©Copyright 2021
Satpreet H. Singh

Machine Learning methods to enable Naturalistic Neuroscience and Neuroengineering

Satpreet H. Singh

A dissertation submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2021

Reading Committee:

Bingni W. Brunton, Chair

Rajesh P. N. Rao, Chair

Eric Shea-Brown

Program Authorized to Offer Degree:
Electrical and Computer Engineering

University of Washington

Abstract

Machine Learning methods to enable
Naturalistic Neuroscience and Neuroengineering

Satpreet H. Singh

Co-Chairs of the Supervisory Committee:
Associate Professor Bingni W. Brunton
Department of Biology

Professor Rajesh P. N. Rao
Department of Computer Science and Engineering, and
Department of Electrical and Computer Engineering

Recent advances in brain recording technology and in algorithms for analyzing behavioral data are enabling the study of neural activity underlying spontaneous behaviors. This new paradigm called "naturalistic neuroscience" goes beyond the confines of traditional neuroscience experiments that rely on cued, repeated trials, and *a priori* experimental design. In this dissertation, we describe how we use machine-learning to study increasingly naturalistic behaviors and associated neural data in two distinctly different settings.

In the first "data-rich" setting, we study large (≈250 GB/subject) opportunistically-collected datasets of simultaneously recorded long-term (7–10 day) electrocorticography and naturalistic behavior video data for 12 human subjects. Our approach uncovers and annotates thousands of instances of human upper-limb movement events from the video recordings, using a pipeline comprising computer-vision, discrete latent-variable modeling, string pattern-matching and event metadata extraction. We curate these events into a database that can be used for many downstream applications in neuroscience and neuroengineering, two of which we prototype – (1) time-frequency analysis and (2) movement initiation decoding. We have published our curated dataset (https://github.com/BruntonUWBio/mining2021), making available a resource that captures naturalistic neural and behavioral

variability at a scale not previously available.

In the second "simulation-based" setting, we study plume-tracking, a complex control problem requiring multimodal sensory integration and robustness to odor intermittency, wind non-stationarity and spatiotemporal plume variability. Flying insects routinely track plumes, often over long distances, in pursuit of odors originating from food or mates. Isolated aspects of this remarkable behavior have been studied in detail in many experimental studies. We take a complementary *in silico* approach, using artificial agents trained with reinforcement learning, to develop an integrated understanding of the behaviors and neural computations that support plume tracking. Specifically, we use deep reinforcement learning to train recurrent neural-network (RNN) based agents to locate the source of simulated turbulent plumes. Interestingly, the agents' emergent behaviors resemble those of flying insects, and the RNNs learn to represent task-relevant variables such as head-direction and time between odor encounters. Exploiting the simulator's flexibility and the full observability of the RNNs' neural activity, we also generate insights into behavior modularity, memory capacity, neural computations and network connectivity that support plume tracking in a variety of easily simulated but hard-to-realize plume configurations. Our *in silico* approach provides key intuitions for an integrated understanding of turbulent plume tracking and motivates future targeted experimental and theoretical developments. Animations accompanying this study can be found at https://github.com/BruntonUWBio/plumetracknets.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First, I would like to thank to my advisors, Drs. Bingni W. Brunton and Rajesh P.N. Rao, for all the effort and time they invested in guiding me; and also for the generous funding of my work and travels to share my work with the broader research community. Thanks especially to Bing for her attention to detail in editing my manuscripts, and for holding me to high standards in communicating my work. Thanks to Raj for sending me reading suggestions, and for bringing the depth and breath of his extensive experience to our discussions. (Thanks also to Raj and Dr. Adrienne Fairhall for their Coursera MOOC on Computational Neuroscience).

Thanks to my PhD committee Drs. Eric Shea-Brown and Daniela M. Witten for their feedback on my work and thought provoking questions! Thanks also to Eric for leading the UW Computational/Theoretical Neuroscience Journal Club. Attending the C/TNJC was an important component of my training at UW; thanks also for the opportunity to co-lead (with Doris Voina) the C/TNJC in 2019-2020. Thanks to Daniela for writing ISLR (which took me from novice to competent in data analysis even before I started grad school). Thanks to Dr. Eli Shlizerman for hosting me in his lab for an important rotation that exposed me to the applications of RNNs in Neuroscience; and to Dr. Charles DeLahunt for initial impressions of the UW ECE department that made me select this program.

Thanks to everyone in the Rao and Brunton labs for being great company! Thanks Seth and Ellie for the memorable trip to Vancouver! Thanks Kam and Biraj for doing the Kernels reading group! Thanks Preston and Pierre for lab meetings that became journal-club sessions! Thanks Alison Mehravari for, among many other things, organizing the NCEC 2018 where I had a discussion with Dr. Ila Fiete (which raised my conviction to pursue the Deep-RL project).

Thanks to my UW friends Sarah Li, Daniel Tabas, Daniel Calderone, Daniel Zdeblick, Danylo Malyuta, Ben Chasnov, Emma Hansen, Thomas Mohren, Eleanor Lutz, Sage Mallingen, Tanner Fiez, Kenji Lee, Megan Morrison, Momona Yamagami, Maneeshika Madduri, Jimin Kim, Narendra Shivaraman, Thomas Miesen, Amanda Chin, Kayla Kuchta for their company and all the fun/serious conversations! Thanks to the many super smart people I met at the UW CNJC, especially those who co/presented papers with me: Alex Hornof, Allison Duffy, Merav Stern, AJ Kruse, Suzanne Lewis, and Mudi Qin.

For the human behavior mining manuscript(s), thanks to Dr. Steve M. Peterson for all his help with processing the brain data, IRB work, and all-round support on the ECoG project; thanks to him and Zoe Steine-Hanson for using the generated dataset to write follow-up manuscripts. Thanks to John So for extensive help with manual annotation of the video data, and Nathan Davis for help fixing a bug with a video annotation script. Thanks to Nancy X.R. Wang for her groundwork towards study approval, initial clinical data procurement, preprocessing, manual annotation, and establishing the plausibility of movement initiation prediction using a subset of this clinical data. Thanks to neurosurgeons Dr. Jeffrey G. Ojemann and Dr. Andrew Ko, and the staff and consenting patients at the University of Washington Harborview Medical Center in Seattle, for supporting this research. Thanks to Pierre Karashchuk, Dr. Kameron D. Harris, Dr. James Wu, Dr. Nile Wilson, Dr. Ariel Rokem, Renshu Gu, David J. Caldwell, and Preston Jiang for helpful discussions and suggestions. For the plume tracking agents manuscript, much thanks to Dr. Floris van Breugel for bringing the breadth and depth of his expertise studying flying insects to this project! Thanks to Aravind Rajeswaran, Stefano Recanatesi, Scott Sterrett, Niru Maheswaranathan, and Steve Brunton for helpful comments and discussions.

Thanks to my parents, family and extended family for their patience and support. My deepest gratitude to some important academic teachers in my life: Mrs. Jyoti Naik, Mrs. Shyamala Shrinivas, Mrs. C. Daniels, Mr. Udayan Kanade, Prof. Kasturi

R. Varadarajan, Prof. Aditya Thakur, Prof. Weng-Keen Wong, Prof. Alan Fern, and Prof. Lillian Ratliff; and to some important mentors during my time in industry: Ken Dallmeyer, Evan Estola, Dr. Wai Gen Yee, Dr. Pascal Collet, Mz. Megan Hughes, Dr. Suman Sundaresh, Dr. Nikita Lytkin, Dr. Wayne Tai Lee, and Dr. Deepak K Kumar

Funding acknowledgements for individual projects are in the respective manuscripts.

Chapter 1

INTRODUCTION

1.1 Naturalistic Neuroscience

Much of our understanding in neuroscience comes from carefully designed, controlled experimental tasks, where timings of cues, stimuli, and behavioral responses are known precisely. However, neuroscientists have always been interested in studying the behavior of neural systems when their experimental subjects are engaged in spontaneous uncued behavior. Recent technological advances in the recording and processing of brain and behavior data have enabled investigation of such increasingly naturalistic behavior, giving rise to a new paradigm called "naturalistic neuroscience" [1–6].

1.2 Key motivations for Naturalistic Neuroscience

A few limitations of traditional controlled studies, that motivate the research community's interest in naturalistic neuroscience, are as follows [1, 7, 8]:

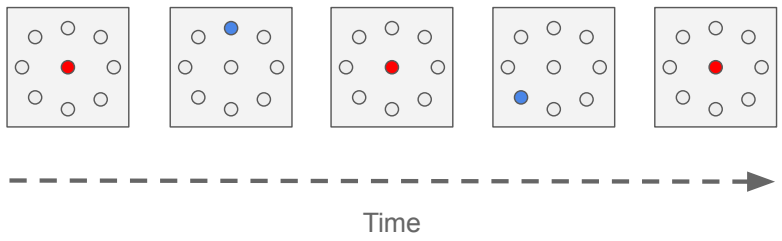

Figure 1.1: Rough illustration of the "center-out task", a traditional motor neuroscience/neuroengineering experimental paradigm: The visual display (grey squares) flashes visual prompts at pre-specified times. The prompt with the red light indicates that the subject should move (or imagine moving) their hand to a 'center/rest' position. The prompt with the blue light indicates that the subject should move (or imagine moving) their hand 'out' in the direction indicated by the light. Behavioral responses and associated neural activity of the subject to these cues are recorded and analyzed, typically assuming statistical independence between each 'center-out' trial.

Limited behavioral repertoire: Though pre-planned controlled experiments are the gold standard and have added much to our understanding in behavioral neuroscience, they can sometimes introduce rigid and artificial constraints on a subject's behavior. Consider a "center-out" task (Figure 1.1), which is a typical paradigm for collecting experimental data

for Brain Computer Interfacing (BCI) applications. Since the subject behaves in response to pre-specified cues, the paradigm largely fails to account for the rich behavioral variability found in spontaneous movements performed during everyday activities. Furthermore, such traditional experiments typically assume statistical independence between each stimulus-followed-by-response trial. This is very different from the continuous sequences of sensory-cognitive-motor activity loops found in spontaneous real-world behavior (that also often contains long-range temporal dependencies) [9]. Models built on neural data collected from such tasks assume inter-trial independence, and largely discard the complex neural variability arising from long-range correlations as (unmodelled) process 'noise' [1].

Generalization: Researchers interested in extracting broad principles and building robust models from data collected in controlled settings have always had to grapple with questions of whether their experimental data will generalize to real-world situations. Specifically, the repeated stimuli used in controlled experiments are unlikely to be found in everyday conditions because one rarely "steps into the same river twice" in real life [8]. This can be especially concerning in the world of BCI, where neural activity decoders need to be robust to real-world signal and noise distributions to meet user-experience expectations. Previously explored solutions to the generalization problem include designing for robustness (e.g. through *adversarial training* [10]) and using adaptive algorithms that track signal distribution changes (e.g. through *adaptive decoding* [11]).

Subject fatigue or boredom: Another complication arising from repeated stimuli used in controlled experiments is that of subject fatigue or boredom. Though researchers design experiments with an eye to keep subjects engaged throughout the course of their experiments, this can become especially concerning when a large number of trials need to be collected.

1.3 Key challenges in Naturalistic Neuroscience (and some solutions)

While Naturalistic Neuroscience experiments address some limitations of traditional experimental paradigms, they bring with them new challenges that demand new approaches to

data collection and analysis. We describe some key challenges below:

Brain recording technology: Traditionally, collecting data from subjects that are mobile or spontaneously behaving has been a significant obstacle. However, technological advances are making such endeavors increasingly within reach. Some recent advances in recording from *human* brains while subjects are mobile include: mobile electroencephalography (EEG) [12], mobile magnetoencephalography (MEG) [13] and mobile functional Near Infrared Spectroscopy (fNIRS) [14]. Among *non-human animals*, such brain recording technologies are perhaps the most developed for smaller model organisms such as rodents [15], zebrafish [16] and nematodes [17]. On the other hand, there relatively limited neural recording technologies for insects engaged in free-form flight [18]. New technologies for high-bandwidth recording from untethered freely behaving non-human primates have recently been demonstrated, with human BCI clinical-trials on the horizon at the time of writing [19].

In Chapter 3, we use intracranial brain recordings from patients undergoing pre-surgical observation at a hospital's epilepsy monitoring unit. These patients have been implanted with one or more electrode arrays which are connected via a tether to a mobile brain-recording interface. Though confined to their bed and its immediate surroundings, these patients are free to perform their daily activities activities (such as talking, eating, meeting people, watching TV, using a computer or phone, sleeping, etc.) during the observation period.

Behavior and subject pose recording technology: Motion capture is a mature but somewhat expensive technology that has been widely applied to both human and non-human subjects [20]. It has traditionally relied on placing markers on subject's bodies, an intervention that can interfere with the spontaneity with which subjects behave. Recently, there has been a flurry of research activity around reducing the cost of motion capture using off-the-shelf cameras, and using markerless pose estimation techniques that minimally interfere with the subject's behavior [21, 22].

Data analysis methods: Most traditional data analysis techniques rely on the availability

of data structured in the form of temporally well-segmented trials, with an assumption of independence between (or easily de-biased) trials. As discussed earlier, such regularities rarely hold in messy real world data. An active emerging research area, called "computational ethology", has recently developed around the challenges arising in the automated analysis of such naturalistic animal behavior [5, 21, 23–26] (see also [27, 28] for a recent surveys and [4] for a perspective). A typical *non-human* naturalistic neuroscience experiment [6, 29, 30] first collects simultaneously recorded behavioral video and neural activity data from one or more freely behaving subjects in an uncontrolled but sufficiently confined environment. Next, the video recordings are processed through an extensive pipeline consisting of steps such as: segmenting the subject(s) from the background, transforming subject pose to common coordinates using affine transformations, estimating pose of body-parts across frames, and higher-level operations such as classifying pose or segmenting pose into actions. Combined with the simultaneously recorded neural data, such naturalistic behavior data are being used to shed light on previously intractable questions in behavioral neuroscience, often at unprecedented scale.

Human action-recognition methods from mainstream computer vision [31] are relevant but not directly applicable to the needs of naturalistic human neuroscience. Traditionally, action-recognition research has concerned itself with discriminating activities at a coarse level, such as sitting vs. walking [32], and has often assumed the availability of a large corpus of labeled training data. In contrast, to study the kinds of behaviors that interest neuroscientists and neuroengineers, we seek to localize fine-grained movements to sub-second temporal resolution, and ideally use the fewest behavioral labels possible [33].

In Chapter 3, we present our automated analysis pipeline for simultaneously recorded *human* brain and behavior data. Since it is not known which behaviors or behavioral characteristics will elicit neural responses worth studying further, we introduce a novel intermediate queryable representation that is flexible enough to support the study of several kinds of behaviors. Recent work by [34] develops such intermediate representations for semi-automated exploration of scenes in general videos across several computer-vision applications.

In Chapter 5, we 'dog-train' the behavior of artificial agents (using some minimalistic assumptions) to find that complex behavior resembling that of flying insects emerges after the training process. We then use behavioral assays across multiple simulated environments, each presenting a different challenge, to understand and characterize the behavior of the agents.

1.4 Organization

The rest of this dissertation is organized as follows:

In Chapter 2, we provide brief summaries of the two studies that have been covered in extensive detail in the rest of this dissertation.

In Chapter 3, we describe how we analyzed large-scale human brain and behavior datasets obtained opportunistically from epilepsy patients undergoing clinical observation prior to epilepsy resection surgery. The data processing pipeline developed in this chapter produces a database of naturalistic upper-limb events, each with associated metadata, that can be used for a variety of downstream applications.

In Chapter 4, we use the naturalistic upper-limb events extracted in Chapter 3 to prototype two downstream applications – (1) a traditional time-frequency analysis of neural activity associated with upper-limb movement-initiation and (2) an upper-limb movement-initiation decoder.

In Chapter 5, we describe an artificial neural network 'agent' trained to perform a challenging control task called plume tracking. We then explore what the agent's behavior and (artificial) neural activity can teach us about how flying insects might perform plume tracking.

Finally, in Chapter 6 we conclude this dissertation with reflections on the aforementioned studies and naturalistic neuroscience.

1.5 Declaration

The work presented in this dissertation is heavily sourced from papers, preprints, abstracts and posters authored by me that have either already been published or will be published soon. This includes:

• SH Singh, F van Breugel, RPN Rao, and BW Brunton, "Emergent behavior and neural dynamics in artificial agents tracking turbulent plumes", Preprint, Sept 2021 [35].

• SH Singh, F van Breugel, RPN Rao, BW Brunton, "Understanding biological plume tracking behavior using deep reinforcement-learning", CSHL Meeting: From Neuroscience to Artificially Intelligent Systems (NAISys), November 2020.

• SH Singh, F van Breugel, RPN Rao, BW Brunton, "Using artificial neural-network agents to dissect biological plume tracking behavior", IROS 2020 Workshop on Robotics-inspired Biology, October 2020.

• SH Singh, F van Breugel, RPN Rao, BW Brunton, "Understanding biological plume tracking behavior using deep reinforcement learning", ALIFE 2020: The 2020 Conference on Artificial Life, July 2020 [36].

• Singh SH, Peterson SM, Rao RP, Brunton BW. "Mining naturalistic human behaviors in long-term video and neural recordings", Journal of Neuroscience Methods, Jul 2021 [37].

• SH Singh, SM Peterson, RPN Rao, BW Brunton, "Enabling naturalistic neuroscience through behavior mining: Analysis of long-term human brain and video recordings", 2019 Conference on Cognitive Computational Neuroscience, Berlin, Germany [38].

• SH Singh, SM Peterson, Dichter B, Scheid M, RPN Rao, BW Brunton, "AJILE12: Long-term naturalistic human intracranial neural recordings and pose", Preprint, Aug 2021 [39].

• SH Singh, SM Peterson, RPN Rao, BW Brunton, "Enabling naturalistic systems neuroscience: Interpretable representations for analysis of long-term human neural activity and behavior," Learning Meaningful Representations of Life Workshop at the 33rd Conference on Neural Information Systems Processing (NeurIPS 2019, Vancouver, Canada)

• SH Singh, RPN Rao, BW Brunton, "Enabling naturalistic neuroscience: Mining large-scale human behavior recordings.", BIOKDD 2019, 18th International Workshop on Data Mining in Bioinformatics (at KDD 2019, Anchorage, AK)

• SH Singh, SM Peterson, NXR Wang, RPN Rao, BW Brunton, "Analysis of large-scale naturalistic human brain and behavior recordings", Ninth International Workshop on the Statistical Analysis of Neuronal Data (SAND9, Pittsburgh, PA)

Chapter 2

PROJECT SYNOPSES

2.1 Behavior mining in long-term human neural and behavior recordings

Summary: Human neuroscience experiments have traditionally been conducted using pre-designed and well-controlled experiments, in which timings of cues, stimuli, and behavioral responses are known precisely. Recent technological advances have enabled us to study longer and increasingly naturalistic brain recordings, giving rise to a new paradigm called "naturalistic neuroscience" where neural computations associated with spontaneous behaviors are studied. Analyzing such unstructured, long-term, and multi-modal data remains a substantial challenge, due in part to the lack of *a priori* experimental design and the difficulty of isolating interpretable behavioral events. Here we develop a highly automated processing pipeline for analyzing large (≈250 GB/subject) datasets of simultaneously recorded human electrocorticography (ECoG) and naturalistic behavior video data. Our pipeline discovers and annotates over 40,000 instances of naturalistic arm movements in long term naturalistic behavior data across 12 subjects (≈18 million video frames over a week/subject) using a combination of computer-vision algorithms, discrete latent-variable modeling, and string pattern-matching. We extract neural correlates associated with these naturalistic events from the simultaneously recorded ECoG data and show that they largely corroborate prior findings from traditional controlled experiments. Our pipeline produces large training datasets for brain-computer interfacing applications, and we show decoding results from a movement initiation detection task; we expect these decoders to be more robust to sources of variability found in real-world scenarios due to the use of naturalistic training data.

Additional Detail - Dataset: Our dataset consists of human intracranial neural recordings (ECoG) and simultaneously recorded behavior video, obtained opportunistically from 12 epilepsy patients for the entire duration of each subject's long-term (7–10 days) clinical observation. Each subject had about 90 electrodes implanted, covering different cortical regions in one brain hemisphere as determined by individual clinical need. Together, the ECoG and video (≈18 million RGB frames/patient for ≈7 days) amount to 250 GB of data per patient.

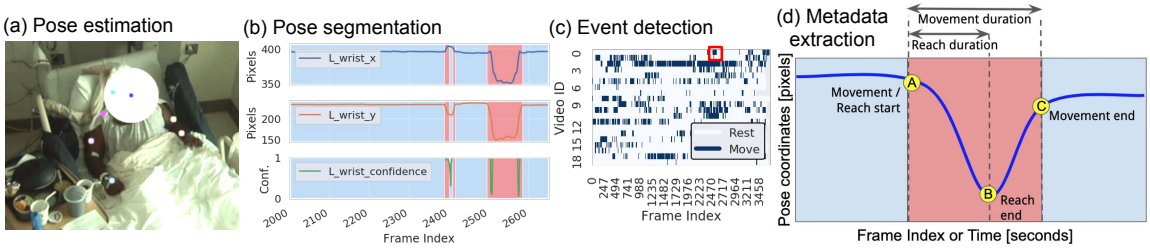

Figure 2.1: Pipeline for behavioral video data processing – Left to right: (a) Anonymized frame showing pose keypoints (colored dots) tracked on human subject. (b) Autoregressive hidden Markov model robustly segments pose trajectory into 'rest' (shaded sky blue) and 'move' (shaded pink) states. (c) Raster plot of pose states for several video-clips for pattern matching at scale. Red box shows example of a detected rest-to-movement event. (d) Post event-mining metadata extraction to get event attributes like movement duration, start and end coordinates, mean pose estimation confidence, among others.

Behavioral video processing pipeline: We take a scalable behavior-mining approach that robustly uncovers instances of naturalistic upper-limb movements in each patient's video data that are amenable to analysis. We first manually annotate several key-points (Fig. 2.1a) on the body of the patient on 1000 randomly selected frames of the video data, and use these to train a pose estimation algorithm [21] to extract pose trajectories for the entire duration of video; we then segment (Fig. 2.1b) these trajectories in time using unsupervised discrete-state latent-variable models [40].

Interestingly, having a discrete, sequential representation of upper-limb pose simplifies the problem of detecting behavioral events to pattern-matching on strings. To detect movement onset-continuation-offset sequences with high temporal precision, we apply regular-expression based queries on single-limb latent-state trajectories (Fig. 2.1c). Additionally, by combining the latent-state sequences from both left and right upper-limbs, we can detect various possible orderings of limb movement (e.g. raising the left and right wrists simultaneously vs. one after the other at different time offsets). Our approach also extracts behavioral metadata associated with the detected events, such as movement reach angle, magnitude, duration, pose estimation confidence over event, and coarse-grained start and

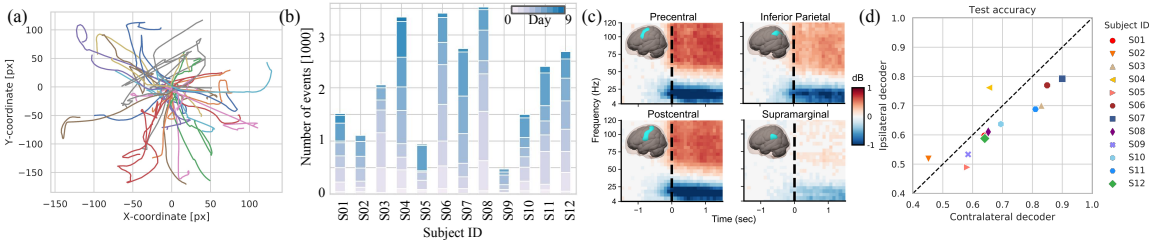

Figure 2.2: Left to right: (a) A sample of 50 typical right-wrist trajectories (translated to start at origin) showing diversity of naturalistic reach movements for a single patient. (b) Number of right-wrist movement initiation events discovered per day for each of 12 subjects, totaling 475 to 3526 events per subject across their entire duration of clinical observation (268 mean ± 123 s.d. per day). (c) Event-locked spectrograms, averaged by brain region (cyan color in insets) across 12 subjects, showed movement-associated high-frequency power increase and low-frequency power decrease. (d) Test set decoding accuracy for initiation of movement of contralateral (side opposite electrode implant) and ipsilateral (same side) wrists. As expected, decoding of contralateral movements is slightly more accurate than ipsilateral in almost all cases.

end locations, among others (Fig. 2.1d).

After filtering using the associated metadata, we construct a large database of naturalistic movement events that captures naturalistic upper-limb movement variability (Fig. 2.2a) and is robust to variation in lighting, camera angle, and level of activity in the video. In total, our dataset includes 12 subjects, with 475 to 3526 individual movement events per subject (268 mean ± 123 s.d. per day, Fig. 2.2b).

Towards naturalistic neuroscience: To understand neural correlates of these behavioral events, we analyze the simultaneously recorded ECoG brain activity. These naturalistic neural correlates largely corroborate results from traditional, controlled experiments [41]; the naturalistic and experimental results also concur on brain regions activated across patients (Figs. 2.2c). Further, to demonstrate the suitability of such behavior-mined opportunistic data for brain-computer interface (BCI) decoding applications, we train classifiers that perform significantly over chance (comparable with [42]) in discriminating upper limb movement-onset events from rest (Figs. 2.2d). Such naturalistic training data could poten-

tially improve the accuracy, robustness, and generalizability of BCI decoders when deployed in real-world settings.

Conclusion: In summary, we present a highly automated, novel workflow for analyzing simultaneously recorded naturalistic long-term human brain and behavioral video data. A key contribution is an approach to generate a robust, temporally precise, and queryable representation of human upper limb poses. We believe that our approach and results are of broad interest to neuroscientists and neuroengineers. We have published our curated dataset and believe that it will be a valuable resource for future studies of naturalistic movements: https://github.com/BruntonUWBio/mining2021.

2.2 Emergent Behavior and Neural Dynamics in Artificial Agents Tracking Turbulent Plumes

Summary: Tracking a turbulent plume to locate its source is a complex control problem requiring multimodal sensory integration and robustness to odor intermittency, wind non-stationarity and spatiotemporal variability in plume statistics. This task is routinely performed by flying insects, often over long distances, in pursuit of food or mates. Isolated aspects of this remarkable behavior have been studied in detail in many experimental studies. Here, we take a complementary *in silico* approach, using artificial agents trained with reinforcement learning, to develop an integrated understanding of the behaviors and neural computations that support plume tracking. Specifically, we use deep reinforcement learning to train recurrent neural-network (RNN) based agents to locate the source of simulated turbulent plumes. Interestingly, the agents' emergent behaviors resemble those of flying insects, and the RNNs learn to represent task-relevant variables such as head-direction and time between odor encounters. Through two separate analyses, we find that reflexive short-memory behaviors are sufficient tracking constant wind-direction plumes, but longer memory timescales are essential for tracking plumes that switch direction. Furthermore, we find that the RNNs' neural activity is low-dimensional and organized into distinct macroscopic structures for a subset of the behavioral modules. Our *in silico* approach provides key intuitions for an integrated understanding of turbulent plume tracking and motivates future targeted experimental and theoretical developments.

Methods: We implement a particle-based 2D plume simulation model that replicates the statistics of real-world turbulent plumes [43]. Our agents are actor-critic networks [44], each comprising a Vanilla Recurrent Neural Network (RNN) followed by a pair of 2 consecutive feedforward layers corresponding to the actor and critic heads of the network. Layers are 64 units wide with *tanh* nonlinearities. At each time step, an agent receives a real-valued 3-tuple input comprising egocentric wind velocities (x, y) and odor concentration at its current location. It produces continuous valued turn and forward-movement actions matched to the capabilities of real flying insects [45, 46]. We train 14 agents with independently randomly initialized networks using Policy Gradient (PPO) [47] and analyse one of the top-5 best

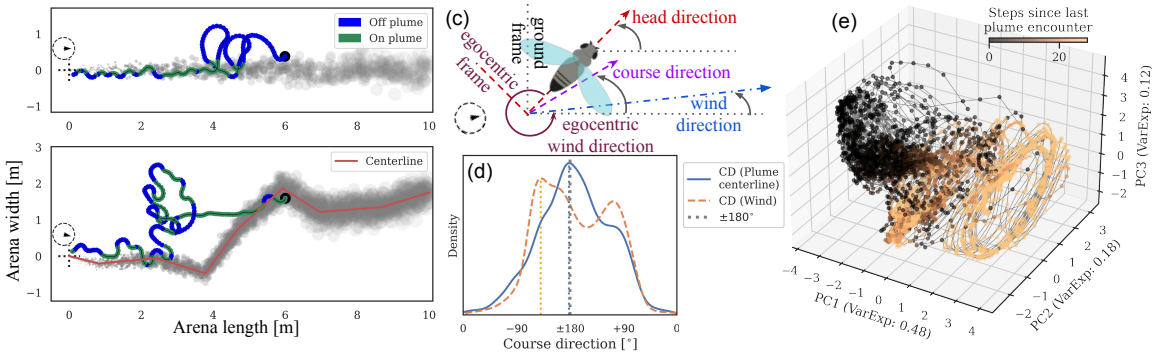

Figure 2.3: **(a-b)** An agent's plume tracking trajectories, each starting at a filled black circle and ending at dotted crosshairs (far left) indicating the plume source. Plume in dull grey color. Wind-direction shown by arrow in dotted circle. Successful tracking episodes in **(a)** a constant wind-direction plume, and **(b)** a plume that switches directions multiple times; **(c)** agent course-direction (CD), measured counter-clockwise from ground x-axis, is the direction the agent actually moves taking into account *slip* due to wind-advection; **(d)** Empirical distribution of CD across multiple successful trajectories in plumes with switching wind-direction suggests that agents track with respect to plume centerline rather than with respect to the wind. **(e)** Neural activity from a variety of episodes covering successful/unsuccessful homing outcomes and constant/non-stationary wind-direction plume configurations, colored by time elapsed since agent last encountered plume (25 steps = 1.0s).

performing agents (as measured by fraction of successful trajectories) in this synopsis.

Results: Trained agents can track plumes across a variety of plume simulator configurations (Figure 2.3a-b) using a set of 3 behavior modules (Fig. 2.4a-b). Emergent behavior modules are reminiscent of *upwind surging*, *crosswind casting* and *U-turn* behaviors previously reported in the literature for flying insects [45, 48–50]. Our analyses suggest an intriguing experimentally testable hypothesis – that agents track plumes in changing wind-directions with respect to their local shape rather than the current wind-direction (Figure 2.3c). Simultaneously, we find that the RNN learns to represent task-relevant variables such as the time elapsed since the last odor encounter (Figure 2.3e). Neural activity is low-dimensional (Fig. 2.4f), and exhibits a signature structure for 2 of the 3 behavior modules (Fig. 2.4c-d), organized in overlapping but distinct regimes (Fig. 2.4e) at a macroscopic level. Using standard notation, the RNN update rule, $\mathbf{h}_t = F(\mathbf{h}_{t-1}, \mathbf{x}_t) = tanh(\mathbf{W}_h\mathbf{h}_{t-1} + \mathbf{W}_x\mathbf{x}_t + b)$, can

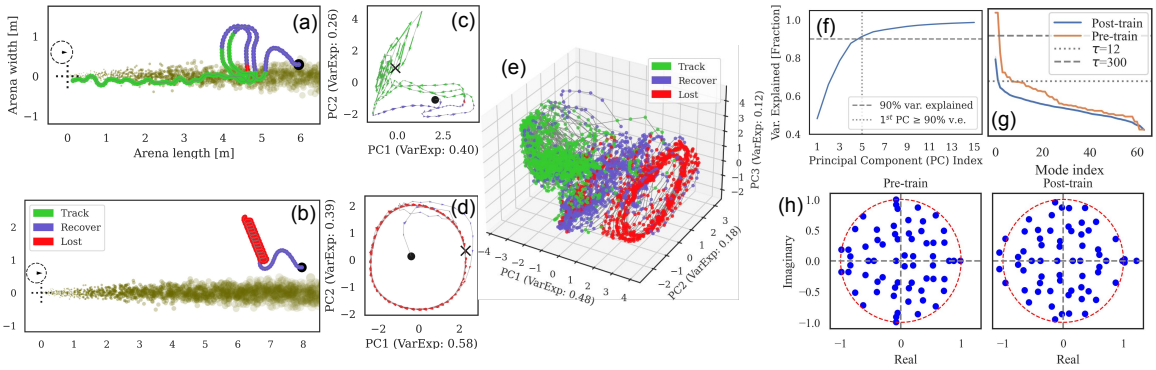

Figure 2.4: **(a-b)** Agent's plume tracking trajectories, each starting at a filled black circle. Plumes (dull grey) originate at dotted crosshairs (far left). Wind-direction (constant) shown by arrow in dotted circle. Agent trajectory for **(a)** a successful tracking episode, and **(b)** an unsuccessful episode that strays from the plume. Trajectories are colored by emergent behavior modules, namely: *track*, where agent inside or skims the boundary of the plume; *lost* where, after an extended duration of plume loss, it spirals (mostly) in place; and *recover*, where, on brief periods of plume loss, it makes large (usually) cross-wind movements to find the plume. **(c-d)** Neural activity corresponding to each row's trajectory, project onto their top-2 Principal Components. Quivers indicate neural activity gradient. **(c)** A 'funnel' like structure (in green) associated with *track*, and a limit-cycle (in red) associated with the periodic *lost* behavior modules. **(e)** Neural activity from multiple trajectories shows overall structure in the neural dynamics, and an amorphous intermediate region associated with the *recover* behavior. **(f)** Like biological recordings, artificial neural activity is low dimensional, with 90% variance explained by top-5 PCs. **(g)** Time-averaged stimulus integration timescales τ associated with stable eigenmodes of recurrence Jacobian $\mathbf{J}^{\text{rec}}$ show a bulk of relatively short (within 12 timesteps, 25 steps = 1.0s) timescales. (Top 5 τs are 56.5, 13.0, 7.7, 6.8 & 5.8 timesteps) Before training, τs associated with $\mathbf{W_h}$'s eigenmodes can be large, even exceeding the length of the training/evaluation episodes (300 steps). **(g)** Eigenvalue spectra of the RNN recurrence matrix $\mathbf{W_h}$ before and after training show how training results in the generation of unstable modes.

be linearized around an arbitrary expansion point $(\mathbf{h}^{\text{e}}\mathbf{x}^{\text{e}})$ on the agent's trajectory, to get the approximate linear system [51, 52], $\mathbf{h}_t \approx F\left(\mathbf{h}^{\text{e}}, \mathbf{x}^{\text{e}}\right) + \mathbf{J}^{\text{rec}}\big|_{(\mathbf{h}^{\text{e}},\mathbf{x}^{\text{e}})} \Delta\mathbf{h}_{t-1} + \mathbf{J}^{\text{inp}}\big|_{(\mathbf{h}^{\text{e}},\mathbf{x}^{\text{e}})} \Delta\mathbf{x}_t$. Like in [52], we use $\tau_i = |(1/\ln|\lambda_i|)|$ to obtain the stimulus integration timescales τ_i associated with stable eigenvalues λ_i of the RNN recurrence Jacobian $\mathbf{J}^{\text{rec}}\big|_{(\mathbf{h}^{\text{e}},\mathbf{x}^{\text{e}})}$ over multiple tracking trajectories. The bulk of these timescales are within $\approx$ 0.5s, suggesting that the

plume tracking task predominantly needs short-timescale memories (Fig. 2.4g). DRL training appears to produce unstable eigenmodes in the RNN connectivity matrix $\mathbf{W}_h$ (Fig. 2.4h).

Conclusion: Our normative RNN-based agent model enables an abstract yet interpretable understanding of the behaviors and neural computations that support turbulent plume tracking. Our findings motivate future targeted neuroethological experiments and theoretical developments. Animations accompanying this study can be found at https://github.com/BruntonUWBio/plumetracknets.

Chapter 3

BEHAVIOR MINING IN LONG-TERM HUMAN BRAIN AND BEHAVIOR RECORDINGS

3.1 Introduction

Neuroscience has long been interested in understanding brain activity associated with spontaneous behaviors in freely behaving subjects. Even so, hypotheses regarding brain function have typically been tested using carefully designed, well-controlled experimental tasks, where timing of cues, stimuli, and behavioral responses are known precisely. Fortunately, recent technological advances have enabled us to study increasingly naturalistic and longer brain recordings, giving rise to a new paradigm called "naturalistic neuroscience" [6, 7, 53–55] where neural computations associated with such spontaneous behaviors are studied. Understanding such unstructured, long-term, and multi-modal data poses a substantial analytic challenge, due in part to the lack of *a priori* experimental design and the difficulty of isolating interpretable behavioral events.

3.1.1 Related work

Our work is most closely related to recent work in human naturalistic neuroscience that combine computer vision with opportunistic clinical brain recordings, including [55], [56], and [57]. In particular, we build on the work of [42] using similar video data and estimate the pose of human upper-body keypoints using neural-networks. [54] use optical flow and image partitioning to detect coarse limb movements from video taken in a clinical setting similar to ours and develop neural decoders for detecting these movements from brain data. Compared to [42], who use a moving window heuristic on pose estimates to detect movements, we take a more principled approach to modeling the pose data. This allows us to localize movement events with finer temporal-resolution and characterize entire movement trajectories, which in turn enables novel applications described later in the paper. We also use newer, more efficient computer vision methods [28, 58] that allow us to process data at a scale that exceeds all of the aforementioned studies taken together in the number of subjects and duration of recordings analyzed. Finally, we focus on curating, characterizing, and making our dataset available to the research community to foster further research and development in this area.

3.1.2 Our approach

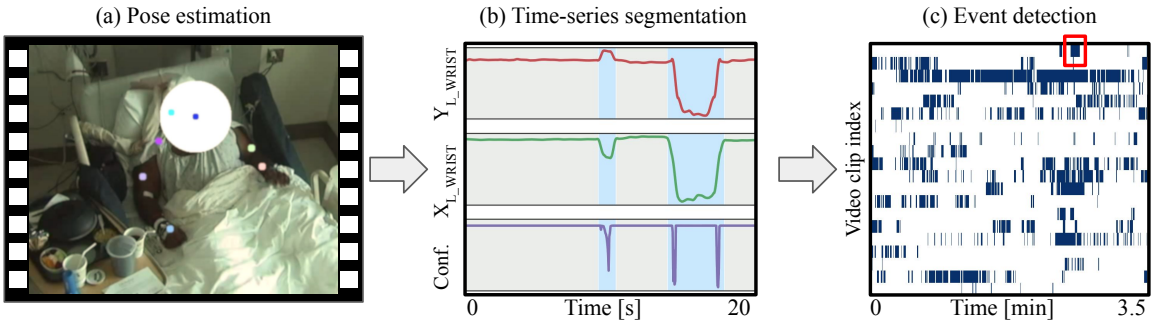

Figure 3.1: Pipeline for behavioral video data processing. (a) Video frame showing estimated pose keypoints (colored dots) on human subject. (See Figure 3.8 for more examples.) (b) Trajectories of estimated (x, y) pose coordinates and associated 'confidence score' for the left wrist. Autoregressive hidden Markov model (Section 3.3.1) robustly segmented pose trajectory into *rest* (shaded grey) and *move* (shaded light blue) states. (c) Raster plot of pose states (*move* in dark blue, *rest* in white) for several video-clips for pattern matching at scale. Red box depicts one *movement initiation* event matching a pattern of 15 contiguous *rest* states (0.5s) followed by 15 contiguous *move* states (0.5s).

We present a scalable behavior-mining approach to analyze simultaneously recorded naturalistic brain and behavior data, obtained opportunistically from human subjects undergoing long-term clinical monitoring prior to epilepsy surgery. Our video processing pipeline (Figure 3.1) first estimates the locations of keypoints (e.g. wrists and elbows) on the upper-body using a neural network trained on each subject [21]. We then segment the trajectory of each keypoint in time using discrete latent-variable models, building a discrete representation of pose dynamics. Interestingly, having a discrete, sequential representation of upper-limb pose simplifies the problem of detecting behavioral *events* to pattern-matching on strings. Using regular-expressions corresponding to patterns of interest, we discover thousands of interpretable events per subject—an order of magnitude more observations than in a typical controlled human experiment. To study the rich naturalistic variability associated with these events, we also extract metadata including movement angle, magnitude, and duration.

Next, we explore the use of these behavioral events for neuroscience and neuroengineer-

ing applications by analyzing the simultaneously recorded brain data. Event-averaged spectrograms associated with our naturalistic human upper-limb movement initiation events corroborate and strengthen previous findings from controlled experiments [41] (see also [59]). Preliminary investigations also suggest that our workflow could produce data useful for training brain-computer interface (BCI) decoders; due to the use of larger sample sizes of training data representative of naturalistic variability, such decoders may perform more robustly in real-world deployments.

Our key contributions in this paper are as follows. First, we present a highly automated, novel workflow for analyzing simultaneously recorded naturalistic long-term human brain and behavioral video data. Second, we develop a domain-relevant, robust, temporally precise, and queryable representation of human upper-limb pose. Third, to showcase our workflow, we demonstrate example applications in neuroscience and neuroengineering, suggesting that our approach and results are of broad interest. Finally, to support open science and facilitate further research in this area, we release our curated dataset consisting of annotated naturalistic events and associated neural recordings.

3.2 Dataset

3.2.1 Human subjects and data collection

Our dataset consists of human intracranial electrocorticography (ECoG) [60] neural recordings and simultaneously recorded behavioral video recordings, obtained opportunistically from 12 patients with epilepsy for the duration of each patient's long-term (7–9 days) continuous clinical observation. The patient population comprised 8 males and 4 females, with an average age of 29.4 years ($\pm$7.9 years s.d.) at the time of recording. The University of Washington Institutional Review Board for the protection of human subjects approved our study and all patients provided their informed written consent. Patient behavior was continuously recorded by a wall-mounted camera (Sony SNC-EP580 series, with day [RGB color] and night [near-infrared B/W] modes) for real-time monitoring by an around-the-clock clinical team, except during intermittent equipment servicing or private times when the camera was switched off or turned away. Patients were observed performing their daily

activities (including talking, eating, watching TV, using a computer or phone, sleeping, receiving clinical care etc.) from the hospital bed while being tethered to a brain-recording interface. Patients continued to receive their clinician-determined doses of anti-epileptic and pain-relieving drugs throughout the recording period. Although there may be some cognitive impairment from medication, using electrocorticographic data from epilepsy patients for sensorimotor neuroscience and brain-computer interface tasks has become accepted practice in the field [61–64].

Each patient had about 90 electrodes implanted under the skull and dura, directly on their brain surface, including either an 8×8 grid or two 8×4 grids of electrodes. (See Table 3.1 for the exact number of electrodes per subject.) These electrodes were placed to monitor a subset of cortical regions (right hemisphere for 5 and left hemisphere for 7 patients), predominantly in one brain hemisphere, as determined by individual clinical need for epilepsy resection surgery and independent of the analysis presented in this manuscript. Grid electrodes are constructed of 3-mm diameter platinum pads spaced 1-cm apart center-to-center and embedded in silastic (Ad-Tech Medical Instrument Corporation Epilepsy Long Term Monitoring subdural grid electrodes). The sampling rates for video and ECoG recordings (Natus Medical Incorporated XLTEK EMU 128 FS Headbox) were 30 frames/second (at 640 pixels $\times$ 480 pixels resolution) and 1000 Hz respectively. Together, the ECoG and video (approximately 18 million frames for a week) totaled about 250 GB of data per patient.

3.2.2 ECoG data preprocessing

We applied the following standard preprocessing steps to the ECoG data [41, 54, 62, 65]: First, we removed DC drift by subtracting the median voltage (across entire recording) of each electrode. Next, in a manner similar to [66], we set to 0 (volts) all electrode data for 2 seconds around high-amplitude artifacts, identified by times of abnormally high electrode-averaged absolute voltage (>50 interquartile range [IQR], across entire recording.) This was implemented using custom Python code. Then, we band-pass filtered ($1 - 200$ Hz, FIR filter), notch filtered (60 Hz and harmonics up to 240 Hz, FIR notch filter) the data to minimize line noise, and downsampled the data to 500 Hz, all using the Python `MNE`

package (version 0.22) [67] functions `filter.filter_data()` and `filter.notch_filter()` respectively. Finally, any drifts common to all electrodes were removed by re-referencing the data at each time-step to the common median across electrodes. Electrodes that had abnormally high standard deviation ($>$ 5 IQR) or kurtosis ($>$ 10 IQR) compared to the across-electrodes median, were excluded from further analysis. All electrode positions were localized and converted to Montreal Neurological Institute and Hospital (MNI) coordinates using the Fieldtrip toolbox (version 20181206) [68, 69] in MATLAB. Additional details about ECoG preprocessing and electrode localization are available in [59]. To aid interpretability, we restricted our analysis in this paper to the 64 grid electrodes covering one hemisphere per subject. However, data for all available electrodes are provided in the publicly released dataset accompanying this paper. In most cases, we also did not analyze the neurally and behaviorally atypical data from the first two days of a patient's hospital stay, since patients were usually heavily medicated during this time while recovering from electrode implantation surgery.

3.2.3 Video data preprocessing

Before processing the video data, we manually inspected and annotated it at a coarse (every 3 minutes or so) level of granularity to create an *omit-list* of video segments that were excluded from further processing. The omit-list included long time-spans of sleep, times when a clinical or research team was actively working with the subject, private times, and times when applying computer-vision algorithms was impossible due to poor lighting conditions or severe occlusion of the subject's body. Almost all camera movement occurs around times when the clinical team is actively working with the patient. Removing these times results in a mostly steady recording configuration as seen in Figure 3.1(a). We also labeled and excluded times when the clinical team had placed seizure restraints on the subjects' hands, since these limited mobility and gave rise to unnatural movements. Completing these manual annotations took about 6–12 hours per subject, depending on their clinical treatment regime, activity and sleep schedule, and length of hospital stay. When analysing ECoG and video data together, the two data-streams were synchronized using metadata extracted

using the equipment manufacturer's software (Natus Medical Incorporated, NeuroWorks Version 9.1.0.2658).

3.3 Methods

3.3.1 Precision extraction of behavioral events at scale

We developed and validated a pipeline to extract temporally precise, interpretable movement events, by processing the video data through pose-estimation, pose time-series segmentation, event detection and finally, event metadata extraction (Figure 3.1).

Markerless pose estimation

To extract a subject's pose from raw video, we trained a state-of-the-art markerless pose estimation tool [21] known for its speed and data efficiency [58]. For training data, we manually annotated around 1000 frames per subject chosen randomly from the entire duration of a subject's video data, preferentially sampling active, daytime hours over times when the subject was asleep. For each frame, we annotated up to 9 keypoints on the subject's body whenever visible [70]. These keypoints were the nose, both wrists, elbows, shoulders, and ears (Figure 3.1a). Note that unlike some professional motion-tracking systems, there are no physical markers placed on the bodies of the subjects during any time.

During prediction, the pose-estimation tool produced the (x, y) coordinates and a *confidence* estimate between $[0, 1]$ for each keypoint per frame (Figure 3.1b). To quantify the performance of keypoint tracking, we estimated the pixel-wise RMS error to be 1.54 mean $\pm$ 0.13 s.d. pixels on the training data and 5.97 mean $\pm$ 1.96 s.d. pixels on the holdout data, both averaged across 12 subjects; holdout data is 5% of the manual annotations, excluding points below confidence threshold of 0.1. As an approximate scaling, 12 pixels in the video span about 4 cm in physical units, which is about the width of a human wrist. We estimated this scale by comparing standard human measurements [71] with median distance between shoulder keypoints (in pixels) at movement onset for a few subjects. On average, estimating pose for the entire duration of a subject's video took 400 GPU-hours per subject using `AWS p2.16xlarge NVIDIA K80` GPUs. We denoised the estimated pose trajectories

by median filtering (window length 11 frames) and smoothing (window length 11, 2nd order Savitzky-Golay [72] filter).

Segmentation of pose trajectories

Next, we segmented the pose time-series into discrete, interpretable states while preserving the temporal precision of the keypoint tracking. We applied a first-order autoregressive hidden semi-Markov model (ARHSMM) [73] with two latent states to each keypoint's time-series (Figure 3.1b for left-wrist). This model converts each keypoint's continuous pose dynamics into discrete latent-state trajectories consisting of two distinct states, that we label as *rest* and *move* **states**. Using a semi-Markov, rather than a Markov model, accounts for the bias that limbs tend to be at *rest* most of the time and mitigates unnecessary switching between latent states. As in [5], we fit the ARHSMM using the `pyhsmm-autoregressive` Python package (version/commit `e6cfde5`) [40]. The resulting states are at video frame-rate resolution and the segmentation is relatively robust to variation in lighting, camera angle, and level of activity in the video.

Behavioral event mining

Discretizing the pose trajectories facilitates the description of scientifically interesting behaviors performed spontaneously by the subject, even though they vary greatly in duration. Specifically, the task of finding different types of behavioral **events** thus reduces to string pattern matching on the discretized dynamics. For the behaviors we explore in the rest of this paper, we looked for *movement initiation* events by matching a pattern of 15 consecutive *rest* states (0.5s), followed by at least 15 consecutive *move* states (0.5s). Similarly, *no-movement* events are state sequences of 90 *rest* states (3.0s) across both wrists and the nose. To create our database of wrist movements, we use regular expressions to quickly find thousands of non-overlapping instances of such patterns in the discretized pose dynamics for each subject.

Parameters for smoothing, hyperparameters for the ARHSMM segmentation model, and the choice of regular expressions for event detection, were picked empirically by assessing

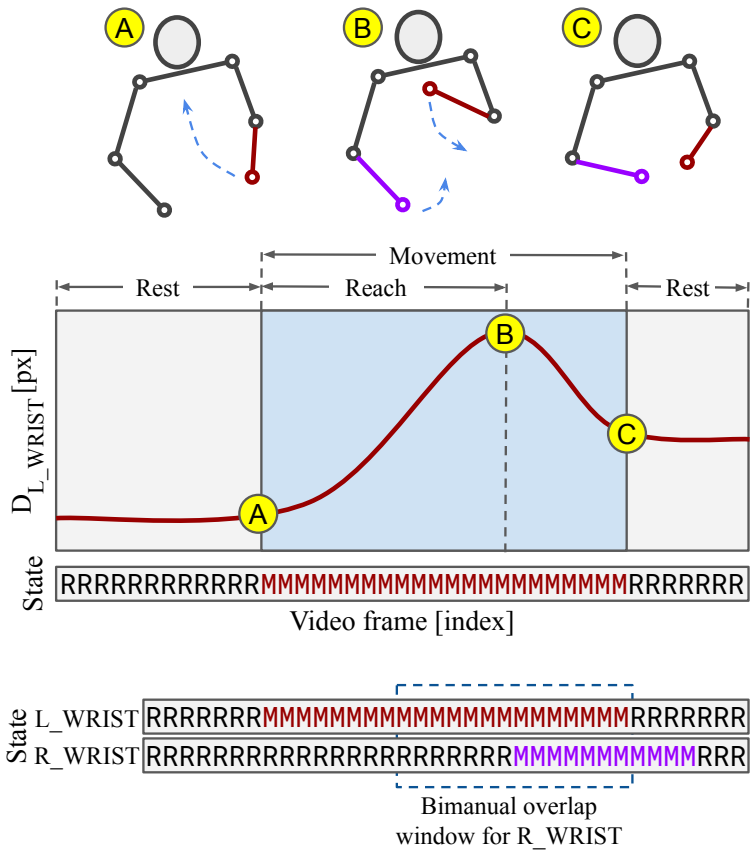

Figure 3.2: Schematic of metadata extraction: [Top] Cartoon showing left-wrist movement at time of (A) movement initiation (B) maximum displacement from initiation (*reach*) and (C) movement end. [Middle] Time course of left-wrist radial distance [px: pixels], and discretized state (R: *rest*, M: *move*). Extracted movement metadata include duration, start and end coordinates, among others (Sec. 3.3.1). [Bottom] Discretized state sequence for both left and right-wrists, now showing movement initiation in the right-wrist while the left-wrist is still in motion. The dashed line shows the window over which *bimanual overlap* metadata is calculated, corresponding to the number of frames for which the opposing (left) wrist is in motion over that duration.

performance on pose time-series derived from a small representative set of subject videos. We confirmed that the temporal accuracy of event boundaries matched our expectations by manually inspecting a few dozen random events of each movement type for each subject. (See Section 3.6 for additional details.)

Event metadata extraction

For each detected movement event, we extracted several metadata features from the continuous pose-dynamics associated with the movement. These include movement-associated metadata (Figure 3.2) like the (x, y) *coordinates* of the keypoint at the start and end of the event, *duration* of the entire movement (up to next rest state), and *rest duration* before and after movement.

Observed naturalistic hand movements often consisted of a hand reaching out, touching, or grabbing an object, then bringing the hand back to the body. Therefore, we defined the *reach* of a wrist movement to be its maximum radial displacement during the course of the event, as calculated from its location at the start of the event. We extracted the *magnitude*, *angle*, and *duration* for each reach.

To measure the *shape* of a movement, we fit 1^{st}, 2^{nd} and 3^{rd}-degree polynomials to a keypoint's displacement trajectory. Differences between the quality of the fit (as measured by R^2) to each polynomial type provide a rough measure of the "curviness" of the movement trajectory. We also estimated a movement's onset and offset *speeds*, by calculating the keypoint's displacement change within short time windows around the start and end of the movement.

Since people often move both hands at the same time (i.e. "bimanually"), we augmented each movement event with metadata about the opposing wrist's movement, if any (Figure 3.2). By juxtaposing the discrete state sequence of both wrists, we calculated when the opposing hand starts to move (*lead/lag time difference*) and how long this movement overlaps with that of the primary hand (*overlap duration*).

Finally, we generated metadata that we used to remove false positives that arise from pose estimation failures and unusual pose states. To compensate for failures in 2D pose

estimation, we calculated movement-weighted *confidence scores* for each event and removed those below a manually determined threshold. To eliminate outlier body postures, we calculated mean *distance* and mean *angle* between *shoulder* keypoints, then removed events from the top and bottom 5 percentiles of these quantities.

3.4 Results

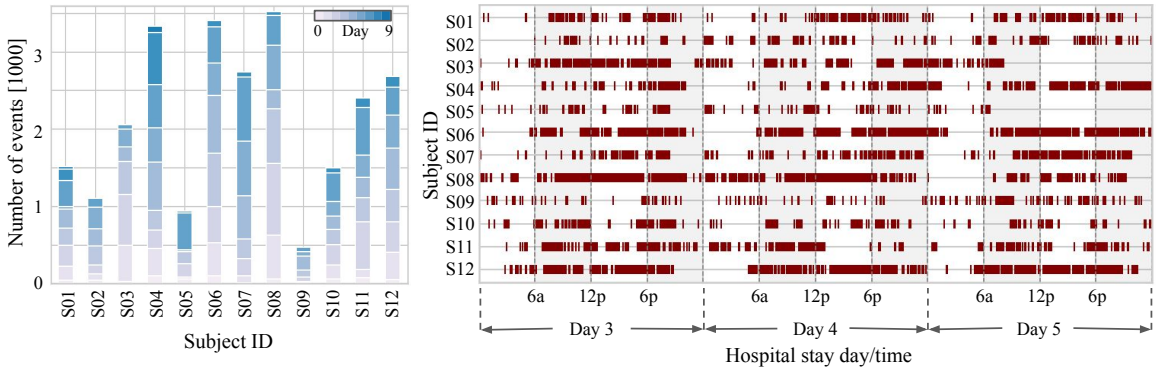

Figure 3.3: [Left] Number of right-wrist movement initiation events discovered per day for each of 12 subjects, totaling 475 to 3526 events per subject across their entire duration of clinical observation (268 mean ± 123 s.d. per day). [Right] Raster plot of right-wrist movement initiation event occurrences showing bursts of activity interspersed with periods of rest or omit-listed (Section 3.3.1) periods. See Figure 3.4 for equivalent plots for left-wrist.

Our pipeline extracted 959 to 6745 individual wrist movement events per subject (487 mean ± 215 per day) across 12 subjects. (Figures 3.3 and 3.4 for right and left wrists, respectively). We observed rich within-subject variability in the event metadata: Figures 3.6 and 3.7 visualize the distributions for several movement associated metadata for right and left wrists, respectively. A random sample of trajectories for one patient (S10) is shown in Figure 3.5.

3.5 Code and dataset release

Code and data to reproduce several key plots in this manuscript is publicly available at: https://github.com/BruntonUWBio/mining2021. Our complete curated dataset consist-

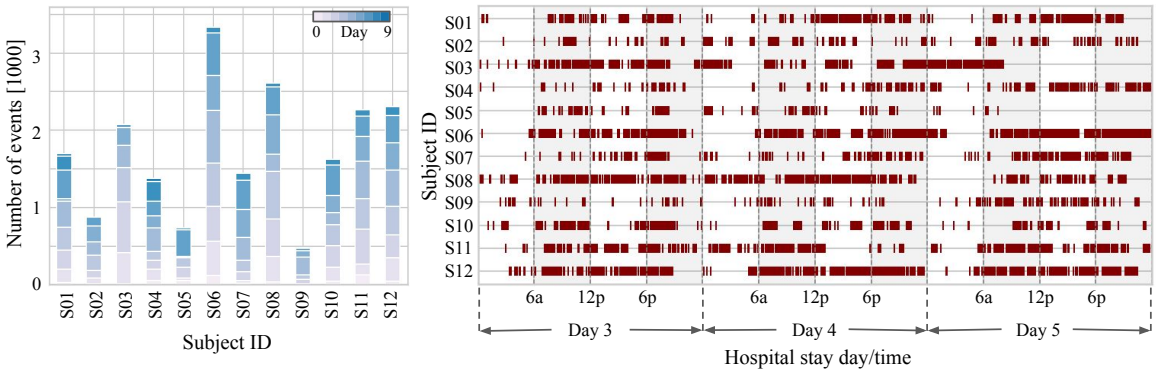

Figure 3.4: Number of left-wrist movement initiation events discovered per day for each of 12 subjects, totaling 484 to 3338 events per subject across their entire duration of clinical observation (219 mean ± 104 s.d. per day). [Right] Raster plot of left-wrist movement initiation occurrences. See Figure 3.3 for equivalent plots for the right-wrist.

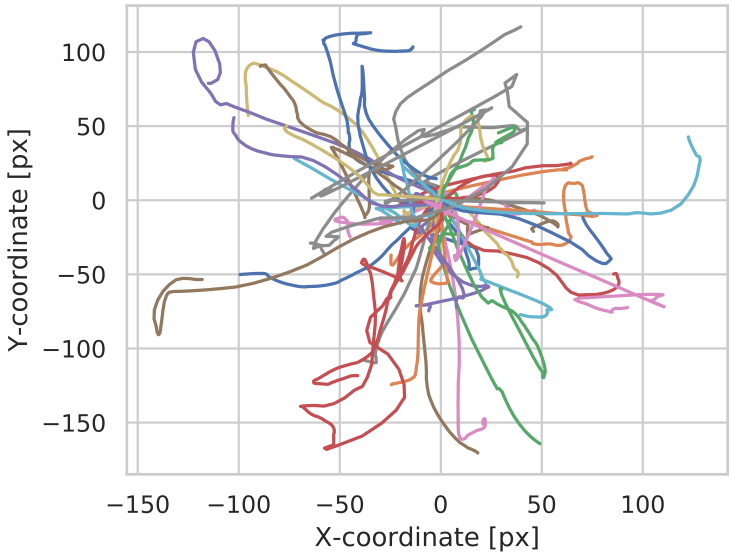

Figure 3.5: A sample of 50 typical right-wrist trajectories (px: pixels; translated to start at origin) showing diversity of naturalistic reach movements for a single subject (S10). Different colors represent different individual trajectories. Note the large variability in the movements, compared to what is normally captured by controlled experiments. (See Figure 3.8 for instructions to access video-snippets of such movements.)

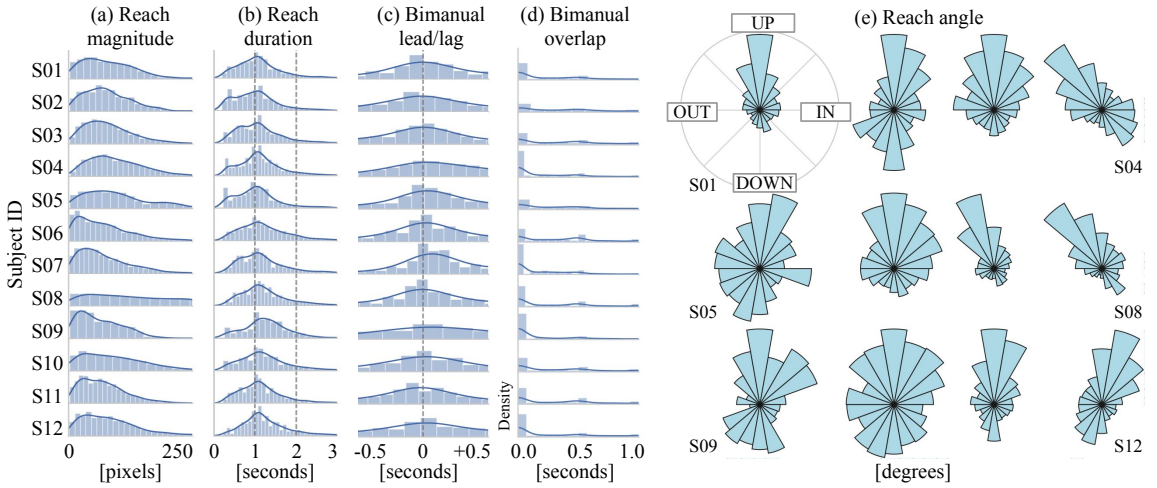

Figure 3.6: Histograms of right-wrist movement initiation event metadata per subject for their entire duration of clinical observation: (a) Reach magnitudes show a dominance of small movements (92 pixels/30.6 cm mean ± 22 pixels/7.3 cm s.d. of subject-wise medians), (b) Reach durations tended to be quite short (1.2s mean ± 0.1s s.d. of subject-wise medians), (c) When both hands moved together ("bimanually"), they tended to start at about the same time, with the left hand start very slightly earlier (leading by 0.2s mean ± 0.1s s.d. of subject-wise medians; negative values in plot imply opposing wrist leading primary), (d) Duration of time left hand was moving during a ±0.5s window around time of right hand movement initiation (0.3s mean ± 0.06s s.d. of subject-wise medians). (e) Polar histograms show that many subjects primarily made upward-downward reaches. However, reaches in almost every other direction were also observed. (95° mean ± 19° s.d. of subject-wise histogram mode bin, with angles measured counterclockwise) (For subject-level metadata statistics, see Table 3.2) See Figure 3.7 for equivalent plots for left-wrist.

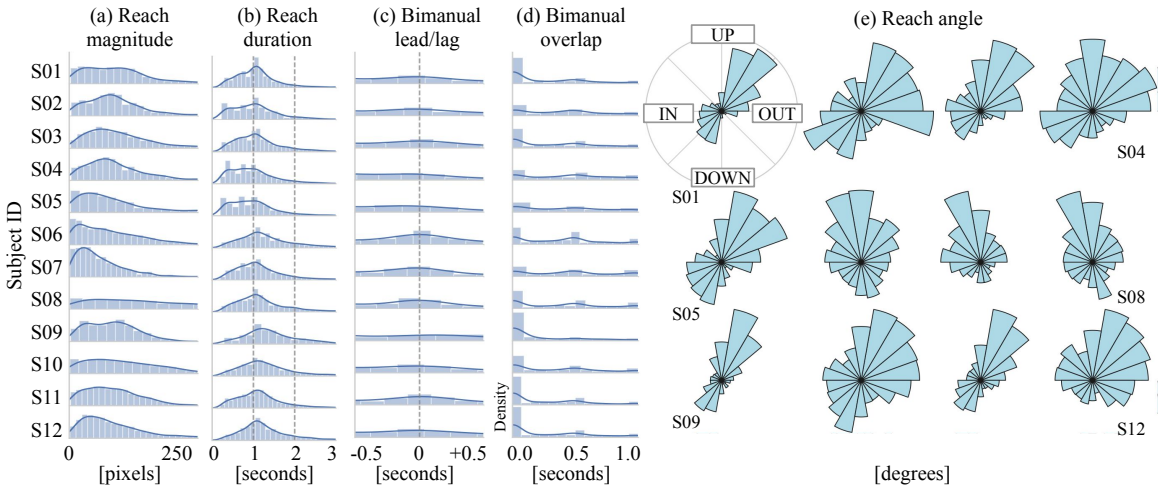

Figure 3.7: Histograms of left-wrist movement initiation event metadata per subject for their entire duration of clinical observation. We see similar trends for the left-wrist as were noted for the right-wrist in Figure 3.6. Distributional statistics for shown metadata are: Reach magnitude (94 pixels mean ± 26 pixels s.d.), Reach duration (37 frames mean ± 5 frames s.d.), Bimanual lead-lag (opposing hand leading by 9 frames mean ± 3 frames s.d.), Bimanual overlap (10 frames mean ± 2 frames s.d., overlap out of a possible maximum of 15 frames), Reach angle (101° mean ± 80° s.d. of subject-wise histogram mode bin, with angles measured counterclockwise) For subject-level metadata statistics, see Table 3.3.

ing of events and their metadata and associated neural data ($\approx$ 40 GB) can also be downloaded following the instructions provided at the aforementioned URL. Video examples of (left, right, & bimanual) wrist movements for one subject (S01) can also be found there.

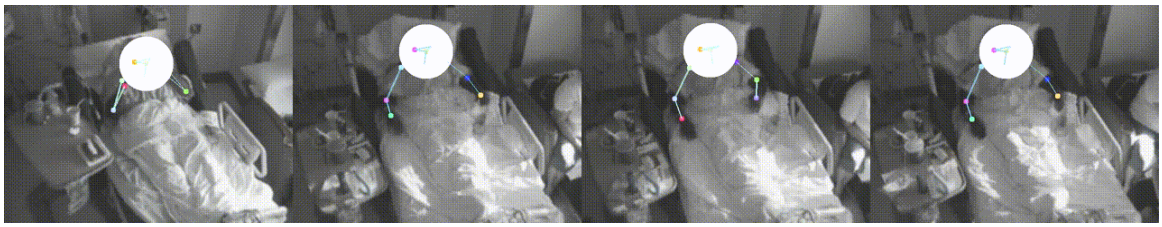

Figure 3.8: **Video frame from supplementary animations**: Additional video examples of extracted movements have been included in the code/data repository accompanying this manuscript: https://github.com/BruntonUWBio/mining2021. We extract 4 examples each of left-wrist, right-wrist and both-wrist ('bimanual') movement initiation.

3.6 Additional details on pose smoothing, pose segmentation and event mining

Here we provide additional details on how we selected parameters for pose smoothing, hyperparameters for the autoregressive hidden semi-Markov model (ARHSMM) pose segmentation model, and how we chose regular expressions for event detection.

Representative video set selection: We first picked 4 subjects that included both male and female subjects, subjects of two ethnic/racial groups, and subjects with electrode implants in either hemisphere. A sample of 15-20 video clips (each of $\approx$ 3 minutes each) per patient was hand selected to select times with different levels of patient activity and background activity, camera orientation, and lighting conditions.

Pose trajectory smoothing: We then processed the estimated pose for 9 keypoints (nose, both wrists, elbows, shoulders, and ears; see Figure 3.1a) for each subject's videos. The pose estimation algorithm, DeepLabCut [21], operates on a a frame-by-frame basis, which allows it to be fast (and therefore cost-effective) but comes at the cost of slightly noisy estimates and occasional jumps (short time-scale large amplitude noise) between frames. We used median filtering and (2nd order) Savitzky-Golay smoothing [72], in that order, to deal with jumps and noisy estimates respectively. A window length of 11 frames for both

filtering and smoothing was observed (manual visual inspection) to reject all noticeable jumps and smooth any remaining pose estimation noise, while not introducing any noticeable lags that could occur when aggregating data over large window-lengths. We picked a 2nd order smoother because it was the lowest-order (simplest) smoother that provided visually satisfactory noise removal. These parameters were found to work across pose trajectories for all 9 estimated keypoints.

Pose trajectory segmentation exploration: We first tried localizing event boundaries (movement initiation, change in movement direction, end of movement etc.) by manual inspection but found that this was a tedious and inexact process. So instead of first creating a ground-truth set and trying to fit a model to that, we chose to explore unsupervised trajectory segmentation techniques. We chose the ARHSMM due to prior demonstrations of its success in fine-grained behavior segmentation in other neuroscience experiments [5, 6, 40, 74] and the availability of a popular open source implementation `pyhsmm-autoregressive` (version/commit `e6cfde5`) in Python [40].

The ARHSMM requires the following hyperparameters to be specified: Number of discrete latent states (which could be roughly interpreted as the maximum number of behavioral states one expects in the data being segmented), observation data conditional probability distribution (which is a standard Gaussian distribution for continuous time-series data), hidden state duration Poisson distribution prior (which specifies how long the model will tend to stay in a state before switching), and autoregressive lag steps (which controls the complexity of dynamics learned by the model). There were additional parameters in the package that controlled other aspects of the model-fitting process, but these were left unchanged to package defaults [40].

As mentioned in the Discussion (Section 6.1.1), we first tried to apply the ARHSMM to simultaneously segment pose time-series from multiple keypoints on the subjects' bodies, while varying the aforementioned hyperparameters across reasonable ranges (number of latent states $[2, 3, \ldots, 10$ states$]$, autoregressive lag steps $[1, 2, \ldots, 5$ video frames$]$, latent state mean duration Poisson distribution prior per state $[15, 30, 60, 90$ video frames$]$). The observation distribution prior was a zero-mean unit-variance multivariate Gaussian distribution. Pose trajectories were mean-centered and standardized across all representative set

videos and keypoints for each subject before model fitting. In the resulting segmentation, we measured the following quantities: number of events produced across different states, the average length of each event for each state, and the average and maximum displacement of a subject's wrists for each state. We also looked at bi-grams and tri-grams (i.e. sequences of two and three consecutive states) in case state-*transitions* (and not the states themselves) produced a more interesting segmentation of the data. Unfortunately, these quantitative metrics did not provide us any significant guidance in deciding upon segmentation hyperparameters. However, manually inspecting video clips associated with each state (or bi/tri-gram) revealed a variety of movements that had been segmented with high temporal precision. Of these, only a few of these were human-interpretable (e.g. single handed reaches, double handed reaches, head movements, complex combinations of multiple keypoint movements etc.). Event-averaged spectrograms or power-spectral densities for the different behavioral states generated using the associated ECoG data, were not easily distinguishable either by manual inspection or by the use of a (Random Forest) classifier.

Pose trajectory segmentation for manuscript: Given the high variability and large scale of the data that we were working with, we chose to *prioritize human-interpretability of the segmentation and the ability to compare with prior literature*, and decided to use the simplified two-state segmentation approach described in the main text. We separately segmented the pose time-series for each of the wrists and nose keypoints, using an ARHSMM with two latent states and an autoregressive lag of one. Before segmentation, each keypoint's time-series was converted to radial distances (i.e. $r = \sqrt{x^2 + y^2}$, with top-left corner of video frame as origin) to further simplify the segmentation task and enable it to scale well to long ($7 - 9$ day) pose time-series. With hyperparameters set to two latent states and an autoregressive lag of one, it is easy to recognize the state with fewer occurrences as the *move* state. Changing either the number of latent states and autoregressive lag makes interpretation of states more complex and was therefore avoided. The observation distribution prior was set to a zero-mean unit-variance Gaussian distribution, and the duration distribution prior was set to a Poisson distribution with a mean duration of 60 frames (2 seconds). However, in practice, we found that the segmentation was indifferent to changes in the prior distributions over the parameters. This is likely due to the fact that the dataset

being fit is enormous and overwhelms the influence of the prior.

Event extraction using regular expressions: We found *movement initiation* and *no-movement* events by performing regular-expression based pattern matching on the discrete state sequences obtained from the ARHSMM-based segmentation. To find *movement initiation* events in each wrist, we use the following Python regular expression to find instances of 15 consecutive *rest* states (0.5s), followed by at least 15 consecutive *move* states (0.5s): `re.compile(''.join([str(x) for x in [0]*15 + [1]*15]))` (`0` represents *rest*, and `1` represents *move*). *no-movement* events are defined to be state sequences of 90 *rest* states (3.0s) across both wrists and the nose. To find these, we first convert discrete state trajectories from both wrists and nose into a single state trajectory by marking a time-step as *rest* only when all three keypoints are simultaneously at *rest*, and marking it as *move* otherwise. We then use the following Python regular expression to find instances of *no-movement* events: `re.compile(''.join([str(x) for x in [0]*90]))`.

Lastly, we repeated the aforementioned two-latent-state segmentation process using a similar representative set of video clips for all 12 subjects with the same hyperparameters as used for the first 4 subjects. A manual inspection of a few dozen random events of each movement type confirmed that the temporal accuracy of event boundaries matched our expectations. Finally, this process was scaled out to the entire duration of the pose data for all 12 subjects, followed by event metadata extraction as described in Section 3.3.1.

3.7 Supplementary Tables

Subject	Implant Hemisphere	# Surface Electrodes	# Depth Electrodes
01	L	86	8
02	R	70	16
03	L	80	0
04	R	84	0
05	R	106	0
06	L	80	0
07	R	64	0
08	R	92	0
09	L	98	28
10	L	86	40
11	L	106	0
12	L	92	24

Table 3.1: **Electrode information for subjects:** Grid and strip electrodes placed on the cortical surface are referred to as Surface electrodes, while Depth electrodes reach deep cortical and subcortical areas. We have only analysed up to 64 Grid electrodes for each patient. However, brain data and metadata for each electrode is included in the dataset accompanying this manuscript.

Subject	Magnitude [px]	Duration [frames]	Bimanual lead-lag [frames]	Bimanual overlap [frames]	Angle mode [degrees CCW]
S01	88.09	34.78	-8.45	7.74	90.0
S02	83.19	32.06	-10.14	11.02	90.0
S03	90.43	34.20	-9.19	10.51	90.0
S04	102.59	34.99	-5.22	6.86	130.0
S05	103.27	32.24	-9.23	9.84	70.0
S06	80.42	40.74	-5.40	10.48	90.0
S07	76.53	37.47	0.07	5.92	110.0
S08	151.37	39.96	-5.77	8.56	130.0
S09	65.83	38.95	-6.35	5.32	90.0
S10	104.53	40.76	-5.12	9.29	90.0
S11	73.25	36.20	-7.88	9.38	90.0
S12	88.09	40.04	-6.75	7.24	70.0

Table 3.2: **Quantitative metrics for right-wrist movements**: Additional data accompanying distributions visualized in Figure 3.6. All quantities shown are medians of respective distributions. Video frame-rate is 30 frames/second. Angle mode (in degrees, measured counter-clockwise with 0° to the right) is the center of the histogram bin with the highest number of events (histogram bin-size is 20°.)

Subject	Magnitude [px]	Duration [frames]	Bimanual lead-lag [frames]	Bimanual overlap [frames]	Angle mode [degrees CCW]
S01	94.66	33.08	-9.63	7.61	50
S02	90.63	33.16	-11.12	11.74	350
S03	97.20	34.50	-9.98	10.00	70
S04	89.22	30.04	-13.81	13.24	90
S05	92.39	30.17	-12.89	12.19	70
S06	81.42	43.18	-6.79	11.34	90
S07	60.67	35.22	-11.57	13.52	110
S08	167.36	38.33	-9.93	11.31	110
S09	81.96	43.13	-6.20	5.92	70
S10	107.64	42.22	-6.68	10.50	70
S11	85.37	36.17	-6.43	7.76	70
S12	84.73	39.92	-7.14	7.72	70

Table 3.3: **Quantitative metrics for left-wrist movements**: Additional data accompanying distributions visualized in Figure 3.7. All quantities shown are medians of respective distributions. Video frame-rate is 30 frames/second. Angle mode (in degrees, measured counter-clockwise with 0° to the right) is the center of the histogram bin with the highest number of events (histogram bin-size is 20°.)

Chapter 4

APPLICATIONS OF NATURALISTIC UPPER-LIMB MOVEMENT EVENTS

In this chapter, we demonstrate the efficacy the event generation pipeline developed in Chapter 3 as a data source for two prototype applications, (1) neural correlates of movement initiation and (2) decoding naturalistic movement initiation. These prototypes have been extensively developed upon in two companion papers [59] and [75].

4.1 Neural correlates of movement initiation

A core scientific question in systems neuroscience is how behaviors are encoded by the coordinated activation of brain regions. To examine the neural correlates of naturalistic movement initiation, we performed a time-frequency (TF) analysis of the neural recordings [76] by averaging event-locked spectrograms for each subject, using hundreds of movement initiation events chosen to match movement statistics (reach magnitude, onset velocity, and shape) of a previous controlled experimental study [41]. Using the aforementioned metadata to guide our search, we selected up to 200 events per day over 5 days for each of 12 subjects, and then further inspected the video for each event to remove any false positives (17.8% mean $\pm$ 9.9% s.d. events). (See Table 4.1 for full list of events selected per subject.)

We observed (see Figure 4.1) movement-associated power increases in a high-frequency band (76–100 Hz) and decreases in a low-frequency band (8–32 Hz) across four cortical areas (Precentral, Inferior Parietal, Postcentral and Supramarginal), with strongest responses in the Precentral area [41, 62].

4.2 Decoding naturalistic movement initiation

A grand challenge in neuroengineering is the development of BCIs that can be used to predict spontaneous activity and intentions outside the lab, in everyday settings [78–82]. Here we performed a preliminary study leveraging our pipeline as a source of training data for a BCI decoder that detects wrist movement initiation events. Specifically, we trained separate classifiers, tailored to each subject, to discriminate between movement initiation events and no-movement epochs for each wrist using only features derived from the ECoG neural recordings.

Our decoder uses the Random Forest (RF) algorithm [73, 83], which is typically considered one of the best off-the-shelf classification algorithms for small/medium sized datasets

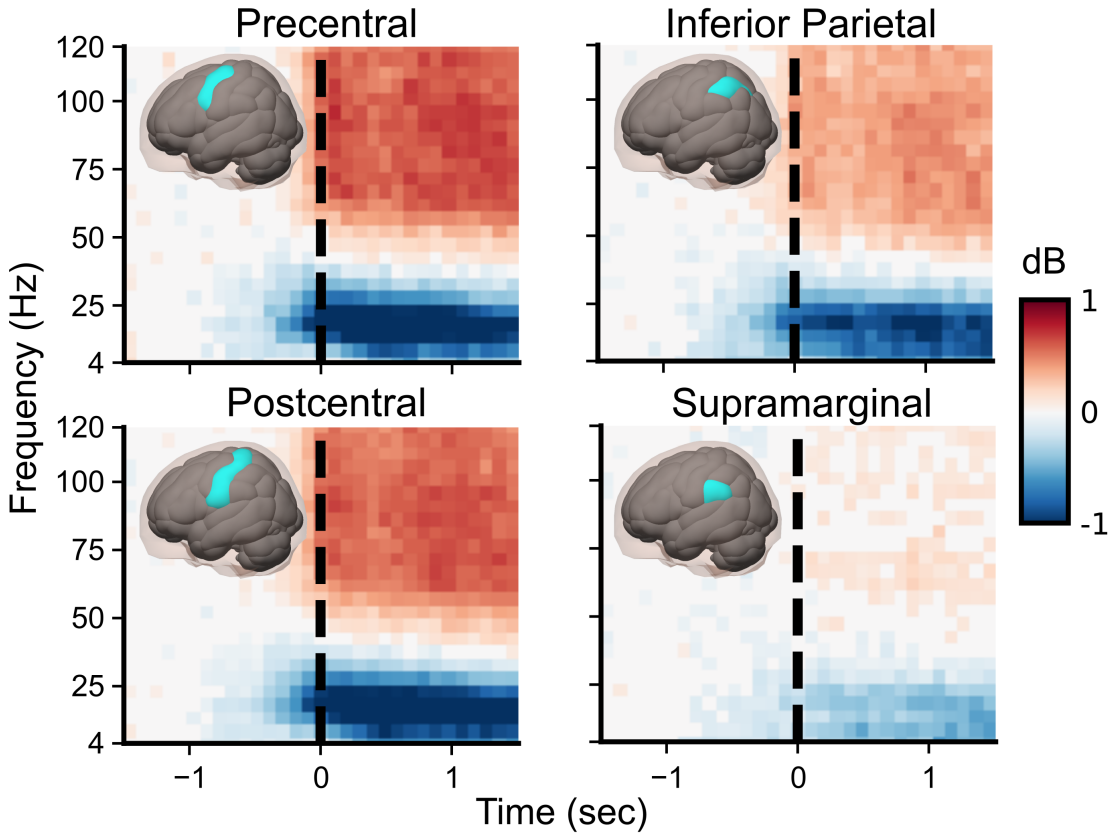

Figure 4.1: Neural correlates of movement initiation: Event-locked spectrograms, averaged by brain region (cyan color in insets) across 12 subjects, showed movement-associated high-frequency power increase and low-frequency power decrease. These patterns corroborate and strengthen previous findings from controlled experiments [41]. See our companion preprint [77] for a deeper exploration of the behavioral and neural variability of these movements.

[84]. We used ECoG data 0.5s before to 0.5s after each event to compute TF spectrograms at each of the 64 grid electrodes and used the flattened vector of TF bins as features for the classifier (TF bins were 200ms $\times$ 5Hz resolution, truncated at 150 Hz; approximately 9000 features total).

Given that the brain's response can drift over the course of days [85, 86], a reduced subset of events from 3 consecutive days (typically days 3 through 5 of clinical monitoring) were used. From these, we used events from the first two days as the training set and events from the last day as the test set. (See Table 4.2 for exact list of days chosen for each subject.) To eliminate the confound of movement initiation in the opposing wrist, we then excluded events with significant movement ($\geq$ 0.2 seconds) in the opposing wrist within the $\pm$0.5s window used for ECoG data. Positive (movement initiation) and negative (no-movement) examples were balanced by down-sampling negative examples. This balancing eliminated bias in the training set and sets up a theoretical baseline performance of 50% accuracy for test set performance. Training and test supports were 633 mean $\pm$ 417 s.d. and 331 mean $\pm$ 203 s.d. examples, respectively. (See Table 4.2 for exact per-subject values.) We tuned the RF using a 20-trial randomized search over two hyperparameters: number of trees (range: $[50, 250]$) and maximum tree-depth (range: $[3, 15]$). For each set of hyperparameters, 5-fold cross-validation holdout accuracy was used to measure performance. Final performance reported is from training using best hyperparameters and corresponds to classifier accuracy on events from the withheld test day. Since false positives (FPs) in the event data establish a ceiling on classifier accuracy, we estimated their prevalence by manually inspecting 100 randomly sampled events per event-type from each subject (5% mean $\pm$ 10% s.d. for no-movement, 22% mean $\pm$ 16% s.d. for contralateral, and 14% mean $\pm$ 10% s.d. for ipsilateral events, aggregated over all subjects).

To interpret the importance of spectral features in the decoder, we visualized Random Forest feature importance scores [83, 84]. We aggregated these scores in two ways to gain insight into their spatial and frequency components. Spectral features are indexed by electrode, time, and frequency. We define Feature Importance aggregated by Electrode $FI_E(e)$

for electrode $e \in \mathcal{E}$ as:

$$FI_E(e) = \sum_{f \in \mathcal{F}, t \in \mathcal{T}} FI(t, f, e),$$

where $\mathcal{E}, \mathcal{F}$ and $\mathcal{T}$ are the sets of electrodes, frequency-bins, and time-bins over which the spectral features are calculated, respectively. For the purpose of visualization, we normalized these values to get Normalized Feature Importance aggregated by Electrode $NFI_E(e)$:

$$NFI_E(e) = \frac{FI_E(e)}{\max_{e \in \mathcal{E}} FI_E(e)}.$$

To understand the contributions of various frequency-bins to decoding, we define analogous formulas for feature importances aggregated by frequency-bin:

$$FI_F(f) = \sum_{e \in \mathcal{E}, t \in \mathcal{T}} FI(t, f, e)$$

$$NFI_F(f) = \frac{FI_F(f)}{\max_{f \in \mathcal{F}} FI_F(f)}$$

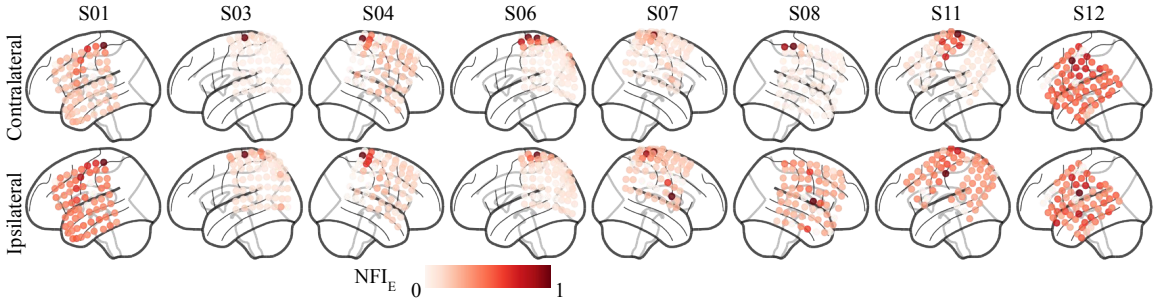

Figure 4.2: Contralateral and ipsilateral wrist movement initiation decoder feature importance scores aggregated by electrode (NFI_E), showing spatial contributions of different brain regions. Scores are normalized by dividing by highest electrode score for each decoder. Electrode coverage over motor cortex is associated with higher decoder accuracy; for instance, subjects having good motor cortex coverage (S07, S06, S03 and S11) have the highest decoding performance (Figure 4.3). See Figure 4.5 for plot with all 12 subjects.

Individual subject classifier performance varied widely between subjects, ranging from around chance levels (50%) to 80% on test accuracy (Figure 4.3). Classifier performance

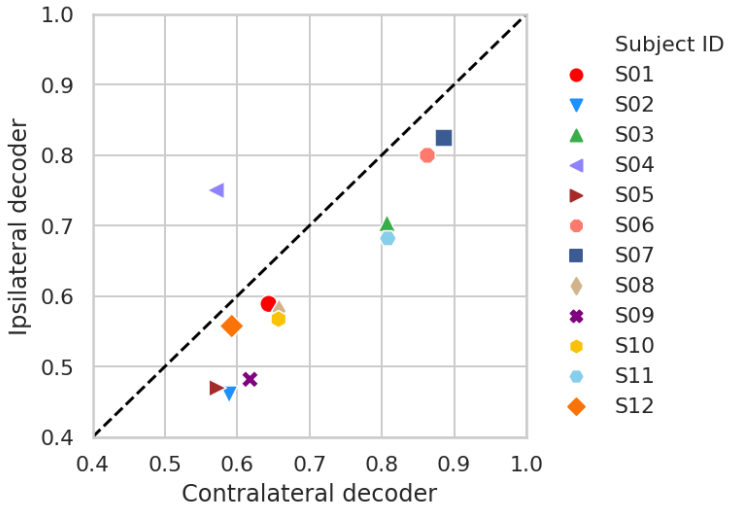

Figure 4.3: Test set decoding accuracy for initiation of movement of contralateral (side opposite electrode implant) and ipsilateral (same side) wrists: As expected, decoding of contralateral movements is slightly more accurate than ipsilateral in almost all cases.

tended to be associated with the extent of motor cortex coverage (Figure 4.2 and 4.5). Test set decoding accuracy (Figure 4.3) was observed to be higher for the contralateral wrist in almost all subjects. Figures 4.2 and Figure 4.4 show the spatial and spectral (frequency) contributions of the classifier's features respectively. We found (Figure 4.2; also see Figure 4.5) that electrodes over sensorimotor cortex, when available, dominated feature importance (e.g. Subjects S07, S06, S03 and S11 in Figure 4.3) in most but not all patients. As seen in Figure 4.4, we found that a low-frequency band ($<$ 35Hz) and a high-frequency band (around 100 Hz) dominates feature importance in most but not all patients. We extensively develop upon the limited prototype described here in our companion paper focused on decoding [75].

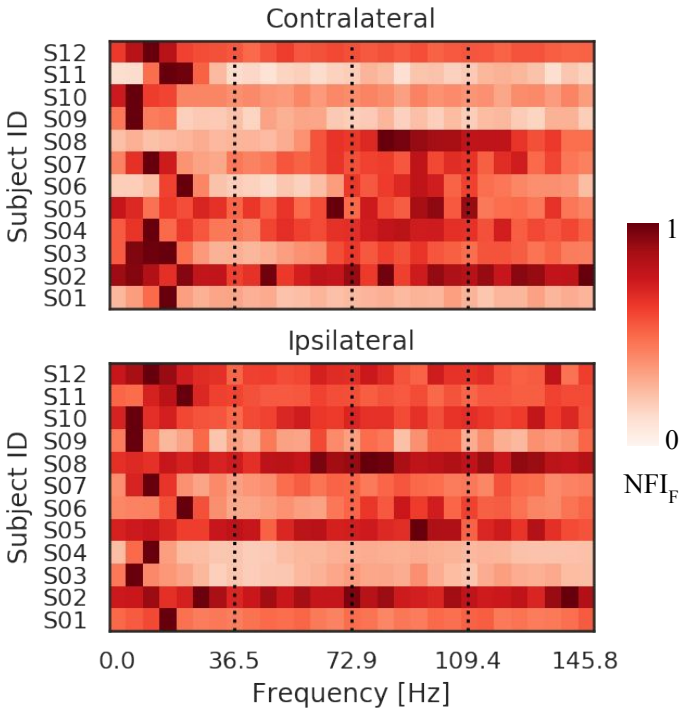

Figure 4.4: Decoder feature importance scores aggregated by frequency and normalized by dividing by score of highest frequency bin per subject (NFI_F). Heatmaps show that the most relevant spectral features tend to come from a low-frequency band (< 35Hz) and a high-frequency band (around 100 Hz), similar to prior findings [41]. When motor cortex electrode coverage is lacking (e.g. contralateral S02 and S05) or if ipsilateral wrist movement is being decoded, spectral feature contributions tend to be more broadly distributed across the frequency spectrum, and associated with lower decoding accuracy.

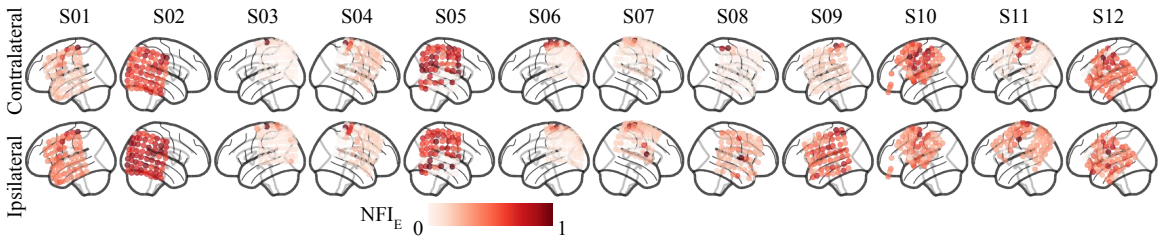

Figure 4.5: Contralateral and ipsilateral wrist movement initiation decoder normalized feature importance scores aggregated by electrode (NFI_E), showing spatial contributions of different brain regions for all 12 subjects. We see the same trend of motor cortex electrode coverage being associated with higher decoding accuracy, as was noted in Figure 4.2. Subjects having good motor cortex coverage (S07, S06, S03 and S11) have the highest decoding performance (Figure 4.3). Additionally, we see that electrodes with high normalized feature importance tend to be more spatially localized in the case where good motor cortex coverage is available.

4.3 Supplementary Tables

Subject	Eligible events	Events selected	False positives
S01	741	659	82
S02	305	209	96
S03	642	584	58
S04	417	268	149
S05	267	203	64
S06	956	869	87
S07	800	675	125
S08	965	850	115
S09	211	151	60
S10	806	620	186
S11	982	896	86
S12	996	947	49

Table 4.1: **Manual event selection for neural correlates of movement initiation**: Table shows the exact number of events Eligible, Selected and number of events Discarded as False Positives for each participant. Eligible events refer to events that passed certain critera for movement statistics including magnitude, onset velocity, and complexity (using shape metadata). See our companion preprint [59] for more details.

Subject	Wrist	Side	Training [Test] Days	Training Support	Test Support	%FP R, L, [NM]	Accuracy
S01	R	C	3, 4, [5]	526	216	27, 15, [5]	0.64
	L	I		474	450		0.59
S02	R	C	3, 4, [5]	218	90	19, 39, [16]	0.59
	L	I		296	176		0.46
S03	R	C	2, 3, [4]	1072	332	11, 12, [13]	0.81
	L	I		944	388		0.70
S04	R	C	7, 8, [9]	294	128	12, 64, [6]	0.57
	L	I		1332	708		0.75
S05	R	C	2, 3, [4]	114	96	38, 23, [4]	0.57
	L	I		216	164		0.47
S06	R	C	3, 4, [5]	1036	540	18, 21, [3]	0.86
	L	I		550	534		0.80
S07	R	C	3, 4, [5]	138	166	4, 1, [2]	0.89
	L	I		596	704		0.82
S08	R	C	3, 4, [5]	908	190	6, 19, [3]	0.66
	L	I		1712	254		0.58
S09	R	C	3, 4, [5]	272	238	10, 2, [4]	0.62
	L	I		188	330		0.48
S10	R	C	3, 4, [5]	432	140	26, 20, [3]	0.66
	L	I		446	134		0.57
S11	R	C	3, 4, [5]	784	240	20, 14, [2]	0.81
	L	I		834	660		0.68
S12	R	C	3, 4, [5]	1104	584	11, 10, [1]	0.59
	L	I		704	488		0.56

Table 4.2: **Additional metadata for naturalistic movement initiation decoders:** Table shows Decoded Wrist (R: Right, L: Left), Electrode Implant Side with respect to decoded wrist (C: Contralateral, I: Ipsilateral), Hospital stay day from which Training and Test set is generated, Percent of False Positive (FPs) detected in a random 100-event subset of these events (R/L: Right/Left wrist movement initiation, NM: No movement), and Training and Test set Supports (# events), and Decoder Test Set Accuracy (Theoretical baseline: 0.50) for each Subject. Note that both S03 and S05 had video recording issues for parts of day 5, so we started a day earlier. Subject S04 has a high percentage of FPs on their left hand because they tended to keep it under a blanket. Subject S09 has relatively fewer events due to omit-listing of long stretches of time when they kept both their hands under a blanket or times when they were asleep. We selected 3 consecutive days (best-effort) such that both the training and test sets for each hand had at least ≈ 100 events.

Chapter 5

EMERGENT BEHAVIOR AND NEURAL DYNAMICS IN ARTIFICIAL AGENTS TRACKING TURBULENT PLUMES

5.1 Introduction

Locating the source of an odor in a windy environment is a challenging control problem due to the intermittency of the odor signal and the variability of the wind direction and plume shape over space and time. Moreover, an agent attempting to track an intermittent plume needs memory, where current and past egocentric odor, visual and wind sensory signals must be integrated to determine the next action. For flying insects, localizing the source of odor plumes arising from potential food sources or mates is critical for survival and reproduction. Therefore, many aspects of their plume tracking abilities have been experimentally studied in great detail [48, 49, 87, 88]. However, most such studies are limited to one or two levels of analysis such as behavior [46], computation [89, 90] or neural implementation [91]. Moreover, despite the wide adoption of wind tunnel experiments [45], generating controlled dynamic turbulent plumes and tracking flight trajectories at high resolution remain expensive and laborious. Whereas behavioral experiments in wind tunnels are tractable, collecting significant neural data during free flight in small insects remains technologically infeasible, and larger insects require larger wind tunnels (see also platforms that use virtual reality [92]). Here we describe a complementary *in silico* approach using artificial recurrent neural network (RNN) agents trained to track simulated turbulent plumes, with the goal of developing an integrated understanding of the behavioral strategies and the associated neural computations that support plume tracking.

In recent years, artificial neural networks (ANNs) have been broadly applied to model and understand different aspects of neural function and behavior, including the visual [93], motor [94] and navigation systems [95–98]. Whereas many of these ANNs have been trained using supervised approaches that rely on labeled trianing data, an alternative emerging algorithmic toolkit, known as deep reinforcement learning (DRL), has made it computationally tractable to train ANNs in the 'agent' setting (Figure 5.1d). In particular, an ANN agent receives sensory observations and task-aligned rewards based on its actions at each step and tries to learn a strategy to maximize its total expected reward [99, 100]. Such learning and optimization based models can be *normative* in the sense that it can prescribe how a neural system *should* behave, rather than describing how it has been observed to behave. More-

over, the process of designing an optimization objective provides insight into systems at an abstract level that descriptive approaches typically cannot [101]. As neuroscience move towards studying increasingly naturalistic behaviors [7, 8, 53], such approaches are becoming increasingly attractive as a way to gain insight, rapidly explore experimental hypotheses, and generate ideas for theoretical development while capturing real-world complexity across multiple levels of analysis [102–106].

Flying insects search for sources of odor using several strategies, depending on the spatial scale being considered and odor source visibility [48] (see Figure 5.1a): Close to the odor source, insects can fly along a straight-line path to the source guided by vision. At longer ranges (from several meters up to about 100 meters [107]) or when the odor source is not yet visible, they can only use their mechanosensation to estimate wind velocity and olfaction to guide their search. At this larger scale, behavioral studies have reported a few stereotyped behavioral sequences important for plume tracking: *upwind surges* when the insect can sense the odor, and *crosswind casts* and *U-turns* to locate the plume body when the insect loses the odor scent [49]. In this paper, we focus on this the larger-scale odor and wind guided regime, where agents have access to only mechanosensory and olfactory cues.

In this paper, we train and analyze RNN agents to track odors in a flexible plume simulation. We analyse our trained agents at multiple levels of analysis including behavior, neural activity and connectivity and explore connections between the emergent features and their biological counterparts: First, we analyze the emergent modular behavior to find that the behavior modules resemble *upwind surging*, *crosswind casting* and *U-turn* behaviors previously reported in flying insects (Section 5.4.1). Second, inspecting tracking episodes in non-stationary wind-direction plumes, we find that agents tend to track plumes with respect to their local shape rather than with respect to the current wind-direction – a finding that motivates future experimental work (Section 5.4.2). Third, to understand the memory requirements of the plume tracking task, we compare our RNN-based agents with alternative feedforward architecture agents with constrained sensory memory (Section 5.4.3). We find that plumes that do not change in direction can be tracked using only a few timesteps of memory, but longer timescales are essential for more complex non-stationary plumes. Fourth, like in real neurobiological recordings, we find that the neural activity

associated with the agent behavior is low-dimensional (Section 5.4.4). Also, we find that the RNN learns to represent variables that are known to be important to flying insect navigation, such as head-direction and time between odor encounters. Fifth, we provide an abstract description of neural dynamics that support plume tracking. We find that the neural dynamics associated with the emergent behavior modules are structured into two distinct regimes with quasi-periodic structure (Section 5.4.5), and transitions between these regimes are asymmetric in duration (Section 5.4.6). Sixth, we report some empirical findings about training-related connectivity and stimulus integration timescale changes in the RNNs (Section 5.4.7). These abstract descriptions and empirical findings motivate future theoretical studies into RNNs and artificial agents that could benefit both the biological and artificial neural networks communities. Finally, to facilitate future work in this area, we will (on manuscript publication) make our trained agents, accompanying behavioral and neural datasets, and code publicly available.

5.2 Related work

Much work in recent years has used DRL to train ANNs that solve tasks closely inspired by tasks from neuroscience. For instance, agents have been trained to study learning and dynamics in the motor-cortex [108, 109], time encoding in the hippocampus [110], reward-based learning and meta-learning in the pre-frontal cortex [111–113], and task-associated representations across multiple brain areas [114]. (also see recent perspectives discussing the relevance of this emerging algorithmic toolkit to neuroscience [115, 116] and ethology [117].)

In particular, our work is most closely related to the following three recent research efforts. [103] recently developed a virtual-reality model of a rodent embodied in a true-to-scale skeleton body and endowed with a deep ANN 'brain.' They trained this model using DRL to solve four tasks and then analyzed the virtual rodent's emergent behavior and neural activity to find similarities at an abstract level between their agent and findings from biological rodent studies. [118] studied the trail tracking strategies of terrestrial animals with one (e.g. one antenna) or two (e.g. two nostrils) odor sensors. They found that RL agents trained on simulated trails recapitulate the stereotypical zig-zagging tracking

behavior seen in such animals. Using a static trail model and an explicit (non neural) probabilistic model for sensory integration, they theoretically and empirically studied the effect of varying several agent and task parameters on the stereotypical zig-zagging behavior that emerged in their trained agents. [119] used a biologically detailed spiking neural circuit model of a fly mushroom body to study sensory processing, learning, and motor control in flying insects when foraging within turbulent odor plumes.

Our work differs from the recent papers studying artificial agents solving neural inspired tasks in a few key ways. We model a different organism and task compared to [103, 118]. Unlike [103], but similar to [118] and [119], we abstract away biomechanical details of the modeled organism's body and joints. We use neural networks, like [103] and [119] (and unlike [118]), but do not impose biological connectivity constraints like [119] or impose network modularity like [103]. Like [103], but unlike [119], we train artificial (rate-coding) recurrent neural networks (RNNs) end-to-end, to learn task-adapted behaviors, neural representations, neural dynamics and network weights from reward alone. We use simpler 'Vanilla' RNNs compared to the gated RNNs used in [103], to enable comparisons with previous dynamical-systems and connectivity analyses in the computational neuroscience literature [51, 52, 120–122]. We adopt an analysis approach from a growing literature that models neuroscientific and cognitive tasks using RNNs and then analyses the resulting (artificial) neural activity. Neural activities are analyzed at the whole-network level rather than at the individual neuron level, which provides us with an abstract understanding of task-relevant neural computations that is robust to small changes in network architecture and training hyperparameters [51, 121–123]. Finally, our plume tracking task is computationally harder than [118, 119] due to the use of a configurable, dynamic and stochastic environment, and is closer in complexity to the multi-task setting of [103]. Yet, since we do not model vision or joint-level motor control as in [103], our neural networks are simpler and can be trained on a computational budget accessible to an academic lab.

5.3 Methods

5.3.1 Plume simulation

We implement a particle-based two-dimensional plume simulation model (see Figure 5.1f) that replicates the statistics of real-world turbulent plumes [43], and has been used in a wide range of domains including olfactory navigation [49], robotics [124] and sensor networks [125]. The simulator (see Figure 5.1b) comprises a spatially homogeneous wind vector-field (0.5 m/s with configurable direction) and an odor source located at the origin that emits a random Poisson distributed number of odor puffs at each simulation iteration. Puffs are initialized with a fixed initial radius and undergo a fixed-rate radial diffusion step at each simulation iteration. Each emitted puff undergoes advection in the wind velocity field and Gaussian random crosswind translation at each simulation iteration. Effectively, each puff performs a biased random walk downwind over time, while diffusing out spatially. Our simulated plumes are two-dimensional for simplicity of analysis, and we neglect the effects of any agent movements in the direction perpendicular to the ground plane. The dimensions of the simulated arena are $[-2m, +10m]$ and $[-5m, +5m]$ in the x and y coordinates respectively, totaling a $120m^2$ arena. Plumes are simulated at 100 iterations/second. The plume's centerline is obtained by simulating puffs that have no random crosswind translation at each iteration (Figure 5.1f).

We simulate the following four wind configurations. First, where the wind-direction is held constant (0°) throughout the simulation ('constant'). Second, where the wind-direction makes one 45° counter-clockwise switch during a tracking episode ('switch-once'). Third, where the wind-direction switches at multiple random times during a tracking episode ('switch-many'). Each wind-direction turn is a random draw from a Gaussian distribution with mean 0 and 45° s.d., truncated at ±60°, and occurs approximately every 3 seconds. Fourth, where the wind-direction is held constant, but puff birth-rate is reduced (0.4x) compared to the constant configuration ('sparse'). See 5.6 for further details on plume simulation.

5.3.2 Agent architecture

Our agents are actor-critic networks (see Figure 5.1e), each comprising a recurrent neural network (RNN) followed by a parallel pair of two consecutive feedforward layers corresponding to the actor and critic heads of the network [44]. The Actor head implements a control policy that maps the RNN's learned state representation to control actions, while the Critic head implements a value function that maps the state representation to a reward-derived scalar value which is used only during agent training and not thereafter. In the DRL literature, two-layer deep heads are typically regarded to be sufficiently expressive for any control problem. At each time step, an agent receives a 3-dimensional real-valued input vector comprising egocentric wind velocities (x, y) and odor concentration at its current location. In response, agents produce continuous valued turn (maximum $\pm 6.25\pi$ radians/s) and forward-movement (maximum 2.5 m/s) actions that are matched to the capabilities of real flying insects [45, 46]. Additionally, to understand the role of memory on tracking performance (section 5.4.3), we use an alternative feedforward-only network (MLP) architecture with fixed-length memory, simulated by appending historical sensory observations onto instantaneous network inputs [126]. Though such MLPs are far from being biologically plausible architectures, they serve as useful tools for abstract comparison since their memory capacities can be controlled precisely. All layers in our networks are 64 units wide with *tanh* nonlinearities. In contrast to the orthogonal initialization typically employed in the mainstream machine learning literature [127], we initialize our RNNs with normally distributed weights to facilitate comparisons with the computational neuroscience literature [128–130].

5.3.3 Agent training and evaluation

We train our agents using the Proximal Policy Gradient (PPO) algorithm [47], which is known to robustly solve continuous observation-space continuous action-space control problems without needing significant hyperparameter tuning. To guide agent training, we use a simple reward function that greatly rewards homing in on odor source, mildly rewards actions that reduce the radial distance between agent and odor source, and penalizes longer

duration trajectories and straying too far from the plume. We train 14 independently randomly initialized networks for each architecture type, i.e. RNNs and MLPs with 2, 4, 6, 8, 10 & 12 timesteps of observation history. Then, we evaluate each trained agent over a behavioral assay comprising a set of 15 fixed initial locations, 2 fixed initial simulation timestamps and 8 fixed initial agent directions totaling 240 episodes, for each of the constant, switch-once and switch-many plume configurations. For each architecture type, we select the top-5 seeds with the best performance as measured by total number of successful episodes across the aforementioned plume configurations. Agent training/evaluation episodes are run at 25 frames per second on a sub-sampled plume, and limited to 300 frames/timesteps (12 seconds of flight) per episode for faster DRL training. See 5.6 for additional details on agent training and evaluation, and 5.7 for a full list of associated [hyper]parameters.

5.4 Analysis of trained agents

We now analyze our trained agents at different levels, starting from behavior, to neural representation and dynamics, and finally, connectivity. Unless otherwise specified, this section describes results from agent 3 (of 5 trained RNNs). These analyses use a randomly selected 120 episode *evaluation subset* of the 240 evaluation episodes for each of the constant, switch-once and switch-many plume configurations. The selected episodes are balanced to include an equal number (60 episodes each) of successful and unsuccessful plume tracking episode outcomes. Whenever there are fewer than 60 episodes of either outcome type (successful or unsuccessful) for any plume configuration, then the selection is trimmed to use an equal number of episodes of the smaller outcome type.

5.4.1 Trained agents can track plumes across varying wind conditions and the emergent behavior exhibits modularity

Our trained RNN agents are able to complete the plume tracking task across varying levels of wind-direction non-stationarity and plume puff birth-rates (see Figure 5.2 for example trajectories). Through visual inspection of trajectories, we find that agents appear to primarily be implementing one of three behavior modules, determined approximately by the time elapsed since the agent last sensed odor (Figure 5.3). We name these three regimes

tracking, *lost* and *recovering*. In the *tracking* module, the agent rapidly moves closer to the plume source, using either straight-line trajectories when it is well within the plume, or a quasi-periodic 'plume skimming' behavior where it stays close to the edge of the plume while moving in and out of it. The interval between the agent's encounters with the plume in this regime is within 0.5 seconds. *Lost* corresponds to a periodic behavior that appears variably across trained agents as either a spiraling in place or a slithering/oscillating motion, with possibly also a slow drift in an arbitrary direction. It is seen when the agent has not encountered the plume for a relatively long time (typically over 1 second). *Recovering* corresponds to a periodic or irregular behavior intermediate to *tracking* and *lost* where the agent makes large (usually cross-wind) motions after having lost track of the plume for a relatively shorter time (between about 0.5 second and 1.0 second) before the *lost* behavior kicks in. See supplemental section 5.8 for exact thresholds used to segment an agent's trajectories into behavior modules.

Manual inspection of several trajectories reveals that successful constant-wind plume tracking trajectories predominantly consist of the agent acting in the *tracking* and *recovering* modules (see associated animations: https://github.com/BruntonUWBio/plumetracknets/blob/main/constant.md). Inspecting successful trajectories across agents on the 'switch-once' (see associated animations: https://github.com/BruntonUWBio/plumetracknets/blob/main/switch-once.md) and 'switch-many' (see associated animations: https://github.com/BruntonUWBio/plumetracknets/blob/main/switch-many.md) plumes reveals that RNN agents employ more complex strategies in the face of changing wind-directions: If an agent is in the *tracking* module and well within the plume at the time of wind-direction change, it continues along its path without changing it's actions till it reaches the edge of the plume. If skimming the edge of the plume, it now has to compensate for the movement of the plume. We notice that agents appear to make more pronounced oscillations in and out of the plume, and this results in the slower progress in their movement towards the odor source. Furthermore, we notice that the shape of the oscillations appears to depend more on the local shape of the plume than the current direction of the wind (explored further in Section 5.4.2). Finally, if the agent cannot keep up with the movement of the plume, it typically orchestrates a sequence of large oscillations and spiral-like movements, corresponding

to the *recovering* and *lost* modules, to try to find the plume boundary. On returning to the plume, it once again resumes *tracking* module behaviors.

5.4.2 In switching wind-direction plumes, agents track the plume with respect to the plume centerline rather than the current wind-direction

A visual inspection of agent trajectories in non-stationary wind-direction plumes suggests that agents take into account the local shape of the plume, rather than just the current wind conditions (see Figure 5.2c-d and supplementary animations mentioned in Section 5.4.1). To quantify this, we look at the empirical distributions of the agent's course-direction as measured with respect to the current wind-direction, and with respect to the centerline (see Figure 5.1f) of the nearby plume. The agent's course-direction (see Figure 5.1c) is defined as the direction of its instantaneous movement with respect to the ground. Subtracting the current wind-direction angle from the course-direction provides the course-direction with respect to the wind. To find the course-direction with respect to the centerline, we first find the median centerline angle using centerline puffs (Section 5.3.1) within a ± 2 c.m. band of the x-coordinate of the agent location, and subtract this from the course-direction with respect to the ground. To generate the empirical distributions, we aggregate timesteps ($N > 2500$) from up to 60 successful trajectories for each of the three plume configurations from our randomized evaluation subset, and consider only timesteps belonging to the *tracking* behavior module. Additionally, for the 'switch-once' configuration, we trim trajectories to consider only the timesteps after the wind-direction switch has occurred.

Figure 5.4 shows the empirical distributions for Agent 3. For 'switch-once' plumes, we see that the (highest) mode of the centerline-referenced course-direction distribution tends to be much more centered than the mode for the wind-referenced distribution. This implies that, predominantly, the agent's flight is aligned (anti-parallel) with the plume centerline, but at an $\approx 45°$ angle with respect to the current wind-direction. We see the same trend again in the 'switch-many' regime, where the wind-referenced distribution seems to indicate that the agent spends a significant proportion of time at an angle diverging from the upwind-direction. However, the centerline-referenced course-direction distribution reveals that the

agent's trajectory is actually aligned with (anti-parallel to) the plume centerline. These trends are seen to hold across all 5 RNN agents (see 5.9 for equivalent data other agents.)

5.4.3 Recurrence supports expanded behavioral repertoire, but simple plume tracking is possible with just a few timesteps of memory

To understand the role of memory capacity in plume tracking, we compare the performance of our trained RNNs to trained feedforward networks (MLPs) with varying amounts of sensory memory. As seen in Figure 5.5, RNNs outperform MLPs across plume tracking tasks of different levels of difficulty, with more pronounced differences on tougher tasks. For MLPs, longer duration sensory memories can lead to much better performance on tougher tracking tasks where the plumes switch more often or are sparser.

5.4.4 Neural activity is low dimensional and represents biologically relevant variables

We now turn our attention to the neural activity that is simultaneously logged while the agents are performing plume tracking. Following a trend moving towards analyzing network populations rather than individual units [123, 131] and due to the abstract nature of our model, we choose to interpret the entire hidden state rather than try to characterize individual units. Principal Components Analysis (PCA) of neural activities aggregated across multiple plume conditions (constant, switch-once & switch-many) reveals a low dimensional structure with 90% of data variance explained by the first-5 Principal Components (see Figure 5.6g).

Next, we look at the variables that the RNNs have learned to represent in its hidden state to get insights into the type of computations it is performing. We find visual evidence (see Figure 5.6a–d) that our RNNs have learned to represent at least three biologically relevant quantities beyond the instantaneous sensory observations that are provided to it by the simulator: (1) agent head-direction, i.e. the orientation of the agent with respect to the ground, (2) time since odor (the plume) was last encountered, which seems to be involved in determining transitions between behavior modules, (3) an exponentially-weighted moving-average recent odor encounters, which we conjecture is a quantity that would be

helpful in the face of an intermittent odor signal arising from a turbulent plume, and (4) an exponentially-weighted moving-average of a discretized odor encounter signal (discretized to be 1 at the first timestep of a stream of perceptible odor inputs and 0 elsewhere.) We determine the window-sizes for (3) and (4) by linearly regressing neural activity onto them for sliding-windows of varying lengths, and choosing the window-size that produces the best fit as measured by the coefficient of determination R^2 (see Figure 5.6e).

To quantify how important these represented variables are to actual task performance, we train a Random Forest (RF) [132] classifier to predict actions taken by the agent over successful trajectories. We uniformly partition the Turn and Move action variable domains into 3 and 2 discrete classes respectively, roughly corresponding to 'left', 'center' and 'right' turns, and 'fast' and 'slow' forward movements. These are concatenated to form a 6-class independent variable. The classifier receives instantaneous sensory observations (egocentric wind speed x and y coordinates w_X, w_Y and odor concentration) and the four aforementioned represented features (1–4) as inputs. Training and test sets are a randomized non-overlapping 80%-20% split of evaluation episodes, balanced across plume configuration and episode outcomes. We do a 20-trial 3-fold cross-validated randomized search over the number-of-estimators (range: [10,50]) hyperparameter, and then train a classifier using the best hyperparameter on the whole training set.

Classifier accuracy for Agent 3 (shown in Figure 5.6) is 83%, which higher than a baseline accuracy of 69% using a classifier receiving just instantaneous sensory observations, and well over the 40% accuracy produced by a dummy majority-class classifier. Represented variables have permutation importance scores [132, 133] within the range covered by the importance scores associated with instantaneous sensory inputs. Note that the estimates provided by this analysis are (unavoidably) approximate due to the discretization of the action data and correlations between features. See 5.10 for results for all 5 agents.

5.4.5 Neural dynamics are characterized by limit cycle like regimes and a transition region

We now look at how an agent's neural activity evolves over the course of a plume tracking episode. Unlike previous papers that were able to characterize the nonlinear dynamics of

their task-performing RNNs by inspecting their fixed-point (FP) manifolds [51, 121, 122], we did not find any such FP structures in our RNNs to do so. Instead, we take an empirical approach and visualize the neural activity to inspect if there is any structure in the dynamics that could be used to characterize it. Indeed, we find that the neural dynamics of our RNNs appear to organize themselves into overlapping but distinctly structured regimes associated with the *tracking* and *lost* behavioral modules (see Figure 5.7). Interestingly, the periodic oscillations (e.g. spirals) seen in the *lost* behavioral state appear to be represented as quasi-limit-cycles in the neural activity (Figure 5.7d). Similarly, the neural dynamics associated with the *tracking* behavior are represented as quasi-periodic 'funnel' like structure (Figure 5.7c). We also see an amorphous transition region associated with the *recovering* behavioral state. Except for one agent, we see the same approximate structure (limit-cycles and funnel) emerge in the neural dynamics for each of the 5 RNN agents. See 5.11, and animation of individual episode dynamics: https://github.com/BruntonUWBio/plumetracknets/. Also see supplementary Table 5.7 for the exact period associated with each RNN's *lost* limit-cycle.

5.4.6 Macroscopic transitions between neural activity regimes are asymmetric in duration

After having found distinct (*tracking* and *lost*) neural activity regimes in the previous section, we now explore differences in transition durations between regimes during the course of plume tracking. Specifically, we look at differences in the durations between (1) when an agent enters the plume and when it 'enters' the *tracking* neural activity regime, and (2) when an agent leaves the plume and when it 'enters' the *lost* neural activity regime. To define entry into a neural activity regime, we first define neural activity 'centroids' associated with the *tracking* and *lost* neural activity regimes by averaging the last 1-second of neural activity from evaluation episodes ending in successful homing (HOME), and those unsuccessfully ending in the agent going too far from the plume i.e. Out-Of-Bounds (OOB), respectively (see Figure 5.8). We then define clusters associated with the HOME and OOB centroids as being comprised of all neural activity with a distance $D/2$ units from the respective centroid, where D is the distance between centroids. Finally, for any unsuccessful tracking episode,

we calculate a 'time to *lost*' (TTL) as the duration between the agent leaving the plume and entering the OOB cluster. Similarly, for successfully homing episodes, we calculate a 'time to *track*' (TTT) as the time taken to enter the HOME cluster after entering the plume. To exclude small excursions outside the plume when the agent is skimming the boundary of the plume, we only consider sub-trajectories where the agent has spent at least the last 12 timesteps (≈ 0.5 seconds) outside the plume (effectively entering the *recovering* behavioral state). Since the agent could have spent even longer outside the plume before entry, i.e. entered the *lost* behavior state, we split TTT into 'time to *track* not *lost*' (TTT-NL), and 'time to *track* after *lost*' (TTT-L).

Comparing evaluation episodes across constant, switch-once and switch-many plumes, we find that TTL durations are significantly longer than TTT-NL and TTT-L durations (Figure 5.8). This trend holds across 4 out of 5 agents. See 5.12 for data on all 5 RNN agents.

5.4.7 Empirical observations on RNN connectivity

In this section, we report some empirical observations related to the weight matrices and recurrence Jacobians of our RNNs. The update rule for a (Vanilla) RNN is given by

$$\mathbf{h}_t = F(\mathbf{h}_{t-1}, \mathbf{x}_t) = tanh(\mathbf{W}_h \mathbf{h}_{t-1} + \mathbf{W}_x \mathbf{x}_t + b),$$

where F is a nonlinear function of the network hidden state $\mathbf{h}_{t-1}$, network input $\mathbf{x}_t$ vectors, and a bias term b respectively. For our RNNs with $tanh$ nonlinearities, $\mathbf{W}_h$ is the hidden-to-hidden layer recurrence matrix (connectivity), and $\mathbf{W}_x$ is the input-to-hidden layer matrix respectively [51]. This equation can be linearized around an arbitrary expansion point $(\mathbf{h}^{\mathrm{e}}, \mathbf{x}^{\mathrm{e}})$ to get a linear dynamical system approximation given by :

$$\mathbf{h}_t \approx F\left(\mathbf{h}^{\mathrm{e}}, \mathbf{x}^{\mathrm{e}}\right) + \mathbf{J}^{\mathrm{rec}}\big|_{(\mathbf{h}^{\mathbf{e}}, \mathbf{x}^{\mathrm{e}})} \Delta\mathbf{h}_{t-1} + \mathbf{J}^{\mathrm{inp}}\big|_{(\mathbf{h}^{\mathbf{e}}, \mathbf{x}^{\mathrm{e}})} \Delta\mathbf{x}_t,$$

where $\Delta\mathbf{h}_{t-1} = \mathbf{h}_{t-1} - \mathbf{h}^{\mathrm{e}}$ is the linear system state, $\Delta\mathbf{x}_t = \mathbf{x}_t - \mathbf{x}^{\mathrm{e}}$ is the linear system input, $\mathbf{J}^{\mathrm{rec}}$ is the recurrence Jacobian $\left(J^{\mathrm{rec}}_{ij}\Big|_{(\mathbf{h}^{\mathbf{e}}, \mathbf{x}^{\mathbf{e}})} = \frac{\partial F(\mathbf{h}, \mathbf{x})_i}{\partial h_j} \right)$, and $\mathbf{J}^{\mathrm{inp}}$ is the input Jacobian

$\left(J^{\text{inp}}_{ij}\Big|_{(\mathbf{h}^e,\mathbf{x}^e)} = \frac{\partial F(\mathbf{h},\mathbf{x})_i}{\partial x_j} \right)$ [52]. Note that $\mathbf{J}^{\text{rec}}|_{(0,0)} = \mathbf{W}_h$ and $\mathbf{J}^{\text{inp}}|_{(0,0)} = \mathbf{W}_x$.

Prior literature has looked at the eigenvalues and eigenvectors of the recurrence Jacobian (and recurrence matrix) to investigate how connectivity affects the dynamics of the network [52, 120]. Specifically [52] obtains the stimulus integration timescale τ_i associated with a stable eigenvalue λ_i (i.e. $|\lambda_i| \leq 1$), by looking at the discrete-time iteration $h_i(t) = \lambda_i^t h_i(0)$ that governs the integration of stimulus in the direction of eigenvector v_i associated with λ_i. They then compare this with the equivalent continuous time equation $h_i(t) = h_i(0)e^{-t/\tau_i}$, to get $\tau_i = |(1/\ln|\lambda_i|)|$.

As seen in Figure 5.9, we find that the agent training process reorganizes the eigenspectrum of the RNN recurrence matrix to produce multiple eigenvalues outside of the complex unit circle. These unstable eigenvalues (along with any external stimuli) drive the network's hidden dynamics in the initial steps of a plume-tracking episode. (See animation showing the dynamics of the eigenspectra over the course of an example tracking trajectory: https://github.com/BruntonUWBio/plumetracknets/blob/main/VRNN3-eigen.md) Next, we calculate the eigenvalues and stimulus integration timescales of the zero-input recurrence Jacobian $\mathbf{J}^{\text{rec}}|_{(\mathbf{h},0)}$ along every step of the trajectory for one randomly chosen successful and one unsuccessful episode across each of three plume configurations (constant, switch-once and switch-many). Comparing the time-averaged stimulus integration timescales from these recurrence Jacobians with those from the untrained RNN recurrence matrix reveals that training adjusts these timescales to lie well within the maximum episode length (300 timesteps). Furthermore, we see that the bulk of these timescales are within about 12 timesteps ($\approx$ 0.5s), suggesting that the plume tracking task predominantly needs short timescale memories. See 5.13 for data on all RNNs,

5.5 Discussion

In summary, we use a normative approach using deep reinforcement learning trained artificial neural network (ANN) agents to better understand the behaviors and neural computations that can support plume tracking in simulated turbulent odor plumes. Despite being trained using only a simple task-aligned reward function, we find that several naturalistic

biological features emerge in the agents at the level of behavioral modules, neural representations, and neural dynamics. We discuss these and some empirical observations on RNN connectivity below.

We find that the complex behavior exhibited by ANN agents can be decomposed into sequences of simpler modules (Section 5.4.1). Using simple threshold-based rules on the duration since the agent last encountered the plume, we temporally segment the agent's trajectories into three behavior modules: *tracking*, *lost* and *recovering*. *Tracking* keeps the agent within the plume or close to the edge of the plume while rapidly moving closer to the plume source (Figure 5.3). The agent is *recovering* if it loses odor input for a relatively short time (between 0.5s and $\approx 1.0s$), and makes large cross-wind movements in order to locate the plume. If it does not find the plume in that time, i.e. it is *lost*, it moves into a periodic spiraling or slithering-like oscillating motion. These modules show features similar to *upwind surging*, *crosswind casting* and *U-turn* behaviors previously reported in previous ethological studies on moths, fruit-flies and other flying insects [45, 48–50]. Furthermore, a mechanism for controlling changes in flight behavior, based on an internal clock that is activated when the insect does not sense odor, has also been previously proposed for moths [134–136].

Analysing the behavior of agents when they are navigating plumes with changing wind-direction reveals that the local structure of the plume is more relevant to the agent's flight strategy than the current wind-direction (Section 5.4.2). Specifically, we look at the empirical distributions of agents' course-direction in the *tracking* module to find that agents spend a significant proportion of their time at an angle to the upwind-direction (Figure 5.4). However, when the same empirical course-direction distribution is referenced to plume centerline, we find it centered around the anti-parallel direction to the plume centerline. Plume tracking flying insects have largely been studied in wind-tunnel experiments with constant wind-direction laminar plumes (though note a recent large scale release-and-recapture experiment that attempts to do so at a multi-kilometer spatial scale at low wind speeds [137]). This result provides us with intuition and a testable hypothesis to guide future experimental wind-tunnel studies with non-stationary wind-direction plumes.

Next, we investigate the role of memory capacity towards task completion by training

feedforward networks (MLPs) with varying amounts of historical sensory inputs, and compare these with recurrent (RNN) agents (Section 5.4.3). We find that longer durations of sensory memory are associated with better performance on tougher tracking tasks, such as those having plumes that switch one or more times. Our data suggests that RNNs generally outperform MLPs across plume tracking tasks of different levels of difficulty, with more pronounced differences on tougher tasks. Prior work analyzing 3D flight trajectories of flies and mosquitoes has reported that short-term memory processes modulate the decision sequences executed by these flying insects during natural plume tracking [90].

As is typical in neurobiological recordings [138, 139] and also in spiking neural network models [140], we find that population activity for our RNNs is also low dimensional, with 5-8 principal components capturing the overwhelming majority ($\geq$ 90%) of total variance (Section 5.4.4). We also identify biologically relevant variables that the RNNs appear to represent (Figure 5.6), namely, (1) agent head-direction, (2) time since odor was last encountered, (3) an exponentially moving-average of a discretized recently sensed odors, and (4) an exponentially moving-average of recently sensed odors. Some of these variables (1–4) have been reported for their relevance to odor navigation and related tasks in previous biological literature. The Ellipsoid Body in the Central Complex of flying insects is a well-studied structure implicated in its role in navigation, and has been found to implement a stable head-direction compass by integrating multi-modal neural signals [141–144]. A more recent study in Drosophila revealed a ring attractor circuit that tracks the insect's head-direction, specifically in a wind-guided compass navigation task [145]. [87] makes a case for the importance of neurally encoding time between odor encounters for olfactory search, and provides mechanisms for how bursting olfactory receptor neurons (bORNs) in many animals can encode the time intervals between successive odor encounters. On similar lines, [146] finds that flies navigating in an experimental turbulent environment use an exponentially-weighted moving-average of odor encounters to assign probabilities to their turn and stop behaviors, with saccadic upwind turns becoming more likely at higher encounter rates [147]. Our choice of (d) is motivated by theoretical results in [52] that suggest that exponentially-weighted averages are a good model for stimulus integration in RNNs. Future experiments could confirm if this variable is indeed important to flying-insect navigation decisions, and

furthermore reveal if there are specific subcircuits that encode for this quantity directly. Interestingly, we find that the window-size of the exponentially-weighted moving-average of odor stimulus (e.g. window-size 8 for agent 3) is significantly shorter than that for odor encounters (e.g. window-size 46 for agent 3), which seems plausible for the quick decision making that would be required for turn decisions during flight. The higher permutation importance of the former compared to the latter corroborates this intuition. It is important to note that there could be other variables, that we have not considered but have higher explanatory value, that are highly correlated with the ones that we have studied. To the best of our knowledge, there are no theoretical tools available yet that would allow us to reverse engineer the true mathematical program being implemented by an RNN.

Inspecting the dynamical features associated with each behavior module's neural activity reveals quasi-periodic trajectories associated with the *tracking* behavior module, and quasi-limit-cycles associated with the *lost* behavior module (Section 5.4.5). Additionally, we find that the time taken to transition into the neural activity cluster associated with the *lost* behavior after leaving the plume, takes longer than the time taken to transition into the neural activity cluster associated with the *tracking* behavior after entering the plume. This asymmetry in macro-scale transitions in the neural state resembles an asymmetry in behavior transitions reported in [45], where the authors observe that Drosophila take about twice as long to cast crosswind (after plume loss) than surge upwind (on encountering attractive odors).

We also make report some patterns associated with recurrent connectivity that we consistently observe across our trained RNNs (Figure 5.9). Comparing eigenmodes of the RNN connectivity matrix before and after training reveals that training leads to the emergence of unstable eigenmodes. Additionally, looking at the stimulus integration timescales associated with stable eigenmodes of recurrence Jacobians of trained agents suggests that only a few long timescales (maximum of 56–117 steps across all 5 agents) are involved, whereas the bulk of timescales are within ≈ 12 steps.

Online supplement

Animations accompanying this manuscript can be found at: https://github.com/BruntonUWBio/plumetracknets Code will be released at this location on manuscript publication.

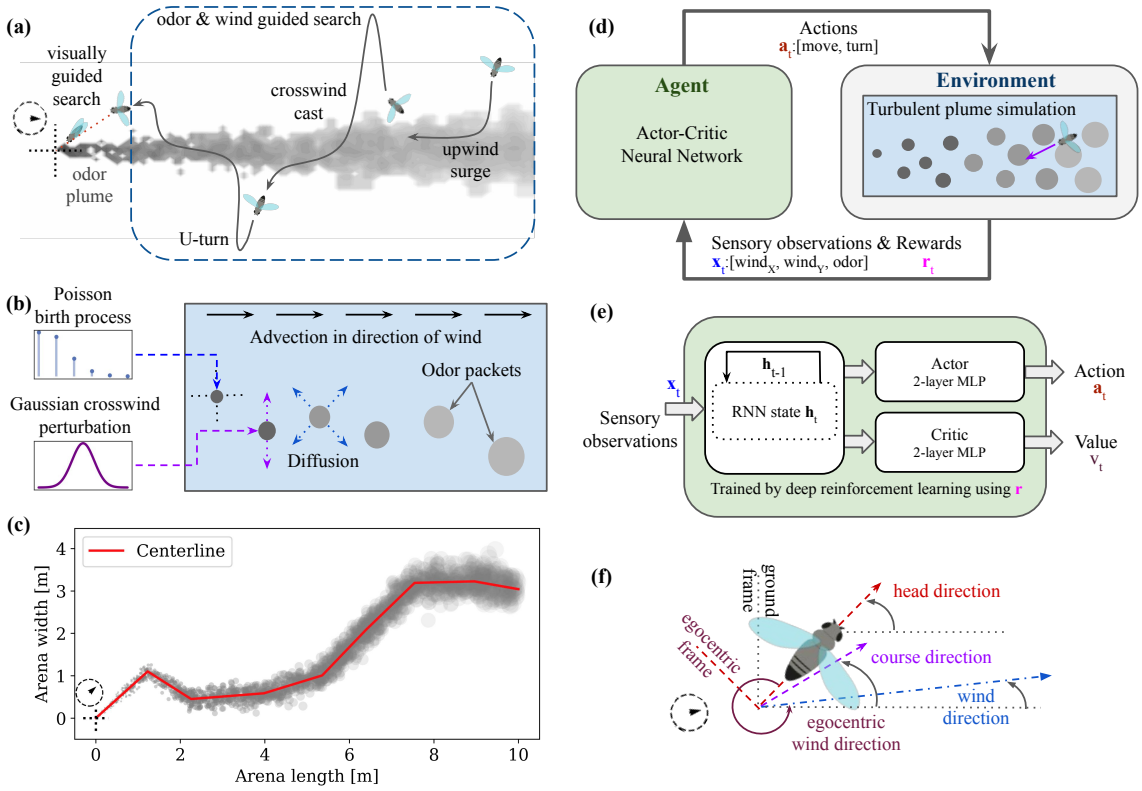

Figure 5.1: **Training artificial agents to track turbulent plumes with deep reinforcement learning. (a)** A schematic of a flying insect performing a plume tracking task, showing *upwind surges*, *crosswind casts*, and *U-turns* behaviors (inspired by a figure in [48]). In this work, we model the spatial scale (dashed rectangle) where the insect can use only olfactory and mechanosensory (to sense wind velocity) cues for plume tracking. **(b)** The plume simulator models stochastic emission of odor packets from a source carried by wind. Odor packets are subject to advection by wind, random cross-wind perturbation, and radial diffusion. **(c)** An example of a plume simulation where the wind direction changed several times. **(d)** A schematic showing how the artificial agent interacts with the environment at each time step. The environment model determines the sensory observations available to the agent $\mathbf{x}$ (egocentric wind direction vector and local odor concentration) and the rewards used in training. The agent navigates within the environment with actions $\mathbf{a}$ (turn direction and magnitude of movement). **(e)** Agents are modeled as neural networks and trained by deep reinforcement learning (DRL). A recurrent neural network (RNN) generates an internal state representation from sensory observations, followed by parallel Actor and Critic heads that implement the agent's control policy and predict the state values, respectively. The Actor and Critic heads are 2-layer, feedforward multi-layer perceptron (MLP) networks. **(f)** A schematic showing a flying agent's head-direction, course-direction, and the wind direction, all measured with respect to the ground and counter-clockwise from the x-axis. Course direction is the direction that the agent actually moves in, accounting for the effect of the wind on the agent's intended direction of movement (head-direction). Egocentric wind direction is the direction of the wind as sensed by the agent and is measured counter-clockwise with respect to the agent's current head-direction.

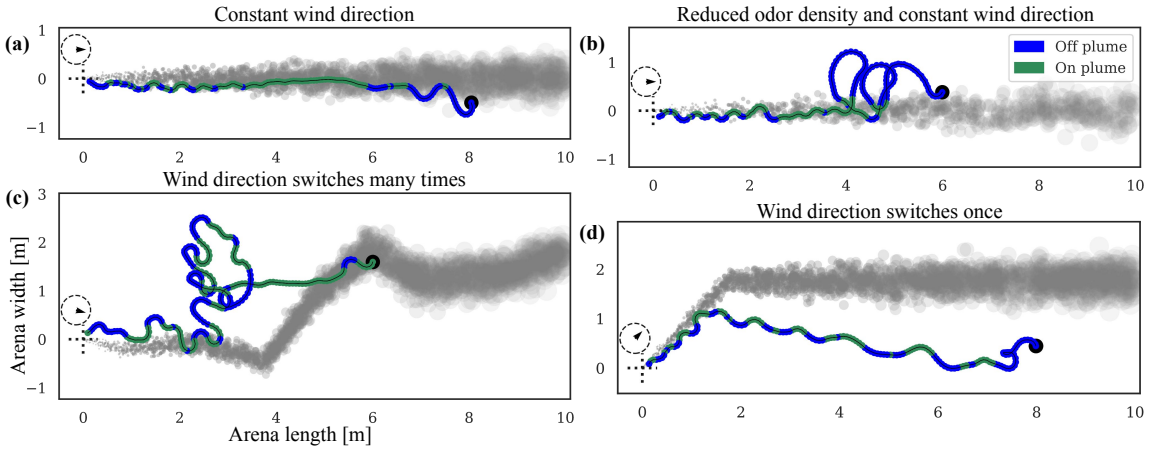

Figure 5.2: **Successful plume tracking trajectories under various plume simulator configurations**: Odor plumes (dark grey), with RNN (RNN) agent trajectories colored by whether the agent was able to sense the presence (green) or absence (dark blue) of odor. Trajectories start at filled black circle and end near dotted cross-hairs in the (lower) left of each sub-figure. Plotted plume snapshot is from end of tracking episode, and plotted trajectory covers entire duration of the tracking episode. All examples use a 0.5 m/s wind. The arrow within dotted circle above the cross-hair shows direction of wind at time of snapshot. The different configurations are: **(a)** 'Constant' left-to-right wind-direction plume. **(b)** 'Sparse' plume with constant same left-to-right wind-direction but reduced (0.4x) birthrate. **(c)** 'switch-many' plume with switches occurring every $\approx 3s$ **(d)** 'switch-once' plume, which makes one 45° counter-clockwise switch during the tracking episode. Episodes are manually chosen across 3 trained agents. Supplementary animations provide additional examples of both successful and unsuccessful tracking episodes: https://github.com/BruntonUWBio/plumetracknets.

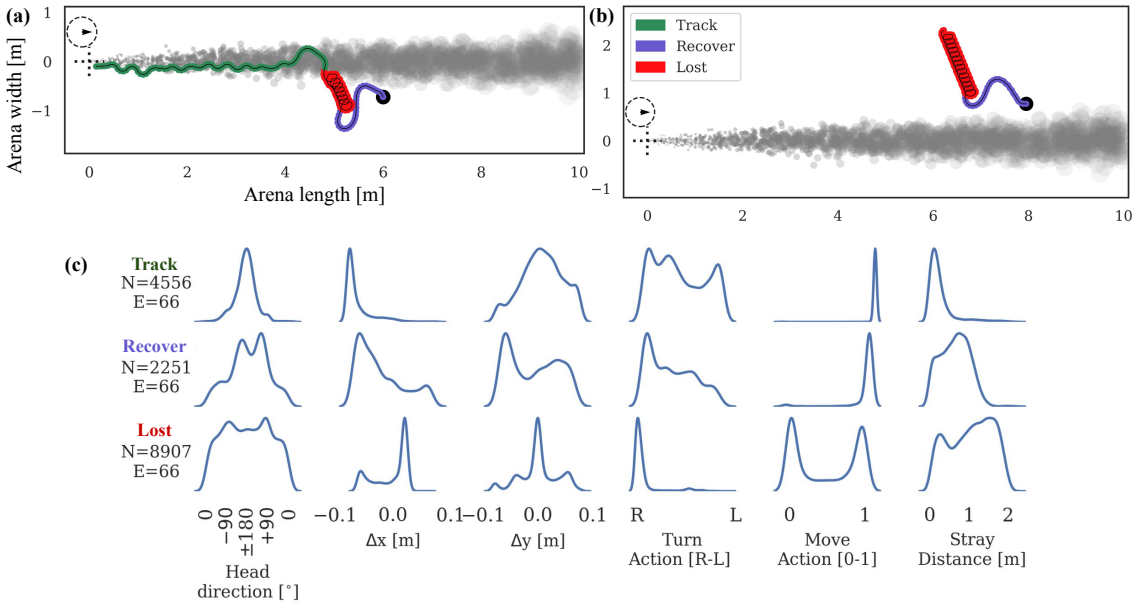

Figure 5.3: **Plume tracking trajectories can be decomposed into behavior modules: (a & b)** Trajectories for successful **(a)** and unsuccessful **(b)** plume tracking episodes showing three distinct behavior modules: tracking (green), *lost* (red) and recovering (blue) **(c)** Histograms show data aggregated from an equal number of successful and unsuccessful constant wind-direction plume tracking episodes (N timesteps, E episodes). Plots reveal differences between three behavior modules across key behavioral measures: Head-direction: Histograms of head-direction (i.e. the orientation of the agent with respect to the ground) are concentrated around $\pm 180°$, a signature of zig-zagging but mostly upwind trajectories; the histogram concentration around the upwind-direction reduces from tracking to recover to lost, accounting for the more complex trajectories encountered in the latter behavior modules. (angle measured counterclockwise with $0°$ indicating directly downwind.) Δx and Δy: Histograms of per-timestep drift in the x-direction (Δx) and y-direction (Δy) show how tracking is characterized by primarily upwind movement in both *tracking* and *recover* modules, but lesser so in the *lost* module. Y-direction drifts are significant in the *recover* and *lost* modules, but minimal in the *tracking* module. Turn action: Left/right turning movements are balanced in the *tracking* module as the agent closely tracks the edge of the plume, but is biased towards clockwise movements in the other regimes, especially the *lost* module. Move action: The agent seems to modulate its forward movement speed in the *lost* module only. Stray Distance: The agent strays from the plume minimally in the *tracking* module, but significantly otherwise. See supplemental section 5.8 for equivalent plots for other agents.

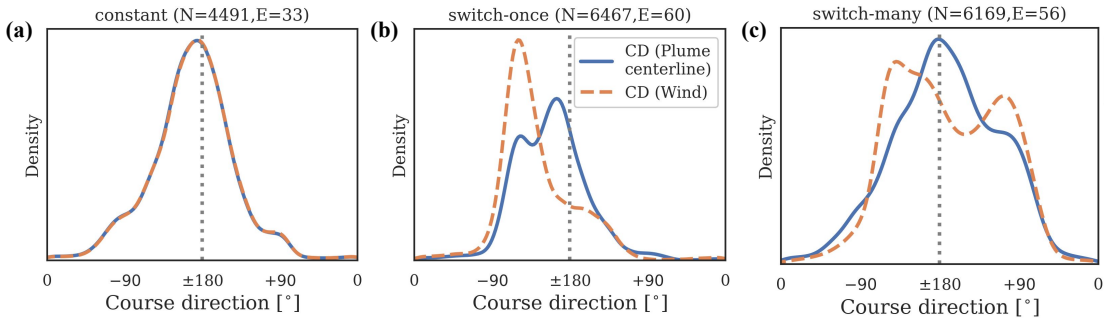

Figure 5.4: **Course-direction empirical distributions suggest that agents track plume with respect to plume centerline rather than current wind-direction:** Solid lines show kernel density estimates of course-direction (CD), and vertical dotted lines indicate distribution peaks (modes). Figure subtitles indicate how many timesteps (N) and how many successful episodes (E) were used for each plot. All angles are measured counter-clockwise; $\pm 180°$ implies anti-parallel movement with respect to the plume centerline, exactly upwind movement with respect to the wind-direction, or right-to-left movement with respect to the ground. **(a)** For the constant wind-direction plume, the three CD distributions are completely aligned. **(b)** In the 'switch-once' regime, a significant proportion of time is spent at a $\approx 45°$ angle to the wind, but is actually aligned (anti-parallel) with the plume centerline. **(c)** We see the same trend again in the 'switch-many' regime, where a significant proportion of time is spent at an angle to the upwind-direction, but is actually aligned with the plume centerline. See 5.9 for equivalent plots for all 5 RNN agents.

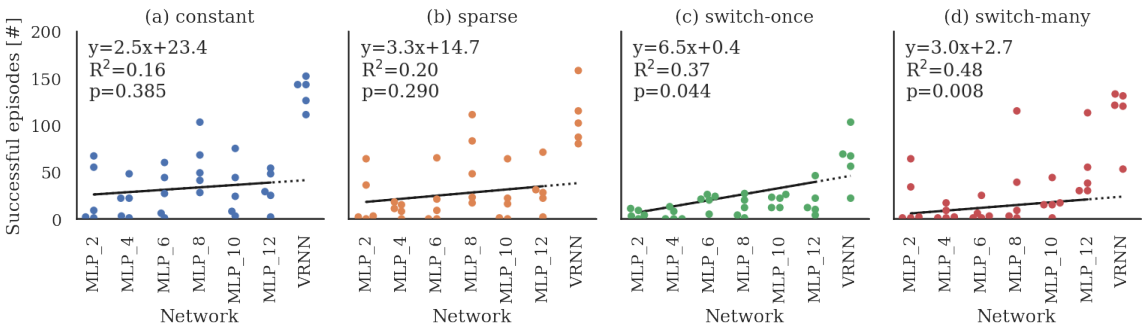

Figure 5.5: **Larger memory capacity improves plume tracking, especially in non-stationary wind-direction plumes:** Number of successful homing episodes for different agent architectures, across different plume configurations for the same set of initial conditions: agent starting location and orientation, and plume state (Section 5.3.3). 'MLP_X' refers to feedforward networks with X timesteps of sensory history. RNNs generally outperform feedforward networks, with differences appearing to become more pronounced for more complex switching wind-direction (switch-once, switch-many) plume tasks. In feedforward networks, performance on plumes with switching wind-direction (switch-once, switch-many) can improve significantly with increasing memory. Regression lines (solid black) are fit on only MLP data, but are extended slightly (dotted line) for comparison with RNN (p-values are for a two-sided Wald Test with null hypothesis that the slope is zero.)

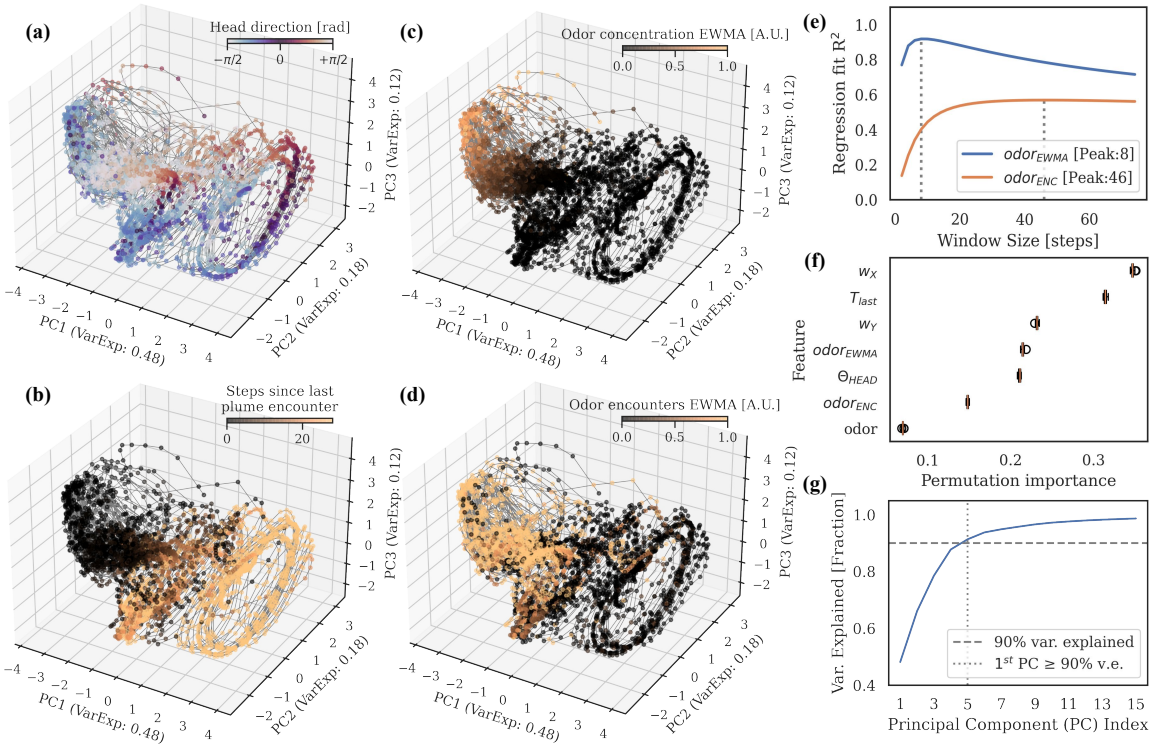

Figure 5.6: **Neural activity is low-dimensional and represents biologically relevant variables.** (a–d) Neural activity trajectories plotted over a diversity of plume conditions and tracking outcomes, **(a)** colored by agent head-direction Θ_{HEAD}, **(b)** steps since last odor encounter T_{last}, **(c)** exponentially-weighted moving-average of odor concentration ($odor_{EWMA}$, window-size = 8 steps), and **(d)** exponentially-weighted moving-average of recent odor encounters ($odor_{ENC}$, window-size = 46 steps). **(e)** Quality of fit (R^2) of a linear model regressing neural activity onto $odor_{EWMA}$ and $odor_{ENC}$ for sliding-windows of varying lengths. The sliding-window size for subfigures **(c)** and **(d)** are determined by identifying the peaks of these curves. **(e)** Permutation importance scores of features of a classifier trained to predict agent actions using the aforementioned plotted features (T_{last}, Θ_{HEAD}, $odor_{EWMA}$, and $odor_{ENC}$), and instantaneous sensory observations (wind w_X, w_Y and odor). **(f)** Plot of cumulative variance explained by top principal components of neural activity aggregated across multiple plume configurations ('constant', 'switch-once' & 'switch-many') suggests a low-dimensional structure. **(g)** 90% of the variance of the 64-dimensional neural activity can be explained by the first-5 principal components. See **??** for equivalent plots for all 5 RNN agents.

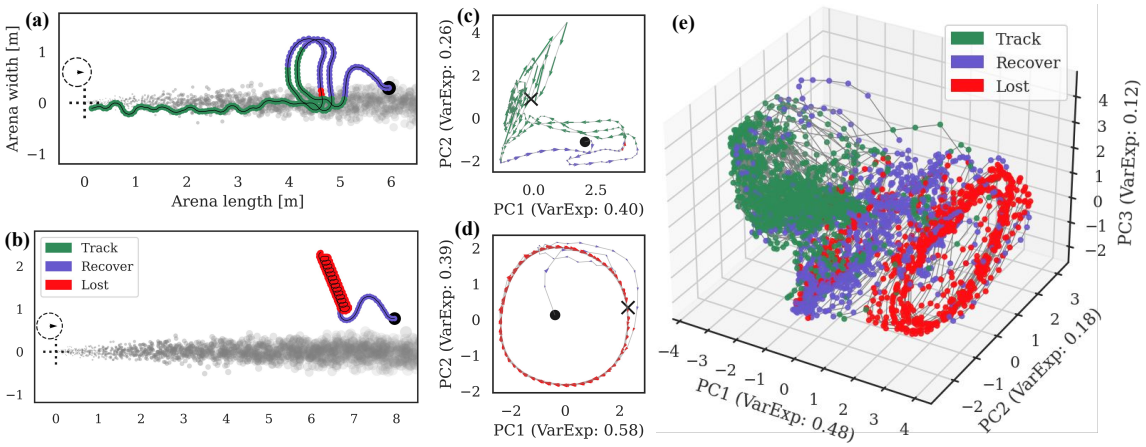

Figure 5.7: **Neural dynamics appear to organize themselves into overlapping yet distinct regimes: (a)** Episode that ends in successful homing-in on the odor source, and **(b)** Unsuccessful episode that strays from the plume and ends up exceeding the simulator's bounds. **(c & d)** Neural activity plots corresponding to each row's trajectory projected on a 2D subspace generated from the top-2 principal components of that episode's neural activity. Quiver arrows correspond to direction of neural activity gradient, and are colored by the agent's current behavior module. **(c)** A 'funnel' like structure (in green) emerges in the state-space corresponding to the *tracking* neural activity regime. **(d)** The agent's periodic 'lost' behavior shows up as a limit-cycle in the state-space (red). **(e)** Neural activity plotted over multiple trajectories comprising a diversity of plume conditions and tracking outcomes, colored by behavior module (section 5.4.1). See supplemental section 5.11 for plots for all 5 RNN agents.

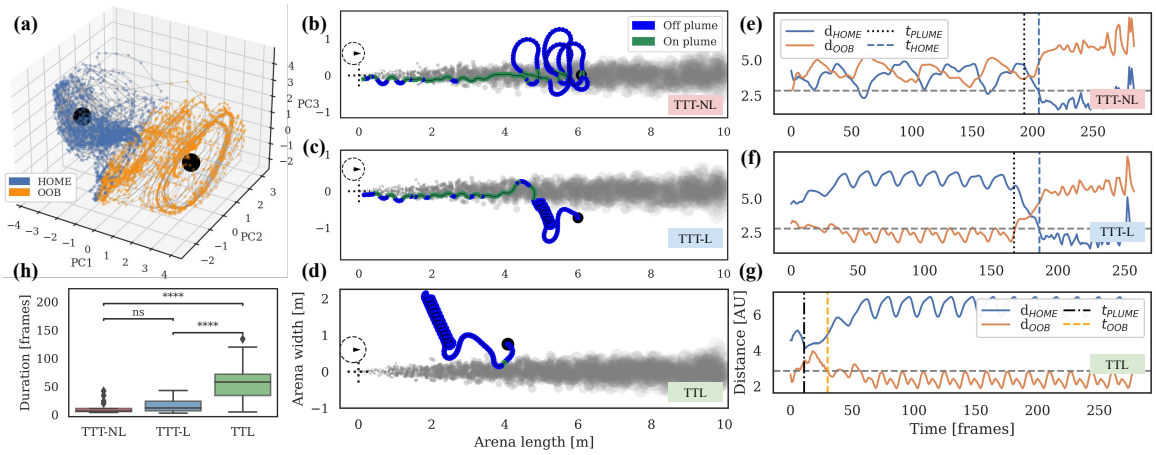

Figure 5.8: **Transitions between neural activity regimes are asymmetric in duration: (a)** Neural activity from multiple tracking episodes plotted on same top-3 principal component subspace as in Figure 5.7. Points are colored by which neural regime centroid they are closest to (see Section 5.12 for centroid definitions; orange color for points nearer to Out-Of-Bounds/OOB centroid and blue for those closer to HOME centroid; black circles denote centroid locations). (b–d) Example trajectories where the agent **(b)** enters the plume from the *recovering* behavior module, **(c)** enters the plume from the *lost* behavior module, and **(d)** enters the *lost* behavior module after a brief encounter with the plume. (e–g) Time courses of neural activity distances to the HOME and OOB centroid (in blue and orange respectively), associated with the respective trajectories in the center column. Dotted vertical lines show time of entering or leaving plume, while dashed vertical lines show time when the agent has entered the target neural activity regime, i.e. neural activity is less than D/2 units away from target centroid, where D is the distance between centroids. **(h)** Box plots compare transition times to target regimes over a large set of trajectories across varying plume conditions (constant, switch-once, switch-many). Transitions into the *lost* neural activity regime (TTL) tend to take longer than transitions into the *tracking* neural activity regime (TTT-NL or TTT-L) (two-sided Mann-Whitney-Wilcoxon test with Bonferroni correction, ****: $p \leq$ 1.00e-04, ns: not significant). See 5.12 for plots of all 5 RNNs.

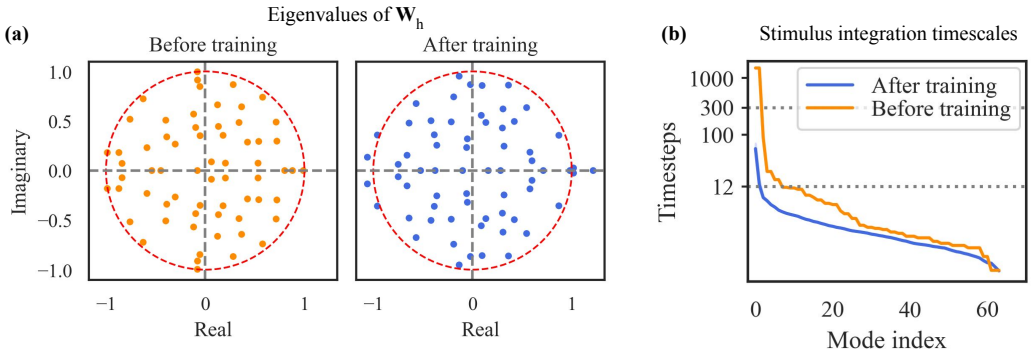

Figure 5.9: **Eigenvalues and stimulus integration timescales: (a)** Eigenvalue spectra of the RNN recurrence matrix $\mathbf{W_h}$ before and after training show how training results in the generation of unstable modes. **(b)** Time-averaged (over 6 episodes) stimulus integration timescales associated with stable eigenmodes of recurrence Jacobian $\mathbf{J}^{\text{rec}}$ show a bulk of relatively short timescales (within 12 timesteps). Top 5 integration timescales for agent shown are 56.5, 13.0, 7.7, 6.8 & 5.8 timesteps. Before training, timescales associated with $\mathbf{W_h}$'s eigenmodes can be large, even exceeding the length of the training/evaluation episodes (300 steps). See 5.13 for equivalent plots and data for all RNN agents.

5.6 Supplementary details on agent training and evaluation

[We repeat some details from the main text for the sake of readability.]

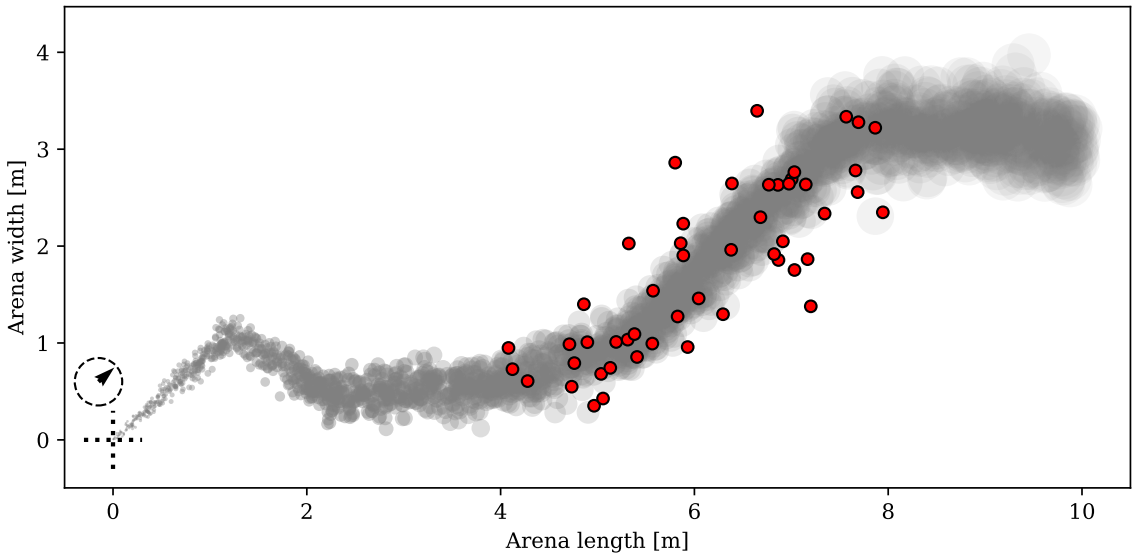

Figure 5.10: **Snapshot of training plume:** 50 points (red) are potential starting locations. Agent is initialized with a uniformly randomly chosen head-direction.

5.6.1 Plume pre-computation:

Puffs are generated at the source located at $(0, 0)$, at the rate of $r_t \sim Poisson(R)$ puffs/step, where $R = 1.0$. Each puff's location $p_t = (x_t, y_t)$ is henceforth governed by the stochastic differential equation, $p_t = p_{t-1} + w_t\delta + \xi$, where w_t is the wind-velocity at time t, $\xi \sim \mathcal{N}(0, \sigma)$ is cross wind i.i.d. random Gaussian noise added per the turbulent plume model of [43]. Each puff trajectory is integrated using a simple forward Euler integrator at 100 frames/sec. Furthermore, each puff starts with a radius $r_0 = 0.01$ m and undergoes a diffusion process that increases it's radius at the rate of 0.01 m/s.

We compute 120-second long (clock time) plumes ahead of training/evaluation time for the 'constant', 'switch-once' and 'switch-many' configurations. A 40s window (60s - 100s) of the 'switch-many' and 'constant' plumes are used for training agents (Figure 5.10). 'Sparse'

plumes are simulated by downsampling the number of puffs simulated in a 'constant' plume simulation.

Wind velocity w_t is held constant at $(0.5, 0.0)$ m/s for all t for the 'constant' plume configuration. For the 'switch-once' plume configuration, wind velocity w_t starts at $(0.5, 0.0)$ m/s till $t = 60.00$ s, when it makes a single 45° counter-clockwise turn and stays there for the rest of the simulation. For the 'switch-many' plume configuration, wind velocity w_t changes once every $3.0 + \tau$ seconds, where $\tau \sim Uniform(-0.3, +0.3)$ is a random shift added i.i.d. at each change. Wind-direction turns are sampled i.i.d. from a $\mathcal{N}(0°, 30°)$ Gaussian distribution truncated at ±60°.

5.6.2 Training

Partially Observable Markov Decision Process (POMDP): To train agents using DRL, we define a Partially Observable Markov Decision Process (POMDP) [100] as follows (also see Figure 5.1):

- *Action space:* Agents provide a two dimensional output a_t at each timestep corresponding to how much they want to turn and how much they want to move forward.

$$\mathbf{a_t} = [a_\theta, a_m], \text{where } a_\theta \in [-\theta_{max}, +\theta_{max}], a_m \in [0, \Delta_{max}]$$

  The maximum turn capacity of an agent (θ_{max}) is 6.25π radians/s (1125 °/s), and the maximum forward movement capacity (Δ_{max}) of an agent is 2.5 m/s.

- *Observation space:* Agents receive a 3-dimensional egocentric sensory observation vector o_t at timestep t, comprising odor-concentration and (x, y) coordinates of relative wind-velocity at the agent's current location and orientation in the plume. Note that the agent's current location and orientation in the plume are tracked and updated by the training environment code.

$$\mathbf{o_t} = [o_c, o_x, o_y], \text{where } o_c \in [c_{min}, 1], o_x, o_y \in [-(\Delta_{max} + |v_{wind}|), (\Delta_{max} + |v_{wind}|)]$$

Here c_{min} is the minimum odor concentration, and has been manually set to be 0.0001 arbitrarily units.

- *Reward function::* Rewards are given to encourage task completion, i.e. home in on the plume source. The agent receives: $+100$ when it reaches within a small fixed radius $r_{homed} = 0.2m$ of the source, $-\epsilon$ per timestep to simulate a 'metabolic cost' to flying and therefore encourage faster homing. We also provide the agent two shaping rewards, without which the training process is infeasibly slow: First, a reward proportional to the decrease in radial distance to the source per timestep $(r_{t-1} - r_t)$ as a form of shaping reward. Here, $r_t = \sqrt{x_t^2 + y_t^2}$ is the euclidean distance of the agent to the source at timestep t. Second, a fixed negative reward of -10 if the agent strays more than $r_{stray} = 2$m away from the plume (i.e. the center of the nearest puff is greater than r_{stray}).

- *Transition function:* The agent's location and orientation within the plume is randomly initialized at the beginning of each training episode (see Figure 5.10 for example locations). The environment then deterministically updates the agent's location and orientation at each timestep taking into account its actions and the wind velocity. Episodes end if the agent reaches within a radial distance r_{homed} of odor source, or if the agent strays more than r_{stray} from the plume, or if the episode exceeds 300 timesteps (12 seconds of clock time.)

- *Augmented observation space for MLPs:* To understand the role of memory on tracking performance, in section 5.4.3, we use feedforward-only networks (MLPs) with fixed-length memory. Memory is simulated by appending historical sensory observations into the MLPs' inputs (known as 'frame stacking' in the DRL literature [126].) Therefore $\mathbf{o_t}$ for an MLP with L timesteps history is now $[o_c^{(0)}, o_x^{(0)}, o_y^{(0)}, \ldots o_c^{(L)}, o_x^{(L)}, o_y^{(L)}]$.

We implement the POMDP environment using the OpenAI Gym [148] and stable-baselines [149] libraries.

Training curricula: We adapt an open source implementation [150] of the Proximal Policy Gradient algorithm with Generalized Advantage Estimation (PPO-GAE) [47, 151] to train our agents.

To train our agents to perform across dynamically varying plumes, we randomize the agent's location, agent's orientation, plume state and plume sparsity at the start of each training episode. Agents are initialized at random starting locations (x, y), where x is chosen uniformly randomly in the range $[30, 80]$ percentile of puff locations; y is chosen by sampling from a normal distribution with mean given by the median y-coordinate of odor puffs in the range $[x-1, x+1]$, and variance given by the $5^{th} - 50^{th}$ percentile y-coordinate difference of the aforementioned odor puffs. Initial agent orientation is selected at random from $[-\pi, \pi]$ radians. The 'switch-many' plume, which changes direction every ≈ 3 seconds, is used for training. Initial plume state is randomized by choosing a random time between 60s - 90s, at which to initialize the precomputed plume. The simulation is sparsified by downsampling the number of puffs to a fraction randomly uniformly chosen in the range $[0.3, 1.0]$. The plume is randomly flipped about the x-axis to mitigate any y-directional biases that might have crept into the finite plume simulation.

Curriculum based training methods are known to improve training performance by gradually increasing the difficulty of the training task over the course of the training process [152]. We train our RNNs using a two stage curriculum, where we first train the RNN for 1 million timesteps on the constant wind-direction plume, and then train it for another 4 million timesteps on the switch-many plume. This two stage process improved the stability and performance of the training process for RNNs, but not for MLPs. MLPs are directly trained for 2 million timesteps on the switch-many plume. Training durations have been chosen such that training updates reliably converge within these times.

Hyperparameter selection: Our training process has hyperparameters relating to (1) training algorithm hyperparameters, (2) training plume parameters, and (3) neural network architecture. (See section 5.7 for a list of all [hyper]parameters.) While PPO is not a very sample efficient algorithm, it is known to work robustly across a wide range of continuous control (continuous observation and action space) problems without needing extensive hyperparameter tuning [47]. Furthermore, exhaustive hyperparameter tuning is computa-

tionally unfeasible on our budget. However, we do try to tweak hyperparameters one-by-one starting off from the parameters suggested in the PPO manuscript for continuous control problems. We also trained Gated Recurrent Units (GRUs) in the same manner as we did our Vanilla RNNs (RNNs), and found that the performance of the GRUs did not significantly exceed that of the RNNs (see Figure 5.11).

5.6.3 Evaluation

We evaluate trained agents over a behavioral assay comprising fixed set of initial locations, initial simulation timestamps and initial agent directions across the aforementioned plume wind-direction and birth-rate configurations, each comprising 240 episodes.

The same set of 240 initial conditions for each episode are used to initialize the agent and simulator, for each agent and dataset evaluated:

- Initial agent head angle (with respect to ground): $0, \frac{1}{4}\pi, \frac{1}{2}\pi, \frac{3}{4}\pi, \pi, \frac{5}{4}\pi, \frac{3}{2}\pi, \frac{7}{4}\pi$ radians
- Initial x-coordinate: 4, 6, and 8 meters
- Initial y-coordinate: 0^{th}, 25^{th}, 50^{th}, 75^{th}, and 100^{th} percentile of the minimum and maximum y-coordinate of the puffs located in a 1-meter band around the initial x-coordinate. For constant wind-direction plumes (including sparse plumes), the task is made harder by selecting only 0^{th}, 50^{th}, and 100^{th} percentiles as described before (i.e. $y_{min}, y_{median}, y_{max}$) and then adding two other locations that are ± 0.5 m outside the plume (i.e. $y_{min} - 0.5$ m and $y_{max} + 0.5$ m)]
- Initial timestamp: 60.00s and 61.00s (58.00s and 59.00s for the switch-once plume as it switches at exactly 60.00s)

We train 14 seeds per model type (RNNs, and MLPs with 2, 4, 6 ..., 12 timesteps of history) and select the top-5 best performing seeds for analysis. Performance here is measured by counting the number of successful episodes across constant, switch-once and switch-many plumes.

All models are trained and evaluated on an Ubuntu Linux v20.04 workstation with Intel Core i9-9940X CPU and a TITAN RTX GPU. Each seed takes ≈ 16 hours to train and evaluate, with MLP and RNN models using 1 and 4 cores in parallel respectively.

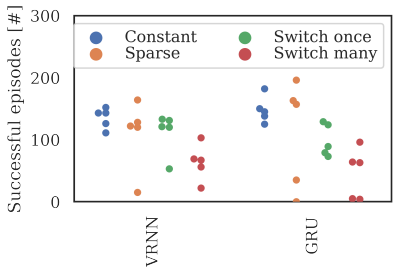

Figure 5.11: Comparison of Vanilla RNNs and GRUs across 4 plume configurations. Vanilla RNN data is same as that in Figure 5.5

.

5.7 Key parameters for simulation, agent, model, and training and evaluation

Parameter description	**Value/Range**
Simulation integration time-step	0.01s
Wind speed	0.5 m/s
Wind speed crosswind noise	$\mathcal{N}(0, 0.005)$ m/s (per timestep)
Puff birth rate (Poisson mean)	1.0 puffs/timestep (at 100 FPS)
Puff initial radius	0.01m
Puff radius growth rate (diffusion)	0.01m/s
Maximum plume extent simulated (x, y)	(-2/+10m, ± 5m)

Table 5.1: Plume [hyper]parameters

Parameter description	**Value/Range**
Environment frame rate	25 FPS
Sensor sampling rate	25 Hz
Forward movement capacity (Δ_{max})	2.5 m/s
Turn capacity (θ_{max})	± 6.25 π radians/s (± 1125 °/sec)
Homing radius	0.2 m
Max. stray from plume allowed	2 m
Odor sensing threshold (minimum)	0.0001 (unitless)

Table 5.2: Agent and environment parameters

Parameter description	**Value/Range**
RNN hidden layer width	64 units
Feedforward hidden layer width(s)	64 units
Neural network nonlinearity	tanh
Layer initialization (Recurrent, Feedforward)	(Normal, Orthogonal)

Table 5.3: Model (neural network) parameters

Parameter description	**Value/Range**
RNN training steps	5M
MLP training steps	2M
Learning Rate	0.0003 (with linear decay)
PPO Entropy Coefficient	0.05
PPO Value Loss Coefficient	0.5
PPO Epochs	10
PPO Gamma	0.99
PPO max. gradient norm	0.5
GAE Lambda	0.95
GAE steps	2048

Table 5.4: Training algorithm, training curriculum and evaluation parameters

5.8 *Regime metadata distributions*

Agent	Agent ID	Lost regime threshold
RNN 1	2760377	30 steps (1.2 s)
RNN 2	3199993	25 steps (1.0 s)
RNN 3	3307e9	35 steps (1.4 s)
RNN 4	541058	38 steps (1.52 s)
RNN 5	9781ba	25 steps (1.0 s)

Table 5.5: Thresholds for defining when the *lost* behavior module kicks in i.e. duration (in timesteps or seconds) since the plume was last encountered.

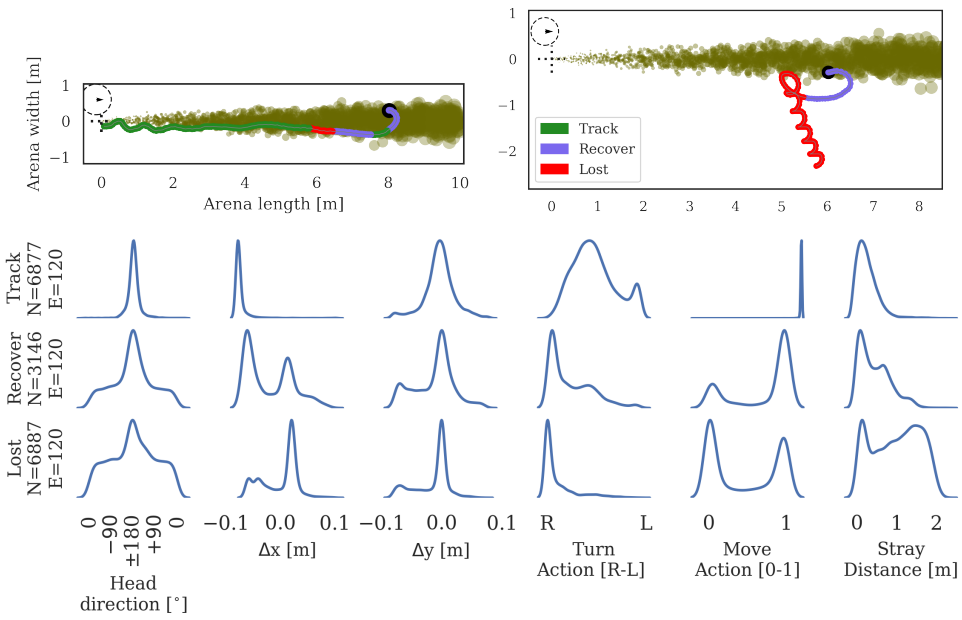

Figure 5.12: Agent 1 (compare with Agent 3 in Figure 5.3)

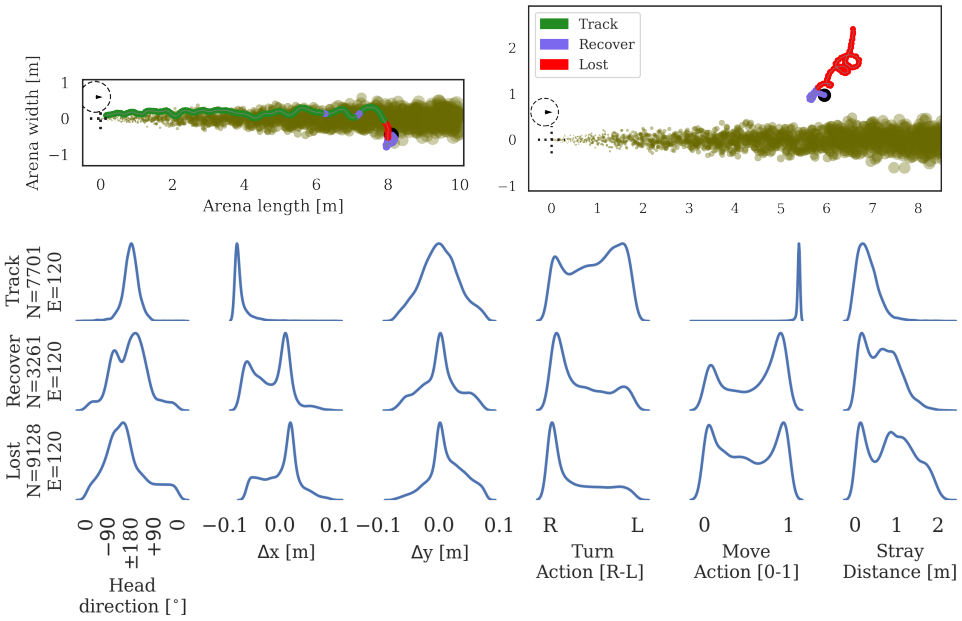

Figure 5.13: Agent 2

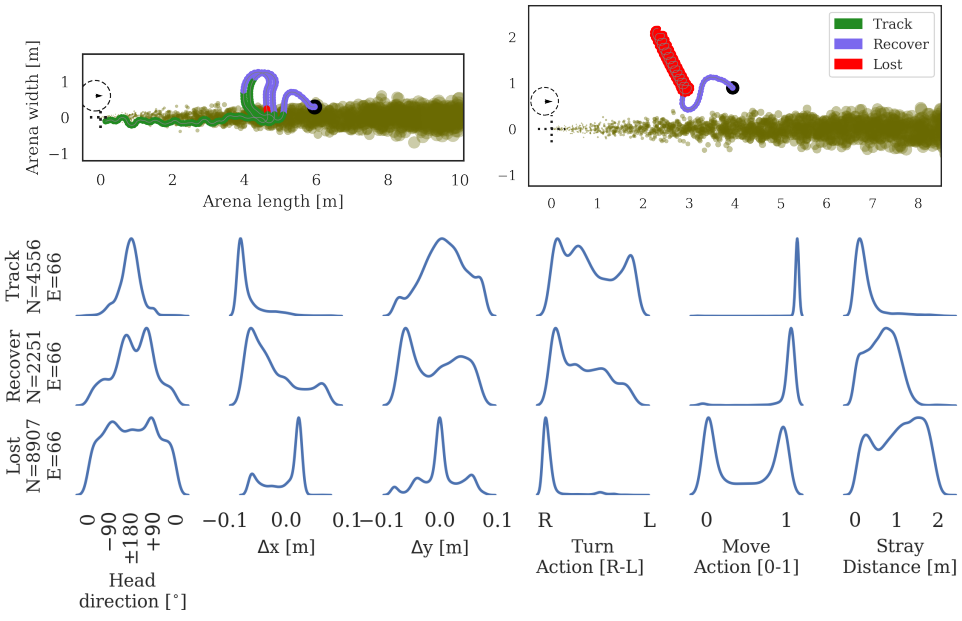

Figure 5.14: Agent 3 (same as Figure 5.3)

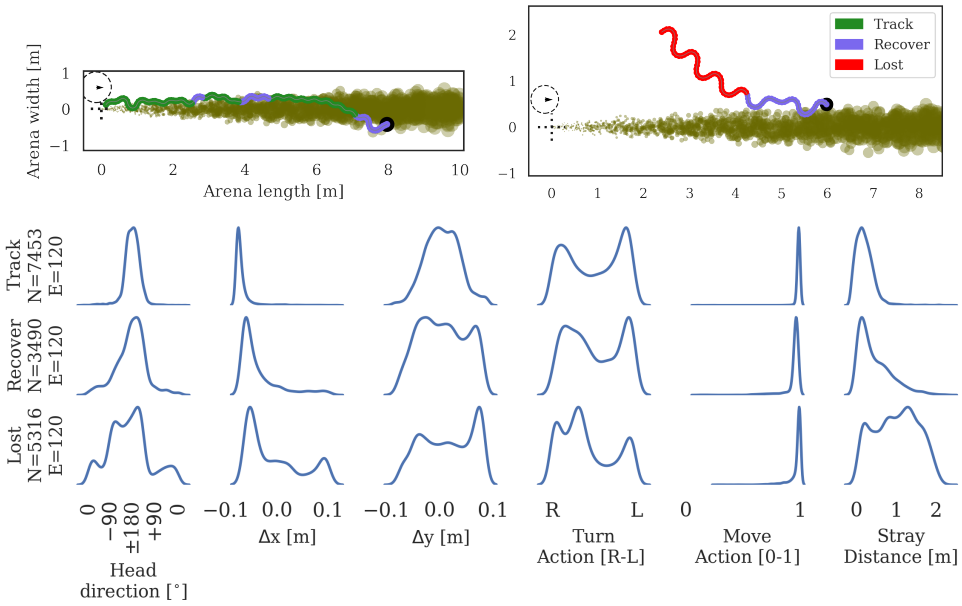

Figure 5.15: Agent 4

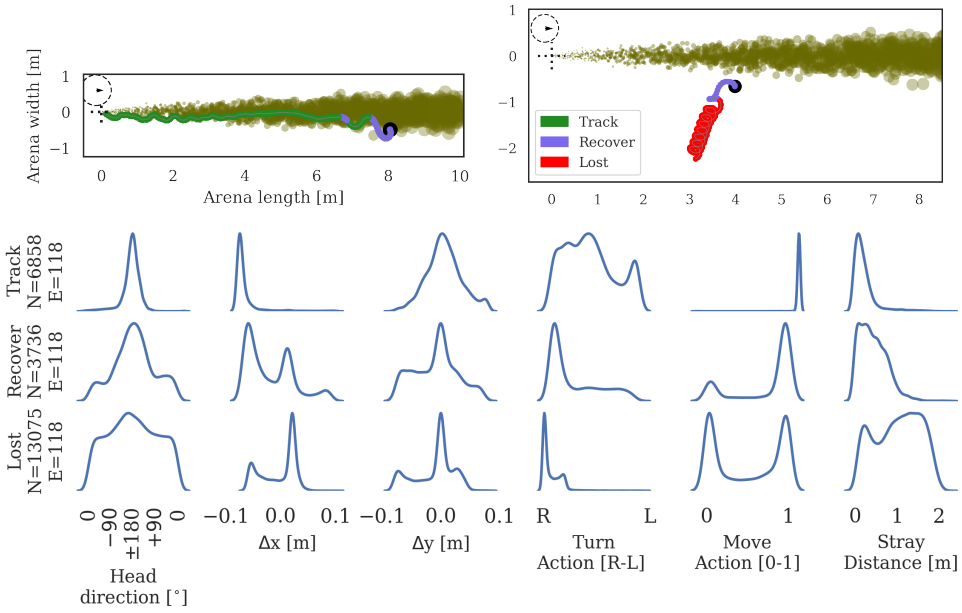

Figure 5.16: Agent 5

5.9 *Comparing reference frames for plume tracking*

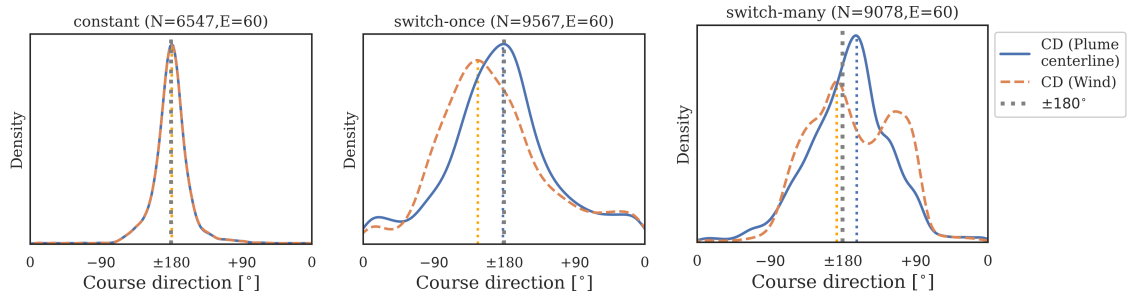

Figure 5.17: Agent 1: Empirical course-direction (CD) distribution

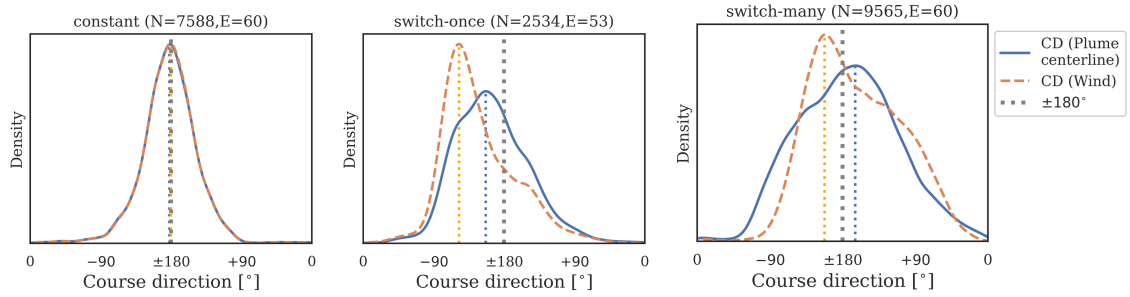

Figure 5.18: Agent 2: Empirical course-direction (CD) distribution

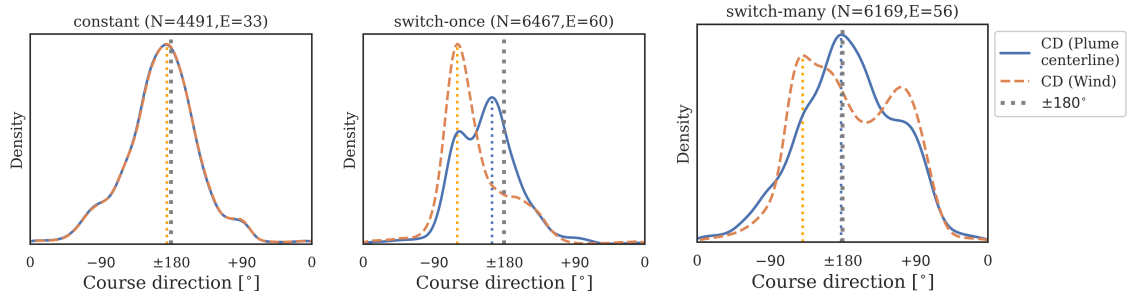

Figure 5.19: Agent 3: Empirical course-direction (CD) distribution (Same as Figure 5.4)

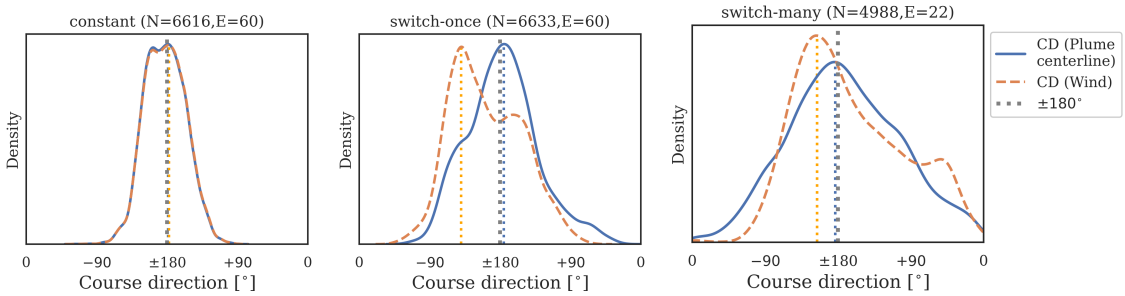

Figure 5.20: Agent 4: Empirical course-direction (CD) distribution

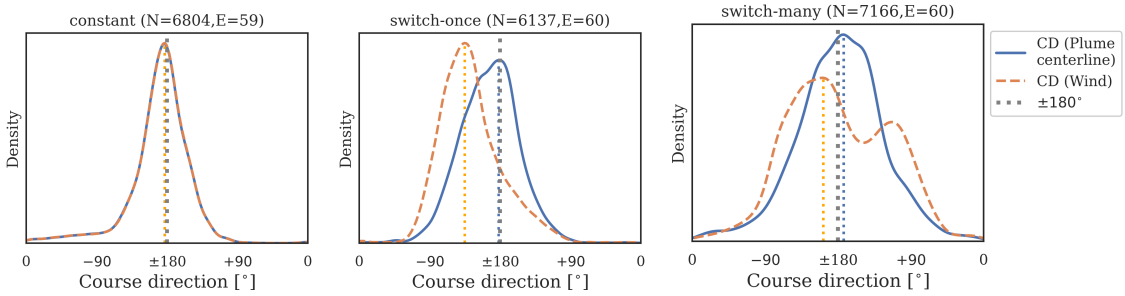

Figure 5.21: Agent 5: Empirical course-direction (CD) distribution

5.10 Neural activity dimensionality and neural representations

Agent	Agent ID	Test set accuracy (All features)	Test set accuracy (Instantaneous only)	Test set accuracy (Most freq. class)
RNN 1	2760377	0.84	0.74	0.33
RNN 2	3199993	0.67	0.49	0.28
RNN 3	3307e9	0.82	0.69	0.39
RNN 4	541058	0.70	0.53	0.44
RNN 5	9781ba	0.84	0.74	0.40

Table 5.6: Classifier based quantification of contribution of represented features: Represented features contribute to higher test accuracy.

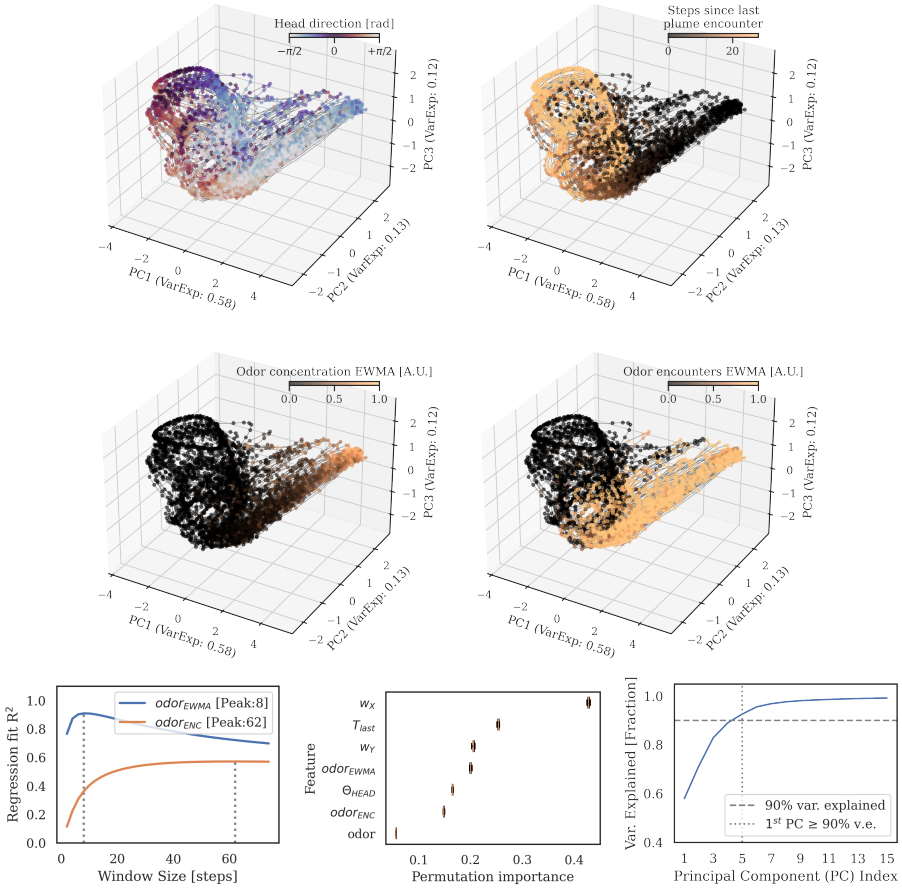

Figure 5.22: Neural representations – Agent 1 (compare with Agent 3 in Figure 5.6)

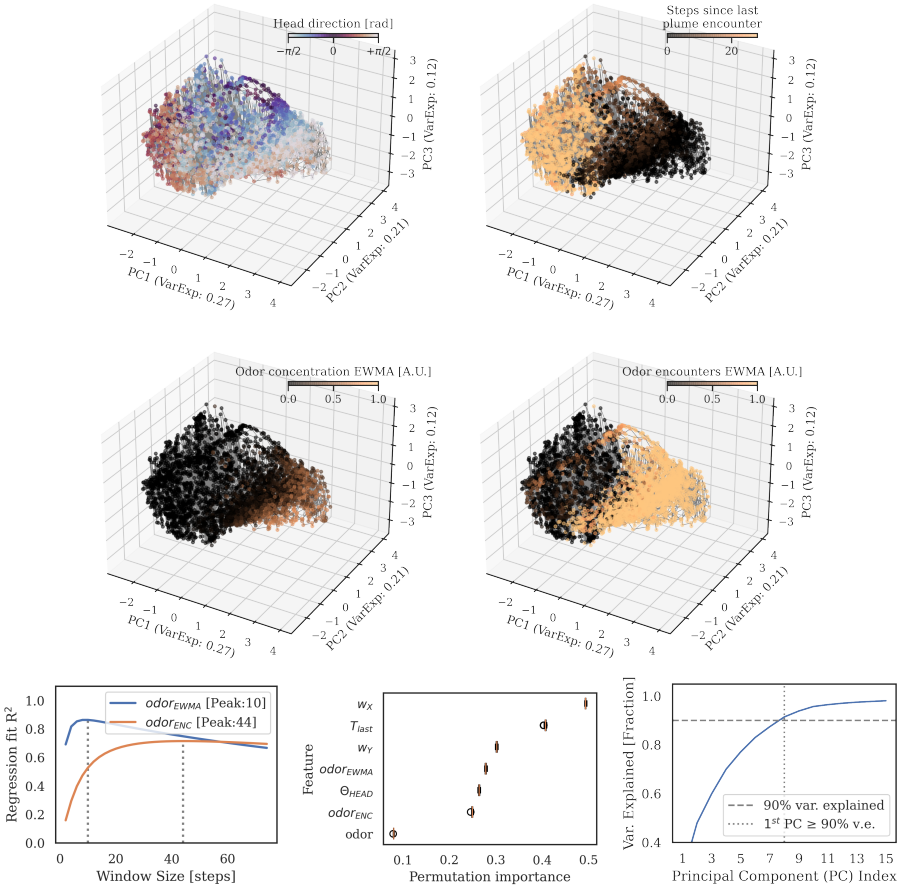

Figure 5.23: Neural representations – Agent 2

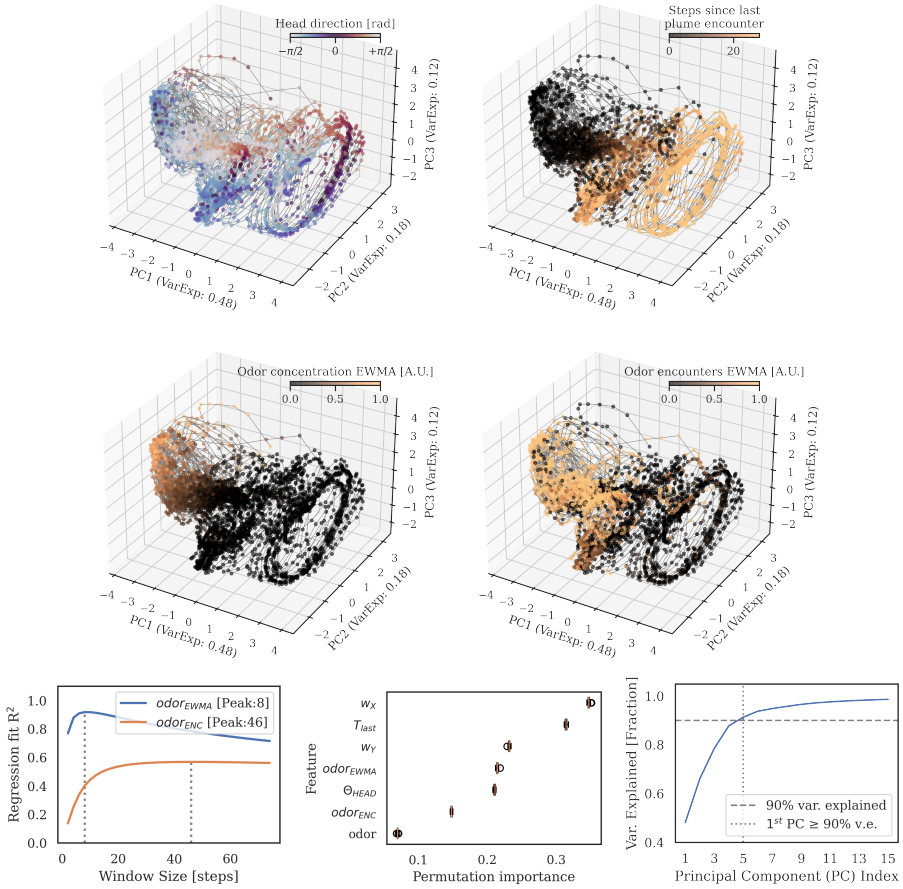

Figure 5.24: Neural representations – Agent 3 (Same agent as in Figure 5.6)

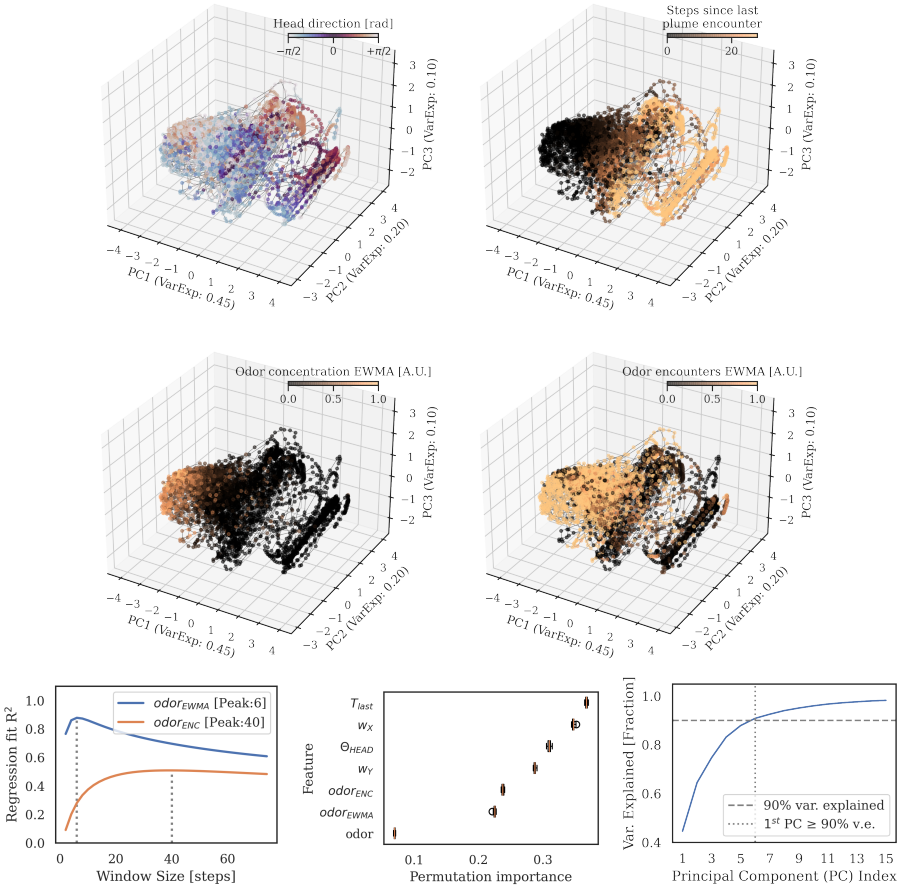

Figure 5.25: Neural representations – Agent 4

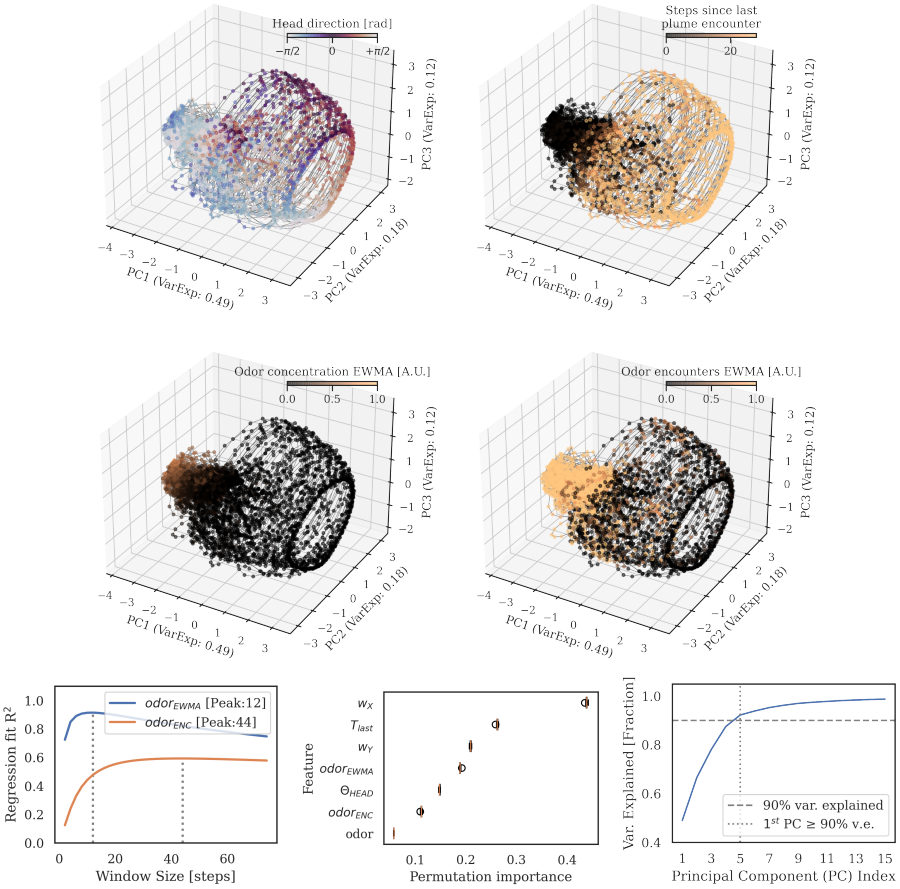

Figure 5.26: Neural representations – Agent 5

5.11 Neural activity regimes

Agent	Agent ID	Limit-cycle period
RNN 1	2760377	19 steps (0.76 s)
RNN 2	3199993	NA (clear periodic structure not observed)
RNN 3	3307e9	17 steps (0.68 s)
RNN 4	541058	28 steps (1.12 s)
RNN 5	9781ba	18 steps (0.72 s)

Table 5.7: Limit cycle periods for each RNN agent

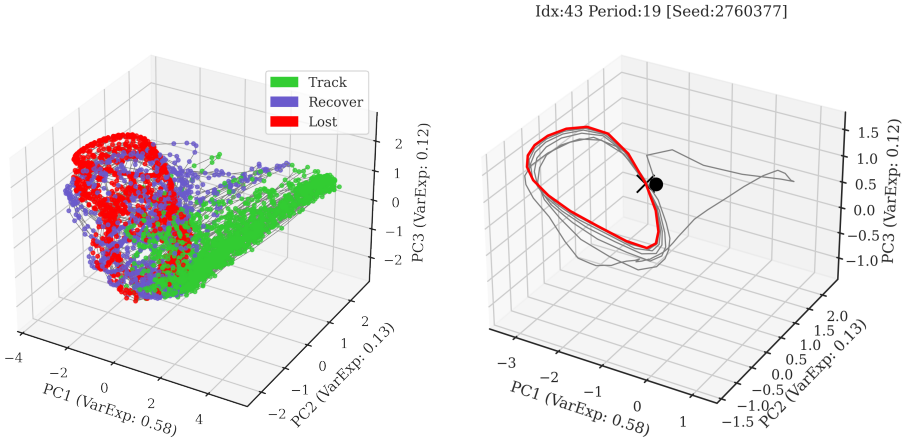

Figure 5.27: Neural dynamics – Agent 1 (compare with Agent 3 in Figure 5.7)

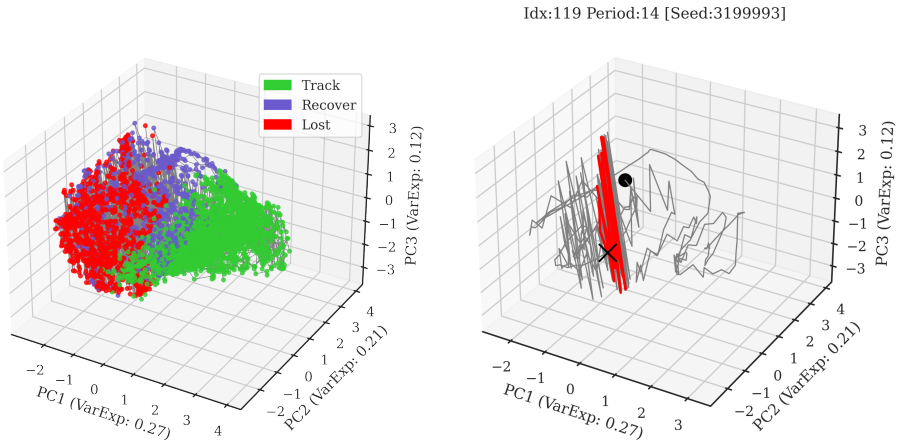

Figure 5.28: Neural dynamics – Agent 2

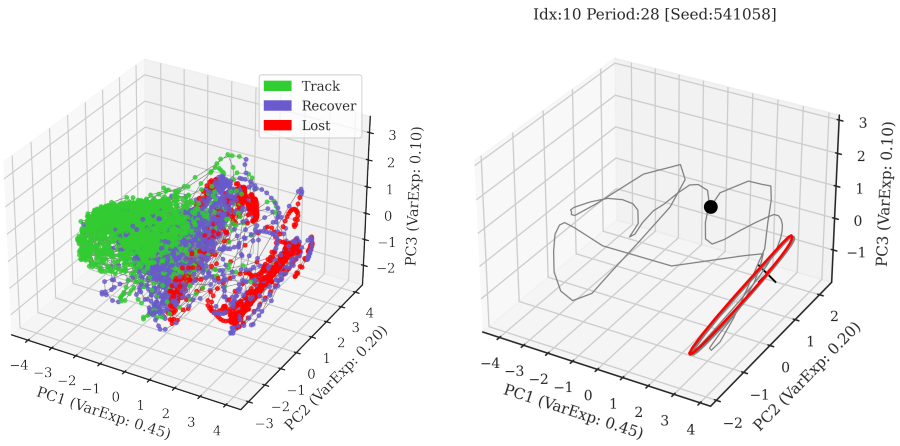

Figure 5.29: Neural dynamics – Agent 3 (same as Figure 5.7)

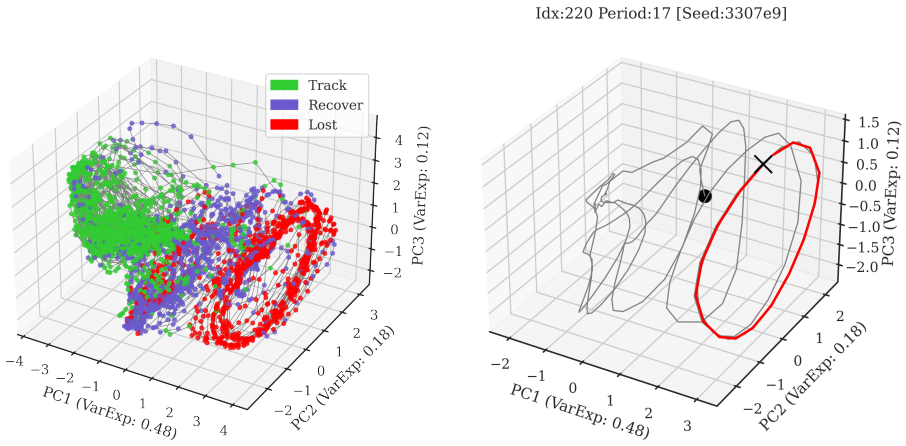

Figure 5.30: Neural dynamics – Agent 4

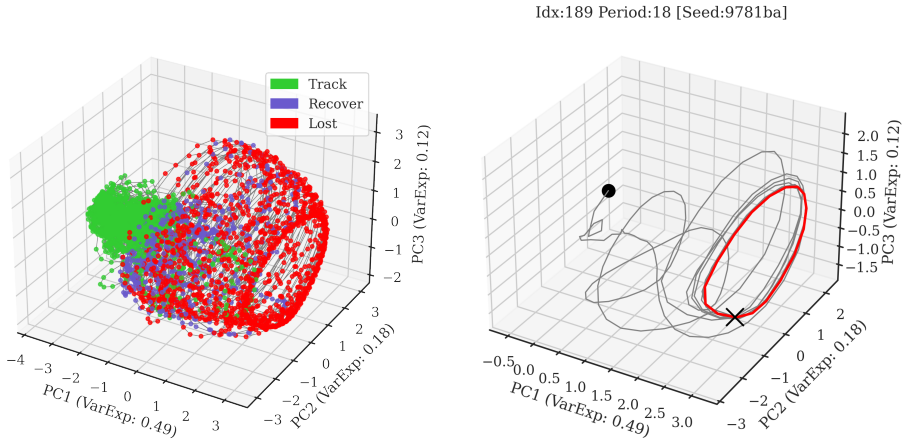

Figure 5.31: Neural dynamics – Agent 5

5.12 Transitions between neural clusters

All plots use the Mann-Whitney-Wilcoxon test two-sided with Bonferroni correction, where p-value annotations indicate:

```
ns: 5.00e-02 < p <= 1.00e+00 (not significant)
*: 1.00e-02 < p <= 5.00e-02
**: 1.00e-03 < p <= 1.00e-02
***: 1.00e-04 < p <= 1.00e-03
****: p <= 1.00e-04
```

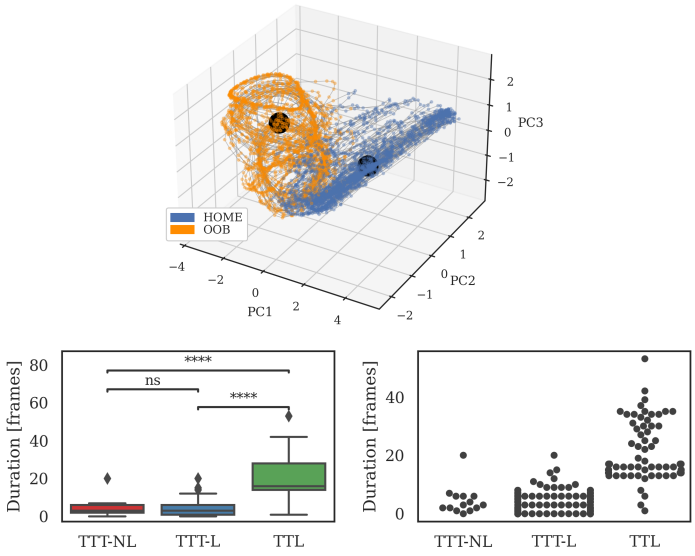

Figure 5.32: Transitions between neural activity regimes – Agent 1 (compare with Agent 3 in Figure 5.8)

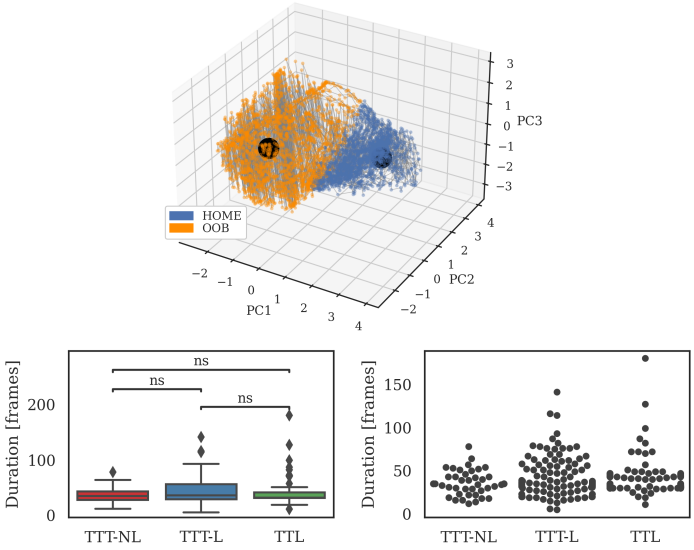

Figure 5.33: Transitions between neural activity regimes – Agent 2 (NB: this agent does not follow the trend)

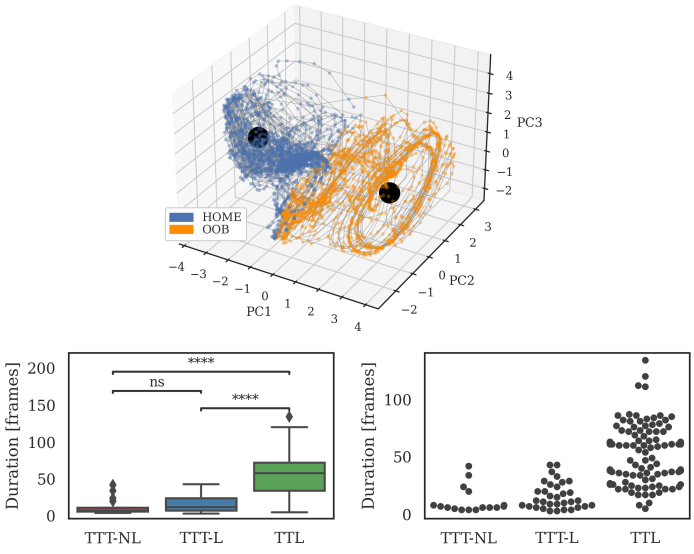

Figure 5.34: Transitions between neural activity regimes – Agent 3 (same data as Figure 5.8)

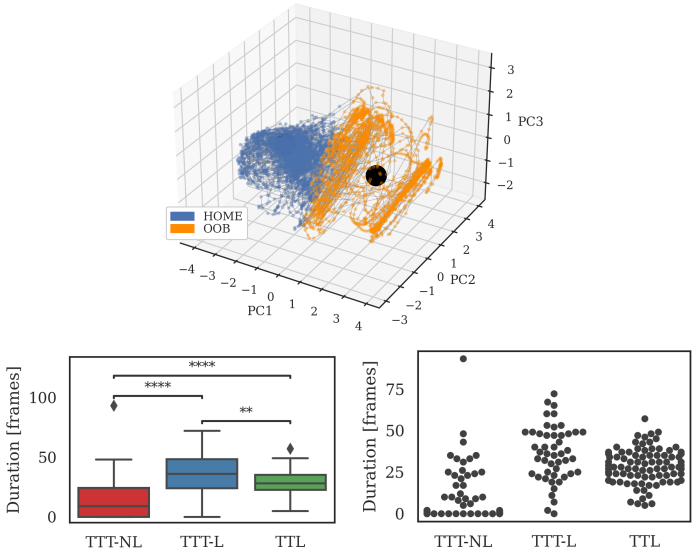

Figure 5.35: Transitions between neural activity regimes – Agent 4

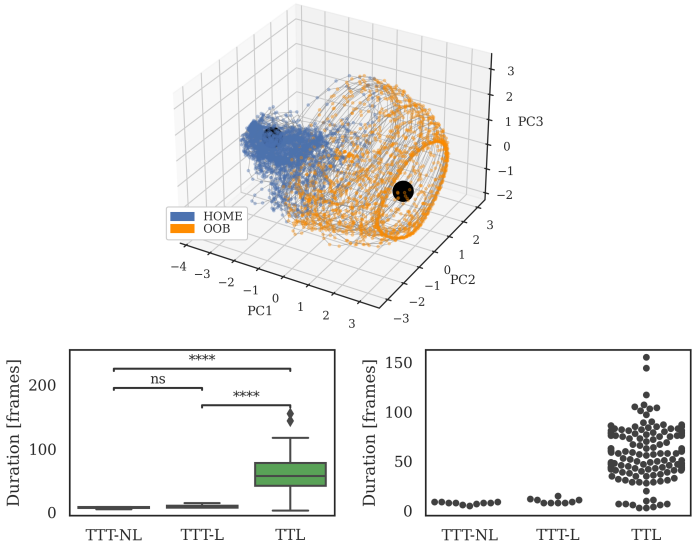

Figure 5.36: Transitions between neural activity regimes – Agent 5

5.13 RNN connectivity and stimulus integration timescales

Agent	Agent ID	Top 5 τs
RNN 1	2760377	116.5, 81.5, 16.9, 13.5, 8.3
RNN 2	3199993	95.7, 61.7, 16.6, 12.0, 9.6
RNN 3	3307e9	56.5, 13.0, 7.7, 6.8, 5.8
RNN 4	541058	86.4, 51.8, 15.1, 12.4, 9.7
RNN 5	9781ba	86.2, 27.4, 8.6, 6.6, 5.6

Table 5.8: Top 5 τs (stimulus integration timescale) for each RNN seed

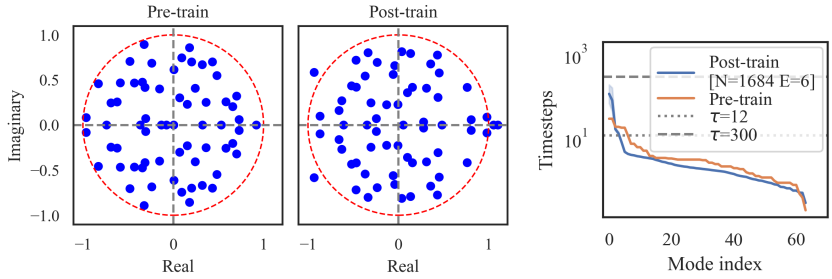

Figure 5.37: Eigenspectra and stimulus integration timescales – Agent 1 (compare with Agent 3 in Figure 5.9)

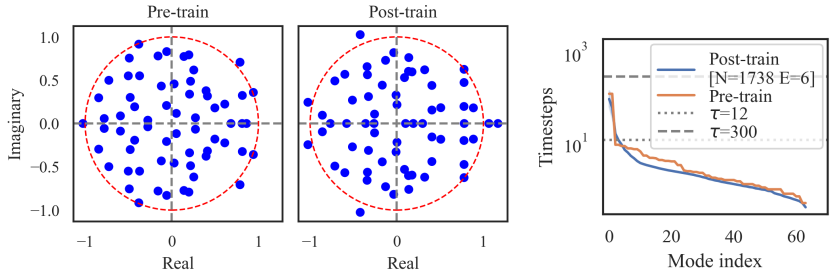

Figure 5.38: Eigenspectra and stimulus integration timescales – Agent 2

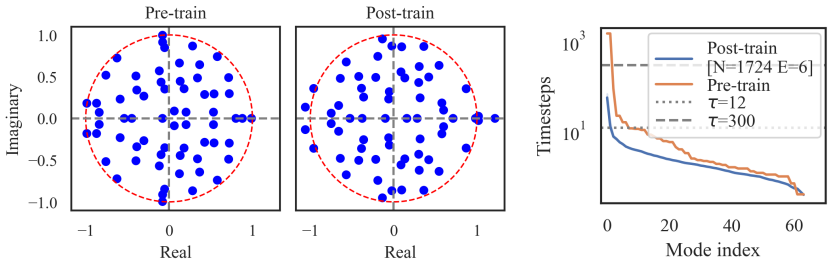

Figure 5.39: Eigenspectra and stimulus integration timescales – Agent 3 (same as Figure 5.9)

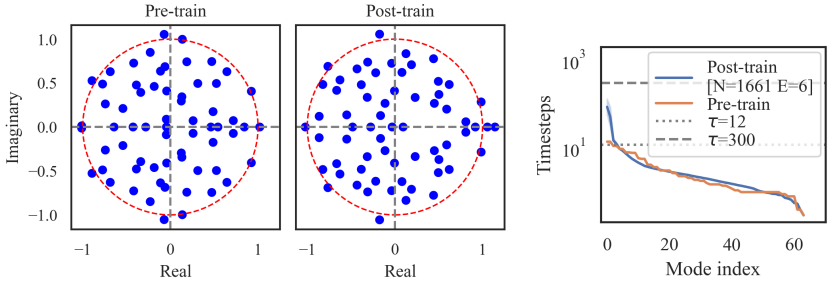

Figure 5.40: Eigenspectra and stimulus integration timescales – Agent 4

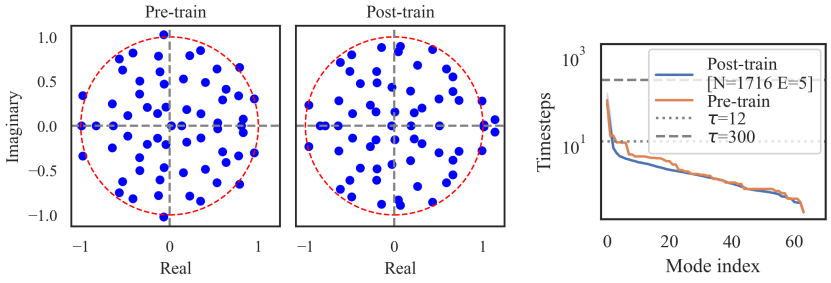

Figure 5.41: Eigenspectra and stimulus integration timescales – Agent 5

Chapter 6

CONCLUSIONS

In this dissertation, we described two different scenarios in which machine learning (ML) was used to facilitate neuroscientific studies of naturalistic behavior and associated neural activity. The two scenarios present distinct engineering & analysis challenges, with one commonality – the complexity of the subjects' naturalistic behavior.

6.1 Mining naturalistic human behaviors in long-term video and neural recordings

In the first "data-rich" scenario (Chapters 3 and 4), we studied naturalistic human upper-limb movements extracted by mining such behaviors from opportunistically-collected long-term human electrocorticography (ECoG) and naturalistic behavior video data for 12 human subjects. We developed a highly automated and scalable approach for analyzing long-term datasets of simultaneously collected human brain and naturalistic behavior data. Our workflow robustly uncovered and annotated thousands of human upper-limb movement events in behavior videos. To detect movement events, we first discretized pose time-series for each wrist into two latent states, indicating movement or rest, and then used regular expressions to look for user-specified patterns in the latent state sequences. This semi-supervised strategy allowed us to rapidly explore movements and their associated brain responses. Importantly, our curated naturalistic dataset supported direct comparison with existing literature from controlled experiments. We found large variability between subjects in the number of events discovered, which we attribute to inter-subject differences in times excluded in the omit-listing process, cycles of sleep and wakeful activity, and clinical treatment regimes (Figures 3.3 and 3.4).

We take a "mining" approach to analysing these large datasets, which contain high variability coming from not only the diversity of patients and human behaviors, but also due to the opportunistic data-collection paradigm. Thus, large chunks of the recordings are discarded, so there is large inter-subject variability in the number of events discovered. We believe this highlights the open research challenges associated with analysing such data. The within-subject variability observed in our event metadata (Figures 3.5, 3.6, and 3.7) is expected to closely reflect the natural statistics of human upper-limb movements while seated, since our subjects received no instructions for when and how to move.

Figure 6.1: Compared to standard computational ethology data analysis pipelines, a novel aspect of our pipeline is that we first learn a temporally precise 'queryable representation' of single limb data (discretization). We then search for patterns (pattern match) that correspond to scientifically interesting behavioral events, e.g. single hand movement, bimanual movements. Finally, we extract metadata about each of these events (e.g. movement duration, speed, start and end locations) that are useful for downstream applications. Separating the pipeline into multiple stages in the manner described allows us to define movement events flexibly, localize them in time precisely and characterize movement trajectories associated with the events.

To demonstrate the applicability of our workflow, we analyzed the brain data associated with the annotated events from two perspectives: characterizing neural correlates of movement initiation, and decoding naturalistic movement initiation using ECoG data. Key to the success of our applications is the availability of a large number of repeated instances of movement initiation events, all available with high temporal precision, which is an essential requirement for generating event-averaged spectrograms [76].

Our prototype spectrograms (Figure 4.1) showed a pattern of movement-associated power increases in a high-frequency band (76–100 Hz) and decreases in a low-frequency band (8–32 Hz) in the same areas as reported in previous controlled laboratory experiments [41, 62]. In our concurrently released preprint [59], we further investigate the consequences of the relatively higher variance of naturalistic movement statistics, model the contributions of the various movement metadata to the observed neural responses, and contrast prior controlled experimental findings with those from our dataset to highlight differences due to the different time-scales of data-collection (a few hours for controlled experiments vs. over several days for ours). In particular, analyses comparing naturalistic and controlled experimental data in [59] were enabled by our ability to match movement statistics between naturalistic and controlled experiments by selecting events by reach magnitude,

onset velocity, and shape metadata.

Similarly, our decoding results show (Figure 4.4) that a low-frequency band (< 35Hz) and a high-frequency band (around 100 Hz) dominates feature importance for most subjects (with some exceptions). This pattern is similar to what has been observed with controlled brain-computer interface studies such as [41], [153], and [154]. Our decoding performance, as measured by test set accuracy, is similar to those in previous studies that also use such opportunistically collected ECoG data [42, 54]. Unlike these previous studies, the ability to select events without opposing wrist activity allowed us to disambiguate confounds when comparing movement decoders for opposing wrists. In most subjects, contralateral limb movement decoding is more accurate than it is for ipsilateral movements, which is expected due to hemispheric lateralization [155]. When motor cortex electrode coverage is limited or unavailable (Figure 4.2), decoding is still possible (Figure 4.3), which is likely due to the decoders exploiting correlations between brain-regions [41, 65, 155]. When motor cortex coverage is available, we see decoder feature importance concentrate itself on this area [41, 42, 54]. The large dataset produced by this approach has also critically enabled the development of a generalized neural-network based decoder adept at transferring across subjects and across recording modalities [75].

6.1.1 Limitations and future work

Our work has a number of limitations that can be improved upon in future work. First, our strategy of discretizing individual keypoint time-series to two latent states and then pattern-matching on latent state sequences may be challenging with more complex behaviors involving coordinated movement of more keypoints. When we increased the number of latent states in the pose segmentation process, we also noticed that behavioral states were harder to interpret and associated ECoG responses were not easily separable. The automated analysis of behavior for simple model organisms such as worms [156], zebrafish [29], flies [9] and mice [6, 74, 157] has advanced to the extent of being able to automatically extract hierarchies of coordinated behavioral sequences (or *grammars*) from naturalistic videos. Except for some very limited work [158, 159], such progress has been elusive in human

computer vision, possibly due to the sheer complexity and variability of human movements in various contexts. Though not tailored to our temporal precision requirements, future research in fine-grained human action recognition in sports [160, 161], domestic [162] and industrial [163] contexts could eventually provide methods that enable the collection of massive datasets of finely annotated human behavior.

All of our data was acquired opportunistically and videos were recorded from a single clinical monitoring camera. Thus, a primary drawback of the event metadata generated by our pipeline is that they are derived from pose-estimation on single-camera RGB images, implying that all pose coordinates are 2D projections and that the fidelity of pose-derived metadata is limited. However, one can still extract utility from the event metadata by coarse-graining or binning them. For example, we are exploring the neural correlates and decoding of coarse reach directions (inwards, outwards, upwards and downwards) instead of exact reach angles in ongoing work. The kinematic dataset could be made significantly richer by the use of additional hardware, such as an RGB-D (RGB with depth) camera or a stereoscopic camera system, that would enable pose-estimation and object-tracking in 3D [22, 164, 165].

We controlled false positives in the event discovery process using a combination of pose-estimation confidence and a tedious manual omit-listing process. We found the confidence estimate provided by our pose-estimation tool to perform well under conditions of good visibility, but it was sensitive to variations arising from naturalistic lighting and occlusions. One potential source of improvement could come from using pose-estimation algorithms that employ body models, such as OpenPose [166]. In our assessment, DeepLabCut [28, 58] offered a better speed (cost) vs. accuracy trade-off at the scale we deployed for pose-estimation. Future work is poised to take advantage of rapid innovations in computer vision, as more tools become available and accessible. While manual creation and review of an omit-list cannot be completely avoided for compliance with human research protocols, we believe that a stereoscopic or depth based camera system could also help detect occlusions better and lead to a reduction in false positives.

Finally, two limitations arise from the opportunistic data-collection paradigm itself. First, we have limited our study to a subject's wrists because they are relatively uncon-

strained and can perform spontaneous naturalistic movements compared to the rest of the subject's body. Our subjects' heads are tethered to a brain recording device that partially restricts the movement of the rest of their upper body. The study of more naturalistic, especially more active, behaviors would require wireless recording. Second, ECoG data such as ours has been obtained opportunistically from a neuro-atypical patient population undergoing long-term monitoring preceding invasive epilepsy resection surgery. Finally, we note with caution that conclusions from analyzing such data might not generalize well to the broader, neuro-typical population.

6.2 Artificial neural network agents modeling turbulent plume tracking in flying insects

In the second "simulator-based" scenario, we studied the behaviors and neural computations associated with turbulent plume tracking by training artificial agents to track simulated turbulent plumes. Despite being trained using only a simple task-aligned reward function, we find that several naturalistic biological features emerge in the agents at the level of behavioral modules, neural representations, and neural dynamics. We believe that our normative neural network agent models enable an abstract yet highly interpretable understanding of the behaviors and neural computations that support plume tracking. We believe that our contributions are a significant first iteration, providing a foundation upon which further work with additional modeling complexity can be developed. We hope that our empirical study will contribute to the growing convergence in the understanding of artificial and biological networks [167, 168], and that attempts to reverse-engineer such neural network agents will help accelerate the development of inverse-models of biological agent behaviors [169, 170]. Our model could also serve as a generative model for complex naturalistic behaviors, which in turn could be used to develop new behavior analysis tools for complex naturalistic behaviors seen in biology [9, 25, 37]. Learnings from these studies could also be used to inform the development of robotic agents with artificial [171] or hybrid [172] olfactory sensing.

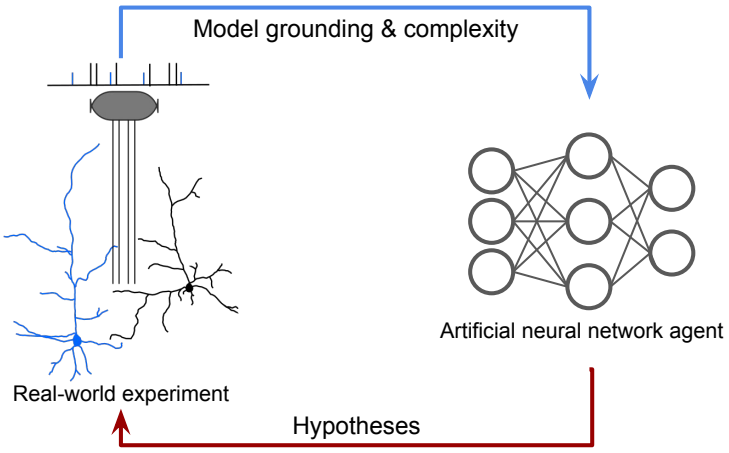

Figure 6.2: Closing the loop between real-world experiments and artificial agent models: Often task-solving artificial neural networks (or agents) exhibit behavioral features or neural computations that resemble their biological counterparts. This makes them useful as tools for generating further hypotheses of behaviors and neural computation in conditions where neural recording, or experimental might be challenging or expensive. Such hypotheses could guide future real-world experiments. Simultaneously, artificial agents could benefit from incorporating mechanisms, architectures and objectives derived from real-world experiments. This could potentially allow artificial models to behave more like biological creatures, that are known to be more robust and fast-adapting when solving complex real-world tasks [173]. A virtuous cycle between real-world experiments and increasingly complex yet biologically grounded artificial models could benefit both the neuroscience and machine intelligence communities. Figure uses a modified version of the open source extracellular recording sketch contributed by Klara Gerlei to scidraw.io [174].

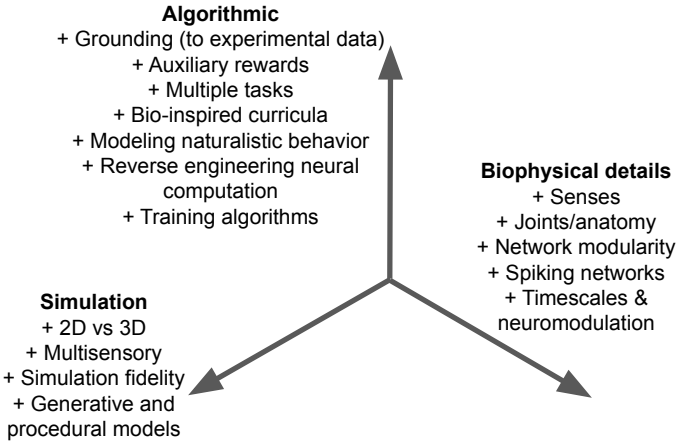

Figure 6.3: We believe that our artificial neural network agents can be improved along three distinct dimensions: Algorithmic, Simulation capabilities, and Biophysical complexity modeled. We describe these in detail in Section 6.2.1.

6.2.1 Limitations and future work

A few limitations of our approach must be kept in mind while drawing conclusions from this work: First, ANNs are coarse approximations of models of real biological circuits, and do not account for information stored in individual neuronal spikes or on the timescale of spike timing differences. Future models could use emerging techniques for reward-based training of spiking neural networks [175–177] to train more biologically plausible spiking neural networks. Furthermore, constraining model wiring during training using connectomes from real flying insects [178] could provide a fine-grained correspondence between biological circuits and the trained artificial model. Such models could also serve as testbeds for the development of network causal inference methods [179–181].

Second, like related research that uses ANNs to model neuronal circuits [96, 97, 182, 183], we train our artificial agents to solve a single task. We expect, however, that training agents on multiple tasks [184] is likely to produce agents with not just a richer repertoire of behaviors, but also more complex neural activity structures with shared task-representations, shared task-dynamics and task-specific adaptations [130, 185–189].

Unlike flying insects exhibit significant speed modulation, our agents mostly flew at

either their maximum speed or slowed down to a near-halt (see Figure 5.3.) Adding penalties for forward movement to the reward function did induce some speed modulation, but was not used because it drastically reduced agent performance even for small penalties. Future work should explore the role of skewing DRL reward functions towards such auxiliary goals that are not aligned with the primary task (plume tracking) being performed by the agent. Other directions for improving the fidelity of the model include modeling vision and other senses, using a biomechanical model of the flying insect [190] and training the model in a 3D virtual-reality environment for simulating more realistic perception [117, 191].

Our work also suggests a few directions for further methodological innovation: Currently popular methods for neural network comparison [192, 193] are focused on comparing the responses of feedforward networks to a library of stimuli, and, by design, do not take into account the order or amount of stimuli previously seen by the network. Though alternative approaches developed for use with RNNs do take stimulus history into consideration, they do not account for the network being deployed in a closed-loop (agent) setting where the network's current state determines it's next action, which in turn affects the next observation seen by it [51, 52, 121]. New theoretical development is required for comparing RNN-based agents, specifically to deal with small differences in stimuli, which in turn can generate actions that result in compounding differences over trajectories. Furthermore, current methods characterize differences between networks by looking at discrete dynamical features such as fixed-points and neural activity transitions between them. It is unclear what measures could be used for comparing the continuous and somewhat amorphous dynamical structures such as the ones we encountered in our work (Figure 5.7).

Learning algorithms that respect biological constraints (e.g. excitation-inhibition balance, Dale's law) have received attention in the supervised-learning setting [194–196] but not in the reinforcement-learning setting. Finally, despite observing similar behavioral modules and neural activity structures across trained agents, we observed a significant variation in performance (Section 5.4.3). It might be possible to mitigate this by constraining training using additional tasks, as suggested earlier. However, alternative solutions could involve using meta-algorithms like evolution [197], using algorithms constrained by morphology [198], or using more complex biologically-inspired training curricula [152].

BIBLIOGRAPHY

[1] A. Huk, K. Bonnen, and B. J. He, "Beyond trial-based paradigms: Continuous behavior, ongoing neural activity, and natural stimuli," *Journal of Neuroscience*, vol. 38, no. 35, pp. 7551–7558, 2018.

[2] G. Felsen and Y. Dan, "A Natural Approach to Studying Vision," *Nature Neuroscience*, vol. 8, pp. 1643–1646, Dec. 2005.

[3] J. Derix, O. Iljina, A. Schulze-Bonhage, A. Aertsen, and T. Ball, ""Doctor" or "Darling"? Decoding the Communication Partner from ECoG of the Anterior Temporal Lobe during Non-Experimental, Real-Life Social Interaction," *Frontiers in Human Neuroscience*, vol. 6, 2012.

[4] D. Anderson and P. Perona, "Toward a Science of Computational Ethology," *Neuron*, vol. 84, pp. 18–31, Oct. 2014.

[5] A. Wiltschko, M. Johnson, G. Iurilli, R. Peterson, J. Katon, S. Pashkovski, V. Abraira, R. Adams, and S. Datta, "Mapping Sub-Second Structure in Mouse Behavior," *Neuron*, vol. 88, pp. 1121–1135, Dec. 2015.

[6] J. Markowitz, W. Gillis, C. Beron, S. Neufeld, K. Robertson, N. Bhagat, R. Peterson, E. Peterson, M. Hyun, S. Linderman, B. Sabatini, and S. Datta, "The Striatum Organizes 3D Behavior via Moment-to-Moment Action Selection," *Cell*, vol. 174, pp. 44–58.e17, June 2018.

[7] S. A. Nastase, A. Goldstein, and U. Hasson, "Keep it real: rethinking the primacy of experimental control in cognitive neuroscience," *NeuroImage*, vol. 222, p. 117254, 2020.

[8] S. Sonkusare, M. Breakspear, and C. Guo, "Naturalistic stimuli in neuroscience: critically acclaimed," *Trends in cognitive sciences*, vol. 23, no. 8, pp. 699–714, 2019.

[9] G. J. Berman, W. Bialek, and J. W. Shaevitz, "Predictability and hierarchy in drosophila behavior," *Proceedings of the National Academy of Sciences*, vol. 113, no. 42, pp. 11943–11948, 2016.

[10] O. Özdenizci, Y. Wang, T. Koike-Akino, and D. Erdoğmuş, "Transfer learning in brain-computer interfaces with adversarial variational autoencoders," in *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*, pp. 207–210, IEEE, 2019.

[11] P. Shenoy, M. Krauledat, B. Blankertz, R. P. Rao, and K.-R. Müller, "Towards adaptive classification for BCI," *Journal of Neural Engineering*, vol. 3, no. 1, p. R13, 2006.

[12] P. Aspinall, P. Mavros, R. Coyne, and J. Roe, "The urban brain: analysing outdoor physical activity with mobile eeg," *British journal of sports medicine*, vol. 49, no. 4, pp. 272–276, 2015.

[13] E. Boto, N. Holmes, J. Leggett, G. Roberts, V. Shah, S. S. Meyer, L. D. Muñoz, K. J. Mullinger, T. M. Tierney, S. Bestmann, *et al.*, "Moving magnetoencephalography towards real-world applications with a wearable system," *Nature*, vol. 555, no. 7698, pp. 657–661, 2018.

[14] A. von Lühmann, Y. Zheng, A. Ortega-Martinez, S. Kiran, D. C. Somers, A. Cronin-Golomb, L. N. Awad, T. D. Ellis, D. A. Boas, and M. A. Yücel, "Towards neuroscience of the everyday world (new) using functional near infrared spectroscopy," *Current opinion in biomedical engineering*, p. 100272, 2021.

[15] E. Dennis, A. E. Hady, A. Michaiel, A. M. Clemens, D. R. G. Tervo, J. Voigts, and S. R. Datta, "Systems neuroscience of natural behaviors in rodents," Oct 2020.

[16] M. Privat and G. Sumbre, "Naturalistic behavior: The zebrafish larva strikes back," *Current Biology*, vol. 30, no. 1, pp. R27–R29, 2020.

[17] J. P. Nguyen, F. B. Shipley, A. N. Linder, G. S. Plummer, M. Liu, S. U. Setru, J. W. Shaevitz, and A. M. Leifer, "Whole-brain calcium imaging with cellular resolution

in freely behaving caenorhabditis elegans," *Proceedings of the National Academy of Sciences*, vol. 113, no. 8, pp. E1074–E1081, 2016.

[18] R. R. Harrison, R. J. Kier, A. Leonardo, H. Fotowat, R. Chan, and F. Gabbiani, "A wireless neural/emg telemetry system for freely moving insects," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pp. 2940–2943, IEEE, 2010.

[19] A. Regalado, "A paralyzed man is challenging neuralink's monkey to a match of mind pong," May 2021.

[20] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Computer vision and image understanding*, vol. 81, no. 3, pp. 231–268, 2001.

[21] A. Mathis, P. Mamidanna, K. Cury, T. Abe, V. Murthy, M. Mathis, and M. Bethge, "DeepLabCut: Markerless pose estimation of user-defined body parts with deep learning," tech. rep., Nature Publishing Group, 2018.

[22] P. Karashchuk, K. L. Rupp, E. S. Dickinson, E. Sanders, E. Azim, B. W. Brunton, and J. C. Tuthill, "Anipose: a toolkit for robust markerless 3D pose estimation," *BioRxiv*, 2020.

[23] E. Batty, M. Whiteway, S. Saxena, D. Biderman, T. Abe, S. Musall, W. Gillis, J. Markowitz, A. Churchland, J. Cunningham, *et al.*, "BehaveNet: nonlinear embedding and bayesian neural decoding of behavioral videos," in *Advances in Neural Information Processing Systems*, pp. 15680–15691, 2019.

[24] T. D. Pereira, D. E. Aldarondo, L. Willmore, M. Kislin, S. S.-H. Wang, M. Murthy, and J. W. Shaevitz, "Fast animal pose estimation using deep neural networks," *Nature Methods*, vol. 16, no. 1, p. 117, 2019.

[25] J. Nassar, S. Linderman, M. Bugallo, and I. M. Park, "Tree-structured recurrent switching linear dynamical systems for multi-scale modeling," in *International Conference on Learning Representations*, 2018.

[26] M. Johnson, D. Duvenaud, A. Wiltschko, R. Adams, and S. Datta, "Composing graphical models with neural networks for structured representations and fast inference," in *Advances in Neural Information Processing Systems 29*, pp. 2946–2954, Curran Associates, Inc., 2016.

[27] M. H. McCullough and G. J. Goodhill, "Unsupervised quantification of naturalistic animal behaviors for gaining insight into the brain," *Current Opinion in Neurobiology*, vol. 70, pp. 89–100, 2021.

[28] M. Mathis and A. Mathis, "Deep learning tools for the measurement of animal behavior in neuroscience," *Current Opinion in Neurobiology*, vol. 60, pp. 1–11, 2020.

[29] R. E. Johnson, S. Linderman, T. Panier, C. L. Wee, E. Song, K. J. Herrera, A. Miller, and F. Engert, "Probabilistic models of larval zebrafish behavior reveal structure on many scales," *Current Biology*, vol. 30, no. 1, pp. 70–82, 2020.

[30] G. Berman, "Measuring behavior across scales," *BMC Biology*, vol. 16, Dec. 2018.

[31] S. Ramasamy Ramamurthy and N. Roy, "Recent trends in machine learning for human activity recognition - A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1254, 2018.

[32] S. Ghorbani, K. Mahdaviani, A. Thaler, K. Kording, D. J. Cook, G. Blohm, and N. F. Troje, "Movi: A large multipurpose motion and video dataset," *arXiv preprint arXiv:2003.01888*, 2020.

[33] N. Seethapathi, S. Wang, R. Saluja, G. Blohm, and K. P. Kording, "Movement science needs different pose tracking algorithms," *arXiv preprint arXiv:1907.10226*, 2019.

[34] D. Fu, W. Crichton, J. Hong, X. Yao, H. Zhang, A. Truong, A. Narayan, M. Agrawala, C. Ré, and K. Fatahalian, "Rekall: Specifying Video Events using Compositions of Spatiotemporal Labels," *arXiv:1910.02993 [cs]*, Oct. 2019.

[35] S. H. Singh, F. van Breugel, R. P. N. Rao, and B. W. Brunton, "Emergent behavior and neural dynamics in artificial agents tracking turbulent plumes," 2021.

[36] S. H. Singh, F. van Breugel, R. P. Rao, and B. W. Brunton, "Understanding biological plume tracking behavior using deep reinforcement-learning," in *Artificial Life Conference Proceedings*, pp. 750–752, MIT Press One Rogers Street, Cambridge, MA 02142-1209 USA journals-info@ mit ..., 2020.

[37] S. H. Singh, S. M. Peterson, R. P. Rao, and B. W. Brunton, "Mining naturalistic human behaviors in long-term video and neural recordings," *Journal of Neuroscience Methods*, vol. 358, p. 109199, 2021.

[38] S. Singh, S. Peterson, R. Rao, and B. Brunton, "Enabling naturalistic neuroscience through behavior mining: Analysis of long-term human brain and video recordings," 01 2019.

[39] S. M. Peterson, S. H. Singh, B. Dichter, M. Scheid, R. P. Rao, and B. W. Brunton, "Ajile12: Long-term naturalistic human intracranial neural recordings and pose," *bioRxiv*, 2021.

[40] M. J. Johnson and A. S. Willsky, "Bayesian Nonparametric Hidden Semi-Markov Models," *Journal of Machine Learning Research*, vol. 14, pp. 673–701, February 2013.

[41] K. Miller, E. Leuthardt, G. Schalk, R. Rao, N. Anderson, D. Moran, J. Miller, and J. Ojemann, "Spectral Changes in Cortical Surface Potentials during Motor Movement," *Journal of Neuroscience*, vol. 27, pp. 2424–2432, Feb. 2007.

[42] N. Wang, A. Farhadi, R. Rao, and B. Brunton, "AJILE movement prediction: Multimodal deep learning for natural human neural recordings and video," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[43] J. A. Farrell, J. Murlis, X. Long, W. Li, and R. T. Cardé, "Filament-based atmospheric dispersion model to achieve short time-scale structure of odor plumes," *Environmental Fluid Mechanics*, vol. 2, no. 1-2, pp. 143–169, 2002.

[44] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," in *Advances in neural information processing systems*, pp. 1008–1014, 2000.

[45] F. van Breugel and M. H. Dickinson, "Plume-tracking behavior of flying drosophila emerges from a set of distinct sensory-motor reflexes," *Current Biology*, vol. 24, no. 3, pp. 274–286, 2014.

[46] F. van Breugel, W. Regan, and H. Lipson, "From insects to machines," *IEEE Robotics & Automation Magazine*, vol. 15, no. 4, pp. 68–74, 2008.

[47] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[48] K. L. Baker, M. Dickinson, T. M. Findley, D. H. Gire, M. Louis, M. P. Suver, J. V. Verhagen, K. I. Nagel, and M. C. Smear, "Algorithms for olfactory search across species," *Journal of Neuroscience*, vol. 38, no. 44, pp. 9383–9389, 2018.

[49] R. T. Cardé and M. A. Willis, "Navigational strategies used by insects to find distant, wind-borne sources of odor," *Journal of Chemical Ecology*, vol. 34, no. 7, pp. 854–866, 2008.

[50] S. A. Budick and M. H. Dickinson, "Free-flight responses of drosophila melanogaster to attractive odors," *Journal of Experimental Biology*, vol. 209, no. 15, pp. 3001–3017, 2006.

[51] D. Sussillo and O. Barak, "Opening the black box: low-dimensional dynamics in high-dimensional recurrent neural networks," *Neural computation*, vol. 25, no. 3, pp. 626–649, 2013.

[52] N. Maheswaranathan, A. Williams, M. Golub, S. Ganguli, and D. Sussillo, "Reverse engineering recurrent networks for sentiment classification reveals line attractor dynamics," in *Advances in Neural Information Processing Systems*, pp. 15696–15705, 2019.

[53] A. Huk, K. Bonnen, and B. He, "Beyond Trial-Based Paradigms: Continuous Behavior, Ongoing Neural Activity, and Natural Stimuli," *The Journal of Neuroscience*, pp. 1920–17, July 2018.

[54] P. Gabriel, K. Chen, A. Alasfour, T. Pailla, W. Doyle, O. Devinsky, D. Friedman, P. Dugan, L. Melloni, T. Thesen, D. Gonda, S. Sattar, S. Wang, and V. Gilja, "Neural Correlates of Unstructured Motor Behaviors," *Journal of Neural Engineering*, vol. 16, p. 066026, Oct. 2019.

[55] N. Wang, J. Olson, J. Ojemann, R. Rao, and B. Brunton, "Unsupervised Decoding of Long-Term, Naturalistic Human Neural Recordings with Automated Video and Audio Annotations," *Frontiers in Human Neuroscience*, vol. 10, Apr. 2016.

[56] A. Alasfour, P. Gabriel, X. Jiang, I. Shamie, L. Melloni, T. Thesen, P. Dugan, D. Friedman, W. Doyle, O. Devinsky, *et al.*, "Coarse behavioral context decoding," *Journal of Neural Engineering*, vol. 16, no. 1, p. 016021, 2019.

[57] C. Chambers, N. Seethapathi, R. Saluja, H. Loeb, S. R. Pierce, D. K. Bogen, L. Prosser, M. J. Johnson, and K. P. Kording, "Computer vision to automatically assess infant neuromotor risk," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 11, pp. 2431–2442, 2020.

[58] T. Nath, A. Mathis, A. Chen, A. Patel, M. Bethge, and M. Mathis, "Using DeepLabCut for 3D markerless pose estimation across species and behaviors," *Nature Protocols*, vol. 14, pp. 2152–2176, 2019.

[59] S. M. Peterson, S. H. Singh, N. X. Wang, R. P. Rao, and B. W. Brunton, "Behavioral and neural variability of naturalistic arm movements," *BioRxiv*, 2020.

[60] J. Parvizi and S. Kastner, "Human intracranial EEG: promises and limitations," *Nature Neuroscience*, vol. 21, no. 4, p. 474, 2018.

[61] J. Parvizi and S. Kastner, "Promises and limitations of human intracranial electroencephalography," *Nature Neuroscience*, p. 1, 2018.

[62] K. J. Miller, "A library of human electrocorticographic data and analyses," *Nature Human Behaviour*, vol. 3, no. 11, pp. 1225–1235, 2019.

[63] E. C. Leuthardt, G. Schalk, J. R. Wolpaw, J. G. Ojemann, and D. W. Moran, "A brain–computer interface using electrocorticographic signals in humans," *Journal of neural engineering*, vol. 1, no. 2, p. 63, 2004.

[64] E. C. Leuthardt, K. J. Miller, G. Schalk, R. P. Rao, and J. G. Ojemann, "Electrocorticography-based brain computer interface-the seattle experience," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 14, no. 2, pp. 194–198, 2006.

[65] G. Schalk, J. Kubanek, K. Miller, N. Anderson, E. Leuthardt, J. Ojemann, D. Limbrick, D. Moran, L. Gerhardt, and J. Wolpaw, "Decoding two-dimensional movement trajectories using electrocorticographic signals in humans," *Journal of Neural Engineering*, vol. 4, no. 3, p. 264, 2007.

[66] A. Delorme, S. Makeig, and T. Sejnowski, "Automatic artifact rejection for eeg data using high-order statistics and independent component analysis," in *Proceedings of the third international ICA conference*, pp. 9–12, Citeseer, 2001.

[67] A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, L. Parkkonen, and M. S. Hämäläinen, "Mne software for processing meg and eeg data," *Neuroimage*, vol. 86, pp. 446–460, 2014.

[68] R. Oostenveld, P. Fries, E. Maris, and J.-M. Schoffelen, "Fieldtrip: Open source software for advanced analysis of MEG, EEG, and invasive electrophysiological data," *Computational Intelligence and Neuroscience*, vol. 2011, pp. 156869–156869, 2011.

[69] A. Stolk, S. Griffin, R. van der Meij, C. Dewar, I. Saez, J. J. Lin, G. Piantoni, J.-M. Schoffelen, R. T. Knight, and R. Oostenveld, "Integrated analysis of anatomical and electrophysiological human intracranial data," *Nature Protocols*, vol. 13, no. 7, pp. 1699–1723, 2018.

[70] L. Bourdev and J. Malik, "Poselets: Body part detectors trained using 3D human pose annotations," in *International Conference on Computer Vision*, sep 2009.

[71] M. A. McDowell, C. D. Fryar, and C. L. Ogden, "Anthropometric reference data for children and adults: United states, 1988-1994.," *Vital and health statistics. Series 11, Data from the national health survey*, no. 249, pp. 1–68, 2009.

[72] R. W. Schafer, "What is a Savitzky-Golay filter?," *IEEE Signal Processing Magazine*, vol. 28, no. 4, pp. 111–117, 2011.

[73] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[74] S. R. Datta, "Q&A: Understanding the composition of behavior," *BMC Biology*, vol. 17, no. 1, p. 44, 2019.

[75] S. M. Peterson, Z. Steine-Hanson, N. Davis, R. P. Rao, and B. W. Brunton, "Generalized neural decoders for transfer learning across participants and recording modalities," *Journal of Neural Engineering*, vol. 18, no. 2, p. 026014, 2021.

[76] M. Cohen, *Analyzing Neural Time Series Data: Theory and Practice*. MIT Press, Jan. 2014.

[77] S. M. Peterson, S. H. Singh, N. X. Wang, R. P. Rao, and B. W. Brunton, "Behavioral and neural variability of naturalistic arm movements," *Eneuro*, 2021.

[78] M. M. Shanechi, "Brain–machine interfaces from motor to mood," *Nature Neuroscience*, vol. 22, no. 10, pp. 1554–1564, 2019.

[79] E. Smalley, "The business of brain-computer interfaces.," *Nature Biotechnology*, vol. 37, no. 9, p. 978, 2019.

[80] M. M. Shanechi, "Brain–machine interfaces," in *Dynamic Neuroscience*, pp. 197–218, Springer, 2018.

[81] D. J. Warren, S. Kellis, J. G. Nieveen, S. M. Wendelken, H. Dantas, T. S. Davis, D. T. Hutchinson, R. A. Normann, G. A. Clark, and V. J. Mathews, "Recording and decoding for neural prostheses," *Proceedings of the IEEE*, vol. 104, no. 2, pp. 374–391, 2016.

[82] K. Shenoy and C. Chestek, "Neural prosthetics," *Scholarpedia*, vol. 7, no. 3, p. 11854, 2012.

[83] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, pp. 5–32, Oct. 2001.

[84] T. Hastie, R. Tibshirani, and J. Friedman, "The elements of statistical learning: data mining, inference, and prediction, Springer Series in Statistics," 2009.

[85] A. Farshchian, J. A. Gallego, J. P. Cohen, Y. Bengio, L. E. Miller, and S. A. Solla, "Adversarial domain adaptation for stable brain-machine interfaces," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, OpenReview.net, 2019.

[86] S. L. Klosterman, J. R. Estepp, J. W. Monnin, and J. C. Christensen, "Day-to-day variability in hybrid, passive brain-computer interfaces: Comparing two studies assessing cognitive workload," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 1584–1590, IEEE, 2016.

[87] I. J. Park, A. M. Hein, Y. V. Bobkov, M. A. Reidenbach, B. W. Ache, and J. C. Principe, "Neurally encoding time for olfactory navigation," *PLoS Computational Biology*, vol. 12, no. 1, 2016.

[88] T. A. Currier and K. I. Nagel, "Multisensory control of navigation in the fruit fly," *Current opinion in neurobiology*, vol. 64, pp. 10–16, 2020.

[89] T. Lochmatter and A. Martinoli, "Theoretical analysis of three bio-inspired plume tracking algorithms," in *2009 IEEE International Conference on Robotics and Automation*, pp. 2661–2668, IEEE, 2009.

[90] R. Pang, F. van Breugel, M. Dickinson, J. A. Riffell, and A. Fairhall, "History dependence in insect flight decisions during odor tracking," *PLoS computational biology*, vol. 14, no. 2, p. e1005969, 2018.

[91] X. Sun, M. Mangan, and S. Yue, "An analysis of a ring attractor model for cue integration," in *Conference on Biomimetic and Biohybrid Systems*, pp. 459–470, Springer, 2018.

[92] P. K. Kaushik, M. Renz, and S. B. Olsson, "Characterizing long-range search behavior in diptera using complex 3d virtual environments," *Proceedings of the National Academy of Sciences*, vol. 117, no. 22, pp. 12201–12207, 2020.

[93] N. Kriegeskorte, "Deep Neural Networks: A new framework for modeling biological vision and brain information processing," *Annual Review of Vision Science*, vol. 1, pp. 417–446, 2015.

[94] D. Sussillo, M. M. Churchland, M. T. Kaufman, and K. V. Shenoy, "A neural network that finds a naturalistic solution for the production of muscle activity," *Nature Neuroscience*, vol. 18, no. 7, pp. 1025–1033, 2015.

[95] I. Kanitscheider and I. Fiete, "Training recurrent networks to generate hypotheses about how the brain solves hard navigation problems," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 4532–4541, 2017.

[96] C. J. Cueva and X.-X. Wei, "Emergence of grid-like representations by training recurrent neural networks to perform spatial localization," in *International Conference on Learning Representations*, 2018.

[97] 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*, 2019.

[98] M. Haesemeyer, A. F. Schier, and F. Engert, "Convergent temperature representations in artificial and biological neural networks," *Neuron*, vol. 103, no. 6, pp. 1123–1134, 2019.

[99] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[100] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT press, 2018.

[101] B. A. Richards, T. P. Lillicrap, P. Beaudoin, Y. Bengio, R. Bogacz, A. Christensen, C. Clopath, R. P. Costa, A. de Berker, S. Ganguli, *et al.*, "A deep learning framework for neuroscience," *Nature neuroscience*, vol. 22, no. 11, pp. 1761–1770, 2019.

[102] F. Le Moël and A. Wystrach, "Towards a multi-level understanding in insect navigation," *Current Opinion in Insect Science*, 2020.

[103] J. Merel, D. Aldarondo, J. Marshall, Y. Tassa, G. Wayne, and B. Ölveczky, "Deep neuroethology of a virtual rodent," *arXiv preprint arXiv:1911.09451*, 2019.

[104] M. B. Ahrens, "Zebrafish neuroscience: Using artificial neural networks to help understand brains," *Current Biology*, vol. 29, no. 21, pp. R1138–R1140, 2019.

[105] A. Banino, C. Barry, B. Uria, C. Blundell, T. Lillicrap, P. Mirowski, A. Pritzel, M. J. Chadwick, T. Degris, J. Modayil, *et al.*, "Vector-based navigation using grid-like representations in artificial agents," *Nature*, vol. 557, no. 7705, pp. 429–433, 2018.

[106] S. Colabrese, K. Gustavsson, A. Celani, and L. Biferale, "Flow navigation by smart microswimmers via reinforcement learning," *Physical Review Letters*, vol. 118, no. 15, p. 158004, 2017.

[107] C. Wall and J. Perry, "Range of action of moth sex-attractant sources," *Entomologia experimentalis et applicata*, vol. 44, no. 1, pp. 5–14, 1987.

[108] A. Weinstein and M. M. Botvinick, "Structure learning in motor control: A deep reinforcement learning model," *arXiv preprint arXiv:1706.06827*, 2017.

[109] S. Song, Ł. Kidziński, X. B. Peng, C. Ong, J. L. Hicks, S. Levine, C. Atkeson, and S. Delp, "Deep reinforcement learning for modeling human locomotion control in neuromechanical simulation," *bioRxiv*, 2020.

[110] D. Lin and B. A. Richards, "Time cell encoding in deep reinforcement learning agents depends on mnemonic demands," *bioRxiv*, 2021.

[111] H. F. Song, G. R. Yang, and X.-J. Wang, "Reward-based training of recurrent neural networks for cognitive and value-based tasks," *Elife*, vol. 6, p. e21492, 2017.

[112] J. X. Wang, Z. Kurth-Nelson, D. Kumaran, D. Tirumala, H. Soyer, J. Z. Leibo, D. Hassabis, and M. Botvinick, "Prefrontal cortex as a meta-reinforcement learning system," *Nature neuroscience*, vol. 21, no. 6, pp. 860–868, 2018.

[113] M. Botvinick, S. Ritter, J. X. Wang, Z. Kurth-Nelson, C. Blundell, and D. Hassabis, "Reinforcement learning, fast and slow," *Trends in Cognitive Sciences*, 2019.

[114] L. Cross, J. Cockburn, Y. Yue, and J. P. O'Doherty, "Using deep reinforcement learning to reveal how the brain encodes abstract state-space representations in high-dimensional environments," *Neuron*, vol. 109, no. 4, pp. 724–738, 2021.

[115] M. Botvinick, J. X. Wang, W. Dabney, K. J. Miller, and Z. Kurth-Nelson, "Deep reinforcement learning and its neuroscientific implications," *Neuron*, 2020.

[116] S. J. Gershman and B. P. Ölveczky, "The neurobiology of deep reinforcement learning," *Current Biology*, vol. 30, no. 11, pp. R629–R632, 2020.

[117] M. Crosby, "Building thinking machines by solving animal cognition tasks," *Minds and Machines*, pp. 1–27, 2020.

[118] G. Reddy, B. I. Shraiman, and M. Vergassola, "Sector search strategies for odor trail tracking," *bioRxiv*, 2021.

[119] H. Rapp and M. P. Nawrot, "A spiking neural program for sensorimotor control during foraging in flying insects," *Proceedings of the National Academy of Sciences*, vol. 117, no. 45, pp. 28412–28421, 2020.

[120] K. Rajan and L. F. Abbott, "Eigenvalue spectra of random matrices for neural networks," *Physical review letters*, vol. 97, no. 18, p. 188104, 2006.

[121] N. Maheswaranathan, A. Williams, M. Golub, S. Ganguli, and D. Sussillo, "Universality and individuality in neural dynamics across large populations of recurrent networks," in *Advances in neural information processing systems*, pp. 15629–15641, 2019.

[122] S. Vyas, M. D. Golub, D. Sussillo, and K. V. Shenoy, "Computation through neural population dynamics," *Annual Review of Neuroscience*, vol. 43, pp. 249–275, 2020.

[123] R. B. Ebitz and B. Y. Hayden, "The population doctrine revolution in cognitive neurophysiology," *arXiv preprint arXiv:2104.00145*, 2021.

[124] G. Kowadlo and R. A. Russell, "Robot odor localization: a taxonomy and survey," *The International Journal of Robotics Research*, vol. 27, no. 8, pp. 869–894, 2008.

[125] M. P. Michaelides and C. G. Panayiotou, "Plume source position estimation using sensor networks," in *Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control, 2005.*, pp. 731–736, IEEE, 2005.

[126] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[127] M. Henaff, A. Szlam, and Y. LeCun, "Recurrent orthogonal networks and long-memory tasks," in *International Conference on Machine Learning*, pp. 2034–2042, PMLR, 2016.

[128] T. P. Vogels, K. Rajan, and L. F. Abbott, "Neural network dynamics," *Annu. Rev. Neurosci.*, vol. 28, pp. 357–376, 2005.

[129] D. Sussillo, "Neural circuits as computational dynamical systems," *Current opinion in neurobiology*, vol. 25, pp. 156–163, 2014.

[130] G. R. Yang, M. R. Joglekar, H. F. Song, W. T. Newsome, and X.-J. Wang, "Task representations in neural networks trained to perform many cognitive tasks," *Nature neuroscience*, vol. 22, no. 2, pp. 297–306, 2019.

[131] S. Saxena and J. P. Cunningham, "Towards the neural population doctrine," *Current opinion in neurobiology*, vol. 55, pp. 103–111, 2019.

[132] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[133] C. Strobl, A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis, "Conditional variable importance for random forests," *BMC bioinformatics*, vol. 9, no. 1, pp. 1–11, 2008.

[134] J. S. Kennedy and D. Marsh, "Pheromone-regulated anemotaxis in flying moths," *Science*, vol. 184, no. 4140, pp. 999–1001, 1974.

[135] J. Kennedy, "Zigzagging and casting as a programmed response to wind-borne odour: a review," *Physiological Entomology*, vol. 8, no. 2, pp. 109–120, 1983.

[136] T. Baker, "Upwind flight and casting flight: complementary phasic and tonic systems used for location of sex pheromone sources by male moth," in *Proc. 10th Int. Symp. Olfaction and Taste, Oslo, 1990*, pp. 18–25, 1990.

[137] K. Leitch, F. Ponce, F. van Breugel, and M. H. Dickinson, "The long-distance flight behavior of drosophila suggests a general model for wind-assisted dispersal in insects," *bioRxiv*, 2020.

[138] R. Pang, B. J. Lansdell, and A. L. Fairhall, "Dimensionality reduction in neuroscience," *Current Biology*, vol. 26, no. 14, pp. R656–R660, 2016.

[139] J. P. Cunningham and M. Y. Byron, "Dimensionality reduction for large-scale neural recordings," *Nature neuroscience*, vol. 17, no. 11, pp. 1500–1509, 2014.

[140] S. Recanatesi, G. K. Ocker, M. A. Buice, and E. Shea-Brown, "Dimensionality in recurrent spiking networks: global trends in activity and local origins in connectivity," *PLoS computational biology*, vol. 15, no. 7, p. e1006446, 2019.

[141] K. Pfeiffer and U. Homberg, "Organization and functional roles of the central complex in the insect brain," *Annual review of entomology*, vol. 59, pp. 165–184, 2014.

[142] J. D. Seelig and V. Jayaraman, "Neural dynamics for landmark orientation and angular path integration," *Nature*, vol. 521, no. 7551, pp. 186–191, 2015.

[143] J. Green, A. Adachi, K. K. Shah, J. D. Hirokawa, P. S. Magani, and G. Maimon, "A neural circuit architecture for angular integration in drosophila," *Nature*, vol. 546, no. 7656, pp. 101–106, 2017.

[144] S. S. Kim, A. M. Hermundstad, S. Romani, L. Abbott, and V. Jayaraman, "Generation of stable heading representations in diverse visual scenes," *Nature*, vol. 576, no. 7785, pp. 126–131, 2019.

[145] T. S. Okubo, P. Patella, I. D'Alessandro, and R. I. Wilson, "A neural network for wind-guided compass navigation," *Neuron*, 2020.

[146] M. Demir, N. Kadakia, H. D. Anderson, D. A. Clark, and T. Emonet, "Walking drosophila navigate complex plumes using stochastic decisions biased by the timing of odor encounters," *bioRxiv*, 2020.

[147] A. Celani, "Olfactory navigation: Tempo is the key," *Elife*, vol. 9, p. e63385, 2020.

[148] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "OpenAI gym," *arXiv preprint arXiv:1606.01540*, 2016.

[149] A. Hill, A. Raffin, M. Ernestus, A. Gleave, A. Kanervisto, R. Traore, P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, and Y. Wu, "Stable baselines." https://github.com/hill-a/stable-baselines, 2018.

[150] I. Kostrikov, "Pytorch implementations of reinforcement learning algorithms," 2021.

[151] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

[152] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.

[153] K. Volkova, M. A. Lebedev, A. Kaplan, and A. Ossadtchi, "Decoding movement from electrocorticographic activity: A review," *Frontiers in Neuroinformatics*, vol. 13, 2019.

[154] H. Yuan and B. He, "Brain–computer interfaces using sensorimotor rhythms: Current state and future perspectives," *IEEE Transactions on Biomedical Engineering*, vol. 61, pp. 1425–1435, 2014.

[155] W. Tam, T. Wu, Q. Zhao, E. Keefer, and Z. Yang, "Human motor decoding from neural signals: a review," *BMC Biomedical Engineering*, vol. 1, no. 1, p. 22, 2019.

[156] S. Gupta and A. Gomez-Marin, "A context-free grammar for caenorhabditis elegans behavior," *BioRxiv*, p. 708891, 2019.

[157] K. Luxem, F. Fuhrmann, J. Kuersch, S. Remy, and P. Bauer, "Identifying behavioral structure from deep variational embeddings of animal motion," *BioRxiv*, 2020.

[158] D. Summers-Stay, C. L. Teo, Y. Yang, C. Fermüller, and Y. Aloimonos, "Using a minimal action grammar for activity understanding in the real world," *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4104–4111, 2012.

[159] Y. Yang, A. Guha, C. Fermüller, Y. Aloimonos, and A. V. Williams, "A cognitive system for understanding human manipulation actions," in *Advances in Cognitive Systems*, pp. 67–86, 2014.

[160] D. Shao, Y. Zhao, B. Dai, and D. Lin, "Finegym: A hierarchical video dataset for fine-grained action understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2616–2625, 2020.

[161] A. Piergiovanni and M. S. Ryoo, "Fine-grained activity recognition in baseball videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1740–1748, 2018.

[162] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele, "A database for fine grained activity detection of cooking activities," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1194–1201, IEEE, 2012.

[163] T. Kobayashi, Y. Aoki, S. Shimizu, K. Kusano, and S. Okumura, "Fine-grained action recognition in assembly work scenes by drawing attention to the hands," in *2019 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pp. 440–446, IEEE, 2019.

[164] L. Hansen, M. Siebert, J. Diesel, and M. P. Heinrich, "Fusing information from multiple 2D depth cameras for 3D human pose estimation in the operating room," *International Journal of Computer Assisted Radiology and Surgery*, vol. 14, no. 11, pp. 1871–1879, 2019.

[165] N. Sarafianos, B. Boteanu, B. Ionescu, and I. A. Kakadiaris, "3D human pose estimation: A review of the literature and analysis of covariates," *Computer Vision and Image Understanding*, vol. 152, pp. 1–20, 2016.

[166] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7291–7299, 2017.

[167] U. Hasson, S. A. Nastase, and A. Goldstein, "Direct fit to nature: An evolutionary perspective on biological and artificial neural networks," *Neuron*, vol. 105, no. 3, pp. 416–434, 2020.

[168] D. Hassabis, D. Kumaran, C. Summerfield, and M. Botvinick, "Neuroscience-inspired artificial intelligence," *Neuron*, vol. 95, no. 2, pp. 245–258, 2017.

[169] M. Kwon, S. Daptardar, P. Schrater, and X. Pitkow, "Inverse rational control with partially observable continuous nonlinear dynamics," *arXiv preprint arXiv:2009.12576*, 2020.

[170] Z. Ashwood, N. A. Roy, J. H. Bak, and J. W. Pillow, "Inferring learning rules from

animal decision-making," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[171] V. Vouloutsi, L. L. LopezSerrano, Z. Mathews, A. E. Chimeno, A. Ziyatdinov, A. P. i Lluna, S. B. i Badia, and P. F. J. Verschure, "The synthetic moth: a neuromorphic approach toward artificial olfaction in robots," *Neuromorphic Olfaction*, pp. 117–152, 2013.

[172] M. J. Anderson, J. G. Sullivan, J. L. Talley, K. M. Brink, S. B. Fuller, and T. L. Daniel, "The "smellicopter," a bio-hybrid odor localizing nano air vehicle," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6077–6082, IEEE, 2019.

[173] A. M. Zador, "A critique of pure learning and what artificial neural networks can learn from animal brains," *Nature communications*, vol. 10, no. 1, pp. 1–7, 2019.

[174] K. Gerlei, "Extracellular recording," Sept. 2021.

[175] S. Jia, T. Zhang, X. Cheng, H. Liu, and B. Xu, "Neuronal-plasticity and reward-propagation improved recurrent spiking neural networks," *Frontiers in Neuroscience*, vol. 15, p. 205, 2021.

[176] M. Yuan, X. Wu, R. Yan, and H. Tang, "Reinforcement learning in spiking neural networks with stochastic and deterministic synapses," *Neural computation*, vol. 31, no. 12, pp. 2368–2389, 2019.

[177] C. Frenkel, D. Bol, and G. Indiveri, "Bottom-up and top-down neural processing systems design: Neuromorphic intelligence as the convergence of natural and artificial intelligence," *arXiv preprint arXiv:2106.01288*, 2021.

[178] L. K. Scheffer, C. S. Xu, M. Januszewski, Z. Lu, S.-y. Takemura, K. J. Hayworth, G. B. Huang, K. Shinomiya, J. Maitlin-Shepard, S. Berg, *et al.*, "A connectome and analysis of the adult drosophila central brain," *Elife*, vol. 9, p. e57443, 2020.

[179] S. B. Wolff and B. P. Ölveczky, "The promise and perils of causal circuit manipulations," *Current opinion in neurobiology*, vol. 49, pp. 84–94, 2018.

[180] A. R. Vaidya, M. S. Pujara, M. Petrides, E. A. Murray, and L. K. Fellows, "Lesion studies in contemporary neuroscience," *Trends in cognitive sciences*, vol. 23, no. 8, pp. 653–671, 2019.

[181] I. E. Marinescu, P. N. Lawlor, and K. P. Kording, "Quasi-experimental causality in neuroscience and behavioural research," *Nature human behaviour*, vol. 2, no. 12, pp. 891–898, 2018.

[182] V. Mante, D. Sussillo, K. V. Shenoy, and W. T. Newsome, "Context-dependent computation by recurrent dynamics in prefrontal cortex," *nature*, vol. 503, no. 7474, pp. 78–84, 2013.

[183] A. J. Kell, D. L. Yamins, E. N. Shook, S. V. Norman-Haignere, and J. H. McDermott, "A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy," *Neuron*, vol. 98, no. 3, pp. 630–644, 2018.

[184] M. Crawshaw, "Multi-task learning with deep neural networks: A survey," *arXiv preprint arXiv:2009.09796*, 2020.

[185] C. D. Marton, G. Lajoie, and K. Rajan, "Efficient and robust multi-task learning in the brain with modular task primitives," *arXiv preprint arXiv:2105.14108*, 2021.

[186] L. Duncker, L. Driscoll, K. V. Shenoy, M. Sahani, and D. Sussillo, "Organizing recurrent network dynamics by task-computation to enable continual learning," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[187] W. F. Młynarski and A. M. Hermundstad, "Efficient and adaptive sensory codes," *Nature Neuroscience*, pp. 1–12, 2021.

[188] W. F. Młynarski and A. M. Hermundstad, "Adaptive coding for dynamic sensory inference," *Elife*, vol. 7, p. e32055, 2018.

[189] A. I. Weber and A. L. Fairhall, "The role of adaptation in neural coding," *Current opinion in neurobiology*, vol. 58, pp. 135–140, 2019.

[190] V. L. Ríos, P. G. Özdil, S. T. Ramalingasetty, J. Arreguit, S. C. Rosset, G. Knott, A. J. Ijspeert, and P. Ramdya, "Neuromechfly, a neuromechanical model of adult drosophila melanogaster," *bioRxiv*, 2021.

[191] M. Crosby, B. Beyret, and M. Halina, "The Animal-AI olympics," *Nature Machine Intelligence*, vol. 1, no. 5, pp. 257–257, 2019.

[192] M. Raghu, J. Gilmer, J. Yosinski, and J. Sohl-Dickstein, "Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability," in *NIPS*, 2017.

[193] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton, "Similarity of neural network representations revisited," in *International Conference on Machine Learning*, pp. 3519–3529, PMLR, 2019.

[194] A. Goulas, F. Damicelli, and C. C. Hilgetag, "Bio-instantiated recurrent neural networks: Integrating neurobiology-based network topology in artificial networks," *Neural Networks*, 2021.

[195] C. B. Delahunt, J. A. Riffell, and J. N. Kutz, "Biological mechanisms for learning: a computational model of olfactory learning in the manduca sexta moth, with applications to neural nets," *Frontiers in computational neuroscience*, vol. 12, p. 102, 2018.

[196] D. B. Ehrlich, J. T. Stone, D. Brandfonbrener, A. Atanasov, and J. D. Murray, "Psychrnn: An accessible and flexible python package for training recurrent neural network models on cognitive tasks," *Eneuro*, vol. 8, no. 1, 2021.

[197] K. O. Stanley, J. Clune, J. Lehman, and R. Miikkulainen, "Designing neural networks through neuroevolution," *Nature Machine Intelligence*, vol. 1, no. 1, pp. 24–35, 2019.

[198] A. Gupta, S. Savarese, S. Ganguli, and L. Fei-Fei, "Embodied intelligence via learning and evolution," *arXiv preprint arXiv:2102.02202*, 2021.