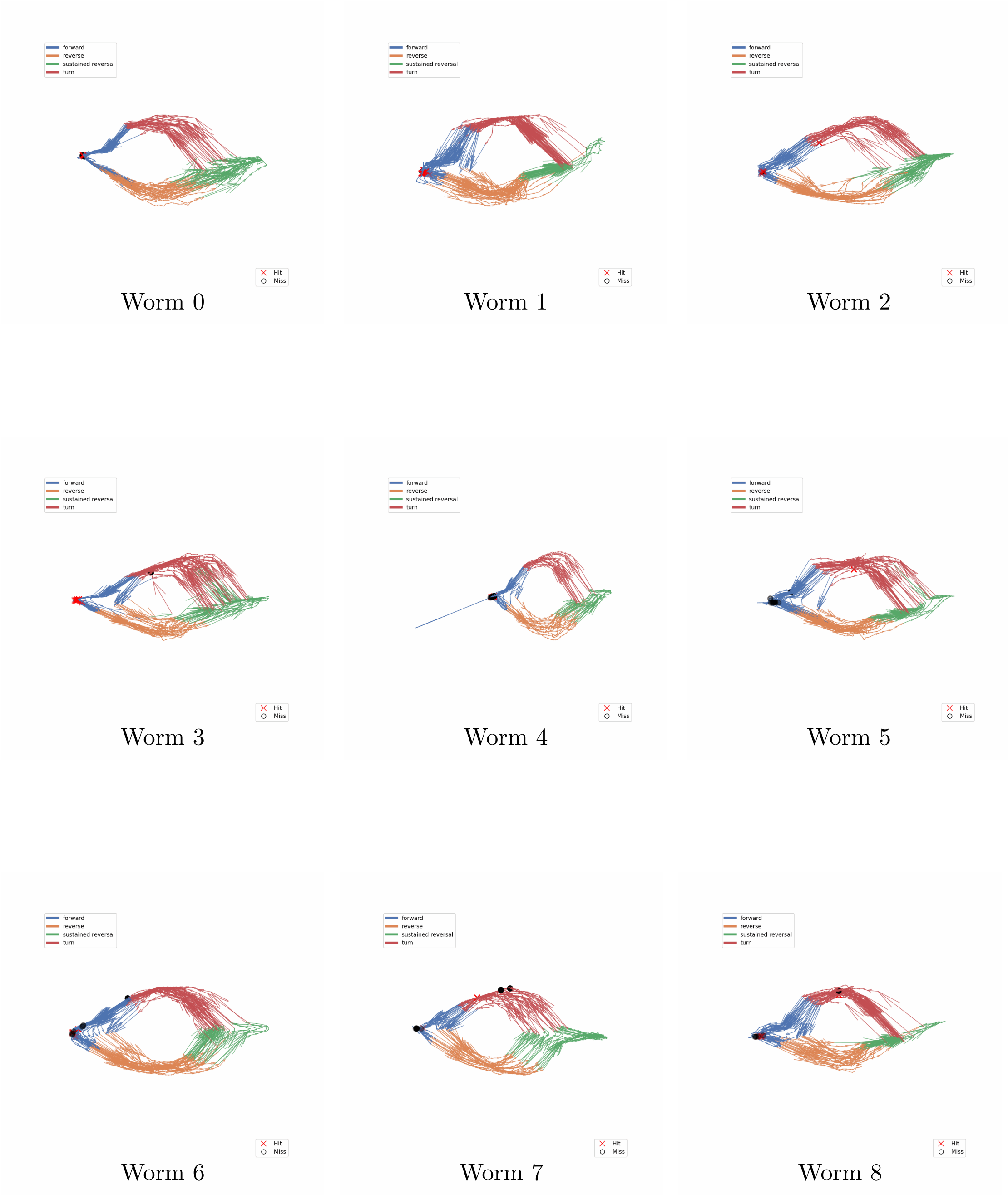


Figure A.4.: Embeddings of all nine worms (original, with past window)

A.6. Embeddings across all worms - original, no past window

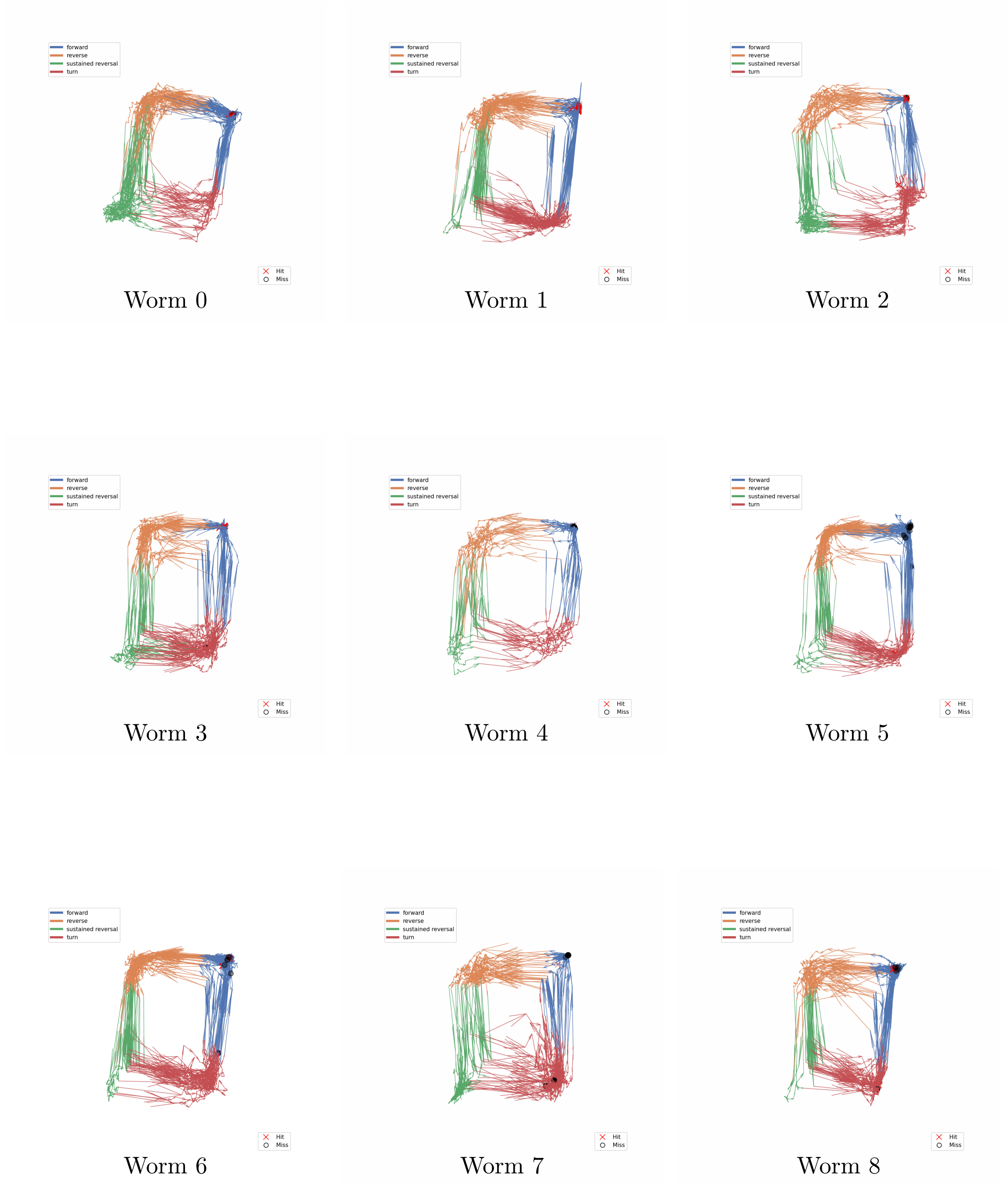
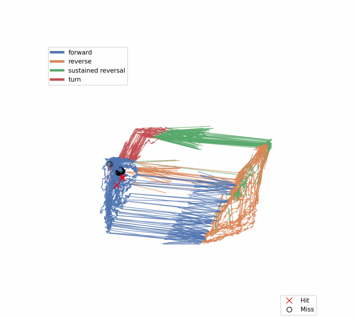
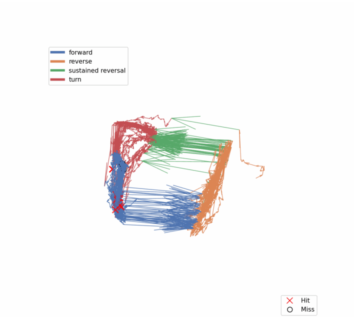


Figure A.6.: Embeddings of all nine worms (original, without past window)

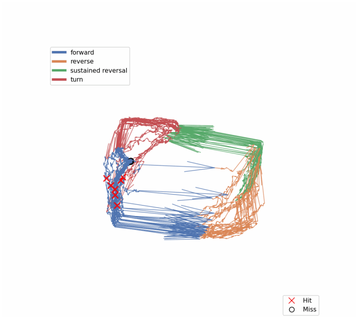
A.7. Embeddings across all worms - multi-step, recursive



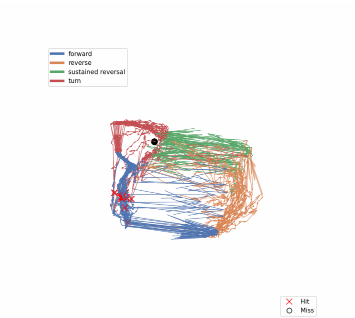
Worm 0



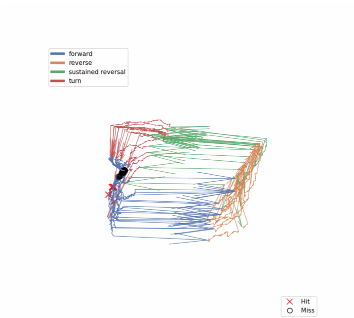
Worm 1



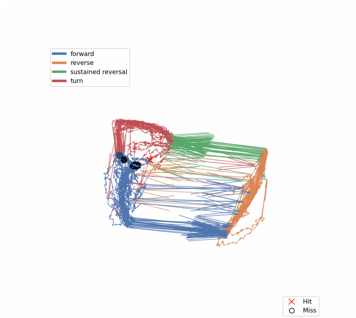
Worm 2



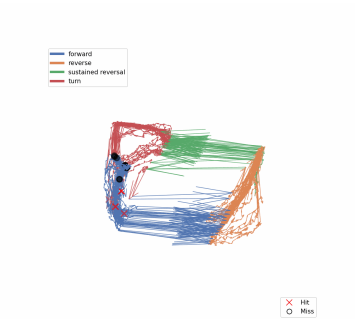
Worm 3



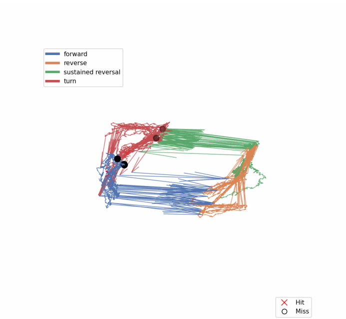
Worm 4



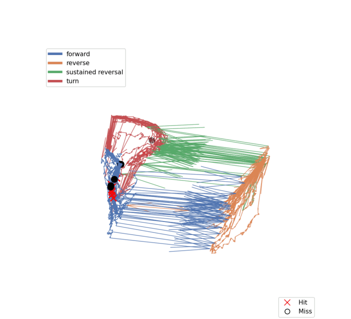
Worm 5



Worm 6



Worm 7



Worm 8

Figure A.8.: Embeddings of all nine worms (multi-step, recursive)