

UCSF

UC San Francisco Electronic Theses and Dissertations

Title

INTERNALIZED DISTRIBUTED MOTOR COMMANDS BRIDGE ACTION AND COGNITION IN C. ELEGANS

Permalink

<https://escholarship.org/uc/item/9544p3hx>

Author

Dunn, Raymond

Publication Date

2025

Peer reviewed|Thesis/dissertation

Internalized Distributed Motor Commands Bridge Action and Cognition in C. elegans

by
Raymond Dunn

DISSERTATION

Submitted in partial satisfaction of the requirements for degree of
DOCTOR OF PHILOSOPHY

in

Neuroscience

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, SAN FRANCISCO

Approved:

DocuSigned by:

Torsten Wittmann

Torsten Wittmann

58C53353ED5C44E...

Chair

DocuSigned by:

Karunesh Ganguly

Karunesh Ganguly

DocuSigned by:

Noelle L'Etoile

Noelle L'Etoile

DocuSigned by:

Saul Kato

Saul Kato

90E64C3F1F734D4...

Committee Members

ACKNOWLEDGEMENTS

I would like to express my deep gratitude for the relationships that have shaped me into the person I am today, both as a scientist and as a man. This gratitude is an expression of the love that fuels me, and keeps me warm on cold days. Graduate school has been immersive and intense, pushing me to think deep and broad. Beyond research, it has been tied to my personal growth. These pursuits helped me find my golden thread, and I will never, ever let it go.

First, I want to acknowledge the scientists who trained me before graduate school. They did an incredible job, as I have been both capable and rigorous in my work. Now, after twelve years of research, I am beginning to understand the time and effort they put into teaching me. Thank you, Maria, Dan, Patrick, Tiago, Michael, and Bridget.

Next, thank you to Gary L., Ted G, Matt W., Mike G., and Luke A. You helped shape my character, which has directly and indirectly substantiated any of my successes.

To my lab mates—especially Hannah G., Jackson B., Laura P., Trang D., and Yutaka S.—your passions and our fun chats fueled my own throughout grad school.

To Noelle—your unwavering passion for science and community in difficult times has been a beacon for me. You helped me balance immersion in research with the pull of reclusion and taught me how to wield my competitiveness for the benefit of those I love.

To my community of friends has enriched my life beyond measure, I can trace so much of who I am to the joy, laughter, and struggles we have shared. To my childhood friends—Brenton, Brian, D’vonne, Mitch, Sam, and Tyler—and to my SF friends—Iris, Josh, James, Danny, Julianne, Lucas, Brennan, Deanna, Tim, John, and Francisco—thank you.

To Saul—when I entered graduate school, I made the decision immerse myself in your intellectual and scientific vision; what questions you find interesting and how you solve problems. You won my loyalty with your immense generosity and by recognizing that I was a sensitive man motivated by personal connection and a nebulous curiosity about the world in which we live. This investment we made in each other has returned orders of magnitude for me in both science and in my personal relationships. You are a cornerstone of what has made my graduate school experience better than I could have dreamed.

To my dad, I love you, I miss you, and I know you would be proud. To Bo, thank you for being my sister, friend, and mentor. To mom—thank you for raising me into the man I am. Your parenting empowered me to my achievements and comforted me in my failures. Finally, to Caitriona—five years ago you entered my life, and together we created a new world. You give me meaning, and I love you to the moon and back.

“The earliest lineal ancestors to today’s [AI] Core personalities were not projects to create artificial intelligence, but incidental efforts to simulate artificial life [...] Hyperlife—self-reproducing, information-storing, interacting, metabolizing, evolving [...] It escaped the tide pools of the individual machines, moving into the embryonic planetary datasphere that they called the Internet.”

Dan Simmons, *The Rise of Endymion*

“Ten million years, Earthman, can you conceive of that kind of time span? A galactic civilization could grow from a single worm five times over in that time.”

Douglas Adams, *Hitchhiker’s Guide to the Galaxy*

INTERNALIZED DISTRIBUTED MOTOR COMMANDS BRIDGE ACTION AND COGNITION IN *C. ELEGANS*

Raymond Lake Dunn

ABSTRACT

Here we find that the nematode *C. elegans* makes goal-oriented turns while foraging and can maintain a working memory of sensory activation prior to the execution of a turn. This information is integrated with body posture to localize appetitive stimuli. We construct a virtual-reality whole-brain imaging and neural perturbation system, and with it we find that this working memory is implemented by the coupled oscillation of two distributed neural motor command complexes. One complex decouples from motor output after sensory evidence accumulation and initiates turn execution. The second gates this process with a reliable time constant. We propose that the function of working memory via internalization of motor oscillations could represent the evolutionary origin of internal neural processing i.e. thought, and a foundation of higher cognition.

TABLE OF CONTENTS

CHAPTER 1: WORKING MEMORY IN <i>C. ELEGANS</i> EMERGES FROM INTERNALIZED DISTRIBUTED MOTOR COMMANDS	1
1.1 INTRODUCTION.....	1
1.2 TOWARDS UNDERSTANDING THE FOUNDATIONS OF COGNITION.....	2
1.2.1 <i>C. elegans</i> executes directed, sensory-guided reorientations.....	4
1.2.2 Distributed dynamical complexes encode motor state and internalized head curvature.....	5
1.2.3 Headswing dynamics scaffold olfactory processing.....	10
1.2.4 Headswing oscillation phase is reset to control turn direction	14
1.2.5 Command state subcomplex is a reference clock for headswing.....	18
1.3 DISCUSSION	25
1.3.1 Caveats and interpretations.....	27
1.4 MATERIALS AND METHODS	29
1.4.1 <i>C. elegans</i> strains.....	29
1.4.2 Behavior imaging and quantification.....	30
1.4.3 Whole-brain Ca ²⁺ imaging of <i>C. elegans</i>	31
1.4.4 Region of interest (ROI) detection in volumetric Ca ²⁺ imaging data.....	32
1.4.5 Identification of neuron genetic identities	33
1.4.6 Neural timeseries derivatives and embeddings	34
1.4.7 Linear predictive modeling framework	35

1.4.8	Statistical tests	35
1.4.9	Command state ramping and phase analysis	36
CHAPTER 2: CONTRIBUTIONS ENABLING DATA STANDARDIZATION		
AND CLOSED-LOOP EXPERIMENTATION.....		37
2.1	ON THE DEVELOPMENT OF TECHNOLOGIES FOR PROBING BRAIN FUNCTION	37
2.2	A REAL-TIME CLOSED-LOOP EXPERIMENTAL FRAMEWORK	41
2.2.1	Problem statement	42
2.2.2	Proposed software solutions	43
2.2.3	CLEF technical architecture.....	46
2.2.4	Implementation case study: wb-live	50
2.2.5	Example 1: Perturbational studies of neural dynamics	57
2.2.6	Example 2: Next-gen brain neuromodulation.....	59
2.2.7	Example 3: Machine learning for brain-machine interfaces.....	59
2.2.8	Conclusion	60
2.3	UNIFYING COMMUNITY WHOLE-BRAIN IMAGING DATASETS ENABLES ROBUST NEURON	
	IDENTIFICATION AND REVEALS DETERMINANTS OF NEURON POSITION IN C. ELEGANS.....	61
2.3.1	Introduction.....	61
2.3.2	Results.....	65
2.3.3	Discussion	85
2.3.4	Methods.....	89
2.4	PYCRO-MANAGER: OPEN-SOURCE SOFTWARE FOR CUSTOMIZED AND	
	REPRODUCIBLE MICROSCOPE CONTROL	106

2.5	MINIMO: A LINKED DATA AND METADATA STORAGE SYSTEM FOR SMALL LABS	112
CONCLUSION		115
REFERENCES		117

LIST OF FIGURES

Figure 1.2.1: Worms make directed turns in a virtual olfactory environment.....	9
Figure 1.2.2: Execution and prediction of sensory-guided turns in paralyzed worms.....	13
Figure 1.2.3: Sensory memory is established by phase-embedding.	17
Figure 1.2.4: Neuron stimulation and TURN prediction reveals coordination of command and headswing subcomplexes.	24
Figure 2.2.1: The closed-loop versus open-loop experimental paradigm.	42
Figure 2.2.2: CLEF architecture diagram	44
Figure 2.2.3: CLEF GUI prototype.....	46
Figure 2.2.4: The <i>wb-live</i> launcher app, complete with lab branding.....	51
Figure 2.2.5: The <i>wb-live</i> Brainalyzer GUI in closed-loop neural imaging mode.....	53
Figure 2.2.6: The <i>wb-live</i> Brainalyzer GUI in closed-loop behavior mode.....	55
Figure 2.2.7: Pulse-triggered averages from stimulating various neurons with optogenetics	58
Figure 2.3.1: Example NWB file contents	66
Figure 2.3.2: Schema for <i>C. elegans</i> extension to the NWB architecture.....	68
Figure 2.3.3: Two software GUI applications with NWB I/O	69
Figure 2.3.4: Harmonized multi-lab atlas of <i>C. elegans</i> neurons.....	74
Figure 2.3.5: Analyses of neuron positions, distances, and positional variability	76
Figure 2.3.6: Improvements in neural identification accuracy.....	81

Figure 2.4.1: Pycro-Manager design (via Henry Pinkard)	109
Figure 2.5.1: <i>minimo</i> architecture.....	113

LIST OF TABLES

Table 1.4.1: List of <i>C. elegans</i> strains used in this study.	29
Table 2.3.1: Summary of aggregated dataset characteristics.	70

CHAPTER 1: WORKING MEMORY IN *C. ELEGANS* EMERGES FROM INTERNALIZED DISTRIBUTED MOTOR COMMANDS

1.1 Introduction

What was the first thought a nervous system ever had? How do you measure cognition? How do we achieve a deeper level of understanding of the brain? How does a brain *work*? These are examples of questions which I have had the privilege of probing during graduate school. These questions have no reasonable answers but taking them seriously has motivated and inspired the past several years of the life and, by extension, this dissertation. My outlook on biology, human intelligence, and how to advance the bleeding edge of knowledge is forever tied to the freedom I have felt over the past several years, by allowing my curiosity to guide me week after week.

In this dissertation I collect my contributions to science that I've made during graduate school. One key insight I have learned in my scientific training is just how satisfying it can be to sit on a question for long periods of time, to wrestle with it from different angles and positions, to see how new intuition grows out of dozens of seemingly insignificant shifts in perspective or approach. Dozens of times over the past several years, I've felt like I had a satisfactory solution

to a problem yet still found deeper and more clear insight and meaning by continuing to wrestle. This has also been my greatest joy in graduate school.

This chapter continues with my principal manuscript from my graduate school research. It encapsulates experiments, theory, and technology which I conducted and developed to advance our understanding of how the brain of the small roundworm *Caenorhabditis elegans* produces behavior. Specifically, this research marks the first report of working memory in worms. Moreover, the working memory appears to be implemented by the phasic interaction of two oscillators for motor control. I speculate that this internalization of motor output for cognitive function represents a foundational step forward in the evolution of cognition.

In the next chapter, I detail four notable methods contributions I made to the research community; two open source software packages for closed-loop real-time microscopy control, one software package for a data management system, and one standardized dataset for training machine learning algorithms.

1.2 Towards Understanding the Foundations of Cognition

Any animal navigating a complex environment stands to benefit from the ability to quickly form impressions of the world, retain them internally, and act on them later. Internal memory can be leveraged for deferred, contingent actions, separating cognition from reflex. But how did nervous systems acquire this foundational ability?

Distributed brain oscillations and their interactions have long been hypothesized to be a fundamental building block of neural computation and cognitive function in complex-brained animals (Buzsaki, 2006). Considerable theoretical work has been devoted to understanding how these observed phenomena might implement various cognitive functions (Engel, 2001). However, experimental establishment of causal roles and mechanistic understanding have been elusive, due at least in part to the sheer complexity of mammalian brain networks as well as the challenge of observing these networks at sufficient sampling density.

By contrast, in simpler brained animals with vastly lower neuron counts, oscillators serving to produce repetitive bodily movement, i.e. central pattern generators, have been closely studied and systematically dissected, often yielding deep mechanistic insight in the production of adaptive rhythmic motor patterns (Harris-Warrick, 1992). But cognitive functions such as sensory-driven decision-making have rarely been studied in simple animals (Briggman, 2005).

The 1mm long nematode *C. elegans* crawls on its right or left side when on a flat surface. During foraging, the worm crawls in a straight or curved forward direction and punctuates bouts of forward crawling with discrete reversal-then-turn reorientation maneuvers, known as pirouettes, either in the dorsal or ventral direction (**Fig. 1a**). The choice of this direction has been traditionally modeled as an unregulated random process, akin to the randomizing tumbles of a bacterium performing biased random-walk chemotaxis (Pierce, 1999), however worm olfactory neurons signal on surprisingly fast timescales (i.e. hundreds of milliseconds, Kato et al., 2014), and mutations which slow temporal dynamics of olfactory receptors drastically disrupt

chemotaxis, suggesting worms integrate sensory information with instantaneous body posture to execute the appropriate behavioral sequence. If true, timed activation of sensory neurons with respect to instantaneous head curvature would mimic naturalistic sensory activation, but with exquisite temporal control.

Here, we construct a closed-loop virtual olfactory environment (**Fig. 1a,b**), and show that worms integrate olfactory sensory dynamics with instantaneous body posture to navigate towards attractants. Fascinatingly, by withholding sensory activation during reorientation, we unambiguously demonstrate that these directed turns depend on a working memory which spans tens of seconds. We then turn to elucidating the mechanism of action of this working memory, which we find is implemented by the phasic interaction of two distributed dynamical complexes associated with motor control. We contend that the internalization of motor commands to direct deferred contingent actions might have been a crucial evolutionary step toward cognition.

1.2.1 *C. elegans* executes directed, sensory-guided reorientations

To evaluate whether an asymmetric sensory experience can drive directional turning in worms, we constructed a closed-loop virtual odor environment, using optogenetic stimulation to simulate an asymmetric odor signal from the worm’s frame of reference. In the absence of explicit sensory cues, such as crawling around on the surface of an agar plate without food, worm post-reversal turns are biased 60-75% towards their ventral side (Croll, 1975). We repeatedly stimulated the

AWA sensory neuron, known to mediate chemotaxis to certain attractants such as diacetyl via pulsatile activity responses (Itskovits et al., 2018), triggered on either ventral or dorsal head-swings during forward motion (**Fig. 1a**). Animals showed a strong preference for resolving the reversal with a turn in the favored direction as they resumed forward locomotion (**Fig. 1b**).

This observation suggests two key features of the sensory control of the behavior. First, to assign the stimulus to a particular direction, the animal needs to integrate olfactory sensation with proprioception or motor commands for head swings. Second, to act appropriately after the reversal, the animal needs some form of working memory (**Fig. 1c**).

1.2.2 Distributed dynamical complexes encode motor state and internalized head curvature

Previous studies have shown that whole-brain dynamics embed high level motor state commands in worms, zebrafish, mice, and humans. In the case of worms, the command state that encapsulates forward crawling (FWD) versus reverse crawling (REV) is broadly distributed throughout about 30% of the neural population. We reasoned that if FWD or REV motor commands are encoded in broadly distributed dynamics, a command state encoding head swings may also be present in a set of neurons.

To look for distributed encoding of head curvature, we restrained worms in microfluidic chips for volumetric whole-brain calcium imaging at cellular resolution (**Fig. 1d,e**). In our

immobilized whole-brain imaging setup, in all trials, we observed cycles of the fictive command state sequence (forward-backward-turn) widely distributed across neurons. To distinguish between the observed behavior of the worm and the fictive command-state cycles which have been previously ascribed to neural dynamics in restrained worms (Kato et al., 2015, **Fig. 1a**), we abbreviate each state: forward (FWD), reversal (REV), and dorsal/ventral (DOR/VEN) turn (TURN). As reported previously, the first principal component of the whole brain recording reliably reveals this command-state-reflecting dynamical process, consisting of cycles of forward and reversal periods, which we term the command state projection (Kato et al., 2015, **Fig. 1f,h**). During forward locomotion, head swings are especially pronounced. We therefore performed PCA on the residuals of $\text{proj}(\text{command})$, restricting to periods for fictive forward locomotion only, which yielded a second strong stereotyped network oscillation (**Fig. 1g,i**).

We found this faster oscillating component to be distributed across 6-10 neurons in the head of the animal that are variously implicated in motor control of the head muscles, sensorimotor integration, and proprioception (Gray et al., 2005; Yeon et al., 2018). The neuron class we observed with highest component loading was SMDV, and its contralateral neuron class SMDD had a large negative loading (**Fig. 1h**). Previous studies have shown SMDV reliably correlates with ventral head curvature during forward crawling and ventral post-reversal turns; SMDD correlates with dorsal turns (Hendricks et al., 2012; Kato et al., 2015). Here, their anti-phasic coupling during FWD suggests their activity corresponds to fictive head swings consistent

with sinusoidal crawling during the forward command state. Therefore we named this component the headswing projection.

During reverse locomotion, the head oscillations present during forward crawling are suppressed – the animal’s head trails behind the body (Alkema et al., 2005). During REV epochs in paralyzed worms, headswing dynamics continue to display oscillatory behavior; however, the magnitude is typically attenuated and the regularity of the oscillation period and waveform appears reduced (see example worm, **Fig. 1g**). We therefore refer to these dynamics as internalized because they persist in the absence of both sensory input (i.e. in paralyzed worms) and motor behavioral output (i.e. during reversal/REV). Toward the end of a reversal epoch, the magnitude of the headswing oscillation typically increases over the course of 1-2 oscillations, exhibited in SMDD/V as well as other neurons. Taken together, we surmise that the headswing oscillation decouples from motor output during reversals but does not disappear, and recouples leading up to a reversal termination in order to effect the post-reversal turn.

Previous studies have implicated dozens of neurons as “controlling reversal” or “controlling forward crawling” [many citations, may want to name neurons too]. These findings are consistent with the notion that the forward/reverse motor state is determined by a unitary bistable process emitted from a collective network state, like a coin flipping sides.

The command state and headswing and projections offer a window into two distinct dynamical processes with striking similarities. They are each spontaneous, oscillatory, and composed of many neurons with push-pull dynamics. Indeed in addition to having anticorrelated

dynamics, many forward-correlated neurons and reverse-correlated neurons directly inhibit one another (e.g. AVA/AVB); the same is true for dorsal-correlated and ventral-correlated neurons (e.g. SMDV/SMDD) (Witvliet et al., 2020).

We term the network structures which produce the dynamics captured in the command state and headswing projections as a dynamical subcomplex (**Fig. 1f**). These subcomplexes need not be strictly defined by the neuron identities participating in them; some neurons can and do come in or out of a shared dynamical motif or exhibit multiplexing (Kato et al., 2015 within command and headswing projections).

We next wondered whether command state and headswing subcomplexes may interact, so we evaluated whether their projections’ timescales may be coordinated (**Fig. 1k**). We measured the duration of paired FWD-REV epochs, expressed in terms of headswing peak-to-peak duration within the same worm. Surprisingly, we found increases in the fraction of command state durations at up to six integer multiples of the headswing timescale. This observation suggests the command state and headswing dynamical complexes interact; either one directs the other, or they influence one another (Kaplan et al., 2020).

Finally, neural trajectories in the $\text{proj}(\text{command})$ and $\text{proj}(\text{headswing})$ state space are more directly interpretable in the context of behavioral command state, and exhibit more symmetry compared to previous visualizations (Kato et al., 2015).

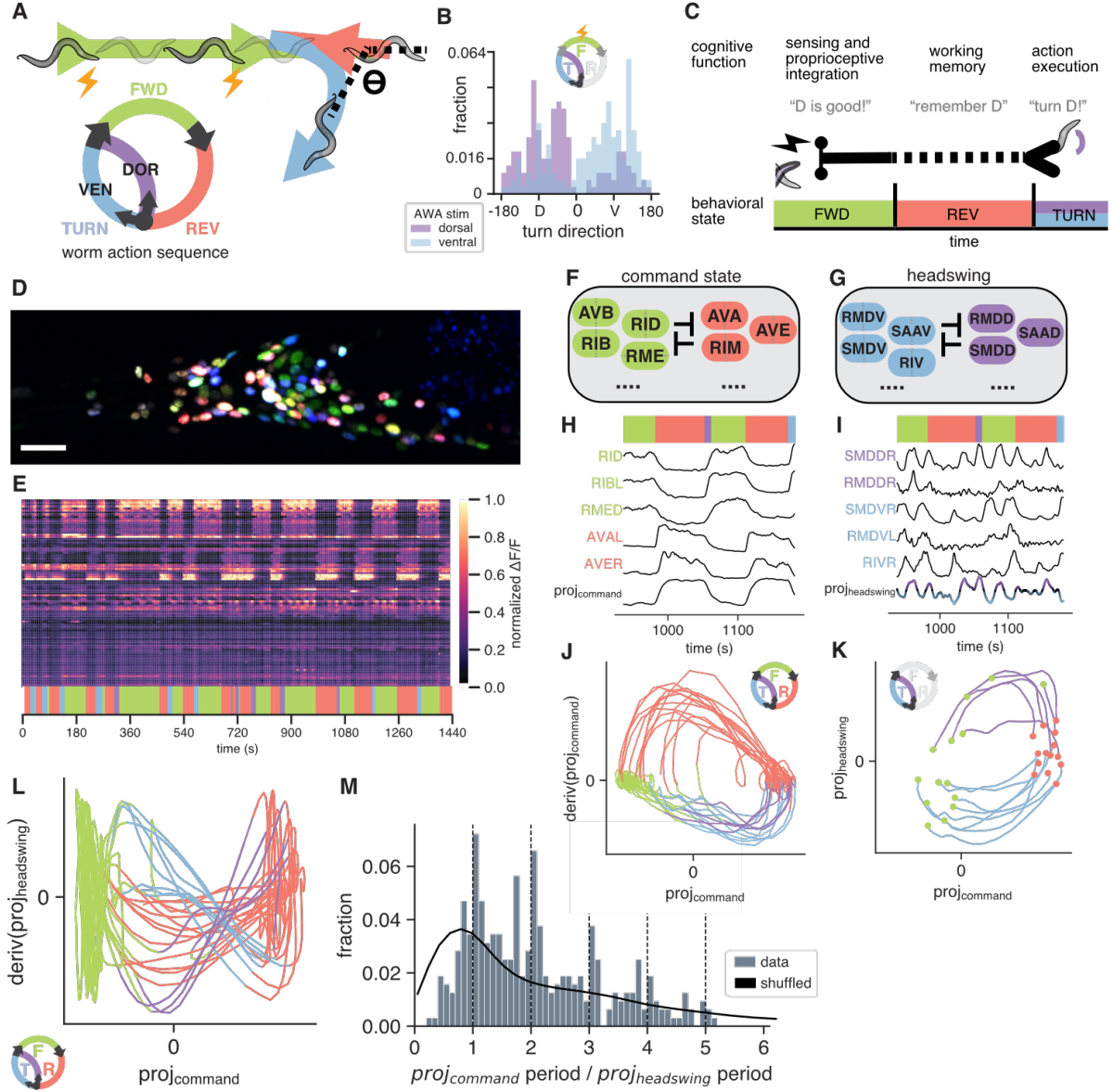


Figure 1.2.1: Worms make directed turns in a virtual olfactory environment.

(A) Schematic of closed-loop optogenetic stimulation of attractant sensory neuron with respect to instantaneous head curvature during forward crawling. [Inset] stereotypical motor command state sequence during foraging. (B) Angle of post-reversal turn following closed-loop stimulation, colored by whether stimulation was triggered by dorsal or ventral head bending during preceding bout of forward crawling. (C) Schema for deferred actions as a model for proto cognitive behavior. Memory of sensory experience is maintained across the reversal interval to turn execution. (D) NeuroPAL and panneuronal nuclear-localized GCaMP6s enable whole-brain imaging at cellular resolution with unambiguous neuron identification in restrained worms. Scale bar 20um. (E) Example heat plot of fluorescence ($\Delta F/F$) time series of 109 segmented head

neurons, one neuron per row. Top horizontal bars specify time intervals in (F, G). Bottom colored bar specifies inferred command state. **(F)** Connectivity diagram illustrating populations of push-pull neurons which underlie command state (FWD/REV) and head curvature (DOR/VEN). The neurons shown represent those which most reliably participate in these patterns of activity. **(G)** Example high-loading neurons in FWD/REV distributed command state. Neuron names are colored according to the state with which they positively correlate. Proj(FWD/REV) is revealed by the first principle component (PC1) of (E). **(H)** Example high-loading neurons in DOR/VEN distributed command state. Neuron names are colored according to the state with which they positively correlate. Proj(DOR/VEN) is revealed by restricting PCA to time-series epochs in the forward command state. **(I)** Example single worm trajectory embedded in proj(FWD/REV) phase space. Data is from (E). **(J)** Example trajectories of DOR/VEN TURN embedded in proj(FWD/REV) and proj(DOR/VEN) state space. Trajectories go from termination of preceding REV (red), until initiation of subsequent FWD (green) state. Data is from E. **(K)** Distribution of FWD/REV state duration, expressed in terms of contemporaneous DOR/VEN command state period. Data aggregated across 31 worms. **(L)** Example single worm trajectory embedded in FWD/REV and DOR/VEN state space.

1.2.3 Headswing dynamics scaffold olfactory processing

To test the hypothesis that sensory information that enters the system at a specific phase of the head swing during forward locomotion and is held in working memory to inform deferred behavioral choice, we combined whole-brain imaging with perturbative optogenetic stimulations during the FWD command state (**Fig. 2a-c**), recapitulating our virtual environment in paralyzed worms. Each 16-24 minute recording captured 10-30 forward-reversal-turn sequences.

Stimulation of AWA shifted the distribution of times until the next REV (**Fig. 2d**) compared to a control distribution computed for spontaneous REV transitions. Reversal command state transitions after AWA stimulation do not appear to be triggered immediately. Furthermore, the second peak in the distribution of time to initiation of REV (~30s) for AWA

stimulations roughly corresponds to one typical period of headswing oscillation (**Fig. 1h, 2d**). Therefore, our sensory stimulus triggers the onset of a fictive reversal not only in the near future, but also one oscillation cycle in the future, preserving the phase relationship between stimulation and internal representation of the proprioceptive state. Additionally, when paired with measurement of AWA activity at stimulation, the phase of headswing projection is predictive for turn direction on an individual trial basis (**Fig. 2i**), demonstrating that the headswing projection phase at the onset of fictive reversal is informative for prediction of turn choice but only following successful AWA stimulation. This suggests the possibility that the setting of the relative phase of headswing and command state phase may in fact be the mechanism for storing the memory of the head direction of a salient AWA stimulus.

We evaluated the TURN produced after the intervening REV state, about 30-90s in the future (**Fig. 2g**), and found that our stimulations bidirectionally biased TURN ratio; the headswing phase, estimated with peak-to-peak linear interpolation, at AWA stimulation stratified dorsal-promoting and ventral-promoting segments of headswing phase.

Next we constructed a decoding framework, trained on individual stimulation trials (see Methods) to predict TURN direction using activity prior to REV. We used balanced accuracy as our error metric because worms display an intrinsic bias for ventral turns. We found that the dorsal/ventral identity of fictive post-reversal turns could be predicted by AWA optogenetic stimulation when combined with phase information of headswing projection at the time of stimulation (**Fig. 2e**). Our predictability improved by including additional features such as

headswing state at REV onset (**Fig. 2j**), and including individual neuron features of headswing-participating-neurons alongside the headswing projection.

This confirms the existence of a functioning working memory in our immobilized setup, lasting up to tens of seconds during fictive reversal periods (**Fig. 2g**), as distributions of reversal times are typically longer in immobilization chips (Kato et al., 2015). The persistence of this working memory capability during immobilization and muscle paralysis demonstrates that the memory is carried by an internal state, rather than being embodied in body posture or muscle activation.

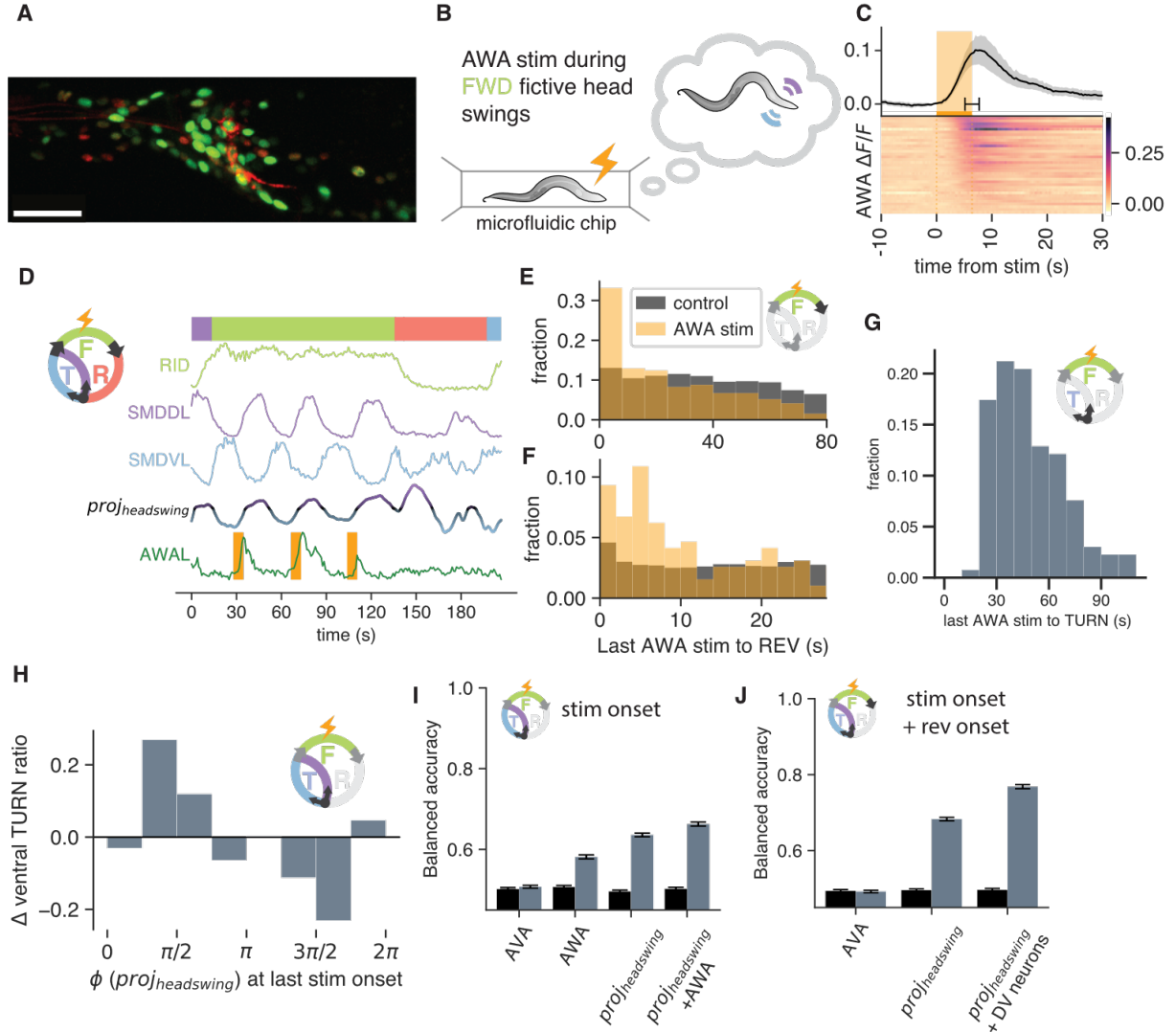


Figure 1.2.2: Execution and prediction of sensory-guided turns in paralyzed worms.

(A) Maximum intensity projection of panneuronal GCaMP with WOrMsChRmine expressed under AWA-specific promoter for acute, transient activation during whole-brain imaging. (B) Diagram of paralyzed worm under whole-brain volumetric calcium imaging with brain-state dependent AWA stimulation. (C) Heatmap of individual AWA 3-5s stimulations during whole-brain imaging. Each row represents a single AWA from a different worm. (D) Example trial of simultaneous whole-brain imaging and closed-loop stimulation in paralyzed worms, recapitulating our simulated virtual olfactory environment in Fig. 1A. Olfactory activation was repeatedly embedded within spontaneous head-swing dynamics during forward crawling. Upon reversal initiation, stimulation was then withheld. (E) Distribution of time after AWA stimulation until next fictive reversal (REV), compared to spontaneous state transition computational control. (F) Same as (E) but with different bin size and axis limits to reveal REV transition timescale. (G) Distribution of time after AWA stimulation until fictive dorsal or

ventral turn. **(H)** Change in ventral TURNs ratio, relative to unstimulated control and stratified by phase of DOR/VEN fictive head swing at the time of stimulation onset. Here phase is calculated by peak-to-trough interpolation of $\text{proj}(\text{DOR}/\text{VEN})$ in derivative space. **(I)** Barplots of balanced accuracy for turn choice (VEN or DOR) prediction using models fit to neural activity at stimulation onset of AVA, AWA, $\text{proj}(\text{DOR}/\text{VEN})$, or AWA+DOR/VEN. Error bars are SEM after cross validation. **(J)** Barplots of balanced accuracy for turn choice (VEN or DOR) prediction using models fit to neural activity at both stimulation onset and REV onset of AVA, $\text{proj}(\text{DOR}/\text{VEN})$ or $\text{proj}(\text{DOR}/\text{VEN})$ as well as individual features of neurons constituting the DOR/VEN subcomplex.

1.2.4 Headswing oscillation phase is reset to control turn direction

Does internalization of the headswing influence properties of its oscillation? Alternating activity of dorsal- and ventral-associated neurons during FWD and REV is strikingly sinusoidal (**Fig. 1**). Therefore we hypothesized that the sensory-proprioception memory may be maintained in the headswing oscillation phase. Sine waves are parameterized by amplitude, frequency, and phase. Therefore we next asked how these parameters of headswing dynamics are related between FWD and the subsequent REV.

We fit single sine waves to smoothened derivatives of the headswing projection as well as a set of high-loading neurons which participate in the headswing complex to compare oscillation amplitude, frequency, and phase between headswing oscillations during FWD and the subsequent REV epoch (**Fig. 3a**). We reasoned that fitting on derivatives of neural activity would be more sensitive because the signal is stationary, and because our functional fitting form was a sine wave, so its derivative could be easily interpreted. Furthermore we conducted these

analyses on trials with and without AWA stimulation, because we reasoned that this phase relationship should be a general property of worm motor sequence coordination.

We observed significant correlations between amplitude and frequency between headswing during FWD and REV (**Fig. 3b,c**), suggesting these properties are intrinsic to the network structure which produces these dynamics. On average the amplitude of the headswing oscillation is reduced during REV. Headswing phase during a FWD epoch may be related to the internalized headswing phase in the subsequent REV in one of three ways: phase is unchanged, kicked (i.e. a circular shift in phase), or reset (i.e. nonlinear discretization). Surprisingly, we found no obvious relationship between the headswing dynamics phase in coupled and uncoupled modes, ruling out these first two hypotheses.

We next wondered how stable and regular are oscillatory headswing dynamics when internalized during REV? Particularly during early REV, the headswing oscillation appears weak. We again fit sine waves to activity during REV, however this time we separately fit on the first and second halves of REV (**Fig. 3e**). We then compared fit parameters between headswing oscillations in early and late REV. We found small but significant correlations in amplitude, frequency, and phase between early and late REV, indicating that the internalized headswing oscillation is stable over the course of a fictive reversal (**Fig. 3f-h**).

We asked whether headswing oscillation fit parameters explain TURN direction. Fascinatingly, we observe that headswing subcomplex dynamics appear to fall into two discrete symmetric bins governed by future turn choice (**Fig. 3k**). This phase discretization explains the

lack of a detectable relationship between headswing phase between FWD and REV. Crucially the headswing phase in the FWD frame of reference also appears to trend this way, however the polar plot reveals that the peaks of each distribution are close (**Fig. 3j**). This observation provides an explanation for why we can predict TURN from headswing dynamics at reversal onset (**Fig. 2j**). This suggests there is some nonlinear operation which serves to discretize the headswing subcomplex into two initial states, separated by phase, which produces headswing dynamics for “future ventral” and “future dorsal” TURNS.

Therefore, we conclude that the phase of the headswing dynamical subcomplex is reset by some nonlinear discretization operation applied on headswing dynamics by REV onset. This phase resetting then sets up an internalized headswing oscillation that will give rise to a future dorsal or ventral turn.

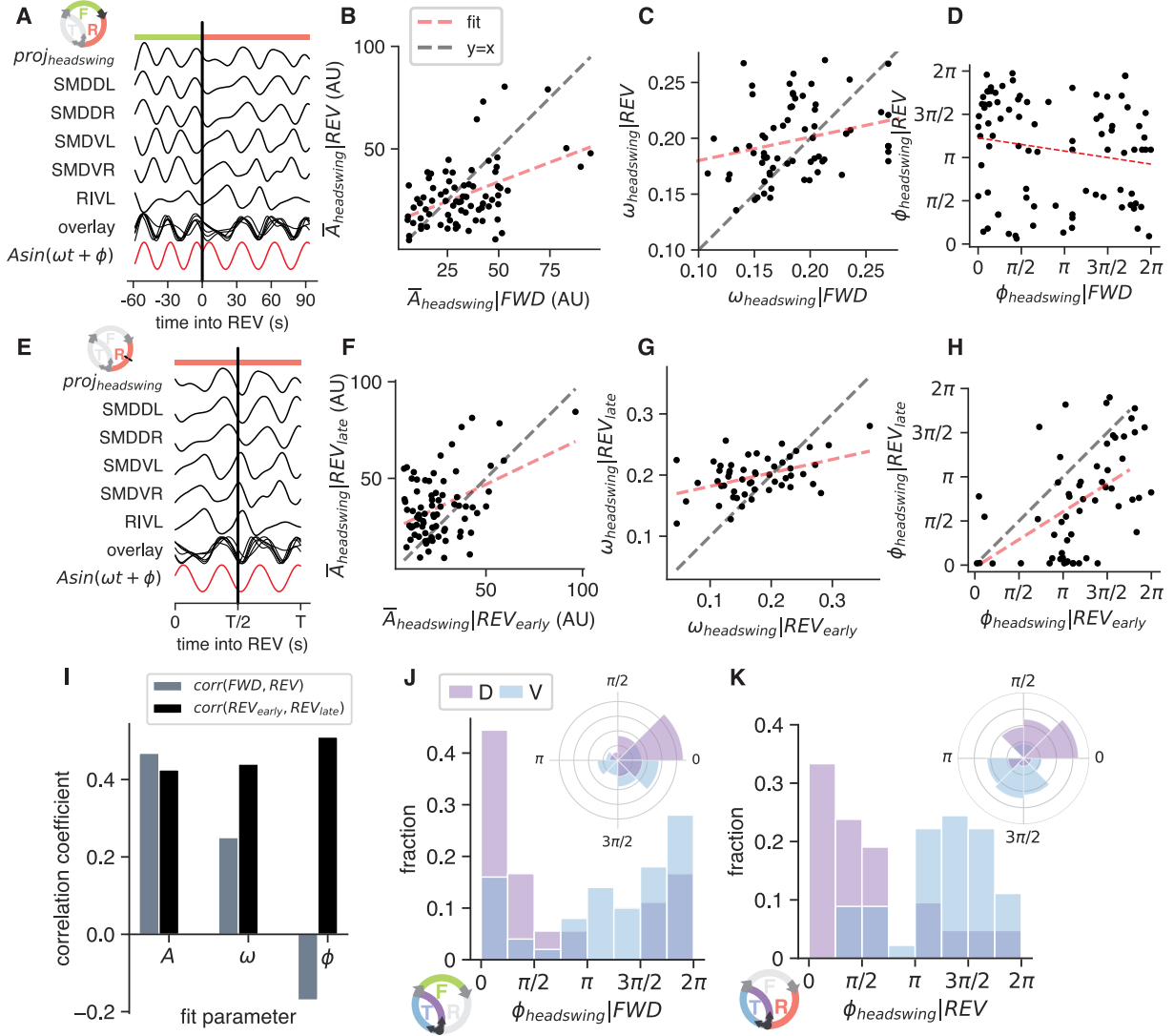


Figure 1.2.3: Sensory memory is established by phase-embedding.

(A) Schematic of functional form fitting. A single sine wave was fit to smoothed derivative of proj(DOR/VEN) as well as the individual neurons composing it, for each paired FWD and subsequent REV segment. By convention, the sign of dorsal-associated neuron loadings was fixed to be positive. Sines were fit on a continuous time coordinates for direct comparison of fit parameters. **(B-D)** Correlation between sine wave neuron loading (b), frequency (c) and phase (d) in FWD and corresponding REV segment. **(E)** Schematic for fitting sine wave separately to first (early) and second (late) halves of a REV segment. **(F-H)** Correlation between sine wave neuron loading, frequency, and phase between first and second halves of a REV segment. **(I)** Summary of correlation coefficients calculated in (b-d, f-h) **(J-L)** Sine wave phase in FWD and REV segments, colored by subsequent TURN direction.

1.2.5 Command state subcomplex is a reference clock for headswing

How does the animal know when to recouple the internalized headswing oscillation, producing the correct directed turn? To store a value (i.e. memory) reliably using the phase of an oscillator over several oscillation periods, we can envision two possible scenarios: (1) the oscillator must maintain a stable frequency and there must be another temporally stable process to function as a reference clock, such as a fixed-timescale ramp-to-threshold process or a slower oscillator with a period that is a near multiple of the first oscillator; (2) in the case of a variable oscillator, the oscillator must be coordinated dynamically with a process capable of maintaining a relative phase over oscillation cycles with jointly varying period.

Our evidence suggests that the internalized headswing oscillation has stable frequency (**Fig. 3i**), and is initialized differently depending on future TURN to be produced. Therefore we hypothesized that this “clock” temporal process would prevent REVs from ending too soon, and be visible in the command state projection. Furthermore, an alternative to the phase-memory hypothesis is that, while many neurons participate in the headswing oscillation, the decision may be ultimately reliant on a single neuron such as SMDV/SMDD activation.

Therefore, to better assess the causal role of particular neurons in holding a sensory memory or driving turn identity, we turned to selectively stimulating neurons present in the headswing projection during whole-brain imaging, during fictive reversals. We started by stimulating SMDD/V neuron classes given that they signal turn choice. The opposing phase of the SMDD and SMDV class neurons and role naturally suggested a hypothesis that the selective

activation of either opposing class leading up to reversal termination would influence the identity of the post-reversal turn. We used a digital micromirror device to spatially restrict our laser stimulation to selectively stimulate either neuron class, and we timed stimulations to occur during fictive reversals (**Fig. 4a,b**). Surprisingly, selective optogenetic stimulation of the ventral or dorsal class of neurons, e.g. SMD neurons, though demonstrably hyperpolarizing the neurons (**Fig. 4l**), did not affect the resulting turn identity (**Fig. 4h**; aggregate data of headswing neuron stimulation, see below). This suggests that the activity of SMD neurons, while clearly signaling the turn identity, are not the locus of the memory trace initially established upon sensory-proprioceptive integration. Interestingly, despite the inability to bias turn identity, stimulation of either SMD neuron class elicited immediate reversal terminations (**Fig. 4c,d**).

In search of other neurons that may encode future turn identity, we performed reversal-timed stimulations of other nonsensory head neurons RIA and RIV. Previous studies implicated RIA in sensory-proprioceptive integration; it exhibits compartmentalized calcium dynamics along the nrV and nrD regions of its nerve ring neurite, which tightly correlate with SMDV and SMDD activation during ventral and dorsal head bending, respectively (Hendricks et al. 2012). RIV has functional overlap with SMDV (Gray et al. 2005), however it is typically thought of as a motor neuron due to its synaptic connectivity (Witvliet et al. 2020). We found that spatially localized optogenetic stimulations of subcompartments of RIA, and as well as RIV, were capable of immediately terminating reversals upon stimulation (**Fig. 4e-g**). This substantiates the

distributed nature of the headswing oscillation, not just in the display of shared activity but in influence on behavior.

One interpretation of the ability of proprioceptive neurons to immediately terminate fictive reversals is that they signal the successful execution of physical bending that produces reverse movement. In addition to their role in reorientations during foraging, reverse crawling is used by *C. elegans* to escape predatory fungal traps and noxious stimuli (Maguire et al., 2011; Kaplan and Horvitz, 1993). Escape reversal bouts can be longer and may partially explain the increase in long REV epochs observed in the chip. We wondered whether a proprioceptive reverse termination signal would interfere with the putative role of the command state subcomplex as a reliable clock supporting phase-based memory.

Interestingly, the capability of RIA subcompartments to terminate fictive reversal was gated by the instantaneous activity of the command state projection at the time of stimulation on a per-trial basis (**Fig. 4i, j**), suggesting that some time-varying process during reversal command state dictates REV termination by headswing neurons. We found that the magnitude of the command projection was a predictor of successful reversal terminations initiated by RIA (nrV or nrD) stimulation. As this projection decays, the system becomes permissive for proprioception-driven reversal termination. Additionally, individual neuron predictivity was positively correlated with command state projection loading (**Fig. 4i**). This suggests that the decay in activity of positive loading neurons such as OLQ, URX, or the ramping of neurons with negative loading such as RME, could potentially function as timers, possibly in conjunction with

other sensory or motor roles previously implicated (Kaplan and Horvitz, 1993; Hart et al., 1995; Zimmer et al., 2009). Interestingly, the backward command neuron AVA, which has the largest command state loading of all neurons and is required for communication of the command state to the downstream body motor system that controls body posture (Kato et al., 2015; Pokala et al., 2014), was not a predictor of induced reversal terminations, suggesting it may not have a casual role in initiating reversal terminations. AVA’s relationship to command state subcomplex dynamics as an output to downstream motor systems or muscles may be analogous to SMD’s relationship to headswing subcomplex dynamics.

Focusing next on stimulations that forced an output from the emerging D/V decision process by terminating reversal, we found that the activity level of the headswing projection, or individual neurons with high headswing loadings (**Fig. 1g**), was predictive of the turn (**Fig. 4l,m**). We suspect the non-monotonic time-course of predictivity is a consequence of oscillatory dynamics in these neurons; the headswing dynamics captures both DOR and VEN depolarizations, whereas individual neurons participating in the headswing oscillation show turn-direction-dependent differential activity during half of the oscillation cycle, enabling better classification. The observation that both command-state-gating and D/V predictability diminish rapidly in models fit on timepoints prior to stimulation suggests that the timescale of command-state-gating and headswing recoupling may be coordinated.

We hypothesized that the sensory memory is stored as a change in the levels of certain neurons, either absolute or relative to other neuron level changes. However, using our decoding

framework, we were unable to find any neurons whose static or monotonic activity levels were predictive of turn identity. We then asked if the sensory memory is stored as a change in the amplitude of oscillating neurons. While we did observe shifting imbalances of oscillation amplitude in SMDV versus SMDD in some recordings, these imbalances were not predictive of the turn identity until roughly one oscillation just prior to the post-reversal turn (**Fig. 3j**), suggesting that the difference in oscillation amplitude of these neurons does not provide the working memory through the early period of reversal when headswing is attenuated. In any case, purely observational studies of neural activity do not discriminate causal activity from epiphenomenal “ride-along” activity.

To test the hypothesis of dynamical coordination between command and headswing subcomplexes, we first fit an exponential to the command projection during REV (**Fig. 4k**) and found that the time constant of command projection ramping is correlated with reversal duration, the timescale over which the memory must be maintained.

Combined with our earlier observation on the period headswing oscillation is embedded within the longer command state oscillation (**Fig. 1k**), we conclude that within a reversal epoch, the command state gates decoupling and recoupling of headswing from motor output during reversal, each of which aligned to headswing phase, producing a directed turn which is informed by prior sensory-priorioceptive sensory experience.

What dynamical system could support spontaneous oscillations like those we observe in the worm brain? Previous theoretical work has shown that spontaneous oscillations emerge from a simple biologically plausible dynamical model, the half-center oscillator (HCO, Skinner 1994), where two units mutually inhibit one another. This model is consistent with the observation that both command state and headswing subcomplexes are composed of anticorrelated neurons which mutually inhibit. Parameterized HCO dynamics are strikingly similar to both the command state and headswing projections (**Fig. 1,4**). Furthermore, incorporation of an interaction term between two HCOs produces state duration nesting analogous to Figure 1j (**Fig. 4n**). This observation raises an intriguing normative hypothesis, that the worm brain may be composed of multiple loosely-coupled dynamical subcomplexes (e.g. HCOs) whose coordination implements cognitive functions such as working memory and ultimately generates evolutionarily successful behavior.

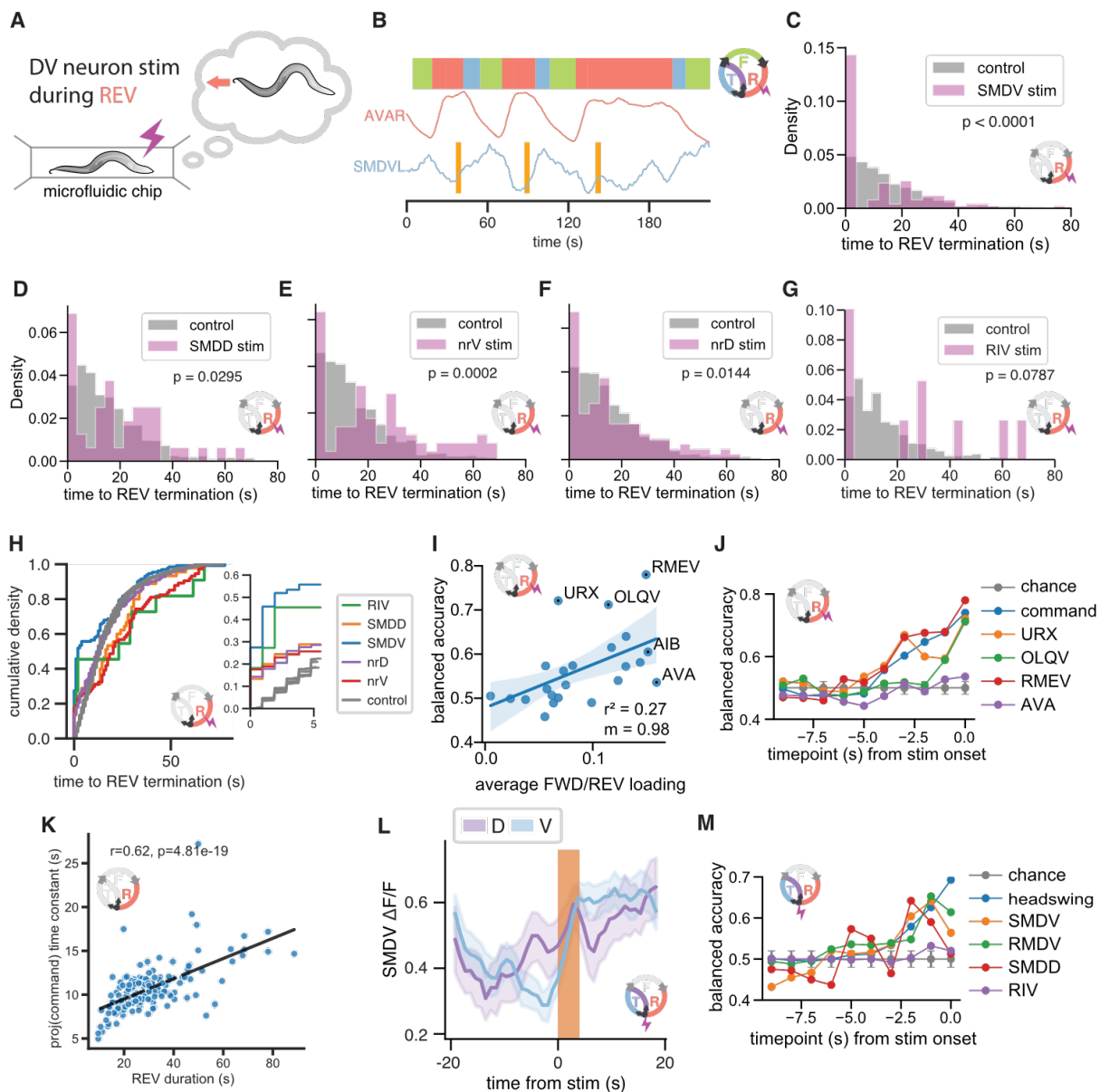


Figure 1.2.4: Neuron stimulation and TURN prediction reveals coordination of command and headswing subcomplexes.

(A) Diagram of paralyzed worm under whole-brain volumetric calcium imaging with brain-state dependent activation of neurons with high magnitude loading in DOR/VEN subcomplex. (B) Example trial of simultaneous whole brain imaging and single neuron optogenetics targeting SMDV timed to fictive reversals, denoted by AVA depolarization. (C-G) Distribution of time after stimulation until REV termination following stimulation of DOR/VEN neurons (SMDV/SMDD/RIV or nrV/nrD subcompartments of RIA). Computational control for spontaneous state transition is shown in grey. (H) Empirical cumulative distributions detailing the effect of stimulating various DOR/VEN neurons (SMDV/SMDD/RIV or nrV/nrD).

subcompartments of RIA) on time after stimulation until REV termination. Inset: expansion of timescale immediately after stimulation. **(I)** Balanced accuracy of decoding successful REV termination after activation of RIA nrV or nrD, by fitting a classifier on the activity of various neurons at the time of stimulation onset or prior to stimulation onset. **(J)** Predictability of REV termination correlates with $\text{proj}(\text{FWD}/\text{REV})$ loading. **(K)** The rise constant of forward command state correlates with the reversal duration ($r=0.62$). **(L)** Average evoked SMDV activity on trials where stimulation was immediately followed by dorsal (purple) or ventral (blue) TURN. Stimulation had no effect on future turn identity. Shaded regions indicated 95% CI. Note that given future D/V turn identity, SMDV activity has already diverged prior to stimulation. Following stimulation, SMDV shows prolonged and sustained depolarization for ventral but not dorsal TURNs, where depolarization is confined to light-on period. **(M)** Balanced accuracy of decoding TURN direction following activation of RIA nrV or nrD, by fitting a classifier on activity of various neurons at or before stimulation onset.

1.3 Discussion

In summary, worms are capable of producing directed turns by gathering sensory evidence and integrating it with instantaneous body posture, revising a longstanding notion that pirouette turning sequences are undirected. Our data suggest the worms navigate odor environments based on the following mechanism: (a) sensory input is received during crawling sinusoidal head swings, (b) AWA activation elicits a turn sequence, beginning with reversal initiation, (c) command state and headswing dynamical subcomplexes are nested, so this reversal initiation is embedded in the headswing oscillation, (d) headswing dynamics are internalized and phase is reset, discretized to the interval $[0, \pi]$ or $[\pi, 2\pi]$ depending on headswing embedding, (e) during REV, a ramping process in the command state dynamical subcomplex gates reversal termination, until (f) a rise in headswing signal indicates successful turning, terminating REV,

and finally (g) the correct directional turn is produced based on whether headswing subcomplex is in DOR- or VEN- associated phase at headswing recoupling.

Our experiments argue against the encoding of working memory in static neural activity levels for the sensory-guided decision-making capability we study here. They instead suggest that working memory is implemented by the maintenance of the relative phase of two coordinated dynamical subcomplexes distributed across many neurons. This study supports the notion that distributed dynamical signals are causal entities, enabling coordination of processes which manifest cognitive functions and drive single-trial consequences.

Of note, we find that the persistence of the headswing dynamical subcomplex oscillation during reversal, in the absence of proprioceptive sensory input or behavioral output, to be intriguing. This is the simplest example we are aware of where an animal uses dynamics in its nervous system to sustain information about the outside world, and act on those dynamics tens of seconds in the future to support goal-oriented decision making.

We surmise that the original function of the head neural oscillator was to produce physical movement, and the motor-decoupled mode of operation only arose later to support the function of working memory, endowing the animal the ability to perform deferred sensory-guided action selection. This evolutionary step, from embodied to internal neural oscillations, may represent the origin of a functional primitive of cognition subsequently recruited to do far more complex forms of thinking.

1.3.1 Caveats and interpretations

The accuracy of the prediction response to a past stimulus and the noise of the measurements of phase relationships and spread of state space trajectories imply the presence of uncontrolled variability. If neural responses were more reliable, perturbative experiments would be likely to be more effective and efficient in dissecting mechanisms. We might drive more reliable responses by exploring closed-loop stimulation parameters, and more comprehensive or precise control of the sensorium as well as the organismal state of the animal. Previous work has found that downstream neural responses are more reliable when AVA optogenetic stimulation is paired with optogenetic stimulation of AWC, another sensory neuron that plays a key role in attractant chemotaxis (Dobosiewicz et al., 2019).

While no neurons in our recordings appeared to explicitly encode of future dorsal/ventral turns by way of modulating their stable activity level or in their amplitude of fluctuations, we cannot rule out the existence of neurons that do not show activity via our pan-neuronal activity indicator, for technical reasons such as low genetic expression levels, possess sub-cellularly localized calcium fluctuations that do not reach the nuclear-localized calcium sensor, or having a intracellular calcium concentration outside the dynamical range of our sensor. We also cannot rule out the possibility that other non-neural activity internal signals, such as neuropeptide concentration, or the facilitation of a set of synapses, carry the dorsal/ventral sensory memory. However, given the evidence of a causal role of relative phase in our perturbative studies, we suggest that phase encoding carries the early memory trace for this working memory system.

We surmise that the original function of the head neural oscillator was to produce physical movement, and the motor-decoupled mode only arose later to support the function of working memory, endowing the animal the ability to perform deferred sensory-guided action selection. This evolutionary step, from embodied to internal neural oscillations, may represent the origin of a functional primitive of cognition subsequently recruited to do far more complex forms of thinking.

1.4 Materials and Methods

1.4.1 *C. elegans* strains

Table 1.4.1: List of *C. elegans* strains used in this study.

<u>Strain</u> <u>name</u>	<u>Experiment</u>	<u>Genotype</u>	<u>Construct</u>	<u>Ref</u>
OH16230	Control	otIs670 V; otIs672.	N/A	Yemini et al. 2021
FC121	Fig. 4	otIs670 V; otIs672.	PFC053 (flp-22p::Chrimson::sl2::mCherry @ 5ng/ul) + unc-122 RFP (50ng/ul)	This study
FC128	Fig. 4	otIs670 V; otIs672.	PFC079 (glr- 3p::WOrMsChRmine::wrmScarlet @ 5ng/ul) + unc-122 RFP (50ng/ul)	This study
FC134	Fig. 1b, Fig. 2	otIs670 V; otIs672.	PFC082 (odr- 10p::WOrMsChRmine::wrmScarlet @ 2.5ng/ul) + unc-122 RFP (50ng/ul)	This study

1.4.2 Behavior imaging and quantification

Worms were prepared 24 hours in advance of imaging: L4 worms were picked onto fresh NGM agar plates with an OP50 lawn. Immediately before imaging, worms were picked onto a fresh NGM agar plate without food, into a drop of M9. Single worms were aspirated out of that drop and washed 3x by aspirating into new drops of M9. Finally, the worm was aspirated onto an NGM agar plate without food, with care taken to re-aspirate excess M9. Worms were imaged while crawling on the surface of the agar without lid for 1-2 recordings of 12 minutes each. Data were acquired on a Leica M205 with TL5000 base. Images were captured on an ORCA-FLASH V3 set to 100 ms exposures. Image FOV was set to 4.78mm. Worms were tracked over the course of an experiment with the microscope’s motorized stage centered on the worm’s centroid.

For experiments involving optogenetics, care was taken to reduce activation of the optogenetic channel from brightfield illumination light. About 24 hours prior to imaging, 80ul of 100uM all-trans retinal in M9 was added to the bacteria lawn of L4s. Worms were then placed in foil at 20C until imaging. Microscope illumination was set to 2. During an imaging session, worms were partially covered if not being immediately prepared for an experiment. Optogenetic illumination light source was the Leica LED3 set to 100%.

1.4.3 Whole-brain Ca²⁺ imaging of *C. elegans*

Two-layer PDMS microfluidic devices were manufactured as described previously (Zimmer et al., 2009, Cáceres et al., 2012). Worms were prepared 24 hours in advance of imaging: L4 worms were picked on to fresh NGM agar plates with an OP50 lawn. Immediately prior to imaging, worms were picked onto fresh NGM agar plates without food, and placed in a drop of M9 solution with 5mM tetramisole. Worms were left in tetramisole for 10 minutes prior to being aspirated into the microfluidic channel. Worms were imaged for 16-24 minutes. Data were acquired on inverted spinning disk microscope (Yokogawa W1) equipped with a sCMOS camera (Telydyne Photometrics Kinnetix). About half of recordings were acquired at 2x2 camera binning. The microscope objective lens was 40x 1.25NA WI. Sample Volumetric scans were performed using a piezo stage with 10-12 Z-planes with z-spacing 2.5-3um. In select recordings, 4 Z-planes with z-spacing 3um was used to measure neurons in lateralized anterior, lateral, dorsal, and ventral ganglia at higher temporal resolution under equivalent optical conditions. Prior to acquiring videos for calcium timeseries, a reference high-resolution structural image was acquire: 40x1um Z planes under 4 different optical conditions to measure NeuroPAL fluorescence (see Yemini et al., 2021). Exposures ranged from 30ms-80ms under 100-200uW illumination to minimize photobleaching.

For experiments involving optogenetics, care was taken to reduce activation of the optogenetic channel via illumination for measuring GCaMP fluorescence. About 24 hours prior to imaging, 80ul of 1uM ATR in M9 was added to the bacterial lawn of L4s. Worms were then

placed in foil at 20C until imaging. During an imaging session, worms were partially covered if not being immediately prepared for an experiment. Worms were imaged under 70-140uW of light at the sample, for 20-30ms exposures within an 80ms duty cycle - 20ms-30ms of light exposure, 50-60ms of blank time. Optogenetic illumination light source was (89NorthLDI-7) at 640nm, powered to 10-20% of maximum illumination intensity. This illumination was directed by a digital micromirror device (DMD, Mightex Polygon P1000). Illumination was brightfield for worms with optogenetic construct expressed under a single-neuron promoter, and localized to about 5x5um for localized illumination of constructs driven by multi-neuronal promoters.

1.4.4 Region of interest (ROI) detection in volumetric Ca²⁺ imaging data

ROI detection from neural timeseries videos was adapted from Kato et al., 2012, implemented by the Napari (Chiu and Clack, 2022b) eats-worm plugin. Briefly, interframe motion was first registered using manual tracking (Schindelin et al., 2012). A reference ROI movie was then generated composed of each image plane by averaging successive blocks of 20-200 movie frames to reduce noise. Each frame of the reference was adaptively thresholded based on median image brightness, median filtered, then convolved with a gaussian kernel. Local maxima were found and merged if peaks were adjacent within a greedy threshold. For each ROI center, a surrounding region with radius 5-7 was defined, with overlapping adjacent regions excluded via Voronoi tessellation with area shrinkage of 0.5 pixels. ROIs in adject timepoints were linked via local greedy matching. Cells below detection threshold were extrapolated based on the motion of

neighboring ROIs. Finally, time-varying multi-plane ROIs were adjoined based on overlap. Each neuron was manually verified by R.L.D. For high temporal resolution half-brain imaging experiments, images were compressed along the Z axis via maximum intensity projection, becoming 3D ROIs. Neural time series extraction was adapted from Kato et al., 2015. Briefly, for each 4D ROI, a single-cell fluorescence intensity was computed taking the average of the brightest 30-60 voxels at every time point after subtracting z-plane specific background intensity. Background values were computed by averaging pixels not belonging to any ROI within a radius of 21 pixels. $DF/F0$ was computed for each neuron, with $F0$ as mean background fluorescence.

1.4.5 Identification of neuron genetic identities

In each recording, we detected 45-140 neurons. Neurons were identified by assessing their anatomical position, relationship to surrounding neurons, and their established activity patterns. Furthermore the strains used for experiments here express the NeuroPAL genetic cassette (Yemini et al., 2021), which uses a genetically defined combination of 4 fluorophores to discriminate neurons based on multi-color reporter expression. Neuron labels were assigned by hand, using NeuroPAL documentation for guidance. In many cases ambiguity still existed, so here we opted for a more conservative approach and chose not to ID neurons which could not be identified beyond a reasonable doubt. In rare cases, (e.g. in **Fig. 1e**) ambiguous identities are denoted in parentheses.

Behavioral decoding of whole-brain recordings was performed as previously described (Kato et al., 2015).

1.4.6 Neural timeseries derivatives and embeddings

Derivatives and PCA on neural time series data was performed as previously described (Chartrand, 2011, Kato et al., 2015). Specifically, total-variation regularization was used to compute de-noised time derivatives while while resolving the accuracy of command state transitions to single frames. Numerical differentiation ensures some degree of stationarity to neural signal, improving subsequent analyses. This approach was also used for the calculation of $\text{proj}(\text{headswing})$. In more detail, for calculating $\text{proj}(\text{headswing})$, we first subtracted the sums of projections to the derivative timeseries onto temporal $\text{proj}(\text{command})$. Next, we subsegment forward command states and remove candidate forward states inconsistent with sustained forward run defined by a threshold ($<4\text{s}$). The resulting segments are concatenated for PCA (Jolliffe 2002) to define loadings. Full timeseries are then projected onto these loadings. Prior to PCA, timeseries are detrended and regularized. For comparison across animals, we applied PC matching by inverting projections such that genetically identified neuron SMDV would be negative. In some cases, such as a large number of neurons drifting out of the focal plane contributing high variance, this procedure was adjusted by first filtering neurons with thresholded (0.5) normalized covariance to SMDV or by further subsegmenting the beginning, or end, of the full timeseries, i.e. temporal cropping.

1.4.7 Linear predictive modeling framework

For predictive modeling in Fig. 2-3, we used linear KNN classifiers implemented by scikit-learn (Pedregosa et al., 2011) with 5-fold cross-validation to limit overfitting. To account for class balance, we report the balanced accuracy metric, which is equivalent to accuracy score with class-balanced sample weights. For timepoint predictive analysis (**Fig. 3j,k**), separate models were fit on neural activity (DFF and derivative as independent features) at single timepoints relative to stimulation.

1.4.8 Statistical tests

In Figs. 2d and 3e-i, the significance of the effect of neuron stimulation on time to reversal was assessed using a Kolmogorov-Smirnov test. The comparable control distribution must account for spontaneous command state transitions unaffected by stimulation. To calculate control null distribution, first we take unstimulated states from control worms and compute the marginal distribution of time until state transition. Then accumulate shifted state durations selected from our experimental group, based on time from stimulation, to generate the marginal distribution of time until state transition. The circular correlation coefficient $\text{proj}(\text{headswing})$ phase at stim onset and $\text{proj}(\text{headswing})$ phase at reversal onset (**Fig. 2h**) was calculated using `astropy.stats.circcorrcoef`. Significance was assessed using permutation testing. Non-significance

of stimulation of dorsal and ventral turn neurons on turn bias in **Fig. 3d** was assessed using a chi-squared test.

1.4.9 Command state ramping and phase analysis

The time constant of `proj(command)` ramping was calculated by fitting a saturated curve to `proj(command)` traces scaled to each reversal. `proj(command)` and `proj(headswing)` phase were calculated by smoothing both traces with a gaussian filter, identifying extrema (`scipy.signal.find_peaks`), and linearly interpolating between peaks and valleys. The ratio of `proj(command)` to `proj(headswing)` period was calculated using the inter-peak-intervals of each signal for the corresponding cycles.

CHAPTER 2: CONTRIBUTIONS ENABLING DATA STANDARDIZATION AND CLOSED- LOOP EXPERIMENTATION

2.1 On the Development of Technologies for Probing Brain Function

Recent advances in whole-brain imaging have revolutionized our understanding of neural function in model organisms. *C. elegans*, with its genetic tractability, fully mapped connectome, and stereotyped 302-neuron nervous system, offers an ideal platform for systems neuroscience research. However *C. elegans* community and neuroscience more broadly has faced significant challenges in data standardization, analysis, and experimental reproducibility across laboratories. This chapter chronicles my efforts to address these fundamental challenges through three interconnected technological contributions aimed at enhancing standardization, research capabilities, and reproducibility. Through a different lens, these technologies were essential in establishing the critical infrastructure necessary for the Kato lab's *C. elegans* research program.

The first contribution introduces a Closed Loop Experiment Framework (CLEF) designed to overcome the computational challenges associated with real-time data acquisition and analysis in neuroscience experiments. The first major wager that I made for my graduate school research was in developing a software system to enable closed-loop perturbative

experiments. I believe that closed-loop experimental design is essential for progress in understanding complex highly interconnected systems like brains. This framework addresses the tension between intensive computational demands and the need for low-latency feedback in closed-loop experiments. By providing a modular, extensible platform that abstracts complex computational resources from experimenters, CLEF enables next-generation perturbation studies of neural dynamics, with distant applications in advanced neuromodulation therapies, and machine learning applications for brain-machine interfaces without requiring extensive programming expertise. I built CLEF to enable the experiments presented in Chapter 1, and I anticipate that the proliferation of the CLEF framework can enable advances in understanding of nervous systems beyond *C. elegans*. Organization of this subchapter and the associated software package into a peer reviewed manuscript is ongoing.

The second contribution presents WormID (see wormid.org), a comprehensive data harmonization framework for *C. elegans* volumetric microscopy data that establishes a standardized format for *C. elegans* whole-brain imaging datasets, pre-processing techniques, and human-in-the-loop machine-learning-based analysis tools. The *C. elegans* community continually produces whole-brain imaging datasets that could in theory be reused for new scientific inquiry, but because the data are produced using disparate equipment and techniques and stored in ad hoc formats, it is challenging to assimilate and reuse this growing collection. By developing NWB extensions specifically designed for volumetric microscopy data, creating preprocessing tools for alignment, and building an extensive corpus of 118 annotated datasets from five laboratories to assimilate data from over 100 animals from five labs, this work enables robust

cross-laboratory analysis and algorithm development. Using this corpus, three existing automated cell-identification algorithms (CPD, StatAtlas, and CRF_ID) were trained to produce a robust, lab-agnostic neural identification atlas system. This system does not require re-training for new labs, discovered biological factors that influence neural positioning, and outperformed previous methods by achieving up to 95% accuracy in automated neuron identification, recovering all human-labeled neurons in some cases. This resource not only facilitates algorithm development and benchmarking but also enables novel biological insights into neuronal positioning and organization in *C. elegans*. This growing resource of data, code, apps, and tutorials enables users to (1) study neuroanatomical organization and neural activity across diverse experimental paradigms, (2) develop and benchmark algorithms for automated neuron detection, segmentation, cell identification, tracking, and activity extraction, and (3) share data with the community and comply with data-sharing policies. This project was led by Yutaka Sprague, a research assistant in the Kato lab and my mentee, who did the lions share of the work. My role specifically was in high-level design, such as with project conception and data analysis objectives. This project was accepted into a peer-reviewed journal

<https://doi.org/10.1016/j.crmeth.2024.100964>.

The third contribution presents two complementary software solutions: Pycro-Manager and minimo. Pycro-Manager bridges the gap between microscopy hardware control and advanced data analysis by creating a Python interface to μ Manager that integrates seamlessly with the scientific computing ecosystem. Its high-level architecture of acquisition events, hooks, and image processors facilitates complex, customized microscopy workflows with concise,

readable code while handling multi-terabyte datasets at speeds exceeding 1 GB/s. The lead on this project was Henry Pinkard from Laura Waller’s lab, and I was included to test components and develop example applications for pycromanager documentation. Furthermore, the current stable version of CLEF was built using pycromanager to interact with microscopy components.

Complementing this, minimo provides a linked data and metadata storage system specifically designed for small laboratories, combining object storage for large raw data with a document database for metadata to ensure context preservation and reproducibility in scientific research. I developed V1 of this software system, and the lead author on the associated manuscript of this work was Jackson Borchardt, who developed minimo V2. Both pycromanager and minimo were published in peer-reviewed journals (<https://doi.org/10.1038/s41592-021-01087-6>, <https://doi.org/10.21105/joss.02979>)

First and foremost, these technologies were necessary for me to complete the primary experimental work of my dissertation. I am optimistic that they may be useful to the wider scientific community. The challenges addressed by these technologies—data standardization, closed-loop experimental design, and data organization are essential to any laboratory. By addressing key technological limitations, this work not only enhances current research capabilities but also lays the groundwork for future innovations in understanding neural function across laboratories and experimental paradigms.

2.2 A Real-Time Closed-Loop Experimental Framework

The traditional paradigm for neuroscientific experiments and interventions has been to design an experiment or interventional session, pre-specify the parameters of the session, then run the experiment and analyze the results. However, there is widespread agreement in the field that performing next-generation neuroscientific experiments and developing personalized, adaptive interventional therapies both require the implementation of closed-loop systems. A closed-loop system is defined as having the capability to modify its parameters or stimulus protocols on-the-fly, responsive to data being read-out during a session. We outline here a proposed collaborative project for enabling closed-loop neuroscience via a modular, extensible platform dubbed CLEF (Closed Loop Experiment Framework).

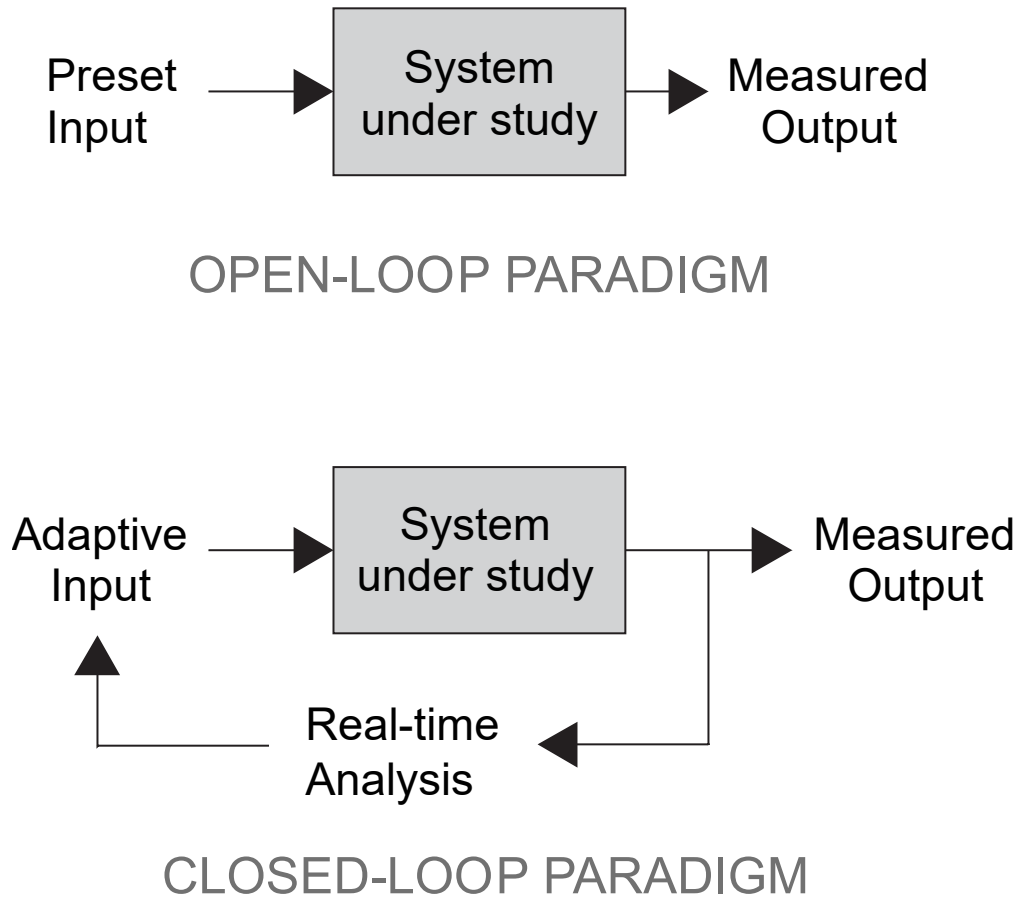


Figure 2.2.1: The closed-loop versus open-loop experimental paradigm.

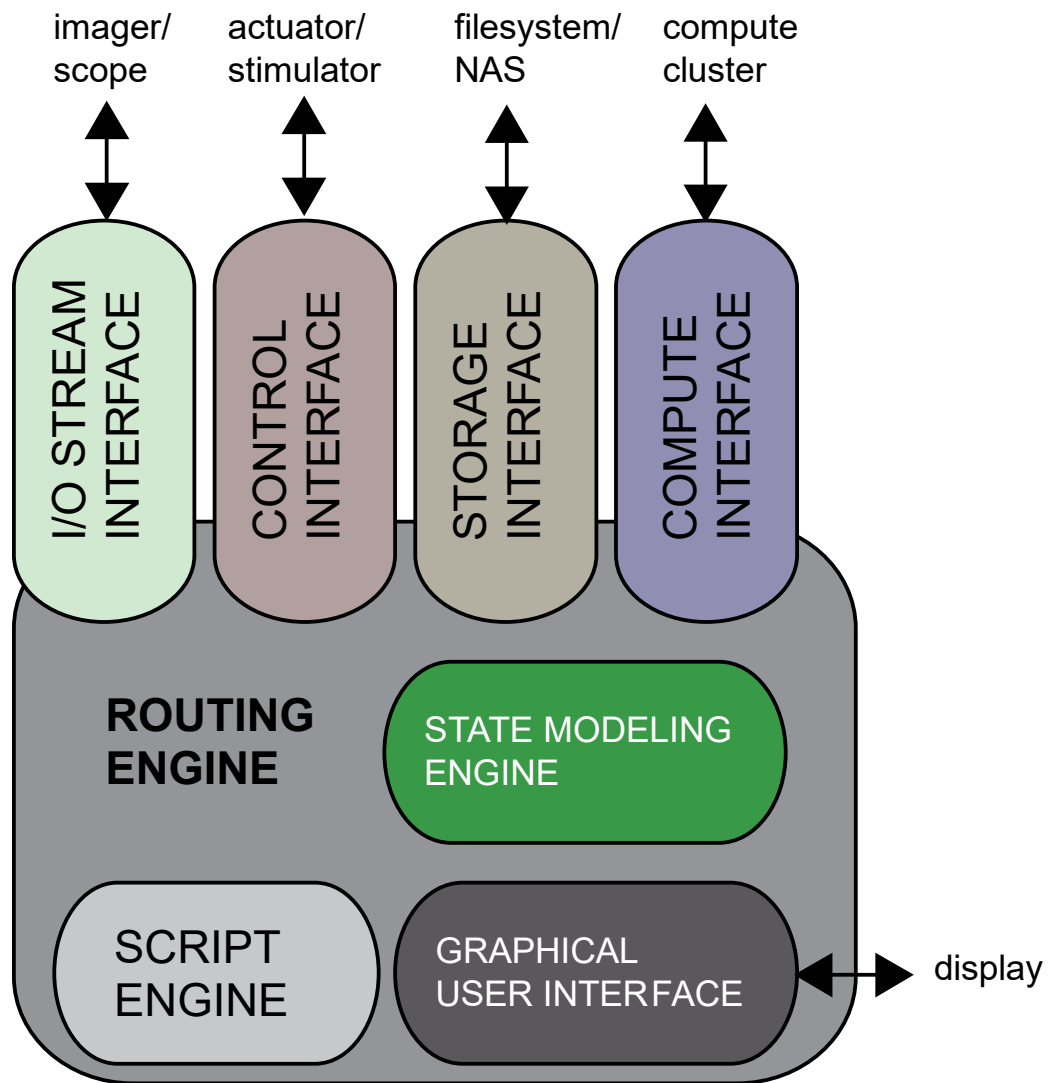
2.2.1 Problem statement

The computational demands of modern imaging and data acquisition, feature extraction, analysis, and intelligent stimulus design exist at tension with the need for real-time speed in a live experimental or therapeutic session. Powerful, dedicated, low-latency computing resources are required, rather than the thin client, high-latency cloud-computing based architectures

appropriate for most traditional contexts. We describe three specific closed-loop projects with significant real-time computational needs.

2.2.2 Proposed software solutions

We propose to implement and provide the ongoing support infrastructure for a modular, extensible software and computer systems framework that provides the “connective tissue” for closed-loop systems, entitled CLEF (Closed Loop Experiment Framework). This framework will leverage specialized systems architectures designed for low-latency high-performance computing, while abstracting these resources away from the experimenter. The framework will provide graphical user interfaces as well as a simplified programmatic environment and APIs for designing and specifying closed-loop experiments and intervention, and for incorporating state of the art machine learning algorithms. A prototype is shown in **Figure 2.2** and **2.3**.



CLEF ARCHITECTURE

Figure 2.2.2: CLEF architecture diagram

The routing engine controls data flow between various external elements, maintains a stateful computational model via the state modeling engine, allows for pre-programmed experimental logic via the script engine, and human operator interaction via the graphical user interface module.

The benefits of the proposed solution are:

- 1) Drastically reduce or eliminate the human software engineering resources and programming expertise required by biological and medical investigators to design closed-loop experiments and trials.
- 2) Enable the effective involvement of data science and machine learning experts in these biomedical studies across all campuses.
- 3) Avoid the notorious technological obsolescence of experimental systems designed under single grants or single lab projects.

Such a framework would draw on the synergistic core competencies of investigators and groups at UCSF, Berkeley, and the National Labs. The design and implementation of CLEF would involve the collaboration of biological investigators, electrical and optical engineers, systems architects, and computer scientists.

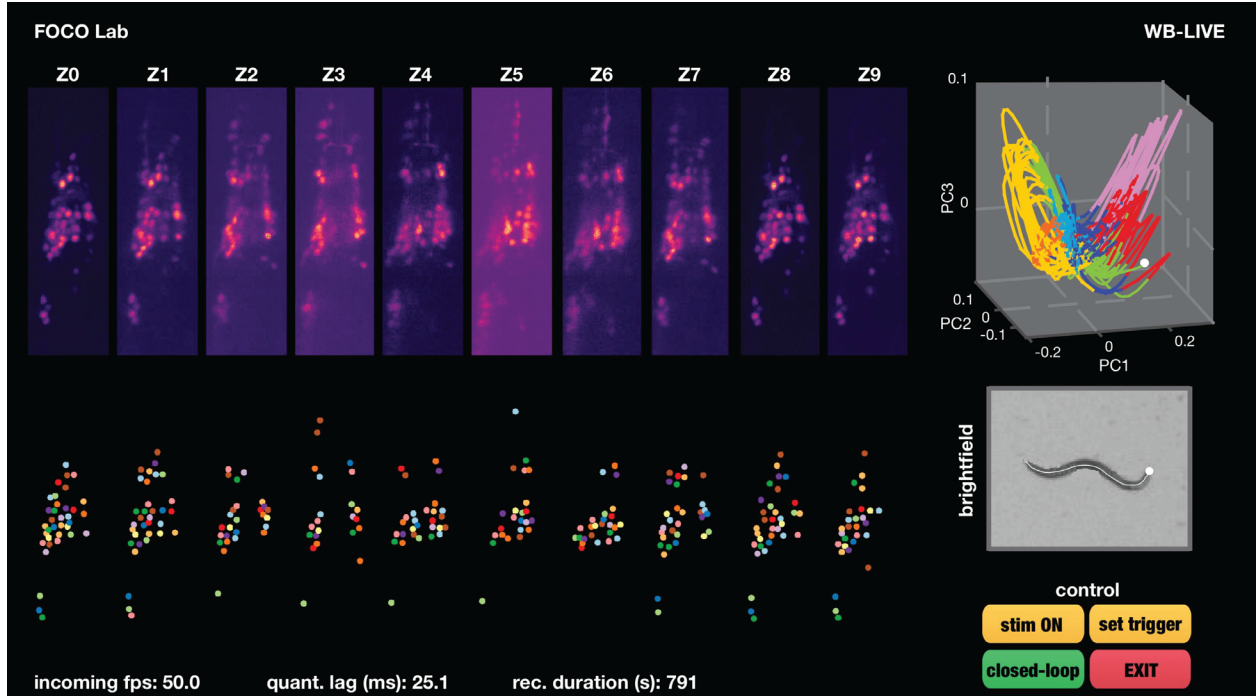


Figure 2.2.3: CLEF GUI prototype

The graphical user interface allows a human operator to control experiments and monitor raw and processed data in real time. The top panel displays imaging planes in real time during volumetric acquisition. The bottom panel displays pseudo-colored segmented neurons. The top right panel displays quantified pixels from neurons transformed into a low-dimensional subspace. The bottom right panel displays segmented and skeletonized worms during freely moving behavioral recording.

2.2.3 CLEF technical architecture

Core components

CLEF is built to be modular, and its architecture includes several key components designed for extensibility and performance:

1. Routing engine: the central hub that manages data flow between all system components, handling both synchronous and asynchronous communication. This component implements a message-passing framework that enables different modules to communicate without tightly coupling their implementations.
2. I/O stream interface: provides standardized connectivity to microscopes, cameras, and other data acquisition hardware. This component supports multiple hardware backends, including MicroManager, direct camera APIs, and custom device drivers. It also handles hardware-specific initialization, configuration, acquisition parameters, and provides a unified data format for downstream processing.
3. Control interface: manages stimulation devices including optogenetic stimulators, laser systems, and digital micromirror devices (DMDs). This component uses precise timing control, spatial patterning capabilities, and protocol management, and it supports multiple common neurobiological stimulation paradigms including pattern stimulation, closed-loop triggering, and timing-critical protocols.
4. State modeling engine: maintains and updates computational models of the experimental system in real-time. This engine supports multiple modeling approaches including statistical models, dynamical systems models, and machine learning based predictive models. The state modeling engine provides a framework for integrating custom algorithmic approaches for specific experimental needs.

5. Script engine: enabling programmable experimental logic without requiring deep knowledge of the underlying system, this Python-based scripting environment provides a simplified API for defining experimental protocols, conditional logic, and adaptive procedures. The script engine includes debugging tools, parameter validation, and experiment templating to facilitate rapid development of new protocols.
6. Graphical user interface: provides real-time visualization and interactive control for experimenters and uses multiple visualization modalities including volumetric data display, time series plotting, feature space representation, and control panels. It supports customizable layouts to accommodate different experimental needs and hardware configurations.
7. Storage interface: manages data persistence and retrieval, connecting to Network Attached Storage (NAS) systems. This component handles efficient write operations for high-bandwidth data streams, metadata management, and experiment configuration storage. It also implements automatic backup strategies and data integrity verification.
8. Compute interface: facilitates distributed computing for intensive analysis tasks. This component enables offloading of computationally demanding tasks to dedicated hardware including GPUs and compute clusters. The compute interface manages resource allocation, job scheduling, and result integration.

Technology stack

CLEF is implemented using a combination of technologies optimized for performance and extensibility:

- Python: the primary programming language, chosen for its widespread adoption in scientific computing and extensive libraries.
- PyQt/PyQtGraph: provides high-performance visualization capabilities for real-time data display.
- Numba: delivers near-native performance for computational bottlenecks through just-in-time compilation.
- NumPy: enables efficient numerical operations on large datasets.
- Shared memory IPC: facilitates low-latency communication between processes using memory regions accessible to multiple processes simultaneously. This approach minimizes data copying overhead and enables efficient parallel processing pipelines.
- Multiprocessing: leverages modern multi-core processors to parallelize data processing pipelines. The framework implements a task-based parallelism model where independent operations can be distributed across available CPU cores.

- OpenCV: provides optimized computer vision algorithms for image processing tasks such as filtering, segmentation, and feature detection. These capabilities are essential for extracting meaningful information from raw imaging data.
- SciPy: supplies scientific computing tools including signal processing, statistical analysis, and optimization algorithms. These components support sophisticated analysis of neural dynamics and experimental data.

2.2.4 Implementation case study: *wb-live*

System overview

The first full implementation of the CLEF framework, known as *wb-live*, was developed for real-time, whole-brain imaging and closed-loop optogenetic stimulation in the *C. elegans* model organism. The system integrates with spinning disk confocal microscopy to capture volumetric neural activity data while simultaneously controlling targeted optical stimulation based on the observed neural dynamics.

The *wb-live* system addresses several key technical challenges:

- volumetric imaging at cellular resolution across the entire *C. elegans* nervous system
- real-time identification and tracking of individual neurons in moving specimens

- extraction of neuronal activity signals from fluorescence imaging data
- detection of specific activity patterns across neural populations
- precise spatial and temporal control of optogenetic stimulation
- seamless integration with existing microscopy hardware

The implementation consists of approximately 20,000 lines of Python code organized into modular components following the CLEF architecture. The system has been successfully deployed in multiple neuroscience laboratories, demonstrating its robustness and usability.

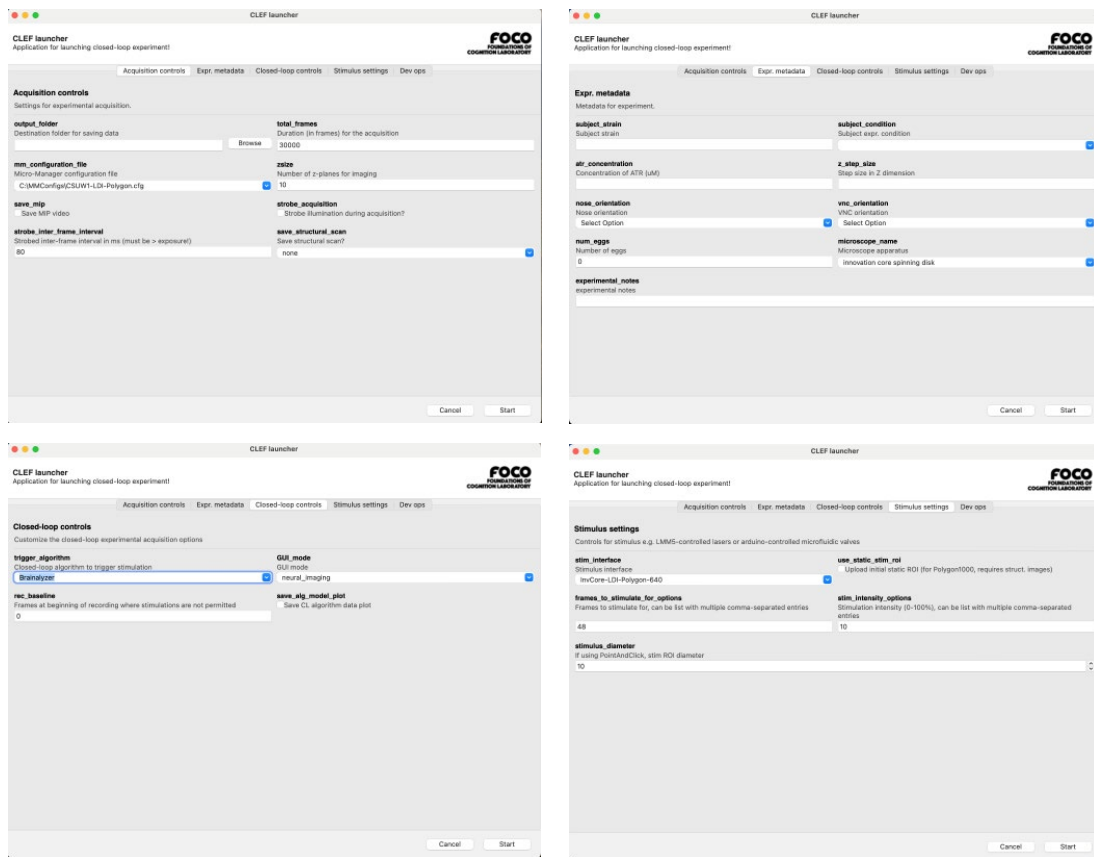


Figure 2.2.4: The *wb-live* launcher app, complete with lab branding

Data acquisition pipeline

The *wb-live* system implements a high-performance data acquisition pipeline:

1. Image acquisition: continuous volumetric imaging at rates up to 5 volumes per second. The system supports z-stack acquisition through piezoelectric focus control synchronized with camera exposures. Acquisition parameters including exposure time, binning, and region of interest are configurable through the user interface.
2. Image processing: real-time filtering, segmentation, and feature extraction from neuronal activity. The processing pipeline includes motion correction to compensate for specimen movement, background subtraction to enhance signal quality, and denoising to improve signal-to-noise ratio. These operations are optimized using Numba and parallel processing to maintain throughput.
3. Feature analysis: dimensionality reduction and pattern detection in neural activity dynamics. The system extracts relevant features using techniques including principal component analysis, time-domain filtering, and threshold-based event detection. These features serve as inputs to the closed-loop decision logic.
4. Decision logic: determines stimulation parameters based on observed neural states. The decision module implements multiple algorithms ranging from simple threshold-based triggers to more sophisticated pattern recognition approaches. This component is designed to be extensible through a plugin architecture.

5. Stimulus control: precise spatial and temporal control of optogenetic stimulation. The system supports multiple stimulation hardware backends including digital micromirror devices, laser scanning systems, and LED arrays. Spatial patterns, timing parameters, and intensity profiles are configurable through the experiment protocol.

The entire pipeline operates with an end-to-end latency of less than 100 milliseconds from image acquisition to stimulation, enabling genuine closed-loop experimental paradigms at behaviorally and physiologically relevant timescales.



Figure 2.2.5: The *wb-live* Brainalyzer GUI in closed-loop neural imaging mode

Closed-loop algorithms

The system supports multiple closed-loop experimental paradigms through a plugin architecture that allows researchers to implement custom decision logic:

1. Threshold-based: triggers stimulation when neural activity exceeds predefined thresholds.

This approach is computationally efficient and conceptually straightforward, making it suitable for initial closed-loop experiments. The threshold parameters can be adjusted dynamically during experiments to tune sensitivity.

2. Pattern recognition: identifies specific patterns in neural dynamics to trigger intervention.

This algorithm detects temporal sequences or spatial configurations of neural activity associated with particular behavioral states or functional circuit dynamics. Pattern templates can be defined manually or learned from previous recordings.

3. State-based: models the neural system as transitioning between discrete states and intervenes

at specific transition points. This approach employs statistical models to classify observed neural activity patterns into predefined states, enabling interventions targeted at specific points in neural state trajectories.

4. Predictive: uses machine learning to predict future neural states and applies preemptive stimulation. This sophisticated approach leverages recurrent neural networks or state space models to forecast neural dynamics, allowing intervention before target states are fully realized.

5. User-directed: allows for manual triggering of stimulation based on real-time visualization of neural activity. This interactive mode enables experimenters to apply their expert judgment to initiate interventions based on complex patterns that may be difficult to algorithmically define.

Each algorithm is implemented as a separate module with a standardized interface, allowing researchers to switch between approaches or combine multiple strategies within a single experiment. Performance characteristics including computational demands, response latency, and detection accuracy are tracked to guide algorithm selection.



Figure 2.2.6: The *wb-live* Brainalyzer GUI in closed-loop behavior mode

Hardware Acceleration

To meet these performance demands, CLEF leverages several hardware acceleration strategies:

- Graphics processing units (GPU) acceleration: for image processing and neural network inference, GPUs provide massive parallelism for operations like filtering, segmentation, and deep learning model execution. The framework includes GPU-optimized implementations of common image processing operations.
- Dedicated computing hardware: high-performance workstations with optimized configurations. These systems typically include multi-core CPUs, substantial RAM configurations, and high-speed storage to handle sustained data rates.
- Custom I/O hardware: direct digital connections between acquisition and stimulation systems. These connections minimize latency for time-critical signals and provide deterministic timing characteristics.
- Memory optimization: shared memory architecture to minimize data transfer overhead. By avoiding unnecessary data copies between processing stages, the system maintains throughput even with high data rates.
- Distribution of computation: separation of critical real-time processing from downstream analysis and storage operations. This approach ensures that essential closed-loop functions maintain consistent performance regardless of storage or analysis loads.

2.2.5 Example 1: Perturbational studies of neural dynamics

Single-cell resolution brain-wide or multi-brain-area imaging has emerged as a powerful paradigm to observe and decipher the real-time operation of biological neural networks in model organisms. The next generation of studies will move from passive, correlational observation or brute-force perturbational approaches to spatiotemporally patterned perturbations under high resolution monitoring. However, a substantial amount of computation is required to implement machine vision algorithms to extract biologically relevant features such as neuronal time series and latent factors. Additionally, computational microscopy has emerged as a powerful method to drastically improve the performance and resolving power of traditional purely optical microscopy techniques. However, these computations require substantial offline processing. A high-performance low-latency compute fabric could enable the use of these techniques in a closed-loop scenario.

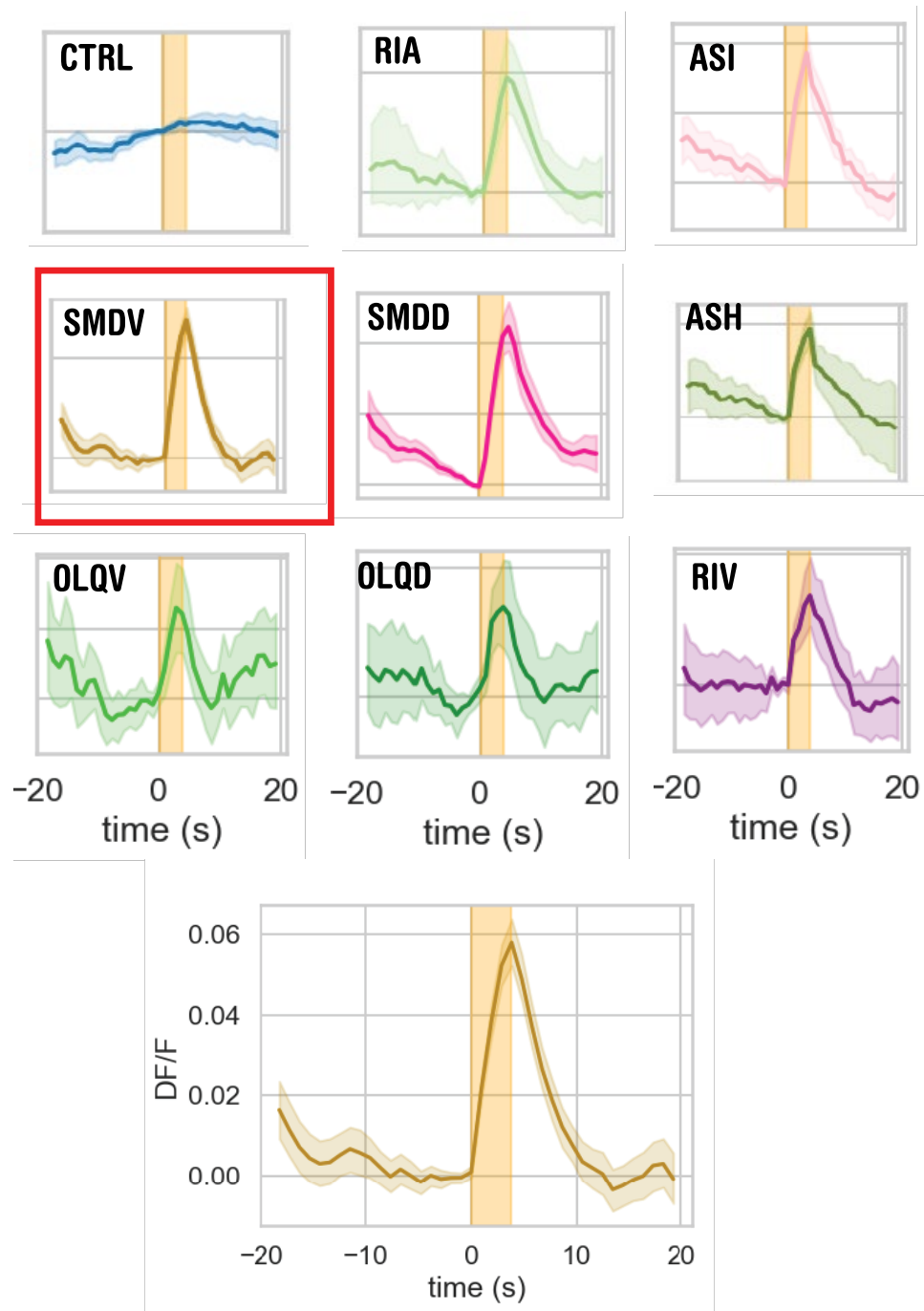


Figure 2.2.7: Pulse-triggered averages from stimulating various neurons with optogenetics

2.2.6 Example 2: Next-gen brain neuromodulation

DBS (Deep Brain Stimulation) and tES (Transcranial Electrical Stimulation) have emerged as important interventional therapies for a wide spectrum of brain disease, though mechanisms remain largely unelucidated. The next generation of these approaches is aimed to be more selective in effect, leading to fewer side effects and tolerance effects. There is consensus that spatially and temporally patterned stimuli are likely required to achieve increased selectivity and potency. Moreover, it is quite likely that circuit specific interventions that are adaptive to the ongoing local and global dynamics will be the most targeted. However, the possible choices of spatiotemporal patterns is combinatorically large, and hence, the only practical way to find these effective patterns is to utilize an adaptive search algorithm responsive to effects observed during a session. With the advent of large-scale recording systems that can simultaneously monitor 500-1000 neurons across multiple brain regions in real time, this approach offers the possibility of decoding and stimulating in a previously unparalleled manner. This approach can potentially be applied to many models of neurological and psychiatric disorders.

2.2.7 Example 3: Machine learning for brain-machine interfaces

Over the past decade, there have been increasingly sophisticated demonstrations of the direct decoding of neural information in real time; this has also resulted in direct neural control of increasingly complex external devices. Strikingly, this has been achieved with relatively low

channel counts, typically ~ 100 channels, likely to increase 10 to 100-fold in the near future. In parallel, there is also great interest in the use of advanced machine learning approaches to decode multiple parallel streams of information. While artificial neural networks have shown early promise in allowing improved neural control, a plethora of novel state-of-art approaches remain unexplored. The combination of large-scale recording technology, machine learning methods and real-time computing platforms offers an incredible opportunity to push the field exponentially forward.

2.2.8 Conclusion

The CLEF framework represents a significant advance in experimental neuroscience methodology, enabling a new generation of adaptive, intelligent experiments that respond to biological systems in real-time. By addressing the technical challenges of closed-loop experimentation through a modular, extensible architecture, CLEF reduces the barrier to entry for researchers seeking to implement sophisticated feedback-based protocols. As computational capabilities continue to advance, this approach promises to accelerate discovery in fundamental neuroscience and the development of personalized neuromodulatory therapies.

2.3 Unifying community whole-brain imaging datasets enables robust neuron identification and reveals determinants of neuron position in *C. elegans*

2.3.1 Introduction

Whole-brain imaging experiments with single-neuron resolution (herein shortened to simply “whole-brain imaging”) have undergone explosive growth since first demonstrated in the millimeter-long nematode worm *Caenorhabditis elegans* and the zebrafish *Danio rerio* in 2013 (Schrödel et al., 2013; Ahrens et al., 2013). Since then, these methods have been widely adopted and advanced in the worm (Kato et al., 2015; Venkatachalam et al., 2016; Nguyen et al., 2016; Yemini et al., 2021; Atanas et al., 2023), zebrafish (Portugues et al., 2014; Prevedel et al., 2014; Royer et al., 2016; Kim et al., 2017; Vladimirov et al., 2018; Chen et al., 2018), and larval (Lemon et al., 2015), and adult (Mann et al., 2017; Aimon et al., 2019) fly communities. Moreover, there have been significant efforts and advances in neuron-resolution imaging of multiple and/or large brain regions in mammals, rapidly approaching whole-brain imaging, especially in mice (Stirman et al., 2016; Skocek et al., 2018; Klioutchnikov et al., 2020; Zong et al., 2022; Manley et al., 2024). In *C. elegans*, whole-brain imaging datasets have enabled characterization of neural network dynamics (Kato et al., 2015; Yemini et al., 2021), functional connectivity (Uzel et al., 2022; Flavell et al., 2022; Randi et al., 2023; Chaudhary et al., 2021) and the roles of individual neurons during behavior (Atanas et al., 2023). These studies leverage

the property of eutely in this organism: each cell has a unique and stereotyped identity, consistent across every animal, that allows for data from individual neurons to be pooled and compared across multiple trials and animals. However, analyses of these experiments are bottlenecked by the need to determine the unique identities of each neuron in 3D volumetric recordings. Manual cell identification from fluorescent microscope images is a notoriously difficult skill, requiring substantial expertise and labor. This task is particularly difficult for neurons labeled with nuclear localized fluorophores, which is typical for whole-brain recordings. We recently developed NeuroPAL (Yemini et al., 2021), the first method where the unique identity of every single neuron can be distinguished by an invariant fluorescent color barcode in living animals at all developmental stages of both sexes (Tekieli et al., 2021).

NeuroPAL has greatly simplified the task of cell identification and has thus seen rapid adoption, with at least six labs (Yemini et al., 2021; Atanas et al., 2023; Uzel et al., 2022; Randi et al., 2023; Chaudhary et al., 2021) publishing whole-brain imaging datasets using these animals and many more labs incorporating the system into their experimental protocols since its release in 2021. Despite this innovation, neural identification remains a challenging task that requires expertise and many hours of manual work. In the past few years, researchers have proposed various algorithmic auto-identification approaches to attack this problem (Chaudhary et al., 2021; Varol et al., 2020; Nejatbakhsh et al., 2020; Skuhersky et al., 2022; Toyoshima et al., 2020; Bubnis et al., 2019). However, none of them have achieved widespread adoption, due at least in part to their incompatibility with different microscopy data formats and low performance on data acquired from different labs. Automatic approaches to the complementary

problem of tracking neurons across video frames have achieved some generalized performance across various datasets (Wu et al., 2022; Yu et al., 2021) but so far, there have not been efforts to perform similar training and benchmarking for automatic cell identification. To build automatic approaches that are robust, accurate, and generalizable, there is a critical need for a standardized format and compatible tools trained and benchmarked on a consolidated corpus of data that reflects the heterogeneity of microscopy equipment, experimental conditions, and protocols across labs.

To address this need, we take a data harmonization approach: a process of combining datasets from different sources and homogenizing them to produce a substantially larger data corpus that, in our case, minimizes non-biological inconsistencies across individual datasets while increasing the overall biological diversity of training and benchmarking data. Harmonization includes (1) aggregating the data, (2) converting it to a standardized format, (3) normalizing it, (4) handling duplicate and missing data, and (5) pre-processing data to register it to a common space and coordinate system. Data harmonization is standard in many data science fields but has seen slower adoption in the life sciences (Cheng et al., 2024). Similar efforts to standardize data formats and build large corpuses of data have been essential in the development and benchmarking of many modern machine-learning algorithms (Deng et al., 2012; Deng et al., 2009; Lhoest et al., 2021).

We introduce WormID (wormid.org). This resource consists of (1) data harmonization tools including a standardized file format for both raw and processed data alongside related metadata that extends the existing Neurodata Without Borders (NWB) format, (2) pre-

processing to align the color and coordinate space of new datasets, and (3) open-source software to analyze whole-brain activity images. We also provide tutorials and documentation that enable researchers to easily incorporate these tools into their data pipelines. Finally, we provide a large online corpus of harmonized *C. elegans* whole-brain activity imaging and structural data that can be used for large-scale experimental analysis, neurobiological modeling, and algorithmic development. This corpus is stored in a popular community archive called the Distributed Archives for Neurophysiology Data Integration (DANDI), which serves as a free, persistent, open access repository for experimental neuroscience data from a variety of model organisms (Halchenko et al., 2024). The addition of new datasets to this corpus are encouraged, facilitated by tutorials and software, and provide a simple means for submitters to comply with data-sharing policies of federal and private funders.

By aggregating a diversity of datasets from multiple labs into a large data corpus, we achieve a substantial boost in the performance of three existing neural auto-identification algorithms, arguably moving into the regime of practical utility for the broader community of users. Furthermore, we mine this corpus to investigate the relationship between neural lineage, synaptic connectivity, and somatic positioning of *C. elegans* neurons to better understand the factors that drive the positioning of neurons in the adult worm.

This corpus and set of tools should be of wide utility to *C. elegans* researchers. We hope it will serve as a seed for continued community aggregation of brain imaging datasets and further the development and improvement of community data-analysis tools applicable across many model organisms.

2.3.2 Results

*A standardized format for whole-brain *C. elegans* recordings enables data aggregation and algorithm interoperability*

Current state-of-the-art whole-brain recordings of *C. elegans* typically consist of a combination of structural images that often use the NeuroPAL multi-channel fluorescent system to determine neuron identities (**Fig. 1a,b**) and time-series images of neural activity acquired by using genetically encoded activity sensors such as GCaMP6s (Chen et al., 2013) (**Fig. 1c**). This imaging is performed either on immobilized worms often constrained within a microfluidic chip to maximize image quality (Schrödel et al., 2013; Kato et al., 2015; Chronis et al., 2007) or on freely moving worms (Venkatachalam et al., 2016; Nguyen et al., 2016). To aid interpretation, herein we visualize whole-brain structural NeuroPAL images via (1) an unrolled “butterfly” plot of neuron positions that projects the 3D worm structure into a 2D plane (**Fig. 1a**), (2) a 2D projection plot of the NeuroPAL color space (**Fig. 1b**), and (3) 2D dorsal-ventral and lateral projection plots of the neurons (**Fig. 1b**). These visualizations facilitate quick comparisons of neuron color and position from different samples and fine-tuning of their global alignment.

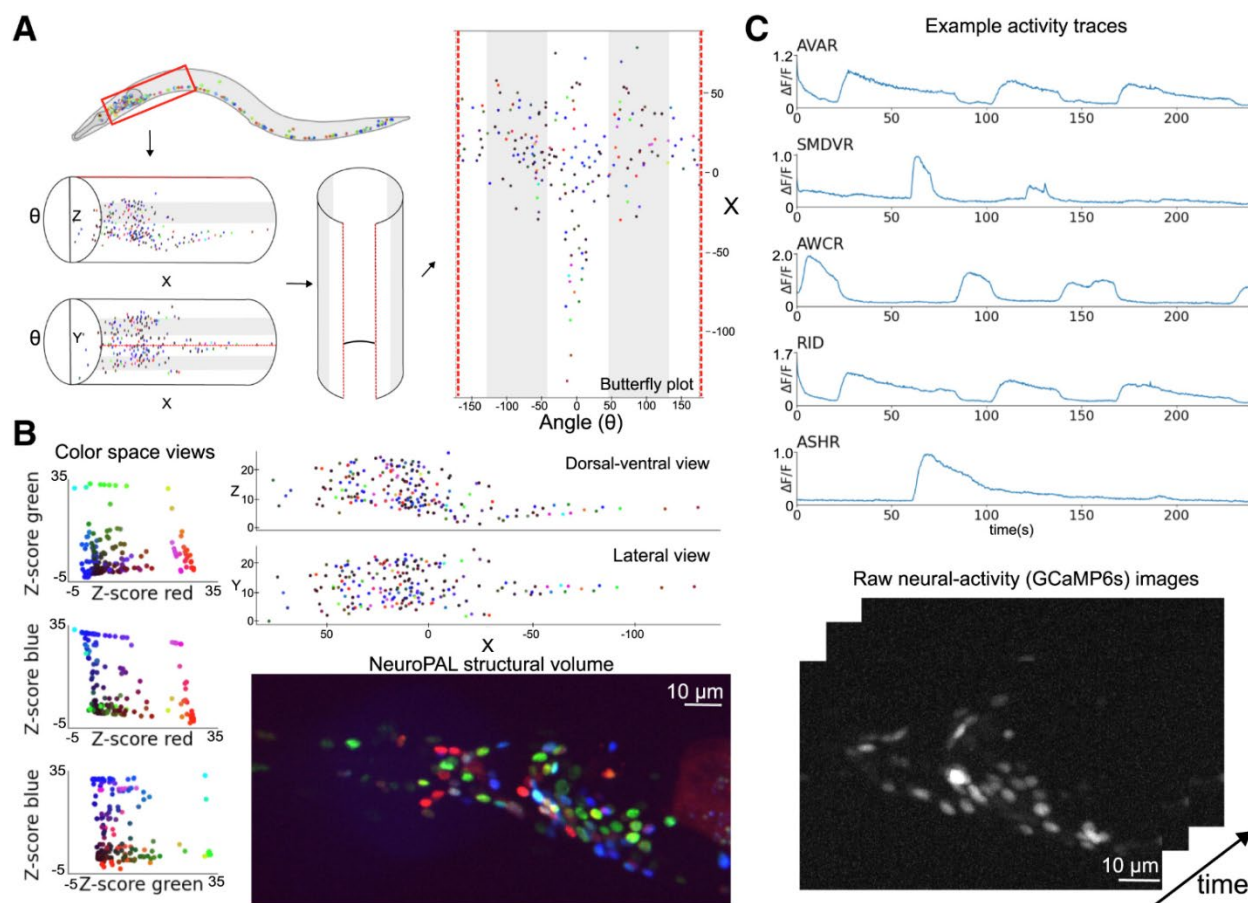


Figure 2.3.1: Example NWB file contents

(A) Illustration of a NeuroPAL worm. The head is highlighted by a red box. A butterfly plot visualizes the full 2D representation of the worm brain by projecting neurons onto the surface of the cylindrical body and then unrolling the cylinder. Neuron centers are colored using their composite NeuroPAL expression. (B) Visualizations of the raw NeuroPAL structural image and 2D projections of its RG, RB, and GB color subspaces and XZ and XY projections of its neuron positions. Neuron centers are colored using their composite NeuroPAL expression. (C) Example activity traces for five neurons contained in the NWB file and of the raw neural-activity (GCaMP6s) images.

All associated raw data and metadata is stored in the standardized NWB41 file format with an additional extension that we developed, *ndx-multichannel-volume* (ndx = neurodata extension), to provide support for multi-channel volumetric recordings and *C. elegans*-specific metadata (**Fig. 2a**). This extension is available in the NWB Extensions Catalog and is now the official NWB standard for data sharing of *C. elegans* whole-brain neural-activity imaging. NWB data are hierarchically organized with basic metadata stored at the file’s root level, raw data stored in the “acquisition module,” and various processed experimental data stored in “processing modules” (**Fig. 2b**). Individual NWB files contain a single experimental run for a single animal. These NWB files are then stored and accessed from the DANDI archive, where they receive a unique persistent digital object identifier (DOI) in accordance with the International Organization for Standardization (ISO). Datasets can then be downloaded or streamed from the DANDI archive using the DANDI interface or API (**Fig. 2c**).

We incorporated NWB *ndx-multichannel-volume* read and write functionality into two software tools. These independent software implementations both offer both user-friendly GUIs that allow the visualization, analysis, and curation of *C. elegans* NeuroPAL structural images and neural activity recordings in immobilized worms (NeuroPAL software https://github.com/Yemini-Lab/NeuroPAL_ID, **Fig. 3a**; eats-worm software <https://github.com/focolab/eats-worm>, **Fig. 3b**). This functionality can be straightforwardly incorporated into other data-analysis pipelines and software.

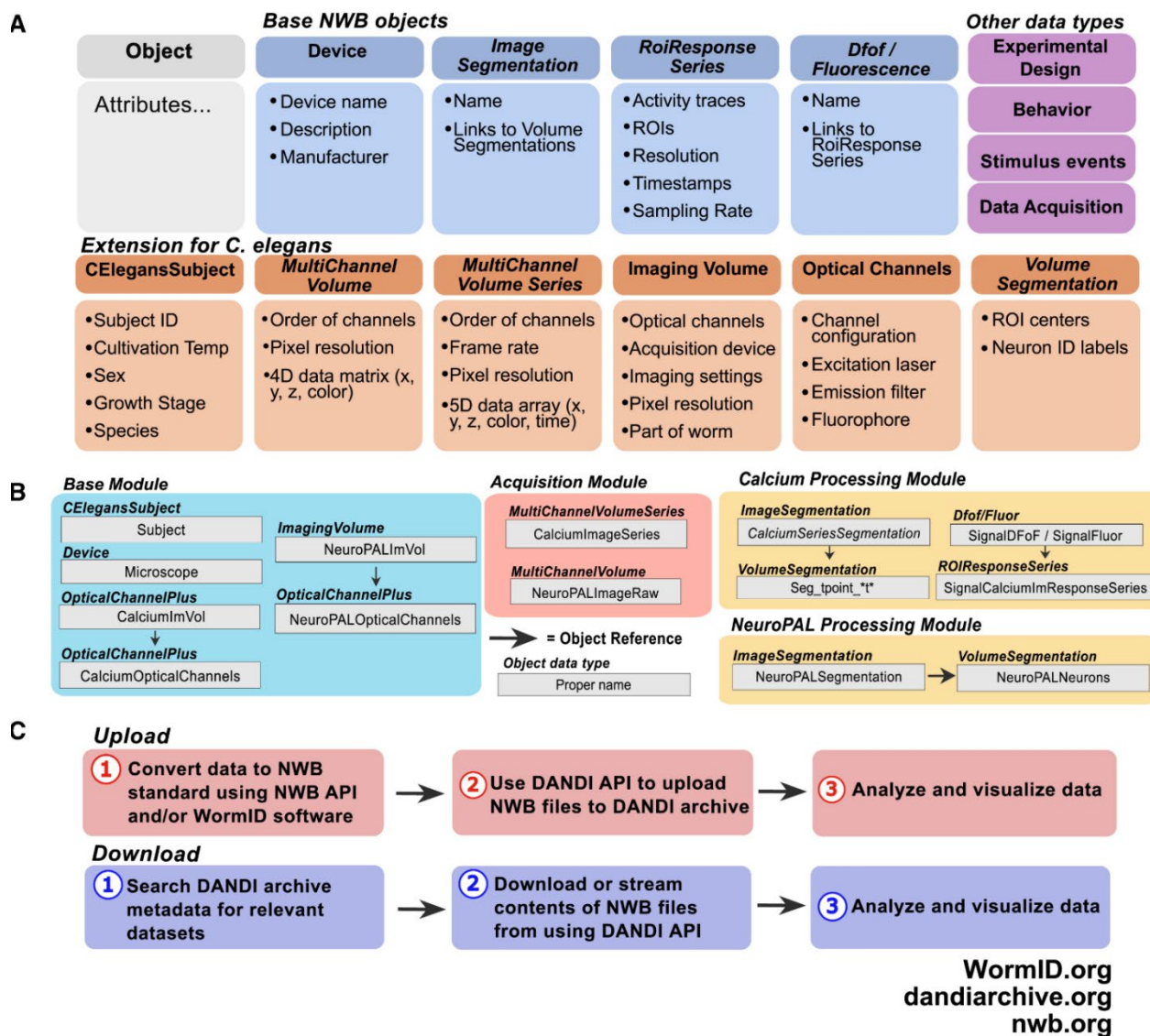


Figure 2.3.2: Schema for *C. elegans* extension to the NWB architecture

(A) Names and content for objects used in *C. elegans* optophysiology NWB files. (B) File organization hierarchy of NWB files for *C. elegans* optophysiology. Modules are structured like folders within the root file in an HDF5-based hierarchy. (C) Flowchart of steps for converting new data to NWB and uploading to the DANDI archive and for using NWB datasets that are already up on DANDI.

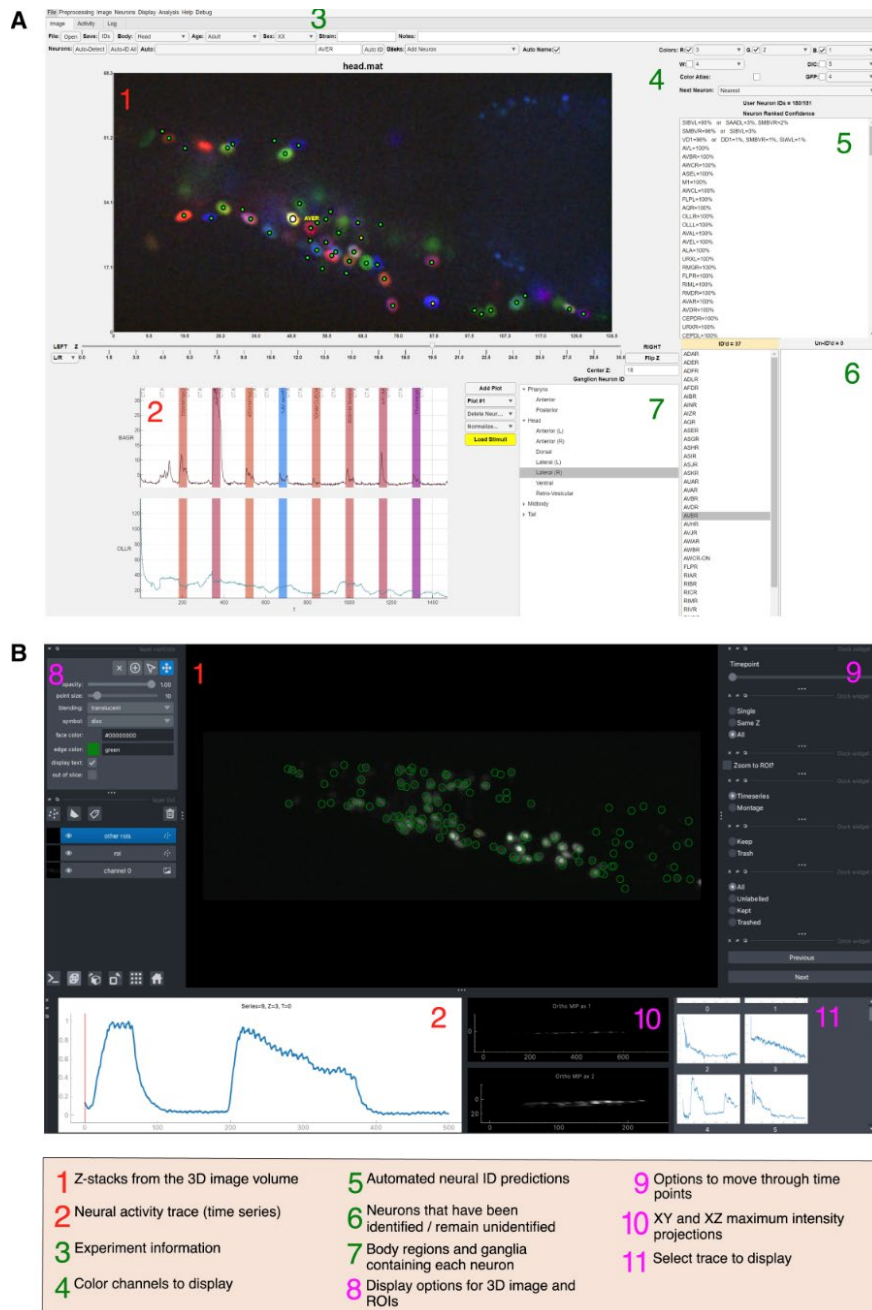


Figure 2.3.3: Two software GUI applications with NWB I/O

(A,B) GUIs with NWB I/O support for visualization and annotation of NeuroPAL structural images, neural detection, and automated identification and time-series of neural activity with stimulus presentation in immobilized worms. Colored numerical callouts highlight specific modules and features in each software. (A) NeuroPAL ID software from Yemini Lab. (B) EATS-worm software from Kato Lab.

Table 2.3.1: Summary of aggregated dataset characteristics.

The NP dataset comes from the original NeuroPAL paper (Yemini et al., 2021). Datasets can be accessed programmatically using the DANDI API and Dandiset ID or by searching the Dandiset ID on dandiarchive.org.

Dataset #	Dandiset ID	# of worms in the dataset	Lab code	NeuroPAL, GCaMP, or both	# of neurons marked (avg.)	# of ID labels (avg.)
NP	000715	10	NP	NeuroPAL	189–196 (193)	186–193 (190)
1	000541	21	EY	Both	166–188 (177)	164–184 (175)
2	000714	9	HL	NeuroPAL	113–125 (119)	58–69 (64)
3	000692	9	KK	Both	149–163 (154)	149–163 (154)
4	000776	38	SF	Both	29–96 (70)	29–96 (70)
5	000565	21	SK1	Both	78–139 (111)	30–82 (48)
6	000472	10	SK2	NeuroPAL	166–180 (173)	38–63 (49)
Summary		118	5 labs		29–196 (126)	29–193 (99)

In **Table 1**, we present a summary of the data we aggregated and harmonized into a corpus: 108 worms from six datasets acquired by five different labs, each with neuron positions and human-labeled neuron identities. This corpus can be mined for biological insights, training and benchmarking of machine-vision approaches, and neurobiological studies of structural and neural-activity time-series data. Each of these datasets is stored on DANDI and range from a few hundred megabytes to several terabytes. DANDI supports streaming from the cloud and allows users to selectively load data objects and data chunks, substantially reducing the local data storage and RAM requirements necessary to work with these data on a personal computer.

*An updated atlas of neuron positions in the *C. elegans* hermaphrodite head*

Our multi-lab data corpus allows data scientists to train and benchmark the performance of algorithms for automated neuron identification using datasets that reflect real-world diversity. In this section, we focus on the statistical atlas approach presented in Varol et al 2020. This approach was the first to take advantage of the color information provided by NeuroPAL and was presented alongside the original NeuroPAL work (Yemini et al., 2021). This neuron identification assignment algorithm was framed as a bipartite graph matching problem, with the goal of minimizing the total assignment cost using the well-known Hungarian algorithm (Kuhn et al., 1955). Cost is calculated by comparing neuron positions and colors in the animal sample with the mean and covariance of neuron positions and colors in a reference statistical atlas. The original atlas presented in the paper was trained on 10 worms from the original NeuroPAL work. We retrained this atlas using the full multi-lab corpus that we present in this work, increasing

the training set by over 10-fold. To train the statistical atlas, we first performed an affine transformation to roughly register each training dataset to one of the 10 original NeuroPAL worms based on ground-truth labels. This roughly aligns the principal axes of the worm as well as the scaling to a common space. The algorithm then calculates the means and covariances of neuron positions and colors in this roughly aligned space for individual recordings. The statistical atlas generated by this approach serves the additional purpose of characterizing the mean and covariance of neuron positions and colors across the whole corpus of data. To the best of our knowledge, this newly generated atlas is the most comprehensive atlas of neuron positions and NeuroPAL coloring. We find that training on the multi-lab corpus significantly decreases the alignment cost for the ground truth labeled neurons across datasets, indicating that the true distribution of neuron positions and variances is more accurately represented in the more comprehensive atlas compared to the original one.

In **Figure 4**, we present visualizations of the statistical atlas of neuron positions and colors trained on 104 of the 118 worms in our consolidated NWB/DANDI dataset as well as on the smaller dataset of 10 worms used in the original NeuroPAL paper (employing the StatAtlas algorithm in Yemini et al., 2021 and Varol et al., 2020). We found that substantial deviations from a linear pose and occluding image artifacts significantly impacted algorithmic alignment and thus accuracy, therefore, 14 worms were omitted from the atlas due to large nonlinear deformities or pronounced imaging artifacts. With 104 worms, this represents, to the best of our knowledge, the most broadly trained statistical atlas for *C. elegans* neuron positions and NeuroPAL coloring available. By leveraging the diversity of the multi-lab corpus this atlas

captures variability between individual worms, strains, and lab-specific experimental conditions. This atlas can be used as a basis for training and testing automatic labeling algorithms as well as biological investigations of neuron positions and structural brain organization. This statistical atlas further complements detailed electron microscopy (EM)-based anatomical atlases by adding cellular structural detail and, notably, it provides nearly 100 more animals in its corpus than the approximately 10 EM ones available now (White et al., 1986; Cook et al., 2019; Witvliet et al., 2021). Although our corpus lacks the synaptic connectivity found in the EM datasets, it provides the complementary functional activity that is not available and cannot be obtained from EM imaging of fixed animals.

WormID.org supplies links to the software, visualization tools, and datasets discussed previously in this paper. Furthermore, wormid.org provides links to the data corpus and related tools to work with whole-brain structural images and activity recordings, convert datasets to NWB, and supplies tutorials and instructions for using these tools. Our aim is that these data standard, data corpus, and atlas of cell positions will be a continually evolving resource for the *C. elegans* neuroscience community and eventually for the communities of other model organisms.

Analysis of biological factors influencing neuron positions

We statistically analyzed the spatial positions of *C. elegans* neuronal somas across individuals, strains, and lab conditions based on their means and covariances in the statistical atlas. We focused on relative pairwise displacements rather than absolute positions because the absolute position of cells is dependent on the pose and deformation of the animal's body during imaging. Pairwise displacements are alignment independent and robust to deformation.

Before analyzing statistical properties on the neuron positions, we assessed the percentage of neurons that were labeled by humans in each dataset. We found that neurons in the ventral ganglion and retrovesicular ganglion were less commonly labeled than neurons in other ganglia. As is shown here and previously in Yemini et al. 2021, 6 neurons in the ventral and retrovesicular ganglia exhibit high relative positional variability, which may explain why fewer of them were confidently labeled by researchers (**Fig. 5b**). For this reason, we explored several different factors hypothesized to contribute to the organization and variability of relative cell positions: (1) gangliar boundaries (e.g., basal lamina and abutting tissue), which may restrict cell movement within the coelem; (2) synaptic connectivity, which may impose energetic costs dependent on neuronal proximity; and (3) developmental-time and cell-lineage effects whereby recently divided cells (i.e., sister cells) remain close together and more distant relatives (e.g., mother and grandmother cells) end up further apart.

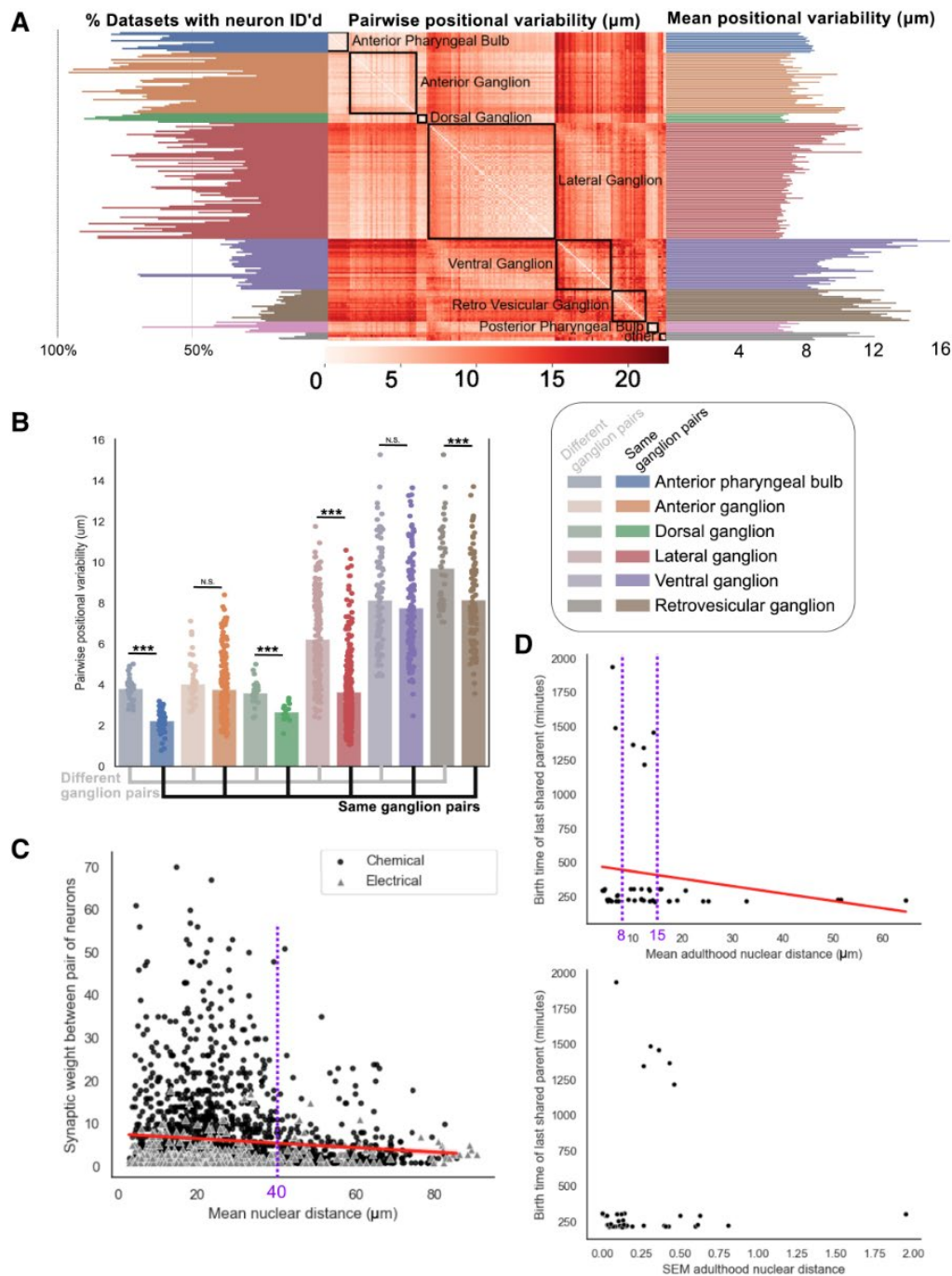


Figure 2.3.5: Analyses of neuron positions, distances, and positional variability

(A) Left: percentage of datasets containing each labeled neuron, organized anterior-to-posterior within each ganglion. Middle: heatmap of the standard deviation of pairwise positional distances between each pair of neurons across datasets. Right: averaged sums of heatmap rows. Neurons with higher mean positional variability have less stereotyped positions within the worm body.

(B) Pairwise positional variability by ganglia for 10 closest neighbors of each neuron, separating

neuron pairs in the same ganglion from pairs in different ganglia. Pairs where the atlas distance was over 20 μm were removed from this analysis. Anterior pharynx: 95% CI effect size = $[-1.79, -1.38]$ μm , $p = 2.5 * 10^{-21}$, $N_{\text{same}} = 52$, $N_{\text{diff}} = 37$. Dorsal: 95% CI effect size = $[-1.15, -0.71]$ μm , $p = 7.1 * 10^{-6}$, $N_{\text{same}} = 13$, $N_{\text{diff}} = 27$. Lateral: effect size 95% CI $[-2.80, -2.38]$ μm , $p = 5.3 * 10^{-31}$, $N_{\text{same}} = 317$, $N_{\text{diff}} = 161$. Retrovesicular: 95% CI effect size = $[-1.94, -1.17]$ μm , $p = 9.3 * 10^{-5}$, $N_{\text{same}} = 89$, $N_{\text{diff}} = 44$. Anterior: $p = 0.161$, $N_{\text{same}} = 176$, $N_{\text{diff}} = 50$. Ventral: $p = 0.252$, $N_{\text{same}} = 135$, $N_{\text{diff}} = 94$. (C) Pairwise relationship between neuron synaptic weights and their mean positional distance for chemical and electrical synapses. Chemical synapses: KendallTau $\tau = -0.036$, $p = 0.021$, Pearson $R = -0.098$, $p = 6.4 * 10^{-6}$, $N = 2119$. Electrical synapses: KendallTau $\tau = 0.009$, $p = 0.80$, Pearson $R = 0.031$, $p = 0.51$, $N = 444$. (D) Relationship between cell birth times and the mean and SEM of their nuclear positional distance in adulthood for sister cells. Mean: KendallTau $\tau = -0.144$, $p = 0.052$, Pearson $R = -0.161$, $p = 0.129$. SEM: KendallTau $\tau = 0.014$, $p = 0.845$, Pearson $R = 0.074$, $p = 0.487$. Most sisters are within 15 μm of each other in adulthood. More sisters that divide embryonically remain close together (<8 μm) than sisters that divide >16 h later at postembryonic larval stages.

To test the first hypothesis, that ganglial boundaries regulate positional organization and variability, we measured the positional variability of neurons that are spatially close and compared pairs within the same ganglion to pairs straddling each other in different ganglia. We found that neurons in the anterior pharyngeal bulb and neurons in the dorsal, lateral, and retrovesicular ganglia all exhibit significantly lower variability for pairs within the same ganglion compared to pairs in different ganglia. Conversely, for neurons in the anterior and ventral ganglia, we observed no significant difference in positional variability between pairs in the same ganglion and pairs in different ganglia (**Fig. 5b**). We used an independent-sample t test to compare pairs within the same ganglion with those in different ganglia. To determine whether disparities in the distributions of pairwise distance between intra- and inter-ganglion groups could trivially account for this difference in variability, we performed distance-matched

comparisons of the variability of inter-ganglia pairs to intra-ganglia pairs in 1 μm bins. We observed a consistently higher variability of inter-ganglia pairs to intra-ganglia pairs regardless of mean distance. We also found no correlation between mean distance and variability for both inter- and intra-ganglia pairs. Known anatomical features of the worm are consistent with our findings and hypothesis of ganglial influence on the spatial relationships of neurons: the pharynx is a muscular epithelial tube (Mango, 2007) that rigidly encases neurons, the remaining ganglia are separated by basal lamina that loosely restricts their boundaries (White et al., 1986) and, finally, the anterior and ventral ganglia (and comparatively smaller retrovesicular ganglion) are completely bounded whereas all other ganglia are open at least at one end, and in White et al. 1986, it had been noted that tight cellular packing in these regions led to “slop,” “uncertainty,” and, in live animals, even “flipping” from side to side of the cells contained therein. This finding suggests that neural identification algorithms could be improved by a hierarchical approach, such as first predicting ganglion membership, then predicting identities within this ganglion.

Next, we explored the relationship between somatic distance and synaptic connectivity. Overall, there was a very weak but statistically significant correlation between nuclear distance and synaptic weight for chemical synapses and no significance or detectable correlation for electrical synapses. However, we found that nearby neurons (mean distance $<40 \mu\text{m}$) exhibit a wide range of chemical synaptic weights ranging anywhere from 0 to 70 synapses (with a median synaptic count of 3), whereas distant neurons (mean distance $>40 \mu\text{m}$) have a maximum synaptic count of ~ 25 synapses (with a median count of 2) (**Fig. 5c**). This choice of distance cutoff was chosen based on the observation of a distinct elbow at $40 \mu\text{m}$ in a 2D kernel density

estimate plot of the scatter data. Our findings suggest that neurons that are strongly wired together tend to be close to each other, although somatic proximity alone is not sufficient to imply strong connectivity. Recent findings in *C. elegans* have substantiated Peter’s rule: neurons with larger colocalized axodendritic regions are more likely to form connections (Cook et al., 2023). Our findings thus lend further support to this principle and suggest that close somatic or nuclear proximity also plays a role in determining neural connectivity.

Lastly, we explored the hypothesis that cell lineage is a determinant of adult cell positioning. Embryonic *C. elegans* are confined to a fixed volume within an eggshell approximately 50 μm in length and 30 μm in diameter (Riddle et al., 1997). After hatching, they grow over 4 times in length from birth ($\sim 250\text{ }\mu\text{m}$) to adulthood (over 1 mm), with an exponential expansion in their volume (Byerly et al., 1976; Stojanovski et al., 2022). Sister cells are those whose lineage differs only at the very last division. We hypothesized that animal growth should lead to both larger distances and higher variability between older sister cells that divided in the embryo versus younger sister cells born much later at postembryonic larval stages of development. Surprisingly, we found no statistically significant correlation between the time of cell division and nuclear distance or between the time of cell division and distance variability as measured by the scanning electron microscopy (SEM) (**Fig. 5D**). In fact, most sisters remained within 15 μm of each other (~ 3 nuclei apart) at adulthood, regardless of when they were born. Strikingly, a substantial cohort of embryonic sisters ended up closer together at adulthood ($< 8\text{ }\mu\text{m}$) than those dividing at larval stages that occur more than 16 h later (**Fig. 5D**). Our

data rule out exponential postembryonic growth spurts as a major determinant of divergence and variability in neuron positions.

Neural identification performance increases for all laboratories and all tested algorithms when trained on a harmonized multi-lab corpus

Our previously published statistical atlas algorithm (“StatAtlas”) for automated neural identification was trained on a homogeneous dataset of 10 NeuroPAL worms (Yemini et al., 2021; Varol et al., 2020). Formerly, this 10-worm training set achieved average accuracies of 86% overall in head neurons that ranged from 50% for the ventral ganglion to 100% for the anterior pharyngeal ganglion. These accuracies facilitate neural identification, but in practice, they require substantial verification and manual correction, necessitating significant time expenditure. Moreover, the algorithm fails to generalize to datasets produced by other labs (**Fig. 6**). We tested the performance of our previously published algorithm, StatAtlas, on each of the six aggregated datasets. Initial performance on these datasets ranged from ~21% to ~65% with an average of 41%. This substantial decrease in accuracy on datasets from different labs exposes the limitation of using single-lab training sets to produce tools intended for use by different labs with different instrumentation, experimental methods, and data acquisition pipelines.

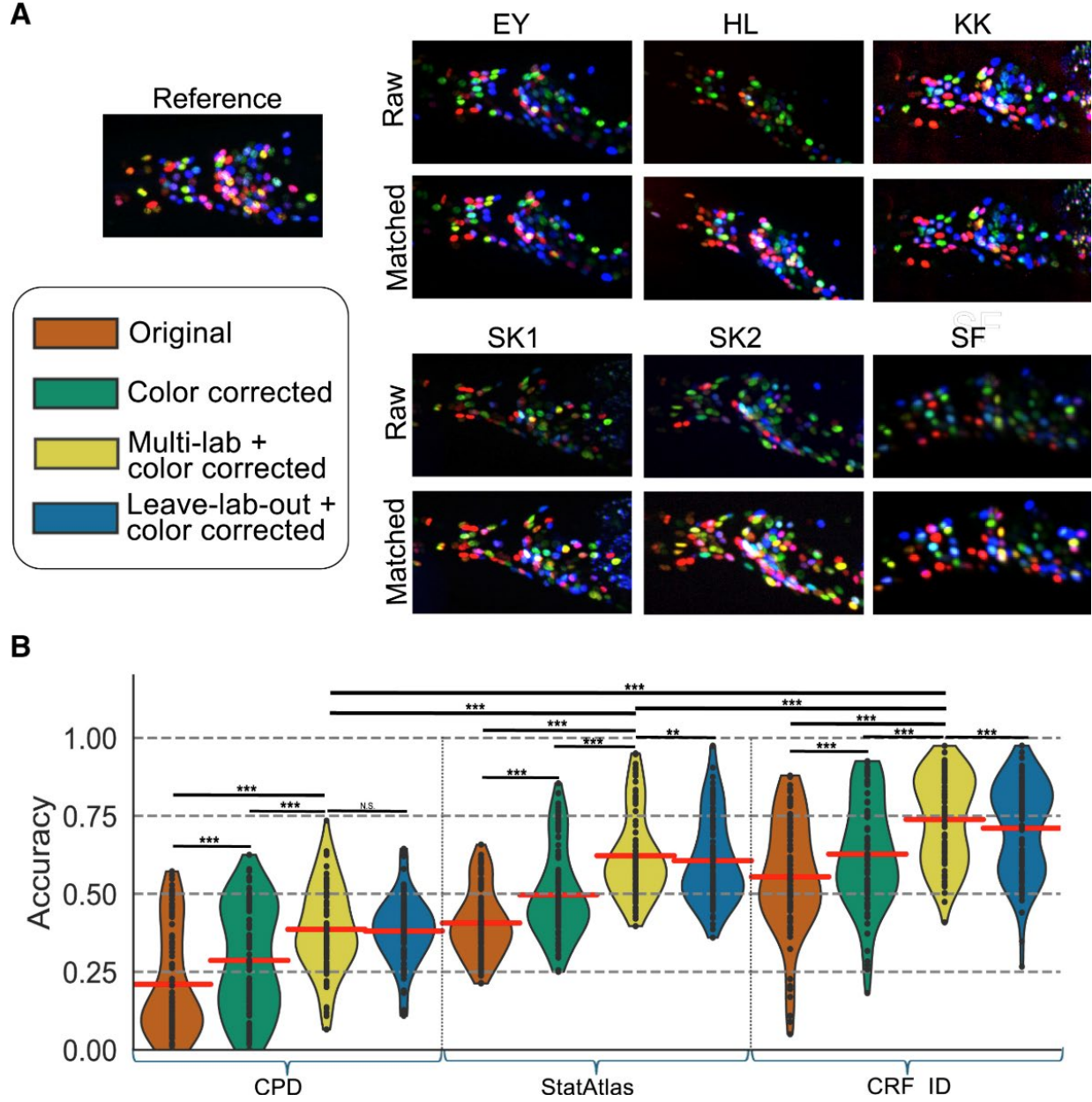


Figure 2.3.6: Improvements in neural identification accuracy

(A) Examples of raw and color-corrected (histogram matched) images from each lab and dataset. (B) Top ranked test accuracy for training set of original 10 reference worms with no color correction (orange), 10 reference worms with color correction (green), the multi-lab corpus with color correction (yellow), and training on all but the test dataset with color correction (blue) for coherent point drift (CPD, left), the statistical atlas model (StatAtlas, middle), and the conditional random field model (CRF_ID, right). All models were tested on the same 94 datasets in the corpus. For multi-lab and leave-lab-out atlases, test datasets were held out during training and then tested using a k-fold cross validation strategy. Algorithmic

performance was evaluated using paired t tests (where $N = 94$ for each test set) to compare the performance of different atlases. Significance is reported using a Bonferroni correction with the convention of * for $p < 0.05$, ** for $p < 0.01$, and *** for $p < 0.001$.

To assess the performance benefits of using a large, harmonized corpus to train commonly used automated neural identification methods, we tested two more popular algorithms: coherent point drift (CPD) (Myronenko et al., 2009) and CRF_ID (Chaudhary et al., 2021). Coherent point drift is an untrained and unsupervised algorithm that (1) globally aligns a sample point cloud of neurons to a reference atlas, then (2) locally matches points from the sample to their nearest neighbors in the atlas, and finally (3) identifies sample neurons (points) by their corresponding matches in the atlas. CRF_ID is a newer graph-based approach that identifies neurons using a combination of statistics describing their individual features (e.g., absolute position and color) and pairwise spatial relationships (e.g., displacement and angle relative to each other). There are no currently published benchmarks for neural identification using CPD. Formerly, CRF_ID demonstrated a high accuracy of 83% when originally trained and tested solely on the HL dataset. Like StatAtlas, when testing the generalizability of the CPD and CRF_ID base models on the full WormID corpus, we observed poor performance with an average overall accuracy of 39% and 59%, respectively.

CPD, StatAtlas, and CRF_ID all take a conceptually similar approach of matching a sample point cloud to a reference atlas, but they each differ in what they use as the reference. For CPD, the reference is simply a single static reference point cloud; in this case, that is a single labeled NeuroPAL image. For StatAtlas, the reference is a statistical atlas of the mean

and variance of neuron positions and colors across a corpus of data. CRF_ID extends the StatAtlas approach by including pairwise statistical spatial features such as the likelihood that one neuron is anterior to another or the relative distances between neuron pairs. These three approaches leverage increasingly more statistical information derived from our comprehensive corpus of data. We thus expected that CRF_ID would perform better than StatAtlas, which would in turn perform better than CPD, and indeed this was the case (**Fig. 6b**).

After inspecting recordings from multiple labs, we hypothesized that lab-to-lab differences in color space may negatively impact algorithmic performance. Potential sources of color space variability include differences in microscope hardware, software and image settings, and configuration of the optical path. Anecdotally, in addition to these known sources of variability, researchers also typically adjust exposure, contrast, and other channel display parameters to make the composite rendered colors appear more like the images in the NeuroPAL reference manual.⁵² In aggregate, this suggested that harmonizing the color space might aid automated algorithms.

Therefore, we developed an approach to match the color histogram of a sample image to a reference histogram representing ideal coloring. Histogram-matching the original small training set improved the accuracy of all three tested algorithms by an average of 8%, 9%, and 7% for CPD, StatAtlas, and CRF_ID respectively. Qualitatively, it also made composite color renderings better match the NeuroPAL reference manual, aiding users in annotating and correcting algorithmic predictions (**Fig. 6a,b**).

Given this success on the original small training set, we used the histogram-matched images to train the StatAtlas and CRF_ID algorithms on the full corpus of data. Test accuracy is reported using 5-fold cross-validation where each worm is tested against an atlas that excluded that worm from the training set. For CPD, we updated the algorithm to select the best template out of the full corpus. This led to significant improvement in accuracy across algorithms (**Fig. 6B**), with an average improvement of 17%, 22%, and 18% for CPD, StatAtlas, and CRF_ID, that further raised average predictive accuracy from 22% to 39%, 41% to 62%, and 55% to 74% for them respectively. This is equivalent to a $\sim 1.3\times$, $\sim 1.6\times$, and $\sim 1.7\times$ reduction in error rate. Accuracy reached as high as 95% for several individual datasets when tested with both StatAtlas and CRF_ID. Furthermore, when considering the top 5 neural identity assignments (rather than just the top 1), the multi-lab models showed average accuracies of 65%, 86%, and 89% for CPD, StatAtlas, and CRF_ID, respectively, with some datasets reaching 100% accuracy for both StatAtlas and CRF_ID (**Fig. 6b**). In addition, we saw similar improvements in accuracy across most datasets for StatAtlas and CRF_ID when training on all except one dataset and then testing on the left-out dataset (**Fig. 6b**). This indicates that most of the benefits from retraining come from achieving a better representation of the full diversity across datasets, rather than capturing the specific nuances of any one dataset. This generalizability should enable labs to use these retrained algorithms out of the box rather than needing to do additional fine-tuning on their own data.

Differences in accuracy between datasets may have been caused by a variety of factors including poor initial alignment, optical quality, non-neuronal artifacts in the images, and

nonlinear deformations of the worm body. Additionally, datasets with fewer annotated neurons had better automatic labeling accuracy, presumably because experimenters only labeled the easiest neurons to identify and left the hardest ones unannotated.

2.3.3 Discussion

Aggregation and harmonization of data from a variety of different sources is necessary to build a corpus for analytical methods and machine-learning tools that generalize across the diversity of real-world data. In this work, we present a data-harmonization pipeline for analyzing whole-brain structural and activity imaging in *C. elegans*. This pipeline includes data aggregation, conversion to a standardized file format, software for analyzing these standardized datasets, pre-processing approaches to align images and color spaces, and spatial registration of sample neuron point clouds to a common atlas.

We used this corpus to study potential biological factors that organize cell position in *C. elegans*. Specifically, we find that (1) restrictions in bounding tissue and ganglionic space likely contribute to variability in neuron positions, (2) neurons with somatic distances less than $\sim 40\text{ }\mu\text{m}$ of each other show higher synaptic connectivity, and (3) sister neurons that divide in the embryo can be found closer together at adulthood than ones dividing at larval stages more than 16 h later. The positive relationship between synaptic connectivity and neuron somatic proximity thus augments the previously observed correlation of synaptic connectivity to axodendritic adjacency, termed Peter's rule. Moreover, the close distances and low positional

variability we measured for embryonically born sister neurons rules out exponential organismal growth as a major cause in driving neurons apart from each other during the establishment of the adult Bauplan.

We then used the corpus to train machine-learning algorithms to automate the intensive task of labeling cells in these datasets, producing an updated statistical atlas of neuron positions and colors. The harmonized data corpus substantially boosts generalized neural identification performance across datasets from contributing labs for each tested algorithm, despite the variability in data from these different groups. Full recovery of human labeling was achieved in the top 5 automatic label assignments for 12 out of 94 datasets using either the StatAtlas or CRF_ID algorithms. This major improvement in accuracy compared to using older atlases indicates that training on our much larger corpus of data allows these systems to learn a more accurate representation of the true distribution of neuron position and color across individuals. CRF_ID performs the best of the three algorithms we tested, which is in line with our expectation because CRF_ID leverages pairwise positional features rather than only the mean and variance of individual neurons. In the future, our corpus can be used to incorporate neuronal shape- and size-based descriptors as well as dynamical time-series features to further improve neural identification algorithms. As new automated labeling algorithms continue to be developed using the tools outlined in this paper, there is the additional possibility of using ensemble methods to improve neuron labeling by leveraging multiple algorithms.

The wormid.org tools and resources are readily applicable to new whole-brain structural and activity imaging datasets, and these new datasets can be easily added to the existing corpus. These tools streamline public data sharing to facilitate both open science and to satisfy data-sharing mandates. We hope this resource will continue to grow in size and breadth to enable the development and benchmarking of new machine-learning tools and algorithms. Our analyses of cell features based on the full corpus of data can immediately be used to inform better feature selection and algorithms to continually improve automated approaches for neuron-subtype identification in volumetric images. Additionally, the large corpus and trained statistical atlas can serve as a descriptive resource of the underlying neurophysiology of *C. elegans*. Moreover, our resources can be incorporated into computational neurobiology courses, such as the Neuromatch Academy (neuromatch.io) to train the next global generation of neuroscientists on real-world datasets. As the community continues to develop new tools, this corpus will allow these new tools to be benchmarked for generalizable performance, spurring innovation.

As of January 2023, the United States National Institute of Health (NIH) has instituted a Data Management and Sharing Policy, and other agencies such as the European Research Council (ERC) have instituted similar policies. Compliance with these policies can require substantial thought, effort, and cost on the part of investigators. The wormid.org tools and resources provide a free and simple mechanism to satisfy these new policies. Furthermore, the DANDI repository permits data embargoes, thus providing researchers with a means to manage timing for the public release of their data. Our standard and resource should significantly aid researchers in policy compliance and facilitate open science and data sharing.

As the community continues to scale up the generation of neural data and increasingly relies on machine learning analysis to tame this “big data,” there is an ever-growing need to unify disparate datasets to produce verifiably robust, accurate, and generalizable analytical approaches. Harmonization efforts such as ours can significantly reduce the activation energy necessary for collaboration, data sharing, and the development of unified community-wide tools across labs. While some of the resources we created are specific to *C. elegans*, the framework and much of our toolkit can be applied to other model organism imaging communities.

Limitations of the study

At present, whole-brain activity imaging in *C. elegans* is performed using nuclear-localized calcium sensors (such as NLS-GCaMP6s) rather than cytosolic or membrane-bound sensors, due principally to the ease of cell segmentation afforded by nuclear localization of fluorophores.

Voltage sensors in *C. elegans* are immature, at the time of this work. The NWB format can be extended to accommodate higher-resolution subcellular activity data as they become available.

Although we identified developmental and anatomical factors that correlate with neural position (ganglionic boundaries, timing of cell divisions, and connectivity), it remains experimentally challenging to perturb these factors to probe their causal influence.

2.3.4 Methods

NWB is an HDF5-based format built specifically for neurophysiology data and has emerged as the de facto standard for storing neurophysiology datasets with associated metadata for reuse and sharing. NWB provides object types for data and metadata including acquisition parameters, segmentation of 3D image regions, fluorescent time series (e.g., for neural activity), experimental design information, multichannel electrophysiology time series data, 3D images, stimulus events during an experiment, and behavioral data (Rübel et al., 2022).

The base NWB schema supports two-dimensional structural and time-series multi-channel images but did not originally support the type of five-dimensional (multi-channel, volumetric, time-series) data that is used in *C. elegans* whole-brain activity imaging or other metadata associated with these types of experiments. To solve this problem, we developed ‘ndx-multichannel-volume’ as a novel extension to the existing Neurodata Without Borders (NWB) standardized file format. More information and resources about NWB can be found at nwb.org.

Our extension adds new objects built from existing ones in the schema to add and improve support for multi-channel, volumetric, time-series images and the metadata associated with those images as well as volumetric segmentation data and metadata fields specific to *C. elegans* (e.g., cultivation temperature and growth stage). This extension and the datasets presented in this work represent, to the best of our knowledge, the first applications of the NWB data format to *C. elegans* and have now been incorporated as the standard for this model

organism. This extension is flexible, open-source, and can be continuously updated to incorporate new types of data for future experiments.

Storage on DANDI archive

Data and associated metadata were uploaded to the DANDI archive [RRID:SCR_017571] using the Python command line tool (<https://doi.org/10.5281/zenodo.3692138>). The data were first converted into the NWB format (<https://doi.org/10.1101/2021.03.13.435173>) and organized into a structure akin to the Brain Imaging Data Structure (BIDS) (<https://doi.org/10.1038/sdata.2016.44>).

All datasets can be streamed or downloaded from the DANDI archive, available on WormID.org as well as these individual URLs:55,56,57,58,59,60,61

Original NeuroPAL: <https://doi.org/10.48324/dandi.000715/0.241009.1514>.

EY: <https://doi.org/10.48324/dandi.000541/0.241009.1457>.

HL: <https://doi.org/10.48324/dandi.000714/0.241009.1516>.

KK: <https://doi.org/10.48324/dandi.000692/0.240402.2118>.

SF: <https://doi.org/10.48324/dandi.000776/0.241009.1509>.

SK1: <https://doi.org/10.48324/dandi.000565/0.241009.1504>.

SK2: <https://doi.org/10.48324/dandi.000472/0.241009.1502>.

We present two software packages with user-friendly GUIs to interface with NWB datasets and run standard analysis pipelines for neural detections/segmentation, identification, tracking, and extracting time-series of neural-activity traces annotated with any experimental stimuli presented. First, we present the NeuroPAL_ID software (**Fig. 3b**) for visualization, annotation, neuronal segmentation and identification, neural tracking, activity trace extraction and stimulus presentation data of volumetric NeuroPAL images and whole-brain activity. This software is pre-compiled for use on MacOS and Windows. The software is open-source, available from https://github.com/Yemini-Lab/NeuroPAL_ID/releases, and is written in MATLAB and Python. It has now been updated to include functionality described in this paper to enable histogram matching, color-corrected image visualization, and automated neural identification using the new StatAtlas. This software is written and managed by the Yemini Lab; further information can be found at <https://www.yeminilab.com/neuropal>.

Second, we present the eats-worm software (**Fig. 3b**) for visualization, segmentation, and activity extraction of neural-activity time series from immobilized worms. eats-worm similarly allows for manual verification and curation of the automatic segmentation and tracking algorithms. The tracking algorithm was optimized for tracking neurons across frames in immobilized worms, but there are currently efforts to extend this functionality to work for freely-moving worms as well. eats-worm is written in Python and is built as a plugin to napari, a popular 3D visualization tool. This software is written and managed by the Kato Lab; further information can be found at <https://github.com/focolab/eats-worm>.

Both software programs have embedded functionality to read and write NWB files. NWB I/O functionality enables a user to quickly run similar analyses on all the datasets presented in this work without the need to develop specific pipelines to read in data from each dataset. Furthermore, this functionality can be easily embedded into MATLAB or Python-based analysis software. All other code used for data conversions and analysis can be found at <https://github.com/focolab/NWBelegans> and is publicly available.

Data acquisition

NeuroPAL structural volumes and neural activity time series volumes were acquired using the protocols outlined in Yemini et al. 2021. After collection of these images, neurons were marked and annotated according to the guidelines in the NeuroPAL manual (Yemini, 2022). Specific immobilization methods, microscope setup, and experimental protocols differ slightly between datasets. All datasets were taken using spinning disk confocal microscopes with xy (lateral) resolution varying from 0.1604–0.54 $\mu\text{m}/\text{pixel}$ and z (axial) resolutions varying from 0.54–1.5 $\mu\text{m}/\text{pixel}$. xy resolution was the same for NeuroPAL structural images and neural activity images (using GCaMP6s) for all datasets, but z resolution varied from 0.54 to 3 $\mu\text{m}/\text{pixel}$. z resolution is generally lower for neural activity images due to limitations in optical sectioning with confocal microscopes. Lower z resolution also reduces the number of frames needed to record a full volume for a single time-point, to allow imaging at a higher temporal resolution. Most images were taken with the worm immobilized in a microfluidic chip

with the exception of the KK dataset (where worms were semi-restricted in a microfluidic device) and the SF dataset (where worms were freely moving). The NWB files and the DANDI datasets that hold them contain metadata for the specific setup and conditions in each dataset. For published datasets, additional information can be found in the associated publications (Yemini et al., 2021; Atanas et al., 2023).

After acquisition of NeuroPAL structural volumes and whole-brain activity time-series, images were annotated using various automated detection/segmentation algorithms ranging from classical computer vision approaches (e.g., template matching) (Gonzales et al., 1977) to deep neural network approaches. These were then manually verified. Ground truth annotations were done using a combination of existing automated identification algorithms followed by manual corrections. Each neuron identity label was either explicitly annotated by experts or manually verified after algorithmic identification. Note that varying levels of completeness in labeling are due to the difficulty of this manual annotation task. For several datasets with lower image quality, even experts could only confidently label 30–50% of segmented neurons in the volume. For neural activity time-series, neuron centers were first tracked across images using various algorithms and then manually verified by experts (Emmons et al., 2021; Wen et al., 2021). Fluorescence activity is then extracted from these tracked ROIs to obtain time series of neural-activity traces. Neurons in the NeuroPAL structural volume were then matched to the ROIs in the neural activity time-series to get labeled activity traces.

Datasets from various labs were converted to the NWB standardized file format using the *ndx-multichannel-volume* extension presented in this work. These files were then uploaded to the DANDI archive where they are now publicly accessible for data streaming, download, or online visualization.

Butterfly plot

To produce the butterfly plot, we first manually found three orthogonal basis vectors to align neuron point clouds to a new cartesian coordinate space. To do so, we used human-guided affine transformation to roughly align these basis vectors to the anterior-posterior, dorsal-ventral, and left-right axes. The XYZ coordinates of each neuron were projected into this new cartesian coordinate space and then converted to cylindrical coordinates by the following equations:

$$x_{cylinder} = -x_{new}$$

$$r = \sqrt{y_{new}^2 + z_{new}^2}$$

$$\theta = \arctan 2(y_{new}, z_{new})$$

We plotted the new x and θ coordinates on a 2D plane to obtain the butterfly plots shown in (**Figures 1A and 4**). This projection is akin to flattening the positions of the neurons along the circumference of a cylinder of the worm body and then unrolling that cylinder into a flattened plane.

Histogram matching

We modified the established approach of histogram matching to apply it to 3-D volumetric, multi-channel data (Gonzales et al., 1977). We created a reference histogram using the 10 worms from the original NeuroPAL work. This data is stored as uint16 (65,536 possible values for each pixel). For each channel, we created a histogram counting the number of pixels within the bin edges, assigning each color value its own bin, and then averaged the values in each of these bins across the 10 images. Practically, these histograms were very similar across these 10 datasets, so the averaged histogram was similar to each of the individual histograms.

To color match a new animal sample, we calculated a histogram for each channel. The number of bins for each channel histogram was equal to the maximum intensity value present in that channel in the image: $bincount_{channel} = \max(x_{channel})$. This meant that there was a different number of histogram bins for each channel in each image because images were collected at different bit depths and with varying levels of saturation.

We then calculated the cumulative density function (CDF) at each color value of both the sample and the reference. We created a lookup table M to associate each gray count value x in the sample to the color value in the reference with the closest CDF value. Next, we created a new matched image with each pixel transformed into the new color space using this lookup table as shown below:

$$M_{channel}(x) = (|cdf_{sample}(x) - cdf_{reference}(x')|)$$

$$MatchedImage_{channel}(i, j, k) = M_{channel}(A(i, j, k))$$

Color extraction

To extract the color values for the neurons in each image, we first calculated the mean and standard deviation of the pixel counts in each channel, and then converted each pixel value into its Z score based these channel values. We then sampled a $3 \times 3 \times 1$ grid of pixel values around each segmented neuron center in each channel. We use the median values of this $3 \times 3 \times 1$ grid as the RGB values for that neuron center. Color values were extracted post-histogram matching when training or testing using histogram-matched images. For non-histogram-matched images, there are no additional color pre-processing steps beyond Z-scoring.

Positional variability analysis

We calculated pairwise positional variability by measuring the Euclidean distance between every pair of canonical head neurons across each structural volume when both neurons in that pair had a ground truth label. We then took the average and standard deviation of these distances for each neuron pair to find mean nuclear distance and pairwise positional variability, respectively. For these analyses we ignored pairs that are not present in at least 5 datasets. We

used pairwise positional variability instead of absolute positional variability because absolute position is extremely sensitive to point-cloud realignment, which would make it hard to disaggregate natural positional variability from alignment errors; furthermore, we are interested in how individual cells vary relative to each other, not in how they vary individually. To get the mean positional variability for a given neuron, we averaged the mean pairwise distance for all neuron pairs that contained that neuron.

We calculated Intra vs. inter ganglion measures as follows: for every neuron in the atlas, we found its n closest atlas neighbors and only performed measurements for these pairwise neighbors. We then separated these pairings based on whether the two neurons in the pair are within the same ganglion or in two different ganglia. Note that pairs in different ganglia will appear twice: e.g., if one neuron in the pair is in the anterior ganglion and the other is in the lateral ganglion, the pair will be counted in the analysis for both the anterior ganglion and the lateral ganglion (**Fig. 5b**). Pairs within the same ganglion are only counted once. We compared this approach for $n = 1-20$. For all numbers of neighbors there is higher positional variability for neighbors in different ganglia when compared to those in the same ganglion. The measurements stabilize around $n = 7$ and hold steady through $n = 20$. Therefore, we selected 10 to use for n in our analysis.

Synaptic connection weights between neuron pairs are derived from the whole-brain connectome of the adult hermaphrodite in Cook et al. 2019

For lineal distance: the cell lineage tree and associated birth times were taken from Sulston et al. 2021. The last shared parent cell between two neurons is the most recent shared parent node in the lineal tree. We used the birth time of the last shared parent cell between two neurons as the lineal distance and explored the relationship between this lineal distance and mean pairwise nuclear distance (**Fig. 5d**). In this analysis, we focus only on sister cells: terminal cells that only divided from each other at the very last stage of their lineal tree.

Alignment of datasets

Before analysis, datasets were first roughly aligned to a common space by learning an affine transformation as described in Varol et al. 2020. The algorithm chooses a reference dataset from the 10 original NeuroPAL worms to align every other dataset to. For each dataset, the algorithm then learns an affine transformation that minimizes the distance of each ground truth labeled neuron to the same neuron in the reference dataset. This roughly aligned the principal axes of the worm as well as the overall scaling across datasets. The accuracy of each automated identification algorithm is highly sensitive to pre-alignment and thus this step is critical for direct comparison of models and quantification of atlas performance. Pre-alignment is accomplished by labeling a small number of easily identifiable neurons.

Coherent point drift (CPD)

Coherent point drift has been a common algorithm for registering two similar point clouds to each other since its introduction in Myronenko and Song 2009. CPD allows for both rigid and non-rigid point set registration. CPD models one point set as a set of GMM centroids that are fit to the second point set by maximizing the likelihood. GMM centroids are set to move coherently to preserve the structure of the point clouds. In the rigid case, the algorithm learns an affine transformation of the GMM centroid locations while in the non-rigid case, the algorithm learns a displacement function on the original centroid positions with an enforced regularization term to enforce smoothness. The objective function is optimized using an iterative EM optimization approach and yields both the aligned point set as well as an $N \times M$ correspondence probability matrix that represents the likelihood that each point n in set 1 corresponds to each point m in point set 2.

In this paper, we use the specific implementation of CPD used in Yu et al. 2021. First, rigid CPD is used to roughly align a test worm point cloud to a template point cloud. Then, non-rigid CPD is used to model non-linear deformations between the semi-aligned test and template. Neuron assignments are then determined by creating a matrix of pairwise Euclidean distances between every neuron’s position and color in the test and every neuron in the template in the aligned space. We then use the Hungarian algorithm on this distance matrix to find the optimal label assignments. To get 2nd ranked assignments, we assigned an infinite cost to each

label assignment from the first pass and reran the Hungarian algorithm. We repeat this for the 3rd-5th order assignments.

Accuracy was calculated by counting the number of neurons whose algorithmic assignment was the same as the ground truth label and then dividing by the total number of neurons that have a ground truth label. Note that neurons without a ground truth label were not included in the accuracy metric but are still part of the cost matrix and received neuron assignments. Since there is no ground truth for these neurons we did not determine the accuracy of their label assignments.

The difference between the use of the ‘original (10) worms ’versus ‘multi-lab corpus ’for CPD is what set is included in possible options for the template. For the ‘original ’group, we compare every test set to each of the original 10 NeuroPAL worms and report the accuracy for the template that has the highest average probability of correspondence after the rigid alignment step. Similarly, for the ‘multi-lab corpus’, each test worm is compared to each possible template worm in the whole multi-lab corpus and accuracy is reported similarly. The accuracy of CPD is highly sensitive to a good rough initial alignment and to similarity of the template and the test point cloud. The template with the highest average probability of correspondence is not necessarily the template that yields the highest accuracy, but it is the template that the algorithm has the highest confidence that it has found the ‘correct ’correspondence.

Statistical atlas training and inference

Statistical atlases used for testing performance were trained using the algorithm described in Varol et al. 2020. This algorithm uses a training set of neuron point clouds with both XYZ and RGB values and takes a block-coordinate descent approach where it iteratively learns affine transformation parameters to align the neuron point clouds, then updates the means and covariances of the positions and colors of each neuron until reaching convergence. This process generates mean and covariance parameters for each neuron as well as an aligned coordinate space for all the worms in the training set. The trained atlas consists of a list of neuron names alongside their associated means and covariances in the aligned position and color space.

We trained three atlases: the original atlas trained on just the original 10 NeuroPAL worms from Yemini et al. 2021, the color corrected atlas trained on these same 10 worms after histogram matching, and the multi-lab + color-corrected atlas which is trained on the full corpus of histogram-matched data.

For the original atlas, we tested every dataset in the full corpus without histogram matching. For the color-corrected atlas, we similarly tested every dataset on the full corpus of data with histogram matching. For the atlas trained on the full corpus, we use k-fold cross-validation. The corpus was split into five equally sized groups. For each group, an atlas was trained on all datasets in the other four groups and performance was reported for the out-of-training set group. The 10 worms used to train the original and color-corrected atlas were included in the training for each of these five groups. These 10 worms were not used to report

testing accuracy for any of the atlases (Fig. 6). The fully trained atlas presented in Figure 4 was trained using the 10 original worms and the full corpus of data presented in this work, without splitting it into groups. This full atlas was embedded into the “Auto ID” functionality of the NeuroPAL ID software shown in **Fig. 3a**.

Neuron point clouds used for testing were pre-aligned by learning an affine transformation from each sample dataset to the aligned coordinates of the atlas based on a subset of the ground truth labeled neurons in the sample. Briefly, assuming N neurons in the test sample and M neurons in the atlas, we calculated an NxM cost matrix using the Mahalanobis distance between each neuron center in the sample and each neuron distribution in the atlas. $\underline{x}_i$ represents the XYZRGB values of neuron i, while $\underline{\mu}_j$ and $\underline{\Sigma}_j$ represent the XYZRGB mean and covariance respectively for neuron j in the atlas:

$$Cos\ t_{i,j} = \left(\underline{x}_i - \underline{\mu}_j \right)^T \underline{\Sigma}_j^{-1} \left(\underline{x}_i - \underline{\mu}_j \right)$$

We then treated this cost matrix as a linear sum assignment problem. Label assignments (for neural identification) were calculated using the Hungarian algorithm (Kuhn, 1955). 2nd-5th order ranked assignments and accuracy are calculated in the same way as described for CPD.

CRF_ID training and inference

CRF_ID atlases and inference are conducted using the algorithm described in Chaudhary et al 2021. This approach follows a probabilistic-graphical-model framework based on conditional random fields. The graph is defined by node features corresponding to unary measures for each neuron center such as position and color and edge features corresponding to pairwise measures for each pair of neurons such as distance, relative angle, or probability that one neuron is anterior to the other. After features are selected, a data-driven atlas is trained on a corpus of data to determine the average values for each of the measured features; then for a test worm, node and edge potentials are calculated based on comparison of each feature in the test worm to the atlas and infer the best global assignment of labels by maximizing an energy function using an approximate inference method. For the analysis in this work, we used the color information solely to define the node potentials, and the pairwise angle relationships only to define the edge potentials. Optimizing the weights of the node and edge features may result in a higher prediction accuracy.

We trained three atlases: the original atlas trained on just the original 10 NeuroPAL worms from Yemini et al. 2021, the color corrected atlas trained on these same 10 worms after histogram matching, and the multi-lab + color-corrected atlas which is trained on the full corpus of histogram-matched data. This training approach follows the same k-fold cross validation approach used for the StatAtlas method.

We use the roughly pre-aligned point clouds used in the StatAtlas algorithm as input to the CRF_ID algorithm to eliminate possible differences in the initial alignment step, which can dramatically change accuracy.

In practice, there are nearly always fewer detected neuron centers in a given image than total cells in the atlas. CRF_ID handles this by modeling a hidden variable $\mathbf{h} \in \{0,1\}^N$ where N is the number of neurons in the atlas. This variable specifies the probability that a given cell is missing in the image. Based on the number of cells in the test image, P cells are uniformly selected across different regions of the head and removed from the atlas. This process is repeated ~ 1000 times to sample multiple possible combinations of $\mathbf{h}$. The top 1–5 predicted assignments are generated by compiling a list of the most frequent labels for each cell in the test image across all runs. Accuracy is reported same as CPD and CRF_ID.

Optimizing the aforementioned energy function using an approximate inference method produces marginal distributions of label assignments for each cell. The top 1–5 predicted labels for each cell were generated by sorting the marginal probability of labels in a descending order. The label that resulted in the highest marginal probability was assigned as top 1.

Quantification and statistical analysis

Descriptions of statistical tests can be found in the text or figure captions where the analysis was conducted. All analyses were done in Python 3.12. Specific packages and versions used in

analyses can be found in the `setup.py` file in the NWBelegans Github repository <https://github.com/focolab/NWBelegans>.

We present WormID.org as an additional resource to guide users through the full process of collecting data, converting to the NWB format, uploading to DANDI, and replicating the analyses in this work. The website contains tutorials, links to code and data, and step-by-step instructions for data conversion.

2.4 Pycro-Manager: open-source software for customized and reproducible microscope control

Cutting-edge innovations in biological microscopy are increasingly blurring the line between data acquisition and data analysis. Computational microscopy and machine-learning-based methods take this paradigm to an extreme, often producing raw measurements that are not human-interpretable without postprocessing (Boominathan et al., 2016; Gustafsson et al., 2000; Pavani et al., 2008). Furthermore, new data-adaptive imaging methods rely on data processing during acquisition to actively control various hardware settings of the microscope.

Testing new ideas and applying them for biological discovery is often impeded by a lack of control software that is capable of meeting demands for speed and performance, integrating new and diverse types of hardware, providing the flexibility to adapt in real time to the data being captured, and providing user-friendly programming interfaces. As a result, researchers often end up developing custom software that works only with specific instruments, using closed-source and/or proprietary programming languages. In addition to slowing development, this lack of a consistent software ecosystem creates barriers to reproduction and replication of new techniques. These include the increased likelihood of bugs in code that is not reused and tested in different contexts, the burden on new users of understanding new code bases, and the extra work of disentangling high-level code that is not properly abstracted from the underlying hardware. In many fields the Python programming language — in particular, the NumPy and SciPy ecosystem (Harris et al., 2020; Virtanen et al., 2020) — has emerged as a common

framework for reproducing research results that rely on complex, multilayer scientific workflows in a portable, scriptable form⁶ that is accessible to researchers with various levels of programming experience.

The central challenge of such a consolidated approach in microscopy is the diversity of hardware used in different types of microscopes, ranging from custom-built components on optical tables to turnkey commercial systems. `µManager` (Edelstein et al., 2010; Edelstein et al., 2014) is an essential tool for controlling microscopes, thanks to an extensive library of ‘device adapters’ for controlling different types of hardware, from cameras to complete microscopes, with a single programming interface. Community contributions of device adapters, plug-ins and scripts provide hundreds of developer-years of microscopy automation.

Despite the power of these libraries, which are written in C++ and Java, they are often difficult to integrate with the latest developments in computer vision and scientific computing, which most readily interface with NumPy. This not only increases the difficulty of developing techniques that rely on both customized data capture and data analysis, but also hinders the dissemination and adoption of these techniques by fragmenting them across multiple tools and programming languages and making them harder to understand and test in new contexts.

Many groups are developing microscopy control software that is written mostly or entirely in Python, including, Python Microscope (Miguel et al., 2021), Python Microscopy Environment (<http://www.python-microscopy.org/>) and Tormenta (Barabas et al., 2016) (a complete list is at <https://doi.org/10.5281/zenodo.4433237>). However, these tools are currently

limited to specific types of microscopy (for example, super-resolution) and/or support a relatively small number of hardware devices. Thus, despite the potential benefits of such a design, they face an enormous task to match μ Manager's flexibility and device support.

To address these needs, we present here Pycro-Manager. Pycro-Manager is built on a translation layer that converts Java objects, functions and data into language-agnostic messages that are reconstituted as Python objects or functions and as NumPy arrays

(**Fig. 1a** and **Supplementary Information**). As a result, the existing capabilities of μ Manager can be called as if they had been written in Python, without forcing users to learn Java, forcing Java developers to learn Python, or abandoning of the relative strengths of either language (**Supplementary Information**).

In addition, Pycro-Manager provides a new library of high-level programmatic building blocks (in Python) for customizing data acquisition. These are designed to facilitate creation of complex data acquisitions with concise, readable code while maintaining the flexibility required for customization and the ability to handle multi-terabyte datasets acquired at speeds over 1 GB s⁻¹ (**Supplementary Information**).

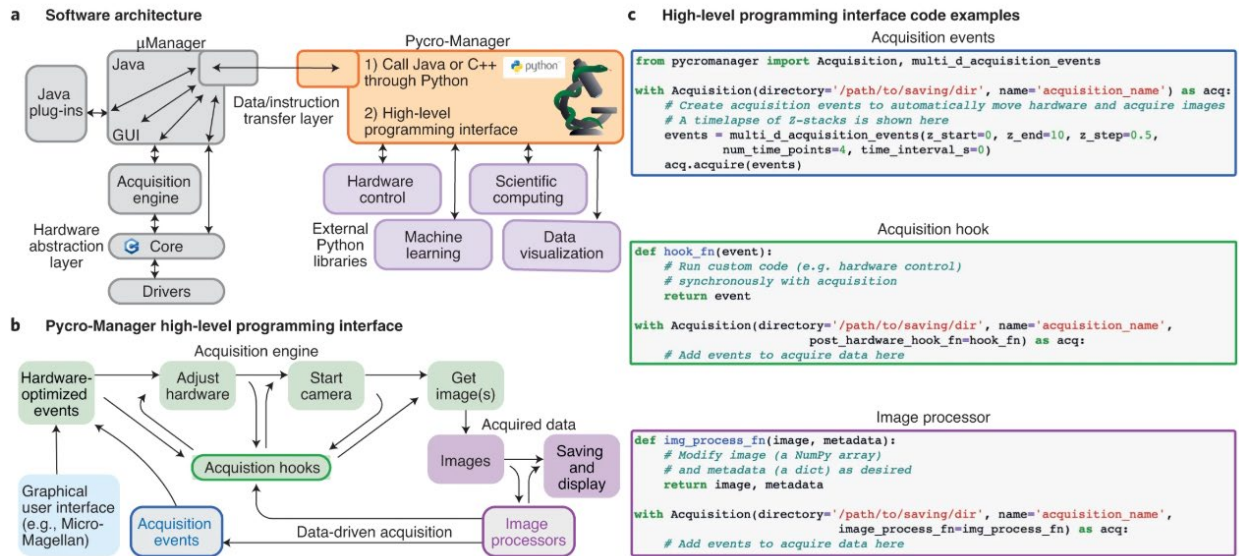


Figure 2.4.1: Pycro-Manager design (via Henry Pinkard)

(a) Software architecture overview. Grey: the existing parts of μ Manager provide generic microscope control abstracted from specific hardware, a graphical user interface (GUI), a Java plug-in interface, and an acquisition engine that automates various aspects of data collection. Orange: Pycro-Manager enables access to these components through Python over a network-compatible transport layer, as well as a concise, high-level programming interface for acquiring data. Purple: these provide integration of data acquisition with Python libraries for hardware control, data visualization, scientific computing and so forth. (b) Pycro-Manager’s high-level programming interface. The data acquisition process in Pycro-Manager starts with a source of acquisition events (blue), from either programming or a GUI. These events are passed to the acquisition engine (green), which optimizes them to take advantage of hardware triggering where available, sends instructions to hardware and acquires images. The resulting images are then saved and displayed in the GUI (magenta). The three main abstractions of the Pycro-Manager high-level programming interface (acquisition events, acquisition hooks, and image processors) enable fine-grained control and customization of this process. (c) Code examples. Code snippets for implementing acquisition events (blue), acquisition hooks (green) and image processors (magenta). (“Python” and the Python logos are trademarks or registered trademarks of the Python Software Foundation, used with permission from the Foundation. C++ logo from <http://isocpp.org/about>).

The high-level application programming interface (API) contains three main abstractions: acquisition events, acquisition hooks and image processors (**Fig. 1b,c**). These building blocks can be used individually or combined for more complex applications. Acquisition events enable customized instructions for how to adjust hardware when acquiring images and how to index those images along arbitrary axes (for example, time, z, etc.) for storage and display. For common microscopy workflows like time-lapses and z-stacks, events can be automatically generated by high-level functions (**Fig. 1c**). Alternatively, they can be created manually from nested Python lists and dictionaries to allow a greater degree of customization. Acquisition hooks enable the execution of arbitrary Python code concurrently at various stages of the data acquisition process. Image processors give access to the image data as soon as it is acquired, for modification or for use in data-driven feedback loops on the acquisition process.

The combination of Jupyter notebooks (<https://jupyter.org/>) — interactive documents that consolidate text, code, equations and results — and the Pycro-Manager high-level API provides a means to describe the full workflow of a research project, from data acquisition to analysis to results, thereby facilitating understanding, dissemination and reproduction of new microscopy technologies. For techniques that rely heavily on computation, these notebooks can be as valuable as (or even more valuable than) their corresponding research papers.

In the **Supplementary Information** and online documentation (<https://pycro-manager.readthedocs.io/en/latest/#>), we provide tutorials describing the basic features of Pycro-Manager, as well as Jupyter notebooks outlining sample applications in microscope

alignment, light sheet microscopy, integrated sample processing, computational microscopy and machine learning, and the control of microscopes over networks. The source code and documentation for Pycro-Manager can be found in the Supplementary Information, and future updates will be posted at <https://github.com/micro-manager/pycro-manager> and <https://pycro-manager.readthedocs.io/en/latest/#>.

Supplementary Information

See https://static-content.springer.com/esm/art%3A10.1038%2Fs41592-021-01087-6/MediaObjects/41592_2021_1087_MOESM1_ESM.pdf for the comprehensive supplementary information.

2.5 minimo: a linked data and metadata storage system for small labs

minimo is a linked data and metadata storage system for small labs. It combines object storage for large raw data with a document database for metadata. It is simple, low-overhead, and modern. All too often we want to revisit data and struggle with a loss of context: old data is hard to find, hard to understand, hard to fuse. But if good hygiene is applied to the initial data commit process – i.e., if raw data are stored alongside metadata capturing the context in which they were collected or created – much of this trouble can be avoided. *minimo* encourages this sort of hygiene by providing a customizable interface for storing, linking, and accessing raw data and metadata. It sets the stage for reproducible science in the context of small labs that need a robust but lightweight solution to hygienic data storage. Raw data and metadata should be inseparably linked. This is the central principle of *minimo*.

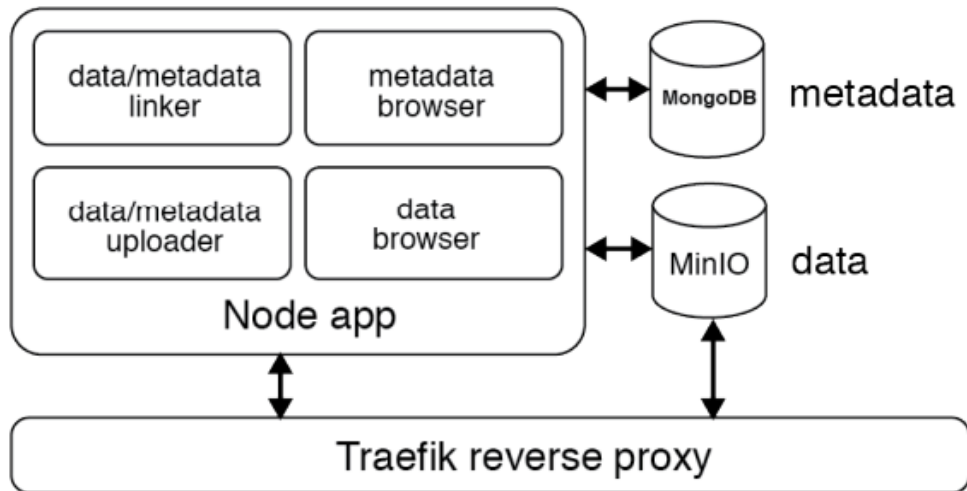


Figure 2.5.1: *minimo* architecture

minimo leverages existing open source solutions for data and metadata storage using modern development patterns, and it provides a lightweight web interface which abstracts the technical details of those solutions away from users. We use an object storage system for raw data, and a document database for metadata. It is designed to accommodate programmatic data access, to be portable across common infrastructure choices and operating systems, and to be deployable both locally and in the cloud. *minimo* consists of the following components: 1) A MinIO server for raw data storage, because big, chunky data should live in immutable object storage. 2) A MongoDB server for metadata storage, because metadata are typically small and benefit greatly from database structure and semantics to allow efficient search and selection. 3) An Express/Node server which provides a user interface for uploading, linking, and accessing raw data and metadata. 4) A Traefik server as reverse proxy for the other three components.

For portability and ease of deployment, we’ve bundled all of these components as a Docker application. This means that *minimo* can be deployed on any common operating system (Linux, Windows, or Mac) simply by cloning its code from GitHub, installing Docker and Docker Compose, and running a single Docker command. This architecture offers several key advantages: it supports very large (up to 5TB) raw data files, provides both easy human access and robust programmatic access to underlying data stores, allows for deployment on both local and cloud hardware, and is lightweight enough to be easily extensible and maintainable. We use this in our lab (Bubnis et al., 2019) and share it to support reproducible and convenient data science.

Conclusion

In conclusion, we consider our discovery of working memory in *C. elegans* to be a major advance in our understanding of the evolution of higher cognitive function. Our findings demonstrate that even in a simple, compact organism with 302 neurons has cognitive abilities which are typically associated with more complex animals. The neural oscillators that first evolved to coordinate movement may have been repurposed to maintain internal representations, an economical adaptation that could have laid the groundwork for more complex cognition. The phase-encoding mechanism we identify in *C. elegans* may represent a rudimentary form of oscillatory coupling, a mechanism broadly implicated in higher cognitive processes across phyla. Moving forward, further investigation into this system may reveal additional computational roles for these dynamics, shedding light on how simple circuits can support flexible behavior.

This work continues in a long history of, through careful experimentation, discovering advanced capabilities in reductionist model organisms. These findings challenge our previously held notion of what systems are needed for simple function. However, some reflection on the content of this memory is warranted; it may be a single bit of information (dorsal vs ventral), and these findings suggest that an output decision is stored. However deferred actions leave an opportunity for additional computation, for example multimodal sensory integration or modulation of this memory mechanism. Moving forward, it will be interesting to continue to probe this memory system to investigate additional facets of control.

The technological advances described in this dissertation—WormID, CLEF, Pycro-Manager, and minimo—collectively address critical challenges in neuroscience research: standardization, real-time experimentation, and data management. These tools not only enabled the discoveries presented here but also provide a framework for future work across laboratories. Indeed, CLEF in particular shows promise for enabling a new generation of experimentation with, in my opinion, superior explanatory power compared to traditional loss-of-function gain-of-function experimentation. However future work will need to substantiate this claim.

References

- Ahrens, M. B., Orger, M. B., Robson, D. N., Li, J. M., & Keller, P. J. (2013). Whole-brain functional imaging at cellular resolution using light-sheet microscopy. *Nature Methods*, 10(5), 413–420. <https://doi.org/10.1038/nmeth.2434>
- Aimon, S., Katsuki, T., Jia, T., Grosenick, L., Broxton, M., Deisseroth, K., Sejnowski, T. J., & Greenspan, R. J. (2019). Fast near-whole-brain imaging in adult *Drosophila* during responses to stimuli and behavior. *PLoS Biology*, 17(2), e2006732. <https://doi.org/10.1371/journal.pbio.2006732>
- Alkema, M. J., Hunter-Ensor, M., Ringstad, N., & Horvitz, H. R. Tyramine functions independently of octopamine in the *Caenorhabditis elegans* nervous system. *Neuron*, 46(2), 247–260 (2005). <https://doi.org/10.1016/j.neuron.2005.02.024>
- Atanas, A., et al. Brain-wide representations of behavior spanning multiple timescales and states in *C. elegans*. *Cell* 186(19), 4134–4151 (2023). <https://doi.org/10.1016/j.cell.2023.07.035>
- Atanas, Adam; Kim, Jungsoo; Wang, Ziyu; Bueno, Eric; Becker, McCoy; Kang, Di; Park, Jungyeon; Kramer, Talya; Wan, Flossie; Baskoylu, Saba; Dag, Ugur; Kalogeropoulou, Elpiniki; Gomes, Matthew; Estrem, Cassi; Sprague, Daniel; Cohen, Netta; Mansinghka, Vikash; Flavell, Steve (2024) Brain-wide representations of behavior spanning multiple timescales and states in *C. elegans* (Version 0.241009.1509) [Data set]. DANDI archive. <https://doi.org/10.48324/dandi.000776/0.241009.1509>

- Barabas, F. M., Masullo, L. A., & Stefani, F. D. (2016). Note: Tormenta: An open source Python-powered control software for camera based optical microscopy. Review of Scientific Instruments, 87(12). <https://doi.org/10.1063/1.4972392>
- Boominathan, V., Adams, J. K., Asif, M. S., Avants, B. W., Robinson, J. T., Baraniuk, R. G., Sankaranarayanan, A. C., & Veeraraghavan, A. (2016). Lensless Imaging: A computational renaissance. IEEE Signal Processing Magazine, 33(5), 23–35. <https://doi.org/10.1109/msp.2016.2581921>
- Brigman, K. L., Abarbanel, H. D. I., & Kristan, W. B. Optical imaging of neuronal populations during decision-making. Science 307(5711), 896–901 (2005). <https://doi.org/10.1126/science.1103736>
- Bubnis, G., Ban, S., DiFranco, M. D., & Kato, S. (2019). A probabilistic atlas for cell identification. ArXiv e-Prints. <https://arxiv.org/pdf/1903.09227.pdf>
- Buzsaki, G. (2006). Rhythms of the Brain. Oxford University Press.
- Byerly, L., Cassada, R., & Russell, R. (1976). The life cycle of the nematode *Caenorhabditis elegans*. Developmental Biology, 51(1), 23–33. [https://doi.org/10.1016/0012-1606\(76\)90119-6](https://doi.org/10.1016/0012-1606(76)90119-6)
- Chartrand, R. (2011). Numerical Differentiation of Noisy, Nonsmooth Data. ISRN Applied Mathematics, 2011, 1–11. <https://doi.org/10.5402/2011/164564>
- Chaudhary, S., Lee, S. A., Li, Y., Patel, D. S., & Lu, H. (2021). Graphical-model framework for automated annotation of cell identities in dense cellular images. eLife, 10. <https://doi.org/10.7554/elife.60321>

- Chaudhary, Shivesh; Sprague, Daniel; Lee, Sol Ah; Li, Yueyi; Patel, Dhaval S; Lu, Hang (2024)
Segmented and labeled NeuroPAL structural images (Version 0.241009.1516) [Data set].
DANDI archive. <https://doi.org/10.48324/dandi.000714/0.241009.1516>
- Chen, T., Wardill, T. J., Sun, Y., Pulver, S. R., Renninger, S. L., Baohan, A., Schreiter, E. R.,
Kerr, R. A., Orger, M. B., Jayaraman, V., Looger, L. L., Svoboda, K., & Kim, D. S.
(2013). Ultrasensitive fluorescent proteins for imaging neuronal activity. *Nature*,
499(7458), 295–300. <https://doi.org/10.1038/nature12354>
- Cheng, C., Messerschmidt, L., Bravo, I., Waldbauer, M., Bhavikatti, R., Schenk, C., Grujic, V.,
Model, T., Kubinec, R., & Barceló, J. (2024). A general primer for data harmonization.
Scientific Data, 11(1). <https://doi.org/10.1038/s41597-024-02956-3>
- Chiu, C. L., & Clack, N. (2022b). napari: a Python Multi-Dimensional Image Viewer Platform
for the Research Community. *Microscopy and Microanalysis*, 28(S1), 1576–1577.
<https://doi.org/10.1017/s1431927622006328>
- Chronis, N., Zimmer, M., & Bargmann, C. I. (2007). Microfluidics for in vivo imaging of
neuronal and behavioral activity in *Caenorhabditis elegans*. *Nature Methods*, 4(9), 727–
731. <https://doi.org/10.1038/nmeth1075>
- Cook, S. J., Jarrell, T. A., Brittin, C. A., Wang, Y., Bloniarz, A. E., Yakovlev, M. A., Nguyen,
K. C. Q., Tang, L. T., Bayer, E. A., Duerr, J. S., Bülow, H. E., Hobert, O., Hall, D. H.,
& Emmons, S. W. (2019). Whole-animal connectomes of both *Caenorhabditis elegans*
sexes. *Nature*, 571(7763), 63–71. <https://doi.org/10.1038/s41586-019-1352-7>

- Croll, N. A. (1975). Components and patterns in the behaviour of the nematode *Caenorhabditis elegans*. *Journal of Zoology*, 176(2), 159–176. <https://doi.org/10.1111/j.1469-7998.1975.tb03191>.
- Dandi. (n.d.). GitHub - dandi/dandi-cli: DANDI command line client to facilitate common operations. GitHub. <https://github.com/dandi/dandi-cli>
- De Carlos Cáceres, I., Valmas, N., Hilliard, M. A., & Lu, H. (2012). Laterally Orienting *C. elegans* Using Geometry at Microscale for High-Throughput Visual Screens in Neurodegeneration and Neuronal Development Studies. *PLoS ONE*, 7(4), e35037. <https://doi.org/10.1371/journal.pone.0035037>
- Deng, J., Dong, W., Socher, R., Li, L., Li, N. K., & Fei-Fei, N. L. (2009). ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition. <https://doi.org/10.1109/cvpr.2009.5206848>
- Deng, N. L. (2012). The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]. *IEEE Signal Processing Magazine*, 29(6), 141–142. <https://doi.org/10.1109/msp.2012.2211477>
- Dunn, Raymond; Sprague, Daniel; Kato, Saul (2024) *C. elegans* whole-brain neuroPAL and immobilized calcium imaging (Version 0.241009.1504) [Data set]. DANDI archive. <https://doi.org/10.48324/dandi.000565/0.241009.1504>
- Dunn, Raymond; Sprague, Daniel; Borchardt, Jackson; Bubnis, Greg; Kato, Saul (2024) NeuroPAL volumetric images (Version 0.241009.1502) [Data set]. DANDI archive. <https://doi.org/10.48324/dandi.000472/0.241009.1502>

- Dobosiewicz, M., Liu, Q., & Bargmann, C. I. Reliability of an interneuron response depends on an integrated sensory state. *eLife*, 8 (2019). <https://doi.org/10.7554/elife.50566>
- Docker, Inc. (2020). Docker compose: Define and run multi-container applications with docker. In GitHub repository. GitHub. <https://github.com/docker/compose>
- Edelstein, A., Amodaj, N., Hoover, K., Vale, R., & Stuurman, N. (2010). Computer Control of Microscopes Using μ Manager. *Current Protocols in Molecular Biology*, 92(1). <https://doi.org/10.1002/0471142727.mb1420s92>
- Edelstein, A. D., Tsuchida, M. A., Amodaj, N., Pinkard, H., Vale, R. D., & Stuurman, N. (2014). Advanced methods of microscope control using μ Manager software. *Journal of Biological Methods*, 1(2), 1. <https://doi.org/10.14440/jbm.2014.36>
- Emmons, S. W., Yemini, E., & Zimmer, M. (2021). Methods for analyzing neuronal structure and activity in *Caenorhabditis elegans*. *Genetics*, 218(4). <https://doi.org/10.1093/genetics/iyab072>
- Engel, A. K., Fries, P., & Singer, W. (2001). Dynamic predictions: Oscillations and synchrony in top-down processing. *Nature Reviews. Neuroscience*, 2(10), 704–716. <https://doi.org/10.1038/35094565>
- Express Technical Committee. (2020). Express: Fast, unopinionated, minimalist web framework for node. In GitHub repository. GitHub. <https://github.com/expressjs/express>
- Flavell, S. W., & Gordus, A. (2022). Dynamic functional connectivity in the static connectome of *Caenorhabditis elegans*. *Current Opinion in Neurobiology*, 73, 102515. <https://doi.org/10.1016/j.conb.2021.12.002>

- Gonzales, R., & Fittes, B. (1977). Gray-level transformations for interactive image enhancement. *Mechanism and Machine Theory*, 12(1), 111–122. [https://doi.org/10.1016/0094-114x\(77\)90062-3](https://doi.org/10.1016/0094-114x(77)90062-3)
- Gray, J. M., Hill, J. J., & Bargmann, C. I. A circuit for navigation in *Caenorhabditis elegans*. *PNAS*, 102(9), 3184–3191 (2005). <https://doi.org/10.1073/pnas.0409009101>
- Gustafsson, M. G. L. (2000). Surpassing the lateral resolution limit by a factor of two using structured illumination microscopy. *Journal of Microscopy*, 198(2), 82–87. <https://doi.org/10.1046/j.1365-2818.2000.00710.x>
- Harris, C. R., Millman, K. J., Van Der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Van Kerkwijk, M. H., Brett, M., Haldane, A., Del Río, J. F., Wiebe, M., Peterson, P., . . . Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357–362. <https://doi.org/10.1038/s41586-020-2649-2>
- Harris-Warrick, R. M. Dynamic biological networks : the stomatogastric nervous system. In MIT Press eBooks (1992). <https://ci.nii.ac.jp/ncid/BA19537454>
- Hart, A. C., Sims, S., & Kaplan, J. M. (1995). Synaptic code for sensory modalities revealed by *C. elegans* GLR-1 glutamate receptor. *Nature*, 378(6552), 82–85. <https://doi.org/10.1038/378082a0>
- Hendricks, M., Ha, H., Maffey, N., & Zhang, Y. Compartmentalized calcium dynamics in a *C. elegans* interneuron encode head movement. *Nature* 487(7405), 99–103 (2012). <https://doi.org/10.1038/nature11081>

- Itskovits, E., Ruach, R., Kazakov, A., & Zaslaver, A. Concerted pulsatile and graded neural dynamics enables efficient chemotaxis in *C. elegans*. *Nature Communications* 9(1). (2018). <https://doi.org/10.1038/s41467-018-05151-2>
- Jolliffe, I.T. Principal Component Analysis. (2002). In Springer eBooks. <https://doi.org/10.1007/b98835>
- Kaplan, J. M., & Horvitz, H. R. (1993). A dual mechanosensory and chemosensory neuron in *Caenorhabditis elegans*. In *Proc. Natl. Acad. Sci. USA* (Vol. 90, pp. 2227–2231). 10.1073/pnas.90.6.2227
- Kaplan, H. S., Thula, O. S., Khoss, N., & Zimmer, M. Nested neuronal dynamics orchestrate a behavioral hierarchy across timescales. *Neuron* 105(3), 562-576.e9 (2020). <https://doi.org/10.1016/j.neuron.2019.10.037>
- Kato, S., et al. Global brain dynamics embed the motor command sequence of *Caenorhabditis elegans*. *Cell* 163(3), 656-669 (2015). <https://doi.org/10.1016/j.cell.2015.09.034>
- Kim, D. H., Kim, J., Marques, J. C., Grama, A., Hildebrand, D. G. C., Gu, W., Li, J. M., & Robson, D. N. (2017). Pan-neuronal calcium imaging with cellular resolution in freely swimming zebrafish. *Nature Methods*, 14(11), 1107–1114. <https://doi.org/10.1038/nmeth.4429>
- Klioutchnikov, A., Wallace, D. J., Frosz, M. H., Zeltner, R., Sawinski, J., Pawlak, V., Voit, K., St J Russell, P., & Kerr, J. N. D. (2020). Three-photon head-mounted microscope for imaging deep cortical layers in freely moving rats. *Nature Methods*, 17(5), 509–513. <https://doi.org/10.1038/s41592-020-0817-9>

- Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1–2), 83–97. <https://doi.org/10.1002/nav.3800020109>
- Lemon, W. C., Pulver, S. R., Höckendorf, B., McDole, K., Branson, K., Freeman, J., & Keller, P. J. (2015). Whole-central nervous system functional imaging in larval *Drosophila*. *Nature Communications*, 6(1). <https://doi.org/10.1038/ncomms8924>
- Lhoest, Q., Villanova, D. M. A., Jernite, Y., Thakur, A., Patrick, V. P., Patil, S., Chaumond, J., Drame, M., Plu, J., Tunstall, L., Davison, J., Šaško, M., Chhablani, G., Malik, B., Brandeis, S., Scao, T. L., Sanh, V., Xu, C., Patry, N., . . . Wolf, T. (2021, September 7). Datasets: a community library for natural language processing. *arXiv.org*. Retrieved March 18, 2025, from <https://arxiv.org/abs/2109.02846>
- Lisman, J. E., & Jensen, O. The theta-gamma neural code. *Neuron*, 77(6), 1002–1016 (2013). <https://doi.org/10.1016/j.neuron.2013.03.007>
- Maguire, S. M., Clark, C. M., Nunnari, J., Pirri, J. K., & Alkema, M. J. The *C. elegans* touch response facilitates escape from predacious fungi. *Current Biology*, 21(15), 1326–1330 (2011). <https://doi.org/10.1016/j.cub.2011.06.063>
- Mango, S. E. (2007). The *C. elegans* pharynx: a model for organogenesis. *WormBook*. <https://doi.org/10.1895/wormbook.1.129.1>
- Manley, J., Lu, S., Barber, K., Demas, J., Kim, H., Meyer, D., Traub, F. M., & Vaziri, A. (2024). Simultaneous, cortex-wide dynamics of up to 1 million neurons reveal unbounded scaling of dimensionality with neuron number. *Neuron*, 112(10), 1694–1709.e5. <https://doi.org/10.1016/j.neuron.2024.02.011>

- Mann, K., Gallen, C. L., & Clandinin, T. R. (2017). Whole-Brain Calcium Imaging Reveals an Intrinsic Functional Network in *Drosophila*. *Current Biology*, 27(15), 2389-2396.e4.
<https://doi.org/10.1016/j.cub.2017.06.076>
- Miguel, D. et al. Preprint at bioRxiv (2021).
- MinIO, Inc. (2020). MinIO: High performance, kubernetes native object storage. In GitHub repository. GitHub. <https://github.com/minio/minio>
- MongoDB Inc. (2020). The MongoDB database. In GitHub repository. GitHub. <https://github.com/mongodb/mongo>
- Myronenko, A., & Song, N. X. (2010). Point Set Registration: Coherent Point Drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12), 2262–2275.
<https://doi.org/10.1109/tpami.2010.46>
- Nejatbakhsh, A., Varol, E., Yemini, E., Hobert, O., & Paninski, L. (2020). Probabilistic Joint Segmentation and Labeling of *C. elegans* Neurons. In *Lecture notes in computer science* (pp. 130–140). https://doi.org/10.1007/978-3-030-59722-1_13
- Neuromatch – Facilitating global collaboration in computational sciences. (n.d.). Retrieved March 18, 2025, from <http://neuromatch.io/>
- NeuroPAL — Yemini Lab. (n.d.). Yemini Lab. Retrieved March 18, 2025, from <https://www.yeminilab.com/neuropal>
- Nguyen, J. P., Shipley, F. B., Linder, A. N., Plummer, G. S., Liu, M., Setru, S. U., Shaevitz, J. W., & Leifer, A. M. (2015). Whole-brain calcium imaging with cellular resolution in

freely behaving *Caenorhabditis elegans*. *Proceedings of the National Academy of Sciences*, 113(8). <https://doi.org/10.1073/pnas.1507110112>

Node.js Core Collaborators. (2020). Node.js JavaScript runtime. In GitHub repository. GitHub. <https://github.com/nodejs/node>

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E.A. Quintero, Charles R Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. (2020) SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17(3), 261-272.

Pavani, S. R. P., & Piestun, R. (2008). Three dimensional tracking of fluorescent microparticles using a photon-limited double-helix response system. *Optics Express*, 16(26), 22048. <https://doi.org/10.1364/oe.16.022048>

Pedregosa, F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., Vanderplas J., Passos A., Cournapeau D., Brucher M., Perrot M., & Duchesnay É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. <https://doi.org/10.5555/1953048.2078195>

- Pierce-Shimomura, J. T., Morse, T. M., & Lockery, S. R. The fundamental role of pirouettes in *Caenorhabditis elegans* chemotaxis. *Journal of Neuroscience* (Vol. 19, Issue 21, pp. 9557–9569) (1999). <https://www.jneurosci.org/content/jneuro/19/21/9557.full.pdf>
- Pokala, N., Liu, Q., Gordus, A., & Bargmann, C. I. Inducible and titratable silencing of *Caenorhabditis elegans* neurons in vivo with histamine-gated chloride channels. *Proceedings of the National Academy of Sciences*, 111(7), 2770–2775 (2014). <https://doi.org/10.1073/pnas.1400615111>
- Poldrack, R. A., Gorgolewski, K. J., & Varoquaux, G. (2019). Computational and informatic advances for reproducible data analysis in neuroimaging. *Annual Review of Biomedical Data Science*, 2(1), 119–138. <https://doi.org/10.1146/annurev-biodatasci-072018-021237>
- Portugues, R., Feierstein, C. E., Engert, F., & Orger, M. B. (2014). Whole-Brain Activity Maps Reveal Stereotyped, Distributed Networks for Visuomotor Behavior. *Neuron*, 81(6), 1328–1343. <https://doi.org/10.1016/j.neuron.2014.01.019>
- Prevedel, R., Yoon, Y., Hoffmann, M., Pak, N., Wetzstein, G., Kato, S., Schrödel, T., Raskar, R., Zimmer, M., Boyden, E. S., & Vaziri, A. (2014). Simultaneous whole-animal 3D imaging of neuronal activity using light-field microscopy. *Nature Methods*, 11(7), 727–730. <https://doi.org/10.1038/nmeth.2964>
- Price-Whelan, A. M., Lim, P. L., Earl, N., Starkman, N., Bradley, L., Shupe, D. L., Patil, A. A., Corrales, L., Brasseur, C. E., Nöthe, M., Donath, A., Tollerud, E., Morris, B. M., Ginsburg, A., Vaher, E., Weaver, B. A., Tocknell, J., Jamieson, W., . . . Zonca, A. (2022). The Astropy Project: Sustaining and Growing a Community-oriented Open-

- source Project and the Latest Major Release (v5.0) of the Core Package*. The Astrophysical Journal, 935(2), 167. <https://doi.org/10.3847/1538-4357/ac7c74>
- Randi, F., Sharma, A. K., Dvali, S., & Leifer, A. M. (2023). Neural signal propagation atlas of *Caenorhabditis elegans*. Nature, 623(7986), 406–414. <https://doi.org/10.1038/s41586-023-06683-4>
- Riddle, D. L., Blumenthal, T., Meyer, B. J., & Priess, J. R. (1998). *C. Elegans* II. <https://pubmed.ncbi.nlm.nih.gov/21413221/>
- Royer, L. A., Lemon, W. C., Chhetri, R. K., Wan, Y., Coleman, M., Myers, E. W., & Keller, P. J. (2016). Adaptive light-sheet microscopy for long-term, high-resolution imaging in living organisms. Nature Biotechnology, 34(12), 1267–1278. <https://doi.org/10.1038/nbt.3708>
- Rübel, O., Tritt, A., Ly, R., Dichter, B. K., Ghosh, S., Niu, L., Baker, P., Soltesz, I., Ng, L., Svoboda, K., Frank, L., & Bouchard, K. E. (2022). The Neurodata Without Borders ecosystem for neurophysiological data science. eLife, 11. <https://doi.org/10.7554/elife.78362>
- Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B., Tinevez, J. Y., White, D. J., Hartenstein, V., Eliceiri, K., Tomancak, P., & Cardona, A. (2012). Fiji: an open-source platform for biological-image analysis. Nature Methods, 9(7), 676–682. <https://doi.org/10.1038/nmeth.2019>

- Schrödel, T., Prevedel, R., Aumayr, K., Zimmer, M., & Vaziri, A. (2013). Brain-wide 3D imaging of neuronal activity in *Caenorhabditis elegans* with sculpted light. *Nature Methods*, 10(10), 1013–1020. <https://doi.org/10.1038/nmeth.2637>
- Skinner, F. K., Kopell, N., & Marder, E. (1994). Mechanisms for oscillation and frequency control in reciprocally inhibitory model neural networks. *Journal of Computational Neuroscience*, 1(1–2), 69–87. <https://doi.org/10.1007/bf00962719>
- Skocek, O., Nöbauer, T., Weilguny, L., Traub, F. M., Xia, C. N., Molodtsov, M. I., Grama, A., Yamagata, M., Aharoni, D., Cox, D. D., Golshani, P., & Vaziri, A. (2018). High-speed volumetric imaging of neuronal activity in freely moving rodents. *Nature Methods*, 15(6), 429–432. <https://doi.org/10.1038/s41592-018-0008-0>
- Skuhersky, M., Wu, T., Yemini, E., Nejatbakhsh, A., Boyden, E., & Tegmark, M. (2022). Toward a more accurate 3D atlas of *C. elegans* neurons. *BMC Bioinformatics*, 23(1). <https://doi.org/10.1186/s12859-022-04738-3>
- Sprague, D. Y., Rusch, K., Dunn, R. L., Borchardt, J. M., Ban, S., Bubnis, G., Chiu, G. C., Wen, C., Suzuki, R., Chaudhary, S., Lee, H. J., Yu, Z., Dichter, B., Ly, R., Onami, S., Lu, H., Kimura, K. D., Yemini, E., & Kato, S. (2025). Unifying community whole-brain imaging datasets enables robust neuron identification and reveals determinants of neuron position in *C. elegans*. *Cell Reports Methods*, 100964. <https://doi.org/10.1016/j.crmeth.2024.100964>

- Stirman, J. N., Smith, I. T., Kudenov, M. W., & Smith, S. L. (2016). Wide field-of-view, multi-region, two-photon imaging of neuronal activity in the mammalian brain. *Nature Biotechnology*, 34(8), 857–862. <https://doi.org/10.1038/nbt.3594>
- Stojanovski, K., Großhans, H., & Towbin, B. D. (2022). Coupling of growth rate and developmental tempo reduces body size heterogeneity in *C. elegans*. *Nature Communications*, 13(1). <https://doi.org/10.1038/s41467-022-29720-8>
- Sulston, J., Schierenberg, E., White, J., & Thomson, J. (1983). The embryonic cell lineage of the nematode *Caenorhabditis elegans*. *Developmental Biology*, 100(1), 64–119. [https://doi.org/10.1016/0012-1606\(83\)90201-4](https://doi.org/10.1016/0012-1606(83)90201-4)
- Suzuki, Ryoga; Wen, Chentao; Sprague, Daniel; Onami, Shuichi; Kimura, Koutarou D (2024) Whole-brain spontaneous GCaMP activity with NeuroPAL cell ID information of semi-restricted worms (Version 0.240402.2118) [Data set]. DANDI archive. <https://doi.org/10.48324/dandi.000692/0.240402.2118>
- Tekieli, T., Yemini, E., Nejatbakhsh, A., Wang, C., Varol, E., Fernandez, R. W., Masoudi, N., Paninski, L., & Hobert, O. (2021). Visualizing the organization and differentiation of the male-specific nervous system of *C. elegans*. *Development*. <https://doi.org/10.1242/dev.199687>
- Toyoshima, Y., Wu, S., Kanamori, M., Sato, H., Jang, M. S., Oe, S., Murakami, Y., Teramoto, T., Park, C., Iwasaki, Y., Ishihara, T., Yoshida, R., & Iino, Y. (2020). Neuron ID dataset facilitates neuronal annotation for whole-brain activity imaging of *C. elegans*. *BMC Biology*, 18(1). <https://doi.org/10.1186/s12915-020-0745-2>

Traefik Labs. (2020). Traefik: The cloud native application proxy. In GitHub repository. GitHub.

<https://github.com/traefik/traefik>

Uzel, K., Kato, S., & Zimmer, M. (2022). A set of hub neurons and non-local connectivity features support global brain dynamics in *C. elegans*. *Current Biology*, 32(16), 3443-3459.e8. <https://doi.org/10.1016/j.cub.2022.06.039>

Varol, E., Nejatbakhsh, A., Sun, R., Mena, G., Yemini, E., Hobert, O., & Paninski, L. (2020). Statistical Atlas of *C. elegans* Neurons. In *Lecture notes in computer science* (pp. 119–129). https://doi.org/10.1007/978-3-030-59722-1_12

Venkatachalam, V., Ji, N., Wang, X., Clark, C., Mitchell, J. K., Klein, M., Tabone, C. J., Florman, J., Ji, H., Greenwood, J., Chisholm, A. D., Srinivasan, J., Alkema, M., Zhen, M., & Samuel, A. D. T. (2015). Pan-neuronal imaging in roaming *Caenorhabditis elegans*. *Proceedings of the National Academy of Sciences*, 113(8). <https://doi.org/10.1073/pnas.1507109113>

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., Van Der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., . . . Vázquez-Baeza, Y. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3), 261–272. <https://doi.org/10.1038/s41592-019-0686-2>

Vladimirov, N., Wang, C., Höckendorf, B., Pujala, A., Tanimoto, M., Mu, Y., Yang, C., Wittenbach, J. D., Freeman, J., Preibisch, S., Koyama, M., Keller, P. J., & Ahrens, M.

- B. (2018). Brain-wide circuit interrogation at the cellular level guided by online analysis of neuronal function. *Nature Methods*, 15(12), 1117–1125.
<https://doi.org/10.1038/s41592-018-0221-x>
- Wen, C., Miura, T., Voleti, V., Yamaguchi, K., Tsutsumi, M., Yamamoto, K., Otomo, K., Fujie, Y., Teramoto, T., Ishihara, T., Aoki, K., Nemoto, T., Hillman, E. M., & Kimura, K. D. (2021). 3DeeCellTracker, a deep learning-based pipeline for segmenting and tracking cells in 3D time lapse images. *eLife*, 10. <https://doi.org/10.7554/elife.59187>
- White, J. G., Southgate, E., Thomson, J. N., & Brenner, S. (1986). The structure of the nervous system of the nematode *Caenorhabditis elegans*. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 314(1165), 1–340.
<https://doi.org/10.1098/rstb.1986.0056>
- Witvliet, D., Mulcahy, B., Mitchell, J. K., Meirovitch, Y., Berger, D. R., Wu, Y., Liu, Y., Koh, W. X., Parvathala, R., Holmyard, D., Schalek, R. L., Shavit, N., Chisholm, A. D., Lichtman, J. W., Samuel, A. D. T., & Zhen, M. (2021). Connectomes across development reveal principles of brain maturation. *Nature*, 596(7871), 257–261.
<https://doi.org/10.1038/s41586-021-03778-8>
- Wu, Y., Wu, S., Wang, X., Lang, C., Zhang, Q., Wen, Q., & Xu, T. (2022). Rapid detection and recognition of whole brain activity in a freely behaving *Caenorhabditis elegans*. *PLoS Computational Biology*, 18(10), e1010594. <https://doi.org/10.1371/journal.pcbi.1010594>
- Yemini, E., et al. NeuroPAL: A multicolor atlas for whole-brain neuronal identification in *C. elegans*. *Cell* 184(1), 272–288 (2021). <https://doi.org/10.1016/j.cell.2020.12.012>

- Yemini, Eviatar; Venkatachalam, Vivek; Lin, Albert; Varol, Erdem; Sprague, Daniel; Paninski, Liam; Samuel, Arivinthan; Hobert, Oliver (2024) NeuroPAL: Atlas of *C. elegans* neuron locations and colors in NeuroPAL worm (Version 0.241009.1514) [Data set]. DANDI archive. <https://doi.org/10.48324/dandi.000715/0.241009.1514>
- Yemini, Ev; Nejatbakhsh, Amin; Venkatachalam, Vivek; Lin, Albert; Sprague, Daniel; Samuel, Aravinthan; Hobert, Oliver (2024) NeuroPAL Microfluidic Chip Images and GCaMP activity (Version 0.241009.1457) [Data set]. DANDI archive. <https://doi.org/10.48324/dandi.000541/0.241009.1457>
- Yeon, J., Kim, J., Kim, D. Y., Kim, H., Kim, J., Du, E. J., Kang, K., Lim, H. H., Moon, D., & Kim, K. A sensory-motor neuron type mediates proprioceptive coordination of steering in *C. elegans* via two TRPC channels. *PLoS Biology*. 16(6), e2004929 (2018). <https://doi.org/10.1371/journal.pbio.2004929>
- Yu, X., Creamer, M. S., Randi, F., Sharma, A. K., Linderman, S. W., & Leifer, A. M. (2021). Fast deep neural correspondence for tracking and identifying neurons in *C. elegans* using semi-synthetic training. *eLife*, 10. <https://doi.org/10.7554/elife.66410>
- Zimmer, M., Gray, J. M., Pokala, N., Chang, A. J., Karow, D. S., Marletta, M. A., Hudson, M. L., Morton, D. B., Chronis, N., & Bargmann, C. I. (2009). Neurons Detect Increases and Decreases in Oxygen Levels Using Distinct Guanylate Cyclases. *Neuron*, 61(6), 865–879. <https://doi.org/10.1016/j.neuron.2009.02.013>

Zong, W., Obenhaus, H. A., Skytøen, E. R., Eneqvist, H., De Jong, N. L., Vale, R., Jorge, M.

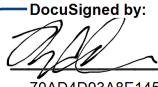
R., Moser, M., & Moser, E. I. (2022). Large-scale two-photon calcium imaging in freely

moving mice. *Cell*, 185(7), 1240-1256.e30. <https://doi.org/10.1016/j.cell.2022.02.017>

Publishing Agreement

It is the policy of the University to encourage open access and broad distribution of all theses, dissertations, and manuscripts. The Graduate Division will facilitate the distribution of UCSF theses, dissertations, and manuscripts to the UCSF Library for open access and distribution. UCSF will make such theses, dissertations, and manuscripts accessible to the public and will take reasonable steps to preserve these works in perpetuity.

I hereby grant the non-exclusive, perpetual right to The Regents of the University of California to reproduce, publicly display, distribute, preserve, and publish copies of my thesis, dissertation, or manuscript in any form or media, now existing or later derived, including access online for teaching, research, and public service purposes.

DocuSigned by:

70AD4D93A8E1453...

Author Signature

3/19/2025

Date