Robust Representation Learning Approaches for Neural Population Activity

Justin Jude

Doctor of Philosophy
Institute for Adaptive and Neural Computation
School of Informatics
University of Edinburgh
2023

Abstract

Understanding communication patterns between different regions of the human brain is key to learning useful spatial representations. Once learned, these representations present a foundation on which new tasks can be learned rapidly. Moreover, the activity patterns generated by the brain are ultimately relayed to the muscles to generate behaviour. By measuring these action potentials from the relevant source regions of the brain directly, we can capture expected behaviour notwithstanding interruption in the neural pathways to downstream muscles. Spinal cord injury is an example of interruption in the case of motor control of arm or leg muscles from the motor cortex of the brain. Multiple electrodes recording action potentials from neurons in the motor cortex in conjunction with a plethora of possible modelling techniques can be used to decode this intended movement. Subsequently, soft or hard robotics can be used to bypass the damaged spinal cord in relaying intended movement behaviour to specific limbs.

This thesis is comprised of two main parts. The first part addresses the question of how representation learning in neural networks can benefit the learning of goal-directed behaviour. Using the learning of spatial representations through recurrent neural networks as a model, this work showed that such a representation can be used as a foundation for rapid learning of navigational tasks using reinforcement learning. This learned representation takes the form of spatially modulated units within the neural network, similar to place cells found in the brains of mammals. Furthermore, an analysis of the simulated neurons showed that these place units within the neural network have multiple characteristics replicating those found in biological place cells, such as precursory firing behaviour.

The second part tackles the issue of variability in neural representations, a phenomenon that causes significant deterioration of the decoding of behaviour from neural population activity over time. Using combined neural and behaviour recordings from monkeys performing motor tasks, this work aims to develop stable decoders that are robust to such fluctuations. Two approaches using unsupervised learning were investigated. The first is based on domain adaptation, where decoders were trained to ”ignore” all aspects of the data subject to fluctuations, and to instead extract the salient, stable aspects of the neural representation of movements. This representation then allows the decoder to generalise well to a completely unseen recording session, thus accurately predicting behaviour intention withstanding significant neuron non-stationaries present between recording sessions. This generalisation to an unseen recording session without retraining or recalibration of a decoder has not been previously shown.

This first approach performed well for data that was obtained close enough in time to the training data, but required a significant number of recording sessions for successful training. To address these limitations, a contrastive learning approach was used next. In this model, synthetic variations of trials from a single recording session were generated. These variations were similar in type and magnitude to the neuron non-stationaries that exist between recording sessions, and used as training data together with the original data for a model that learns to remove these non-stationaries to recover stable dynamics related to behaviour. This method produced a very stable decoder capable of accurately inferring intended behaviour for up to a week into the future. This training paradigm is an example of self-supervised learning, whereby the model is trained on perturbed versions of data.

Taken together, in this thesis I explore approaches which lead to robust representations being learned within neural networks. These representations are shown to be neurally realistic and robust, allowing for a high degree of generalisation.

Lay Summary

Around 80 billion neurons in the human brain communicate continuously to allow us to perform a plethora of bodily functions. In order to perform these functions, large populations of neurons from various brain regions interact within these populations in order to trigger downstream functions. These range from neurons in the motor cortex interacting to perform the movement of limbs, to neurons in the hippocampus interacting to perform spatial navigation. Spatial navigation, for instance, occurs when individual neurons within a population of neurons in the hippocampus region of the brain are active for distinct regions of an environment. Evolutionarily, this has been converged on as the most energy efficient mechanism by which humans (and other mammals) navigate the world.

Although individual bodily functions are processed through these interactions occurring amongst many thousands of neurons, the functions themselves can be fairly simple. For example, the movement of one's arm up versus down is controlled by the same large population of neurons in the motor cortex. However, neuron firing patterns from this population of neurons as a whole will differ wildly for both of these arm actions. Finding an informative, low-dimensional representation of such a population of neurons is then vital to inferring arm movement direction. The expectation is that the low-dimensional representation of neural activity in the motor cortex will be sufficiently unique for an arm movement up versus an arm movement down. This is the starting point for building brain-computer interfaces (BCIs), which can infer muscle movement (and other human behaviour) when even a small subset of these neuron populations are recorded from. With recent advances in BCI effectiveness, individuals with life-changing tetraplegia as a result of neuromuscular disorders have the potential to regain motor functions through the use of BCI controlled prosthetics. The primary issue with current BCI systems is that once trained, their long-term accuracy wanes quickly due to instabilities in neuron activity and minute movement of recording apparatus.

In this thesis, I show that using a learning strategy akin to rodents in experiments in Neuroscience exhibits spatial firing patterns in artificial models similar to that of mammals. I next present two modelling approaches which significantly improve the long-term accuracy of BCI decoders, reducing the frequency with which these decoders are required to be recalibrated to facilitate continued accurate decoding of behaviour.

Acknowledgements

I would like to thank my primary supervisor Dr. Matthias Hennig for his highly supportive supervision style and for helping me with all of my work in my PhD. You've aided me in the discussion of many ideas over the last 4 years and given me sound advice in all areas. I have learned so much from you that I will apply in the rest of my career that I very likely would not have, had I not had your supervision. I have particularly valued your extensive help with motivating all of my ideas within the context of Neuroscience, as well as invaluable guidance with manuscript writing. I would also like to thank my second supervisor Dr. Arno Onken, Dr. Nina Kudryashova, Dr. Cole Hurwitz and all of the other members of the Neurons and Systems journal club for crucial discussions on my work and introducing me to new areas of research.

Most importantly, I would like to thank my entire family for consistently supporting me throughout my Doctorate and throughout my life. I am particularly and continually grateful to my parents - without their immense and unimaginable sacrifice I would not have had the prosperity, opportunities and education that I have been granted.

Lastly, I would like to thank the School of Informatics at The University of Edinburgh as well the Engineering and Physical Sciences Research Council (EPSRC) for their financial support.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

The following previously published work of mine features prominently within this dissertation. Each chapter details the relevant relations to my previous work.

- J. Jude and M. H. Hennig. Hippocampal representations emerge when training recurrent neural networks on a memory dependent maze navigation task. *arXiv preprint arXiv:2012.01328*, 2020

- C. Hurwitz, A. Srivastava, K. Xu, J. Jude, M. Perich, L. Miller, and M. Hennig. Targeted neural dynamical modeling. *Advances in Neural Information Processing Systems*, 34, 2021

- J. Jude, M. Perich, L. Miller, and M. Hennig. Robust alignment of cross-session recordings of neural population activity by behaviour via unsupervised domain adaptation. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 10462–10475. PMLR, 17–23 Jul 2022

- J. Jude, M. G. Perich, L. E. Miller, and M. H. Hennig. Capturing cross-session neural population variability through self-supervised identification of consistent neuron ensembles. In *Proceedings of the NeurIPS Workshop on Symmetry and Geometry in Neural Representations*, volume 197 of *Proceedings of Machine Learning Research*. PMLR, Dec 2022

(*Justin Jude, London, England, 2023*)

Table of Contents

Chapter 1

Introduction

Artificial Neural Networks (ANNs), especially those optimised using backpropagation [48], have emerged as powerful and versatile models, capable of both class prediction and novel data generation. It has been shown that ANNs are neurobiologically analogous models of how the brain processes stimuli [47]. Moreover, when trained to perform and control tasks akin to those performed in the field of experimental Neuroscience, ANNs exhibit features and activation patterns mirroring those found in rodents (but also present in other mammals) performing the same tasks. Recurrent Neural Networks (RNNs) in particular have shown to be a fruitful paradigm for modelling areas of the mammalian brain. This is due to the resemblance between the recurrent nature of RNN training to the way stimuli are processed by the mammalian brain. For example, when performing image classification and object detection tasks, RNNs exhibit similar processing of images as the mammalian visual system [54, 92]. Furthermore, units of RNNs performing spatial tasks in artificial environments display the emergence of place and grid representations found in mammals [14, 5, 71, 13, 87].

Within the mammalian brain, the hippocampus is a formation in the medial temporal lobe. It is crucial for the consolidation and retrieval of memory, especially in the short-term. The hippocampus also serves the important function of spatial localisation and navigation. This was shown in a landmark study in 1971 by John O'Keefe [66, 65] where hippocampal neurons of rats traversing a maze exhibited firing patterns correlated with specific locations in the maze. These neurons were termed "place cells" and spatial locations which they correspond to, "place fields". Further studies into hippocampal function find further properties of these place cells which facilitate more complex spatial reasoning, particularly where rewarding stimuli are concerned. These include place cells drifting towards rewarding locations throughout maze conditioning

[49], a higher concentration of place fields at rewarding locations [58] and preemptive firing of successions of place cells in anticipation of distinct movement directions [39].

In this thesis I hypothesise that ANN units are analogous to these hippocampal neurons when ANNs are trained on similar tasks to those of rodents. This is done by training ANNs using the machine learning paradigm of reinforcement learning in order to guide a simulated agent towards rewarding locations of a virtual maze. Once I have shown this resemblance, and that robust representations akin to those of the hippocampus can be used for downstream tasks, I proceed to show that such representations are also useful for stably modelling neurons which are responsible for expressing animal behaviour.

Animal behaviours (such as movement, speech and visual stimuli) undoubtedly correspond to activity from biological neurons in the brain [85, 17]. In recent years, the number of neurons that can be recorded from simultaneously has increased by several orders of magnitude. This is due to advances in microchip density and transmission rates between extracellular recording electrodes and external decoding devices. The increase in the number of electrodes used in many of these recordings gives far greater insight into behaviour planning and execution.

The predictive potential of decoding models used to infer behaviour intent corresponding to these high dimensional data streams is also significantly improved. Classical linear decoders which can be effective on individual or small numbers of neurons do not have the capacity or complexity to decode from a large number of neurons or channels. Non linear models with large numbers of parameters such as neural networks have been shown to be much more practical in predicting behaviour from these high dimensional recordings [25, 83, 36].

Accurate decoding is possible because high-dimensional neural population activity typically occupies low dimensional manifolds [12, 18, 23, 31, 59, 75]. These manifolds can be extracted using suitable latent variable models. Over time however, drifts in activity of individual neurons and instabilities in neural recording devices can be substantial, causing inconsistency in the extraction of a stable neural manifold. This results in stable decoding over days and weeks being impractical with previous methods.

This variability due to neural drift and other non-stationaries is one of a number of hurdles preventing brain computer interfaces from becoming commonplace. Retraining a decoder over consecutive days would maintain accuracy but with the significant time and computational cost this would entail. While neural drift cannot be predicted on an individual neuron level, population level variations over consecutive recording sessions such as differing sets of neurons and varying permutations of consistent neurons in recorded data may be learnable by a latent variable model based on neural networks when the underlying manifold is stable over time.

In this thesis I will present two approaches which aim to correct for this variability. The first is a model trained using the principle of domain adaptation which aims to unify data by class irrespective of source distribution. The second uses self-supervised learning to decode behaviour from unseen recordings up to a week into the future with no further decoder training. This is achieved by training a model on anticipated future changes to neuron populations and firing patterns.

I first present background on hippocampal function and how it functions as a predictive map of an environment. I then review previous and current models for extracting interpretable latent variables from high dimensional neural data as well as comparisons of current behaviour decoding methods. Next, I introduce three chapters of original work where I show that artificial neural networks exhibit properties shared with their biological counterparts as well as how these neural networks can conversely be used to model biological neuron firing patterns. Finally, I conclude with a discussion of each of my works and my future research plans.

Chapter 2

Background

2.1 The Hippocampus and navigation

The hippocampus is a complex brain formation in the temporal lobe which serves as a memory consolidator and has projections to and from many other brain regions. One of these regions is the entorhinal cortex, itself within the medial temporal lobe. Input to the dentate gyrus region of the hippocampus from the medial entorhinal cortex (MEC) is a crucial pathway with respect to pattern recognition and the encoding of memories. This encoding mechanism of the hippocampus is used to encode memory of various stimuli (such as visual landmarks) at different locations in space. The resulting cognitive map [86] of a given environment takes the form of an ensemble of place cells, which are position-sensitive neurons in the hippocampus which actively produce action potentials when the animal is at a corresponding place field [66], i.e location in an environment. Many place cells can map to a single place field (location) and a single place cell can map to more than one place field. The combination of active firing place cells when the animal is at a given location allows for allocentric self-positioning. In fact, place cells have been shown to have a causal role in the formation of these cognitive maps as the stimulation of individual place cells triggers the positioning behaviour associated with the location of place fields corresponding to said place cells [72], causing the animal to determine it is at a position that it does not actually occupy in space.

In conjunction with grid cells in MEC [30], the animal can navigate to different locations with a different place field from its current location. Grid cells are neurons which fire at multiple locations in an environment in a periodic hexagonal manner and form the coordinate system used by mammals to navigate space. Grid cells essentially act as a mapping from one place field to another, supporting navigation through novel routes and taking into account obstacles as well as other environmental changes.

The layout of place fields is not constant, and can adjust to small changes within known environments through partial remapping [61] and flexibly adjust to entirely new environments through global remapping. This remapping of place fields can correct for obstacles within an environment, changes to stimuli, or a consistent environment rotation [7]. A linear environment transformation is corrected through the stretching or contraction of place fields [64]. These hippocampal neurons have also been observed in the encoding of non-spatial information, showing that hippocampal neurons can be sensitive to distinct frequency ranges of sound [2] This shows that the memory encoding function of hippocampal neurons is general-purpose and can be applied to many tasks within various contexts. Place cells can also be specialised in order to identify the boundaries of a given environment [50].

2.1.1 Task related place cell properties

The activity of place cells has been shown to be task dependent through experimental results. Although from a spatial perspective, place cell firing activity was originally thought to be consistent when the environment remains persistent, results from various maze-based rodent experiments show that place cell firing is more malleable and conductive to task success. For instance, [39] show that place cells in mice in the CA3 region of the hippocampus frequently fire nonlocally, with place cells corresponding to place fields ahead of the mouse firing prior to the mouse approaching critical decision locations in a maze. [27] show that a far higher proportion of hippocampal neurons in the CA1 region in rats performing an episodic task in a T-shaped maze encode the phase of the task rather than spatial information (in Chapter 3 this relates to trajectory direction). [1] show CA1 place cells encode rewarding destination location at the start position of a maze by firing as the rodent begins navigating towards a reward. [49] demonstrate that place fields of CA1 neurons gradually drift toward reward locations throughout reward training on a T-shaped maze - this can be interpreted as a form of continual remapping. [82] identify hippocampal CA1 neurons whose activity are modulated not only by spatial location in a maze but also by lap number. This shows that place cells can encode distinct events in time in addition to location.

2.1.2 Biological parallels of recurrent neural networks

There has been increasing evidence that recurrent neural networks (RNNs) are able to faithfully model brain activity, particularly that of the hippocampus and entorhinal cortex. Grid [14, 5] and place [71] representations mimicking that of real grid and place cells form once the recurrent network has learned a predictive task in the context of a complex environment. [5] in particular shows that navigation via routes that were not present during maze traversal training can be achieved when training an agent using an RNN architecture and reinforcement learning. This is akin to the function of biological grid cells which are critical for path integration and trajectory planning.

[13] demonstrates not only the emergence of characteristic neural representations, but also hallmarks of head direction system cells such as compass neurons when training a recurrent network on a simple angular velocity integration task. [87] show a plethora of location based cells forming when differently training three RNNs on a range of simulated navigation tasks, in particular showing anticipatory activity at decision locations. These emergent cells include head direction, boundary vector and egocentric boundary cells and their prevalence varies across each RNN. Additionally, based on the results of the study, the authors suggest that when the learned representation of the model is remapped across environments, the proportions of various location based cells remain consistent.

2.2 Modelling neural data

Neural recordings have recently seen significantly increasing dimensionality due to higher probe count and a rise in the number of channels per probe [62, 4, 6, 19, 21, 43, 56, 60, 90], allowing for large scale recordings of neural populations. The raw neural activity captured by these recording probes is usually spike sorted [51], whereby a threshold is applied to the activity to detect action potentials (or spikes) on the individual channels. These spikes are then clustered by firing profile and assigned to individual neurons.

Modelling approaches which can utilise this dimensionality in order to predict firing rates or behaviour must either directly model the joint activity of all recorded neurons, capturing specific population wide firing patterns and interactions across neurons, or extract informative low dimensional latent dynamics from the high dimensional data. Latent variable models allow us to extract low dimensional dynamics which describe firing patterns in high dimensional recordings without needing to observe the activity of all recorded neurons. Modelling approaches which do not reduce the dimensionality in some way are highly computationally expensive [37].

Brain-Computer Interface (BCI) systems which generally take neural activity as input and predict behaviour are an increasingly prominent application of this neural modelling. Most of this behaviour prediction currently takes the form of predicting movement intent from the neural activity of the motor cortex of human and non-human primates. The implications of accurate real-time movement decoding has the potential for tetraplegic human patients to be life changing. Current work in the field focuses on predicting intended cursor movement as input to a personal computer, allowing patients to interface with the world digitally.

The input signal to an intracortical BCI system for decoding movement usually consists of raw recordings taken from the motor cortex of a participant using surgically implanted electrodes (such as a Utah array [32]). Movement kinematics are usually recorded alongside neural activity for decoder model training purposes. The decoding pipeline for the stated BCI system would then consist of applying a spike detection algorithm to the raw data and spike sorting the resulting action potentials (as outlined above) into separate spike trains from individual neurons. Various modelling techniques can then be used to decode corresponding movement from the spike sorted neurons.

Decoding techniques can be categorised as being offline or online. Offline decoders refer to those approaches which require entire spike trains from the neurons in a recorded population in order to predict behaviour. Online decoders on the other hand only require small windows of around 150ms or less to predict momentary behaviour. Effective BCI decoding for real-world uses such as for prosthetic implementations require online decoders as these allow use in real-time.

Currently most online decoders in use are based on the Kalman filter [44] and involve repeated Bayesian inference for each time step. The Kalman filter assumes that spike trains from all neurons are noisy observations of a hidden latent state. The method allows for behaviour inference from the noisy spike observations and assumes that the relationships between neurons are linear with a Gaussian noise model. Nonlinear state estimators which build on the Kalman filter such as the Extended Kalman filter (EKF) and the Unscented Kalman filter (UKF) more accurately model the nonlinearities in recorded intra-cortical neural data. The UKF in particular has been shown to be more effective as a decoder in BCI systems than the standard Kalman filter [52, 53].

[57] propose a variant of the Kalman filter based on the restricted Boltzmann machine (RBM) [34] to model non-linearities which achieves state-of-the-art online decoding performance. The filter they introduce, termed the recurrent exponential family harmonium (rEFH), models spike trains per neuron as a Poisson process, allowing for non-linear dynamics. As the RBM on which the model is based is unsupervised, temporal correlations in spiking data can be explained by latent dynamics instead of directly relating to behaviour.

Offline decoding meanwhile is more straightforward, with higher overall decoding accuracy than online decoding due to offline models being exposed to many more timesteps of neuron spike trains. Non-linear models such as recurrent neural networks can be expected to achieve almost perfect decoding accuracy on held out trials of data from the same recording session after training, especially when the minimum trial length is at least 300ms.

2.2.1 Latent variable models

Neural population activity relating to behaviour has been shown to be inherently low-dimensional despite the observed high dimensionality of data recorded using multi-electrode arrays [12, 18, 23, 31, 59, 75] . The implication of this is that activity patterns across the entire neuron population are responsible for downstream behaviour, not the spiking of individual neurons. Therefore, predicting behaviour from neural population recordings has been shown to be most effective when using latent variable models. Dimensionality reduction methods such as Principal Component Analysis (PCA) and Variational Autoencoders (VAEs) [46] can be used to extract a low dimensional representation of the activity of a population of neurons.

2.2.1.1 Linear Models

While PCA and VAEs can be used to extract a latent representation of neural data, these methods do not model temporal dependencies across timesteps of each neuron. These methods are therefore considered static state-space models. Linear dynamical state-space models on the other hand aim to model temporal correlations across timesteps. Linear dynamical systems (LDS) [79] and jPCA [12] are examples of latent variable models which can capture the linear relationships between latent states. Gaussian Process Factor Analysis (GPFA) [89] aims to reduce the dimensionality of neural spiking data in order to visualise the trajectories of spike trains. GPFA first reduces dimensionality by applying factor analysis to spiking data and simultaneously smoothes the resulting low-dimensional trajectories by fitting a Gaussian process model to them. Unlike the state-space models described above, Gaussian process models provide both a measure of uncertainty along with model selection. The expectation-maximisation algorithm is used to estimate parameters of the Gaussian process from the data.

2.2.1.2 Non-linear Models

Although the methods outlined above find some dynamical structure outlining neuron firing patterns, these linear approaches do not model non-linear dynamics which have been shown to correspond to motor control [23]. An extension of GPFA, Gaussian Process Factor Analysis via Dynamical Systems (GPFADS) [74], proposes dynamical priors over trajectories which encourage temporal non-reversibility, allowing GPFADS to disentangle latent trajectories when applied to motor cortex neural data. Explicit temporal non-linear models such as recurrent neural networks (RNNs) fare better in modelling non-linear spiking data. An RNN based latent variable model such as Latent Factor Analysis via Dynamical Systems (LFADS) [68] acts as a sequential autoencoder and models non-linear dynamics, extracting a much more informative and interpretable latent space than the above methods.

When the latent space of LFADS is highly interpretable, we find that it is well disentangled with regard to properties of individual trials. These properties include recording session, recording subject and different aspects of behaviour or stimulus to which the neural activity corresponds to. We can visualise this trial separation by reducing the dimensionality of the latent space to two dimensions using PCA or t-SNE and then plotting the resulting variables. In many cases, these extracted latent variables can be used to reconstruct the original behaviour corresponding to the neural activity. In a well regularised LFADS model, latents can be inferred from the neural activity of previously unseen recording trials (within the same recording session), which can then be used to infer corresponding behaviour.

Targeted Dynamical Neural Modelling (TNDM) [38] utilises a variation of LFADS to separate neural activity into behaviourally relevant and irrelevant latent spaces, similar to [77]. This results in an even better separation of trials by corresponding behaviour in the behaviourally relevant latent space than is possible using LFADS, and therefore more accurate inferred reconstruction of behavioural variables such as hand movement from unseen trials.

2.3 Stable behaviour decoding

Neural activity recorded over multiple sessions which correspond to consistent simultaneously recorded distinct behaviours is highly unstable. This is largely due to drift in the activity of individual neurons and non-stationaries such as the minute movement of recording apparatus. In addition, neurons can move and be replaced by different neurons, therefore keeping track of neurons across sessions becomes infeasible. Spike sorting methods will place any neurons which are consistent across recording sessions in different positions in the resulting downstream neural data for each session, with no consistent ground truth with respect to neuron identity. Current decoding approaches such as gated RNNs [35] and latent variable models such as LFADS and TNDM are highly sensitive to the positions of individual neurons within neural data. Therefore reliable decoding across sessions is impractical with these methods.

[22] show that spiking data over many days from the same subject share a latent space. The authors apply PCA to high dimensional neural data over several days (over the course of a year) and show that this neural activity has underlying dynamics which are stable and recoverable over many days by alignment using canonical correlation analysis (CCA). After applying CCA, neural activity can be reconstructed for a recording session many days into the future. Furthermore, accurate decoding of behaviour is possible from these future sessions once alignment has occurred.

Recent work aims to allow for reliable decoding of behaviour across recording sessions with only relatively little retraining using neural data from an unseen session. For example, [20] produce an aligner capable of mapping neural data from any session to an original ”ground truth” session. The authors train an adversarial model to align EMG neural activity over sessions recorded from many days from a single monkey. A discriminator network based on a VAE is trained to autoencode neural activity from a day 0 recording session. Subsequently, the discriminator is trained to maximise the difference between the neural reconstruction losses of day 0 and other future days. Then a generator network is simultaneously trained to align neural activity from other future days to that of the day 0 session by minimising the neural reconstruction losses. This then allows for reliable behaviour decoding from the latent variables of the discriminator network across recording sessions as these latent variable values have been maintained across days for consistent behaviours.

Similarly to [20], [45] aim to produce an aligner model without any behaviour data required for recalibration. The authors train a model akin to LFADS on a day

0 session, with an added readout layer to predict behaviour from the generator RNN. Neural activity from a day k session is then used to recalibrate the model. This is done by training the model (using backpropagation) to minimise the Kullback-Leibler (KL) divergence between the output of the generator RNN on day 0 and day k. Reliable behaviour decoding from the readout layer of the generator RNN which was only trained on day 0 is then possible for day k.

The above three methods are unsupervised in the sense that they can be recalibrated for optimal decoding performance on new recording sessions with only neural activity and without access to behaviour such as kinematic data from a session. This is useful for BCI applications where simultaneous behaviour recordings are not available for recalibration, such as for continuous BCI decoding from tetraplegic patients.

[84] train an RNN to predict consistent behaviours from many months of previously recorded neural data. The decoder is made robust through training on the plethora and variability of the recorded data over many sessions. In addition, the authors further perturb this neural data during training in order to inject increased artificial robustness into the decoder.

[88] use a generative adversarial network to generate synthetic neural data from a held out session by utilising the behaviour of that session. When this synthetic neural data is utilised in conjunction with some real neural data from the held out session, relatively high behaviour decoding accuracy is reported by the authors on a held out session. Less neural data from the held out session is used than with the above approaches.

Although the above approaches are somewhat successful in recalibrating a model to decode behaviour from a new session of recorded neural activity, this recalibration still requires some data from the new session, and importantly for real-world continuous use cases, this data acquisition requires time to collect and computation for model parameter updates. Therefore these models are not truly robust to unseen sessions of neural recordings corresponding to a subject performing consistent behaviours.

With respect to latent variable modelling, while a low dimensional manifold is recoverable across sessions, alignment is still required in order to predict behaviour from the corresponding latent variables of an unseen session. Without this alignment, behaviour decoding performance is impractical. The next two sections outline approaches from machine learning which are used in chapters 4 and 5 to create truly robust decoders capable of predicting behaviour with high accuracy from trials which have been recorded in sessions which are completely unseen to the decoder with no further recalibration.

2.3.1 Domain Adaptation

Domain Adaptation is a field of machine learning which aims to predict consistent classes from various differing sources. When each of these sources varies substantially they are considered different domains. For example, Figure 2.1 shows examples from a dataset with differing domains (house number styling) where the classes to be predicted (digits from 0 to 9) are consistent.

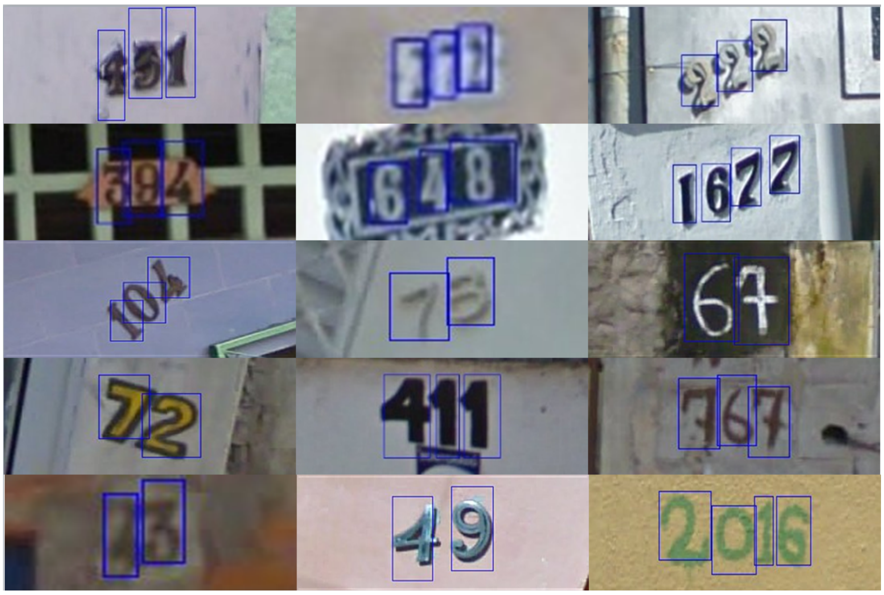

Figure 2.1: Samples from Street View House Numbers dataset.

In order to predict consistent classes from differing domains, [24] use an unsupervised adversarial approach where a reverse gradient is utilised to predict classes from domain invariant features. The reverse gradient is implemented between a domain classifier, which aims to predict the domain identity of each data sample, and a feature extractor which is optimised in turn to produce features from which domain identity is difficult for the domain classifier to predict, thereby unifying domains and improving class prediction performance.

[26] utilise a similar principle to remove inter-experimental variability across experiments where large scale two-photon imaging data was gathered. The authors aim to predict cell-type classes from imaging data which has been aligned across experiments and aim for this alignment to improve prediction performance. Their model structure is relatively similar to that of [24] where a reverse gradient is applied between a domain

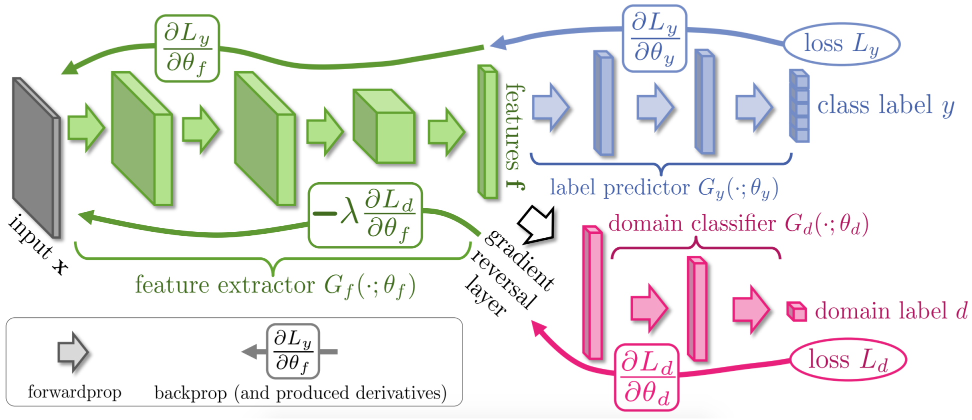
$\frac{\partial L_y}{\partial \theta_f}$
$\frac{\partial L_y}{\partial \theta_y}$
loss L_y
class label y
features $\mathbf{f}$
label predictor $G_y(\cdot;\theta_y)$
$-\lambda \frac{\partial L_d}{\partial \theta_f}$
input $\mathbf{x}$
feature extractor $G_f(\cdot;\theta_f)$
gradient reversal layer
domain classifier $G_d(\cdot;\theta_d)$
domain label d
$\frac{\partial L_d}{\partial \theta_d}$
loss L_d
forwardprop
backprop (and produced derivatives)

Figure 2.2: Adapted from [24]. Model structure showing reverse gradient between domain classifier and feature extractor.

(or experiment ID) classifier and an encoding network. A class predictor then predicts cell type from the output of the encoding network. The reverse gradient is instrumental in adversarially forming a domain invariant encoder.

Other domain adaptation methods such as Pixel-DA [8] and CycleGAN [91] are useful for mapping data from a single source domain to a target domain but are not effective in merging data from across many domains. In chapter 4 I treat individual recordings sessions from motor cortex as different domains and aim to create an invariant latent space across 12 domains (recording sessions) in order to create a robust decoder which can predict movement behaviour from a completely unseen recording session. Therefore I present a model based on [24] as I aim to merge many sessions instead of mapping individual sessions to a single target session.

2.3.2 Self-supervised Learning

Self-supervised learning is a machine learning paradigm where auxiliary tasks associated with unlabelled and labelled data are used for training as opposed to direct classification training of said data. Examples include identifying which augmentations have been applied to samples of data, such as predicting the amount by which image data has been randomly rotated. This paradigm of learning has been shown to be successful in learning useful representations of data for downstream tasks in computer vision and reinforcement learning [15, 69, 33, 11, 28, 9, 80, 67, 29, 78].

Useful representations are learned by minimising the representational distance within a given model between various perturbed versions of the same data sample, while simultaneously maximising the distance between perturbed versions of other data samples. The primary method to achieve this is contrastive learning, where positive examples (of the same perturbed data samples) are compared to perturbed versions of other data samples (negative examples) by a proposed metric. The model is then optimised to have dissimilar representations between these positive and negative examples.

Recent work in self-supervised learning applied to neural data with the aim of predicting behaviour include [3]. By applying a range of augmentations to neural data such as temporal jitter, the authors aim to train a model capable of finding nearby neighbours in the representation of the model and subsequently predict the latent variables of one trial of neural data from another nearby trial. This method surpasses previous self-supervised methods such as [28] in predicting the reach direction of trials of two monkeys performing a centre out reach task from the corresponding neural data.

[55] propose a self-supervised approach which aims to learn disentangled representations of neural data. The model used is based on a VAE [46], which is optimised to maximise the representational similarity in the model latent space of different transformed views of the same data samples (trials). Similarly to [3], the authors apply augmentations such as temporal jitter and neuron dropout to trials of neural data. For each trial, two different views are created. The VAE encoder then predicts latent variables from each of these views. The latents are split between a ”context” space and a ”style” space.

The goal is to have the context space be consistent across both views and the style space be specific to each view. To achieve this, the authors swap the context space of the two views before the latents of each view are input to the VAE decoder network, which reconstructs the original view associated with the input to the encoder before swapping has occurred. This ensures that the context segment of the latent space remains consistent across views and thus stable to the augmentations applied to create the views. Behaviour such as movement direction can then be accurately predicted from the context space of the latents of an unseen trial. The authors indeed show good decoding accuracy from unseen trials within a single session, but do not show decoding performance across sessions. Good decoding accuracy across sessions is highly unlikely with this model as the temporal dependencies across each trial of neural data are not modelled.

The above recent work using self-supervised learning in modelling neural activity shows the effectiveness of selectively perturbing neural data in order to learn relevant latent variables. These models take different views of the same neural data and align the latent spaces of these views once passed through an encoder, with the ultimate aim of reconstructing these views. In chapter 5, I utilise a similar technique to train our sequential autoencoder by aligning the latent variables of perturbed versions of the same data and aim to generate the activity of the original unperturbed trial. Whereas [55] propose a model which is invariant to the specific neurons used to represent the neural state within training data, I look at unseen sessions and so do not aim to produce a model invariant to new neurons, but one that is able to identify and utilise seen neurons to reconstruct completely unperturbed trials.

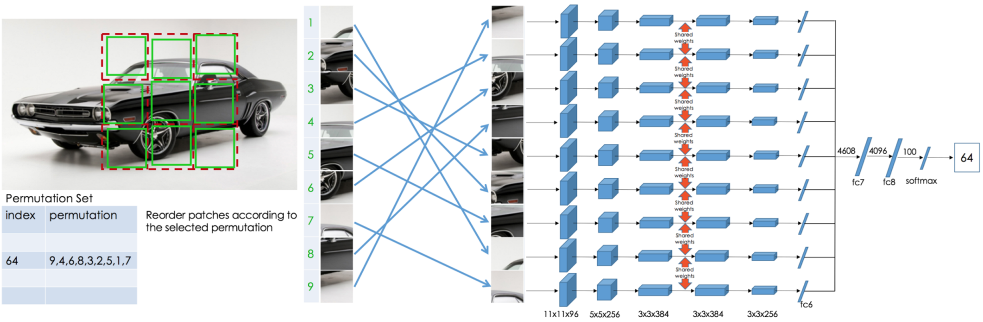

Figure 2.3: Adapted from [63] - Authors form 9 subsets of images and form random permutations of these subsets. The model is then tasked with predicting the identity of a given permutation. The representation resulting from training on this auxiliary task is then used to predict classes of images from imagenet with good accuracy in a self-supervised fashion (without knowledge of class labels).

[63] is an example of a self-supervised model from computer vision which is trained to understand the structure of separate classes of images (and how this structure differs between classes) by learning an auxiliary task (shown in Figure 2.3). This model is robust in predicting the class of unseen images without being trained on any class labels and is capable of state-of-the-art downstream class prediction on ImageNet. I use this principle of learning structure of features of image data and shift neurons in our neural data by a random amount for each trial. I also jitter the trials in time slightly and dropout neurons at random. The dropped out neurons are replaced with randomly generated spike trains to simulate new neurons in an unseen session of data. I aim to apply the notion of image structure learning to neural data in order to reverse changes to neuron ordering over several days of recordings in chapter 5.

2.4 Thesis

2.4.1 Scope

In this thesis I introduce one piece of work showing that signatures and firing patterns of hippocampal cells emerge when a recurrent neural network is trained on similar tasks to that of a mammalian rodent. Next I introduce two pieces of work which aim to produce session robust behaviour decoders of neural activity. These works utilise methods from machine learning in order to decode behaviour from completely unseen neural recording sessions.

Scope Extrusion Throughout this thesis I will not explicitly outline the process of spike sorting or of how neuron populations are recorded from. The data used in chapters 4 and 5 to train the models have been spike sorted by our collaborators. It should also be noted that the methods described in chapters 4 and 5 are only applied to offline decoding of entire trials of neural data and not in real time as is required for effective use with brain-computer interfaces (BCIs). I do not discuss the state-of-the-art in terms of online behaviour decoding in this thesis but refer the reader to [57].

2.4.2 Contributions

The key contributions of this thesis are as follows:

Recapitulation of experimentally found hippocampal representations

- I show that the place cells which emerge within the representation of an RNN controlling an agent performing a maze traversal task present firing selectivity consistent with that shown in experimental neuroscience.
- I then show that this learned representation is advantageous when the artificial agent is navigating towards rewarding locations in the maze.
- I show that this learned representation tends to sweep ahead of the agent and replay the path ahead of it, confirming results from experimental neuroscience.

Session invariant latent representation via domain adaptation

- I treat each recording session of neural activity as a separate domain.

- I then propose a model which is trained on many recording sessions of neural activity, each acting as a separate domain.
- I show that by utilising a negative gradient, a domain invariant latent space can be obtained.
- I show that, due to this session invariant latent space, this model generalises to a completely unseen session when trained using the paradigm of unsupervised domain adaptation.

Capturing neural variability using self-supervised learning

- I introduce a set of perturbations which are applied to all trials of a single recording session which are intended to mimic real inter-session variability.
- I then propose another model trained using a self-supervised learning technique, whereby each perturbed version of a trial is mapped to the original.
- I show that, when trained only on perturbed trials of the single recording session of neural activity, this model generalises to unseen sessions for up to a week into the future.

Chapter 3

Hippocampal representations emerge when training recurrent neural networks on a memory dependent maze navigation task

In this work I aimed to explore if representations of space form in the units of a recurrent neural network when trained to predict subsequent environmental stimuli while an artificial agent performs random walks of a maze. This is a task-optimised model, where, instead of explicitly hand crafting a computational model of the hippocampus based on experimental data, the model is trained on tasks the hippocampus is adept at. If behaviours of a computational model trained in this way are similar to that of the hippocampus, then there is evidence to suggest that the hippocampus learns in the same way. My starting hypothesis is based on the predication that the mammalian hippocampus acts as a predictor, and learns to predict subsequent sensory stimuli given actions (movements) in a given environment [70]. Furthermore, I find that spatial representations which form in the RNN units are highly comparable to those in the mammalian hippocampus.

I confirm the above hypothesis, and subsequently show that the resulting spatial representation learned by the RNN can be employed to efficiently learn goal-directed behaviour in a reinforcement learning task by directly controlling the actions of the artificial agent with the RNN. I show that simultaneous predictive learning of the environment and Q-learning to reach reward locations using a pre-trained RNN converges to perfect performance on a reward navigation task much faster than Q-learning alone with an environmentally pre-trained RNN or without pre-training the RNN on the environment at all.

This is likely due to the learned representation of the environment within the RNN being overwritten during Q-learning alone, instead of this representation being directly utilised for navigation, as is the case when both tasks are optimised simultaneously.

This combined loss training paradigm also recapitulates many key observations from hippocampal place cells within the units of the RNN:

1. Non-metric attractors form in the activation space of our network units in the way of place cells, uniformly covering the maze environment. [65]
2. Extrafield firing of these units at locations outside of their apparent place fields. [39]
3. Non-local forward sweeping representation of the network. [39]
4. Place fields drifting towards reward locations throughout reward training. [49]
5. A high proportion of network units with place fields at the maze start location encode reward locations. [1]
6. A higher proportion of network units encode task phase than turn direction. [27]

3.1 Contribution

I am the first author and lead of this work. As such, I conceptualised the model, implemented all versions of the model, ran and evaluated the methods, and wrote the manuscript along with Matthias Hennig.

3.2 Paper

Hippocampal representations emerge when training recurrent neural networks on a memory dependent maze navigation task

Justin Jude
University of Edinburgh
justin.jude@ed.ac.uk

Matthias H. Hennig
University of Edinburgh
m.hennig@ed.ac.uk

Abstract

Predicting the future outcomes of actions forms the basis for reinforcement learning to shape goal-directed behaviours. Recent work showed that learning based on predicting future sensory experience using the current state and action of an agent leads to representations that resemble those in the brain, for instance place and grid cells of the medial temporal lobe. Here we ask if combining ongoing predictive learning of sensory events and of notional value of actions leading to rewards forms representations that enable more efficient goal-directed learning than reinforcement learning alone. We find that once a recurrent network is trained to learn the structure of its environment solely based on sensory prediction, a landscape forms resembling hippocampal place cells. Next, we introduce cued rewards, and train the network to predict state-action Q-values which are used to guide subsequent behaviour. A network previously exposed to the same environment without rewards learns the task faster than a network trained using Q-learning alone, or without previous exposure. Interestingly, this training paradigm causes non-local neural activity to sweep forward in space at decision points, anticipating the future path to a rewarded location. Moreover, prevalent choice and cue-selective neurons form in this network, again recapitulating experimental findings. Together, these results indicate that a simple combination of predictive, unsupervised learning of environment structure and of reinforcers yields efficient representations to support goal-directed behaviour and exhibit dynamics also found experimentally in the hippocampus when learning similar tasks.

1 Introduction

Recurrent neural networks have been used to perform spatial navigation tasks and the subsequent study of their internal representations has yielded dynamics and structures that are strikingly biological. Metric (Cueva & Wei, 2018; Banino et al., 2018) and non-metric (Recanatesi et al., 2019) representations mimicking grid (Fyhn et al., 2004) and place cells (O'Keefe & Nadel, 1978) respectively form once the recurrent network has learned a predictive task in the context of a complex environment. Cueva et al. (2020) demonstrates not only the emergence of characteristic neural representations, but also hallmarks of head direction system cells when training a recurrent network on a simple angular velocity integration task. Biologically, non-metric representations are associated with landmark spatial memory, in which place cells within the mammalian hippocampus fire when the associated organism is present in a corresponding place field. Non-metric place representations differ from metric grid representations as place cells do not inherently encode mappings or distances across positions in space, and in this sense they cannot be readily used exclusively in mammalian navigation. Therefore, in contrast to grid cells, place cells only encode individual (or sometimes multiple) environment locations in isolation. In this paper, we show that although place representations do not encode mappings between distant locations, they are effective as a consolidated foundation of space for difficult navigation tasks.

Extrafield firing of place cells occurs when these neurons spike outside of these contiguous place field regions. Here we show that recurrent neural networks (RNNs) produce representations with internal dynamics that closely resemble those found experimentally in the hippocampus when performing goal-directed behaviour in a predictive learning framework. Importantly, experimental research in neuroscience shows that non-metric representations are not entirely context-free, but exhibit task-related activity. For instance, Johnson & Redish (2007) show that spatial representations in mice in the CA3 region of the hippocampus frequently fire nonlocally. Griffin et al. (2007) show that a far higher proportion of hippocampal neurons in the CA1 region in rats performing an episodic task in a T-shaped maze encode the phase of the task rather than spatial information (in this case trajectory direction). Ainge et al. (2007) show CA1 place cells encode destination location at the start position of a maze. Lee et al. (2006) demonstrate that place fields of CA1 neurons gradually drift toward reward locations throughout reward training on a T-shaped maze. The interpretation of these results is however currently unclear.

In this work we show that a recurrent neural network learning a choice-reward based task using reinforcement learning, in conjunction with predictive sensory learning produces an internal representation with consistent extrafield firing associated with consequential decision points. In addition we find that the network's representation, once trained, follows a forward sweeping pattern as reported experimentally by Johnson & Redish (2007). We then show that a higher proportion of units in the trained network show strong selectivity for the encoding or choice phase of the task than the proportion showing selectivity for spatial topology. Importantly, these properties only emerge during predictive learning, where task learning is much faster compared to traditional deep Q learning.

2 Method

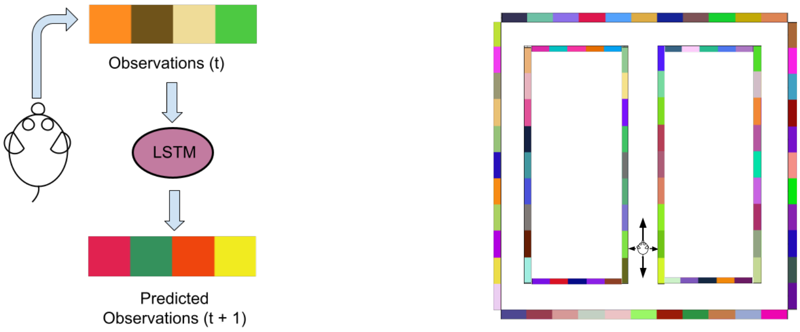

Figure 1: Left, the wall observation received by the network at each timestep. Right, the artificial agent performs exploratory trajectories though the maze environment, from which we task the LSTM network to learn a predictive representation.

We use a form of the maze used by Johnson & Redish (2007) which has a central T structure with returning arms, shown in Figure 1. All walls of the maze are tiled with distinct RGB colours which are generated at random and remain fixed throughout. The length of each outer edge of the maze is 11, with the width of the maze being 1 at all locations.. An agent with a 1 square footprint is initially learning to predict the next sensory stimulus during exploration of the maze. This exploration performed by the agent is constrained such that the agent always moves in the forwards direction and at a constant speed. When the agent approaches the top or bottom of the maze stem, the agent moves in one of two random directions which is not the reverse direction of its current heading. This combination of unsupervised learning and exploration has been shown previously to produce place cell-like encoding of the agent's position (Recanatesi et al., 2019). Later, rewards at four possible locations are introduced and the agent is tasked with associating a cue with the rewarding trajectory. The agent has four vision sensors, one in each cardinal direction, reading the wall RGB colours they intersect at any distance, with a small amount of added Gaussian noise. Therefore, at each time step the RNN receives a total of 12 values (red, green and blue components for each of 4 wall colours intersected by the agent's sensors). The distance of the agent to a particular wall segment does not affect the RGB values received.

The agent is controlled by a simple recurrent neural network comprised of a single 380 unit Long-Short term memory (Hochreiter & Schmidhuber, 1997) (LSTM) network with a single layered readout for the prediction of RGB values. We do not impose any regularisation on the network. We first pre-train the network by tasking it with predicting the subsequent observation of wall colours from the currently observable wall colours given its trajectory through the maze. As in the experiments by Johnson & Redish (2007), during pre-training the agent does not choose any of its actions and is only learning to predict the sensory inputs it encounters. In a given pre-training iteration, we collect all observations as the agent traverses the maze until it returns to the start location at the bottom of the central stem and finally train the LSTM on the entire collected trajectory. The network is trained with a mean-squared error loss of predicted and target RGB wall colours (Eq. 1), with model parameters optimised using Adam (Kingma & Ba, 2015) and a learning rate of 0.0002:

$$loss_{rgb} = \frac{1}{T-1}\sum_{t=1}^{T-1}(rgb_{t+1} - \tanh\left(W_{rnn}.[h_{t-1}, rgb_t] + b_{rnn}\right))^2 \tag{1}$$

where T is the total number of timesteps in a given trajectory, rgb_t is a 12 x 1 sized vector containing the RGB values for the 4 vision sensors of the agent at timestep t. rgb_{t+1} indicates the RGB sensor values of the subsequent timestep, which the network aims to predict. W_{rnn} and b_{rnn} are the parameters of the RNN, h_{t-1} is the state RNN at the previous timestep with h_0 being zero.

To solve this predictive task, the network has to maintain the sequence of encountered wall observations in its internal memory for several time steps in order to predict subsequent wall colours. In our model, this is achieved through the network forming a non-metric representation of the maze environment, as also demonstrated by Xu & Barak (2020). Unsurprisingly, the RNN has difficulty in predicting the subsequent set of wall colours when the agent reaches the top or bottom of the maze stem, due to the randomly chosen turn directions at these locations.

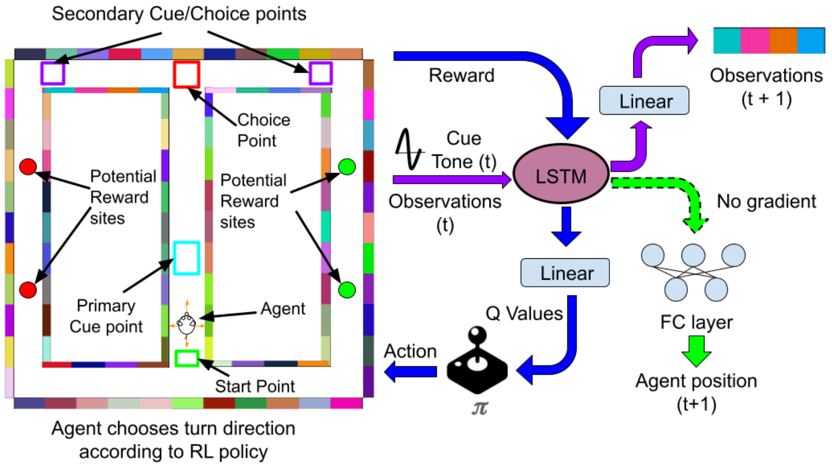

Figure 2: For the joint prediction and reinforcement learning task to be learned by the LSTM network, we introduce a cue which indicates future reward locations to the agent. The cue is first played halfway up the central maze stem (primary cue location). At secondary cue locations, the same cue tone is repeated if and only if the agent has proceeded in turning in the direction corresponding to the cue tone frequency given at the primary cue location. The agent is free to choose the next action to be taken when traversing the maze at either the choice point at the top of the stem of the maze or at the secondary cue locations. There are two potential reward sites on both returning arms, with the reward sites being active if the agent is on the returning arm corresponding to the cue tone frequency.

Once the LSTM has formed an internal representation of the maze, the agent is tasked with navigating towards potential reward sites whose location is indicated by a cue signal: a low frequency cue indicates active reward sites on the left return arm and a high frequency cue indicates active reward sites on the right return arm - the cue tone and corresponding side of active reward sites are together chosen randomly at each iteration with a secondary cue given if the agent has turned correctly. In this phase there are three choice points, while the movements are constrained to follow the forward maze direction elsewhere: at the top of the maze stem and at the two secondary choice points (Figure 2), with initially random movement at these points during reward training. There are 5 steps between the cue and choice points and 7 steps from the choice point to the first reward site on either return arm. The inclusion of the secondary cues as additional choice points was motivated by the experimental set up used by Johnson & Redish (2007), to compare the network activity at these points to experimental data. These secondary points also give the agent the opportunity to backtrack on its decision made at the primary choice point in light of further environmental observation (the presentation or lack thereof of the secondary cue), and make learning more efficient in our model. This may explain how it speeds up training the animals in the same task.

We additionally introduce a new single layered readout for the LSTM network which predicts state-action values associated with the four cardinal directions in relation to the agent's current position and direction. At each timestep, this ensemble receives the agent's environment observation and the agent follows an epsilon-greedy policy (starting with fully random movement at choice points and a decaying epsilon thereafter) for choosing optimal actions of those available at each of the three choice points. The recurrent network controlling the agent is trained on a weighted combined loss of a reinforcement learning (RL) task loss and the previously described predictive wall colour loss:

$$loss_{combined} = |Q(s,a) - (r + \gamma \cdot Q'(s', \arg\max_{a'} Q(s',a'))| + \lambda \cdot loss_{rgb} \tag{2}$$

The first component of this loss is the difference between predicted and observed state-action values which are represented by Q-values (Watkins & Dayan, 1992), which are a prediction of future global reward:

$$h_t = \tanh\left(W_{rnn} \cdot [h_{t-1}, rgb_t] + b_{rnn}\right) \tag{3}$$

$$Q(s,a) = W_Q h_t + b_Q \tag{4}$$

where h_t is the output of the RNN at each timestep and W_Q and b_Q are the parameters of the new readout layer we use to predict Q-values. We use double-Q learning (Van Hasselt et al., 2016) to train the agent on the task, updating the target Q value predictor (Q' - a LSTM with same number of units) every 15 training iterations. Double-Q learning allows for optimal performance on the reward task in drastically fewer agent maze traversals and network training iterations than with standard DQN (Mnih et al., 2013) based Q-learning which suffers from overestimation of Q-values. We settle on a discount factor (γ) of 0.8 as values higher than this regularly cause the network to converge on solutions wherein the agent does not take the most direct path to reward locations, with backtracking at secondary choice points. The second loss component is the sensory prediction task which we used to pre-train the network (λ decays from 10 to 0.02 throughout training). This loss component is included when training the network on the reward task so that the spatial map of the maze environment formed during pre-training is maintained throughout Q-learning. This ensures the map is not overwritten as would happen when Q-learning is performed alone, and leads to faster task learning (see results). We optimise the network for this joint task using Adam and a learning rate of 0.0002, which we find improves the rate of convergence with optimal task performance, as opposed to higher learning rates which still converge but with backtracking at secondary choice points often inherent in task solutions.

In contrast to much of the previous work on spatial representations in recurrent networks, we do not give the network any indication of the agent's location or movement. This makes the task considerably more difficult due to the unpredictable movement possible at choice points during the reward task. The network is coerced into storing the current movement direction of the agent in its representation, in addition to storing the cue frequency. As such, a network of Gated Recurrent Units (Cho et al., 2014) (GRUs) or vanilla RNN units was unable to perform well in either the pre-training or joint RL task due to these prevalent long term dependencies (18 steps between cue and final reward).

To analyse the representations formed by the network, we train a further single layered fully connected network (shown in green in Figure 2) to predict the agent's next location using the activity of the LSTM. There is no backpropagation of gradients between this predictor and the LSTM network, and the predictor is trained at the end of reward training. This network followed by a softmax layer, generates a distribution indicating the probability of agent location inferred from LSTM activity. The plots in Figure 5 are examples of this. This is used in place of the decoding algorithm used by

Johnson & Redish (2007) to predict the neuronally inferred maze location of rats when performing a cue based task.

3 Results

The agent learns the sensory prediction task to a high degree of recall and after around a thousand training iterations (combined loss with pre-training in Figure 3), the agent is able to achieve perfect performance on the reward task when the LSTM network has 380 or more units (Fig. 3, right). We trained the reinforcement learning (Eq. 2) portion of the task in an epsilon greedy manner, with a steadily decaying epsilon to ensure that the agent would choose the rewarding path consistently once actions were chosen at choice points completely by the network. Notably, the agent did not turn at either of the secondary choice points once training had completed - only at the primary choice point.

We attempted to run the reinforcement learning task alone in a maze with no sensory input except the reward cue. In this scenario the network is not able to learn the task due to a lack of self-localisation and is unable to perform the task based on step counting between the cue and choice point. In addition, the reward based reinforcement learning task was attempted using Q-learning alone with a loss function that did not include the wall colour prediction error, both with and without pre-training (shown in Fig. 3, left). In both cases we find that the reward task is not learnable with the same higher rate of epsilon decay we use for the combined loss function with pre-training, as the network quickly forgets the consolidation of the maze formed during pre-training, which we maintain through the combined loss (Eq. 2). We also find the network can solve the reward task using the combined loss without pre-training, albeit in around 3 times the number of maze traversals as with the use of the spatial map formed in the pre-trained case (Fig. 3, left).

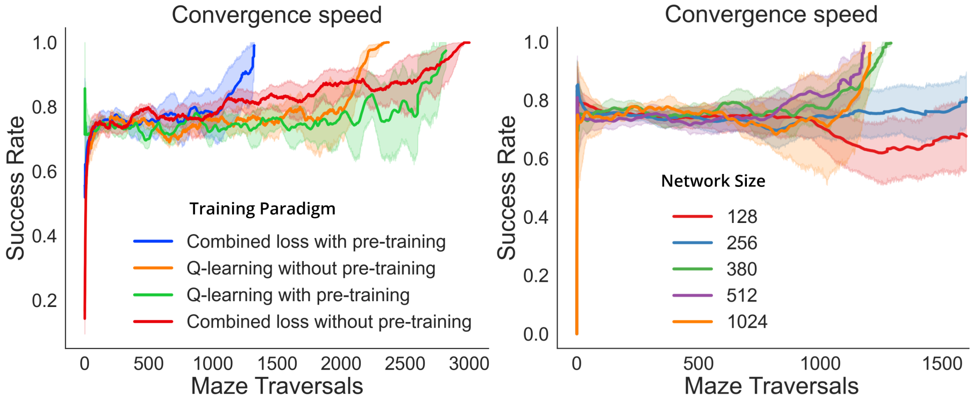

Figure 3: Left: Success rate (proportion of direct traversals to reward locations) of each set of training paradigms on the reward task, averaged over 10 initial conditions and random wall colours using optimal rate of epsilon decay for each paradigm, each shown with a 95% confidence interval. Place representation formed during pre-training alongside combined loss allows network to achieve perfect performance on reward task in relatively few maze traversals. Q-learning alone without pre-training also achieves perfect performance in more than twice the number of maze traversals. Q-learning alone with pre-training takes far more maze traversals to converge (and is less likely to be optimal) due to the non-random initial state of network and inability to utilise the spatial map formed. Combined training without pre-training also takes relatively many maze traversals to converge due to a relatively difficult joint task with no biased initial state. Right: Pre-trained network optimised with combined loss converges at similar rates with different network sizes above 380 units.

3.1 Extrafield place cell firing

First, we investigate the representation learned by the network. Pre-training causes the formation of a consolidated representation that resembles place cells in the hippocampus. We observe a substantial increase in activity in a particular unit when the agent moves across its respective place field. Individual units in the network generally have well isolated place fields, which together cover the whole maze and therefore allow reliable decoding of agent location. After training on the reward task, the network units exhibit substantial extrafield firing with respect to the previously formed place fields. We especially see units with extrafield activity at the primary and secondary cue points (Fig. 4E).

In the top row of Figure 4(A-D) we show activity in four reward trained LSTM units obtained through the collection of unit activity from a full left sided trajectory from the maze start point returning to the start point with cues presented, together with a full right sided trajectory. We show all activity from this activity collection in the top row of Figure 4(A-D). Maze areas for each unit with activity higher than 30% of the peak activity of each unit are defined as place fields (mirroring the experimental threshold used by Johnson & Redish (2007)). In experiments, rodents seem to pause at high consequence decision points (Johnson & Redish, 2007) with alternating head movement behaviour signifying vicarious trial and error (VTE) (Muenzinger, 1938; Hu & Amsel, 1995). In the activity plots in the bottom row of Figure 4(A-D), we simulate this using our reward trained model by running the agent from the start position at the bottom of the maze stem, then pausing it at the top of the stem, with a left cue presented halfway up. We show activity above 60% of unit peak activity (identified with the previously collected aggregated activity) shown in addition to the previously identified place fields.

The network representation seems to have substantial activity corresponding to both return arms, with surprisingly high extrafield activity in the shown LSTM units when the agent is paused at the maze choice point, a location for which these units do not usually have corresponding activity (Fig. 4B-D). To generate the extrafield firing maps in the bottom row of Figure 4A-D), we first define extrafield firing as unit activity averaged over the number of paused timesteps which exceeds 60% of peak timestep averaged unit activity while the agent is paused at the top of the maze stem. For Fig. 4B-D) bottom row, the number of timesteps usually taken by the agent during normal motion to reach each dotted place field appears to be a sufficient number of steps before timestep averaged activity at the paused choice point reaches 60% of timestep averaged peak activity.

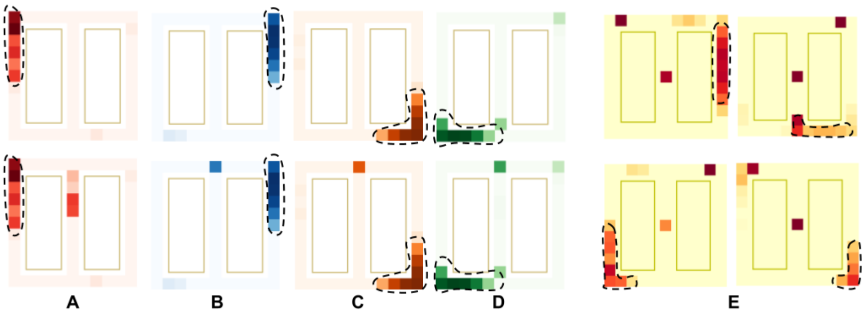

Figure 4: **Two forms of extrafield firing emerge: A-D) Firing of place units with place fields ahead of the paused agent:** *Top row*: Activity maps showing well isolated place fields of four RNN units (acting as place cells) form after predictive pre-training, indicated in dotted regions. *Bottom row*: After reward training, we see that when the agent moves from the start to the top of the maze stem (with a left cue presented), then kept stationary at the choice point with the RNN repeatedly receiving observation from choice point for several timesteps thereafter, we see extrafield firing, indicating that these units with place fields ahead of the agent are active while the agent is paused (shown in addition to previously determined unit place fields in dotted regions). **A**) Strong extrafield firing between cue and choice point, due to place field being present on left side of maze (firing in conjunction with the sole presentation of a left cue). This example shows that extrafield firing at the choice point does not always occur. **B**, **C**, **D**) High extrafield firing at choice point while agent is paused at top of stem for many timesteps. **E**) **Extrafield firing outside of contiguous place fields at primary and secondary cue points:** Place fields outlined in dotted areas (determined from average activity on both trajectories) of four RNN units forming after pre-training. After training on the reward task we observe high levels of consistent extrafield firing at primary and secondary cue points in 56% of RNN units.

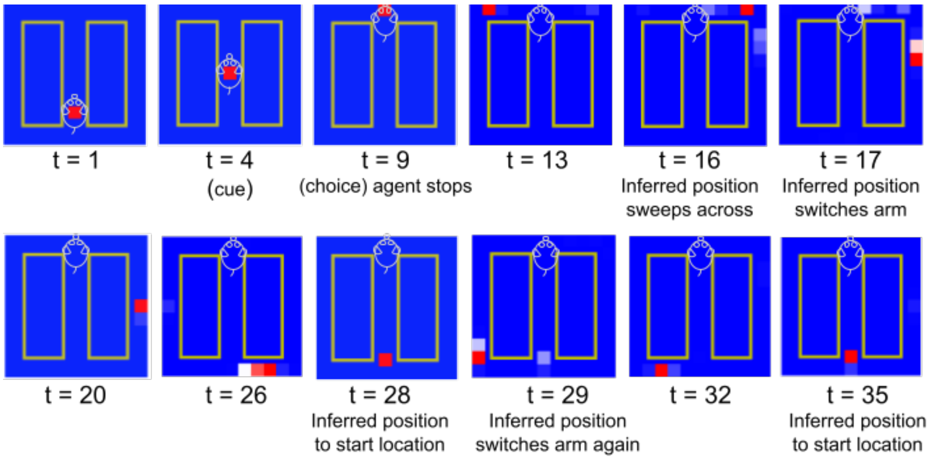
t = 1
t = 4
(cue)
t = 9
(choice) agent stops
t = 13
t = 16
Inferred position sweeps across
t = 17
Inferred position switches arm
t = 20
t = 26
t = 28
Inferred position to start location
t = 29
Inferred position switches arm again
t = 32
t = 35
Inferred position to start location

Figure 5: **RNN inferred position of the agent moves ahead of the real agent position, with the network replaying previously learned spatial trajectories through the maze**. We run the agent from the start position to the top of the stem of the maze at timestep 9 with a low frequency (left) cue tone at timestep 4. The agent is stopped at this position with the LSTM network receiving the environment observation from this position for the remainder of the shown timesteps. The RNN representation inferred agent position (inference shown in green in Figure 2), moves in the direction corresponding to the frequency of the given low frequency cue tone (left). Then between timesteps 13 and 17, the inferred position jumps from the return arm with active reward sites to the alternate arm, with the inferred position moving from this position to the start location fairly consistently. Then the inferred position jumps again at timestep 29 to the rewarding return arm and moves constantly to the start position.

3.2 Forward moving representation

In contrast to the static dynamics of the LSTM network after predictive pre-training, following training on the reward task, the forward representation of the LSTM is looking ahead of the agent and is now displaying sweeping behaviour (Fig. 5) which is identified experimentally in rats by Johnson & Redish (2007) when performing cue based tasks. When the agent is stationary at the choice point, once it has moved from the bottom to the top of the maze stem receiving a low frequency cue, we observe the representation (inferred agent position) moving ahead of the agent - first in the direction corresponding to the cue given at the first cue point and then abruptly down the opposing arm of the maze towards the starting location. Thereafter the representation moves down the correct arm (corresponding to the cue) and becomes stationary at the maze start location. This path switching behaviour is reliably observed in networks trained on the combined loss (Eq. 2) with and without pre-training, with differing numbers of units and initial conditions as long as the reward task is solved without backtracking at secondary cue locations. The network lacks a sweeping or forward moving representation when trained on the reward task with Q-learning alone, regardless of pre-training. Thus pre-training does not contribute to sweeping or path switching behaviour.

We visualise these dynamics using Uniform Manifold Approximation and Projection (UMAP) (McInnes et al., 2018) in Figure 6. This shows generally connected manifolds over time for each trajectory, with closer inspection revealing the dynamics which leads to the sweeping arm behaviour in Figure 5 when the agent is stationary at the primary choice point. Zeroing visual input while the agent is paused at the choice point gives comparable representation dynamics to that observed in Figures 5 and 6.

Separately, we find that place fields of particular LSTM units drift forwards from their original firing positions after pre-training, towards the reward locations on the return arms throughout reward training, as shown experimentally in CA1 neurons in Lee et al. (2006). We observe this behaviour in 56 out of 380 network units (15%) which have final resting locations of place fields at reward locations (Figure 7) out of a total of 116 units with place fields which move generally, with the remaining 60 having place fields moving to other random locations. We use chi-square to test if

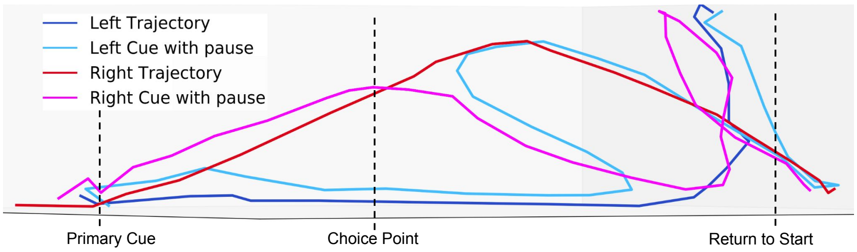

Figure 6: **Low dimensional manifold of network dynamics shows trajectory switching phenomenon**. We apply UMAP dimensionality reduction to the activations of the LSTM at each timestep. The x-axis represents timesteps, with the primary cue, the choice point and the return to the maze start position indicated. The y-axis is the 1-dimensional UMAP representation of the 380 unit LSTM and we show how this value relates over timesteps across different trajectories. We show the manifold of dynamics of a complete left (with low frequency cue presented) trajectory (dark blue) and complete right trajectory (red) shown along with manifold of dynamics when agent run from start location to choice point with left cue (light blue) and right cue (pink) given at cue point and agent paused in place at the top of the maze stem (as seen in Figure 5). A few timesteps after the agent is paused, the dynamics of the left cue paused agent (light blue) switches manifold path abruptly from running alongside the complete left trajectory path (blue) and joins the right trajectory path (red), following this for many timesteps before ultimately resulting at the same manifold end position as the complete left trajectory manifold path (blue). This is analogous for the right cue paths (red and pink).

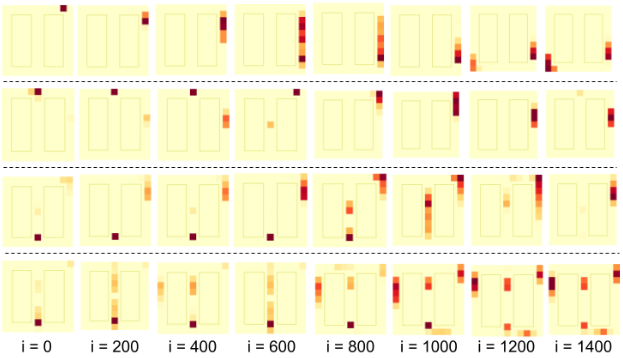

Figure 7: **Place fields of several RNN units drift towards reward positions throughout reward training.** We show place fields of four representative LSTM units (one in each row), starting from $i = 0$ with the pre-trained network (where i is the number of complete maze traversals throughout reward training), drifting forwards towards reward locations throughout reward training as the total number of maze traversals increases. The place fields ultimately rest at maze reward locations at the end of reward training ($i = 1400$).

significantly more RNN units with place fields which move throughout reward training, move to reward locations versus other random locations in the maze. We show that place field movement throughout reward training is not random but significantly preferential toward reward locations ($p < 1.78e{-}38$). This is possibly explained by the gradient of Q values (prediction of predicted reward) spreading backwards from reward locations (Hasselmo, 2005) and becoming stronger throughout training.

3.3 Selectivity of neuronal units

In addition to a forward sweeping representation, this trained network also exhibits neural selectivity that closely matches hippocampal circuits. Griffin et al. (2007) reported that after reward learning, hippocampal neurons were more strongly selective for the encoding or choice phase of a task rather than the direction of the organism's trajectory. We garner the preference of selectivity of each neuronal unit in our network using a discrimination index used by Griffin et al. (2007) for the turn direction selectivity (DI_{turn}) and the phase selectivity (DI_{phase}):

$$DI_{\text{turn}} = \frac{FR_{\text{right}} - FR_{\text{left}}}{FR_{\text{right}} + FR_{\text{left}}} \quad DI_{\text{phase}} = \frac{FR_{\text{cue}} - FR_{\text{choice}}}{FR_{\text{cue}} + FR_{\text{choice}}} \tag{5}$$

where FR_{right} for a particular LSTM unit is the mean firing rate from the cue point on the central stem to the choice point at the top of the stem on trajectories where the agent turns right at the choice point. Similarly FR_{left} is the mean stem firing rate when the agent turns left. FR_{cue} is the firing rate at the cue (encoding) point averaged over both left and right trajectories and similarly FR_{choice} is the averaged firing rate at the choice (sampling) point.

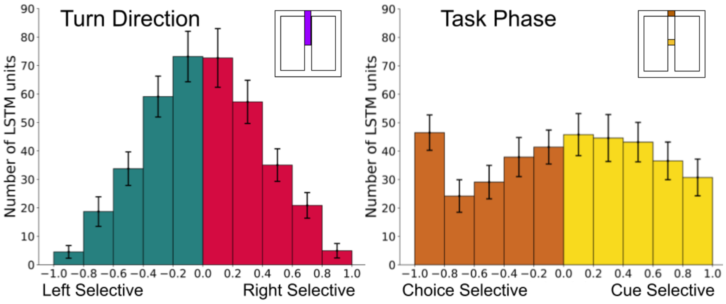

Figure 8: **RNN units are more task selective than direction selective.** Histograms showing LSTM unit discrimination index for turn direction selectivity (DI_{turn}) vs task phase selectivity (DI_{phase}). Counts averaged over 30 trained models, with error bars indicating one standard deviation of error. A highly negative selectivity index for turn direction indicates a neuronal unit which exhibits high levels of selectivity (uniquely high network activity) for a leftward trajectory and a highly positive selectivity index indicates selectivity for a rightward trajectory. A negative selectivity for task phase indicates a neuron which is highly selective for the choice (retrieval) phase of the goal based task whereas a positive index indicates a neuron which is highly selective for the cue (encoding) phase of the task.

The firing areas used for selectivity measurement are insets in Figure 8. We use the stem above the cue point to assess turn direction selectivity, and the cue/choice points to assess encoding and sampling (DI_{phase}). Figure 8 shows a higher proportion of LSTM units are strongly task selective rather than turn selective, with significantly more units having large absolute DI_{phase} indices than DI_{turn} indices.

In addition, the reward trained network is found to have a disproportionately high number of units (163 out of 380 LSTM units) with place fields at the start location of the maze. Moreover, we find evidence of conditional destination encoding in these units which were heavily differentiated in their firing with respect to particular rewarding locations, as shown experimentally in CA1 hippocampal place cells (Ainge et al., 2007; Wood et al., 2000; Ferbinteanu & Shapiro, 2003). 59.5% of units with a place field at the maze start location fired uniquely at this point for rewarding locations on a particular return arm.

4 Discussion

In this work we show that networks trained with a combined predictive and goal-based objective exhibit functional dynamics and selectivity behaviour coinciding with that of hippocampal neurons. We demonstrate that extrafield firing activity of network units emerge when a simulated agent, which is trained on a goal based reward task in a T-shaped maze, pauses at decision points - suggesting intrinsic dynamics are encoding the future trajectory of the agent. This mirrors experimental results in hippocampal place cells in rats (Johnson & Redish, 2007; Frank et al., 2000). At the same time, we find that networks using this combined objective, following exploratory pre-training only on a sensory prediction task, can learn the correct goal-directed behaviour much faster than an equivalent network with only a Q learning objective.

Previous work shows that metric neural representations of environments form when an RNN is optimised to predict agent position from agent velocity (Cueva & Wei, 2018; Banino et al., 2018) and non-metric representations form when an RNN is trained to predict future sensory events given direction of movement (Recanatesi et al., 2019). When training our model we do not provide the LSTM network with any explicit information about location or direction, it only receives sensory information. This is similar to the purely contextual input received by the model pre-trained by Xu & Barak (2020) where no velocity input is given, however, the network used by these authors is still trained on position and landmark prediction in a supervised way.

Instead, our training paradigm forces the LSTM to maintain an implicit notion of movement within its internal state in relation to environmental observations. This, in conjunction with the consideration that model-free RL methods such as Q-learning perform poorly on tasks in dynamic environments such as ours (Dolan & Dayan, 2013), and the long term dependency on the delayed cue in perspective of the choice location, makes the task outlined in Figure 2 challenging. Thus the LSTM is constrained to solve the reward task by storing its current trajectory in its state, as it receives no input concerning its last action or its current direction. This gives rise to a forward looking representation when combined with predictive learning.

Training on a sensory predictive task causes the formation of a non-metric place cell like representation in the activations of network units, similarly to Recanatesi et al. (2019). This allows relatively fast learning of the cue-reward task when using a combined predictive and RL loss - this is due to the network being able to localise itself even in the absence of any location or direction based input.

These pre-trained LSTM units demonstrate nonlocal extrafield firing after reward training (Figure 4). We observe that cue or choice point extrafield activity is evident in most LSTM units after training on the reward task. This is likely due to the increased precedence these points have in the agent reaching reward locations. Together the trained LSTM network units form a representation which sweeps along the paths available to the agent, first down the reward path and then the other, as shown in Figure 5 and demonstrated in rats in Johnson & Redish (2007). The sweeping dynamics of the representation may arise due to the lack of a second cue as the agent is paused at the choice point.

Although hippocampal place cells are critical for spatial memory (Nakazawa et al., 2002; Florian & Roullet, 2004; Sandi et al., 2003; Redish & Touretzky, 1998; Miller et al., 2020), it is unclear by what mechanism an ensemble of place cells contributes to a representation of goal-directed behaviour. Our model and training paradigm is in keeping with the hypothesis that the hippocampus is involved in maintaining a conjunctive representation of cognitive maps and sensory information (Whittington et al., 2019). We show that this paradigm can be extended with predictive learning of Q-values of anticipated future reward, and show that the resulting representation is well suited for learning actions leading from a cue to a reward. Importantly, this representation emerges solely from sampling sensory inputs and predicted rewards, while reinforcement learning itself remains model-free and is initially random. The surprising similarity of the task-dependent activity in our simulations and experimentally recorded neural activity in similar tasks suggests that the model may replicate central aspects of learning and planning in the hippocampus. Our trained model could improve understanding of hippocampal function by testing hypotheses regarding previously unobserved dynamics inexpensively. This could be performed on maze environments such as this work, or more open arena settings once the model is retrained.

A limitation of our model is that it has no generative component from which to sample observations. Future work would focus on an agent capable of sampling future trajectories in order to truly plan its upcoming actions in a complex environment.

References

James A. Ainge, Minija Tamosiunaite, Florentin Woergoetter, and Paul A. Dudchenko. Hippocampal CA1 place cells encode intended destination on a maze with multiple choice points. *Journal of Neuroscience*, 27(36), 2007. ISSN 02706474. doi: 10.1523/JNEUROSCI.2011-07.2007.

Andrea Banino, Caswell Barry, Benigno Uria, Charles Blundell, Timothy Lillicrap, Piotr Mirowski, Alexander Pritzel, Martin J. Chadwick, Thomas Degris, Joseph Modayil, Greg Wayne, Hubert Soyer, Fabio Viola, Brian Zhang, Ross Goroshin, Neil Rabinowitz, Razvan Pascanu, Charlie Beattie, Stig Petersen, Amir Sadik, Stephen Gaffney, Helen King, Koray Kavukcuoglu, Demis Hassabis, Raia Hadsell, and Dharshan Kumaran. Vector-based navigation using grid-like representations in artificial agents. *Nature*, 2018. ISSN 14764687. doi: 10.1038/s41586-018-0102-6.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2014. ISBN 9781937284961. doi: 10.3115/v1/d14-1179.

Christopher J. Cueva and Xue Xin Wei. Emergence of grid-like representations by training recurrent neural networks to perform spatial localization. In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR, 2018.

Christopher J. Cueva, Peter Y. Wang, Matthew Chin, and Xue-Xin Wei. Emergence of functional and structural properties of the head direction system by optimization of recurrent neural networks. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=HklSeREtPB`.

Ray J. Dolan and Peter Dayan. Goals and habits in the brain, 2013. ISSN 08966273.

Janina Ferbinteanu and Matthew L. Shapiro. Prospective and retrospective memory coding in the hippocampus. *Neuron*, 40(6), 2003. ISSN 08966273. doi: 10.1016/S0896-6273(03)00752-9.

Cédrick Florian and Pascal Roullet. Hippocampal CA3-region is crucial for acquisition and memory consolidation in Morris water maze task in mice. *Behavioural Brain Research*, 154(2), 2004. ISSN 01664328. doi: 10.1016/j.bbr.2004.03.003.

Loren M. Frank, Emery N. Brown, and Matthew Wilson. Trajectory encoding in the hippocampus and entorhinal cortex. *Neuron*, 27(1), 2000. ISSN 08966273. doi: 10.1016/S0896-6273(00)00018-0.

Marianne Fyhn, Sturla Molden, Menno P. Witter, Edvard I. Moser, and May Britt Moser. Spatial representation in the entorhinal cortex. *Science*, 2004. ISSN 00368075. doi: 10.1126/science.1099901.

Amy L. Griffin, Howard Eichenbaum, and Michael E. Hasselmo. Spatial representations of hippocampal CA1 neurons are modulated by behavioral context in a hippocampus-dependent memory task. *Journal of Neuroscience*, 27(9), 2007. ISSN 02706474. doi: 10.1523/JNEUROSCI.4083-06.2007.

Michael E. Hasselmo. A model of prefrontal cortical mechanisms for goal-directed behavior. *Journal of Cognitive Neuroscience*, 17(7), 2005. ISSN 0898929X. doi: 10.1162/0898929054475190.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 1997. ISSN 08997667. doi: 10.1162/neco.1997.9.8.1735.

Dan Hu and Abram Amsel. A simple test of the vicarious trial-and-error hypothesis of hippocampal function. *Proceedings of the National Academy of Sciences of the United States of America*, 92 (12), 1995. ISSN 00278424. doi: 10.1073/pnas.92.12.5506.

Adam Johnson and A. David Redish. Neural ensembles in CA3 transiently encode paths forward of the animal at a decision point. *Journal of Neuroscience*, 27(45), 2007. ISSN 02706474. doi: 10.1523/JNEUROSCI.3761-07.2007.

Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.

Inah Lee, Amy L Griffin, Eric A Zilli, Howard Eichenbaum, and Michael E Hasselmo. Gradual Translocation of Spatial Correlates of Neuronal Firing in the Hippocampus toward Prospective Reward Locations. *Neuron*, 51(5):639–650, 2006. ISSN 0896-6273. doi: https://doi.org/10.1016/j.neuron.2006.06.033. URL `http://www.sciencedirect.com/science/article/pii/S0896627306005836`.

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. UMAP: Uniform Manifold Approximation and Projection. *Journal of Open Source Software*, 3(29), 2018. ISSN 2475-9066. doi: 10.21105/joss.00861.

Thomas D. Miller, Trevor T.J. Chong, Anne M.Aimola Davies, Michael R. Johnson, Sarosh R. Irani, Masud Husain, Tammy W.C. Ng, Saiju Jacob, Paul Maddison, Christopher Kennard, Penny A. Gowland, and Clive R. Rosenthal. Human hippocampal CA3 damage disrupts both recent and remote episodic memories. *eLife*, 9, 2020. ISSN 2050084X. doi: 10.7554/eLife.41836.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning, 2013.

Karl F. Muenzinger. Vicarious Trial and Error at a Point of Choice: I. A General Survey of its Relation to Learning Efficiency. *Pedagogical Seminary and Journal of Genetic Psychology*, 53(1), 1938. ISSN 08856559. doi: 10.1080/08856559.1938.10533799.

Kazu Nakazawa, Michael C. Quirk, Raymond A. Chitwood, Masahiko Watanabe, Mark F. Yeckel, Linus D. Sun, Akira Kato, Candice A. Carr, Daniel Johnston, Matthew A. Wilson, and Susumu Tonegawa. Requirement for hippocampal CA3 NMDA receptors in associative memory recall. *Science*, 297(5579), 2002. ISSN 00368075. doi: 10.1126/science.1071795.

J. O'Keefe and L. Nadel. *The hippocampus as a cognitive map*. Clarendon Press, Oxford, United Kingdom, 1978.

Stefano Recanatesi, Matthew Farrell, Guillaume Lajoie, Sophie Deneve, Mattia Rigotti, and Eric Shea-Brown. Signatures of low-dimensional neural predictive manifolds. *Cosyne Abstracts 2019, Lisbon, PT.*, 2019.

A. David Redish and David S. Touretzky. The Role of the Hippocampus in Solving the Morris Water Maze. *Neural Computation*, 10(1), 1998. ISSN 08997667. doi: 10.1162/089976698300017908.

Carmen Sandi, Heather A. Davies, M. Isabel Cordero, Jose J. Rodriguez, Victor I. Popov, and Michael G. Stewart. Rapid reversal of stress induced loss of synapses in CA3 of rat hippocampus following water maze training. *European Journal of Neuroscience*, 17(11), 2003. ISSN 0953816X. doi: 10.1046/j.1460-9568.2003.02675.x.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double Q-Learning. In *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, 2016. ISBN 9781577357605.

Christopher J. C. H. Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8(3-4), 1992. ISSN 0885-6125. doi: 10.1007/bf00992698.

James CR Whittington, Timothy H. Muller, Shirley Mark, Guifen Chen, Caswell Barry, Neil Burgess, and Timothy EJ Behrens. The Tolman-Eichenbaum Machine: Unifying space and relational memory through generalisation in the hippocampal formation. *bioRxiv*, pp. 770495, September 2019. doi: 10.1101/770495.

Emma R. Wood, Paul A. Dudchenko, R. Jonathan Robitsek, and Howard Eichenbaum. Hippocampal neurons encode information about different types of memory episodes occurring in the same location. *Neuron*, 27(3), 2000. ISSN 08966273. doi: 10.1016/S0896-6273(00)00071-4.

Tie Xu and Omri Barak. Implementing inductive bias for different navigation tasks through diverse rnn attrractors. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=Byx4NkrtDS`.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Discussion

(c) Did you discuss any potential negative societal impacts of your work? [N/A]

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...

   (a) Did you state the full set of assumptions of all theoretical results? [N/A]

   (b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments...

   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] In Supplemental Material

   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] In Methods Section

   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Figures 3 and 8

   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] In Supplemental Material - Just CPU

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

   (a) If your work uses existing assets, did you cite the creators? [N/A]

   (b) Did you mention the license of the assets? [N/A]

   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Code

   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

3.3 Discussion

In this chapter I introduced a modelling approach which first trains an RNN on a predictive task, with the aim of learning random trajectories through a maze environment, based on the wall colours of said maze. From this pre-training, I find that the RNN units are spatially modulated, with firing preferences of individual RNN units dictated by specific positions in the maze. Once the RNN has learned the environment, the RNN is tasked with controlling an artificial agent and guiding it towards stochastic reward locations in the maze. The RNN learns this task much more quickly with the environmentally trained RNN than with a tabula rasa RNN. Furthermore, I show that training the RNN on the reward task alongside the predictive task increases the speed of convergence on the reward task.

Once training on the reward task has converged, I show that the RNN units have firing responses akin to those found in rodent place cells after having learned a similar navigation task. The numerous resemblances found could be indicative of the utility in using artificial neural networks in modelling biological neural systems.

Although these recapitulations are noteworthy, the virtual environment used is somewhat simplistic and it remains to be seen whether similar representations will emerge on larger, more complicated and open environments. Furthermore, despite the place representations which emerge in the units of the RNN, can these actually be used to effectively model the mammalian brain? Could further results from an artificial neural network predicate those to be eventually found empirically in a biological network? Much recent work has recapitulated experimental findings [13, 14, 5, 71, 87, 81], however there is currently no work where artificial neural network models have successfully predicted experimental findings.

Another potential unknown is whether the replay shown in section 3.2 of the paper above is a result of repeated training on the same trajectories or an actual innate planning mechanism of the network. If a result of training, is this also the case in biology? Does the biological forward moving representation found in [39] come about through explicit planning or simply through repetition? Moreover, the representation of the maze environment learned by the RNN clearly helps with navigation training but is this the best auxiliary task to pre-train the network on? Could some other pre-training task other than subsequent maze wall colour prediction be more effective in learning a foundational representation for downstream tasks? Future work should be focused on answering these questions.

3.3.1 RNN representation with backtracking in task solution

In contrast to Figure 5 in the paper where the navigation task is solved without any backtracking, here we show how the RNN's representation behaves when the navigation task is solved with the agent often backtracking at secondary choice points in the maze before moving to the correct reward locations. This backtracking is achieved by either increasing the learning rate from 0.0002 to a higher value or increasing the discounting factor γ in Q-learning (see Equation 2 in the paper) from 0.8 to a higher value. Here we use a learning rate of 0.001 and a discount factor of 0.9 (but only one of these is required to cause backtracking).

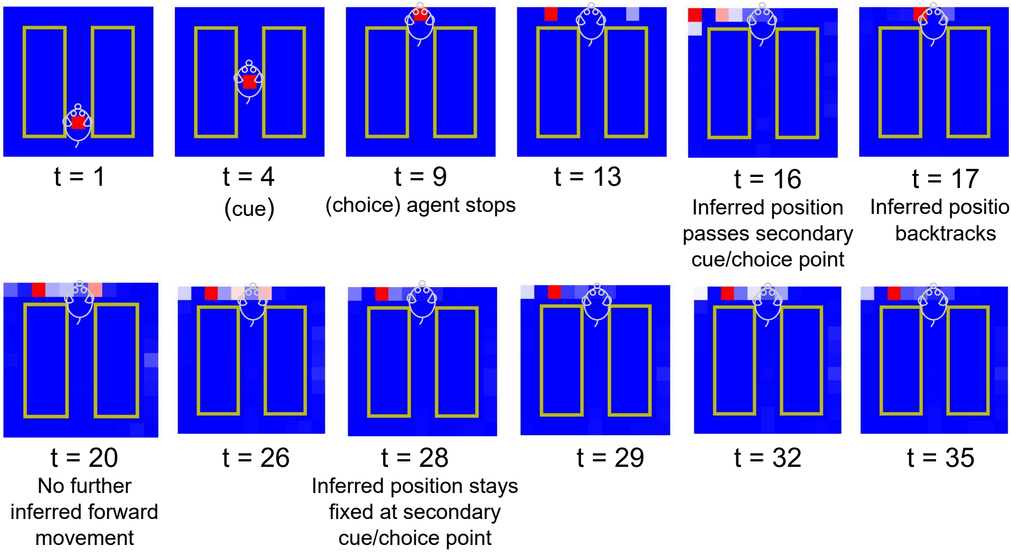
t = 1
t = 4
(cue)
t = 9
(choice) agent stops
t = 13
t = 16
Inferred position passes secondary cue/choice point
t = 17
Inferred position backtracks
t = 20
No further inferred forward movement
t = 26
t = 28
Inferred position stays fixed at secondary cue/choice point
t = 29
t = 32
t = 35

Figure 3.1: **Backtracking causes RNN inferred position of the agent to move to secondary cue/choice point and remain at this position**. We run the agent from the start position to the top of the stem of the maze at timestep 9 with a low frequency (left) cue tone at timestep 4. The agent is stopped at this position with the RNN receiving the environment observation from this position for the remainder of the shown timesteps. The RNN representation (inferred agent location) moves in the direction corresponding to the frequency of the given cue tone. Then between timesteps 13 and 17, the inferred position moves between the primary and secondary choice points slightly, finally resting at the secondary choice point for the remainder.

As seen in Figure 3.1, when backtracking is present in the model's solution for the navigation task, the forward movement of the RNN's representation while the agent is paused is minimal, with the uncertainty caused by backtracking resulting in the inferred agent position resting at the secondary cue/choice point instead of switching maze arms and returning to the maze start position, as is the case when backtracking is not present in the task solution (Figure 5 in the paper).

This difference in movement of RNN representation is due to the uncertainty caused by backtracking of the agent being inherent at the secondary cue/choice points. This backtracking means that when the agent is paused at the top of the maze stem and the RNN representation moves ahead of the agent, reaching the secondary cue/choice ”position”, the lack of secondary cue being presented to the RNN causes ambiguity as to which secondary point (left or right) the RNN believes the agent is at. In the above example, with a low frequency cue given to the agent and reward sites active on the left side of the maze, the agent should turn left at the primary choice point. The RNN representation inferred agent location does move in this direction, but does not move past the left secondary cue/choice point. The lack of secondary cue after 4 timesteps causes the RNN to believe the agent is at the secondary cue/choice point on the right side of the maze, but this is not reconciled with the RNN’s inferred location. This leads to the RNN representation remaining stationary at the left secondary cue/choice point.

Chapter 4

Robust alignment of cross-session recordings of neural population activity by behaviour via unsupervised domain adaptation

In this chapter I introduce a modelling approach which successfully unifies the latent representations of many sessions of intracortical neural recordings. I have already discussed how neural population activity relating to behaviour is assumed to be inherently low-dimensional despite the observed high dimensionality of data recorded using multi-electrode arrays [12, 18, 23, 31, 59, 75]. Therefore, predicting behaviour from neural population recordings has been shown to be most effective when using latent variable models [37]. The primary issue with this however is that the latent representations of trials of differing recording sessions of neural activity diverge considerably, requiring the retraining or recalibration of behaviour decoders. The only way to avoid this is to form a latent representation which is robust to the considerable neural drift and variation (such as probe movement) present between recording sessions.

[22] have shown that latent representations of trials formed using PCA from recording sessions months apart can be reconciled using straightforward linear transformations. Therefore the drift and non-stationaries between many recording sessions could be learnable, with a latent variable model capturing the variability between these sessions. The model could interpolate or extrapolate this learned variability to a completely unseen session chronologically close to these learned sessions. [38] show that it is advantageous to separate the latent space of Latent Factor Analysis via Dynamical

Systems (LFADS) [68] into behaviourally relevant and behaviourally irrelevant subspaces. This shows that training the model to reconstruct spike trains as well as predict behaviour results in a more disentangled latent space.

In this work, I aim to produce a session invariant decoder of intracortical neural activity from monkeys performing a centre-out reach task [23]. I show that applying an adversarial loss on neural reconstruction between the encoder and decoder of a sequential autoencoder based on LFADS causes the latent space to become invariant to recording session. Simultaneously, behaviour is decoded from the same latent space, causing trial disentanglement by behaviour. This results in a latent space which is well separated by behaviour (in this case movement direction) but with no separation by recording session. The model therefore importantly captures inter-session variability, successfully creating a robust latent space.

As a result, the model generalises effectively to a completely unseen session (held out from training), both chronologically in between training sessions and chronologically after training sessions, with high decoding accuracy. Notably, this requires no further retraining of the model, as is the case with all previous methods. To my knowledge, no prior model is capable of decoding from a held out session with high accuracy. This approach is similar to [24] and [26] who use domain adaptation by backpropagation to train a domain invariant encoder.

4.1 Assessing the efficacy of SABLE on synthetic data

Prior to testing our model on real neural data recorded from the motor cortex of two monkeys performing a centre-out reach task, we first assess the performance of our model (SABLE) by applying it to synthetic data generated from a deterministic nonlinear Lorenz system (Figure 4.1). This system is a set of nonlinear equations for three dynamic variables. It is widely used [68] to generated synthetic spiking data due to its limited dimensionality, which allows its entire state space to be visualized. We simulated a population of 50 neurons with firing rates given by randomly weighted linear read-outs of the Lorenz variables as they evolve, followed by an exponential nonlinearity. Spikes from these firing rates were then generated by a Poisson process.

We generated trials by starting the Lorenz system with a 8 sets of randomly chosen initial conditions and running the system for 1 second. We slightly vary the weights of the linear read-out for trials within each session and randomly sample completely new weights for the linear read out of the Lorenz variables in each session (to simulate

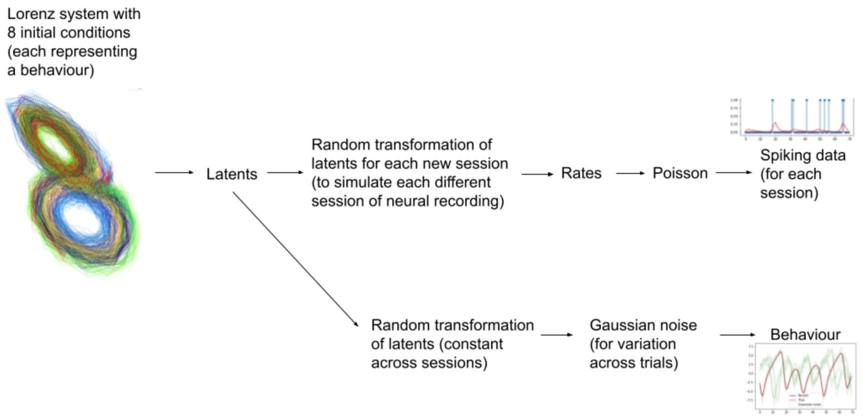
Lorenz system with 8 initial conditions (each representing a behaviour)
Latents
Random transformation of latents for each new session (to simulate each different session of neural recording)
Rates
Poisson
Spiking data (for each session)
Random transformation of latents (constant across sessions)
Gaussian noise (for variation across trials)
Behaviour

Figure 4.1: Lorenz system used to generate a synthetic dataset to represent multiple sessions of recording while performing consistent behaviours.

variability across recording sessions in non-human primates). We generate a total of 160 trials for each synthetic session, 20 for each set of Lorenz initial conditions (each corresponding to a different behaviour).

To simulate behaviour corresponding to this synthetic spiking data, we apply a separate randomly sampled linear read-out of the Lorenz variables which we keep constant across trials and sessions, with some Gaussian noise applied to the read out values (to simulate variation in behaviour across trials). Using SABLE, we aim to predict this corresponding behaviour from the synthetic spiking data. SABLE was trained using varying numbers of these synthetic sessions (19, 9 and 6) and tested on a single completely held out synthetic session. We hypothesise that our model can generate latent variables for each trial such that the latent space is separated by set of Lorenz initial conditions (corresponding to each behaviour), with no separation by session. If this disentanglement occurs and the SABLE encoder learns to model the variability across sessions, stable behaviour decoding should follow.

As shown in Figure 4.2, training SABLE on 9 and 19 synthetic training sessions leads to a robust latent space, with trials separated well by behaviour (set of Lorenz initial conditions) and not separated by training session. We also observe high behaviour decoding accuracy (R^2) when testing on a held-out session when training with 9 or 19 sessions. 6 sessions seems to contain too little overall variability for the model to generalise to trials from the unseen synthetic session, therefore we see little overlap of trials from the held-out test session and trials from the training sessions in the latent space (Figure 4.2). From these results on synthetic data, we hypothesise that SABLE

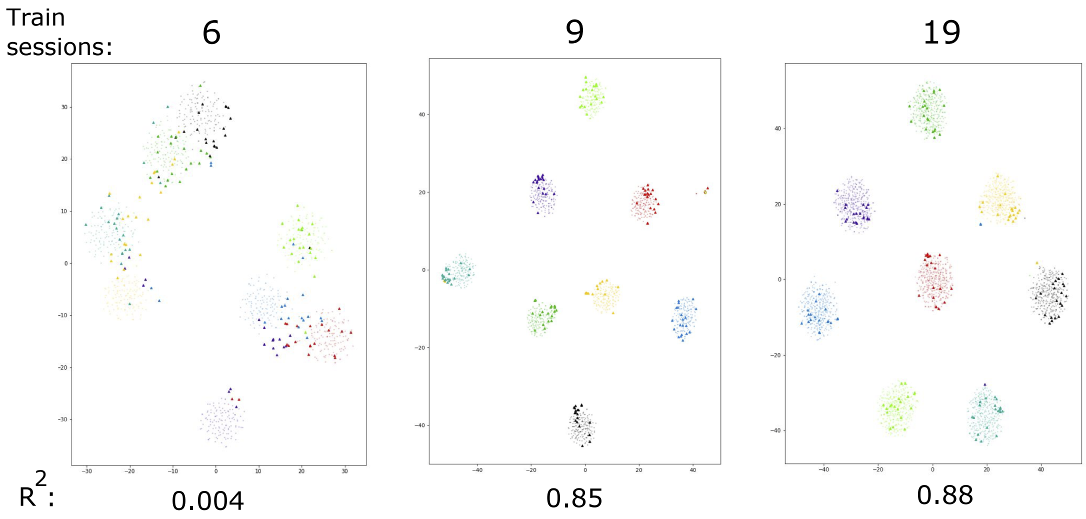

Figure 4.2: t-SNE dimensionality reduced latent space of SABLE when applied to an increasing number of synthetic training sessions and testing on a single unseen synthetic test session. Dots in each latent space indicate trials from synthetic training sessions while triangles indicate trials from a held-out session, with each colour indicating a distinct set of Lorenz initial conditions. *Above:* Number of training sessions we train SABLE on to produce corresponding latent space. *Below:* R-squared decoding accuracy of SABLE predicted behaviour vs. ground truth synthetic behaviour.

will be effective in generalising to trials from an unseen session of real neural recordings when trained on trials from a sufficient number of recording sessions to capture inter-session variability.

4.2 Contribution

I am the first author and lead of this work. As such, I conceptualised the model, implemented all versions of the model, ran and evaluated the methods, and wrote the manuscript along with Matthias Hennig. Matthew Perich and Lee Miller provided the dataset used to train and evaluate the model.

4.3 Paper

Robust alignment of cross-session recordings of neural population activity by behaviour via unsupervised domain adaptation

Justin Jude[1] **Matthew G. Perich**[2] **Lee E. Miller**[3] **Matthias H. Hennig**[1]

Abstract

Neural population activity relating to behaviour is assumed to be inherently low-dimensional despite the observed high dimensionality of data recorded using multi-electrode arrays. Therefore, predicting behaviour from neural population recordings has been shown to be most effective when using latent variable models. Over time however, the activity of single neurons can drift, and different neurons will be recorded due to movement of implanted neural probes. This means that a decoder trained to predict behaviour on one day performs worse when tested on a different day. On the other hand, evidence suggests that the latent dynamics underlying behaviour may be stable even over months and years. Based on this idea, we introduce a model capable of inferring behaviourally relevant latent dynamics from previously unseen data recorded from the same animal, without any need for decoder recalibration. We show that unsupervised domain adaptation combined with a sequential variational autoencoder, trained on several sessions, can achieve good generalisation to unseen data and correctly predict behaviour where conventional methods fail. Our results further support the hypothesis that behaviour-related neural dynamics are low-dimensional and stable over time, and will enable more effective and flexible use of brain computer interface technologies.

1. Introduction

In the brain, stimuli and behaviour can be decoded from the activity of populations of neurons, and it is well established that correlations or co-variations between neurons are a key ingredient in neural population codes (Saxena & Cunningham, 2019). There has been considerable success developing methods for decoding external variables from recordings of even modestly sized populations of 10s or 100s of neurons (Hurwitz et al., 2021a), raising hopes that brain computer interfaces (BCIs) can be an effective assistive technology for severely disabled patients. However, a decoder, once trained, requires stable recordings to perform well. Over the course of days and weeks, the signals recorded from implanted extracellular probes will inevitably change and drift due to factors such as impedance changes, gliosis and probe and brain movement (Chestek et al., 2011). Non-invasive systems such as electromyography (EMG) sensors will not be worn permanently and positioned slightly differently every time, creating even stronger variations in recorded signals. Moreover, the activity of individual neurons can change considerably over similar time scales due to neural plasticity (Rule et al., 2019). Together these fluctuations will lead to degradation of decoder performance over time, thus to be effective, frequent recalibration of BCI systems would be inevitable.

Given the limited long-term stability of recorded neural signals, reports of relatively stable behaviour decoding over days with the same decoder may seem surprising (Chestek et al., 2007). Recent work by Gallego et al. (2020) however showed that some aspects of the population activity of cortical neurons remain very stable even over months and years. Specifically, this study showed that neural population activity in the primary motor cortex is highly restricted to and evolves along a low-dimensional manifold that is stable even when single neuron activity constantly fluctuates.

Low-dimensional neural dynamics can be effectively extracted from neural population activity with latent variable models (Hurwitz et al., 2021a). These models use an often small number of latent variables (or factors) together with an appropriate observation model that relates latent variables to the recorded activity. Importantly, the latent variables in such models often predict stimuli or behaviour very well even when they were only optimised to reproduce neural activity (Hurwitz et al., 2021a). Nonlinear state space models such as LFADS are particularly powerful in predicting

[1] School of Informatics, University of Edinburgh, Edinburgh, Scotland, EH8 9AB [2] Université de Montréal and Mila, Montréal, QC, Canada H3C 3J7 [3] Feinberg School of Medicine, Northwestern, Chicago, IL 60611. Correspondence to: Justin Jude <justin-jude@me.com>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

single trial activity and behaviour in test data (Pandarinath et al., 2018).

Therefore, instabilities in neural recordings can be successfully compensated for by re-training the part of a model that translates neural activity into the latent dynamics, which are assumed stable over time (Farshchian et al., 2019; Dabagia et al., 2022; Degenhart et al., 2020; Wen et al., 2021; Herrero-Vidal et al., 2021). As a behaviour decoder, this can be more data-efficient than re-training a decoder from scratch, but still requires regular interventions. Here we ask if it is possible to recover the latent dynamics without any re-training.

Our approach uses a domain adaptation inspired solution. Sources of session to session variability in neural recordings are shown in Figure 1 and include existing neurons lost from recording electrodes, existing neurons replaced by unseen neurons, and all recording electrodes shifting systematically due to probe array shift. This variability is substantial enough that each recording session can be constituted as a separate domain.

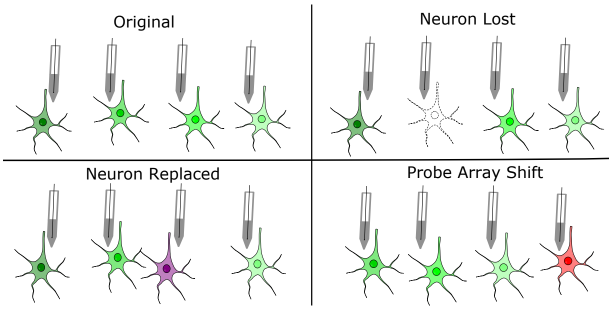

Figure 1. Causes of session-to-session variability in recordings from neural populations. Neurons from the original recording session can be lost from the recording array, original neurons can be replaced by unseen neurons and the entire probe array can shift, causing a systematic change in neuron position. In addition, spike sorting can induce variability as the signal to noise ratio of individual neurons changes between sessions. Domain adaptation of many varied sessions enables our model to learn these sources of variability.

We treat each recording session as a separate domain, each of which can be used to predict the same set of behaviours. The model is optimised using both recorded activity and behaviour to recover the same latent variables irrespective of the domain so it is capable of predicting behaviour correctly for a previously unseen session without need for re-calibration. In contrast, latent variable models without domain adaptation fail to generalise to unseen data, and instead partition the latent space into distinct parts corresponding to the individual recording sessions. We test this model with long-term recordings from the primate motor cortex during a reach task and show that, provided sufficient training data, it can predict behaviour well for previously unseen sessions. BCI decoders that can generalise well to unseen sessions or subjects without any re-training have not yet been demonstrated. We believe this is the first work to show such cross-session decoder generalisation without recalibration.

2. Related Work

This issue of neural stability is investigated by Gallego et al. (2020) where the dynamics of a set of a single animal's M1 cortex neurons are recorded from over many days. The authors find that the underlying dynamics of these neurons over time are indeed reconcilable. Principal component analysis (PCA) is used to reduce the dimensionality of the neural activity on each day, and these variables are then aligned using canonical correlation analysis (CCA). After alignment, neural activity is regenerated for up to 16 days with close similarity and accurate decoding of behaviour.

Farshchian et al. (2019) take this approach a step further and utilise an adversarial approach with a non-linear model (ADAN) to directly align neural activity over many days in order to accurately predict EMG during movement. A discriminator network is trained in a similar fashion to LFADS, tasked with autoencoding neural activity from day 0. A generator neural network is optimised to align neural population activity to that recorded at day 0. The autoencoding discriminator is tasked with maximising the alignment loss.

Sussillo et al. (2016) build a robust decoder capable of utilising large amounts of training data and maintaining decoding performance in the face of recording condition changes such as neuron turnover. Herrero-Vidal et al. (2021) introduces a robust probabilistic approach for neural alignment in a common low dimensional manifold. Their method is applied to recordings from the mouse olfactory bulb, revealing low-dimensional population dynamics that are odour specific and have consistent structure across animals. Following the same idea, Wen et al. (2021) uses adversarial generative modelling to generate large amounts of synthetic spike data from just the behaviour of a separate recording session or subject, mimicking the spike data of that session/subject. Together with this generated synthetic spike data and a small amount of real spike data from the unseen session, the authors are able to achieve relatively good behaviour decoding accuracy on the held out session. This model is more data efficient than the previously mentioned approaches. Nevertheless, in all these cases data from all recording sessions or animals is required for good behaviour decoding, and the models are not capable of generalisation to unseen sessions.

Hurwitz et al. (2021b) combines ideas from Pandarinath et al. (2018) and Sani et al. (2021) to jointly model the neural activity and external behavioural variables by separating the latent space into behaviourally relevant and behaviourally irrelevant components; the relevant dynamics are used to reconstruct the behaviour through a flexible linear decoder and both sets of dynamics are used to reconstruct the neural activity through a linear decoder with no time lag. This work shows that an LFADS-like model can jointly model neural activity and associated behaviour or movement, hence potentially isolate invariant behaviour-related dynamics that can be used for cross-session decoding.

Domain adaptation broadly aims to predict classes from labelled data of a similar nature, albeit from differing sources or domains. The method relevant to this work is by Ganin & Lempitsky (2015), who use a negative gradient between a domain classifier and feature extractor in order to coerce the feature extractor to produce domain invariant features from which a label predictor can infer data classes reliably. This method of domain unification is unsupervised.

An application of domain adaptation to correct for variability in experimental data by Gonschorek et al. (2021) used an autoencoder model and a domain classifier to align two-photon imaging data across experiments. The authors successfully align their recording sessions but they do not test efficacy on unseen sessions. They also explicitly use experimental session ID as domains and show efficacy on non-sequential data in this respect. In this work we show that it is beneficial to not explicitly use session/experiment ID for domain adaptation but to instead use neural patterns directly to align recording sessions for high dimensional sequential data.

In this work we model each session of neural recording as a separate domain and predict behaviour from all of these sessions simultaneously. Domain-invariant latent variables are obtained using the paradigm of unsupervised domain adaptation via a negative gradient, which are then optimised to reconstruct the observed behaviour.

3. Model

This model is based on the hypothesis that behaviour y is encoded in a stable latent space with variables z, and that the two are related linearly as $y = f(z)$. Equally, neural activity x is related to the latent variables through a simple function, and as in related models we choose a linear read-out with a Poisson link function to generate non-negative firing rates (Pandarinath et al., 2018). However, this function will differ between recording sessions (or domains) d as we expect to observe different neurons in each session, and the activity of neurons may change over time. The problem is thus to find the correct encoding function $z = g(x)$ to transform neural activity into the latent space which then allows decoding of behaviour. As explained above, re-training this part of the model for each session can successfully align different sessions. Here we show that this can be achieved without the need for re-training.

Specifically, as proposed by Pandarinath et al. (2018) we assume that the latent dynamics evolve autonomously provided a set of initial conditions z_i that are modelled as Gaussian random variables. These latent variables are produced for each trial by an encoder network consisting of bidirectional Gated Recurrent Units (Cho et al., 2014) (GRU). They are used to simultaneously predict behaviour, and to reconstruct the original trial-specific neural activity. We apply recurrent and kernel regularisation to the encoder GRU to enable better generalisation to unseen sessions.

A further bidirectional GRU is used as a decoder for neural reconstruction and a final separate GRU is used to predict behaviour from the generated latent variables. Training is based on a mean squared error loss for behaviour and Poisson likelihood for neural activity. Importantly, we reverse the backpropagation gradient between the neural reconstruction decoder and the encoder. This gradient reversal layer leads to maximisation of the neural reconstruction loss in the encoder network while, at the same time, the neural decoder network is adversarially optimised to minimise neural reconstruction loss. This implicitly encourages the encoder to generate latent variables which are not separated by session of data collection.

The behaviour decoder meanwhile forces the encoder to generate latent variables which are differentiated by behaviour. Ultimately, this produces a latent space separable by behaviour but not by session of data collection. The complete model is illustrated in Figure 2.

The model is trained using real neural activity which corresponds to consistent behaviours (movement directions in a centre-out reach task, see below). The generative process of our model is as follows:

$$z_i = W_{enc}(\text{GRU}_{\theta_{enc}}(x_{i,1:T})), \quad (1)$$

$$g_{1:T} = \text{GRU}_{\theta_{dec}}(z_i), \quad (2)$$

$$b_{1:T} = \text{GRU}_{\theta_{beh}}(z_i), \quad (3)$$

$$r_t = exp(W_{rate}(W_{fac}(g_t))), \quad (4)$$

$$\bar{x}_t \sim \text{Poisson}(r_t), \quad (5)$$

$$\bar{y}_t = W_{beh}(b_t) \quad (6)$$

where i indicates a particular trial and T is the total number of timesteps per trial. θ_{enc}, θ_{dec}, θ_{beh} are the parameters of the GRUs used to encode spike trains into latent variables, decode spike trains from the generated latent variables, and to decode behaviour from the latent variables respectively.

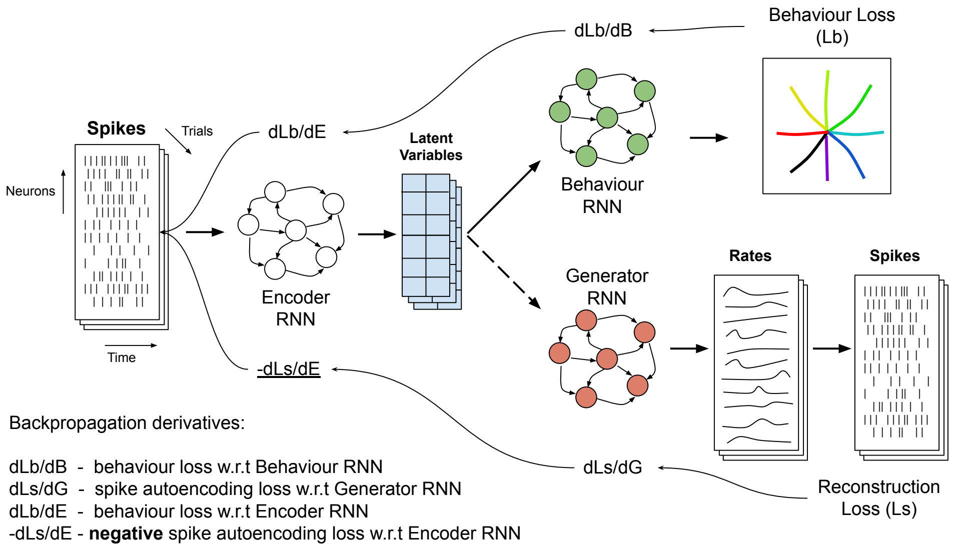
dLb/dB
Behaviour Loss
(Lb)
Spikes
Trials
Neurons
Time
dLb/dE
Latent
Variables
Encoder
RNN
Behaviour
RNN
Generator
RNN
Rates
Spikes
-dLs/dE
dLs/dG
Reconstruction
Loss (Ls)
Backpropagation derivatives:
dLb/dB - behaviour loss w.r.t Behaviour RNN
dLs/dG - spike autoencoding loss w.r.t Generator RNN
dLb/dE - behaviour loss w.r.t Encoder RNN
-dLs/dE - **negative** spike autoencoding loss w.r.t Encoder RNN

Figure 2. Our model (SABLE) consists of a sequential variational autoencoding approach combined with a sequential behaviour decoder. Notably, we implement a reverse gradient layer between the neural decoder and encoder GRUs. The encoder can then learn to extract the invariant latent dynamics determining behaviour from data obtained in different sessions with variability in the recorded neural activity.

W_{enc}, W_{fac}, W_{rate} and W_{beh} are non-linear layers which produce latent variables, neural activity factors, generate firing rates and predict behaviour respectively at each time step per trial.

At each training iteration the following three losses are optimised using Adam (Kingma & Ba, 2015) asynchronously:

$$L_{rec} = -\sum_{t=1}^{T} \log(\text{Poisson}(x_{i,t}|r_t)) \tag{7}$$

$$L_{beh} = \frac{1}{T}\sum_{t=1}^{T}(y_{i,t} - \bar{y}_{i,t})^2 \tag{8}$$

$$\begin{aligned} L_{kl} &= D_{KL}[\text{GRU}_{\theta_{enc}}(z_i|x_i)||\mathcal{N}(0,I)] \\ &= -\frac{1}{2}[\log(z_{i,\sigma}^2) - z_{i,\mu}^2 - z_{i,\sigma}^2 + 1] \end{aligned} \tag{9}$$

where i indicates a particular trial, T is the total number of timesteps per trial, y_i is the true behaviour per trial and $\bar{y}_i$ is the predicted behaviour. The loss in Eq. 7 is maximised by the encoder network and minimised by the neural decoder network (and not applicable to the behaviour decoder network). This adversarial training is the most crucial aspect of our model. As the encoder maximises the neural reconstruction loss throughout training, it produces increasingly spike pattern-invariant latent variables.

Behaviour loss (Eq. 8) is minimised by both the encoder and behaviour decoding network while the Kullback–Leibler (KL) divergence loss (Eq. 9) (between a multivariate standard Gaussian distribution and the encoder generated latent variables) is minimised by just the encoder network. Thus the total error for all parameters in the model across all training trials can be summarised as:

$$\begin{aligned} &E(\theta_{enc}, W_{enc}, \theta_{dec}, W_{fac}, W_{rate}, \theta_{beh}, W_{beh}) = \\ &\sum_{i=1}^{N}\Big(L_{beh}^{i}(\theta_{enc}, W_{enc}, \theta_{beh}, W_{beh}) \\ &+ L_{rec}^{i}(\theta_{dec}, W_{fac}, W_{rate}) \\ &+ \lambda_{kl}L_{kl}^{i}(\theta_{enc}, W_{enc})\Big) - \lambda_r\sum_{i=1}^{N}L_{rec}^{i}(\theta_{enc}, W_{enc}) \end{aligned} \tag{10}$$

where N is the total number of training trials, λ_{kl} is the weight of KL divergence and λ_r is the weight of the reverse gradient applied to the encoder RNN. λ_{kl} rises exponentially as training progresses while λ_r decays exponentially (thereby increasing session invariance over training). We denote our model Stable Alignment of Behaviour through spike-invariant Latent Encoding (SABLE).

SABLE does not require specific hyperparameter tuning for either monkey tested in Section 6, however, in subjects or experimental setups where neural drift is more variable between recording sessions, tuning recurrent dropout and kernel regularisation values may be beneficial for optimal behaviour decoding performance from unseen session trials.

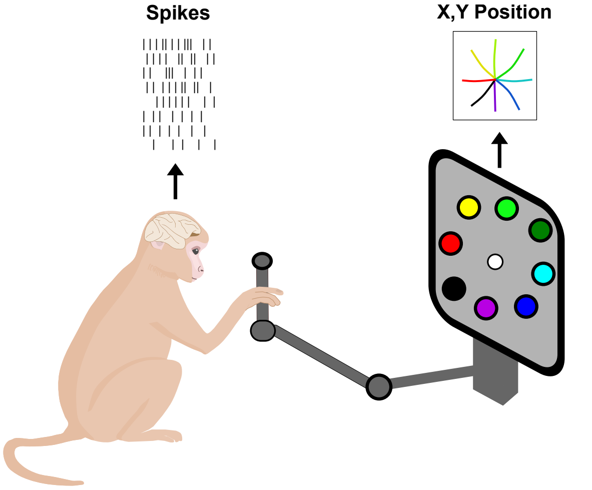

Figure 3. Experimental setup: In each trial one randomly chosen target direction (indicated by one of 8 coloured circles) appears on screen, and the monkey is instructed to control the cursor (white circle) by moving the manipulandum. The monkey moves the cursor to the target location after a go cue. The collected data for each trial consists of the neural spikes and monkey hand position across all timesteps. Our model is tasked with predicting hand position from neural spikes at each timestep.

4. Data

4.1. M1 neural recordings during reach task

We verify that SABLE is able to predict behaviour from unseen neural activity by applying it to data from a previously published experiment (Gallego et al., 2020). In this experiment, two monkeys are trained to perform a center-out reach task towards eight outer targets. On a go cue, each monkey moves a manipulandum along a 2D plane to guide a cursor on a screen to the target location (Figure 3). On successful trials a liquid reward is given. Spiking activity from the motor cortex (M1) along with the 2D hand position are recorded during each trial. Spike trains are converted into spike counts in 10ms bins, and behaviour variables are used at the same resolution. Only successful trials are used, all trials are aligned to movement onset and cut from movement onset to the shortest reach time across all trials.

For our analysis, we train SABLE on many consecutive days of recorded data and test on a subsequent held out day of recordings for each monkey. In total there are 13 near consecutive days of recordings for monkey 1 and 6 near-consecutive days of recordings for monkey 2, with fewer recorded neurons and timesteps for monkey 2 overall. Each day for each monkey consists of one recording session.

5. Models for comparison

We compare the ability of SABLE to predict behaviour from sessions of unseen spike data against existing methods and against a variation of our own model. We look at the following existing models: LFADS (Pandarinath et al., 2018) and RAVE+ (Gonschorek et al., 2021). We also compare against our own model where we do not reverse the gradient between the encoder and decoder, which we denote SABLE-noREV. In addition, we compare against a baseline RNN (GRU) with a linear readout layer optimised to predict movement from spiking data without autoencoding.

LFADS has been shown to have good efficacy at neural reconstruction across trials and sessions with some separation of behaviour in its latent space in previous work. We implement RAVE+ as an autoencoding model with GRUs for the encoder and decoder as our data are time series, and treat recording sessions as separate domains. As with our own model, the encoder is tasked with generating a small number of latent variables following a multivariate standard Gaussian distribution from neural data while the decoder is tasked with reconstructing the data from the latent variables. We use a non-linear layer as a domain classifier on the latent space between the encoder and decoder and implement a negative gradient between this classifier and the encoder network, thus encouraging the encoder to produce session-invariant latent variables. For all models we use the same regularisation techniques in the encoder or predictor as we do for SABLE to maximise generalisation.

For LFADS and RAVE+ we use a separately trained GRU to predict behaviour from the latent space of these models. We do not include ADAN (Farshchian et al., 2019) or the generative adversarial model by Wen et al. (2021) as both require at least some training data from held out session or subject to be effective. Implementation details of SABLE and all comparison models can be found in the Appendix (Section B).

6. Results

6.1. Application to motor cortex neural recordings during a reach task

We train all models on varying numbers of training sessions and for both monkeys, testing behaviour (2D hand position) prediction on intermediate and subsequent held out recording sessions. Our results, summarised in Figure 4, show that SABLE is capable of generalising to unseen data provided a sufficient number of training sessions are provided. In all cases tested SABLE outperforms the comparison models. For example, decoding accuracy for SABLE on an unseen intermediate session for monkey 1 with 12 training sessions is 0.91, which exceeds all other models by at least 0.25. For comparison, the RNN decoder typically yields an accuracy

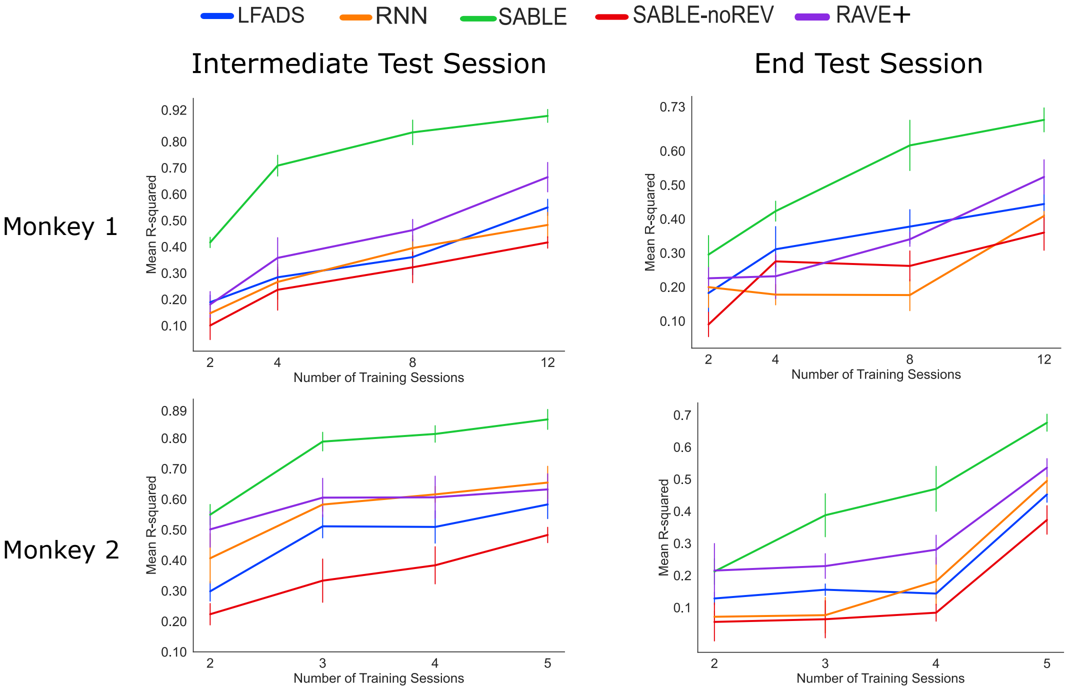

Figure 4. Behaviour prediction performance when testing all models on a completely unseen recording session. We report the mean r-squared between the inferred and true x,y positions. Each model is tested on a held out session while trained on different numbers of training sessions for both monkeys. The left column shows results for a held-out testing session which is in the chronological centre of the training sessions whereas the right column shows results for a test session recorded after all training sessions. Each test condition is run 10 times with different random seeds, with error bars showing standard deviation.

of around 0.92 on held-out data when trained and tested on a single session, indicating that SABLE can achieve saturation performance on unseen data. RAVE+ has a relatively high decoding performance when a large number of training sessions are used, likely because its domain adaptation paradigm removes some session specific variance in this case. In contrast, SABLE-noREV and LFADS have low decoding performance across monkeys and session numbers although they gradually improve with increasing session numbers.

Comparing the performance between the two monkeys, we see generally better overall decoding performance for monkey 2 at the same number of train sessions as monkey 1, although monkey 1 has far more total training data available (12 total consecutive sessions from monkey 1 vs. 5 for monkey 2) and so has higher peak behaviour decoding performance for all models. In addition, we limit the number of neurons for each monkey to the lowest number of neurons in any given session. Therefore, monkey 1 has 42 neurons of neural data across sessions whereas monkey 2 has 16.

Next we compare the difference between test performance for all models on both monkeys for different held out test session ordering (intermediate or end). While SABLE achieves end test session decoding performance exceeding that of current methods (0.71 mean r-squared with 12 train sessions), performance on any given intermediate test session is substantially higher (0.90 mean r-squared with 12 train sessions). Moreover, the performance of SABLE decreases noticeably faster when applied to an end test session when the number of training sessions is reduced versus an intermediate test session. This confirms that drift in recordings is gradual, not random.

6.2. Latent space analysis

Figure 5 shows T-SNE embeddings of the latent space of all autoencoding models. Each colour represents a different target direction with respect to behaviour, and embeddings of the training data are shown as circles while the test trials are shown as triangles.

The embeddings of the LFADS latent space show a clear separation that corresponds to the different training sessions. This shows that there are indeed significant differences between the sessions that are captured in the latent space and prevent generalisation to unseen sessions. Within each session cluster there is a good separation by target direction, indicating that the latent variables extract behaviourally rele-

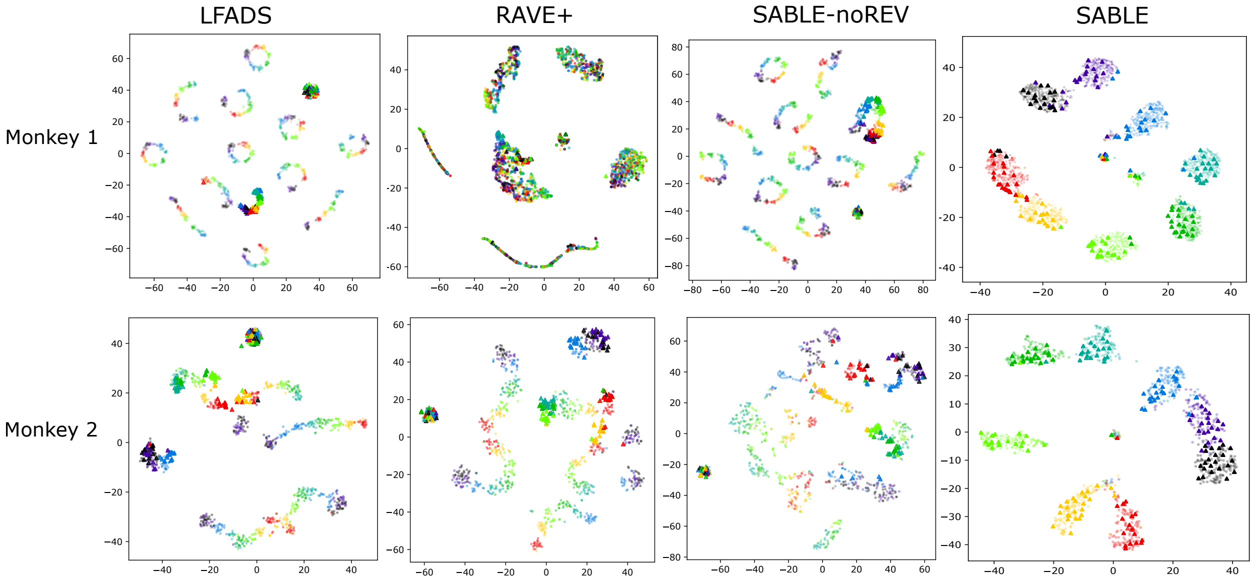

Figure 5. T-SNE embeddings of latent space for autoencoder models. In each embedding, points denoted by a circle are trials from 12 training sessions for monkey 1 and 5 training sessions for monkey 2. Points denoted by a triangle are trials from a held out intermediate test session for both monkeys. Each colour represents a target direction for the centre out reach task.

vant information from the neural activity. In contrast, many trials from the test session form a cluster in a region not covered by the training data, with some degree of separation by behaviour in monkey 1 (where more sessions are available for training). Here the model fails to assign these trials to meaningful latent variables as the differences in activity are too large to be mapped appropriately. We denote trials such as these hereafter as unassigned. Some trials are assigned locations in the latent space also occupied by the training data, indicating that despite session differences occasionally a matching of unseen to training data can be achieved, again with some separation by target direction. However there is also a cluster of unseparated test trials which the encoder of the model has failed to produce meaningful latent variables for due to these test trials being too disparate from the training trials.

For LFADS applied to monkey 2, the encoder still manages to separate trials by train session, but to a lesser degree. We suspect this is due to fewer neurons being available for the sessions of monkey 2. Here there is no longer a cluster of separated test trials, instead some test trials are assigned to existing train clusters as they are fairly similar. Here again however we see a large cluster of unassigned test trials. Overall we see that LFADS clusters train trials and coinciding test trials well but its encoder cannot effectively generate latent variables for dissimilar test trials for behaviour decoding.

The picture is different for RAVE+, where the latent space seems better aligned (more session invariant) but no longer well separated by behaviour for monkey 1 (larger training set and more neurons). In contrast, for monkey 2 we see little session alignment but better behaviour separation is achieved (fewer sessions and fewer recorded neurons). In this case there is some degree of merging of session clusters and organisation by behaviour target direction, but this is insufficient for good test behaviour decoding. For monkey 1 we see 7 clusters, 4 of which have some separation by direction for train session trials. However test trials are not separated well at all by direction. For monkey 2 there is far less clustering by session and some of these clusters separate by direction. Here there is also a large cluster of unassigned test trials. The domain adaptation method used in RAVE+ (reverse gradient based on session ID explicitly) thus does not seem to prevent trials clustering by session.

SABLE-noREV, our model without the reverse gradient, produces a result very similar to LFADS. For both monkeys, there is a cluster of well separated test trials and also many test trials that are unassigned to any cluster (either by direction or session). Therefore, using the latent space for both neural reconstruction and behaviour decoding simultaneously, as proposed by Hurwitz et al. (2021b), is not beneficial to test behaviour decoding across sessions.

Finally, applying SABLE to either monkey produces latent spaces which are very well separated by behaviour and almost entirely training-session invariant. We denote each unseen test trial as correctly classified in terms of target direction by observing whether a given trial gives a behaviour decoding r-squared of above 0.6. When applied to monkey 1 we see a small degree of misclassification of test trials by direction (13% of total test trials), but only when the correct and incorrect target directions are adjacent to each other spatially in the task outlined in Figure 3. This confirms that more similar behaviours in a task have more similar neural patterns and may be difficult for any decoder to disentangle. There are also a small number of unclassified test trials (3% of total test trials) in the centre of the embedding plot, we suspect these may be trials with highly contrasting spiking patterns to any train trial. When applied to monkey 2, we see less misclassification (4% of total test trials) by test behaviour direction and just a couple of unclassified test trials. We also note that both SABLE embeddings are topographically similar and correspond to the spatial aspect of the movement directions of the task outlined in Figure 3.

The stark differences in latent variables seen between SABLE-noREV and SABLE are quite surprising considering the only difference between these models is the reverse gradient between neural decoder and encoder in SABLE versus a positive gradient in SABLE-noREV. This shows the importance of a negative neural reconstruction gradient in training the SABLE encoder network to generate session invariant latent variables. In addition, we suspect that SABLE's encoder generates far fewer unassigned latent variables than the other autoencoding models due to the simpler and more behaviourally structured latent space.

6.3. Behaviour decoding

Examining the decoded behaviour of monkey hand position using SABLE (Figure 6) shows good overall reconstruction of movement trajectories, especially when testing on an intermediate test session. The intermediate test session behaviour decoding for both monkeys mirrors the SABLE T-SNE embedding in Figure 5. Test trials which are incorrectly assigned with respect to movement directions (Figure 5) are decoded correspondingly (Figure 6). Therefore, the behaviour decoder network of SABLE directly utilises latent variables in a particular cluster and decodes one particular direction of movement. Our model is thus consistent with the hypothesis outlined above.

When decoding from an end test session however, this phenomenon is less pronounced as the encoder seems to be less certain of the clusters formed in the latent space. There are more wrongly assigned test trials and the decoded movement trajectories are more spread out, leading to a lower overall mean r-squared when predicting behaviour.

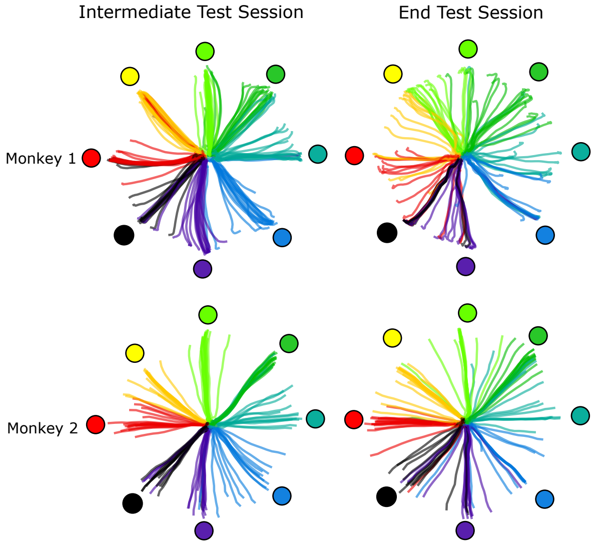

Figure 6. Predicted 2D monkey hand position of test trials using SABLE when trained on 12 train sessions for monkey 1 and 5 train sessions for monkey 2 and tested on an unseen intermediate or end test session.

6.4. Predicting behaviour from an unseen subject

Next we predict behaviour from the unseen neural data from 37 sessions of monkey 2 when SABLE is trained on 14 recording sessions of monkey 1. We use all available recording sessions available for monkey 1 spread across 3 years to train SABLE as we believe this gives the best opportunity for cross subject generalisation. However, for the held-out data we only obtain a mean r-squared of 0.03, so the model fails to generalise to a different animal. Examining the T-SNE embedding in Figure 7 shows that the trials from the training sessions of monkey 1 cluster well by movement direction but the trials from monkey 2 do not map to these direction clusters as the separated sessions of either just monkey 1 or monkey 2 do (as seen in Figure 5). Therefore it appears that the relationship between the recovered latent dynamics and behaviours differs between the two animals, and may require an extra alignment step.

7. Discussion

In this work we present a new method, SABLE, for aligning neural activity with complex temporal dynamics from different recording sessions to allow for consistent behaviour decoding across sessions. We apply it to neural recordings from primate motor cortex during a reaching task where the considerable variability between recording sessions prevents generalisation for a conventional decoder.

The model is trained as a variational autoencoder similar to LFADS (Pandarinath et al., 2018), with an additional gradient from behaviour decoder that disentangles the latent space to enable improved behaviour decoding (Hurwitz et al., 2021b). Reversing the gradient from the neural reconstruction encourages the model to ignore variability in the activity that is irrelevant for decoding activity, which in turn results in an session-invariant encoding of behaviour-relevant factors.

Unlike other domain adaptation methods, our model does not require domain labels, but is trained on a single data set that contains different experimental sessions. This is an advantage in potential BCI applications as variability may not only exist between single sessions, but also within a single session, and in addition the degree of variability may differ as well. As a result, the model still requires considerable amounts of session data for good behaviour reconstruction. We found that performance did not saturate when it was trained on 12 sessions. In contrast, good behaviour decoding with our baseline RNN model could be achieved from a single session with less than 200 trials. Yet we expect that the number of trials per session required is much less than used here.

A main limitation of this method, which may also limit its direct application in a BCI system, is that it assumes that behaviour is generated by autonomous neural dynamics which relies on specification of appropriate initial conditions that form the latent variables in the model. This approach has been shown to successfully capture neural dynamics in a range of scenarios (Pandarinath et al., 2018) and has the advantage of a relatively compact and behaviorally relevant latent encoding that supports discovering invariant features in the neural activity. A possible extension to remove this limitation may be the inclusion of a controller input that models additional temporal dynamics to better account for behavioural variability (Pandarinath et al., 2018). This extension of the latent space can be trained in the same manner and may allow modelling of more complex and variable behavioural paradigms.

We compare our model to RAVE+ (Gonschorek et al., 2021), to our knowledge currently the only other method for domain adaptation of inter-session data. RAVE+ does show some indication of alignment when sufficient individual recording sessions are available, but its latent space fails to capture behaviourally relevant structure. As a result, behaviour decoding for unseen test data is poor. We suspect that the RAVE+ fails because the temporal dynamics in our data are too variable between trials. As pointed out by the authors, RAVE+ requires consistent temporal dynamics between trials, which can be controlled in experiments where stimuli are presented, but that are rarely obtained in behavioural experiments. The other models shown here (LFADS, RNN decoder) are included to contrast domain adaptation to conventional encoders, and not as a baseline for generalisation performance.

Our results are consistent with recent reports showing that motor control is based on low-dimensional latent neural dynamics that are very stable over time despite ongoing neural drift (Gallego et al., 2020). Our model can be used to discover these latent dynamics in data with high variability. Tests we performed on synthetic data (a Lorenz system) indicate that this approach is also successful when neural dynamics are generated from latent dynamics with random transformations (not shown).

Our finding that SABLE has a better performance for intermediate held-out sessions than for sessions at the end of a sequence of training sessions suggests that performance will likely eventually decline for long time intervals between train and test sessions. As long as the latent dynamics are stable however, we expect that training the model with more sessions will eventually stabilise generalisation performance. Taken together these results are encouraging for BCI application as they suggest highly consistent recordings may not be required for good performance as long as it is possible to recover relevant latent dynamics.

References

Chestek, C. A., Batista, A. P., Santhanam, G., Byron, M. Y., Afshar, A., Cunningham, J. P., Gilja, V., Ryu, S. I., Churchland, M. M., and Shenoy, K. V. Single-neuron stability during repeated reaching in macaque premotor cortex. *Journal of Neuroscience*, 27(40):10742–10750, 2007.

Chestek, C. A., Gilja, V., Nuyujukian, P., Foster, J. D., Fan, J. M., Kaufman, M. T., Churchland, M. M., Rivera-Alvidrez, Z., Cunningham, J. P., Ryu, S. I., et al. Long-term stability of neural prosthetic control signals from silicon cortical arrays in rhesus macaque motor cortex. *Journal of Neural Engineering*, 8(4):045005, 2011.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2014. doi: 10.3115/v1/d14-1179.

Dabagia, M., Kording, K. P., and Dyer, E. L. Comparing high-dimensional neural recordings by aligning their low-dimensional latent representations. *arXiv preprint arXiv:2205.08413*, 2022.

Degenhart, A. D., Bishop, W. E., Oby, E. R., Tyler-Kabara, E. C., Chase, S. M., Batista, A. P., and Yu, B. M. Stabilization of a brain–computer interface via the alignment of low-dimensional spaces of neural activity. *Nature Biomedical Engineering*, 4(7), 2020. doi: 10.1038/s41551-020-0542-9.

Farshchian, A., Gallego, J. A., Miller, L. E., Solla, S. A., Cohen, J. P., and Bengio, Y. Adversarial domain adaptation for stable brain-machine interfaces. In *7th International Conference on Learning Representations, ICLR 2019*, 2019.

Gallego, J. A., Perich, M. G., Chowdhury, R. H., Solla, S. A., and Miller, L. E. Long-term stability of cortical population dynamics underlying consistent behavior. *Nature Neuroscience*, 23(2), 2020. doi: 10.1038/s41593-019-0555-4.

Ganin, Y. and Lempitsky, V. Unsupervised domain adaptation by backpropagation. In *32nd International Conference on Machine Learning, ICML 2015*, volume 2, 2015.

Gonschorek, D., Höfling, L., Szatko, K. P., Franke, K., Schubert, T., Dunn, B. A., Berens, P., Klindt, D. A., and Euler, T. Removing inter-experimental variability from functional data in systems neuroscience. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=lVmIjQiJJSr.

Herrero-Vidal, P., Rinberg, D., and Savin, C. Across-animal odor decoding by probabilistic manifold alignment. *Advances in Neural Information Processing Systems*, 34, 2021.

Hurwitz, C., Kudryashova, N., Onken, A., and Hennig, M. H. Building population models for large-scale neural recordings: Opportunities and pitfalls. *Current Opinion in Neurobiology*, 70:64–73, 2021a.

Hurwitz, C., Srivastava, A., Xu, K., Jude, J., Perich, M., Miller, L., and Hennig, M. Targeted neural dynamical modeling. *Advances in Neural Information Processing Systems*, 34, 2021b.

Kingma, D. P. and Ba, J. L. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.

Pandarinath, C., O'Shea, D. J., Collins, J., Jozefowicz, R., Stavisky, S. D., Kao, J. C., Trautmann, E. M., Kaufman, M. T., Ryu, S. I., Hochberg, L. R., et al. Inferring single-trial neural population dynamics using sequential autoencoders. *Nature Methods*, 15(10):805–815, 2018.

Rule, M. E., O'Leary, T., and Harvey, C. D. Causes and consequences of representational drift. *Current Opinion in Neurobiology*, 58:141–147, 2019.

Sani, O. G., Abbaspourazad, H., Wong, Y. T., Pesaran, B., and Shanechi, M. M. Modeling behaviorally relevant neural dynamics enabled by preferential subspace identification. *Nature Neuroscience*, 24(1), 2021. ISSN 15461726. doi: 10.1038/s41593-020-00733-0.

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

Sussillo, D., Stavisky, S. D., Kao, J. C., Ryu, S. I., and Shenoy, K. V. Making brain-machine interfaces robust to future neural variability. *Nature Communications*, 7, 2016. ISSN 20411723. doi: 10.1038/ncomms13749.

Wen, S., Yin, A., Furlanello, T., Perich, M., Miller, L., and Itti, L. Rapid adaptation of brain–computer interfaces to new neuronal ensembles or participants via generative modelling. *Nature Biomedical Engineering*, pp. 1–13, 2021.

A. Cross-subject decoding T-SNE Embedding

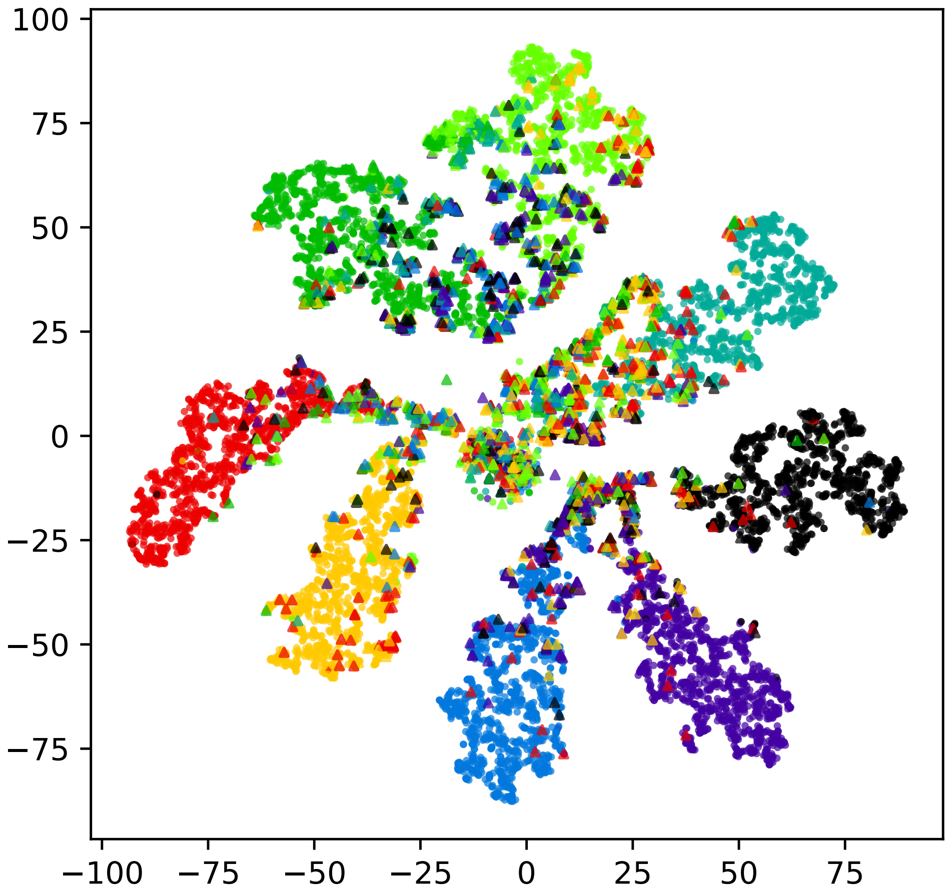

Figure 7. T-SNE embedding of SABLE latent space when training on 37 sessions of monkey 1 and testing on 14 sessions of monkey 2. Training trials are denoted by circles and test trials by triangles. Each colour denotes a particular movement direction.

B. Model Details

Below are implementation details for all models used in this paper.

SABLE		
Parameter	Value	Notes
Encoder		
- RNN Units	512 X 3	Stacked Gated Recurrent Unit
- RNN L2 Kernel Regularisation	1000	
- RNN L2 Recurrent Regularisation	1000	
- Recurrent Dropout	0.2	
- W_{enc} Units	512	Non-linear layer
- W_{enc} Dropout	0.8	
- W_{enc} L2 Regularisation	1000	
- Latent space dimension	64	
Neural Decoder		
- RNN Units	256	Gated Recurrent Unit
- RNN L2 Kernel Regularisation	0.1	
- RNN L2 Recurrent Regularisation	0.1	
- W_{fac} Units	128	Non-linear layer
- W_{fac} Dropout	0.2	
- W_{fac} L2 Regularisation	10	
Behaviour Decoder		Batch Normalisation on all layers
- RNN Units	256 X 2	Stacked Gated Recurrent Unit
- W_{beh} Units	512	Non-linear layer
- W_{beh} Dropout	0.1	
- W_{beh} L2 Regularisation	1.0	
Training		
Kullback–Leibler (KL) divergence weighting (λ_{kl})	0.01 to 10000	Rising exponentially **(between encoder and neural decoder)**
Reverse Gradient weighting (λ_r)	1.0 to 0.000000001	Decaying exponentially

SABLE-noREV		
Parameter	Value	Notes
Encoder		
- RNN Units	512 X 3	Stacked Gated Recurrent Unit
- RNN L2 Kernel Regularisation	1000	
- RNN L2 Recurrent Regularisation	1000	
- Recurrent Dropout	0.2	
- W_{enc} Units	512	Non-linear layer
- W_{enc} Dropout	0.8	
- W_{enc} L2 Regularisation	1000	
- Latent space dimension	64	
Neural Decoder		
- RNN Units	256	Gated Recurrent Unit
- RNN L2 Kernel Regularisation	0.1	
- RNN L2 Recurrent Regularisation	0.1	
- W_{fac} Units	128	Non-linear layer
- W_{fac} Dropout	0.2	
- W_{fac} L2 Regularisation	10	
Behaviour Decoder		Batch Normalisation on all layers
- RNN Units	256 X 2	Stacked Gated Recurrent Unit
- W_{beh} Units	512	Non-linear layer
- W_{beh} Dropout	0.1	
- W_{beh} L2 Regularisation	1.0	
Training		
Kullback–Leibler (KL) divergence weighting (λ_{kl})	0.01 to 10000	Rising exponentially
Reverse Gradient weighting (λ_r)	N/A	**Constant positive gradient of 1**

LFADS		
Parameter	Value	Notes
Encoder		
- RNN Units	512 X 3	Stacked Gated Recurrent Unit
- RNN L2 Kernel Regularisation	1000	
- RNN L2 Recurrent Regularisation	1000	
- Recurrent Dropout	0.2	
- W_{enc} Units	512	Non-linear layer
- W_{enc} Dropout	0.8	
- W_{enc} L2 Regularisation	1000	
- Latent space dimension	64	
Neural Decoder		
- RNN Units	256	Gated Recurrent Unit
- RNN L2 Kernel Regularisation	0.1	
- RNN L2 Recurrent Regularisation	0.1	
- W_{fac} Units	128	Non-linear layer
- W_{fac} Dropout	0.2	
- W_{fac} L2 Regularisation	10	
Behaviour Decoder		**Trained separately to rest of model**
		Batch Normalisation on all layers
- RNN Units	256 X 2	Stacked Gated Recurrent Unit
- W_{beh} Units	512	Non-linear layer
- W_{beh} Dropout	0.1	
- W_{beh} L2 Regularisation	1.0	
Training		
Kullback–Leibler (KL) divergence weighting (λ_{kl})	0.01 to 10000	Rising exponentially
Reverse Gradient weighting (λ_r)	N/A	**Constant positive gradient of 1**

RAVE+		
Parameter	Value	Notes
Encoder		
- RNN Units	512 X 3	Stacked Gated Recurrent Unit
- RNN L2 Kernel Regularisation	1000	
- RNN L2 Recurrent Regularisation	1000	
- Recurrent Dropout	0.2	
- W_{enc} Units	512	Non-linear layer
- W_{enc} Dropout	0.8	
- W_{enc} L2 Regularisation	1000	
- Latent space dimension	64	
Neural Decoder		
- RNN Units	256	Gated Recurrent Unit
- RNN L2 Kernel Regularisation	0.1	
- RNN L2 Recurrent Regularisation	0.1	
- W_{fac} Units	128	Non-linear layer
- W_{fac} Dropout	0.2	
- W_{fac} L2 Regularisation	10	
Behaviour Decoder		**Trained separately to rest of model**
		Batch Normalisation on all layers
- RNN Units	256 X 2	Stacked Gated Recurrent Unit
- W_{beh} Units	512	Non-linear layer
- W_{beh} Dropout	0.1	
- W_{beh} L2 Regularisation	1.0	
Domain Classifier		
- Non-linear layer Units	256 X 2	Batch Normalisation
- Dropout	0.1	
- L2 Regularisation	0.001	
Training		
Kullback–Leibler (KL) divergence weighting (λ_{kl})	0.01 to 10000	Rising exponentially
		(between encoder and Domain Classifier)
Reverse Gradient weighting (λ_r)	1.0 to 0.000000001	Decaying exponentially

4.4 Discussion

In this chapter I developed a novel domain adaptation approach (SABLE) for unifying the representations of many recording sessions of neural data. The primary mechanism used for this is a reverse gradient between the encoder and decoder of a sequential autoencoder. I show how crucial this reverse gradient is by testing the decoding performance of a comparison model with the same architecture as SABLE but lacking this reverse gradient (SABLE-noREV). SABLE-noREV is unable to form a session invariant encoder and so the latent space produced is similar to that of LFADS, whereby trials are separated by behaviour (movement direction) as well as by session.

On intracortical recordings from two monkeys performing a centre-out reach task, I demonstrate that SABLE is effective in forming a session invariant representation of several recording sessions for each monkey. I show that this invariant representation can generalise to a held out session which was recorded close in time to the sessions used for training. This allows for accurate behaviour prediction from the unseen session without any retraining of the model on the unseen session. This has not been previously shown and represents a novel and significant result.

While SABLE is robust to nearby unseen recording sessions, behaviour decoding accuracy is only high on an intermediate session, not on an end session which is a more realistic use case, for example as a BCI system. SABLE is also highly parameterised utilising three stacked Gated Recurrent Units (GRUs) with 512 units each in order to achieve held out session generalisation. The model also performs most effectively with a high level of regularisation, namely dropout of fully connected layers, but also L2 and recurrent dropout applied to GRUs. Training time is also somewhat slow due the high number of sessions which need to be trained on (12 sessions were used for training in the case of monkey 1).

SABLE also does not explicitly take into account the neural variability between sessions, so it is inherently difficult to see how the model is actually able to generalise to an unseen session where neuron positions have shifted and firing patterns have changed. In addition, SABLE, being based on LFADS and being a latent variable model, is only effective when trained on whole trials of neural data offline and cannot operate online on small windows of trial data as would be required for a real world BCI application.

Moreover, I demonstrate that SABLE cannot generalise to trials from recording sessions across monkeys. This is to be expected, as generalising across subjects is implausible even with recalibration. This is in part due to there being no direct (albeit unknown) mapping of neurons across subjects as there is across sessions of the same subject.

Chapter 5

Capturing cross-session neural population variability through self-supervised identification of consistent neuron ensembles

It has been shown that high dimensional neural population activity occupies a low dimensional manifold, therefore we can expect separate sessions of neural recordings from the same subject to occupy the same low dimensional manifold. However, drifts in activity of individual neurons can be substantial from day to day [73, 10]. While this drift is random and cannot be predicted on an individual neuron level, other population level variations over consecutive recording sessions such as differing sets of neurons and varying permutations of consistent neurons in recorded data may be learnable (Figure 5.1). The model outlined in the last chapter (SABLE) does not explicitly model neural drift or population variation (such as that due to probe movement). Here I aim to create a model capable of learning population variation.

In this chapter, I show that classification of consistent and unfamiliar neurons, along with detection of the order and presence of recording neurons over subsequent sessions of recordings is crucial in maintaining behaviour decoding performance on an unseen session of neural data. I propose forming variations of individual trials from a single recording session of two monkeys performing a centre-out reach task. These variations are formed by applying perturbations to these trials, mirroring the non-stationaries existing across sessions. I train a deep recurrent neural network to perceive these perturbations and subsequently leverage this perception to train a se-

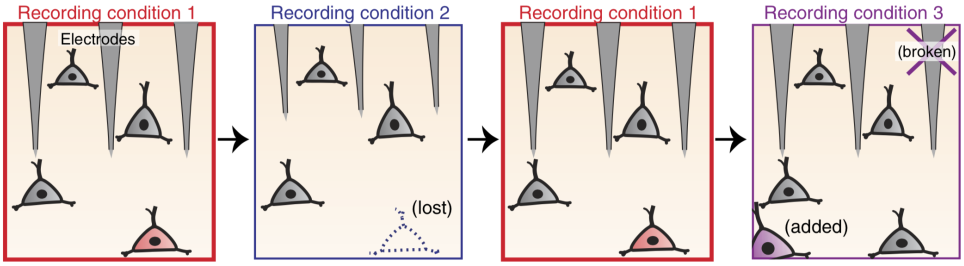

Figure 5.1: Adapted from [84] - Neurons in recording session on day 0 can be lost and new neurons can be added over subsequent recording sessions when recording using chronic intracortical microelectrodes.

quential autoencoding model to test decoding performance on completely unseen sessions of neural activity up to a week into the future, from just one session of training recording. This is a notable advance from SABLE in the previous chapter, which is only capable of generalising to a single (subsequent day) recording session.

I train the aforementioned RNN to perceive perturbations by optimising it to identify the initial positions of neurons in the original unperturbed trial. Once trained, I use the activations of this network as an embedding to aid in training an LFADS [68] model which will be tasked with taking perturbed trials of neural data and predicting original trials. For each trial of neural data we generate several perturbations such that, through contrastive learning, the encoder of the LFADS model learns to map variations of the same trial to similar latent variables and variations of other trials to distant latents. Together, this ensemble of the neuron locator RNN and the LFADS variant is termed CAPTIVATE.

The LFADS variant trained is highly robust to newly introduced neurons in unseen sessions due to the neuron locator RNN's embedding. I observe high movement decoding accuracy on several sessions of unseen neural data from just one training session, as opposed to the many training sessions required to achieve high movement decoding accuracy on an unseen session in the model (SABLE) outlined in the chapter above. This is due to the model and training paradigm described in this chapter more directly modelling the neuronal changes occurring between sessions versus SABLE.

5.1 Contribution

I am the first author and lead of this work. As such, I conceptualised the model, implemented all versions of the model, ran and evaluated the methods, and wrote the manuscript along with Matthias Hennig. Matthew Perich and Lee Miller provided the dataset used to train and evaluate the model.

5.2 Paper

Capturing cross-session neural population variability through self-supervised identification of consistent neuron ensembles

Justin Jude
University of Edinburgh
justin.jude@ed.ac.uk

Matthew G. Perich
Icahn School of Medicine at Mount Sinai
New York, NY 10029
mperich@gmail.com

Lee E. Miller
Feinberg School of Medicine
Northwestern
Chicago, IL 60611
lm@northwestern.edu

Matthias H. Hennig
University of Edinburgh
m.hennig@ed.ac.uk

Abstract

Decoding stimuli or behaviour from recorded neural activity is a common approach to interrogate brain function in research, and an essential part of brain-computer and brain-machine interfaces. Reliable decoding even from small neural populations is possible because high dimensional neural population activity typically occupies low dimensional manifolds that are discoverable with suitable latent variable models. Over time however, drifts in activity of individual neurons and instabilities in neural recording devices can be substantial, making stable decoding over days and weeks impractical. While this drift cannot be predicted on an individual neuron level, population level variations over consecutive recording sessions such as differing sets of neurons and varying permutations of consistent neurons in recorded data may be learnable when the underlying manifold is stable over time. Classification of consistent versus unfamiliar neurons across sessions and accounting for deviations in the order of consistent recording neurons in recording datasets over sessions of recordings may then maintain decoding performance. In this work we show that self-supervised training of a deep neural network can be used to compensate for this inter-session variability. As a result, a sequential autoencoding model can maintain state-of-the-art behaviour decoding performance for completely unseen recording sessions several days into the future. Our approach only requires a single recording session for training the model, and is a step towards reliable, recalibration-free brain computer interfaces.

1 Introduction

Neural decoders require stable neurons in a recorded population in order to accurately predict behaviour such as movement or to allow decoding of stimuli. However, over time instabilities in the recording equipment and drift in neural activity lead to instabilities that prevent re-using a decoder trained on one day for a session recorded on another day [Huber et al., 2012, Ziv et al., 2013, Driscoll et al., 2017]. At the same time, neural population activity is highly structured and often confined to low-dimensional manifolds [Cunningham and Byron, 2014] that can be recovered using latent variable modelling approaches [Hurwitz et al., 2021]. Importantly, recent work showed that movement-related latent neural dynamics in population activity from the primate motor cortex is stable and could be recovered over intervals as long as two years [Gallego et al., 2020]. This suggests that despite the variability at the level of single neurons, in each session a subset of neurons will remain informative about behaviour. A stable cross-session decoder therefore has to be able to identify these neurons and utilise them for decoding. Therefore, here we focus on identifying known recording neurons in unseen sessions. In particular, we hypothesised that a latent encoding of neural activity can be augmented by information about which neurons were seen during training, and at which position in the input. We show that this is sufficient to decode behaviour (in our case different cued arm movements by a monkey with simultaneous motor cortex recordings) with high accuracy across unseen sessions.

We achieve this with a self-supervised approach through training a recurrent neural network (RNN) to predict original neuron positions following data perturbation in a manner mirroring session to session variability. In essence, the closer our perturbations mimic real inter-session variability (as shown in Figure 1), the higher our behaviour prediction performance on an unseen session. These perturbations include adding spikes to existing neurons from randomly generated neurons, removing spikes from existing neurons, shifting the entire neuron population by a constant amount, slightly shifting neurons in time, replacing neurons with randomly generated neurons and eliminating neurons entirely.

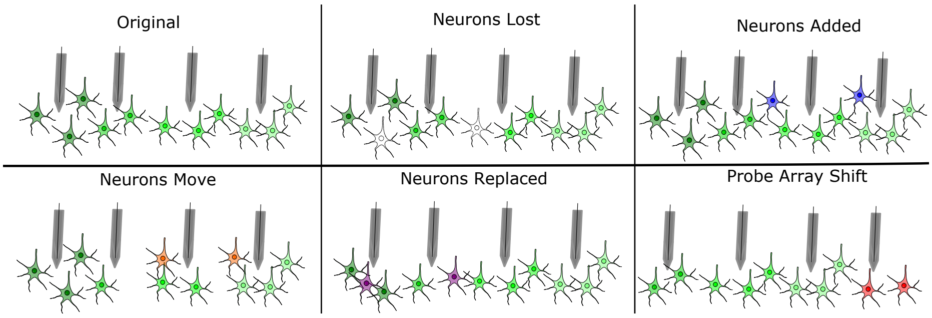

Figure 1: Inter-session ensemble variability possible when recording from neural populations. Neurons from the original recording session can be lost to the recording array, new neurons can become visible, neurons can move between electrodes, original neurons can be replaced by unseen neurons and the entire probe array can shift, causing a systematic change in neuron position. In addition, spike sorting can induce variability as the signal to noise ratio of individual neurons changes between sessions. The perturbations we apply to each trial of recordings is in response to each of these sources of variability. We model each unseen test trial as an instance of a perturbed seen train trial and subsequently, our sequential autoencoder model attempts to map each unseen trial to a known trial.

This neuron locator RNN is trained to predict original neuron position within a single recording session from many perturbed variations of trials of this training session. Once trained to predict original neuron positions, a separate network, which in this case is a sequential autoencoder based on Latent Factor Analysis via Dynamical Systems (LFADS) [Pandarinath et al., 2017], is trained to predict original unperturbed neural recording trials from perturbed variations of trials from the same session. The encoder of this sequential autoencoder receives as additional input the embedding of the neuron locator RNN activations, conditioning the encoder to produce latent variables which are informative enough to accurately reconstruct the original recording. The encoder produces latent variables which are separated by behaviour (arm movement direction) in a self-supervised manner, from which behaviour can be predicted without the model being explicitly trained on behaviour.

Importantly, the joint neuron locator RNN and LFADS encoder ensemble can predict behaviourally relevant latent variables for unseen recording sessions that yield high decoding accuracy. Currently, there are no existing approaches to accurately predict behaviour from an unseen recording session when training on just one single session. We not only show this is possible with our method, but that our approach is robust to inter-session variability for up to 8 days when a sufficient number of neurons are persistent across sessions.

2 Related Work

There have been many recent approaches to creating robust behaviour decoders of neural activity [Gallego et al., 2020, Farshchian et al., 2019, Sussillo et al., 2016, Wen et al., 2021, Karpowicz et al., 2022, Wimalasena et al., 2021]. However these methods are not capable of decoding behaviour from a previously unseen recording session if the recorded activity is subject to random fluctuations.

Recent work in modelling neural activity shows the consequences of selectively perturbing neural data in order to learn relevant latent variables in a self-supervised way using an autoencoder [Liu et al., 2021, Azabou et al., 2021, Zhu et al., 2021]. These models take different views of the same neural data and align the latent spaces of these views once passed through an encoder, with the ultimate aim of reconstructing these views. We utilise a similar technique to train our sequential autoencoder by aligning the latent variables of perturbed versions of the same data and aim to generate the activity of the original unperturbed trial. Importantly, Liu et al. [2021] propose a model which is invariant to the specific neurons used to represent the neural state within training data; in this work we look at unseen sessions and so do not aim to produce a model invariant to new neurons, but one that is able to identify and utilise seen neurons to reconstruct unperturbed trials.

Gonschorek et al. [2021] and Jude et al. [2022] use domain adaptation to align data across recording sessions. In both studies the authors use an autoencoder model and a domain classifier. However these models require training on many days of recording sessions for good behaviour decoding accuracy. For instance, Jude et al. [2022] requires as many as 12 training sessions and training on behaviour explicitly in order to produce high behaviour decoding accuracy on an unseen test session. In this work we achieve state-of-the-art behaviour decoding performance on an unseen test recording session using just one training recording session, and show that this decoding accuracy can be maintained many days into the future without recalibration.

We train an RNN to predict original neuron position from perturbed trials and utilise this network to inform the sequential autoencoder model. This is considered self-supervised learning as we do not train our model on behaviour explicitly but instead train on the subtasks of predicting original neuron positions and reconstructing unperturbed trials from perturbed ones. This approach is similar to that used in Noroozi and Favaro [2016], where authors form 9 subsets of images and randomly permute these subsets, then task the model with predicting the permutation.

3 M1 Recordings

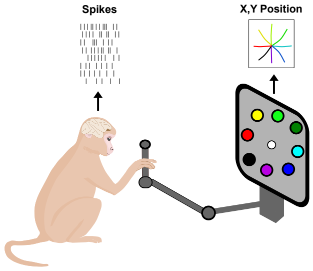

Figure 2: Experimental setup: In each trial one randomly chosen target direction (indicated by one of 8 coloured circles) appears on screen, and the monkey is instructed to control the cursor (white circle) by moving the manipulandum. The monkey moves the cursor to the target location after a go cue. The collected data for each trial consists of the neural spikes and monkey hand position across all timesteps. We predict hand position from neural spikes at each timestep.

We apply our model to data from a previously published experiment [Gallego et al., 2020]. In this experiment, two monkeys were trained to perform a center-out reach task towards eight outer targets. On a go cue, each monkey moves a manipulandum along a 2D plane to guide a cursor on a screen to the target location (Figure 2). On successful trials a liquid reward is given. Spiking activity from the motor cortex (M1) along with the 2D hand position were recorded during each trial. Spike trains were converted into spike counts in 10ms bins, and behaviour variables are used at the same resolution. In this work, only successful trials are used, all trials are aligned to movement onset and cut from movement onset to the shortest reach time across all trials.

For our analysis, we train our model on one session of recorded data from a single day which we denote day 0 (containing 173 trials for both monkeys) and test on subsequent held out days of recordings for each monkey. A comparison of the activity between sessions shows considerable variability, caused by shifts in the order neurons appear as well as disappearance of neurons and the appearance of new ones (see Appendix B, Figure 8). These changes are particularly pronounced for longer time intervals, but are already significant in recordings one day apart. In total we used 5 days of recordings for both monkeys, with 55 recorded neurons across all sessions for Monkey C and 17 for Monkey M. Each day for each monkey consists of one recording session.

4 Data Perturbations

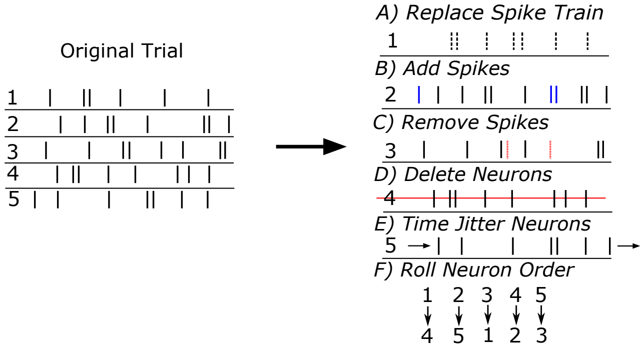

Figure 3: Perturbations applied simultaneously to each trial of neural data, demonstrated with a simple 5 neuron system. A) Replace entire spike train with a randomly generated neuron of the same firing rate as the original neuron. B) Spikes randomly added to spike train proportional to average firing rate of all neurons in a given trial, to mirror influence of nearby newly added unknown neurons. C) Spikes randomly removed to mimic removal or movement of nearby known neurons. D) Deletion of entire neurons to simulate neuron loss between sessions, with randomly generated neurons introduced as the first or last neuron of the trial to keep neuron number consistent. E) Small random time jitter of all neuron spike trains to simulate experimental variation between sessions. F) Constant random shift of the order of all neurons to mirror probe shift.

Fig. 3 outlines the perturbations forming each variation of a single trial during the training of our model. Perturbations A) to D) in Fig. 3 are applied with equal probability to a given neuron of a given trial. Perturbation E) is applied to all neurons, time jitter is chosen randomly between -30ms and +30ms. Perturbation F) is applied to all trials, the amount of this neuron shift is chosen randomly between 0 and 25% of the total number of neurons. We hypothesise that this combination of transformations sufficiently mirrors the real day to day changes of recorded neuron ensembles.

5 Model

Our modelling approach is based on the hypothesis that the perturbations mentioned above can capture the substantial variability between recording sessions from the same animal. We also expect neural activity x is related to the latent variables z through a simple function, however, this function will differ between recording sessions as we expect to observe different neurons in each session. The problem is thus to find the correct encoding function $z = f(x)$ to transform perturbed neural activity into a consistent latent space which then allows decoding of behaviour. In addition, for the same behaviour we require z_i for each trial i to be similar despite variations in the activity x_i.

We first train a fully connected layer and an RNN to predict original neuron position in perturbed trials. We apply the perturbations from Figure 3 to each trial, then task the network to predict the original position of each neuron in the recording data or whether it was previously unseen. As shown in Figure 4, for each neuron in the recording data we project a softmax linear read-out layer from the RNN which each form a probability distribution of predicted original neuron position across all possible positions (plus an extra position indicating that the neuron was randomly generated). Each of these is compared against a one hot encoding of the original neuron position before any perturbations have been applied. If the neuron is randomly generated then the one-hot encoding is one at the dedicated extra position. Predictions of original neuron position are made as follows:

$$\bar{x}_{i,1:T} = \text{Perturb}(x_{i,1:T}), \tag{1}$$

$$\text{acts}_i = \text{GRU}_{\theta_{\text{pos}}}(f_{\text{pos}}(\bar{x}_{i,1:T})), \tag{2}$$

$$\text{pos}_{i,n} = \text{softmax}(W^n_{\text{neuron}}.\text{acts}_i) \tag{3}$$

The predicted position for trial i and neuron n is then: $\operatorname{argmax} \text{pos}_{i,n}$

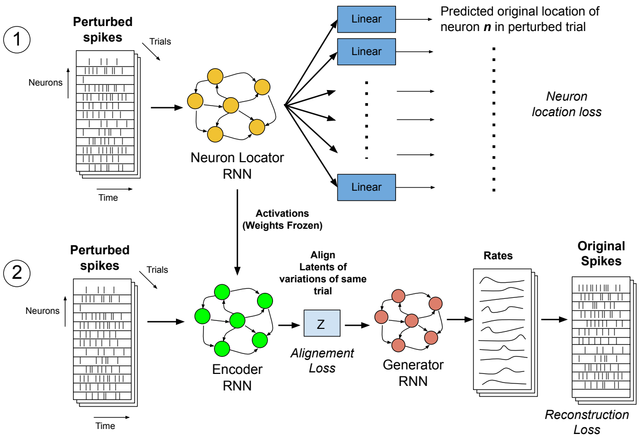

Figure 4: Our model consists of a neuron locator RNN (1) combined with a sequential variational autoencoding approach (2). The neuron locator (1) is trained first to identify original neuron position (or if the neuron is randomly generated) in each trial after perturbations have been applied. Then the neuron locator's weights are frozen and its activations are given as additional input to condition the encoder of the sequential autoencoder (2). Notably, we perturb recording trials when training both the neuron locator and sequential variational autoencoder. The sequential autoencoder is tasked with reconstructing the original unperturbed recording trials. The encoder of the sequential autoencoder maps perturbed versions of the same trial to similar latent variables. This is accelerated by imposing an alignment loss across the latent variables of variations of the same trial. The generator RNN of the sequential autoencoder predicts original trials from latent variables produced by the encoder RNN.

Perturb is the simultaneous application of all perturbations outlined in Section 4 to a given trial, so $x_{i,1:T}$ are the original trials and $\bar{x}_{i,1:T}$ are the perturbed trials. i indicates a particular trial and T is the total number of timesteps per trial. f_{pos} is a fully connected layer and θ_{pos} are the parameters of the locator network used to predict original neuron position, with acts$_i$ being the final RNN hidden state (activations) of this locator network for each trial (which we use later as a conditional embedding). W^n_{neuron} is the set of linear layers (one linear layer for each neuron in the data) used to predict original neuron position, producing a probability distribution when combined with a softmax layer for each neuron.

Once trained, the weights of this neuron locator network are frozen, and the activations of the RNN are used as additional input to the encoder of an LFADS-inspired sequential autoencoder. This input conditions the encoder in predicting latent variables used to generate original trials from perturbed trials. As proposed by Pandarinath et al. [2017] we assume that the latent dynamics evolve autonomously provided a set of initial conditions z_i that are modelled as Gaussian random variables. These latent variables are produced for each trial by an encoder network consisting of bidirectional Gated Recurrent Units [Cho et al., 2014] (GRU). They are used to reconstruct the original trial-specific neural activity from the perturbed trials. A further bidirectional GRU is used as a generator for neural reconstruction of unperturbed trials from latent variables z_i. Training is based on Poisson likelihood for unperturbed neural activity reconstruction (as in [Pandarinath et al., 2017]). The model is trained using real neural activity which corresponds to consistent behaviours (movement directions in a centre-out reach task). The generative process of our model is as follows:

$$z_i = f_{\text{enc}}(\text{GRU}_{\theta_{\text{enc}}}(\bar{x}_{i,1:T}; \text{acts}_i)), \tag{4}$$

$$g_{1:T} = \text{GRU}_{\theta_{\text{gen}}}(z_i), \tag{5}$$

$$r_t = exp(W_{\text{rate}}.f_{\text{fac}}(g_t)), \tag{6}$$

$$\hat{x}_t \sim \text{Poisson}(r_t) \tag{7}$$

where θ_{enc} and θ_{gen} are the parameters of the GRUs used to encode perturbed spike trains into latent variables and subsequently generate original unperturbed spike trains from the latent variables. f_{enc} and f_{fac} are fully connected layers which produce latent variables and neural activity factors respectively. W_{rate} is a linear transformation used to generate firing rates at each time step per trial.

At each training iteration the following three losses are optimised with Adam [Kingma and Ba, 2015]:

$$L_{\text{rec}} = -\sum_{t=1}^{T} \log(\text{Poisson}(x_{i,t}|r_t)) \tag{8}$$

$$L_{kl} = D_{KL}[\text{GRU}_{\theta_{\text{enc}}}(z_i|\bar{x}_i; \text{acts}_i)||\mathcal{N}(0, I)] = -\frac{1}{2}[\log(z_{i,\sigma}^2) - z_{i,\mu}^2 - z_{i,\sigma}^2 + 1] \tag{9}$$

$$L_{\text{align}} = \frac{1}{P}\sum_{j=1}^{P}\sum_{k\neq j}^{P}(z_{i,j} - z_{i,k})^2 \tag{10}$$

Together L_{rec} and L_{kl} are the usual evidence lower-bound of the marginal log-likelihood in a VAE [Kingma and Welling, 2014]. L_{rec} is minimised by the encoder network and the neural generator network. As in Liu et al. [2021], we apply an alignment loss (L_{align}) across latent variables produced from perturbed trials (where P is the number of perturbations of a given trial) of the same original trial z_i which reduces training duration. We form 2 perturbed variations of each trial in a given batch at each training iteration. Kullback–Leibler (L_{kl}) divergence loss (between a multivariate standard Gaussian distribution and the encoder-generated latent variables) and L_{align} are minimised by just the encoder network. We name our model CAPTure and Identify Variability at Target Ensembles (CAPTIVATE). Further implementation details can be found in Appendix A.

5.1 Comparison models

We compare the ability of CAPTIVATE to predict behaviour from sessions of unseen spike data against existing methods and against a variation of our own model where we do not use the locator network trained on original neuron position to aid in aligning perturbed trials. We denote this model variation CAPTIVATE-noLoc. In addition, we look at vanilla LFADS [Pandarinath et al., 2017] in autoencoding trials without any perturbations. We also compare against a baseline RNN (GRU) with a linear readout layer explicitly trained to reconstruct movement behaviour from neural activity.

For all autoencoding models we use a separately trained GRU network to predict behaviour from the day 0 training session latent space. We do not include ADAN [Farshchian et al., 2019], NoMAD [Karpowicz et al., 2022] or the generative model by Wen et al. [2021] as all require training data from a held out session or subject to be effective. We also do not test against Gonschorek et al. [2021] or [Jude et al., 2022] as these approaches require many training sessions to be effective in predicting behaviour from an unseen session whereas we aim to do this with just one training session.

6 Results

Figure 5 shows behaviour decoding performance of CAPTIVATE for an unseen session that was recorded the day after the training session for different total rates of perturbation. A total perturbation rate of 40% (i.e a rate of 10% for each perturbation A) - D) in section 4) for both monkeys appears to be optimal. At perturbation rates above 40%, neural activity from perturbed day 0 train trials with a particular target movement direction begin to resemble original trials of other movement directions, and thus hurt alignment. Perturbation rates below 40%, particularly for Monkey C, are not sufficient to simulate the inter-session variability between day 0 and day 1. Training the neuron locator RNN on a total perturbation rate of 40% for both monkeys yields 85% and 93% accuracy on predicting original neuron position from day 0 perturbed trials from Monkey C and Monkey M respectively. Indeed, the neuron locator network is 76% accurate at identifying original neuron position in a simulated unseen session created with a total perturbation rate of 80% (see Appendix D, Figure 10).

Using the optimal rate of 40% of perturbation to trials from both monkeys when training CAPTIVATE leads to the results summarised in Figure 6. For both monkeys we see high behaviour decoding performance on the unseen session from day 1, surpassing previous methods. CAPTIVATE maintains high behaviour decoding performance for Monkey C on an unseen session up to 8 days after the day 0 training session was recorded. CAPTIVATE also accurately maps neurons from trials across unseen sessions of Monkey C up to 8 days into the future to known neurons from trials of the day 0 train session (see Appendix C, Figure 9). Notably, behaviour decoding for Monkey C is much more stable for future unseen sessions than for Monkey M. This is likely due to sessions from Monkey C containing more than 3 times as many neurons as Monkey M. However, we see in Appendix E, Figure 11 that training CAPTIVATE with 20 neurons from the Monkey C day 0 session is sufficient

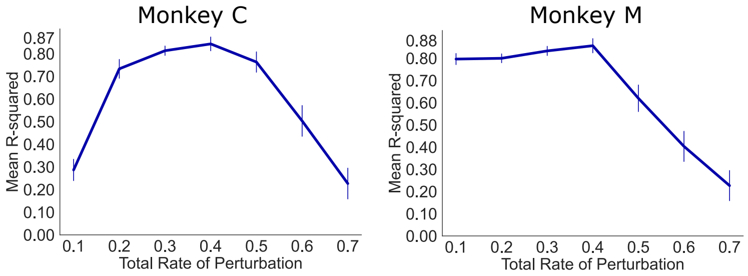

Figure 5: Behaviour decoding performance on an immediately subsequent unseen session (day 1) of CAPTIVATE at different rates of total perturbation. Total perturbation rate is the sum of the rates of perturbations A) - D) outlined in section 4, each of which are applied at equal rates.

to achieve an R^2 of 0.68 when testing on 20 neurons of the day 8 session, indicating our model can be robust to a low number of neurons.

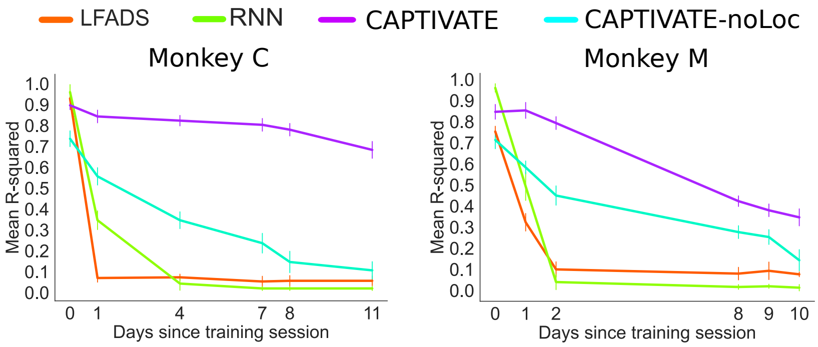

Figure 6: Behaviour prediction performance when testing all models on 30% of held-out trials from day 0 and subsequent days of completely unseen recording sessions. We report the mean R^2 between the inferred and true x,y positions. Each model is tested on held out trials from day 0 and trials from unseen sessions recorded an increasing number of days into the future from the original training session (day 0) for both monkeys. Each day 0 train session is run 10 times with different random seeds, with error bars showing standard deviation when applied to each unseen session.

Notably in the case of Monkey M, day 1 decoding performance is high at all levels of perturbation from 0.1 to 0.4 (Figure 5), therefore it is likely that the session to session variability between day 0 and day 1 is small. Thus, for a subject with fewer neurons in recorded data, CAPTIVATE may only require a low rate of total perturbation when aligning nearby unseen sessions.

CAPTIVATE-noLoc, Vanilla LFADS or an RNN model cannot capture session-to-session variability even for the day 1 unseen session, as shown in Figure 7. CAPTIVATE-noLoc cannot accurately reconstruct original trials from perturbed variations of the day 0 train session, but has a similar day 0 and day 1 session behaviour decoding accuracy, implying our perturbations closely mirror inter-session variability. This indicates that poor performance of CAPTIVATE-noLoc on both monkeys is due to the inability of the encoder of this model to recognise known neurons and thus, shows how crucial the neuron locator network is in recognising known neuron ensembles in unseen recordings.

LFADS is trained solely on unperturbed trials and so cannot recognise the shifts that occur between sessions as variations of the day 0 training session, and thus cannot create an appropriate latent encoding. The RNN model is also trained on unperturbed trials and is even less robust to later unseen sessions than LFADS, however, this RNN baseline can recognise some behaviour in both monkeys for the day 1 unseen session, indicating a relatively low level of variability in adjacent day recordings. Similarly, we see a mean R^2 of 0.37 ($\pm$ 0.02) when training an RNN on day 7 for monkey C and testing on day 8 and an R^2 of 0.42 ($\pm$ 0.04) when training an RNN on day 9 and testing on day 10 for monkey M. Importantly, none of these models overfit as they yield high decoding accuracies for a held-out portion of day 0 trials for both monkeys and for all models, especially the RNN. Therefore the performance drop of the RNN model when applied to unseen sessions is a clear indication of substantial variations between sessions.

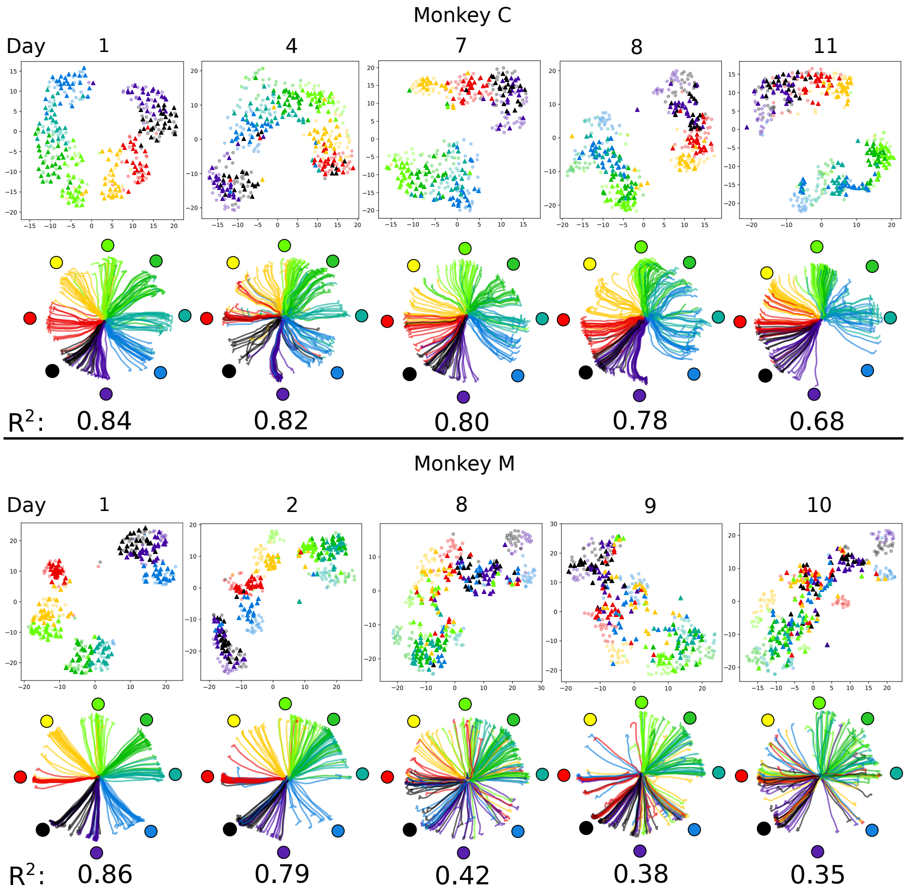

Figure 7: For each monkey, *Top row*: t-SNE embeddings of latent space for CAPTIVATE when applied to each unseen session. In each embedding, points denoted by a circle are trials from the day 0 training session. Points denoted by a triangle are trials from the named unseen session. Each colour represents a target direction for the centre-out reach task. *Bottom row*: Predicted 2D monkey hand position of trials using a separately trained RNN decoder trained only on the day 0 latent space of CAPTIVATE when applied to each unseen session, with mean R^2 between all positions of each predicted and ground truth trajectory shown across all trials in a given unseen session.

Figure 7 shows t-SNE visualisations of the latent space and behaviour predictions made from the latent space of CAPTIVATE when trained on the day 0 session and applied to unseen sessions. For Monkey C, the majority of trials from all unseen sessions are correctly aligned with the corresponding trials in the training data set (Figure 7, compare dots and triangles in the t-SNE plots where colour indicates movement direction; note that the latent space is well partitioned by behaviour although the model is only trained on neural activity). Occurrences where the unseen trials correctly overlap known train trials, in turn, yields correctly decoded behaviour. The alignment becomes progressively worse for later sessions, and as the alignment is less precise, behaviour predictions also become worse. In contrast, for Monkey M the alignment of trials beyond day 2 becomes increasingly worse, a consequence of the smaller number of neurons in the recording.

Ablations of individual perturbations (as outlined in Figure 3) applied when training on the day 0 session reveal that perturbations which introduce randomly generated neurons and alter the continuous ordering of neurons have the highest impact on unseen session behaviour decoding performance. This analysis is summarised in Table 1 and shows that neuron deletions, replacements and probe shifts cause the majority of inter-session neuron ensemble variability. Nonetheless, a combination of all perturbations are necessary for the decoding performance achieved by CAPTIVATE in Figure

Table 1: Mean decoding performance effects of ablating individual perturbations when training on day 0 session and tested on immediately subsequent (day 1) unseen session for both monkeys. For reference, the full CAPTIVATE model trained on day 0 achieves mean decoding R^2 performance of 0.84 (± 0.02) on monkey C and 0.86 (± 0.03) on monkey M when applied to the day 1 session.

Ablation	No-Replace	No-Add	No-Remove	No-Delete	No-Jitter	No-Reorder
C Mean R^2	0.66 (± 0.03)	0.79 (± 0.01)	0.74 (± 0.02)	0.49 (± 0.04)	0.77 (± 0.01)	0.70 (± 0.03)
M Mean R^2	0.71 (± 0.03)	0.75 (± 0.02)	0.81 (± 0.01)	0.63 (± 0.03)	0.81 (± 0.01)	0.77 (± 0.02)

6. We also train CAPTIVATE without the alignment loss in Eq. 10, which produces a behaviour decoding mean R^2 of 0.82 on Monkey C and 0.85 on Monkey M on trials from the day 1 unseen session. This minimal drop in decoding performance when training without an explicit alignment loss is consistent with results from [Liu et al., 2021]. Additionally, decoding performance across unseen sessions when training on day 0 and day 11 sessions separately is almost symmetrical (as shown in Appendix F, Figure 12), indicating that our model can effectively capture neural variability from unseen sessions both forwards and backwards in time. We further assess robustness by testing CAPTIVATE on a variable number of neurons across sessions (similar to a real BCI setting) and show good generalisation, even surpassing performance of the model trained with 55 neurons (as in Figure 6) for some unseen sessions (see Appendix G, Figure 13).

7 Discussion

In this paper we use a self-supervised approach, CAPTIVATE, to train a model to recognise and correct for session-to-session variability in neural recordings. We then show that the combination of this approach with a latent variable model that identifies low-dimensional dynamics in neural activity yields a model that is now robust variability between recordings sessions. The model is capable of successfully predicting behaviour with high accuracy from unseen sessions, surpassing previous work by Jude et al. [2022] when comparing against subsequent day decoding performance. Furthermore, our approach leads to relatively high and stable behaviour decoding performance on unseen sessions many days into the future when a sufficient number of neurons are persistent across sessions. As a result, this method performs better for data sets with more recorded neurons (Monkey C), while for fewer neurons the performance degrades more quickly, only producing good results for sessions close in time to the training session (Monkey M).

With CAPTIVATE we achieve stable behaviour decoding performance for up to 8 days, which is followed by a slow decline in performance. The decline is due to an increase in variability that could no longer be compensated. This would require a model to correct even stronger perturbations, but training a model this way leads to an overall decrease in performance even for short time intervals (Figure 5). Therefore long-term stable decoding currently still requires re-training of the components of a latent variable encoder model such that the altered neural dynamics are re-aligned with the latent dynamics [Wen et al., 2021, Karpowicz et al., 2022, Farshchian et al., 2019]. Equally, our model fails to successfully decode behaviour from recordings from an unseen animal (not illustrated) as this requires a more complex mapping function between activity and latent space [Wen et al., 2021].

References

M. Azabou, M. G. Azar, R. Liu, C.-H. Lin, E. C. Johnson, K. Bhaskaran-Nair, M. Dabagia, B. Avila-Pires, L. Kitchell, K. B. Hengen, W. Gray-Roncal, M. Valko, and E. L. Dyer. Mine your own view: Self-supervised learning through across-sample prediction, 2021. URL https://arxiv.org/abs/2102.10106.

K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2014. ISBN 9781937284961. doi: 10.3115/v1/d14-1179.

J. P. Cunningham and M. Y. Byron. Dimensionality reduction for large-scale neural recordings. *Nature Neuroscience*, 17(11):1500–1509, 2014.

L. N. Driscoll, N. L. Pettit, M. Minderer, S. N. Chettih, and C. D. Harvey. Dynamic reorganization of neuronal activity patterns in parietal cortex. *Cell*, 170(5):986–999, 2017.

A. Farshchian, J. A. Gallego, L. E. Miller, S. A. Solla, J. P. Cohen, and Y. Bengio. Adversarial domain adaptation for stable brain-machine interfaces. In *7th International Conference on Learning Representations, ICLR 2019*, 2019.

J. A. Gallego, M. G. Perich, R. H. Chowdhury, S. A. Solla, and L. E. Miller. Long-term stability of cortical population dynamics underlying consistent behavior. *Nature Neuroscience*, 23(2), 2020. ISSN 15461726. doi: 10.1038/s41593-019-0555-4.

D. Gonschorek, L. Höfling, K. P. Szatko, K. Franke, T. Schubert, B. A. Dunn, P. Berens, D. A. Klindt, and T. Euler. Removing inter-experimental variability from functional data in systems neuroscience. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. URL `https://openreview.net/forum?id=1VmIjQiJJSr`.

D. Huber, D. A. Gutnisky, S. Peron, D. H. O'connor, J. S. Wiegert, L. Tian, T. G. Oertner, L. L. Looger, and K. Svoboda. Multiple dynamic representations in the motor cortex during sensorimotor learning. *Nature*, 484(7395):473–478, 2012.

C. Hurwitz, N. Kudryashova, A. Onken, and M. H. Hennig. Building population models for large-scale neural recordings: Opportunities and pitfalls, 2021. ISSN 18736882.

J. Jude, M. G. Perich, L. E. Miller, and M. H. Hennig. Robust alignment of cross-session recordings of neural population activity by behaviour via unsupervised domain adaptation, 2022. URL `https://arxiv.org/abs/2202.06159`.

B. M. Karpowicz, Y. H. Ali, L. N. Wimalasena, A. R. Sedler, M. R. Keshtkaran, K. Bodkin, X. Ma, L. E. Miller, and C. Pandarinath. Stabilizing brain-computer interfaces through alignment of latent dynamics. *bioRxiv*, 2022. doi: 10.1101/2022.04.06.487388. URL `https://www.biorxiv.org/content/early/2022/04/08/2022.04.06.487388`.

D. P. Kingma and J. L. Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 2014.

R. Liu, M. Azabou, M. Dabagia, C.-H. Lin, M. G. Azar, K. B. Hengen, M. Valko, and E. L. Dyer. Drop, swap, and generate: A self-supervised approach for generating neural activity. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 9910 LNCS, 2016. doi: 10.1007/978-3-319-46466-4{_}5.

C. Pandarinath, D. O'Shea, J. Collins, R. Jozefowicz, S. Stavisky, J. Kao, E. Trautmann, M. Kaufman, S. Ryu, L. Hochberg, J. Henderson, K. Shenoy, and D. Sussillo. Inferring single-trial neural population dynamics using sequential auto-encoders. *Inferring single-trial neural population dynamics using sequential auto-encoders*, 2017. ISSN 1548-7091. doi: 10.1101/152884.

D. Sussillo, S. D. Stavisky, J. C. Kao, S. I. Ryu, and K. V. Shenoy. Making brain-machine interfaces robust to future neural variability. *Nature Communications*, 7, 2016. ISSN 20411723. doi: 10.1038/ncomms13749.

S. Wen, A. Yin, T. Furlanello, M. G. Perich, L. E. Miller, and L. Itti. Rapid adaptation of brain–computer interfaces to new neuronal ensembles or participants via generative modelling. *Nature Biomedical Engineering*, 2021. ISSN 2157846X. doi: 10.1038/s41551-021-00811-z.

L. N. Wimalasena, J. F. Braun, M. R. Keshtkaran, D. Hofmann, J. Á. Gallego, C. Alessandro, M. C. Tresch, L. E. Miller, and C. Pandarinath. Estimating muscle activation from emg using deep learning-based dynamical systems models. *bioRxiv*, 2021. doi: 10.1101/2021.12.01.470827.

F. Zhu, A. Sedler, H. A. Grier, N. Ahad, M. Davenport, M. Kaufman, A. Giovannucci, and C. Pandarinath. Deep inference of latent dynamics with spatio-temporal super-resolution using selective backpropagation through time. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 2331–2345. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper/2021/file/1325cdae3b6f0f91a1b629307bf2d498-Paper.pdf`.

Y. Ziv, L. D. Burns, E. D. Cocker, E. O. Hamel, K. K. Ghosh, L. J. Kitch, A. El Gamal, and M. J. Schnitzer. Long-term dynamics of ca1 hippocampal place codes. *Nature Neuroscience*, 16(3):264, 2013.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Results, Section 6.
   (b) Did you describe the limitations of your work? [Yes] See Discussion, Section 7.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Perturbation and model instructions in sections 4 and 5 respectively. Code included in Supplement.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] In Section 5 and further implementation details in Appendix A.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See figures in Section 6.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Yes, see details of compute used in Section 7.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 3.
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Code included in supplemental material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Implementation and training details

Below are implementation details for the CAPTIVATE model.

CAPTIVATE		
Parameter	Value	Notes
Neuron Locator Network		Layer Normalisation on all layers
- RNN Units	784 X 3	Stacked Gated Recurrent Unit
- W_{pos} Units	1024 X 3	Non-linear layer
- W_{pos} Dropout	0.5	
- W_{pos} L2 Regularisation	100.0	
Sequential Autoencoder Encoder		
- RNN Units	784 X 3	Stacked Gated Recurrent Unit
- RNN L2 Kernel Regularisation	0.1	
- RNN L2 Recurrent Regularisation	0.1	
- W_{enc} Units	1024 X 3	Non-linear layer
- W_{enc} L2 Regularisation	0.1	
- Latent space dimension	64	
Sequential Autoencoder Generator		
- RNN Units	512 X 3	Stacked Gated Recurrent Unit
- RNN L2 Kernel Regularisation	1.0	
- RNN L2 Recurrent Regularisation	1.0	
- W_{fac} Units	512	Non-linear layer
Training		
- KL divergence weighting (λ_{kl})	0.02 to 1.0	Rising exponentially
- Batch size (Train Neuron Locator)	16	
- Batch size (Train Seq. Autoencoder)	4	
- Learning rate (Train Neuron Locator)	0.0001	Adam Optimizer
- Learning rate (Train Seq. Autoencoder)	0.00001	Adam Optimizer

B Changes in recorded neural activity across sessions

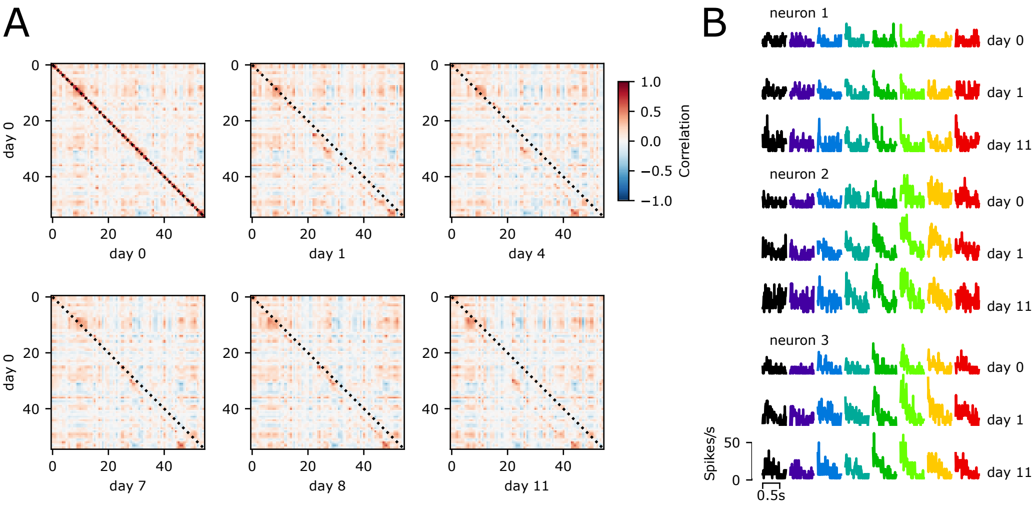

Figure 8: **A**, Pairwise correlations of trial-averaged activity of single neurons between two sessions. For each neuron, the average firing rate was computed for each of the eight movement directions (see part B for examples) at 10ms resolution. The activity for the eight movement directions was concatenated and the Pearson correlation coefficients computed between all neuron pairs. Each plot shows the correlation matrix for activity from session from a different day and activity from the first day (day zero, the training data set in Figure 6). This analysis shows that some neurons from the first session can be matched to neurons recorded at subsequent days, but the relative position of these matched neurons in the recording tends to shift (see high off-diagonal correlations). As the average correlations do not change systematically over this period of time (not illustrated), the gradual changes in neuron identity is a main factor that prevents reliable decoding from unseen sessions in previous models. **B**, Examples of trial-averaged firing rates of three neurons that were tracked over all recording sessions. This matching is based on the similarity of the firing rates, experimentally it is hard to determine if these are indeed the same neurons. In all cases, the time course of the activity is similar and shows consistent differences between trial type (indicated by colour) across sessions. Also note that while these neurons appear to reliably encode movement direction, the activity of a single neuron alone is too noisy to allow for reliable direction decoding from single trials, instead a population decoding approach is required. All data illustrated here is from Monkey C.

C CAPTIVATE accurately maps perturbed neurons and neurons from unseen sessions to known neurons from the Day 0 training session

CAPTIVATE is trained by mapping perturbed trials to known trials. If trials from unseen sessions are similar to the perturbed trials then generalisation to these sessions is possible. Therefore, we aim for the encoder network of CAPTIVATE to map perturbed trials and trials from unseen sessions to day 0 trials. This entails that neurons across unseen sessions (even after neural drift and ensemble change) are mapped directly to neuron positions of the day 0 session at the session. For trials of each movement direction from unseen sessions, we expect that the trial average firing rates of these neurons will map to the day 0 average firing rates for each neuron. As seen below for 4 neurons across 3 sessions (2 unseen), the CAPTIVATE generator network produces trial average firing rates matching the day 0 train session firing rates.

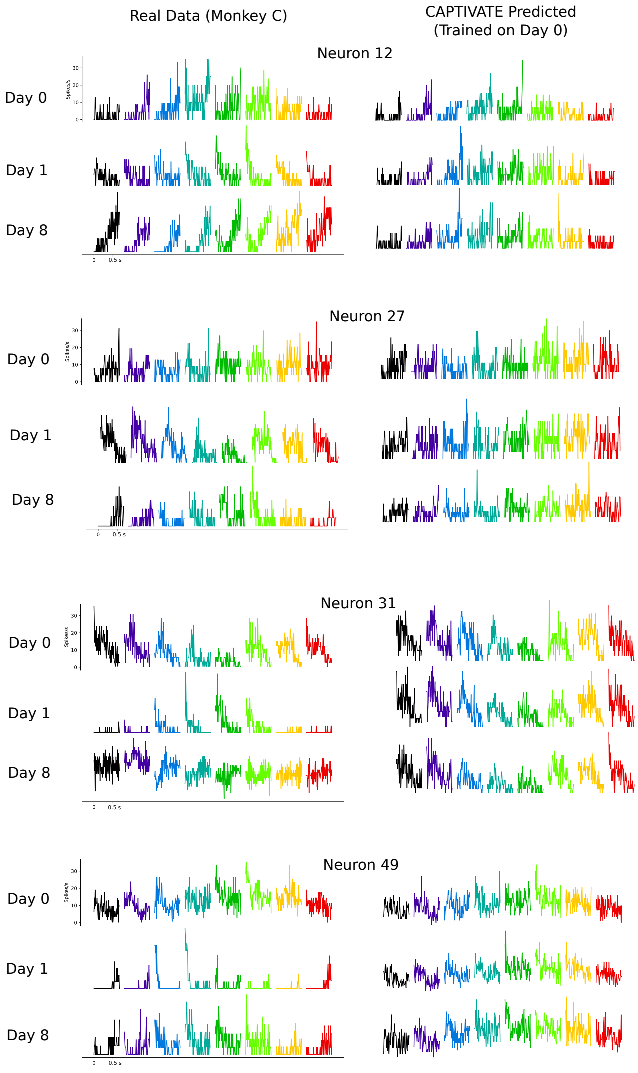

Figure 9: CAPTIVATE is trained on the day 0 session of Monkey C. On the left we show real trial averaged firing rates for each movement direction across 4 randomly selected neurons across the day 0 session and 2 unseen sessions. On the right we show predictions from the generator network of CAPTIVATE. If generalisation is achieved the generator should accurately map neurons across unseen sessions to the neurons of day 0. We see that this is the case as the predicted firing rates are closely matched in the unseen sessions to the day 0 firing rates across movement directions.

D Neuron Locator performance over simulated neural variation

As we do not have ground truth neuron identities from unseen sessions (with respect to the day 0 train session), we simulate inter-session variability by increasing perturbation rate and testing against CAPTIVATE trained on the day 0 session from monkey C with a total perturbation rate of 0.4 (as in the results shown in Figure 6). We see that the neuron locator network of CAPTIVATE can predict neuron identity with 68% accuracy even at a very high total perturbation rate of 1.0.

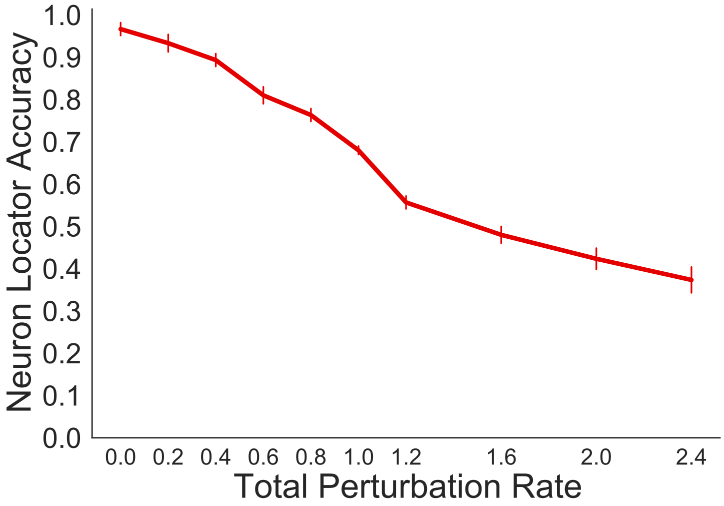

Figure 10: Neuron locator network accuracy when predicting neuron identity (with respect to unperturbed day 0 monkey C train session) as the total rate of perturbation is increased. We are simulating neural drift and ensemble shift across sessions. As we know the ground truth neuron identities, we can assess how well the neuron locator can predict neuron identity.

E Training and testing CAPTIVATE with different numbers of original neurons

Here we test the varying numbers of neurons across sessions of Monkey C when using CAPTIVATE. We see that only 20 neurons are required across sessions for good generalisation for up to 8 days.

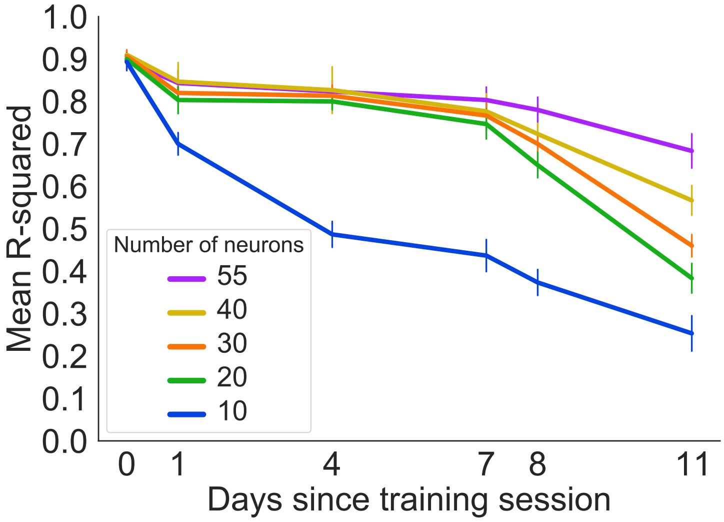

Figure 11: Behaviour prediction performance when training CAPTIVATE on varying numbers of neurons of the day 0 session recorded from Monkey C and testing on all other unseen sessions of monkey C, using the same number of neurons as used in the training session. We also test all neuron number variations of CAPTIVATE on a held out portion of trials from day 0. We report the mean R^2 between the inferred and true x,y positions for the entire movement trajectory of each trial. Each day 0 train session is run 10 times with different random seeds, with error bars showing standard deviation when applied to each unseen session.

F Changing calibration session

Here we show that by training our model on perturbed trials we can generalise to neural drift and recording array movement. CAPTIVATE accounts not only for session variability in the future but also in the past.

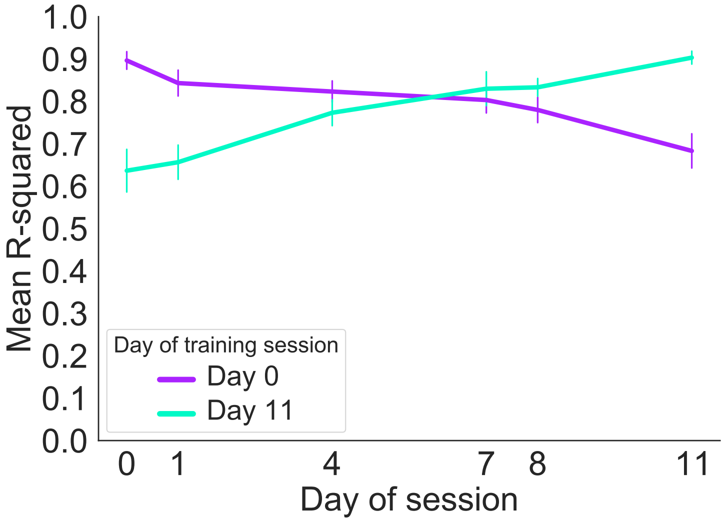

Figure 12: Behaviour prediction performance when separately training CAPTIVATE on the day 0 and day 11 sessions of Monkey C and testing on all other unseen sessions. We see that performance across unseen sessions when training on these sessions is almost symmetrical, indicating that our model can effectively capture neural variability from sessions both backwards and forwards in time.

G Variable neuron number per session

When using an implanted recording array we may lose electrodes or neurons due to spike sorting error over a period of time. Here we show our model can account for this variable neuron number.

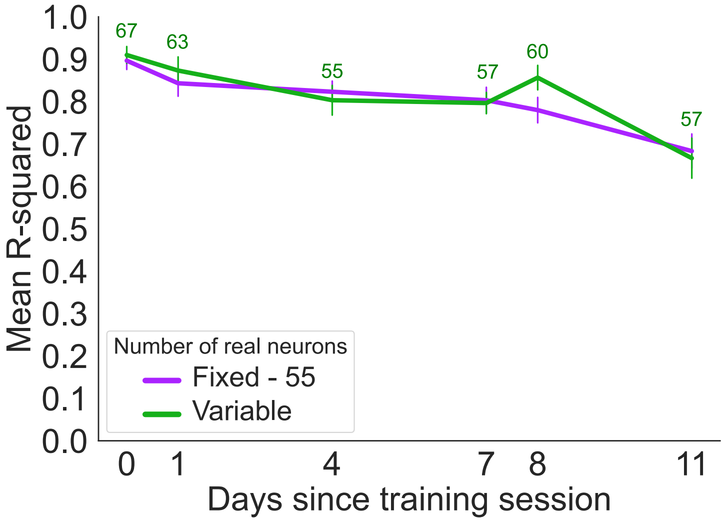

Figure 13: We test CAPTIVATE with a variable number of neurons per session of recording from Monkey C. In our original experiment we only utilise the first 55 neurons of each recording session as this is the minimum number across all sessions. Here we use every neuron available per session (number of neurons per session shown in Figure) and train CAPTIVATE on the day 0 session with 67 neurons. For all other sessions we add randomly generated neurons to compensate. We see that CAPTIVATE is robust to the number of original neurons being variable across sessions. Note the increase in generalisation performance when the model is applied to the day 8 session. This is due to this session having a relatively high number (60) of original neurons, and is thus easier for the model to map trials from this session to known trials from the day 0 training session than it is from other later unseen sessions.

H Testing trained model on known neural variability

We test the resilience of our whole model against an increasing total rate of perturbation in order to ascertain how much variability the model can account for. For the results below, CAPTIVATE is trained with a total perturbation rate of 0.4 on the day 0 session of Monkey C. Note that the model is trained to map perturbed trials to original unperturbed trials.

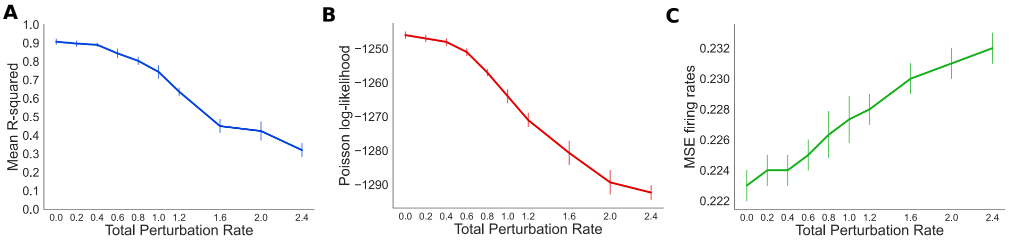

Figure 14: We train CAPTIVATE on trials from the day 0 session of Monkey C with a 0.4 total rate of perturbation on each trial. We then test the trained model on trials of the same session but with increasing rates of total perturbation applied to trials. A) Mean r-squared error of movement predicted from the latent space of the model vs. real movement trajectory of each trial. B) Mean Poisson log-likelihood for neural activity reconstruction by the model generator of original day 0 unperturbed trials. C) Mean squared error of model predicted firing rates vs. real firing rates of original unperturbed day 0 trials.

I Testing trained model on known neural variability across held-out trials

We test the resilience of our whole model against an increasing total rate of perturbation in order to ascertain how much variability the model can account for. For the results below, CAPTIVATE is trained with a total perturbation rate of 0.4 on 70% of the trials of the day 0 session of Monkey C. We show test performance on 30% of the trials of the day 0 session which are withheld from training. Note that the model is trained to map perturbed trials to original unperturbed trials.

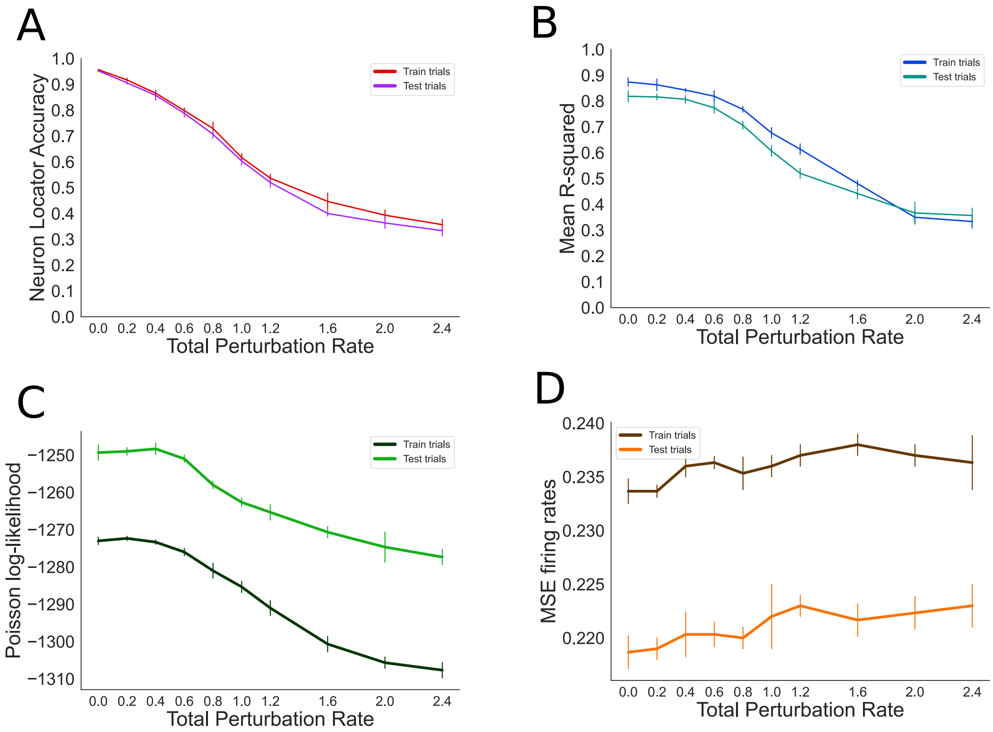

Figure 15: We train CAPTIVATE on 70% of the trials from the day 0 session of Monkey C with a 0.4 total rate of perturbation on each trial. We then test the trained model on the remaining 30% of trials of the same session but with increasing rates of total perturbation applied to these held-out trials. A) Neuron locator network accuracy when predicting neuron identity (with respect to unperturbed day 0 Monkey C train session) as the total rate of perturbation is increased. We are simulating neural drift and ensemble shift across sessions. As we know the ground truth neuron identities, we can assess how well the neuron locator can predict neuron identity. B) Mean r-squared error of movement predicted from the latent space of the model vs. real movement trajectory of each trial. C) Mean Poisson log-likelihood for neural activity reconstruction by the model generator of original day 0 unperturbed trials. D) Mean squared error of model predicted firing rates vs. real firing rates of original unperturbed day 0 trials.

5.3 Discussion

In this chapter I introduce CAPTIVATE, a modelling approach which produces a stable decoder of behaviour from spiking neural activity, capable of generalising for up to a week into the future from a single training session. This is due to the model having captured the variability of neural data between sessions during training, thus the model is able to remain robust to these changes. I additionally discover that non-stationaries which interrupt the continuous ordering of neurons in the monkey datasets (neuron loss and replacement), are most prominent across recording sessions.

Although this is a vast improvement on the work in the previous chapter (SABLE), the perturbations applied to each trial may need to be adjusted if applied to a different species of primate or to a human. Recordings from humans for example, may present more non-stationaries between days as more movement of the recording electrodes from a Utah array is likely, due to a higher degree of freedom for movement of the subject versus monkeys in a laboratory setting who are relatively fixed in place. Even for changes to recording apparatus, adjustments to the perturbations will most likely be required, however it is only possible to tell if this is necessary once this data is collected.

Therefore, the cost of using CAPTIVATE is that the optimal perturbation rate and the types of perturbations will need to be identified, but this is only a one-time cost if subject or recording equipment is constant. These perturbations for a new subject or recording apparatus also need to be learnable by the neuron locator RNN, and so cannot be too complex. If optimal performance were not achieved when training the neuron locator RNN, the ability of the encoder RNN in reversing the perturbations applied to a trial would be severely hampered causing a deterioration in decoding performance.

Furthermore, as with SABLE, CAPTIVATE is also relatively highly parameterised. The neuron locator RNN in particular has three stacked Gated Recurrent Units (GRUs) in order to learn as many variations of the trials in the training session as possible. As mentioned in the paper, the approach is only successful on the monkey with a far higher neuron count. However, a high neuron count does not seem to be a precondition of good decoder generalisation using CAPTIVATE (see Appendix of paper).

Similarly to SABLE, CAPTIVATE is only effective when trained on entire trials of neural activity and cannot be used in online decoding. Although not realistic for BCI systems in its current form, I believe the training approach used in CAPTIVATE will be fruitful in the future. The notion of mapping synthetic variations of neural activity with realistic non-stationaries to ground truth activity could be utilised to stabilise online decoders, and this endeavour is my current research focus.

5.4 Testing efficacy of CAPTIVATE when decoding across subjects

I hypothesise that CAPTIVATE should not be able to generalise well across subjects as the modelling approach described above aims to recognise consistent neurons across datasets (recordings) from the same animal. With separate animals (monkeys), there are clearly no consistent neurons across their recording sessions. However, there may be a subset of neurons in the motor cortex with fairly similar firing patterns across the two monkeys (C and M) when performing similar behaviours.

[76] show that animals of the same species performing the same behaviours have behaviourally relevant neural population latent dynamics which are identifiable across individuals, regardless of the small differences of each individual's brain. Indeed, the modelling approach used in CAPTIVATE (mapping perturbed trials to original trials) creates a high level of stability across sessions of the same animal. High accuracy cross-subject decoding should be possible using our approach if there are a sufficient number of neurons in the motor cortex of monkey C which are recorded from which have coinciding firing patterns across behaviours within the same centre-out reach task as the perturbed neurons (across perturbed trials) of monkey M.

I test how well behaviour from a recording session of monkey C with many trials (1026) can be decoded using CAPTIVATE trained with a single session of monkey M (173 trials). As the dataset for monkey M consists of 17 neurons, we also use 17 neurons from the monkey C session. As seen in Figure 5.2, trials from monkey M are expectedly well separated by behaviour in the latent space of CAPTIVATE. Trials from monkey C (previously unseen) are far less disentangled in this regard and trials pertaining to particular movement directions in the centre-out reach task do not overlap similar trials from monkey M; this is evident in subsequent decoded behaviour (Figure 5.3). Therefore there is only a mean decoding accuracy of 0.224 (R^2) of trials from monkey C.

Although this result is expected, there is some unexpected disentanglement of trials of monkey C in the model latent space, particularly in sets of trials from movement directions which are spatially adjacent (blue, purple, black vs. light green, dark green and

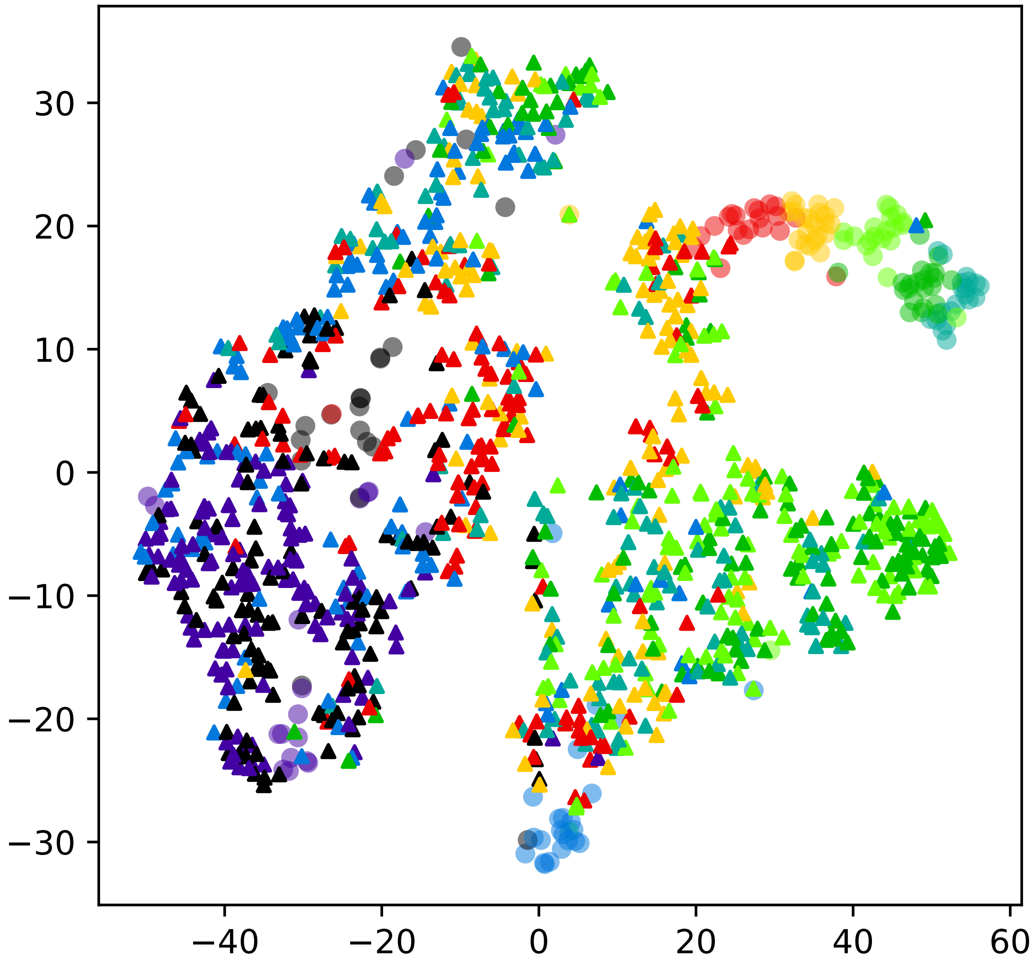

Figure 5.2: t-SNE dimensionality reduced latent space of CAPTIVATE trained on a single session recorded from monkey M and applied to a single unseen session (1026 trials) recorded from monkey C. Dots in each latent space indicates trials from monkey M training session while triangles indicate trials from a unseen monkey C session, with each colour indicating a distinct movement direction in the centre-out reach task.

teal in Figure 5.2). Therefore CAPTIVATE trained on monkey M seems to be somewhat effective at generalisation to other monkeys performing the same task/behaviours, although there appears to be minimal overlap of neurons with similar firing rates across the two monkeys across movement directions. Nonetheless, with a rapidly increasing number of neurons recorded in datasets in the future, this overlap becomes more likely - here only 17 neurons are used across both monkeys therefore there is little possibility of cross-subject neuron similarity across behaviours.

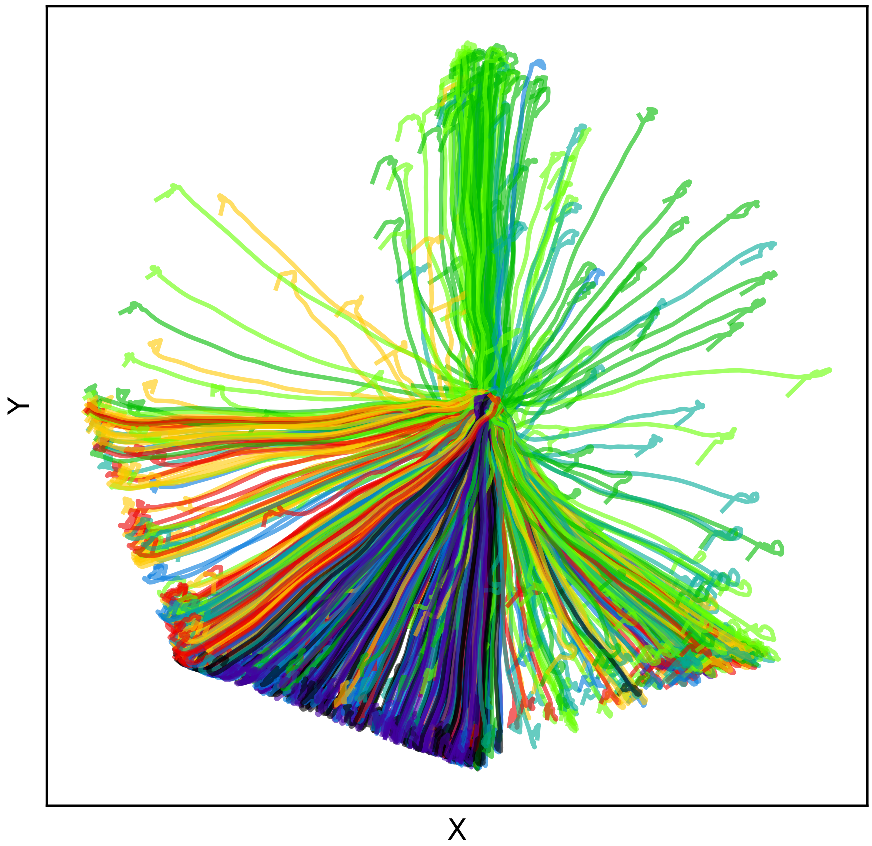

Figure 5.3: Behaviour decoding of CAPTIVATE when applied to an unseen recording session from monkey C after being trained on a single session from monkey M, with each colour indicating a distinct movement direction in the centre-out reach task.

Chapter 6

Conclusion

In this thesis, I introduced three works that present approaches which aim to learn robust representations of neural populations. The first of these demonstrates that just through performing a predictive task during maze traversal, reminiscent of experimental neuroscience, a recurrent neural network model learns a spatially modulated representation of the maze environment. This spatial modulation is in the form of place units. I show that utilisation of this learned representation is advantageous for learning downstream tasks, such as navigating towards reward locations. The neural network units then demonstrate a further multitude of place cell properties, recapitulating many experimental results. This task-optimised model helps to understand more complex brain function in place of other artificial hand crafted models which can be too complicated.

In the second work, I introduced a robust decoder of behaviour from neural activity, brought about through unsupervised domain adaptation. For the first time, I showed that it is possible to train a model capable of generalising to a completely unseen neural recording session. No other preexisting method is capable of this stability without some form of recalibration or retraining. This is achieved through the formation of a stable sequential autoencoder latent space, where the sequential autoencoder is trained using an adversarial technique. Finally, in the third work, I introduced a self-supervised model trained with a contrastive learning approach which aimed to incorporate real neural non-stationaries into the training data, creating a model with a highly stable latent representation. This representation was able to remain constant and thus generalise to neural data from recording sessions up to a week into the future, allowing for an even longer horizon of behaviour decoding without recalibration than the approach presented in the second work. While each of these works were presented and discussed in the context of neural populations, their principles and model architectures can have an impact on other fields of machine learning and on other areas of Neuroscience research.

6.1 Implications of consolidated spatial representations

My work on spatial representations (chapter 3) for use with Q-learning could be useful for future work in reinforcement learning (RL), as this form of learned representation is useful as a succinct foundation for downstream tasks, especially when agents are navigating in generally consistent environments but with dynamic elements or minor changes. Future work should focus on testing the training paradigm with larger, more complex arenas with dynamic elements to see whether such a representation allows for efficient learning of tasks. The RNN model could be used in conjunction with a convolutional neural network so that an artificial agent can receive three dimensional visual input (instead of static wall colours). The RNN could then build a stable spatial representation from random walks of a complex environment with dynamic 3D objects. Additionally, when trained on a more complex environment, the model could be examined for conjunctive responses, where RNN units encode multiple spatial properties simultaneously, such as place and distance to reward.

6.2 Implications and challenges of stable BCI decoding

As previously discussed, my works in stable predictors of behaviour from neural activity have significant implications for brain-computer interfaces (BCIs), through the substantial reduction in the frequency of decoder retraining required for accurate behaviour prediction over prior methods. Stable BCI decoders are even more imperative in human subjects than the laboratory primates which the models in chapters 4 and 5 were tested on. Free moving human subjects tend to have a higher rate of variability in between sessions (or even within sessions) than fixed monkeys. Using current methods, recalibration may be required every hour or so for accurate decoding with human subjects. My work is vital in improving this recalibration frequency. The crucial capacity to learn robust yet informative latent representations of high-dimensional time-series data is additionally applicable to fields such as natural language processing, speech processing and financial modelling.

Chapters 4 and 5 discuss methods (SABLE and CAPTIVATE) which are successfully evaluated on intracortical neural data, proving these approaches are effective when applied to data collected from invasive recording apparatus. However, there are very few instances of invasive recording equipment being implanted in humans, instead, non-invasive techniques such as Electroencephalography (EEG) are much more commonly used. This is due to invasive surgery and monitoring thereafter not being required for EEG recording. Future work could focus on adapting the techniques from chapters 4 and 5 for use with non-invasive recordings - the difficulty here would primarily be that EEG data is not spike based and is highly prone to interference. As previously stated, robust online decoding is a major unsolved problem. Generally, there exists a 100-150ms latency between neural activity and behaviour, so decoding does not need to be strictly real-time - yet there are currently no online approaches which are robust to non-stationaries existing across recording sessions. Future work utilising the domain adaptation and contrastive learning approaches from SABLE and CAPTIVATE for stable (close to) real-time behaviour decoding without calibration from invasive recordings may be more promising with respect to BCI applications in humans in the short-term. In general, these works leave room for iteration with the hope of non-invasive (or semi-invasive) real-time BCI systems which require only infrequent calibration.

6.3 Future potential for cross-subject decoding

In this thesis I test cross-subject decoding using SABLE in chapter 4 (Figure 7 in paper) and using CAPTIVATE in chapter 5 (Figure 5.2). In both circumstances there is little generalisation to the unseen animal's neural activity, resulting in poor behaviour decoding. This is not surprising considering that there is rarely any observed similarity in spiking patterns across animals, despite performing the same consistent behaviours.

However, I believe that CAPTIVATE in particular would benefit in regard to generalisation from a far larger number of recorded neurons across the training and test subjects. This would increase the likelihood of encountering neurons with overlapping spiking patterns across subjects. This is more likely with CAPTIVATE due to the significant perturbations we apply to training data. There is already some behaviourally relevant disentanglement of trials with CAPTIVATE when applied to an unseen animal when training and testing with just 17 neurons (Figure 5.2), so this is not infeasible.

With SABLE, I hypothesise that due to an increasing number of training sessions from the same animal being favourable for generalisation to an unseen session, training SABLE with sessions from several animals will allow the model to capture the cross-subject variability present, potentially allowing for generalisation to an unseen animal. However I believe this to be implausible without a significant increase in the number of parameters used in the SABLE model due to the severity of the variability existing across animals.

SABLE and CAPTIVATE are designed to generalise to unseen neural data without any fine-tuning or recalibration. Other approaches using a large amount of data from across animals [76] show that neural population dynamics are preserved across animals performing similar behaviour. Therefore, a few particular trials from an unseen animal may be effective in sufficiently fine-tuning a model trained entirely on another animal; this could be a more realistic approach to achieve cross-subject decoding. The expected likelihood of a particular pair of neurons across animals with overlapping spiking for all consistent behaviours is minute. Thus an ensemble modelling approach which aims to match neurons across populations (subjects) with similar spiking for even a single consistent behaviour per pair of neurons (if such a pair exists) may be fruitful. Through few-shot learning (one trial for each behaviour), each model of the ensemble could then be used for each behaviour by mapping coinciding neurons, allowing for potential generalisation.

6.4 Nature of representational drift in neural circuits

Representational drift refers to continual changes in firing patterns in single neurons or in a neuron population while corresponding behaviour remains unchanged [73]. Although this drift plays an important role in continual learning [16], it eventually renders neural decoders of behaviour ineffective, often in a matter of hours. In this thesis I have introduced a model (SABLE) which has been shown to account for and overcome this representational drift, permitting good behaviour decoding accuracy for at least for a single future day of recording. CAPTIVATE on the other hand, is designed to account for changes to neuron ensembles across recording sessions and does not explicitly account for this drift, although some of the perturbations utilised may inherently model drift, improving cross-session generalisation.

Although stable behaviour decoding is a central consideration for modelling representational drift, there are many noteworthy applications for the investigation of the nature of this drift. Although drift may appear random, by modelling this drift in whole neuron populations, we could identify the mechanisms by which learning and memory encoding takes place, affirming the need for drift in these large distributed systems [73]. Furthermore, the purpose of high dimensional neural data pertaining to low dimensional behaviours [23] could be better understood. Currently, redundancy is thought to be the central reasoning for this high dimensionality, making neural populations robust to failure in individual neurons and to external factors. However, there is a high energy cost to maintaining many more neurons than is realistically required for each possible behaviour. Therefore modelling neural drift in populations can aid in concretely understanding this phenomenon.

In chapter 3 I look at representations of space, modelled by place units. Within areas of the brain such as the hippocampus, the firing patterns of these place cells need to be relatively fixed in order for efficient navigation of a previously explored environment. However, due to representational drift, these place cells will fire differently over time. Therefore, there must be a mechanism to correct for this drift so that a mammal can navigate in a learned environment in the future. It is thought that internal error signals could detect and correct drift [73], such that plasticity in other regions of the hippocampus be used to compensate for changes in the relevant place cells. By investigating drift across multiple regions of the hippocampus, this mechanism for stable spatial reasoning can be identified.

6.5 Final Statement

I would like to thank the reader for reading my work; I hope my ideas put forward in this thesis are useful to your future research efforts and for the building of effective BCI systems.

References

[1] J. A. Ainge, M. Tamosiunaite, F. Woergoetter, and P. A. Dudchenko. Hippocampal CA1 place cells encode intended destination on a maze with multiple choice points. *Journal of Neuroscience*, 27(36), 2007.

[2] D. Aronov, R. Nevers, and D. W. Tank. Mapping of a non-spatial dimension by the hippocampal-entorhinal circuit. *Nature*, 543(7647), 2017.

[3] M. Azabou, M. G. Azar, R. Liu, C.-H. Lin, E. C. Johnson, K. Bhaskaran-Nair, M. Dabagia, B. Avila-Pires, L. Kitchell, K. B. Hengen, W. Gray-Roncal,

M. Valko, and E. L. Dyer. Mine your own view: Self-supervised learning through across-sample prediction. *arXiv preprint arXiv:2102.10106*, 2021.

[4] M. Ballini, J. Muller, P. Livi, Y. Chen, U. Frey, A. Stettler, A. Shadmani, V. Viswam, I. L. Jones, D. Jackel, M. Radivojevic, M. K. Lewandowska, W. Gong, M. Fiscella, D. J. Bakkum, F. Heer, and A. Hierlemann. A 1024-channel CMOS microelectrode array with 26,400 electrodes for recording and stimulation of electrogenic cells in vitro. *IEEE Journal of Solid-State Circuits*, 49(11), 2014.

[5] A. Banino, C. Barry, B. Uria, C. Blundell, T. Lillicrap, P. Mirowski, A. Pritzel, M. J. Chadwick, T. Degris, J. Modayil, G. Wayne, H. Soyer, F. Viola, B. Zhang, R. Goroshin, N. Rabinowitz, R. Pascanu, C. Beattie, S. Petersen, A. Sadik, S. Gaffney, H. King, K. Kavukcuoglu, D. Hassabis, R. Hadsell, and D. Kumaran. Vector-based navigation using grid-like representations in artificial agents. *Nature*, 2018.

[6] L. Berdondini, P. D. Van Der Wal, O. Guenat, N. F. De Rooij, M. Koudelka-Hep, P. Seitz, R. Kaufmann, P. Metzler, N. Blanc, and S. Rohr. High-density electrode array for imaging in vitro electrophysiological activity. *Biosensors and Bioelectronics*, 21(1), 2005.

[7] E. Bostock, R. U. Muller, and J. L. Kubie. Experience-dependent modifications of hippocampal place cell firing. *Hippocampus*, 1(2), 1991.

[8] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, volume 2017-January, 2017.

[9] Q. Cai, Y. Wang, Y. Pan, T. Yao, and T. Mei. Joint contrastive learning with infinite possibilities. In *Advances in Neural Information Processing Systems*, volume 2020-December, 2020.

[10] J. M. Carmena, M. A. Lebedev, C. S. Henriquez, and M. A. Nicolelis. Stable ensemble performance with single-neuron variability during reaching movements in primates. *Journal of Neuroscience*, 25(46), 2005.

[11] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *37th International Conference on Machine Learning, ICML 2020*, volume PartF168147-3, 2020.

[12] M. M. Churchland, J. P. Cunningham, M. T. Kaufman, J. D. Foster, P. Nuyujukian, S. I. Ryu, K. V. Shenoy, and K. V. Shenoy. Neural population dynamics

during reaching. *Nature*, 487(7405), 2012.

[13] C. J. Cueva, P. Y. Wang, M. Chin, and X.-X. Wei. Emergence of functional and structural properties of the head direction system by optimization of recurrent neural networks. In *International Conference on Learning Representations*, 2020.

[14] C. J. Cueva and X. X. Wei. Emergence of grid-like representations by training recurrent neural networks to perform spatial localization. In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR, 2018.

[15] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1422–1430, 2015.

[16] L. N. Driscoll, L. Duncker, and C. D. Harvey. Representational drift: Emerging theories for continual learning and experimental future directions. *Current Opinion in Neurobiology*, 76:102609, 2022.

[17] N. F. Dronkers. A new brain region for coordinating speech articulation. *Nature*, 384(6605), 1996.

[18] G. F. Elsayed and J. P. Cunningham. Structure in neural population recordings: An expected byproduct of simpler phenomena? *Nature Neuroscience*, 20(9), 2017.

[19] B. Eversmann, M. Jenkner, F. Hofmann, C. Paulus, R. Brederlow, B. Holzapfl, P. Fromherz, M. Merz, M. Brenner, M. Schreiter, R. Gabl, K. Plehnert, M. Steinhauser, G. Eckstein, D. Schmitt-Landsiedel, and R. Thewes. A 128 × 128 CMOS Biosensor Array for Extracellular Recording of Neural Activity. In *IEEE Journal of Solid-State Circuits*, volume 38, 2003.

[20] A. Farshchian, J. A. Gallego, L. E. Miller, S. A. Solla, J. P. Cohen, and Y. Bengio. Adversarial domain adaptation for stable brain-machine interfaces. In *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[21] U. Frey, J. Sedivy, F. Heer, R. Pedron, M. Ballini, J. Mueller, D. Bakkum, S. Hafizovic, F. D. Faraci, F. Greve, K. U. Kirstein, and A. Hierlemann. Switch-matrix-based high-density microelectrode array in CMOS technology. *IEEE Journal of Solid-State Circuits*, 45(2), 2010.

[22] J. A. Gallego, M. G. Perich, R. H. Chowdhury, S. A. Solla, and L. E. Miller. Long-term stability of cortical population dynamics underlying consistent behavior. *Nature Neuroscience*, 23(2), 2020.

[23] J. A. Gallego, M. G. Perich, L. E. Miller, and S. A. Solla. Neural Manifolds for the Control of Movement. *Neuron*, 94(5), 2017.

[24] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *32nd International Conference on Machine Learning, ICML 2015*, volume 2, 2015.

[25] J. I. Glaser, A. S. Benjamin, R. H. Chowdhury, M. G. Perich, L. E. Miller, and K. P. Kording. Machine learning for neural decoding. *eNeuro*, 7(4), 2020.

[26] D. Gonschorek, L. Höfling, K. P. Szatko, K. Franke, T. Schubert, B. A. Dunn, P. Berens, D. A. Klindt, and T. Euler. Removing inter-experimental variability from functional data in systems neuroscience. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[27] A. L. Griffin, H. Eichenbaum, and M. E. Hasselmo. Spatial representations of hippocampal CA1 neurons are modulated by behavioral context in a hippocampus-dependent memory task. *Journal of Neuroscience*, 27(9), 2007.

[28] J. B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko. Bootstrap your own latent a new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, volume 2020-December, 2020.

[29] Z. D. Guo, B. A. Pires, B. Piot, J. B. Grill, F. Altché, R. Munos, and M. G. Azar. Bootstrap latent-predictive representations for multitask reinforcement learning. In *37th International Conference on Machine Learning, ICML 2020*, volume PartF168147-5, 2020.

[30] T. Hafting, M. Fyhn, S. Molden, M. B. Moser, and E. I. Moser. Microstructure of a spatial map in the entorhinal cortex. *Nature*, 436(7052), 2005.

[31] T. M. Hall, K. Nazarpour, and A. Jackson. Real-time estimation and biofeedback of single-neuron firing rates using local field potentials. *Nature Communications*, 5, 2014.

[32] R. R. Harrison, P. T. Watkins, R. J. Kier, R. O. Lovejoy, D. J. Black, B. Greger, and F. Solzbacher. A low-power integrated circuit for a wireless 100-electrode neural recording system. *IEEE Journal of Solid-State Circuits*, 42(1), 2007.

[33] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2020.

[34] G. E. Hinton. *A Practical Guide to Training Restricted Boltzmann Machines*,

pages 599–619. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[35] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 1997.

[36] T. Hosman, M. Vilela, D. Milstein, J. N. Kelemen, D. M. Brandman, L. R. Hochberg, and J. D. Simeral. BCI decoder performance comparison of an LSTM recurrent neural network and a Kalman filter in retrospective simulation. In *International IEEE/EMBS Conference on Neural Engineering, NER*, volume 2019-March, 2019.

[37] C. Hurwitz, N. Kudryashova, A. Onken, and M. H. Hennig. Building population models for large-scale neural recordings: Opportunities and pitfalls. *Current Opinion in Neurobiology*, 70, 2021.

[38] C. Hurwitz, A. Srivastava, K. Xu, J. Jude, M. Perich, L. Miller, and M. Hennig. Targeted neural dynamical modeling. *Advances in Neural Information Processing Systems*, 34, 2021.

[39] A. Johnson and A. D. Redish. Neural ensembles in CA3 transiently encode paths forward of the animal at a decision point. *Journal of Neuroscience*, 27(45), 2007.

[40] J. Jude and M. H. Hennig. Hippocampal representations emerge when training recurrent neural networks on a memory dependent maze navigation task. *arXiv preprint arXiv:2012.01328*, 2020.

[41] J. Jude, M. Perich, L. Miller, and M. Hennig. Robust alignment of cross-session recordings of neural population activity by behaviour via unsupervised domain adaptation. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 10462–10475. PMLR, 17–23 Jul 2022.

[42] J. Jude, M. G. Perich, L. E. Miller, and M. H. Hennig. Capturing cross-session neural population variability through self-supervised identification of consistent neuron ensembles. In *Proceedings of the NeurIPS Workshop on Symmetry and Geometry in Neural Representations*, volume 197 of *Proceedings of Machine Learning Research*. PMLR, Dec 2022.

[43] J. J. Jun, N. A. Steinmetz, J. H. Siegle, D. J. Denman, M. Bauza, B. Barbarits, A. K. Lee, C. A. Anastassiou, A. Andrei, Aydin, M. Barbic, T. J. Blanche, V. Bonin, J. Couto, B. Dutta, S. L. Gratiy, D. A. Gutnisky, M. Häusser, B. Karsh, P. Ledochowitsch, C. M. Lopez, C. Mitelut, S. Musa, M. Okun, M. Pachitariu, J. Putzeys, P. D. Rich, C. Rossant, W. L. Sun, K. Svoboda, M. Carandini, K. D.

Harris, C. Koch, J. O'Keefe, and T. D. Harris. Fully integrated silicon probes for high-density recording of neural activity. *Nature*, 551(7679), 2017.

[44] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering, Transactions of the ASME*, 82(1), 1960.

[45] B. M. Karpowicz, Y. H. Ali, L. N. Wimalasena, A. R. Sedler, M. R. Keshtkaran, K. Bodkin, X. Ma, L. E. Miller, and C. Pandarinath. Stabilizing brain-computer interfaces through alignment of latent dynamics. *bioRxiv*, 2022.

[46] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 2014.

[47] N. Kriegeskorte. Deep Neural Networks: A New Framework for Modeling Biological Vision and Brain Information Processing. *Annual Review of Vision Science*, 1(1), 2015.

[48] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 1998.

[49] I. Lee, A. L. Griffin, E. A. Zilli, H. Eichenbaum, and M. E. Hasselmo. Gradual Translocation of Spatial Correlates of Neuronal Firing in the Hippocampus toward Prospective Reward Locations. *Neuron*, 51(5):639–650, 2006.

[50] C. Lever, S. Burton, A. Jeewajee, J. O'Keefe, and N. Burgess. Boundary vector cells in the subiculum of the hippocampal formation. *Journal of Neuroscience*, 29(31), 2009.

[51] M. S. Lewicki. A review of methods for spike sorting: The detection and classification of neural action potentials. *Network: Computation in Neural Systems*, 9(4), 1998.

[52] S. Li, J. Li, and Z. Li. An improved unscented kalman filter based decoder for cortical brain-machine interfaces. *Frontiers in Neuroscience*, 10(DEC), 2016.

[53] Z. Li, J. E. O'Doherty, T. L. Hanson, M. A. Lebedev, C. S. Henriquez, and M. A. L. Nicolelis. Unscented kalman filter for brain-machine interfaces. *PLOS ONE*, 4(7):1–18, 07 2009.

[54] Q. Liao and T. Poggio. Bridging the gaps between residual learning, recurrent neural networks and visual cortex. *arXiv preprint arXiv:1604.03640*, 2016.

[55] R. Liu, M. Azabou, M. Dabagia, C.-H. Lin, M. G. Azar, K. B. Hengen, M. Valko, and E. L. Dyer. Drop, swap, and generate: A self-supervised approach for generating neural activity. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[56] C. M. Lopez, S. Mitra, J. Putzeys, B. Raducanu, M. Ballini, A. Andrei, S. Severi, M. Welkenhuysen, C. Van Hoof, S. Musa, and R. F. Yazicioglu. A 966-electrode neural probe with 384 configurable channels in 0.13µm SOI CMOS. In *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*, volume 59, 2016.

[57] J. G. Makin, J. E. O'Doherty, M. M. Cardoso, and P. N. Sabes. Superior arm-movement decoding from cortex with a new, unsupervised-learning algorithm. *Journal of Neural Engineering*, 15(2), 2018.

[58] O. Mamad, L. Stumpp, H. M. McNamara, C. Ramakrishnan, K. Deisseroth, R. B. Reilly, and M. Tsanov. Place field assembly distribution encodes preferred locations. *PLoS Biology*, 15(9), 2017.

[59] V. Mante, D. Sussillo, K. V. Shenoy, and W. T. Newsome. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, 503(7474), 2013.

[60] J. Müller, M. Ballini, P. Livi, Y. Chen, M. Radivojevic, A. Shadmani, V. Viswam, I. L. Jones, M. Fiscella, R. Diggelmann, A. Stettler, U. Frey, D. J. Bakkum, and A. Hierlemann. High-resolution CMOS MEA platform to study neurons at subcellular, cellular, and network levels. *Lab on a Chip*, 15(13), 2015.

[61] R. U. Muller and J. L. Kubie. The effects of changes in the environment on the spatial firing of hippocampal complex-spike cells. *Journal of Neuroscience*, 7(7), 1987.

[62] E. Musk. An integrated brain-machine interface platform with thousands of channels. *Journal of Medical Internet Research*, 21(10), 2019.

[63] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 9910 LNCS, 2016.

[64] J. O' Keefe and N. Burgess. Geometric determinants of the place fields of hippocampal neurons. *Nature*, 381(6581):425–428, 5 1996.

[65] J. O'Keefe. Place units in the hippocampus of the freely moving rat. *Experimental Neurology*, 51(1), 1976.

[66] J. O'Keefe and J. Dostrovsky. The Hippocampus as a Spatial Map. *Brain Research*, 1971.

[67] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[68] C. Pandarinath, D. O'Shea, J. Collins, R. Jozefowicz, S. Stavisky, J. Kao, E. Trautmann, M. Kaufman, S. Ryu, L. Hochberg, J. Henderson, K. Shenoy, and D. Sussillo. Inferring single-trial neural population dynamics using sequential auto-encoders. *Inferring single-trial neural population dynamics using sequential auto-encoders*, 2017.

[69] L. Perez and J. Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

[70] S. Recanatesi, M. Farrell, G. Lajoie, S. Deneve, M. Rigotti, and E. Shea-Brown. Signatures of low-dimensional neural predictive manifolds. *Cosyne Abstracts 2019, Lisbon, PT.*, 2019.

[71] S. Recanatesi, M. Farrell, G. Lajoie, S. Deneve, M. Rigotti, and E. Shea-Brown. Predictive learning as a network mechanism for extracting low-dimensional latent space representations. *Nature Communications*, 12(1), 2021.

[72] N. T. Robinson, L. A. Descamps, L. E. Russell, M. O. Buchholz, B. A. Bicknell, G. K. Antonov, J. Y. Lau, R. Nutbrown, C. Schmidt-Hieber, and M. Häusser. Targeted Activation of Hippocampal Place Cells Drives Memory-Guided Spatial Behavior. *Cell*, 183(6), 2020.

[73] M. E. Rule, T. O'Leary, and C. D. Harvey. Causes and consequences of representational drift. *Current Opinion in Neurobiology*, 58, 2019.

[74] V. M. Rutten, A. Bernacchia, M. Sahani, and G. Hennequin. Non-reversible Gaussian processes for identifying latent dynamical structure in neural data. In *Advances in Neural Information Processing Systems*, volume 2020-December, 2020.

[75] P. T. Sadtler, K. M. Quick, M. D. Golub, S. M. Chase, S. I. Ryu, E. C. Tyler-Kabara, B. M. Yu, and A. P. Batista. Neural constraints on learning. *Nature*, 512(7515), 2014.

[76] M. Safaie, J. C. Chang, J. Park, L. E. Miller, J. T. Dudman, M. G. Perich, and J. A. Gallego. Preserved neural population dynamics across animals performing similar behaviour. *bioRxiv*, 2022.

[77] O. G. Sani, H. Abbaspourazad, Y. T. Wong, B. Pesaran, and M. M. Shanechi. Modeling behaviorally relevant neural dynamics enabled by preferential subspace identification. *Nature Neuroscience*, 24(1), 2021.

[78] M. Schwarzer, A. Anand, R. Goel, R. D. Hjelm, A. Courville, and P. Bachman. Data-efficient reinforcement learning with self-predictive representations. *arXiv preprint arXiv:2007.05929*, 2020.

[79] A. C. Smith and E. N. Brown. Estimating a state-space model from point process observations. *Neural Computation*, 15(5), 2003.

[80] J. Song and S. Ermon. Multi-label contrastive predictive coding. In *Advances in Neural Information Processing Systems*, volume 2020-December, 2020.

[81] K. L. Stachenfeld, M. M. Botvinick, and S. J. Gershman. The hippocampus as a predictive map. *Nature Neuroscience*, 20(11), 2017.

[82] C. Sun, W. Yang, J. Martin, and S. Tonegawa. Hippocampal neurons represent events as transferable units of experience. *Nature Neuroscience*, 23(5), 2020.

[83] D. Sussillo, P. Nuyujukian, J. M. Fan, J. C. Kao, S. D. Stavisky, S. Ryu, and K. Shenoy. A recurrent neural network for closed-loop intracortical brain-machine interface decoders. *Journal of Neural Engineering*, 9(2), 2012.

[84] D. Sussillo, S. D. Stavisky, J. C. Kao, S. I. Ryu, and K. V. Shenoy. Making brain-machine interfaces robust to future neural variability. *Nature Communications*, 7, 2016.

[85] K. Svoboda and N. Li. Neural mechanisms of movement planning: motor cortex and beyond. *Current Opinion in Neurobiology*, 49, 2018.

[86] E. C. Tolman. Cognitive maps in rats and men. *Psychological Review*, 55(4), 1948.

[87] B. Uria, B. Ibarz, A. Banino, V. Zambaldi, D. Kumaran, D. Hassabis, C. Barry, and C. Blundell. The spatial memory pipeline: a model of egocentric to allocentric understanding in mammalian brains. *bioRxiv*, 2020.

[88] S. Wen, A. Yin, T. Furlanello, M. G. Perich, L. E. Miller, and L. Itti. Rapid adaptation of brain–computer interfaces to new neuronal ensembles or participants via generative modelling. *Nature Biomedical Engineering*, 2021.

[89] B. M. Yu, J. P. Cunningham, G. Santhanam, S. I. Ryu, K. V. Shenoy, and M. Sahani. Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity. *Journal of Neurophysiology*, 102(1), 2009.

[90] X. Yuan, S. Kim, J. Juyon, M. D'Urbino, T. Bullmann, Y. Chen, A. Stettler, A. Hierlemann, and U. Frey. A microelectrode array with 8,640 electrodes enabling simultaneous full-frame readout at 6.5 kfps and 112-channel switch-matrix readout at 20 kS/s. In *IEEE Symposium on VLSI Circuits, Digest of Technical Papers*, volume 2016-September, 2016.

[91] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2017-October, 2017.

[92] T. Zwickel, T. Wachtler, and R. Eckhorn. Coding the presence of visual objects in a recurrent neural network of visual cortex. *BioSystems*, 89(1-3), 2007.