Scalable Tools for Information Extraction and Causal Modeling of Neural Data

Amin Nejatbakhsh

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
under the Executive Committee
of the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2022

Abstract

Scalable Tools for Information Extraction and Causal Modeling of Neural Data

Amin Nejatbakhsh

Systems neuroscience has entered in the past 20 years into an era that one might call "large scale systems neuroscience". From tuning curves and single neuron recordings there has been a conceptual shift towards a more holistic understanding of how the neural circuits work and as a result how their representations produce neural tunings [Kriegeskorte and Wei, 2021].

With the introduction of a plethora of datasets in various scales, modalities, animals, and systems; we as a community have witnessed invaluable insights that can be gained from the collective view of a neural circuit which was not possible with small scale experimentation [Urai et al., 2022]. The concurrency of the advances in neural recordings such as the production of wide field imaging technologies and neuropixels with the developments in statistical machine learning and specifically deep learning has brought system neuroscience one step closer to data science. With this abundance of data, the need for developing computational models has become crucial. We need to make sense of the data, and thus we need to build models that are constrained up to the acceptable amount of biological detail and probe those models in search of neural mechanisms.

This thesis consists of sections covering a wide range of ideas from computer vision, statistics, machine learning, and dynamical systems. But all of these ideas share a

common purpose, which is to help automate neuroscientific experimentation process in different levels. In chapters 1, 2, and 3, I develop tools that automate the process of extracting useful information from raw neuroscience data in the model organism *C. elegans*. The goal of this is to avoid manual labor and pave the way for high throughput data collection aiming at better quantification of variability across the population of worms. Due to its high level of structural and functional stereotypy, and its relative simplicity, the nematode *C. elegans* has been an attractive model organism for systems and developmental research. With 383 neurons in males and 302 neurons in hermaphrodites, the positions and function of neurons is remarkably conserved across individuals. Furthermore, *C. elegans* remains the only organism for which a complete cellular, lineage, and anatomical map of the entire nervous system has been described for both sexes. Here, I describe the analysis pipeline that we developed for the recently proposed NeuroPAL technique in *C. elegans*. Our proposed pipeline consists of atlas building (chapter 1), registration, segmentation, neural tracking (chapter 2), and signal extraction (chapter 3). I emphasize that categorizing the analysis techniques as a pipeline consisting of the above steps is general and can be applied to virtually every single animal model and emerging imaging modality. I use the language of probabilistic generative modeling and graphical models to communicate the ideas in a rigorous form, therefore some familiarity with those concepts could help the reader navigate through the chapters of this thesis more easily.

In chapters 4 and 5 I build models that aim to automate hypothesis testing and causal interrogation of neural circuits. The notion of functional connectivity (FC) has been instrumental in our understanding of how information propagates in a neural circuit. However, an important limitation is that current techniques do not dissociate between causal connections and purely functional connections with no mechanistic correspondence. I start chapter 4 by introducing causal inference as a unifying language for the following chapters. In chapter 4 I define the notion of interventional connectivity (IC) as a way to summarize the effect of stimulation in a neural circuit providing a more mechanistic description of the information flow. I then investigate which functional connectivity metrics are best predictive of IC in simulations and real data. Following this framework, I discuss how stimulations and interventions can be used to improve fitting and generalization properties of time series models. Building on the literature of model identification and active causal discovery I develop a switching time series model and a method for finding stimulation patterns that help the model to generalize to the vicinity of the observed neural trajectories. Finally in chapter 5 I develop a new FC metric that separates the transferred information from one variable to the other into unique and synergistic sources.

In all projects, I have abstracted out concepts that are specific to the datasets at hand and developed the methods in the most general form. This makes the presented methods applicable to a broad range of datasets, potentially leading to new findings. In addition, all projects are accompanied with extensible and documented code packages, allowing theorists to repurpose the modules for novel applications and experimentalists to run analysis on their datasets efficiently and scalably.

In summary my **main contribution** in this thesis are the following:

- Building the first atlases of hermaphrodite and male *C. elegans* and developing a generic statistical framework for constructing atlases for a broad range of datasets.
- Developing a semi-automated analysis pipeline for neural registration, segmentation, and tracking in *C. elegans*.
- Extending the framework of non-negative matrix factorization to datasets with de-

formable motion and developing algorithms for joint tracking and signal demixing from videos of semi-immobilized *C. elegans*.

- Defining the notion of interventional connectivity (IC) as a way to summarize the effect of stimulation in a neural circuit and investigating which functional connectivity metrics are best predictive of IC in simulations and real data.
- Developing a switching time series model and a method for finding stimulation patterns that help the model to generalize to the vicinity of the observed neural trajectories.
- Developing a new functional connectivity metric that separates the transferred information from one variable to the other into unique and synergistic sources.
- Implementing extensible, well documented, open source code packages for each of the above contributions.

Table of Contents

List of Figures

Acknowledgements

Doing science can be lonely at times. Progress cannot be made without countless hours of running experiments or codes that occasionally provide inconclusive or confusing results. If one finds their so-called flow state when focused on their problem, this seemingly painful process turns into an immersive, wonderful, and rewarding experience. In the flow state, the time works very strangely. For me, the past 5 years felt like a blink of an eye, but thinking back I have gained a lifetime's worth of memories and experiences. Being an ambitious and energetic youngster, I traveled a long way to the United States five years ago and began the story of this thesis. My life in the past five years was fulfilled with enriching experiences turning me into who I am today. Throughout my Ph.D., I have had the privilege of learning from many incredible individuals each of which shaped part of who I am today.

First, I thank my adviser Liam Paninski for his unbounded support and patience. He managed to turn my (somewhat unrealistic) intellectual ambition into concrete steps of doing proper and rigorous science. Apart from his great insights in statistics and neuroscience, at a higher level, I learned from him how to think biologically about statistics and statistically about biology. Next, I thank Larry Abbott, David Blei, and John Cunningham for serving as my thesis committee members. They have been instrumental to my academic growth and development. Larry has been my role model for thinking computationally about the brain. Much of what I know about statistics and machine learning comes from attending class discussions and reading groups with David and John. Their level of statistical insight is exemplary, always leaving room for more curiosity. I also thank Konrad Kording for serving as the external member of my defense committee and providing insightful feedback about my work. I thank Erdem Varol for being a great friend, colleague, and mentor to me. We have co-authored papers, solved late-night math problems together, finished running races and triathlons, and jumped in the Hudson River.

The results in this thesis would not have been possible without the collaboration of top-level experimentalists in the field. Eviatar Yemini, the Hobert Lab, and the Venkatachalam Lab provided datasets that are central to the results of the first three chapters of this thesis. The experimental work by the Kiani Lab motivated the ideas presented in chapter 4.2. In addition to the exceptional level of experimental precision that all our collaborators have, their insightful suggestions have been a critical component of the presented results. In addition to the above, I have been enjoying the mentorship and collaborations of exceptional theoreticians. Specifically, I am thankful for the following scientists. Luca Mazzucato and Francesco Fumarola helped with developing the theoretical ideas in chapter 4.2. Chapter 4.3 is based on the ideas that were initially discussed with Laureline Logiaco and later on continued with Manuel Beiran. Results in chapter 5 are applications of the ideas developed by Ari Pakman.

I am thankful to the Center for Theoretical Neuroscience (CTN) at Columbia University for its excellent research environment and brilliant scientists with diverse backgrounds. Specifically, I am thankful for the enlightening discussions that I have had with Sean Escola, Scott Linderman, Ken Miller, Kanaka Rajan, Josh Glaser, Scott Linderman, Shreya Saxena, Rudy Behnia, Geoff Pleiss, Yashar Ahmadian, Shuonan Chen, Dan Biderman, Jung Park, Ramin Khajeh, Dar Gilboa, Stevie Hamilton, Denis Turcu, Rahim Hashim, Francisco Sacadura, Amol Pasarkar, Minni Sun, Jack Lindsay, Jasmine Stone, Ben Antin, Elom Amematsro, Ishani Ganguli, Jacob Portes, Zhenrui Liao, Sean Bittner, Rainer Engelken, Kenny Kay, Ramon Nogueira, Sam Muscinelli, Matt Whiteway, Ji Xia, Tahereh Toosi, and Salomon Muller. In addition, I had the opportunity to participate in reading groups and discussions in the statistics department with Yixin Wang, Luhuan Wu, Julien Boussard, Keyon Vafa, Claudia Shi, and Inigo Urteaga. Thanks to the CTN admins and Columbia Neurobiology and Behavior (NBB) admins and co-directors for preparing and maintaining this great environment. Specifically, I thank Allison Ong, Mahham Fayyaz, Aleyna Silcott, Alla Kerzhner, Rozanna Yakub, Christina Doyle, Ken Miller, Wes Grueber, Darcy Kelley, and Carol Mason.

I thank my cohort for their never-ending support. I am grateful for having Dan Tyulmankov, Gordon Petty, Patricia Cooney, Mike Cohanpour, You-Nah Jeon, Claire Everett, Andres Villegas, Marjorie Xie, Taiga Abe, Michelle Stackman, and Liz Pekarskaya as my Ph.D. cohort. I thank Zuckerman Institute Athletics Club, and the soccer and basketball teams for helping me stay physically active and mentally healthy. I also thank Columbia Iranian Students Association 2018, GISU Dance Group, and CU Ballroom for helping me express my artistic self through cultural events and dance practices. In addition to the ones I met through Columbia, I have been lucky to have a fairly large support group. I thank Marzieh Jadidi, Sahand Seif, Alireza Bordbar, Kourosh Ghobadi, Mina Dalirrooyfard, Tina Torkaman, Mahdi Torkashvand, Maedeh Seyedolmohadesin, Nadia Ghobadi, Negin Alizadeh, Nooshin Azami, Amirhosein Parvardi, Niloufar Saharkhiz, Zohreh Jalilvand, Mina Latifi, Milad Bakhshizdeh, Saeid Karimimehr, Menoua Keshishian, Armita Arian, Kasra Jalaldoust, Aria Halavati, Erfan Tavakkoli, Mohammad Razeghi, Soheil Arabzadeh, Mohammad Chavoshian, Fatemeh Haghpanah, Farshid Soltanveis, Hamed Dashti, Ramin Khajeh, Navid Mousavi, Yassi Ayman, Parham Jafari, and Aryan Rahimieh.

Finally, I thank those who gave me unconditional love and support. My grandma, my sisters Fati, Naomi, and Nazi, and my mom. I especially thank Fati, she never stopped believing in me through ups and downs.

Dedication

To my grandma Aba, who was my biggest fan when she was among us.

CHAPTER 1

Statistical Atlas: How to Build an Average Brain

1.1 Introduction

Constructing atlases of biological structures such as the brain helps summarize normative patterns in a population and quantifies variability across individuals. An atlas also provides a common coordinate framework to serve as a target for image registration and normalization, which can help decouple and quantify different sources of variability observed in the data [Greitz et al., 1991; Jones et al., 2009; Roland et al., 1994; Scheffer and Meinertzhagen, 2019]. The sources of variability observed in images could be both due to biological factors, such as ganglia placement, posture, and morphology, as well as non-biological factors such as photobleaching of fluorophores, illumination artefacts, and camera placement. With the introduction of new imaging technologies [Ahrens et al., 2013b; Cong et al., 2017; Prevedel et al., 2014] that capture complex biological signals such as those found in neural circuits or the musculoskeletal system, atlas building is a valuable step before downstream analyses.

The *C. elegans* nervous system has been an attractive target for atlas building in recent years due to its high level of structural and functional stereotypy, and its relative simplicity [Choe and Strange, 2007,?; Kaiser and Hilgetag, 2006; Szigeti et al., 2014; Varol et al., 2020; Yemini et al., 2021], the number, positions, and function of neurons is remarkably conserved across individuals. The hermaphrodite has 302 neurons in its entire nervous system, which can be simultaneously imaged using fluorescence microscopy [Venkatachalam et al., 2016c; White et al., 1986a]. Several atlases of neural positions in the *C. elegans* hermaphrodite have been introduced, utilizing a variety of shape and pose models [Bubnis et al., 2019; Long et al., 2009; Skuhersky et al., 2021; Toyoshima et al., 2019; Varol et al., 2020].

In contrast, construction of the male *C. elegans* nervous system has been more challenging for several factors: 1) Males have roughly 30 percent more neurons than hermaphrodites, and additional ganglia enclosing these neurons, primarily in their tail [Sulston and Horvitz, 1977], 2) males show greater variability in their neuron positions and perhaps even their gangliar positions [Tekieli et al., 2021], 3) the male body size is smaller than that of hermaphrodites, resulting in a higher neuron density [Emmons and Sternberg, 2011]. Therefore, models that normalize hermaphrodite neuron positions do not necessarily generalize to males.

Another simple species that is known to exhibit significant stereotypy is the fruit fly, *Drosophila melanogaster*. In the fly, one suitable structure for atlas building is the wing. Although *Drosophila* wings can be evaluated qualitatively or by metrics such as length and surface area, they are often measured within a geometric morphometric framework [Houle et al., 2010]. Landmarks are based on vein intersections with semi-landmarks defining curves. Biometric facial recognition tools have succeeded in classifying images of *Drosophila* wings into biological categories [Dworkin and Gibson, 2006]. However, a probabilistic atlas that quantifies the sources of structural variability amongst wings of different phenotypes and sexes is not yet established.

While the above cases exemplify the scenarios in which atlases can help quantify biological variability, atlas building so far has been hand-tailored to accommodate the specifications of a single organism or a single experimental condition, providing an obstacle to experimentalists that require atlases for novel biological datasets that they curate [Heckscher et al., 2014].

This chapter provides a probabilistic framework for building atlases for various model organisms. The chapter is organized as follows. In the first section, we describe the development of the hermaphrodite *C. elegans* statistical atlas of neural positions [Varol et al., 2020]. Next, in the second section using similar techniques we develop the male *C. elegans* statistical atlas and draw comparisons with the hermaphrodite atlas [Tekieli et al., 2021]. In the last section we extend the methodology to a broader range of assumptions and describe how to build supervised, semi-supervised, and unsupervised atlases of biological structures directly from their images. We then construct the fruit fly wing atlas using images to showcase an application (unpublished).

1.2 Hermaphrodite C. elegans Neural Atlas

Imaging-based atlases of human and animal brains have enabled the principled and standardized means of hypothesis testing in a wide variety of domains [Bubnis et al., 2019; Dickie et al., 2017; Jones et al., 2009; Lein et al., 2007; Mazziotta et al., 2001; Oh et al., 2014; Toyoshima et al., 2020]. Common procedures that atlases enable are the registration of population samples to a common space [Zitova and Flusser, 2003], discriminating pattern differences across samples [Ashburner and Friston, 2000], segmentation into regions of interest [Cabezas et al., 2011], and regularizing complex Bayesian models [Saxena et al., 2019]. Importantly, atlases enable the formation of large-scale population studies due to their ability to gather high-dimensional data into a commensurate space.

C. elegans is a widely studied model organism with a simple nervous system that consists of 302 neurons in the adult hermaphrodite [White et al., 1986b]. Its simplicity and stereotypy have enabled highly-reproducible experimental settings which have been crucial in elucidating neuroscientific hypotheses. Furthermore, to date, *C. elegans* is the only animal whose connectome is completely mapped [Cook et al., 2019b; Jarrell et al., 2012b; White et al., 1986b]. Despite this atlas of connectivity, attempts at quantifying the variability of the neuron positions therein has been limited, capturing only a partial subset of these neurons [Toyoshima et al., 2020]. This is due to the limited number of samples available from electron micrograph reconstructions and an inability to identify neural identities via position alone [Yemini et al., 2019b]. The recent introduction of NeuroPAL, a strain for complete neural identification in *C. elegans*, has enabled efficient and precise annotation of neuron positions in multiple worms.

Using a NeuroPAL dataset, encompassing all head and tail neurons from 10 worms, we propose a latent multivariate statistical model that captures the canonical positions and covariances of *C. elegans* neurons. The observed neurons were captured by fluorescent volumetric imaging. These were then modeled as a multivariate sample, drawn from a latent distribution subjected to a random affine transformation. Given this statistical model, we infer the canonical means and covariances of all neurons present in the head and tail of the worm, yielding a novel positional statistical atlas. To improve our statistical atlas with additional, incompletely annotated worms, we propose a semi-supervised approach for cell-identification. As shown in [Yemini et al., 2019b], using our trained atlas, we can automatically identify neurons in out-of-sample worms with more than 86% accuracy in the head and 94% accuracy in the tail. These accuracies represent the current state of the art, improving the accuracies reported in [Kainmueller et al., 2014] and [Toyoshima et al., 2020]. Furthermore, we demonstrate an additional application of our atlas to obtain a correlation analysis of neural positions, which sheds

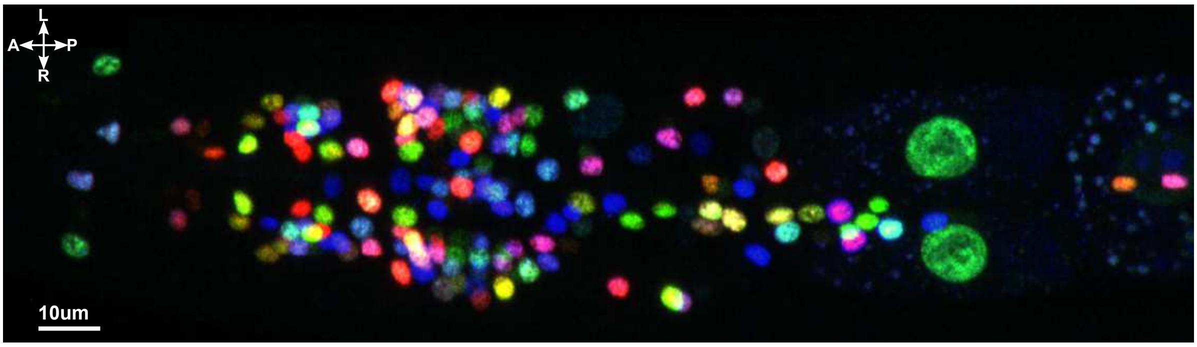

Figure 1.1: **Example NeuroPAL image** Deterministic coloring of *C. elegans* neurons, in a NeuroPAL strain, enables the complete neural identification across a population of worms. See [Yemini et al., 2019b] for details.

light on the structural organization of neurons and their potential connections to genetic lineages.

1.2.1 Data and Pre-processing

To construct the statistical atlas of *C. elegans* neurons, we used volumetric images of both heads and tails from 10 worms (strain OH15262). All worms were imaged on a Zeiss LSM 880 confocal with 32 detector channels and the following laser lines: 405nm, 488nm, 561nm, and 633nm. Volumetric resolution was approximately (X,Y,Z): 0.2 μ m × 0.2 μ m × 0.8 μ m. Images were acquired with four color channels, corresponding to the NeuroPAL fluorophores: mTagBFP2, CyOFP1, mNeptune2.5, and TagRFP-T [Yemini et al., 2019b]. See Figure 1.1 for a representative maximum intensity projection from a head sample. The volumetric images were subsequently annotated by an expert to denote the approximate center for each neuron and its corresponding identity. In total, 240 neurons were annotated in each worm, 195 from the head and 45 from tail. The remaining neurons from the midbody were not imaged for this study.

1.2.2 Method

Due to variability in illumination and the pose of the worm when imaged, observed neuron positions and their exact color balance may vary across imaged worms. This presents a significant challenge when attempting to obtain correspondence between worms to infer the identities of neurons. Therefore, to normalize the random variability that occurs across different worms, prior to identifying neurons in any given microscopy image, we estimate a statistical atlas of neuron positions and colors.

The approach we take resembles the joint expectation-maximization alignment of point sets technique of [Evangelidis and Horaud, 2018], with several important differences discussed below. The dataset we are modeling consists of a collection of point sets: each worm corresponds to one point set, with each point in the set corresponding to the position and color of a single detected neuron. We model each of these positions and colors as samples from a statistical atlas that is common across worms. Each neuron i has a corresponding mean and covariance in this atlas, denoted as $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$, respectively. After drawing all the positions and colors for a given worm j we apply a random affine transformation (parametrized by a matrix $\boldsymbol{\beta}_j$ and translation vector $\boldsymbol{\beta}_j^0$). Finally, since the order of neurons in each point set is arbitrary, we scramble the identities of the neurons with a random permutation, parameterized by a permutation matrix $\boldsymbol{P}_j$. This generative model is summarized in Figure 4.8. See also [Bubnis et al., 2019] for a related model (without the alignment term, and with an inference approach that differs from the methods we describe below).

We build on the methods in [Evangelidis and Horaud, 2018] to infer the parameters of this generative model (i.e., the means and covariances of the statistical atlas, the random transformations, and the random permutations), in a completely unsupervised fashion, using a three-way expectation-maximization procedure. However, in our dataset, we have access to fully annotated neuron detections. We take advantage of this supervised data to simplify the inference problem.

Now we can describe our model in detail. Neuron positions are three-dimensional, and there are three color channels in this dataset (given our three neuron-specific fluorophore channels, we discard the panneuronal TagRFP-T channel as uninformative); therefore, if we use $\boldsymbol{w}_{i,j}$ to denote the appended position and color vector of the i-th neuron in worm j (as output by the detection step described in the previous section), then $\boldsymbol{w}_{i,j} \in \mathbb{R}^6$. Each of these observed $\boldsymbol{w}_{i,j}$ vectors has a corresponding latent vector $\boldsymbol{z}_{i,j}$ in the aligned atlas space. We model this latent vector as a Gaussian,

$$\boldsymbol{z}_{i,j} \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \tag{1.1}$$

with means $\boldsymbol{\mu}_i \in \mathbb{R}^6$ and covariances $\boldsymbol{\Sigma}_i \in \mathbb{S}_+^6$ that do not depend on the worm index j. We model the covariance $\boldsymbol{\Sigma}_i$ with block structure of the form $\boldsymbol{\Sigma}_i = \begin{bmatrix} \Sigma^i_{\text{position}} & \mathbf{0} \\ \mathbf{0} & \Sigma^i_{\text{color}} \end{bmatrix}$, since position and each color are independently varying.

Now the latent vectors $\boldsymbol{z}_{i,j}$ in the atlas space and observed data $\boldsymbol{w}_{i,j}$ extracted from

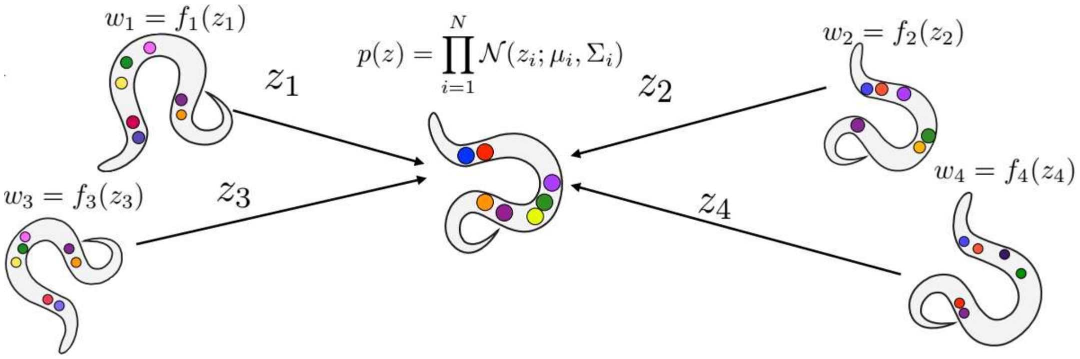

Figure 1.2: **Schematic of the generative model of neuron position and color expression** First we draw a position and color for each neuron i from a distribution with mean $\boldsymbol{\mu}_i$ and covariance $\boldsymbol{\Sigma}_i$; then, to create the observed data + $\boldsymbol{w}_{i,j}$ (the color and position of the i-th neuron of the j-th worm) we apply a random affine transformation and a random permutation encoded by $f(\cdot)$.

the imaged worm j are connected by a worm-specific random affine transformation and permutation. We denote the intermediate affine-transformed variables as $\boldsymbol{x}_{i,j}$:

$$\boldsymbol{x}_{i,j} = \boldsymbol{z}_{i,j}\boldsymbol{\beta}_j + \boldsymbol{\beta}_j^0, \tag{1.2}$$

with $\boldsymbol{\beta}_j$ a 6×6 matrix (with a similar block structure as $\boldsymbol{\Sigma}_i$) and $\boldsymbol{\beta}_j^0 \in \mathbb{R}^6$. then we obtain $\boldsymbol{w}_{i,j}$ by scrambling the labels via the permutation p_j (corresponding to a permutation matrix $\mathbf{P}_j$):

$$\boldsymbol{w}_{i,j} = \boldsymbol{x}_{p_j(i),j}. \tag{1.3}$$

The summary of the generative process is illustrated in Figure 4.8 by combining the permutation operation and the transformation together as a latent function $f(\cdot)$. Note that this model permits partial and variable observations of neurons across different animals if we allow the permutation matrix to be unbalanced (not square), indicating the existence of neurons that are not observed in individual animals.

Given these modeling assumptions, for a dataset of m worms and n_j detected neurons in each worm, we can express the likelihood as:

$$P(\boldsymbol{w}|\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{P},\boldsymbol{\beta},\boldsymbol{\beta}^0) = \prod_{j=1}^{m}\prod_{i=1}^{n_j} \frac{e^{-(1/2)(\boldsymbol{w}_{i,j}-\boldsymbol{\mu}_{\boldsymbol{p_{i,j}}}\boldsymbol{\beta}_j-\boldsymbol{\beta}_j^0)(\boldsymbol{\beta}_j\boldsymbol{\Sigma}_{p_{i,j}}\boldsymbol{\beta}_j^T)^{-1}(\boldsymbol{w}_{i,j}-\boldsymbol{\mu}_{\boldsymbol{p_{i,j}}}\boldsymbol{\beta}_j-\boldsymbol{\beta}_j^0)^T}}{(2\pi)^{d/2}\det((\boldsymbol{\beta}_j\boldsymbol{\Sigma}_{p_{i,j}}\boldsymbol{\beta}_j^T))^{1/2}} \tag{1.4}$$

Since the term $\sum_j\sum_i(1/2)\log\det((\boldsymbol{\beta}_j\boldsymbol{\Sigma}_{p_{i,j}}\boldsymbol{\beta}_j^T)$ is permutation invariant, we can write it as $\sum_j\sum_i(1/2)\log\det((\boldsymbol{\beta}_j\boldsymbol{\Sigma}_i\boldsymbol{\beta}_j^T)$ and thus the maximum likelihood estimate (MLE) for our generative model involves optimizing the negative log-likelihood:

$$\underset{\boldsymbol{P},\boldsymbol{\beta},\boldsymbol{\beta}^0,\boldsymbol{\mu},\boldsymbol{\Sigma}}{\text{minimize}} \sum_{j=1}^{m}\sum_{i=1}^{n_j}(\boldsymbol{w}_{i,j}-\boldsymbol{\mu}_{\boldsymbol{p_{i,j}}}\boldsymbol{\beta}_j-\boldsymbol{\beta}_j^0)(\boldsymbol{\beta}_j\boldsymbol{\Sigma}_{p_{i,j}}\boldsymbol{\beta}_j^T)^{-1}(\boldsymbol{w}_{i,j}-\boldsymbol{\mu}_{\boldsymbol{p_{i,j}}}\boldsymbol{\beta}_j-\boldsymbol{\beta}_j^0)^T)+\log\det((\boldsymbol{\beta}_j\boldsymbol{\Sigma}_i\boldsymbol{\beta}_j^T) \tag{1.5}$$

1.2.3 Optimization

To infer the parameters of the generative model, we take an iterative block-coordinate descent approach, similar to [Evangelidis and Horaud, 2018]: we fix $(\boldsymbol{P},\boldsymbol{\beta},\boldsymbol{\beta}_0)$ (with $\boldsymbol{P}$

abbreviating the collection of permutations $\boldsymbol{P}_j$ for all worms j, and similarly for $\boldsymbol{\beta}, \boldsymbol{\beta}_0$) and solve for $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then fix $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and solve for $(\boldsymbol{P}, \boldsymbol{\beta}, \boldsymbol{\beta}_0)$. Below are the update steps for each of these blocks.

1.2.3.1 Inference of the statistical atlas parameters $\boldsymbol{\mu}, \boldsymbol{\Sigma}$:

Let $\boldsymbol{P}_j \in \mathscr{P}^{n \times n}$ denote the permutation matrix, $\boldsymbol{W}_j = [\boldsymbol{w}_{1,j}^T \ldots \boldsymbol{w}_{n,j}^T]^T \in \mathbb{R}^{n \times d}$ denote the row stacked features of the neurons of the jth worm, and let $\boldsymbol{\mu} = [\boldsymbol{\mu}_1^T \ldots \boldsymbol{\mu}_n^T] \in \mathbb{R}^{n \times d}$ denote the row stacked neuron means. The generative model can be written in matrix form as: $\boldsymbol{W}_j = \boldsymbol{P}_j \boldsymbol{\mu} \boldsymbol{\beta}_j + \mathbf{1}\boldsymbol{\beta}_j^0 + \boldsymbol{E}$ where $\boldsymbol{E}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\beta}_j \boldsymbol{\Sigma}_{\boldsymbol{P}_{i,j}} \boldsymbol{\beta}_j^T)$ denotes the row stacked uncertainty terms.

Since $\boldsymbol{P}_j^T \boldsymbol{P}_j = \mathbf{I}$ because $\boldsymbol{P}$ is a permutation matrix and assuming that $\boldsymbol{\beta}_j$ is a non-degenerate transformation, its inverse exists and can be used to write the system as: $\boldsymbol{P}_j^T \boldsymbol{W}_j \boldsymbol{\beta}_j^{-1} - \mathbf{1}\boldsymbol{\beta}_j^0 \boldsymbol{\beta}_j^{-1} = \boldsymbol{\mu} + \boldsymbol{V}$ where $\boldsymbol{V}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_i)$ is a term to quantify uncertainty.

This equation can be used to infer $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ in closed form by computing the first and second moments of $\boldsymbol{V}$:

$$\boldsymbol{\mu}^* = \frac{1}{m} \sum_{j=1}^{m} \boldsymbol{P}_j^T \boldsymbol{W}_j \boldsymbol{\beta}_j^{-1} - \mathbf{1}\boldsymbol{\beta}_j^0 \boldsymbol{\beta}_j^{-1} \tag{1.6}$$

$$\boldsymbol{\Sigma}_i^* = \frac{1}{m} \sum_{j=1}^{m} (\boldsymbol{P}_{j,i}^T \boldsymbol{W}_j \boldsymbol{\beta}_j^{-1} - \boldsymbol{\beta}_j^0 \boldsymbol{\beta}_j^{-1} - \boldsymbol{\mu}_i)^T (\boldsymbol{P}_{j,i}^T \boldsymbol{W}_j \boldsymbol{\beta}_j^{-1} - \boldsymbol{\beta}_j^0 \boldsymbol{\beta}_j^{-1} - \boldsymbol{\mu}_i) \tag{1.7}$$

1.2.3.2 Inference of the transformation terms $\boldsymbol{\beta}, \boldsymbol{\beta}_0$:

We can infer the transformation and translation terms $\boldsymbol{\beta}, \boldsymbol{\beta}_0$ by solving a weighted linear regression problem with a Mahalanobis norm for each neuron quantified by the covariance terms, $\boldsymbol{\Sigma}_i$:

$$\underset{\boldsymbol{\beta}_j^{-1}, \boldsymbol{\beta}_j^0 \boldsymbol{\beta}_j^{-1}}{\text{minimize}} \sum_{i=1}^{n_j} (\boldsymbol{P}_{j,i}^T \boldsymbol{W}_j \boldsymbol{\beta}_j^{-1} - \boldsymbol{\beta}_j^0 \boldsymbol{\beta}_j^{-1} - \boldsymbol{\mu}_i) \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{P}_{j,i}^T \boldsymbol{W}_j \boldsymbol{\beta}_j^{-1} - \boldsymbol{\beta}_j^0 \boldsymbol{\beta}_j^{-1} - \boldsymbol{\mu}_i)^T. \tag{1.8}$$

This system admits a fixed point iteration that yields the global minimum [Evangelidis and Horaud, 2018]. First, the closed form solution for $\boldsymbol{\beta}_j^0\boldsymbol{\beta}_j^{-1}$ is given by:

$$\boldsymbol{\beta}_j^0\boldsymbol{\beta}_j^{-1*} = \Big(\sum_{i=1}^{n}(\boldsymbol{P}_{j,i}^T\boldsymbol{W}_j\boldsymbol{\beta}_j^{-1} - \boldsymbol{\mu}_i)\boldsymbol{\Sigma}_i^{-1}\Big)\Big(\sum_{i=1}^{n}\boldsymbol{\Sigma}_i^{-1}\Big)^{-1} \tag{1.9}$$

To analytically solve for $\boldsymbol{\beta}_j^{-1}$, we use the fact that $\mathrm{vec}(ABC) = (C^T \otimes A)\mathrm{vec}(B)$ where $\mathrm{vec}(\cdot)$ denotes the vectorization operation and $\otimes$ denotes Kronecker product. This yields the following vectorized closed form update for $\boldsymbol{\beta}_j^{-1}$:

$$\mathrm{vec}(\boldsymbol{\beta}_j^{-1*}) = \Big(\sum_{i=1}^{n}(\boldsymbol{\Sigma}_i^{-1} \otimes (\boldsymbol{P}_{j,i}^T\boldsymbol{W}_j)^T(\boldsymbol{P}_{j,i}^T\boldsymbol{W}_j))\Big)^{-1}\Big(\sum_{i=1}^{n}\mathrm{vec}((\boldsymbol{P}_{j,i}^T\boldsymbol{W}_j)^T(\boldsymbol{\beta}_j^0\boldsymbol{\beta}_j^{-1} + \boldsymbol{\mu}_i)\boldsymbol{\Sigma}_i^{-1})\Big) \tag{1.10}$$

1.2.3.3 Permutation inference:

Lastly, we can solve for the doubly-stochastic matrix, $\boldsymbol{P}_j$ by setting up a $n \times n_j$ transport matrix $\boldsymbol{D}$ where

$$\boldsymbol{D}_{u,v} = (\boldsymbol{\mu}_u\boldsymbol{W}\boldsymbol{\beta}_j + \boldsymbol{\beta}_j^0 - \boldsymbol{W}_{j,v})\boldsymbol{\Sigma}_u^{-1}(\boldsymbol{\mu}_u\boldsymbol{W}\boldsymbol{\beta}_j + \boldsymbol{\beta}_j^0 - \boldsymbol{W}_{j,v})^T \tag{1.11}$$

and obtaining $\boldsymbol{P}_j$ through the solving the entropic optimal transport problem [Peyré et al., 2019] using the Sinkhorn-Knopp algorithm [Sinkhorn and Knopp, 1967] which minimizes the following objective:

$$\boldsymbol{P}_j^* = \underset{p\in\mathscr{P}}{\arg\min}\sum_{u,v} p_{u,v}\boldsymbol{D}_{u,v} - \gamma p_{u,v}\log p_{u,v} \tag{1.12}$$

Further details of permutation inference for neuron identification can be found in [Mena et al., 2020].

1.2.4 Results

In words, our algorithm operates in the following way. First, the targeted inference parameters are initialized using the neuron centers and colors for a random worm. Then,

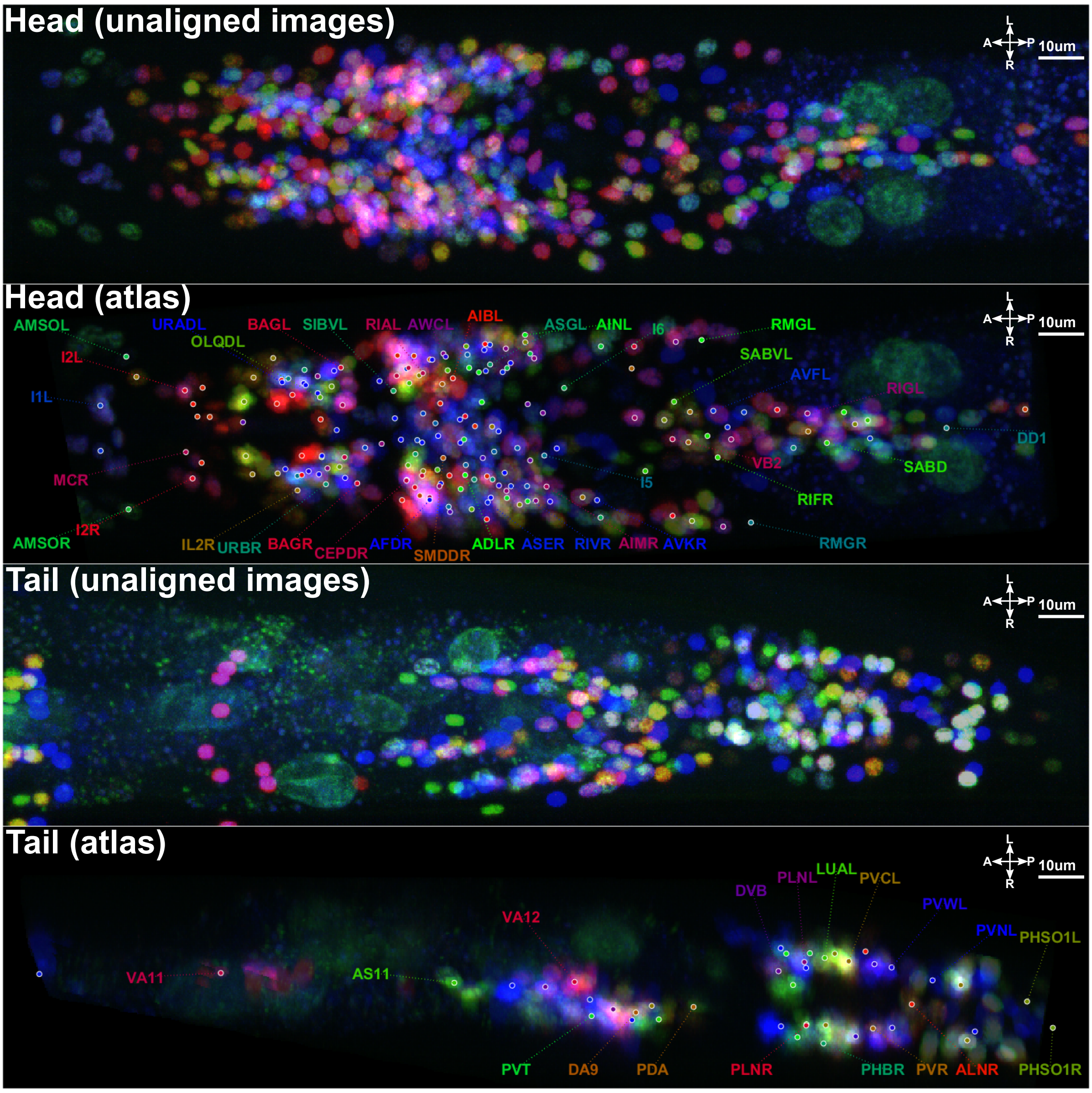

Figure 1.3: **Statistical atlas of *C. elegans* neurons** The construction of the statistical atlas of *C. elegans* neurons in the head and tail is demonstrated by contrasting the superposition of unaligned images of the 10 NeuroPAL worms (top row for head, third row for tail) with the superposition of aligned images to the converged atlas (second row for head, fourth row for tail). The canonical neuron positions and their NeuroPAL colors are represented as colored dots. A limited selection of neurons are annotated to avoid overcrowding in the figure. Note that the nerve ring (the hollow space in the head, one-third distance from the anterior) and the empty boundaries separating many of the worm ganglia, are distinct in the aligned images while indistinguishable in the unaligned images. See [Yemini et al., 2019b] Figure 2 for further details.

the remaining worms are affinely aligned to the hypothetical atlas by solving the linear system for $\{\boldsymbol{\beta}_j, \boldsymbol{\beta}_j^0\}$ in equation 1.10. The means and covariances of the aligned neurons are then used to update the atlas parameters of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. This procedure is iteratively repeated until convergence. See Fig. 1.3 for an illustration.

1.3 Male C. elegans Neural Atlas

It is generally appreciated that nervous systems are sexually dimorphic on a gross anatomical level. However, sex differences in nervous systems have been carefully mapped out, with single-cell resolution, in only very few animals. The nematode *C. elegans* is the only organism for which a complete cellular, lineage, and anatomical map of the entire nervous system has been described for both sexes (Fig. 1.7) [Cook et al., 2019b; Jarrell et al., 2012a; Sulston and Horvitz, 1977; Sulston et al., 1980]. With 383 neurons total, the nervous system of the male is almost 30% larger than that of the hermaphrodite (302 neurons). Based on lineage and anatomy and molecular profiles, 294 neurons are shared between both sexes. Hermaphrodites, which are somatic females, contain an additional 8 hermaphrodite-specific neurons that fall into two classes: the well characterized HSN and VC motor neuron classes, both of which control egg laying behavior [Schafer, 2005]. The male contains an additional 93 neurons that fall into 27 anatomically distinct classes [Cook et al., 2019a; Molina-García et al., 2020; Sammut et al., 2015; Sulston et al., 1980]. These 27 neuron classes are extensively interconnected and the structure of their interconnectivity displays a number of notable features, including modular substructures regulating subsequences of male mating behavior; multiple, parallel and short synaptic pathways directly connecting sensory neurons to end organs and recurrent, reciprocal connectivity among the male's many sensory neurons [Cook et al., 2019a; Jarrell et al., 2012a].

Of the 27 male-specific neuron classes, two are the head sensory neuron classes

CEM and MCM, two are the ventral nerve cord motor neuron classes CA and CP, and the remaining 23 classes are located in the tail of the animal. Some of the 27 male-specific neuron classes are composed of only a single neuron or two bilaterally symmetric neurons. Other neuron classes are composed of multiple class members: the A- and B-type ray sensory neurons are each composed of nine distinct bilateral pairs. With the exception of the four CEM sensory neurons in the head, which are born in the embryo and induced to die in hermaphrodites, all male-specific neurons are generated during postembryonic development from blast cells that proliferate and differentiate in a male-specific manner [Sulston et al., 1980]. Based on cell divisions patterns, the 87 postembryonically generated male-specific neurons are generated at different larval stages. Each individual larval stage contributes to the generation of some of these postembryonic neurons [Sulston et al., 1980]. However, when exactly these neurons terminally differentiate is poorly understood. Moreover, in his classic lineage studies Sulston also noted that the number of two male-specific neuron classes, DX and EF, display a variable number of class members [Sulston et al., 1980]. Since this observation was originally based on Nomarski optics and limited sample size, this variability has not been well characterized and has not been observed elsewhere within or outside the nervous system of *C. elegans*.

The vast majority of the sex-shared nervous system is generated in the embryo and synaptically connected by the first larval stage. Thus, one fascinating problem presented by the male-specific nervous system is how the many postembryonically generated, male-specific neurons become integrated into already existing circuitry. Of the 27 male-specific neuron classes, all but one (PCC) make synaptic contacts to sex-shared neurons. Understanding how such integration occurs may provide interesting insights for more complex vertebrate nervous systems, which are similarly characterized by the addition of new neurons throughout many stages of juvenile and even adult stages.

Despite many interesting aspects of the male nervous system, it has received little attention over the years when compared to the nervous system of the hermaphrodite. A number of studies have illuminated aspects of the development and function of male-specific neurons, but those studies only dealt with a limited set of neurons [Barr et al., 2018; Emmons, 2014, 2018; García and Portman, 2016; Garcia et al., 2001; Liu and Sternberg, 1995; Portman, 2017]. Hence, many aspects of the development and function of the 93 male-specific neurons remain uncharted territory. With some notable exceptions, including the systematic mapping of neurotransmitter identities [Gendrel et al., 2016; Pereira et al., 2015; Serrano-Saiz et al., 2017], marker analysis in the ray sensory neurons [Lints et al., 2004] and ventral nerve cord [Kalis et al., 2014], few molecular markers have been developed that label male-specific neurons. Single-cell transcriptome approaches have so far exclusively focused on the hermaphrodite [Cao et al., 2017; Packer et al., 2019; Taylor et al., 2021]. This dearth of molecular markers not only limits the ability to assess, for example, cell fate in specific mutant backgrounds, but also complicates the means by which cellular expression patterns in the male tail can be unambiguously identified.

Here, we address these shortcomings by showing that NeuroPAL, a previously described multicolor transgene that distinguishes all neuron classes in the hermaphrodites [Yemini et al., 2021], can also be used to disambiguate the 93 neurons of the male nervous system. We find that the NeuroPAL transgene, which harbors more than 40 promoters that drive the expression of four distinct fluorophores, generates a color map that provides sufficient discriminatory power to reliably identify all male-specific neurons. We provide proof-of-principle examples that show how to use NeuroPAL to identify gene expression patterns in the nervous system, and use the NeuroPAL color map to provide a number of insights into the development of the male-specific nervous system.

1.3.1 Results

NeuroPAL provides discriminatory color barcodes for all male-specific neurons With the exception of neurotransmitter pathway genes [Gendrel et al., 2016; Lints and Emmons, 1999; Pereira et al., 2015; Serrano-Saiz et al., 2017], few molecular markers have been comprehensively described for male-specific neurons (www.wormbase.org). For several related neuron classes, for example the ray neurons, molecular markers are available, but they do not provide sufficient resolution to distinguish between all individual class members [Lints and Emmons, 1999; Lints et al., 2004]. We set out to test whether the NeuroPAL transgene that we previously described for the *C. elegans* hermaphrodite [Yemini et al., 2021] would provide a similarly information rich molecular map of the male-specific nervous system.

The NeuroPAL transgene was designed to provide color codes to all neurons of the *C. elegans* hermaphrodite [Yemini et al., 2021]. This was achieved through the judicious use of four fluorophores with separable emission spectra (mTagBFP2, CyOFP1, Tag-RFP-T, mNeptune2.5), expressed under the control of a set of 43 different promoters with overlapping expression profiles (39 neuron-type specific promoters + 4 distinct, but fused panneuronal promoters) [Yemini et al., 2021]. Promoter choices were dictated by the goal of having neighboring neurons display distinct color codes, thereby unambiguously discriminating neighboring neuron identities from one another.

Using NeuroPAL to address stereotypy in the male-specific nervous system We first used NeuroPAL to address questions that relate to stereotypy of the male-specific nervous system. In his original lineage analysis of the male tail, John Sulston reported on an unusual phenomenon, not observed anywhere else in the entire organism: descendants of the U ectoblast produce variable numbers of DX and EF neurons, a notion indicated by stippled lines in Sulston's original lineage diagram. This violates the complete stereotypy and deterministic nature of all cell lineages, both neuronal and non-neuronal. Moreover, according to the Sulston lineage diagram, this variability is restricted to the EF and DX neurons that descend from the U neuroblast and that are located in the preanal ganglion (the EF3 & 4 and DX3 & 4 neurons). In contrast, the DX and EF neurons that are produced from the F neuroblast (EF1 & 2, DX1 & 2), located in the dorsal rectal ganglion, were generated in an apparently invariant manner (as per the Sulston lineage diagram). However, no quantification of this was provided. Because the lineage analysis entirely relied on cleavage pattern alone, it was also not clear to what extent the variably produced DX and EF neurons acquire a differentiated state.

Using NeuroPAL, we examined 22 young adult males and found variability in the presence of fully differentiated EF and DX neurons in the preanal ganglion Fig. 1.6A – assessed by wild-type expression of NeuroPAL colors in these neurons. Within the F-derived dorsorectal ganglion, 22/22 animals invariably showed two fully differentiated DX neurons (DX1 and DX2) and two EF neurons (EF1 and EF2), corroborating John Sulston's observations. In the U-derived preanal ganglion, 19/22 animals show one DX and one EF neuron (= DX3 and EF3), 1/22 had one additional EF (= EF4), and 2/22 had one additional EF (= EF4) and one additional DX (= DX4).

The EF and DX neurons are also the neurons with the greatest inter-animal variability in their relative positioning. We arrived at this conclusion by closely considering the overall variability of positioning of both sex-shared, as well as sex-specific neurons in the tail of the animal. We had previously shown that in the hermaphrodite head, where the vast majority of neurons are generated embryonically, most cells are positioned within a small volume of variability [Yemini et al., 2021] and we observer a similar extent of variability in the male head Fig. 1.6. However, in the tail, where the vast majority of the postembryonically added male-specific neurons are located, there is substantially more positional variability, both in the sex-shared neurons as well as in the sex-specific neurons Fig. 1.6. The EF and DX neurons stand out in the extent of variability in their positioning. It will be interesting to investigate whether the inter-animal variability in neuronal soma position in the male tail also translates into variability in neuronal process adjacency, and hence connectivity, between individual animals.

1.4 Extensions to Deformable Models and Unsupervised Atlases

1.4.1 Methods

In brief, we model each observation, e.g., individual images of *C. elegans* brains or fruit fly wings as a random draw from a probability distribution subjected to a random postural transformation. We call the parameters of this latent probability distribution, the "atlas" (**Fig.1.8A**). We infer the latent atlas parameters and the transformation terms by formulating the generative process through a neural network and use spatial transformers [Jaderberg et al., 2015] to perform differentiable optimization (**Fig.1.8B**). Details of these steps can be found in the following sections.

Notation: We start by introducing the notation. We denote the atlas as a latent variable $\boldsymbol{Z} \in \mathbb{R}^D$ following the distribution $P_{\boldsymbol{\theta}}(\boldsymbol{Z})$. Both $\boldsymbol{X}, \boldsymbol{Z}$ random variables can be high-dimensional or low dimensional depending on the application. For example a statistical atlas of *C. elegans* neural positions is constructed using point clouds that are lower dimensional compared to an atlas that is constructed using pixelwise images [Varol et al., 2020].

Generative model: Given the atlas, i.e. a distribution over the random variable $\boldsymbol{Z}$ the observations $\boldsymbol{X_i}$ are samples from the prior $\boldsymbol{Z_i}$ that are transformed according to some biological transformation $f_{\boldsymbol{\beta_i}} \in \mathscr{F}$ where $\mathscr{F}$ is a function family containing feasible transformations between the atlas and observations. For example, in the *C. elegans* hermaphrodite, the feasible transformation family is the space of rigid transformations $\{\boldsymbol{S},\boldsymbol{T}\}$ where $\boldsymbol{S} \in \mathscr{SO}_3$ and $\boldsymbol{T} \in \mathbb{R}^3$ and piecewise rigid transformations.

Inference and optimization: Once we specify $\mathscr{F}$ and the functional form of $P_{\boldsymbol{\theta}}$ then our goal is to solve the inverse problem and find $\boldsymbol{\theta},\boldsymbol{\beta}_{1:n}$. To do this, we write a probabilistic cost function informed by our statistical model and optimize it w.r.t. $\boldsymbol{\theta},\boldsymbol{\beta}_{1:n}$:

$$\boldsymbol{Z}_i \sim P_{\boldsymbol{\theta}}(\boldsymbol{Z}) \quad \boldsymbol{X}_i|\boldsymbol{Z}_i \sim P(\boldsymbol{X}|\boldsymbol{Z}_i) = f_{\boldsymbol{\beta}_i}(\boldsymbol{Z}_i) + \epsilon_i$$

$$\mathscr{L}(\boldsymbol{\theta},\boldsymbol{\beta}_{1:n}) = \log P_{\boldsymbol{\theta},\boldsymbol{\beta}_{1:n}}(\boldsymbol{X}_{1:n},\boldsymbol{Z}_{1:n}) = \sum_{i=1}^{n} \log P_{\boldsymbol{\beta}_i}(\boldsymbol{X}_i|\boldsymbol{Z}_i) + \log P_{\boldsymbol{\theta}}(\boldsymbol{Z}_i)$$

We take an alternating approach for optimizing $\mathscr{L}$ where we iteratively optimize $\mathscr{L}$ w.r.t. $\boldsymbol{\theta}$ and $\boldsymbol{\beta}_{1:n}$. Given our estimate of the values $\boldsymbol{\beta}_{1:n}$ denoted by $\hat{\boldsymbol{\beta}}_{1:n}$ we find the best fit $\hat{\boldsymbol{\theta}}$ to the data in the following way:

$$\hat{\boldsymbol{\theta}} = \max_{\boldsymbol{\theta}} \mathscr{L}(\boldsymbol{\theta}|\hat{\boldsymbol{\beta}}_{\boldsymbol{1:n}}) = \max_{\boldsymbol{\theta}} \sum_{i=1}^{n} \log P_{\boldsymbol{\theta}}(f_{\hat{\boldsymbol{\beta}}_i}^{-1}(X_i)) \tag{1.13}$$

Notice that here we are trying to find the sufficient statistics of $P_{\boldsymbol{\theta}}$ from known observations $\boldsymbol{Z}_{1:n}$. For the case of multivariate normal distributions where $\boldsymbol{\theta} = \{\boldsymbol{\mu},\boldsymbol{\Sigma}\}$ the maximum likelihood estimate (MLE) is the empirical mean and covariance. However, our formulation allows for incorporating arbitrarily complex distributions where we solve the MLE problem using stochastic variational inference in the parameter space. This is facilitated by probabilistic programming where generic algorithms for MLE and MAP estimation are provided. In the result section, we show an example of using Dirichlet as the prior distribution over the colors in *C. elegans* point clouds.

On the other hand if we have a reasonable estimate of $\boldsymbol{\theta}$ then in order to update our estimates of $\boldsymbol{\beta}_{1:n}$ we need to solve the following for each i:

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{i}} = \max_{\boldsymbol{\beta}_i \in \mathscr{F}} \mathscr{L}(\boldsymbol{\beta}_i|\hat{\boldsymbol{\theta}}) = \max_{\boldsymbol{\beta}_i \in \mathscr{F}} \log P_{\hat{\boldsymbol{\theta}}}(f_{\boldsymbol{\beta}_i}^{-1}(X_i)) \tag{1.14}$$

Depending on the function family $\mathscr{F}$ analytical solutions might exist but in general specific algorithms need to be developed for particular choices of $\mathscr{F}$. We have provided derivations for rigid and piecewise rigid function families in the appendix but here we provide a general deep learning architecture for arbitrarily complex function families.

We consider $\boldsymbol{\beta}_i$ to be a function of the input $\boldsymbol{X}_i$ and use a neural network parameterized by ϕ to learn the transformation family. The neural network architecture is determined based on the input data type where we use convolutional architectures for volumetric or planar image data and fully connected for point clouds. The output of the network provides $\boldsymbol{\beta}_i$, for example, rigid transformations are determined by 6 parameters in 3 dimensions with 3 parameters for the rotation angles along different axes and 3 parameters for the translations. Hence the output dimension of the neural network for rigid transformation family is 6. In the case of piecewise rigid transformations, the number of parameters depend on the number of pieces with 6 parameters for each piece.

Learning the transformation family: If the transformation parameters are known (e.g. for rigid or affine transformations), or can be driven analytically or algorithmically, we can train the network by directly minimizing the error of transformation parameters $\mathscr{L}(\phi) = \sum_{i=1}^{n} \left\| \bar{\boldsymbol{\beta}}_i - \boldsymbol{\beta}_i(\boldsymbol{X}_i;\phi)) \right\|^2$. Otherwise, we train the neural network by optimizing over the loss function in equation 1.14. To compute the gradients of loss w.r.t. ϕ we need to apply the inverse transformation in a differentiable way w.r.t. ϕ. For point clouds, this is straightforward since the transformation family is assumed differentiable. For image inputs, differentiable transformation is made possible by the recent development of differentiable grid sampling for spatial transformers [Jaderberg et al., 2015] where transformation parameters are used to define a parameterized sampling grid that maps the image to a target location.

1.4.2 Results

1.4.2.1 Supervised atlas of hermaphrodite *C. elegans* neuron positions:

We used a public dataset of 10 point clouds of hermaphrodite *C. elegans* tail neurons with 42 neurons per worm [Yemini et al., 2021]. Each neuron in each worm has a 3D location and a RGB color represented by a 6D vector. We chose the prior distribution over the positions to be multivariate normal (MVN) as suggested by prior work [Bubnis et al., 2019; Varol et al., 2020]. However, for the color distribution we experimented with Dirichlet and MVN distributions. We also experimented with two different transformation family for the spatial component of the point clouds, namely rigid (R) and regularized piecewise rigid (PR) transformations. The transformation family for the color component is a simple `softmax` operator.

We used a fully connected architecture for $\boldsymbol{\phi}$ and parameterized rigid and piecewise rigid transformations using 3 angle and 3 translation parameters per piece. The optimization is performed using `Adam` optimizer with learning rate $1e-4$. The updates of $\boldsymbol{\beta}_{1:N}$ are performed by backpropagating the gradients of $\boldsymbol{\phi}$ while we used `Pyro SVI` tool for maximum likelihood estimation of $\boldsymbol{\theta}$. Details on the optimization and implementation can be found in the supplementary.

In **Fig. 1.9**, we illustrate the atlas parameters $\boldsymbol{\theta}$ and aligned point clouds $\boldsymbol{Z}_{1:N}$ as well as the training and testing likelihoods. Our results show that Dirichlet captures the color distribution better than MVN evaluated by test log likelihood (5-fold cross validated) while R and PR transformation families achieve comparable test log likelihood.

1.4.2.2 Semi-supervised atlas of male *C. elegans* images and partial annotations:

Male *C. elegans* images contain denser subsets of neurons in smaller regions making it more difficult to annotate all the neurons manually. Here we showcase the flexibility of our framework in this semi-supervised setting by applying it directly to the image space and using the partial annotations to guide the transformation. The inputs in this case are 12 images of male *C. elegans* (not point clouds), hence we used a convolutional architecture for $\boldsymbol{\phi}$ but the transformation families are chosen to be R and PR as before. We experimented using subsets of annotations with various sizes (5, 10, 20) and observed that the test error (in terms of the number of pixels) drops with more annotations, shown in **Fig. 1.10f,g**. Furthermore, the alignment parameters lead to more biologically feasible transformations when we include more annotations (**Fig. 1.10a-d**). We then applied the transformations found in the image space to the neural point clouds and constructed a semi-supervised atlas of male neurons shown in **Fig. 1.10e**. The transformations for the semi-supervised atlas is inferred using 66 neurons.

1.4.2.3 Unsupervised atlas of transgenic *D. melanogaster* wings:

We used a public dataset [Sonnenschein et al., 2015] of 128 fruit fly images from 4 genotypes and 2 sexes to infer a latent atlas that represents an average wing that is corrected for postural differences by a piecewise rigid motion model. The resulting atlas can be seen in **Fig. 1.11**. Using the atlas coordinate framework, we performed pixelwise t-test on the aligned wings of females and males to observe statistically significant differences in the wing tip density in the medial part of the wing. Furthermore, our results show morphological differences between genotypes, with *egfr* known to modulate the morphology of veins [Roch et al., 2002].

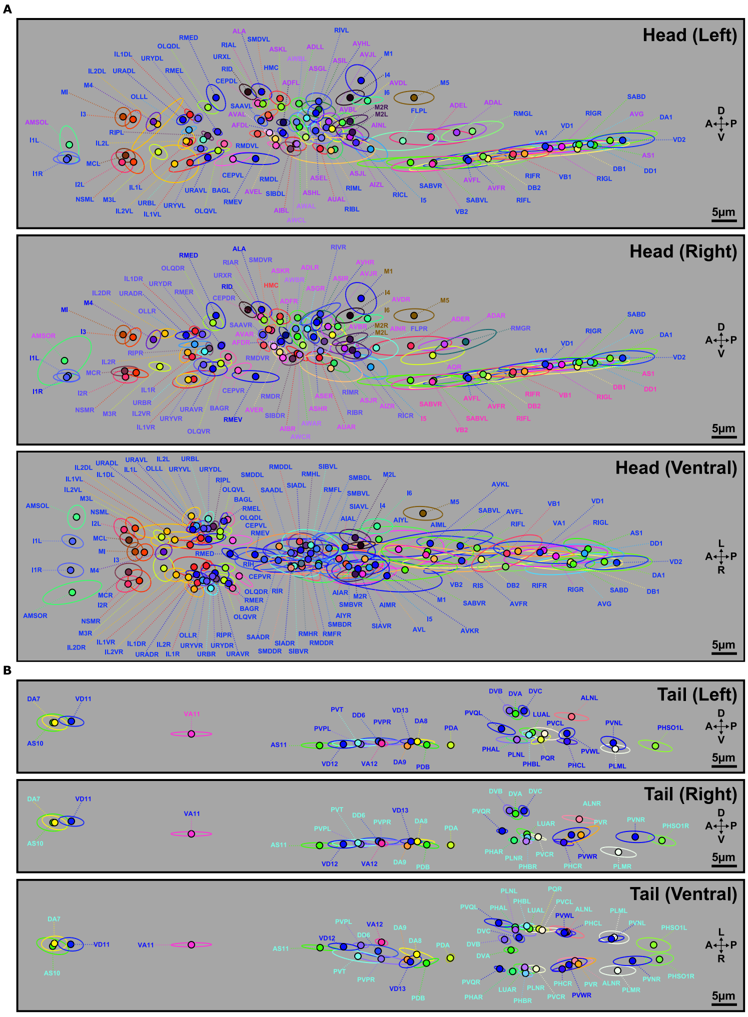

Figure 1.4: **Neuron locations and their positional variability** (A) Neuron locations and variability, in the retrovesicular ganglion, taken from electron micrographs of three adult hermaphrodites N2S, N2T, and N2U [White et al., 1986a]. (B) An example of substantial positional variability. The OLL left (OLLL) and right (OLLR) neurons, within a single animal, should share equivalent positions. Instead they show substantial anterior-posterior displacement relative to each other. The transgenic reporters and their pseudo colors are noted on the figure. (C,D) Canonical neuron locations (filled circles displaying the NeuroPAL colors) and their positional variability (encircling ellipses with matching colors) for all ganglia, as determined by NeuroPAL (otIs669). Positional variability is shown as the 50% contour for neuronal location (measured as a Gaussian density distribution), sliced within a 2D plane. We show both the left-right and dorsal-ventral planes to provide a 3D estimation of positional variability. (C) Left, right, and ventral views of the head neuron positions. OLLR exhibits over twice the positional variability of OLLL in its anterior-posterior axis, echoing the displacement seen with the non-NeuroPAL transgene in panel B. (D) Left, right, and ventral views of the tail neuron positions.

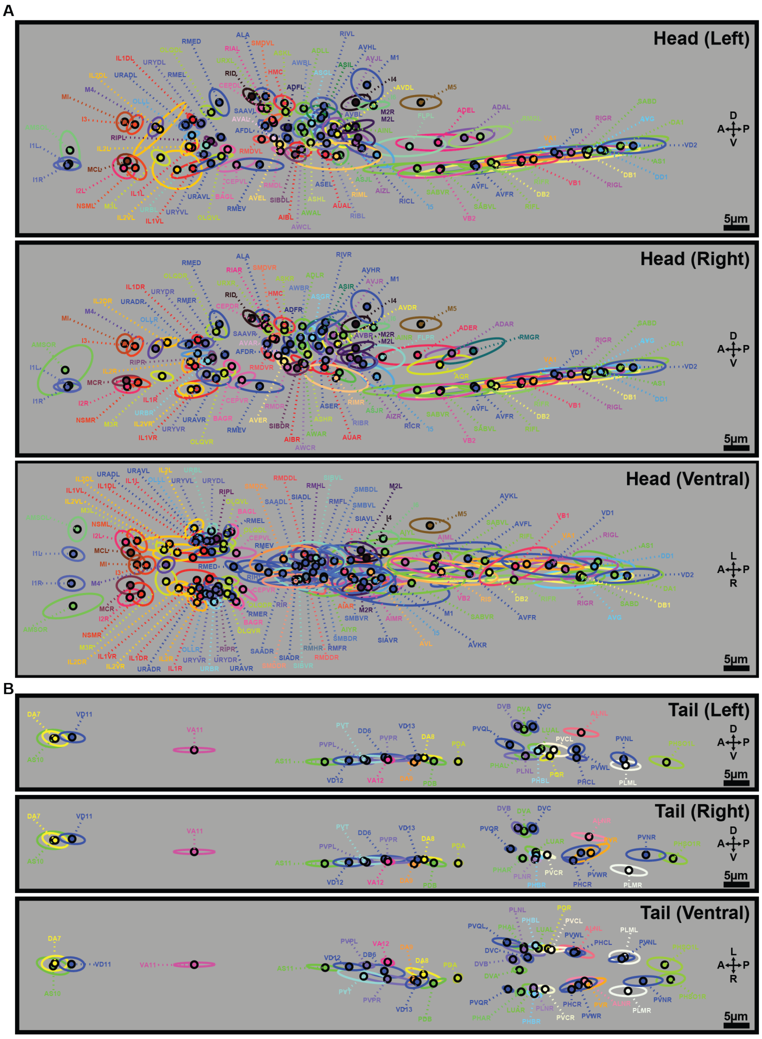

Figure 1.5: **Canonical neuron locations and their positional variability** Canonical neuron locations (filled circles with their NeuroPAL coloring) alongside their positional variability (encircling ellipses with matching color) for all ganglia in the head (**A**) and tail (**B**), as determined by NeuroPAL (*otIs669*). Positional variability is displayed as the 50% contour for neuronal location (measured as a Gaussian density distribution), sliced within a 2D plane; because we are restricted to a planar view, we show both the left-right and dorsal-ventral planes to provide a 3D estimation of the true contour bounding positional variability. Left, right, and ventral views of neuron position variability is shown.

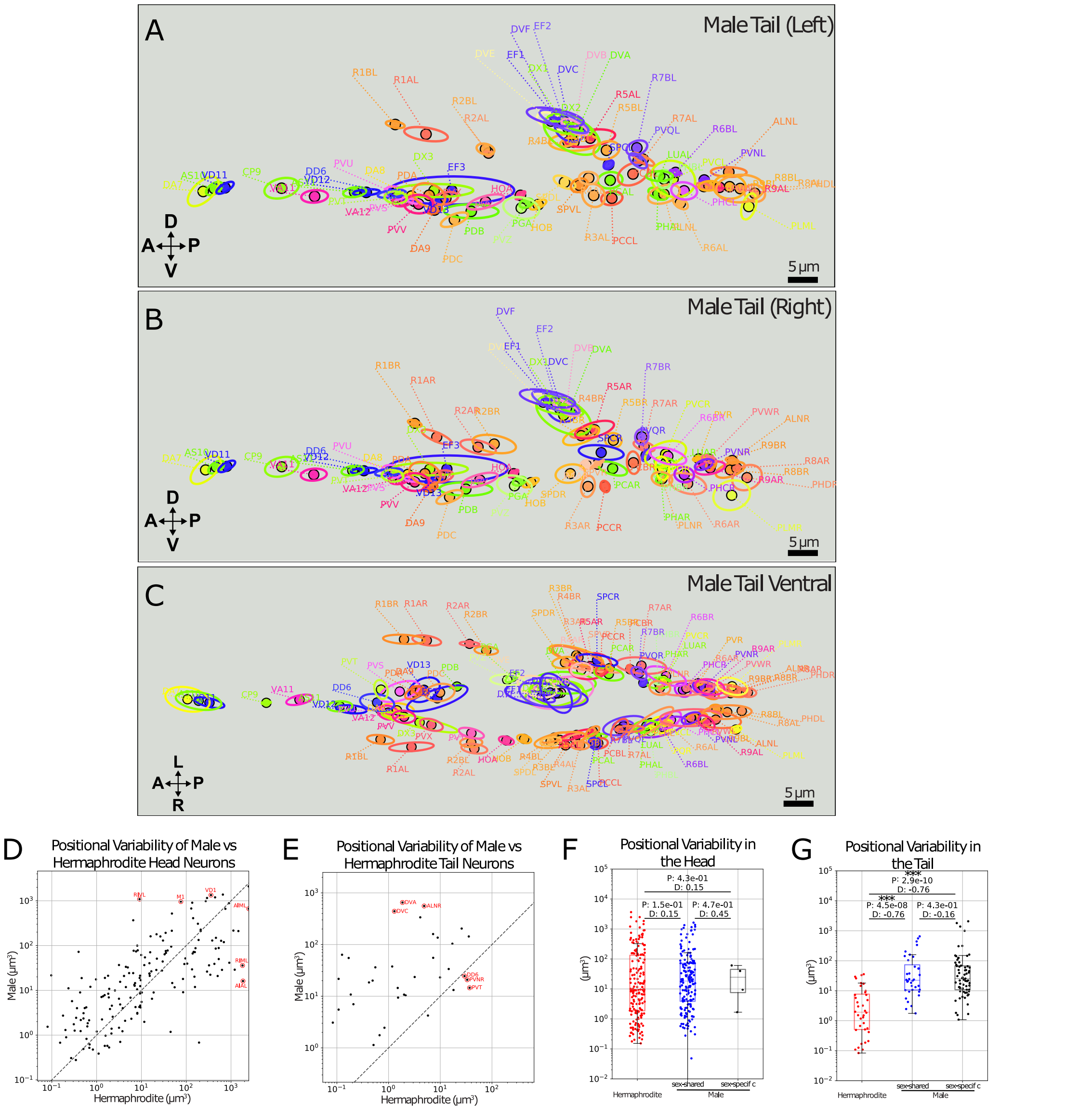

Figure 1.6: **Variability of cell generation and position in the adult male tail** A-C: The atlas of male tail neuron positional variability (based on 13 male tails) for the left (A), right (B), and ventral (C) sided views. Dots indicate the mean position of each neuron. Ellipses indicate the positional variability of each neuron in the given axis. Neurons colors approximate those in NeuroPAL but have been brightened for visibility.

Figure 1.6: (cont. from previous page) D-E: Positional variability of the individual hermaphrodite versus male neurons in the head (E) and tail (F). Six neurons that show maximal differences between both sexes are circled and identified. F-G: Quantification of positional variability for the collection of all head (G) and tail (H) neurons of the hermaphrodite (which are all sex-shared) versus the male sex-shared and sex-specific neurons. In the head, the positional variability is nearly the same for these three neuron groups. In the tail, the positional variability for the group of hermaphrodite neurons is far less than that of the male sex-shared and sex-specific neurons. We report the P-value (Mann-Whitney U test) for differences between hermaphrodites and males and the effect size (Cohen's D). For the head N = 10 hermaphrodites, 12 males, 182 sex-shared neurons, and 6 male-specific neurons, with a mean of 9.6 neurons/hermaphrodite and 9.8 neurons/male. For the tail N = 10 hermaphrodites, 13 males, 41 sex-shared neurons, and 69 male-specific neurons, with a mean of 9.6 neurons/hermaphrodite and 11.6 neurons/male. Further hermaphrodite and male atlases can be found in Fig. 1.7.

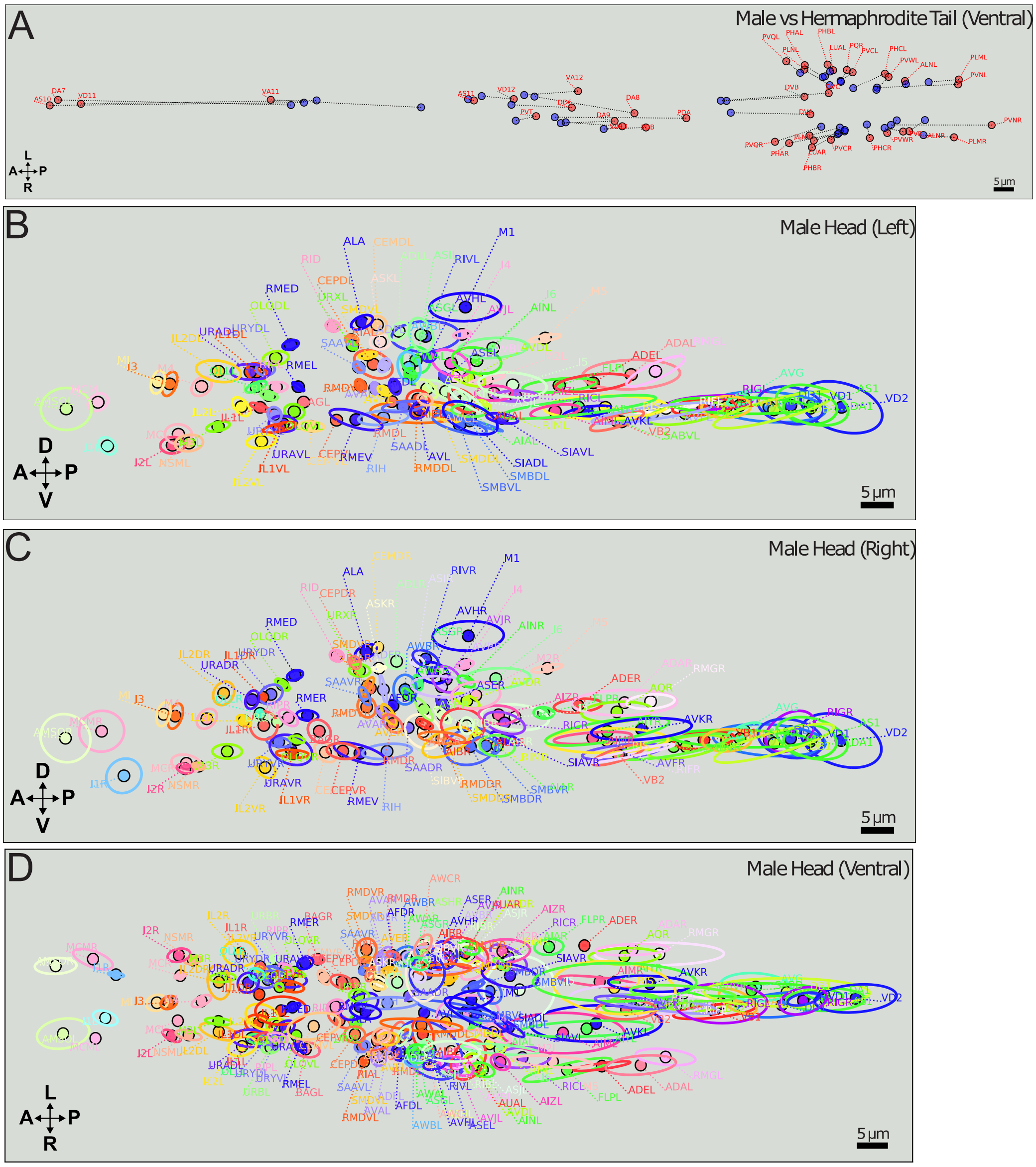

Figure 1.7: **Positional variability in the male versus hermaphrodite head** A: Ventral view of the positions of the hermaphrodite (red) and male (blue) neurons (circles) in the tail. The sex-shared neurons are linearly aligned to each other, labeled, and corresponding pairs are connected by a line. Note that, whereas most sex-shared neurons are positioned similarly in both sexes, the beginning of the hermaphrodite VNC is displaced anterior to its male counterpart and, in contrast, the hermaphrodite dorsorectal ganglion neurons are displaced posterior to their male counterparts. B-D: The atlas of male neuron positional variability (based on 12 male heads) for the left (B), right (C), and ventral (D) sided views. Dots indicate the mean position of each neuron. Ellipses indicate the positional variability of each neuron in the given axis. Neurons colors approximate those in NeuroPAL but have been brightened for visibility.

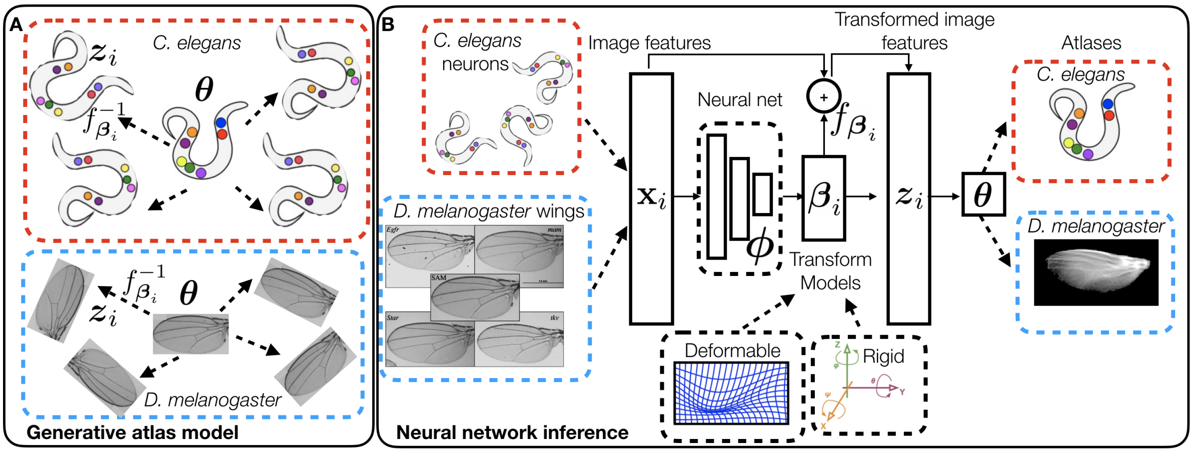

Figure 1.8: **Schematic of generative model of atlas construction A**: Each observation ($\boldsymbol{Z}_i$) is modeled as a random draw from an atlas parametrized by $\boldsymbol{\theta}$ and perturbed by transformation $f_{\boldsymbol{\beta}_i}^{-1}$. **B**: We infer atlas parameters ($\boldsymbol{\theta}$) from observations ($\boldsymbol{X}_i$) by optimizing a neural network loss function that penalizes the distance of each transformed observation ($\boldsymbol{Z}_i$) to the latent atlas ($\boldsymbol{\theta}$). Through this process, we also learn the transformation model parameters, ($\boldsymbol{\beta}_i$) that minimizes the loss. Inference is performed using differentiable grid sampling [Jaderberg et al., 2015].

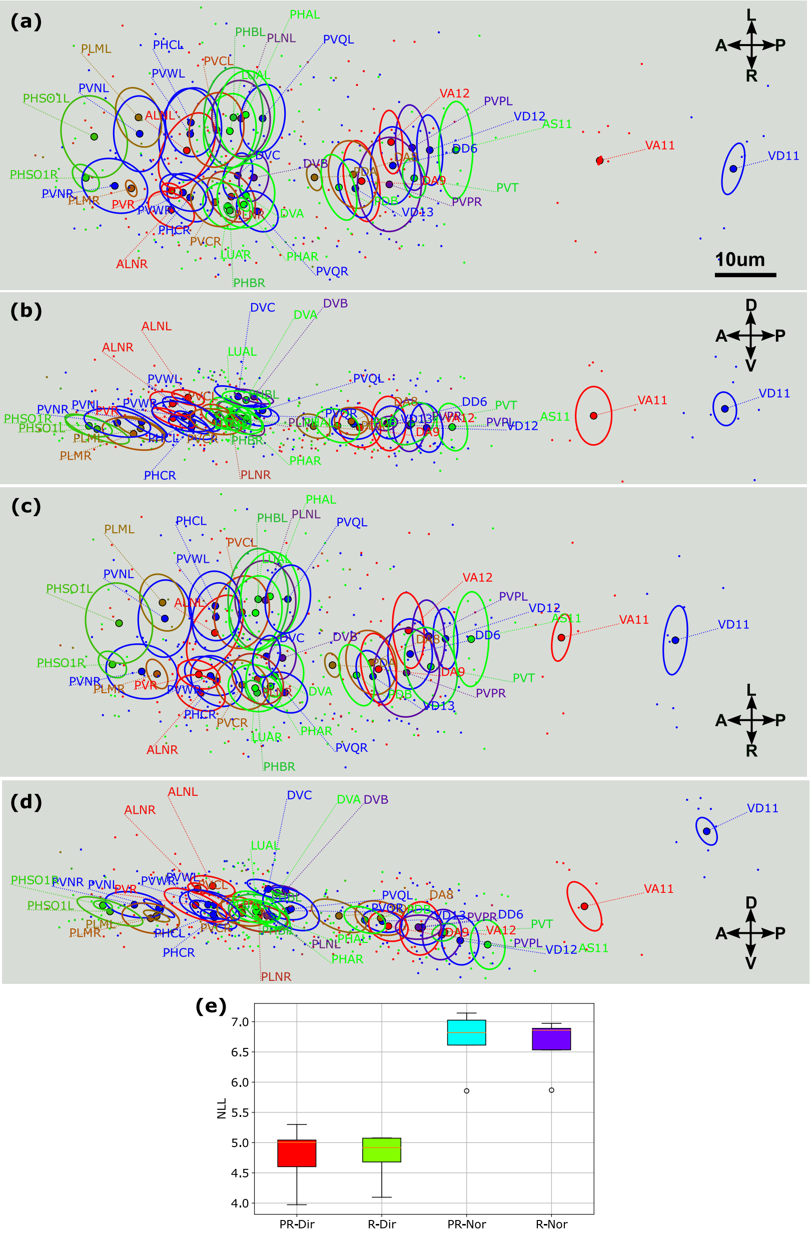

Figure 1.9: **Supervised positional and color atlas of tail neurons of hermaphrodite *C. elegans*A,B:** Dorsal ventral (**A**) and left-right view (**B**) of the atlas constructed by piecewise rigid transformations. Small dots indicate individuals' neural positions, larger dots indicate mean positions in the atlas and ellipses indicate one standard deviation of mass. **C:** The training negative log-likelihood (NLL) under different transformations and color models (PR: piecewise rigid, R: rigid, Dir: Dirichlet, Nor: normal). **D:** Testing error. Piecewise rigid motion model with Dirichlet color model has the lowest NLL in both training and testing samples, indicating appropriateness of modeling motion and color. Additional results can be found in the supplementary material.

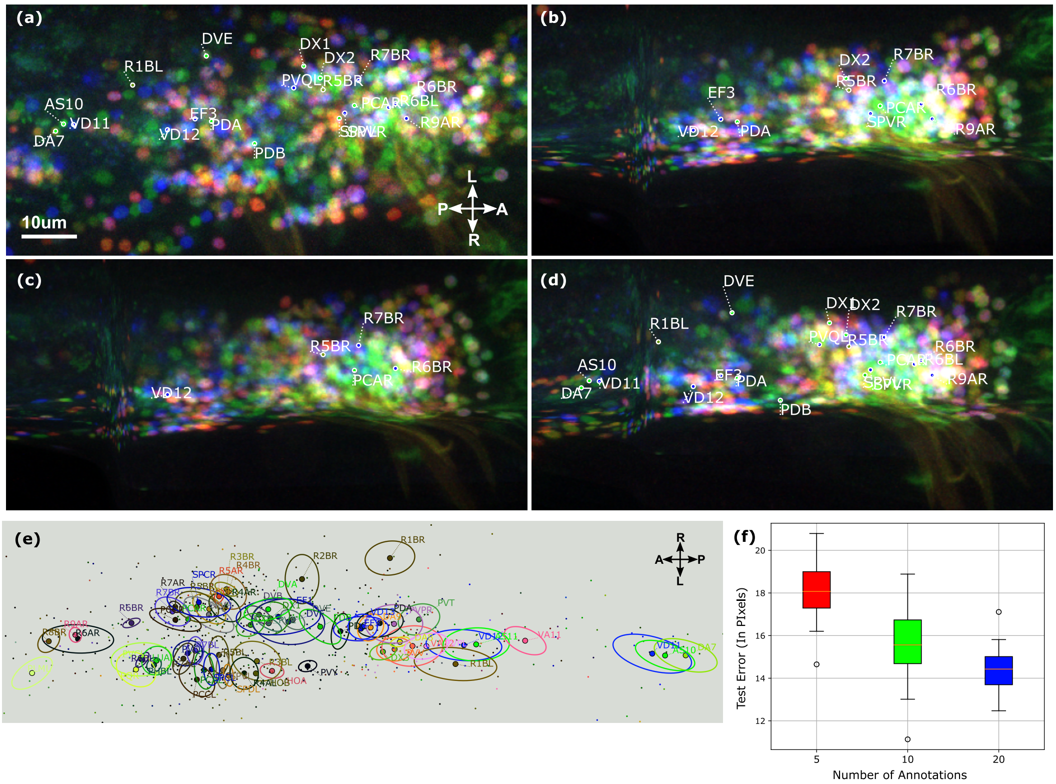

Figure 1.10: **Semi-supervised positional and color atlas of tail neurons of male *C. elegans*** We show the progressive effect of including more annotated neural positions to atlas quality.**A:** Superposition of unaligned NeuroPAL [Yemini et al., 2021] strain male worms.**B-D:**Superposition of worms aligned to an atlas that is trained using 5,10,20 neuron annotations per worm. **E:** The means and covariances of neural positions and colors inferred using semi-supervised atlas training. **F:** Out of sample alignment error decreases with increasing number of neural annotations. **G:** Training loss is minimized when more annotations are provided. This is because posture can be better estimated with more information about neuron location. Additional results can be found in supplementary material.

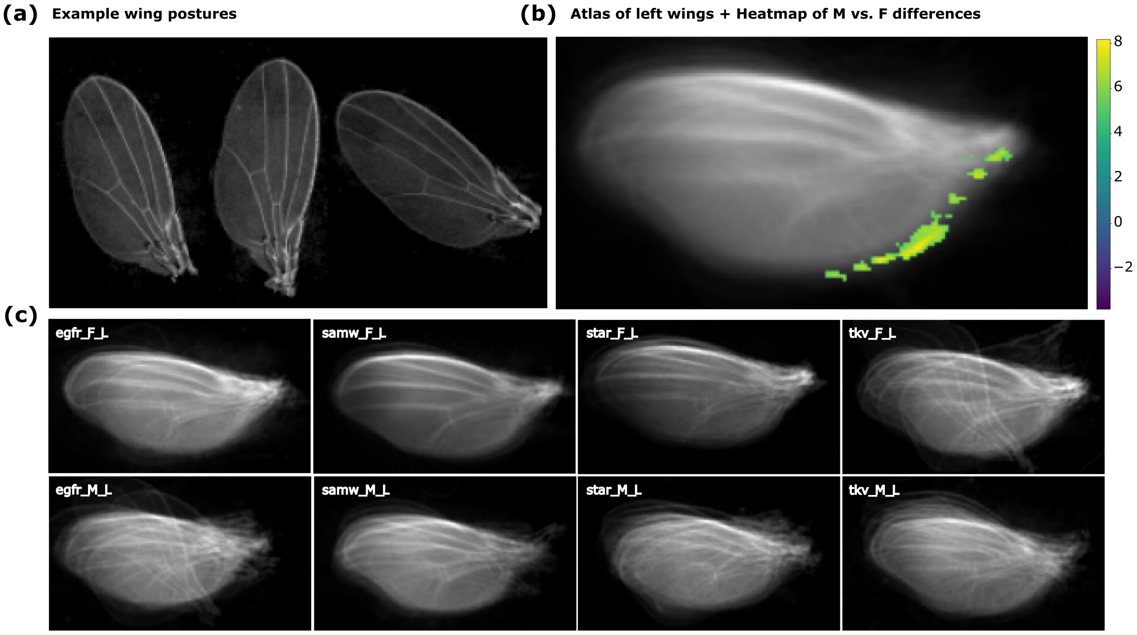
(a) Example wing postures
(b) Atlas of left wings + Heatmap of M vs. F differences
8
6
4
2
0
−2
(c)
egfr_F_L
samw_F_L
star_F_L
tkv_F_L
egfr_M_L
samw_M_L
star_M_L
tkv_M_L

Figure 1.11: **Unsupervised atlas of fruit fly wing** We infer a latent canonical atlas wing in the pixel space without the use of any markers or annotations (**B**) using 128 example images of fruit fly wings in varying poses (**A**). **C:** Averaging wing images of different genotypes and sexes enables a visual comparison of morphological differences between these groups. **B-heatmap:** Pointwise t-statistics ($q < 0.05$) between males and females yields a heatmap that shows that females have more mass in the the medial part the wing than males.

CHAPTER 2

From Raw Data to Scientific Discovery in C. elegans

2.1 Introduction

Limited by technology, classical neuroscience focused on recording from single neurons or hand-selected neurons responsive to specific task parameters. Small scale recordings led to the discovery of neural tuning curves and feature selective neurons such as face neurons or place cells. More recently, neuroscientists have become able to record from much larger populations of neurons. These recordings gave rise to a new understanding of how neural circuits work in coordination, complementing previous discoveries [Urai et al., 2022]. For example, a recent paper argues that neural tunings can be interpreted as projections of a latent structure in the neural state space with specific geometrical properties well suited for performing particular computations [Kriegeskorte and Wei, 2021]. This and many other examples corroborate the necessity of large scale recordings for a more holistic characterization of neural circuits specifically in the context of behavior.

There has been an explosion in the experimental technologies for recording from large neural populations. With microscopy imaging being in the front line of these techniques, new imaging modalities are continuously proposed enabling us to access functional or structural information about the underlying tissue. Some imaging techniques make the structures visible without needing exogenous markers while others such as fluorescence microscopy rely on labelling cells using fluorescence markers. The tissues are then imaged using various imaging modalities including light-sheet, light-field, wide-field or multi-photon microscopy. The images can vary largely due to the imaging techniques and modality, and the properties of the tissue itself. Ultimately, the collected datasets are merely a proxy for the actual signals of interest. It is then crucial to build tools for efficient and scalable extraction of the signals from datasets. Although deep learning revolutionized image and video processing for biological applications in recent years, but the vast majority of existing techniques rely on large training datasets annotated by experts. These datasets often do not exists for novel applications and new imaging modalities. Thus there is a critical need to analysis pipelines that can assist experimentalists without requiring large training datasets.

In this chapter, we describe the analysis pipeline developed for the recently proposed imaging modality NeuroPAL. Our proposed pipeline consists of registration [Nejatbakhsh and Varol, 2021], segmentation [Nejatbakhsh et al., 2020b], neural tracking [Yu et al., 2022], and signal extraction [Nejatbakhsh et al., 2020c] each of which is described in more detail in one of the subsequent sections. Notably,our pipeline enabled computation of the functional connectivity of *C. elegans* neurons and identification of neuronal differentiation defects in *C. elegans* mutants [Yemini et al., 2021]. We emphasize that categorizing the analysis techniques as a pipeline consisting of the above steps is general and can be applied to virtually every single animal model and emerging imaging modality. All components of our pipeline are included in an open source software with graphical user interface allowing experimentalists to efficiently detect neurons and uniquely resolve their identities in *C. elegans* (Fig. 2.1).

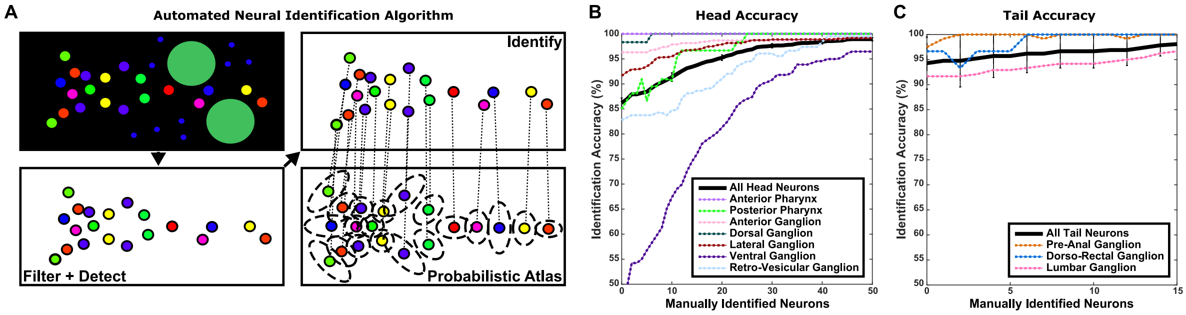

Figure 2.1: **NeuroPAL software: an algorithm for semi-automated neuronal identification and an algorithm to generate optimal-coloring solutions for cell identification** See **Text S1-S2** for algorithmic details and validation. (A-C) The algorithm used for automated neural identification. (A) Raw images are filtered to remove non-neuronal fluorescence and neurons are detected in the filtered image. Detected neurons are identified by matching them to a statistical atlas of neuronal colors and positions. (B,C) Automated neuronal identification accuracy begins at 86% for the head and 94% for the tail. Manually identifying eight neurons raises the head accuracy above 90%. Overall accuracy is displayed as a black line. Accuracy for each ganglion is displayed as a dotted, colored line (see legend). Many of the neurons and ganglia have high identification accuracy and confidence. The ventral ganglion is a problematic area, likely due to the high positional variance therein. (D-E) The algorithm used to generate optimal-coloring solutions for cell identification (for any collection of cells in any organism). We show simulations of two theoretically-optimal alternatives to NeuroPAL, one that permits as many reporters as NeuroPAL (D) and one that restricts the transgene to only 3 reporters (E). With the exception of the number of reporters, both alternatives were generated using parameters similar to NeuroPAL: three landmark fluorophores, where each fluorophore is distinguishable at three intensities (high, medium, and low). Reporters were chosen by the algorithm from those available in WormBase, a community-curated database of cell-specific reporter expression. Similar databases are available for other model organisms (e.g., fly, fish, and mouse). We evaluated the two NeuroPAL alternatives by computing the percentage of their color violations, defined as neighboring neuron pairs with indistinguishable colors.

2.2 C. elegans Neural Point Cloud Registration

Point set registration is one of the central problems in computer vision that involves the optimization of a transformation that aligns two sets of point clouds [Tam et al., 2013; Van Kaick et al., 2011]. Point set registration have been applied in numerous fields including but not limited to robotics [Zhang and Singh, 2015], medical imaging [Audette et al., 2000], object recognition [Drost et al., 2010], panorama stitching [Bazin et al., 2014] and computational neuroscience [Bubnis et al., 2019]. The types of allowable transformations and energy functions utilized in the cost function have differentiated varying methods [Aiger et al., 2008; Besl and McKay, 1992; Bustos et al., 2019; Enqvist et al., 2009; Hast et al., 2013; Indyk et al., 1999; Irani and Raghavan, 1999; Maron and Lipman, 2018; Mellado et al., 2014; Mount et al., 1999; Myronenko and Song, 2010; Pokrass et al., 2013; Tam et al., 2013; Yang et al., 2020; Zhou et al., 2016]. In general, point set registration methods employ an iterative strategy of solving the transformation and updating the matching which works well in practice but there are no guarantees for reaching the global optima [Chetverikov et al., 2002]. Only a few methods have provided approximate globally optimal solutions [Yang et al., 2016; Zhou et al., 2016]. These methods rely on severe constraints of the transformation domains, such as the 3D rotation group SO(3), in order to employ branch and bound techniques on discretizations.

Theoretical analysis of the recovery guarantees of point set registration has not been performed for a general number of dimensions until recently when it was termed as *unlabelled sensing* by [Unnikrishnan et al., 2015] as a problem with duality connections with the well-known problem of compressed sensing [Donoho et al., 2006]. In this problem, similar to linear regression, the response signal is modeled as a linear combination of a set of covariates. However, the correspondence of the responses to the covariates is modeled as having been shuffled by an unknown permutation matrix. For this reason, the problem has also been termed as *linear regression with shuffled labels* [Abid et al., 2017], *linear regression with an unknown permutation* [Pananjady et al., 2016], *homomorphic sensing* [Tsakiris and Peng, 2019] or *linear regression without correspondence* (RWOC) [Hsu et al., 2017], the latter of which will be used to refer to the problem herein. Although RWOC is, in general, an NP-hard problem [Pananjady et al., 2016], there have been several advances in recent years to propose signal to noise ratio (SNR) bounds for recovery of the permutation matrix and the regression coefficients [Pananjady et al., 2016; Unnikrishnan et al., 2018]. Conversely, the same works have also analyzed the SNR and sampling regime by which no recovery is possible.

Nevertheless, the computer vision community has attempted to solve the point set registration problem through consideration of outliers and missing correspondences, which are typically encountered in real-world applications. A common technique used in point set registration to robustify the optimization against outliers is to employ random sampling consensus (RANSAC) subroutines [Fischler and Bolles, 1981; Torr and Zisserman, 2000; Yang and Carlone, 2019]. The main advantages of RANSAC are that the randomization procedure employed can severely reduce the computational cost of an otherwise combinatorial search.

Motivated by applications in computational neuroscience such as matching the neuronal populations of *Caenorhabditis elegans* (*C. elegans*) across different nematodes, we aim to unify the ideas presented in RWOC literature and robust point set registration methods to provide provably approximate solutions to the RWOC problem in the presence of outliers and missing measurements commonly encountered in fluorescence microscopy data. Robustly and automatically matching and identifying neurons in *C. elegans* could expedite the post-experimental data analysis and hypothesis testing cycle [Bubnis et al., 2019; Kainmueller et al., 2014; Nguyen et al., 2017b; Yemini et al., 2019b].

2.2.0.1 Main contributions

The main contributions presented in this paper are the introduction of randomized algorithms for the recovery of the regression coefficients in the RWOC problem that takes into account noise, missing data, and outliers. Hsu et al. [Hsu et al., 2017] provide algorithms for the noisy case without generative assumptions; their algorithm takes into account square permutation matrices, which assumes that the entire signal is captured in the responses and does not take into account any missing correspondences or outliers.

Unnikrishnan et al. [Unnikrishnan et al., 2015, 2018] provide combinatorial existence arguments. Tsakiris et al. [Tsakiris and Peng, 2019] provide an algorithm that takes into account missing correspondences or outliers but not both. Our method is designed for the practical purpose of matching point clouds that may have noisy measurements, missing correspondences, and outliers. Missing data can be thought of as outliers in the source point set, but they can have different interpretations. For example, if the goal is to register an image onto an already existing atlas, then the parts of the atlas that are not present in the image are called missing data. The assumption is that the atlas contains a complete set of objects while the image could be missing some parts for reasons such as incomplete field of view, mutant defects, individual differences, etc. This is undoubtedly the case in the application domain of neuron tracking and matching in biological applications where structures of interest might be missing from the field of view or other unrelated confounding biological structures might exist and potentially be captured by the detection algorithms. Specifically, we demonstrate the efficacy of the proposed method in the identification and tracking of in-vivo (*C. elegans*) neurons where it is possible that some neurons are missing and adversarial objects that might be confused as neurons are present.

In summary, our contributions are four-fold:

1. We introduce the notion of *"robust" regression without correspondence* (rRWOC) that models missing correspondences between responses and covariates as well as completely missed associations in the form of outliers and missing data. In contrast with standard point set registration methods, we further consider the case of adversarial outliers.

2. We introduce a polynomial-time algorithm to find the exact solution for the one-dimensional noiseless rRWOC and the approximate solution in the noisy regime.

3. We introduce a randomized approximately correct algorithm that is more efficient than pure-brute force approaches in multiple dimensional rRWOC.
4. We demonstrate the computational neuroscience application of our approach to point-set registration problems in the context of automatically matching and identification of the cellular layout of the nervous system of the nematode *C. elegans*.

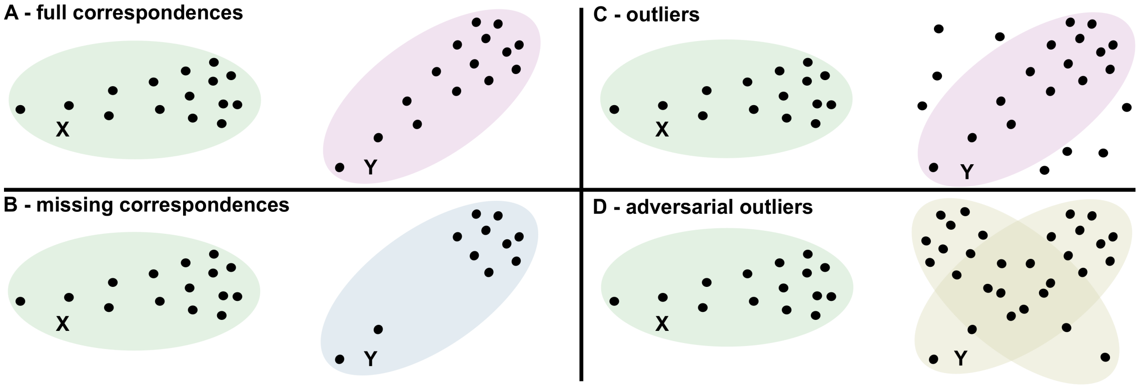

Figure 2.2: **Demonstration of various problem settings of regression without correspondence** A: Full set of hidden correspondences between source and target multisets. **B:** Missing correspondences in the target set. **C:** Unstructured outliers in the target set. **D:** Adversarial outliers in the target set – this setting imposes a theoretical ceiling of 50% outliers in the target set. However, in practice, more than 50% ratio of unstructured outliers can be handled.

2.2.1 Regression model

First, we introduce notation. Let $\boldsymbol{X} = [\boldsymbol{x}_1|\boldsymbol{x}_2|\dots|\boldsymbol{x}_m]^T \in \mathbb{R}^{m\times d}$ and $\boldsymbol{Y} = [\boldsymbol{y}_1|\boldsymbol{y}_2|\dots|\boldsymbol{y}_n]^T \in \mathbb{R}^{n\times d}$ denote two d-dimensional point sets consisting of m and n points, respectively. Let us call $\boldsymbol{X}$ the reference or source set. Let $\boldsymbol{Y}$ denote the target set which may contain outliers and missing correspondences. Note that the points in $\boldsymbol{X}$ that are missing correspondences in $\boldsymbol{Y}$ can be seen as outliers in the source set, hence justifying our claim that we model outliers in both the source and target sets.

Let the set of indices $\mathscr{I} = \{i_1, \ldots, i_{|\mathscr{I}|}\} \subseteq [n]$ denote the indices of $\boldsymbol{y}_j$ which are inliers. Conversely, let $\mathscr{O} = \{o_1, \ldots, o_{|\mathscr{O}|}\} \subseteq [n]$ denote set of indices of $\boldsymbol{y}_j$ which are outliers. By construction, these sets are a disjoint partition of the entire index set of target points: $\mathscr{I} \bigcup \mathscr{O} = [n]$ and $\mathscr{I} \bigcap \mathscr{O} = \emptyset$. Let $\boldsymbol{P} \in \mathscr{P}^{n \times m}$ denote a possibly unbalanced permutation matrix where there are at most $\min\{n, m\}$ ones placed such that no row or column has more than a single one. All other entries are zeroes. Let $\pi(i)$ denote the location of the one in the ith row of the permutation matrix $\boldsymbol{P}$. Next, let $\boldsymbol{\beta} \in \mathbb{R}^{d \times d}$ denote the regression coefficients and $\epsilon \sim \mathscr{N}(0, \nu \boldsymbol{I})$ denote zero-mean Gaussian noise. Lastly, let $\mathrm{U}[\mathscr{C}]$ denote the uniform distribution within some closed convex set $\mathscr{C}$. Given these definitions, we can define the **robust regression without correspondence** (rRWOC) model as

$$\begin{aligned} \boldsymbol{y}_{i_j} &= \boldsymbol{x}_{\pi(i_j)} \boldsymbol{\beta} + \epsilon && \text{for } i_j \in \mathscr{I} \\ \boldsymbol{y}_{o_l} &\sim \mathrm{U}[\mathscr{C}] && \text{for } o_l \in \mathscr{O} \end{aligned} \tag{2.1}$$

Note that the bias terms in the regression can be modeled by padding $\boldsymbol{x}$ with a constant column of ones.

In contrast with linear regression, where the sole objective is to recover the coefficients $\boldsymbol{\beta}$, the two-fold objective of RWOC is to recover the correct permutation matrix $\boldsymbol{P}$, and the regression coefficients $\boldsymbol{\beta}$. To add to the complexity of the problem, the three-fold objective of rRWOC is to recover the inlier set $\mathscr{I}$, the permutation $\boldsymbol{P}$, and the coefficients $\boldsymbol{\beta}$.

2.2.2 Algorithms

To aid in the recovery of the solution in rRWOC, we introduce the following assumption.

Assumption 1 (Maximal inlier set)**.** For point sets $\boldsymbol{X}, \boldsymbol{Y}$, there exists a triple $\{\mathscr{I}^*, \boldsymbol{\beta}^*, \boldsymbol{P}^*\}$ that is maximal in the sense that $n \geq |\mathscr{I}^*| \geq |\mathscr{I}'|$ such that any other triple $\{\mathscr{I}', \boldsymbol{\beta}', \boldsymbol{P}'\}$ is not considered to be the underlying regression model.

Assumption 1 allows the identifiability of whether a given hypothetical index set can be considered to be the true underlying inlier set or not. In practical terms, suppose we generate simulated data with n points in $\boldsymbol{Y}$ of which $k > n/2$ are outliers generated uniformly and the remainder generated with respect to a coefficient $\boldsymbol{\beta}^{\mathscr{I}}$ such that $\boldsymbol{Y}_{[\mathscr{I}]} = \boldsymbol{X}_{\pi(\mathscr{I})}\boldsymbol{\beta}^{\mathscr{I}} + \epsilon^{\mathscr{I}}$. There may be cases such that uniformly generated "outliers", $\boldsymbol{Y}_{[\mathscr{O}]}$, are structured such that there exists a coefficient $\boldsymbol{\beta}^{\mathscr{O}}$ and permutation $\boldsymbol{P}^{\mathscr{O}}$ such that $\boldsymbol{Y}_{[\mathscr{O}]} = \boldsymbol{X}_{\pi(\mathscr{O})}\boldsymbol{\beta}^{\mathscr{O}} + \epsilon^{\mathscr{O}}$ where $\mathrm{Var}(\epsilon^{\mathscr{I}}) \geq \mathrm{Var}(\epsilon^{\mathscr{O}})$. In this case, $\boldsymbol{\beta}^{\mathscr{O}}$ is identifiable but not verifiable as "correct." In practical terms, assumption 1 puts a ceiling on the maximum proportion of outliers that any regression without correspondence algorithm can handle. In a simplest example, if the target point set consists of two duplicate copies of rotated and transformed source point set, it is impossible to identify the correct matching. However, if one of the duplicates has less points, then we can invoke the principle of the maximal inlier set to identify the correct target set. See figure 2.2 for a visualization.

Equipped with the rRWOC model and assumption 1, we now demonstrate the progressive increase in the complexity of recovery of ordinary linear regression, RWOC, and rRWOC in one-dimension.

2.2.2.1 Optimal regression in $d = 1$

Linear regression in one-dimension with known correspondences, no offset term and no outliers can be obtained in $O(n)$ time using the univariate normal equation: $\beta_{OLS} = \frac{\sum_i^n y_i x_{\pi(i)}}{\sum_i^n x_{\pi i}^2}$. On the other hand, RWOC in the one-dimensional case with *no noise* can be solved in $O(n\log(n))$ steps via the method of moments and a simple sorting operation. Namely, first, the regressor β_{RWOC} can be estimated using the ratio of the first moments

Algorithm 1 One dimensional robust regression without correspondence - Exhaustive approach

Input:Reference set: $\{x_1, \dots, x_m\}$, target set: $\{y_1, \dots, y_n\}$, outlier margin: v
Require: $k < \frac{n}{2}$ (number of outliers)

```
for $i = 1, \dots, n$ do
    for $j = 1, \dots, m$ do
        Compute $\beta^{i,j} = y_i / x_j$
        Compute linear assignment [Kuhn, 1955]:
        $\boldsymbol{P}^{i,j} \leftarrow \underset{\boldsymbol{P} \in \mathcal{P}^{n \times m}}{\arg\min} \| \boldsymbol{x}\beta^{i,j} - \boldsymbol{P}^T \boldsymbol{y} \|_2^2$
        Compute hypothetical inliers:
        $\mathcal{I}^{i,j} = \{l : |x_{\pi^{i,j}(l)} \beta^{i,j} - y_l| \leq v\}$
    end for
end for
return $(i^*, j^*) = \underset{(i,j)}{\arg\max} |\mathcal{I}^{i,j}|$ , $\mathcal{I}^* = \mathcal{I}^{i^*,j^*}$, $\boldsymbol{P}^* = \boldsymbol{P}^{i^*,j^*}$, $\beta^* \leftarrow \frac{\sum_{l \in \mathcal{I}^*} y_l x_{\pi^*(l)}}{\sum_{l \in \mathcal{I}^*} x_{\pi^*(l)}^2}$
```

[1]

of the covariates to the responses:

$$\beta_{RWOC} = \frac{\sum_{i=1}^{n} y_i}{\sum_{i=1}^{n} x_i} \tag{2.2}$$

and then the permutation can be recovered using the re-arrangement inequality [Beckenbach and Bellman, 2012],

$$\min_{\boldsymbol{P}} \sum_{i=1}^{n} (y_i - \hat{y}_{\pi(i)})^2 = \sum_{i=1}^{n} (y_{(i)} - \hat{y}_{(i)})^2 = \tag{2.3}$$

$$\|\boldsymbol{P}_y \boldsymbol{y} - \boldsymbol{P}_{\hat{y}} \hat{\boldsymbol{y}}\|_2^2 \longrightarrow \boldsymbol{P}_{RWOC} = \boldsymbol{P}_y^T \boldsymbol{P}_{\hat{y}}$$

where $y_{(i)}$ denotes sorted y_i and $\hat{y}_{(i)}$ denotes sorted $x_i \beta_{RWOC}$ and $\boldsymbol{P}_y$ and $\boldsymbol{P}_{\hat{y}}$ denote the permutation matrices that capture the sorting operations.

In the case with outlier elements in $\boldsymbol{y}$, the problem is non-trivial, even in one dimension, since sorting does not allow the identification of outliers[1]. To solve the one dimensional rRWOC, we introduce algorithm 1 which recovers the triplet $\{\mathcal{I}^*, \boldsymbol{\beta}^*, \boldsymbol{P}^*\}$ in an exhaustive fashion.

[1] See supplementary material section 5 for a toy example experiment.

Proposition 1 (Correctness of Algorithm 1)**.** Suppose there exist $n-k$ inliers in $\boldsymbol{y}$ and that $k < n/2$. Then algorithm 1 yields the correct regression coefficient $\beta^* = \beta$ with probability 1 for noiseless data and with high probability for noisy data with an appropriately selected margin parameter ν.

Proof. (The full proof is included in supplementary material) The overview of the proof is as follows. In the noiseless case, if $j = \pi(i)$ then $\beta^{i,j} = \frac{y_i}{x_j} = \beta^*$. The projection $\boldsymbol{x}\beta^{i,j}$ maps all reference points to their exact corresponding reference points. Thus the Hungarian algorithm will yield these as the assignments since they incur minimal cost. Therefore, we will have $|\mathscr{I}^{i,j}| \geq n-k$. The cardinality of inliers is lower bounded and not equal to $n-k$ since outlier points may by chance be transformed to points in $\boldsymbol{y}$ as well. Contrarily, suppose the transformation $\beta^{i,l}$ for $l \neq \pi(i)$ yields a larger hypothesized inlier set $\mathscr{I}^{i,l}$, such that $|\mathscr{I}^{i,l}| > |\mathscr{I}^{i,j}|$ then this means that there are more points in $\boldsymbol{x}\beta^{i,l}$ that are closer to $\boldsymbol{y}$ than $\boldsymbol{x}\beta^{i,j}$, contradicting the assumption that $n-k$ is the maximal inlier set. ■

The time complexity of algorithm 1 can be analyzed as follows. The main computational cost is due to linear assignment which incurs a cost of $O(\max\{m,n\}^3)$ if [Jonker and Volgenant, 1986] variant is used. Linear assignment is repeated mn times. If m and n are of the same order, then algorithm 1 has complexity $O(n^5)$.

However, if the ratio of inliers to outliers is relatively high, then it is possible to use randomization procedures like RANSAC [Fischler and Bolles, 1981; Torr and Zisserman, 2000] to speed up the algorithm to yield the correct regression coefficient with high probability. This is demonstrated in algorithm 2.

Proposition 2 (Correctness of Algorithm 2)**.** Suppose there are $n-k$ inliers in $\boldsymbol{x}$ and that $k < n/2$. In $q \geq \frac{\log(1-\delta)}{\log(1-\frac{n-k}{mn})}$ iterations, algorithm 2 yields the correct regression coefficient

Algorithm 2 One dimensional robust regression without correspondence - Randomized approach

Input:Reference set: $\{x_1,\dots,x_m\}$, target set: $\{y_1,\dots,y_n\}$, δ (probability of success), outlier margin: ν

Require: $k < \frac{n}{2}$ (number of outliers)

```
for $t = 1,\dots,q$ do
    Sample $i \sim [n]$ and sample $j \sim [m]$
    Compute $\beta^t = y_i/x_j$
    Compute linear assignment [Kuhn, 1955]:
    $\boldsymbol{P}^t \leftarrow \underset{\boldsymbol{P}\in\mathscr{P}^{n\times m}}{\arg\min} \|\boldsymbol{x}\beta^t - \boldsymbol{P}^T\boldsymbol{y}\|_2^2$
    Compute hypothetical inliers:
    $\mathscr{I}^t = \{l : |x_{\pi^t(l)}\beta^t - y_l| \le \nu\}$
end for
return $t^* = \underset{t}{\arg\max}|\mathscr{I}^t|$ , $\mathscr{I}^* = \mathscr{I}^{t^*}$,
    $\boldsymbol{P}^* = \boldsymbol{P}^{t^*}$, $\beta^* \leftarrow \frac{\sum_{l\in\mathscr{I}^*} y_l x_{\pi^*(l)}}{\sum_{l\in\mathscr{I}^*} x_{\pi^*(l)}2}$
```

$\beta^* = \beta$ with probability $\delta \in (0,1)$ for an appropriately selected margin parameter ν.

Proof. The success of algorithm 1 relies on the fact that the exhaustive search eventually hits a tuple (i,j) such that $j = \pi(i)$ which yields the correct regression coefficient. Therefore, when randomly sampling $(i,j) \sim [n]\times[m]$, the probability of choosing a corresponding pair is $\frac{n-k}{n}\frac{1}{m}$. The probability of iterating q times such hat no correct correspondence is selected is $(1-(n-k)/(nm))^q = (1-\delta)$ where δ is the desired success rate. Taking logs yields, $q = \frac{\log(1-\delta)}{\log(1-(n-k)/(nm))}$ ■

The time complexity of randomized algorithm 2 is $O\left(\frac{\log(1-\delta)}{\log(1-(n-k)/n^2)}n^3\right)$.

2.2.2.2 Randomized approximation algorithm ($d \ge 2$)

The exhaustive approach for the $d \ge 2$ dimensional case requires $\binom{n}{d}\binom{m}{d}$ d-subset comparisons of $\boldsymbol{X},\boldsymbol{Y}$ in order to guarantee hitting correct (in the noiseless case) or approximately correct (in the noisy case) regression coefficients, with complexity $O(m^d n^d)$.

Algorithm 3 Robust regression without correspondence - Randomized approach

Input: $\boldsymbol{X} = [\boldsymbol{x}_1|\dots|\boldsymbol{x}_m]^T \in \mathbb{R}^{m\times d}$ (reference points), $\boldsymbol{Y} = [\boldsymbol{y}_1|\dots|\boldsymbol{y}_n]^T \in \mathbb{R}^{n\times d}$ (target points), δ (probability of success), ν (outlier margin)

Require: $k < \frac{n}{2}$ (number of outliers)

```
for $t = 1,\dots,q$ do
    Sample $\boldsymbol{i} = (i_1,\dots,i_d) \sim [n]^d$ w/o replacement
    Sample $\boldsymbol{j} = (j_1,\dots,j_d) \sim [m]^d$ w/o replacement
    Compute $\boldsymbol{\beta}^t = \arg\min_{\boldsymbol{\beta}} \|\boldsymbol{X}_{[\boldsymbol{j}]}\boldsymbol{\beta} - \boldsymbol{Y}_{[\boldsymbol{i}]}\|_F^2$
    Compute linear assignment via [Kuhn, 1955]:
    $\boldsymbol{P}^t \leftarrow \arg\min_{\boldsymbol{P}\in\mathscr{P}^{m\times n}} \|\boldsymbol{X}\boldsymbol{\beta}^t - \boldsymbol{P}\boldsymbol{Y}\|_F^2$
    Compute hypothetical inliers:
    $\mathscr{I}^t = \{l : \|\boldsymbol{x}_{\pi^t(l)}\boldsymbol{\beta}^t - \boldsymbol{y}_l\|_2 \le \nu\}$
end for
return $t^* = \arg\max_t |\mathscr{I}^t|$, $\mathscr{I}^* = \mathscr{I}^{t^*}$,
    $\boldsymbol{P}^* = \boldsymbol{P}^{t^*}_{\mathscr{I}^*}, \boldsymbol{\beta}^* \leftarrow \arg\min_{\boldsymbol{\beta}} \|\boldsymbol{X}_{\pi^*(\mathscr{I}^*)}\boldsymbol{\beta} - \boldsymbol{Y}_{\mathscr{I}^*}\|_F^2$
```

However, especially in higher dimensions, the randomized procedure enables a substantial reduction of iterations to yield a high probability correct triplet of inlier set, permutation, and regression coefficients. The randomized algorithm for rRWOC in $d \ge 2$ is demonstrated in algorithm 3. Random ordered d-tuples of reference and target point sets are sampled and are used to align the remainder of the point set. The number of hypothetical inliers for each hypothetical correspondence is assessed by checking whether the transformed reference points are arbitrarily close to a target point. With high probability, if correct a d-tuple correspondence is captured, the number of transformed reference points matching a target point will be high (Figure 2.2 top), otherwise it will result in a partial coverage (Figure 2.2 bottom).

Proposition 3. For $q \ge \frac{\log(1-\delta)}{\log\left(1-\frac{\binom{m-k}{d}}{\binom{m}{d}\binom{n}{d}}\right)}$, algorithm 3 recovers $\boldsymbol{\beta}^*$ and $\boldsymbol{P}^*$ and the set of inliers for the noiseless case with probability $(1-\delta)$ using arbibrarily small ν. For sufficiently small noise variance and appropriately chosen ν, algorithm 3 recovers approximate $\boldsymbol{\beta}^*$ with high probability.

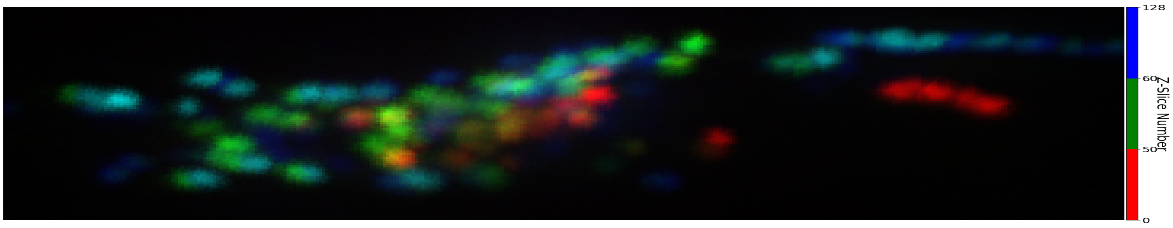

Figure 2.3: **2D projection of 3D fluorescence microscopy image of *C. elegans* head in [Yemini et al., 2021] dataset** Superimposed annotation points denote neuron locations. Outliers are detections that do not correspond to neurons and missing data are undetected neurons.

Proof. Analogous to the analysis of algorithm 2, the probability of drawing d inliers out of n points with k outliers in $\boldsymbol{Y}$ is $\frac{\binom{n-k}{d}}{\binom{n}{d}}$. The probability of matching the drawn inliers with the d corresponding sampled reference points in $\boldsymbol{X}$ is $\frac{1}{\binom{m}{d}}$. Probability that any draw is not going to match is $1 - \frac{\binom{n-k}{d}}{\binom{m}{d}\binom{n}{d}}$. The probability that q draws will be incorrect is $\left(1 - \frac{\binom{m-k}{d}}{\binom{m}{d}\binom{n}{d}}\right)^q$. If we set this to be the probability of failure $(1-\delta)$, we then have the estimate for the number of draws we need to make as $q(\delta, n, m, k) \geq \log(1-\delta)/\log\left(1 - \frac{\binom{m-k}{d}}{\binom{m}{d}\binom{n}{d}}\right)$ ■

The complexity of algorithm 3 can be analyzed as follows. In each inner loop, the regression coefficient solution requires $O(d^3)$ time, the Hungarian algorithm requires $O(nmd)$ to compute the input distance matrix and then $O(\max\{n,m\}^3)$ to optimize the permutation matrix. The rest of the operations are $O(d)$. Therefore, the overall time complexity is

$$O\left(\frac{\log(1-\delta)}{\log\left(1 - \frac{\binom{m-k}{d}}{\binom{m}{d}\binom{n}{d}}\right)}(d^3 + nmd + \max\{n,m\}^3)\right). \tag{2.4}$$

2.2.2.3 Margin parameter (ν) selection

Both of the proofs of the noiseless and the noisy cases of proposition 1 rely on knowledge of the true regression coefficient and the noise variance in order to estimate the margin coefficient ν and output the optimal regression coefficient with high probability. However, in practice, as in many RANSAC-like robust regression settings, these parameters cannot be known apriori, and ν is typically determined via empirical heuristics and or cross-validation [Fischler and Bolles, 1981].

In the noiseless case, an appropriate heuristic is choosing ν arbitrarily small since the correct regression should yield zero residual. However, for the noisy case, if available, supervised data should be used with known correspondences to estimate the actual dispersion of point correspondences.

2.2.3 Numerical Results

To verify the theoretical guarantees of the proposed algorithms, simulated data in 3 dimensions was generated in both noisy and noiseless regimes. Furthermore, iterative solutions of $\boldsymbol{\beta}$ and $\boldsymbol{P}$ were obtained to demonstrate the suboptimality of local minima found using block coordinate descent for this non-convex problem.

The neuroscience application of rRWOC was demonstrated in the context of point set matching of neurons of *C. elegans* worms recorded using fluorescence microscopy imaging. The matching accuracy with respect to ground truth was assessed for rRWOC as well as a robust variant of the iterative closest point (ICP) algorithm [Besl and McKay, 1992] known as trimmed ICP [Chetverikov et al., 2002]. We also compared to the state of the art algorithm for regression without correspondence, termed homomorphic sensing (HS) [Tsakiris and Peng, 2019].

Computational setup and code: All experiments were performed on an Intel i5-7500 CPU at 3.40GHz with 32GB RAM. MATLAB code for 3D versions of algorithm 3 are included in supplementary material along with sample *C. elegans* neuron point clouds.

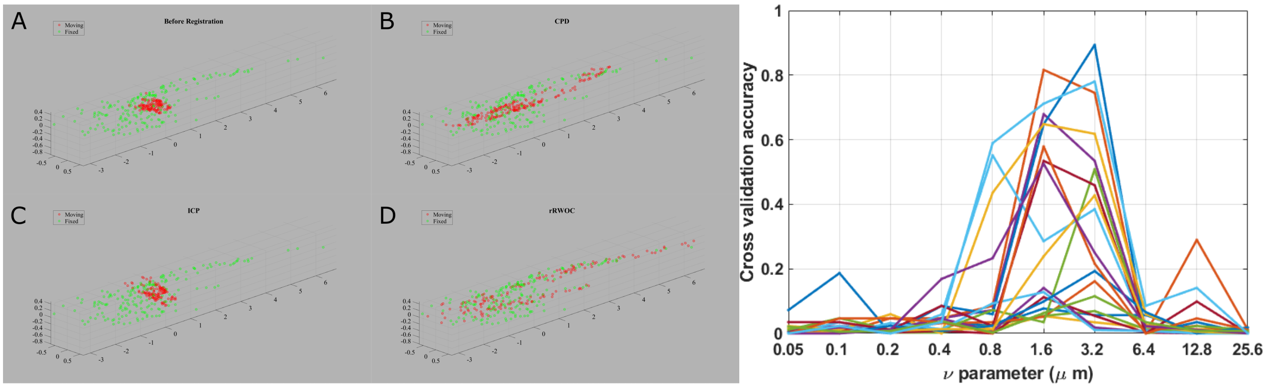

Figure 2.4: **Comparison of rRWOC with other methods and hyperparameter sensitivity results** Left: A: Unaligned point sets of reference *C. elegans* neuron positions (red) and target neuron positions (green) B: Alignment with coherent point drift algorithm [Myronenko and Song, 2010], C: Alignment with iterative closest point algorithm [Chetverikov et al., 2002], D: Alignment with proposed algorithm 3. Right: Margin parameter (ν) estimation in the *C.elegans* dataset.

	Method	*TP*	*FP*	*FN*	*ACC*	*F1*	*PREC*	*REC*	*MD*
C. elegans Head	rRWOC	**135±28**	**57±23**	**60±28**	**0.53±0.15**	**0.69±0.13**	**0.70±0.12**	**0.69±0.14**	**2.63±0.27**
	ICP [Chetverikov et al., 2002]	41±58	151±58	153±59	0.15±0.23	0.21±0.30	0.21±0.30	0.21±0.30	4.18±1.59
	CPD [Myronenko and Song, 2010]	5±2	188±4	190±2	0.01±0.01	0.03±0.01	0.03±0.01	0.03±0.01	11.13±0.34
	HS [Tsakiris and Peng, 2019]	110±24	70±23	80±23	0.45±0.15	0.60±0.13	0.50±0.09	0.53±0.11	3.2±0.34
C. elegans Tail	rRWOC	33±6	10±6	11±6	0.61±0.17	0.75±0.14	**0.76±0.14**	0.74±0.13	2.14±0.31
	ICP [Chetverikov et al., 2002]	2±1	42±1	43±1	0.02±0.01	0.04±0.02	0.04±0.02	0.04±0.02	7.83±1.66
	CPD [Myronenko and Song, 2010]	3±1	41±1	42±1	0.03±0.02	0.04±0.02	0.04±0.02	0.04±0.02	7.43±1.32
	HS [Tsakiris and Peng, 2019]	**36±4**	**9±4**	**10±5**	**0.65±0.13**	**0.78±0.12**	0.72±0.13	**0.82±0.12**	**1.9±0.32**
Fish Unstructured	rRWOC	**28 ± 13**	**18 ± 13**	**27 ± 13**	**0.42 ± 0.21**	**0.61 ± 0.29**	**0.67 ± 0.31**	**0.56 ± 0.26**	**0.12 ± 0.01**
	ICP [Chetverikov et al., 2002]	2 ± 1	45 ± 1	54 ± 1	0.02 ± 0.01	0.04 ± 0.02	0.05 ± 0.02	0.04 ± 0.02	0.23 ± 0.05
	CPD [Myronenko and Song, 2010]	1 ± 2	46 ± 2	55 ± 2	0.01 ± 0.03	0.02 ± 0.06	0.02 ± 0.06	0.02 ± 0.05	0.12 ± 0.00
	HS [Tsakiris and Peng, 2019]	14 ± 0	33 ± 0	42 ± 0	0.17 ± 0.01	0.30 ± 0.02	0.33 ± 0.02	0.28 ± 0.01	0.22 ± 0.00
Fish Adversarial	rRWOC	**28 ± 13**	**18 ± 13**	**26 ± 13**	**0.43 ± 0.28**	**0.62 ± 0.28**	**0.67 ± 0.30**	**0.57 ± 0.26**	**0.14 ± 0.07**
	ICP [Chetverikov et al., 2002]	0 ± 18	47 ± 18	55 ± 18	0 ± 0.34	0 ± 0.40	0 ± 0.43	0 ± 0.37	0.30 ± 0.05
	CPD [Myronenko and Song, 2010]	0 ± 0	47 ± 0	55 ± 0	0 ± 0.00	0 ± 0.01	0 ± 0.01	0 ± 0.01	0.10 ± 0.02
	HS [Tsakiris and Peng, 2019]	15 ± 8	32 ± 8	40 ± 8.5129	0.19 ± 0.11	0.33 ± 0.18	0.36 ± 0.19	0.31 ± 0.17	0.19 ± 0.11

Table 2.1: Transformation recovery and permutation recovery by rRWOC, ICP, CPD and HS algorithms in the C. elegans and `fish` dataset. TP = true positive, FP = false positive, TN = true negative, FN = false negative, ACC = accuracy, F1 = F1 score , PREC = precision, REC = recall, MD = mean distance

2.2.3.1 Neuron matching of *C. elegans*

For this application, we have used the publicly available *C. elegans* fluorescence imaging dataset of Nguyen et al. [Nguyen et al., 2017b] found at `http://dx.doi.org/10.21227/H2901H` as well as the neuronal position dataset provided in [Yemini et al., 2019b]. The worm *C. elegans* is a widely known model organism for studying the nervous system due to the known structural connectome of the 302 neurons it contains. The data provided 3D z-stack images of the head of 14 worms that each consists of approximately 185 to 200 neurons captured under confocal microscopy using florescent tagged protein GFP. In figure 2.3, the depth-colored 2D projection of an image frame can be seen superimposed with annotation points delineating the locations of neurons. Figure 2.3 also highlights the need for a method of matching and aligning worm point clouds that is robust to outliers or missing associations. Here, we define outliers as points where there is no neuron present and define missing data as neurons with no detection present.

Of the 14 datasets of the head neurons of *C.elegans* worms, random pairs were drawn to be the source and target point sets. From the remaining worms, the positional covariance of each neuron was estimated using the supervised alignment method of [Evangelidis and Horaud, 2018]. Since the positional variance of each neuron was uniquely identified using training data, we used variable margin parameters for rRWOC such that $v_l = \max_{i=1,2,3} \lambda_i(\Sigma_l)$ where Σ_l is the covariance matrix of the lth neuron and $\lambda_i(\cdot)$ denotes the ith eigenvalue. Randomized RWOC (algorithm 3) was deployed with $\delta = 0.9$. The results were compared with iterative closest point(ICP) [Besl and McKay, 1992] as well as coherent point drift (CPD) [Myronenko and Song, 2010] algorithms.

The demonstration of the *C. elegans* application of rRWOC is seen in figure 2.4. Here the source point set is a statistical atlas neuron positions [Yemini et al., 2019b] and the target point set is neuron detections which may be corrupted by non-neuronal outliers. The outcome is that the detected neurons are identified correctly using the proposed algorithm.

The recovery rates in terms of recovering the transformation $\boldsymbol{\beta}^*$ as well as the permutation $\boldsymbol{P}^*$, are summarized in table 2.1. In general, rRWOC was able to recover both the transformation and permutation better than ICP and CPD, which tend to be initialization-dependent as well as HS which is a global method. In all of the experiments, ICP and CPD were initialized with random rotation. rRWOC and HS are invariant to initialization since they are not descent-based methods. HS performs slightly better in the tail of C.elegans than rRWOC since the tail dataset tends to have fewer outliers which HS is more sensitive to. Contrarily, rRWOC does better than HS in the head since there are more outliers and missing correspondences.

2.3 Joint Segmentation and Labeling of C. elegans Neurons

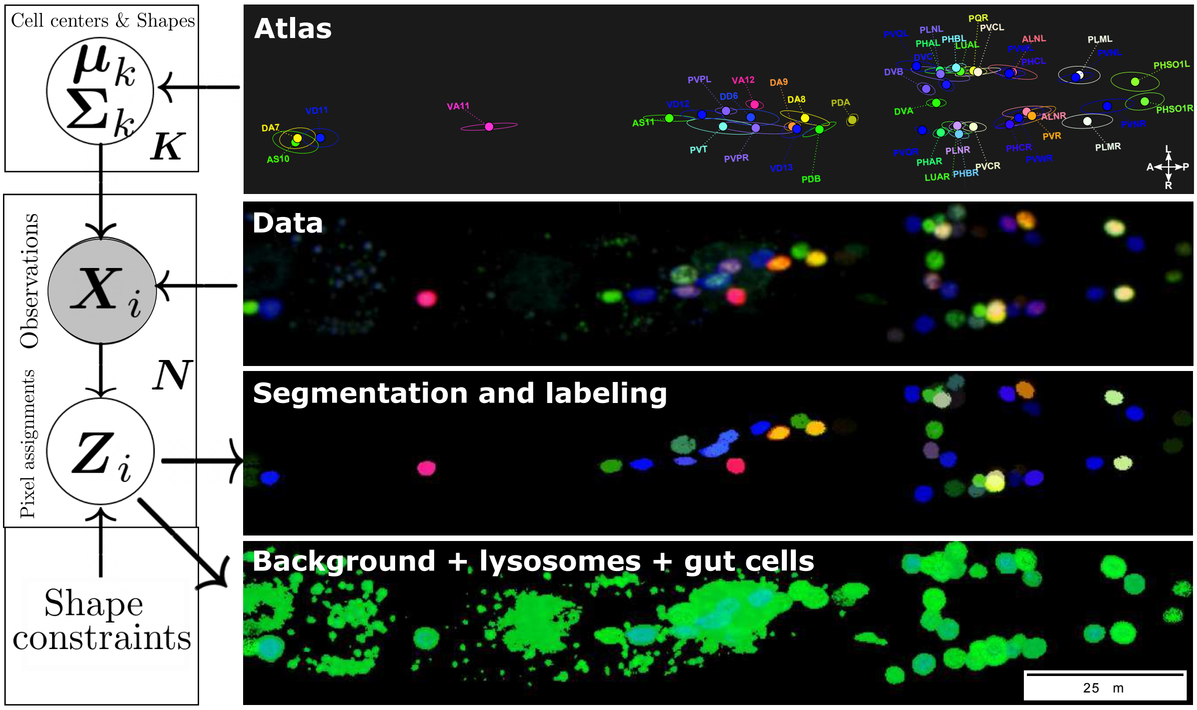

Figure 2.5: **Segmentation and labeling of fluorescently-colored neurons, in image volumes of NeuroPAL worms, using the proposed method** Left: The graphical model of the probabilistic inference procedure employed to identify and segment neurons. Right: The model uses the atlas from [Yemini et al., 2019a] as a prior to assign latent neuron identities to each observed pixel, subject to constraints on the total mass assigned to each cell.

Whole-brain functional imaging of *Caenorhabditis elegans* has been recently introduced to enable the measurement of neural activity at unprecedented temporal and spatial resolution [Schrödel et al., 2013]. Obtaining a complete measurement of neuron positioning and activity enables the study of a wide range of hypotheses including the identification of brainwide dynamic networks involved in action sequences and decisions, decoding of nervous system responses to repulsive and attractive stimuli from distinct modalities, and monitoring of neural identity to reveal neural-fate alterations in the presence of gene mutations [Yemini et al., 2019a]. However, a significant analysis bottleneck is the segmentation and identification of all imaged neurons. Automated segmentation and identification of *C. elegans* neurons would enable high-throughput experiments for many applications.

There have been many recent works towards segmenting and labeling cells and in particular, neurons. Several methods address the cell labeling problem directly without segmenting their shapes [Aerni et al., 2013; Hirose et al., 2018; Tokunaga et al., 2014; Toyoshima et al., 2019]. Broadly, current algorithms for cell labeling can be categorized into two classes: the first and more common approach is to split the labeling problem into multiple steps. These steps include a filtering step to eliminate background components and non-cellular objects. The second step is to detect potential locations of the cells. This step is prone to the false detection of non-cellular objects, depending on how accurate the filtering and detection steps are. The final step to label the cells is to run a point matching algorithm to find the correspondences between the cells and their features in an atlas and the set of detected cells. A very recent work in this class of models [Chaudhary et al., 2020] trains an atlas of the pairwise distances between the neurons, and finds the correspondence between the points such that the pairwise distances match the atlas. The second approach for cell labeling is to directly model the pixels as the observations of a model with cell centers as unobserved variables. The goal in these models is to classify pixels and to infer the unobserved cell centers simultaneously. An example of this paradigm is demonstrated in [Qu et al., 2011], where the authors consider a subset of well separated and large non-neuronal cells for segmentation and annotation.

A novel transgenic strain of *C. elegans* called “NeuroPAL” (a Neuronal Polychromatic

Atlas of Landmarks) has introduced differential fluorescent coloring of neurons to resolve all unique neural identities [Yemini et al., 2019a]. This has enabled the construction of a complete statistical atlas of neuron positions and colors. Here, we present a novel statistical pipeline for joint segmentation and labeling of neural identities in NeuroPAL images. We formulate the segmentation problem as a posterior inference over the latent variables of a mixture model and show that the neural labeling arises naturally from our formulation. We further present a novel technique to constrain the posterior distribution in the expectation-maximization (EM) algorithm to enforce prior knowledge about cell sizes using the Sinkhorn-Knopp algorithm [Sinkhorn and Knopp, 1967].

Our experimental results illustrate that the resulting "Sinkhorn EM" (**sEM**) approach outperforms vanilla EM (**vEM**) both in terms of segmentation quality as well as neuron identification accuracy. We further show that we outperform the multi-step method for neural identification developed in [Yemini et al., 2019a].

2.3.1 Methods

2.3.1.1 Probabilistic Model

First we introduce notation. Let N represent the total number of pixel observations in our multi-colored volumetric image. Each observed pixel in the volume is in the form of a 4-D tensor where the first 3 dimensions are spatial coordinates x, y, z and the 4th dimension corresponds to the different color channels. To facilitate notation within our probabilistic model, we represent the set of pixels as N tuples $\boldsymbol{X}_i = (\boldsymbol{l}_i, \boldsymbol{c}_i) \in \mathbb{R}^{3+C}$ where C is the total number of color channels. The first part of this tuple, $\boldsymbol{l}_i = (x_i, y_i, z_i) \in \mathbb{R}^3$, corresponds to the location of pixel i. The next part of this tuple, $\boldsymbol{c}_i = (c_i^1, \ldots, c_i^C)$, is a vector indicating the color intensity of pixel i in all channels. Next, we model these pixel tuples as random variables drawn from a mixture model.

Let $\boldsymbol{\theta}$ denote the set of parameters of the mixture distribution that these observations are assumed to be drawn from. Since we are trying to segment images comprising of neurons as well as non-neuronal background components, we model this mixture in terms of K components that correspond to neurons that we are trying to segment and B components that capture the background. Given these N pixel observations, $\boldsymbol{X} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N\}$ and distribution parameters, $\boldsymbol{\theta}$, we model the data using the following joint probabilistic mixture model:

$$P(\boldsymbol{X},\boldsymbol{\theta}) = P(\boldsymbol{\theta}) \prod_{i=1}^{N} P(\boldsymbol{X}_i|\boldsymbol{\theta}) = P(\boldsymbol{\theta}) \prod_{i=1}^{N} \sum_{l=1}^{K+B} \pi_l P_l(\boldsymbol{X}_i|\boldsymbol{\theta}). \tag{2.5}$$

Here π_l denotes the membership weights for the lth component and $P_l(\boldsymbol{X}_i|\boldsymbol{\theta})$ denotes the likelihood of the ith pixel given the lth component. Next, we explicitly model the distributions of neurons and the background. We model pixel observations of neurons as multivariate normal distributed in both position and color, i.e. we expect to observe pixels corresponding to the neuron VA11 (the solo magenta neuron roughly one-third from the left-side of Figure 4.8) in the general vicinity of where neuron VA11 is positioned and in colors close to the stereotypical color of the neuron VA11. Furthermore, the *C. elegans* nuclei imaged here are roughly ellipsoidal, which make Gaussian modeling plausible. On the other hand, we model background components to be positioned uniformly throughout the volume but with multivariate-normal distributed colors; i.e., lysosomes (indicated by the green speckles in figure 4.8) could be positioned arbitrarily but usually are in a shade of green. The likelihood of this model can be expressed as:

$$\begin{gathered}\sum_{l=1}^{K+B} \pi_l P_l(\boldsymbol{X}_i|\boldsymbol{\theta}) = \\ \underbrace{\sum_{k=1}^{K} \pi_k^n \mathcal{N}((\boldsymbol{l}_i, \boldsymbol{c}_i)|\boldsymbol{\mu}_k^n, \boldsymbol{\Sigma}_k^n)}_{\textbf{Neurons}} + \underbrace{\sum_{j=1}^{B} \pi_j^b \mathcal{U}(\boldsymbol{l}_i|\boldsymbol{l}_{\min}, \boldsymbol{l}_{\max}) \mathcal{N}(\boldsymbol{c}_i|\boldsymbol{\mu}_j^b, \boldsymbol{\Sigma}_j^b)}_{\textbf{Background components}}.\end{gathered} \tag{2.6}$$

Here $\mathcal{N}(\cdot)$ is a multivariate normal distribution, and $\mathcal{U}(\cdot)$ is a multi-dimensional uniform distribution defined in a hyper-cube that ranges from $\boldsymbol{l}_{\min}$ to $\boldsymbol{l}_{\max}$ where $\boldsymbol{l}_{\min}$ denotes the lower bound of pixel coordinates and $\boldsymbol{l}_{\max}$ denotes the upper bound.

The multivariate Gaussian distributions to model neurons are parametrized by $\boldsymbol{\theta}^n = \{\boldsymbol{\beta}, \boldsymbol{\mu}^n_{1:K}, \boldsymbol{\Sigma}^n_{1:K}\}$ where $\boldsymbol{\beta} \in \mathbb{R}^{4\times 3}$ denotes an affine transformation of the observed neuron positions from their stereotypical position (encoded by an atlas for example). $\boldsymbol{\mu}^n_k \in \mathbb{R}^{3+C}$ denotes the sterotypical position and color of the kth neuron and $\boldsymbol{\Sigma}^n_k \in \mathbf{S}^{3+C}_{++}$ denotes its covariance. Background components are modeled similarly with respect to color, but are permitted to occupy any position in the volume. Namely, $\boldsymbol{\theta}^b = \{\boldsymbol{\mu}^b_{1:B}, \boldsymbol{\Sigma}^b_{1:B}\}$ where $\boldsymbol{\mu}^b_j \in \mathbb{R}^C$ and $\boldsymbol{\Sigma}^b_j \in \mathbf{S}^C_{++}$ denote the mean and covariance of the jth background component color.

The prior distribution of our model is a multivariate normal distribution that encodes the canonical locations and colors of the neurons aligned to the image using the affine transformation term $\boldsymbol{\beta}$. We use the atlas described in [Yemini et al., 2019a] and shown in Fig. 4.8:

$$P(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{1:K} | \boldsymbol{\beta}\boldsymbol{\mu}^a_{1:K}, \boldsymbol{\beta}\boldsymbol{\Sigma}^a_{1:K}\boldsymbol{\beta}^T), \tag{2.7}$$

where the super-script a denotes the atlas parameters, here the mean $\boldsymbol{\mu}^a$ and the covariance $\boldsymbol{\Sigma}^a$ of a Multivariate Normal distribution. Here we assume the existence of an affine transformation matrix that roughly aligns the atlas to the image. The $\boldsymbol{\beta}$ is fit using a few landmark cells and is updated further through iterations using the update rules discussed in the supplementary.

Given the set of N pixel observations $\boldsymbol{X} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N\}$, we seek to find the maximum a posteriori (MAP) estimate of the parameters given the observations. In other words, our objective is to maximize the following log-posterior:

$$\mathscr{L}(\boldsymbol{\theta}) = \log P(\boldsymbol{\theta}|\boldsymbol{X}) = \log P(\boldsymbol{\theta}) + \sum_{i=1}^{N} \log P(\boldsymbol{X}_i|\boldsymbol{\theta}) + C', \tag{2.8}$$

where C' is a constant with respect to $\boldsymbol{\theta}$ that can be ignored optimizing the cost with respect the parameters.

2.3.1.2 Optimization

Vanilla EM algorithm: To find the local MAP estimate of the model parameters, a common strategy is to introduce a latent assignment variable Z that assigns each observation to one of the mixture components. We then maximize the expected complete log likelihood where the expectation is taken under the posterior distribution of the assignment variable.

$$P(\boldsymbol{X},\boldsymbol{\theta}|Z=l) = P(\boldsymbol{X}|\boldsymbol{\theta}_l)P(\boldsymbol{\theta}) \tag{2.9}$$

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta_t}) = \mathbb{E}_{P(Z|\boldsymbol{X},\boldsymbol{\theta}_t)}[\log P(\boldsymbol{X},\boldsymbol{\theta},Z)] \tag{2.10}$$

Here $\boldsymbol{\theta}_t$ denotes the estimate of model parameters at the tth iterate. This function lower bounds $\mathscr{L}(\boldsymbol{\theta})$ and maximizing it improves $\mathscr{L}(\boldsymbol{\theta})$ in each iteration [Dempster et al., 1977], yielding the vanilla[2] Expectation-Maximization (**vEM**) algorithm:

$$\textbf{vEM:} \begin{cases} \textbf{E-step:} \quad \text{update } \boldsymbol{\gamma} \text{ by evaluating } Q(\boldsymbol{\theta}|\boldsymbol{\theta}_t) \\ \textbf{M-step:} \quad \text{solve } \boldsymbol{\theta}_{t+1} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}|\boldsymbol{\theta}_t) \end{cases} \tag{2.11}$$

The E-step consists of computing a term $\boldsymbol{\gamma}$ known as the **responsibility matrix** with $\gamma_{l,i} = P(Z_i = l|\boldsymbol{X}_i,\boldsymbol{\theta})$. For each pixel, this variable defines a probability space over the mixture components and provides a soft assignment of the pixels to components. For example, for a fixed row, $\gamma_{l,:}$ denotes the distribution of the lth component across space of pixels, roughly encoding the spatial extent and shape of the lth object. Conversely, the

[2] We add the term "vanilla" to disambiguate the standard EM meta-algorithm from the proposed variant described later in the text.

ith column of γ, $\gamma_{:,i}$ denotes the membership of the ith pixel amongst the k components. The analytical derivation of the EM parameter updates for the mixture model in (2.5) is included in the supplementary material.

Once we optimize the objective introduced in the previous subsection, the responsibility matrix and parameter estimates can then be used to drive the segmentation and neuron labeling, respectively. Namely, we can use responsibility matrix terms, $\gamma_{l,i}$, to infer whether the ith pixel is occupied by the kth neuron, or the bth background component. Additionally, we can infer the neuron positions and colors with the $\boldsymbol{\mu}_k^n$ estimates. Lastly, $\boldsymbol{\Sigma}_k^n$ can inform us about the neuron shapes and color variability.

Sinkhorn EM algorithm: By definition, the rows of the responsibility matrix, $\gamma_{l,:}$, must sum to one, in order to be a bonafide probability. In other words, the lth component must exist somewhere within the image. However, in **vEM** (2.11), the only way to control the row sum of this matrix is through the constraints on component proportions or distribution-specific component parameters (such as constraining covariance eigenvalues to stay within a range for the Gaussian distribution). Both of these types of constraints effectively act as regularization on the responsibility matrix, $\boldsymbol{\gamma}$. However, in practice, it is common that responsibilities for a component can collapse to zero through the mode collapse phenomenon [Archambeau et al., 2003]. This effectively prevents the segmentation of the lth object from the image, leading to false negatives.

In image segmentation, there often exists some information about the size of each component to be segmented. In our application of neuron segmentation, we have an estimation of how many pixels a neuron should occupy. To incorporate this information into the EM algorithm, we can explicitly constrain the row sum of the responsibility matrix, $\boldsymbol{\gamma}$, to match the desired number of pixels (or weights), while keeping the column sum normalized to one. Specifically, in each iteration of EM algorithm, we aim to find a matrix $\hat{\boldsymbol{\gamma}}$ that is close to γ while satisfying $\sum_l \gamma_{l,i} = 1$ and $\sum_i \gamma_{l,i} = \alpha_l$ where α_l encodes the proportion of pixels that the lth object must occupy. This procedure we describe has been explored by Sinkhorn and Knopp in [Sinkhorn and Knopp, 1967] in what is known as the *matrix balancing algorithm*. In each iteration of EM, We use Sinkhorn's algorithm, to efficiently approximate $\hat{\boldsymbol{\gamma}}$ by iteratively normalizing the row and column of $\boldsymbol{\gamma}$ matrix to sum to the pre-determined marginals α_l (Fig. 4.8). We term the resulting algorithm "Sinkhorn Expectation Maximization" (**sEM**) and study its empirical performance in comparison with **vEM** in the following subsections.

2.3.2 Results

More formally, **sEM** deviates from **vEM** in the evaluation of the responsibilities. Instead of evaluating the expectation of the complete log likelihood, we base our algorithm on the recent finding that the E-step can be modified to be cast as an entropic optimal transport problem. Mena et al. in [Mena et al., 2020] have shown that this modification of the E-step still yields a monotonic increase in the log-posterior function and enjoys better convergence properties. We perform the iterations:

$$\textbf{Sinkhorn EM:} \begin{cases} \textbf{E-step:} \quad \text{update } \hat{\boldsymbol{\gamma}} \text{ by solving:} \\[2ex] \qquad \hat{\boldsymbol{\gamma}} = \underset{\boldsymbol{\gamma} \in \Pi(\boldsymbol{\alpha}, \frac{\mathbf{1}}{N})}{\arg\min} \sum_{i,l} -\log P_l(X_i, \theta) \boldsymbol{\gamma}_{l,i} + \mathscr{H}(\boldsymbol{\gamma} | \boldsymbol{\alpha} \otimes \frac{\mathbf{1}}{N}) \\[2ex] \textbf{M-step:} \quad \text{solve } \boldsymbol{\theta}_{t+1} = \underset{\boldsymbol{\theta}}{\arg\max}\, Q(\boldsymbol{\theta} | \boldsymbol{\theta}_t) \end{cases} \tag{2.12}$$

Here $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{K+B})$ is a vector that encapsulates our desired component proportions, $\mathbf{1}$ denotes a vector of ones with length N, $\Pi(\boldsymbol{V}_1, \boldsymbol{V}_2)$ is the set of all matrices with marginals equal to vectors $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$, and $\mathscr{H}(\boldsymbol{\gamma}|\boldsymbol{A})$ is the relative entropy between probability measures $\boldsymbol{\gamma}, \boldsymbol{A}$ which in our case simplifies to $\sum_{l,i} \gamma_{l,i} \log \gamma_{l,i}$. As noted above, the E-step here can be solved by Sinkhorn iterations, the details of which can be found in the supplementary material.

Datasets: We ran the proposed algorithm on a *C. elegans* dataset consisting of images of 10 heads and 10 tails from the NeuroPAL strain. The images were captured using a spinning-disk confocal microscope with resolution (x,y,z)=(0.27,0.27,1.5) microns. There were three color channels encoding red, green, and blue fluorescence excitation. Each head and tail image roughly consists of about 190 and 40 neurons, respectively, which were annotated by an expert who determined their positions. Processing and annotation details are described in [Yemini et al., 2019a]. Note that the only ground truth available were point markers that denoted the neuron centers. Complete ground truth segmentation of the cell shapes was not provided.

Compared Methods: We compare our algorithm, **sEM**, with a method that is designed specifically toward neural identification in NeuroPAL strains of *C. elegans* [Yemini et al., 2019a]. The neuron identification algorithm in [Yemini et al., 2019a], termed **CELL-ID**, employs a 3 (or 2) step process of filtering, detection, and identification. We hereafter refer to detect+identify version of this method as **CELL-ID (2)** and the filter+detect+identify version as **CELL-ID(3)**. The filtering steps here involve heuristic methods and the detection system uses a greedy matching pursuit algorithm. Lastly, the neuron identification is done by a variant of the iterative closest point algorithm [Besl and McKay, 1992].

Additionally, we compare with the segmentation and labeling performance of **vEM** as described in equation 2.11. Due to a lack of ground truth, we evaluated the segmentation results qualitatively by visualizing the spread and sharpness of the segmentation maps.

Neuron Identification: For each worm image, we first spatially smoothed it using a small 3D Gaussian filter (width 0.5 μm in each dimension). We then removed the low-intensity background pixels using a small threshold to ensure that only dark background pixels are removed (70-th percentile).

Each of **sEM** and **vEM** outputs the cell centers, colors, and shapes (in terms of 3D covariance matrices) as well as a $\boldsymbol{\gamma}$ matrix that includes neuron-specific probabilistic segmentation maps. **CELL-ID** on the other hand only outputs the centers and identities of neurons. The cell centers are used to quantify the accuracy of **sEM** in comparison to **vEM** and **CELL-ID** (2)/(3). For each method, the accuracy is computed by counting the number of mixture components that are within a radius of 3 microns from their true location (annotated by an expert), dividing this by the total number of neurons. We detail the quantitative neuron identification results in Fig.2.7A-B. In Fig.2.7A, we show the neuron identification accuracy for the four compared methods. Due to the higher density of neurons in the head, the accuracies of all methods tended to be lower in the head than in the tail. However, **sEM** displayed significantly higher accuracies than all compared methods, with about 72% accuracy in the head and 89% in the tail. Similarly, Fig. 2.7B demonstrates that the distance of the cell centers inferred by **sEM** is less than 3 microns away from the expert annotations, on average, roughly corresponding to the average diameter of neurons.

Robustness to Initialization and the Number of Landmark Cells: To evaluate the robustness of **sEM** and **vEM** to the selection of landmark cells, we did the following experiment: For $\kappa \in \{5,7,9,11,13\}$ we randomly selected a set of κ cells. Both sEM and vEM were run such that these positions and the identity of these cells were used as landmark cells for computing the initial alignment, $\boldsymbol{\beta}$, that transforms the neuron positions from their atlas-based locations. The average accuracy increases with more landmark cells (Fig. 2.7C), but the best results are obtained if the landmark cells accurately portray the posture of the worm.

We further evaluated the robustness of our algorithm to initialization. Instead of initializing the centers and colors using an atlas, we initialized them using a random subset of the points selected uniformly from the observed pixels. Although initialized randomly, the effect of the atlas as a prior on the cell centers and colors led both **sEM** and **vEM** to converge quickly toward the actual cell locations, with accuracies reaching 80% (std = 0.07) for **sEM** and 79% (std = 0.1) with **vEM**.

Segmentation Quality: The probabilistic segmentation maps are used to qualitatively evaluate **sEM** and **vEM**. In Fig. 2.6, we observe that **sEM** provides sharper and more localized segmentation maps for neurons with boundaries that are more visible and more similar to the presumed borders. To quantify these sharpness attributes, we computed the spatial spread of the segmentations for the lth neuron by computing the spatial variance of using the following formula:

$$\textbf{Location spread}(l) = \mathrm{Var}_{\boldsymbol{l}_i \sim \gamma_{i,l}}(\boldsymbol{l}_i), \tag{2.13}$$

where we compute the variance of the lth neuron's map by using the γ matrix as weights. Note that $\boldsymbol{l}_i$ denotes the coordinates of the ith pixel. We then compared this metric between **vEM** and **sEM** across all neurons of the head and tail (Fig. 2.7D). The results show that while both methods perform similarly in the tail, **sEM** yields significantly sharper and more localized segmentation maps in the denser head images.

Our results show that in addition to the theoretical properties examined in [Mena et al., 2020], another advantage of **sEM** over **vEM** in the context of image segmentation with shape or size priors, is that mode collapse is prevented due to matrix balancing on γ (see **vEM** components of LUAR and PVR in Fig. 2.6).

2.4 Tracking Neurons in Behaving C. elegans

Imaging sparse fluorescent signals has become a standard tool for observing neuronal activity. To place that activity in the context of behavior, it becomes increasingly important to perform that imaging in naturally behaving animals [Lin et al., 2022]. Tracking the fluorescent sources through the moving and deforming tissue of these behaving animals is a challenging instance of a multiple object tracking (MOT) problem, and this step is typically a bottleneck for extracting clean measures of activity [Luo et al., 2021].

Recently, deep learning with convolutional neural networks has been leveraged for many MOT problems with video data including controlling self-driving cars, inferring postural dynamics in humans and animals (DeeperCut [Insafutdinov et al., 2016], DeepLabCut [Mathis et al., 2018], etc. [Wu et al., 2020]), and computational video editing (non-tracking CGI problems). These advances don't immediately generalize to videos of fluorescence reported dynamics in living tissue for several reasons.

(1) In contrast to applications like human or vehicle tracking where each object has unique identifiers that can be exploited, two fluorescence signals in the same video are often generated by nearly identical sources and therefore lack distinguishable features [Mathis et al., 2018; Meijering, 2012; Tinevez et al., 2017; Wu et al., 2020]. (2) While transfer learning has been successfully implemented in scientific applications involving natural videos (a horse galloping) [Jia Deng et al., 2009; Mathis et al., 2018], the low-level spatial and temporal features detected by these networks rarely reflect structures found in fluorescence microscopy data [Moen et al., 2019; Weigert et al., 2020]. Thus, this approach rarely reduces the quantity of additional training data required [Jia Deng et al., 2009; Mathis et al., 2018; Wang et al., 2021; Wen et al., 2021; Yu et al., 2021]. Approaches that successfully reduce training data must make hard assumptions about the underlying structure via direct parameter reduction, regularization, or data augmentation [Chaudhary et al., 2021; Nguyen et al., 2017a; Schulter et al., 2017; Wu et al., 2021]. (3) At the finest spatial scale, convolutional networks rely on images composed of many discriminable textures that typically fill an image [Weng and Zhu, 2015]. Fluorescence microscopy data, however, often has regions of interest with similar fluorescent cells surrounded by voids of black pixels. The combination of sparse global

distributions and locally dense homogeneous peaks are less well-suited to convolutional networks, as it becomes harder for convolutional networks to extract useful features for downstream tasks [Maška et al., 2014; Yan et al., 2018]. Some methods are proposed to improve the performance of convolutional networks on sparse data but their utility is not shown in the context of MOT [Jaritz et al., 2018; Yan et al., 2018]. (4) Biological videos often exhibit complex motion patterns with nonlinear deformations whereas, in contrast, most vehicle and pedestrian tracking algorithms use linear models or random walks to capture the motion [Hallinen et al., 2021].

With sufficiently high frame rates, temporal information can be used to search the vicinity of a cell's previous location and match identities by minimizing displacement over time. However, motion can often preclude achieving such a frame rate, especially when serially imaging slices of a volume or attempting to recover a signal from a dim fluorescent source. Furthermore, this motion often provides critical context for the problem being investigated (e.g. imaging neuronal dynamics to understand behavior [Susoy et al., 2021]). In these cases, it becomes beneficial to constrain a motion model by maintaining relative positions of cells, correlated motion, and priors for fluorescence dynamics.

Cell tracking methods can be categorized into the following two groups: (1) detect and link, and (2) registration-based. Detect and link algorithms have two distinct steps [Chaudhary et al., 2021; Magnusson et al., 2015; Nguyen et al., 2017a; Tinevez et al., 2017; Wang et al., 2021; Yu et al., 2021]: (1) Detection, where identity-blind candidate locations for objects are proposed by a segmentation or keypoint detection algorithm at each time frame independently. (2) Linking, where temporal associations between detected objects are determined to establish a single continuous worldline across all frames for each individual object. A major drawback of this two-step approach is the propagation of errors from the detection step. Errors that occur in the detection step are difficult to recover from, and they can have detrimental effects on linking and overall tracking quality. Several linking methods have been proposed that are robust to detection outliers, but they either require training with large amounts of manually produced ground-truth data, or are not scalable to lengthy videos [Magnusson et al., 2015; Yu et al., 2021].

An alternative approach is to directly operate in the image space and optimize some transformation parameters that align a frame to some other frame [Detlefsen et al., 2018; Lee et al., 2019; Ma et al., 2016; Mathis et al., 2018; Park et al., 2022; Schulter et al., 2017; Wen et al., 2021]. This is done by mapping the underlying image grid from the source to the reference space using the transformation parameters and interpolated pixel values. The transformation parameters must be optimized for each new image over a number of iterations.

Fortunately, recent advances in spatial transformers and differentiable grid sampling have dramatically decreased computational burden and increased performance via GPU acceleration [Detlefsen et al., 2018; Lee et al., 2019; Mazza and Pagani, 2021; Sandkühler et al., 2018; Schneider et al., 2012]. Similarly, modern optimization packages such as PyTorch allow the construction of dynamic computational graphs that support more complex nonlinear transformation families and novel cost functions with various regularizers.

Here, we build upon these recent advances to develop ZephIR, a semi-supervised multiple object tracking algorithm with a novel cost function that can incorporate a diverse set of spatio-temporal constraints that can change dynamically during optimization. Our proposed method is capable of efficiently and accurately tracking a wide range of 2D or 3D videos. It allows the user to tune a number of easily interpretable parameters controlling the relative strengths of the registration loss and other constraints, and hence generalizes well to a wide range of biological assumptions. To showcase the efficacy and versatility of our method, we demonstrate its performance on a number of biological applications, including cell tracking and posture tracking.

2.4.1 Methods

ZephIR tracks a fixed set of keypoints within a volume over time by matching keypoints between an annotated *reference* frame and an unlabeled *child* frame. This matching is done by minimizing a loss function $\mathscr{L}$ with four contributions:

$$\mathscr{L} = \lambda_R \mathscr{L}_R + \lambda_N \mathscr{L}_N + \lambda_D \mathscr{L}_D + \lambda_T \mathscr{L}_T = \boldsymbol{\lambda} \cdot \boldsymbol{\mathscr{L}}$$

We measure overlap of local image features around the keypoint via $\mathscr{L}_R$. We measure relative elastic motion between keypoints via $\mathscr{L}_N$. We measure the distance of each keypoint to the nearest candidate location from a precomputed set via $\mathscr{L}_D$. We measure smoothness of keypoint-determined dynamical features (e.g. fluorescence or motion) via $\mathscr{L}_T$. Each is described in more detail below.

The relative weights of each term, $\boldsymbol{\lambda}$, can be freely adjusted by the user to better fit a particular dataset. The user can also set the relative weights to change while tracking a single frame to allow the algorithm to shift focus to different loss components over a number of optimization iterations.

Image registration, $\mathscr{L}_R$

The first term of our algorithm measures overlap of local image descriptors.

For each keypoint i in a child frame, $I^{(c)}$, an image descriptor (a low-dimensional representation of the local image information), D, is sampled according to a sampling grid centered around that keypoint's coordinates in 3D space, $\rho_i^{(c)}$ (with fixed $z = 0$ for 2D images). We define a set of parameters, $\theta_i^{(c)}$, that is closely related to $\rho_i^{(c)}$ but may include additional transformation models, such as rotation, to characterize the sampling grid, i.e. how each descriptor is sampled from the child frame: $D(I^{(c)}, \theta_i^{(c)})$.

The descriptors are foveated to prioritize more local information relative to the neighboring features. In lieu of image pyramids [Thévenaz et al., 1998], we dynamically increase the effective resolution of the descriptors by applying a Gaussian blur at the start of optimization. The blur is decreased in magnitude every few registration iterations. Doing so avoids vanishing or exploding gradients, both of which can occur in regions with sharp, well-defined edges surrounded by a uniform background. On the other hand, restoring the original resolution of the image still provides the best available information for fine-tuning tracking results towards the end of the optimization loop.

Similarly, a set of reference descriptors that serve as registration targets are sampled from a reference frame, $I^{(r)}$. These are sampled around the user-defined annotations for that reference frame, $\rho_i^{(r)}$, according to a fixed set of parameters, $\theta_i^{(r)}$.

Using the two sets of image descriptors, our registration loop optimizes the transformation parameters, $\theta_i^{(c)}$, to minimize the following loss term:

$$\mathscr{L}_R(\theta^{(c)}) = \sum_i \left[1 - \text{CorrCoef}\Big(D(I^{(r)}, \theta_i^{(r)}), D(I^{(c)}, \theta_i^{(c)}) \Big) \right]$$

The optimized parameters $\theta_i^{(c)}$ are then used to calculate the desired results, the keypoint coordinates for the child frame, $\rho_i^{(c)}$. Note that these coordinates are also used for different loss components below, but as $\rho_i^{(c)}$ is calculated from $\theta_i^{(c)}$, gradients are always accumulated at $\theta_i^{(c)}$.

Spatial regularization, $\mathscr{L}_N$

Cellular motion within a tissue tends to be highly correlated, but these correlations can be hidden in sparse fluorescent movies that only highlight a small number of cells (or subcellular features) [Luo and Bhandarkar, 2005]. Even in less sparse movies, correlations between nearby keypoints may not be well-captured by descriptors, especially when deformations, noise, or lighting conditions prevent descriptor alignment. In order to reintroduce a similar spatial structure to the data without relying on highly specialized skeletal models, we add an elastic spring network between neighboring keypoints [Freifeld et al., 2015, 2016; Luo and Bhandarkar, 2005]. The resulting penalty to relative displacement of neighboring keypoints prevents unreasonable deformations, providing a simple and flexible spatial heuristic of the global structure and motion present in the data.

Despite ZephIR tracking a fixed a number of keypoints across the video, the spring network makes it robust to fluctuations in the number of keypoints visible in a frame. In frames where a keypoint may not be visible or present, it fails to produce useful image descriptors for registration, but the spring connections to its neighboring keypoints allow us to keep track of its approximate location.

Each of the i keypoints being tracked is connected to j nearest neighbors to define the following loss term:

$$\mathscr{L}_N = \sum_{i,j} k_{ij} \left| d_{ij}^{(c)} - d_{ij}^{(r)} \right|$$

where

$$d_{ij}^{(t)} = \|\rho_i^{(t)} - \rho_j^{(t)}\|$$

describes the distance between keypoints i and j in the frame t.

When multiple reference frames are available, the stiffness of each spring connection, k_{ij}, is further adjusted to better model the spatial patterns in the data:

$$k_{ij} = \mathrm{cov}(\rho_i^{(r)}, \rho_j^{(r)})$$

This ensures that connections between highly covariant keypoints are made stronger while connections between keypoints with more weakly correlated motion are weakened or cut accordingly.

Feature detection, $\mathscr{L}_D$

For this component of the algorithm, we solve an easier problem of identity-blind feature detection, as such detection algorithms have been shown to be fruitful in the context of tracking [Tinevez et al., 2017]. Namely, we identify key features (such as the center of a cell) present in a volume *without* matching them to a specific feature in some other volume.

This object or feature detection problem has been well-studied, and a wide variety solutions have been proposed. Solutions can range from more parameter-free algorithms (e.g. Richardson-Lucy deconvolution [Lucy, 1974; Richardson, 1972]), to algorithms requiring more fine-tuning (e.g. watershed [Beucher and Lantuejoul, 1979]). More recently, deep convolutional neural networks have shown to be powerful, effective solutions as well (e.g. StarDist [Weigert et al., 2020]). Importantly, each of these approaches may work better or worse on different classes of images. Generalization to new datasets can be hard to predict, especially for neural networks that are trained on data generated from a single source.

Our approach is to automatically evaluate simple combinations of these established algorithms by using a shallow model-selecting network. After identifying a set of candidate models, we provide the outputs of these models as input channels to a shallow and narrow convolutional neural network (CNN). If a particular model is best suited for a dataset, network weights for the corresponding input channel are increased during training while suppressing other channels. The low number of learnable parameters in the network also allows fast training for each new type of data or imaging condition, which in turn allows rapid experimentation with new selections of models to test as inputs.

The ultimate output of this selector network, $C(I^{(c)})$, is formulated as a probability map, where each pixel of the original image is assigned some probability of being a desired feature. We use this information to push tracking results towards detected features:

$$\mathscr{L}_D = \sum_i \left(1 - C(I^{(c)})[\rho_i^{(c)}]\right)$$

Temporal smoothing, $\mathscr{L}_T$

Given a sufficiently fast imaging rate, we expect pixel intensity values to be smooth across a small local patch of frames, even for cellular datasets where pixel intensities represent smoothly-varying dynamical signals [Clark et al., 2009; Dufour et al., 2015]. Thus, we attempt to maintain smoothly-varying local pixel intensities as a form of temporal regularization. For datasets where expected dynamics are appreciably slower than the imaging rate, the strongest version of this regularization is to penalize any deviation from a local zeroth-order fit. We apply this across a small patch of frames $(c-\epsilon,\ldots,c+\epsilon)$ that are registered at once, and add this to the loss for the center frame, c:

$$\mathscr{L}_T = \sum_i \sum_{t=c-\epsilon}^{c+\epsilon} \left| I^{(t)}[\rho_i^{(t)}] - I^{(c)}[\rho_i^{(c)}] \right|$$

Note that since the loss term is applied for the center frame only, it does not affect the results for the other frames despite registering all frames in the patch together. Additionally, this component of the algorithm requires registration (or approximate registration) of nearby frames, making it more appropriate in low-motion conditions or after initial coarse registration is complete.

Frame sorting

Using all or some of the loss terms listed above, a single *child frame* is registered to a *reference frame*, and all keypoints in the frame are tracked simultaneously. To fully analyze a movie, we need to register every frame to a reference frame.

For many datasets, it is best to register every child frame directly to a coarsely similar reference frame, and let annotations for that reference frame provide initial guesses for keypoints in the child frame (Fig. 2.9A). For this, we must identify a set of representative reference frames that capture the range of deformation patterns present in the movie, and we must assign each remaining frame to one of those reference frames. A pairwise distance between all pairs of frames is determined by some similarity metric (e.g. correlation coefficient) applied to low-resolution thumbnails. A k-medoids clustering algorithm is applied to these pairwise distances to identify a small number of median frames to best serve as reference frames for all other frames in the corresponding cluster (Fig. 2.9) [Mathis et al., 2018; Park et al., 2022].

In other datasets, the registration results from one frame in a cluster may provide useful insight into the solution for a different frame in that cluster. For example, a frame that is close (in deformation space) to the reference may be easy to track. The tracked results from that frame, in turn, may provide a better guess for keypoints in a frame that is further away from the reference. This can reduce the distance between the initial guess and correct positions, and thus reduce the difficulty of the optimization problem. Thus, every child frame being registered is associated not only with a reference frame (a registration target), but also a previously registered *parent frame*, which provides the initialization prior to optimization (Fig. 2.9B).

Additionally, the learning rate for the child frame is partly determined by the distance between the parent and child frames. We expect that when a parent-child pair are close in the deformation space, the keypoints do not undergo significant local displacements. Hence, a low learning rate is applied for a similar parent-child pair, scaling up to a high learning rate in the case of a dissimilar pair to allow tracking of features much further away. The combination of these effects produces a flexible limit on the range of possible optimization results for the child frame based on coarse similarity to its parent frame [Keskar et al., 2016; Ruder, 2016; Schaul et al., 2012].

To take full advantage of this parent-child interaction, we sort all frames into distinct sequences of parent-child frames based on similarity. Each of the resulting *branches* begins from a previously selected median reference frame. The subsequent child frames are selected to minimize the distance from a parent frame until every frame is assigned to a branch. Doing so produces unique sets of frames that stem from each reference frame, naturally forming clusters that separate similar frames from dissimilar ones. This is particularly useful for datasets that repeatedly sample from a limited set of postures or global spatial structures (e.g. locomotion).

However, not all datasets have temporal patterns that can reliably make use of the similarity-based initialization method. For such datasets, a chronologically sorted queue may be more reasonable and provide better accuracy overall, where a branch simply stems from each reference frame both forwards and backwards in time until it encounters the first frame, the last frame, or another branch (Fig 2.9). Note that the parent-child interactions during tracking are still the same regardless of the sorting method. For a chronologically sorted queue, the controlled variation of learning rates effectively allows us to adapt to different capture frame rates. A high frame rate video often captures smooth motion that benefits from low learning rates but a low frame rate video does not.

User intervention

Our pipeline allows a user to dramatically improve tracking quality in various ways by providing further supervision. Providing additional fully annotated frames will improve registration targets to better match descriptors from similar frames. Strategically selecting a new reference frame can have dramatic impacts on frame sorting as well, creating opportunities to form tighter clusters of parent-child branches.

Furthermore, when multiple reference frames are present, covariance of keypoints in those frames helps better define an implicit global spatial structure by modulating stiffnesses of the spring connections between neighboring keypoints, k_{ij}. Any additional reference frames can provide more accurate covariances, and thus a spatial model that is more accurately tailored for that particular dataset.

Partially annotated frames are not used to seed sorted frame branches nor used to sample reference descriptors. Still, all user annotations present in the frame are utilized to improve the tracking quality of the remaining keypoints in that frame (Fig. 2.8D).

Firstly, prior to gradient descent, displacements between all available annotations and their corresponding coordinates from the parent frame are used to interpolate a flow field. This flow field serves as a rough model of the global motion between the two frames [Freifeld et al., 2015; Ma et al., 2016; Schulter et al., 2017]. We sample from the flow field at the remaining keypoints coordinates in the parent frame and apply the resulting estimated displacements to initialize the keypoints closer to their new positions in the child frame. This is particularly helpful for pairs of parent-child frames with large motion between them, and the flow field can always be improved in both precision and accuracy by adding more annotations for the child frame.

Secondly, the spatial regularization during the optimization process, $\mathscr{L}_N$, also makes good use of any partial annotations. The annotations are fixed in place, but the spring connections to their neighbors remain a crucial component of the backwards gradient calculations and helps to "pull" the connected keypoints into place.

To streamline the process of providing user supervision, we offer a browser-based graphical user interface that provides an intuitive, simple environment to produce and save further annotations. Since our approach lacks a slow "training" phase, any new annotations can be applied to tracking a frame directly from the GUI. A macro available in the GUI executes a temporary state of the algorithm quickly and efficiently, allowing users to see the precipitated improvements immediately.

Additionally, the GUI provides an opportunity for users to provide supervision without creating new annotations. The user may upgrade individual results into annotations or entire frames into new reference frames by marking them as correctly tracked. These user-confirmed frames will be treated as a regular reference frame next time the algorithm is executed, benefiting from all the improvements to tracking quality discussed previously. These improvements to the rest of the results can be observed immediately by executing the algorithm from the GUI.

2.4.2 Results

Neurons in crawling worms (*C. elegans*)

Optical methods based on fluorescence activity of calcium binding indicators has become a standard tool for observing neuronal activity in *C. elegans*. To do so, it is necessary to track fluorescent signals from individual neurons across every frame in a recording. This poses a significant challenge, particularly when the animal is allowed to freely crawl. The worm's brain undergoes fast, dramatic, nonaffine deformations, exhibiting a large variety (forward and backward motion, omega turns, coils, pharyngeal pumping, etc.) and magnitude (up to ten microns relative to an internal reference frame) of movements as the animal behaves [Hallinen et al., 2021; Nguyen et al., 2016a; Susoy et al., 2021; Venkatachalam et al., 2016a].

Many solutions have been proposed to track fluorescent neurons in *C. elegans*. Two step (detect and link) approaches often suffer from the lack of reliable detection algorithms and require relatively low frame-to-frame motion in order to accurately link the detected neurons [Nguyen et al., 2017a; Tinevez et al., 2017; Wen et al., 2021]. Similarly, deep learning approaches are limited by insufficient training data, often failing to generalize across different animals, even those within the same strain [Lagache et al., 2020b; Park et al., 2022; Yu et al., 2021]. While these approaches have provided important insight and progress, there remains substantial need for improvement in accuracy and efficiency when tracking many neurons in freely behaving worms.

Fig. 2.10 describes the workflow and performance of ZephIR on tracking a set of 178 fluorescent neurons in the head of a freely behaving worm across a 3D recording of approximately 4.4 minutes (1060 frames @ 4Hz). We collected this data for the purpose of testing this algorithm using a microscope and technique similar to that described in [Venkatachalam et al., 2016a]. The video has been centered and rigidly rotated to maintain a consistent orientation of the worm, but no further straightening has been done. With only a few manually annotated reference frames, ZephIR already achieves state-of-the-art MOT accuracy [Chenouard et al., 2014; Maška et al., 2014; Matula et al., 2015] as reported on similar datasets in recently published works [Nguyen et al., 2017a; Wen et al., 2021; Yu et al., 2021] (Fig. 2.10A,B).

We further improve on the accuracy of the initial results by providing additional supervision. We randomly selected ten neurons uniformly distributed throughout the brain to verify and use as partial annotations across all frames. Because the initial results already achieved high accuracy, they only required correction for a subset of frames (≈ 15%). After this correction and validation, annotations for these ten neurons were re-classified as manual annotations in all frames. The partial annotations produce a dramatic improvement in accuracy (red data point in Fig. 2.10B) without the need to verify entire frames.

Through this workflow, we are able to achieve a sufficiently high accuracy to extract good, meaningful neuronal activity traces across the entire recording (Fig. 2.10D) [Clark et al., 2009; Dufour et al., 2015]. Many neurons show clear correlation with observed behaviors, and the activity patterns are comparable to previously published works [Hallinen et al., 2021; Leifer et al., 2011; Nguyen et al., 2017a; Shipley et al., 2014].

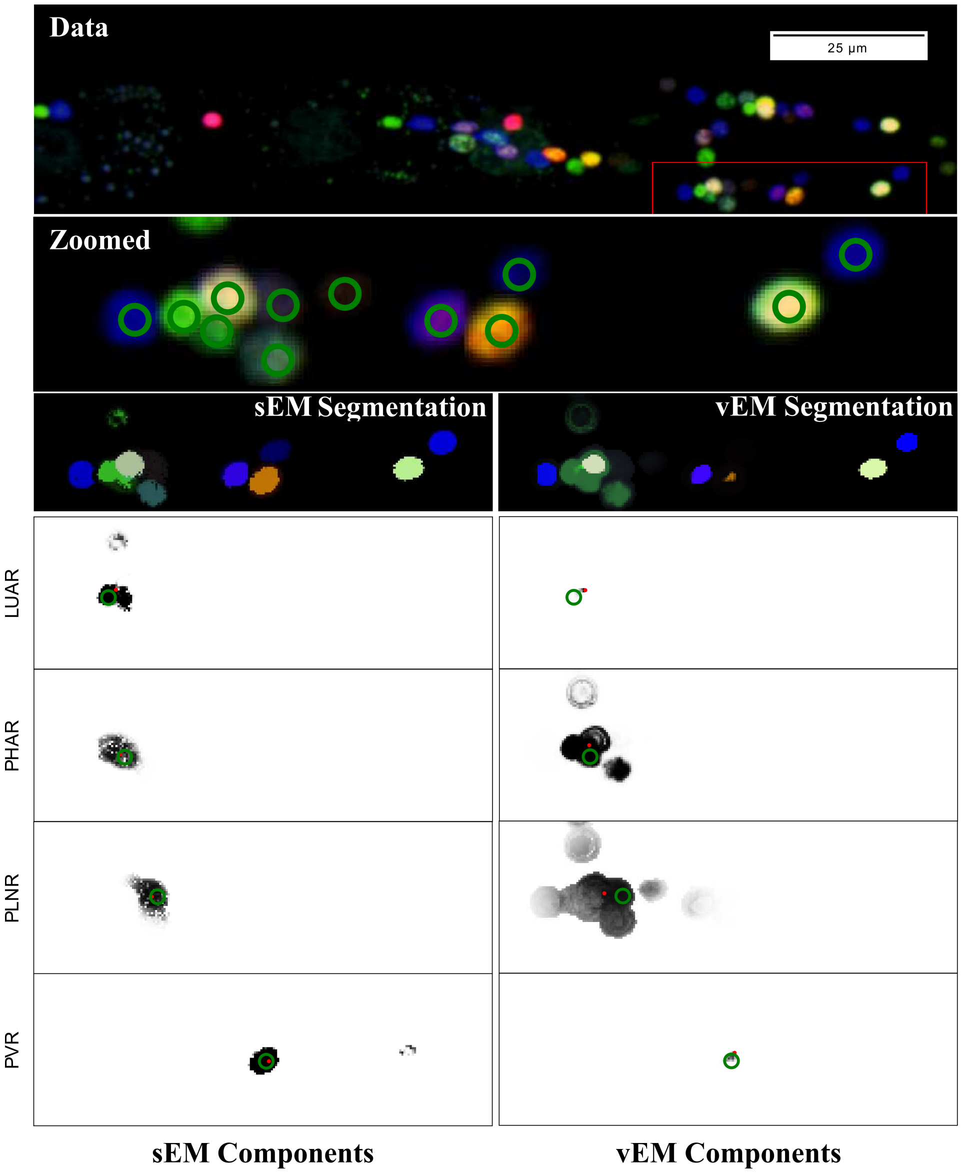

Figure 2.6: **Qualitative comparison of sEM and vEM segmentation maps** A maximum intensity projection of the volumetric NeuroPAL image is shown in the Data panel. To show the segmentation performance we zoom in on the Right Lumbar Ganglion (Zoomed panel; red rectangle in Data panel). Green circles represent expert annotations. Segmentation panels show the probabilistic segmentation of the components ($\boldsymbol{\gamma}_l$). Each component is shown in its inferred color ($\boldsymbol{\mu}_l^n$) using both algorithms; **sEM** provides segmentation maps that are sharper and more cell-like, with more visible boundaries and less spread. Components panels show example per-component probabilistic segmentation maps ($\boldsymbol{\gamma}_l$), with darker pixels having posterior assignment probabilities closer to one for each component. Red dots are the centers inferred by both algorithms. vEM tends to miss some components (LUAR, PVR) and spreads its mass to irrelevant regions for some other components (PHAR, PLNR).

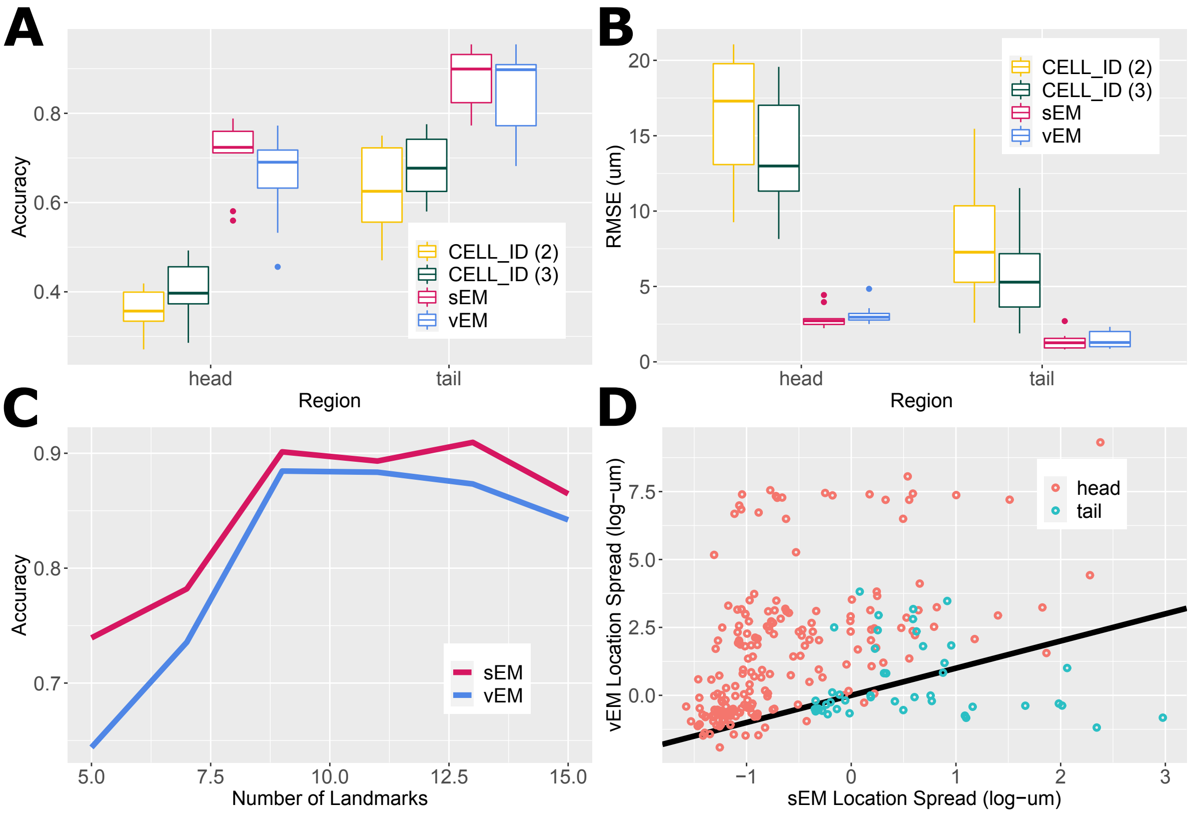

Figure 2.7: **Quantitative evaluations for neuron identification and segmentation sharpness A:** Comparison of neuron identification accuracy between **sEM**, **vEM**, **CELL-ID** (2) (detect+id) [Yemini et al., 2019a], and **CELL-ID** (3) (filter+detect+id); **sEM** accuracy is significantly higher than all other methods in the head, and slightly better than **vEM** in the tail. Both **vEM** and **sEM** outperform the multi-step **CELL-ID** approach. **B:** Root Mean Squared Error (RMSE) between the inferred neural locations and their expert annotated location; similar to **A**, **vEM** and **sEM** outperform **CELL-ID**, and **sEM** achieves lower RMSE than **vEM** in both head and tail. **C:** Average accuracy of **sEM** and **vEM** as a function of randomly chosen landmark cells; **sEM** slightly outperforms **vEM** and the accuracy increases as we use more landmark cells for the initial alignment. **D:** Median spatial spread of the neuron segmentation maps resulting from **sEM** and **vEM**. Here each dot indicates the spread of a particular neuron, with median taken across the population of worms. **vEM** spreads its mass for each component more than **sEM**, resulting in lower confidence in segmentation assignments. This is visually observed in Fig. 2.6 where the segmentation maps of **sEM** are more localized and sharper than those from **vEM**.

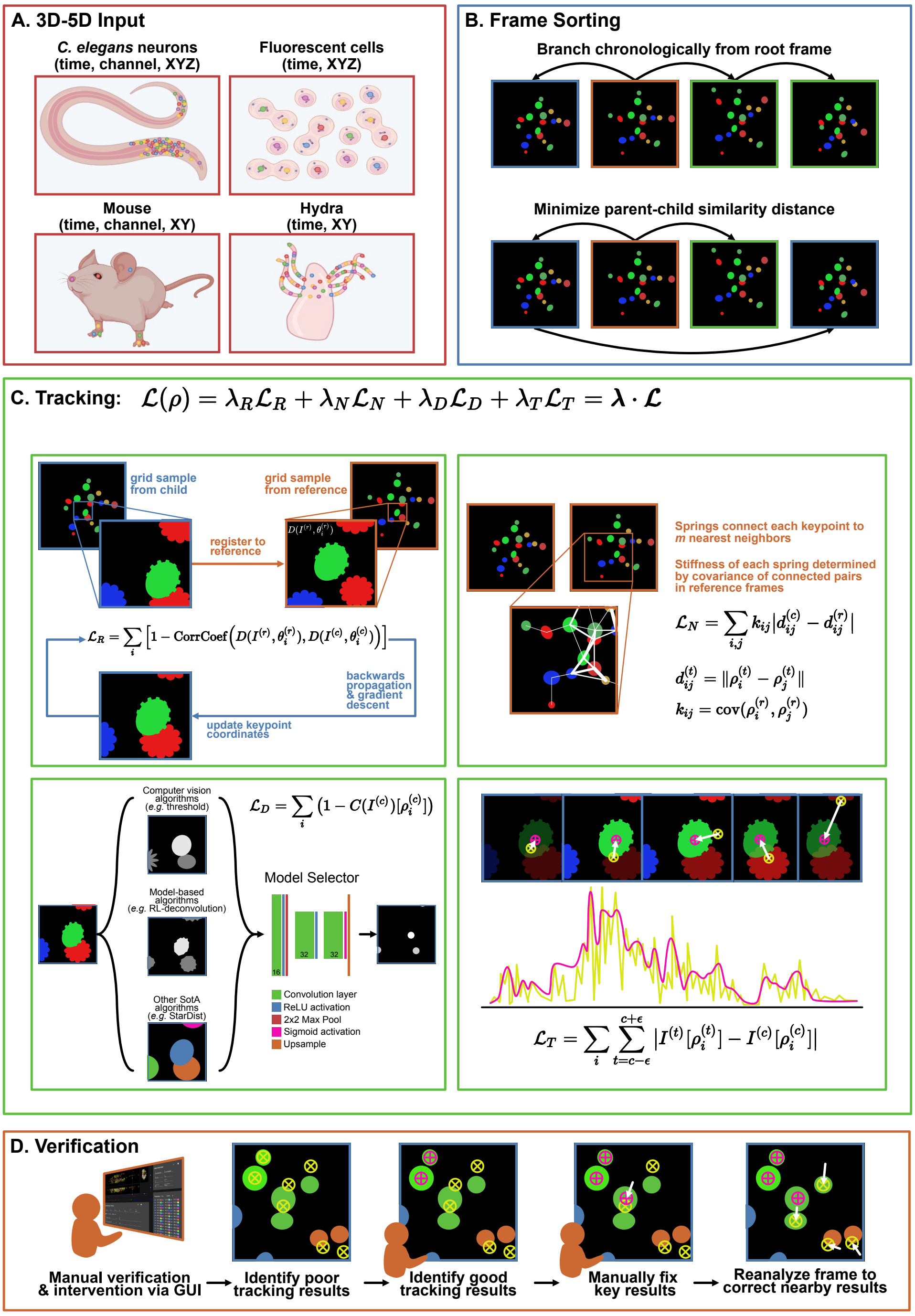
A. 3D-5D Input
C. elegans neurons (time, channel, XYZ)
Fluorescent cells (time, XYZ)
Mouse (time, channel, XY)
Hydra (time, XY)
B. Frame Sorting
Branch chronologically from root frame
Minimize parent-child similarity distance
C. Tracking: $\mathcal{L}(\rho) = \lambda_R \mathcal{L}_R + \lambda_N \mathcal{L}_N + \lambda_D \mathcal{L}_D + \lambda_T \mathcal{L}_T = \boldsymbol{\lambda} \cdot \boldsymbol{\mathcal{L}}$
grid sample from child
grid sample from reference
register to reference
$D(I^{(r)}, \theta_i^{(r)})$
$\mathcal{L}_R = \sum_i \left[1 - \mathrm{CorrCoef}\left(D(I^{(r)}, \theta_i^{(r)}), D(I^{(c)}, \theta_i^{(c)})\right)\right]$
backwards propagation & gradient descent
update keypoint coordinates
Springs connect each keypoint to *m* nearest neighbors
Stiffness of each spring determined by covariance of connected pairs in reference frames
$\mathcal{L}_N = \sum_{i,j} k_{ij} |d_{ij}^{(c)} - d_{ij}^{(r)}|$
$d_{ij}^{(t)} = \|\rho_i^{(t)} - \rho_j^{(t)}\|$
$k_{ij} = \mathrm{cov}(\rho_i^{(r)}, \rho_j^{(r)})$
Computer vision algorithms (*e.g.* threshold)
Model-based algorithms (*e.g.* RL-deconvolution)
Other SotA algorithms (*e.g.* StarDist)
$\mathcal{L}_D = \sum_i (1 - C(I^{(c)})[\rho_i^{(c)}])$
Model Selector
16
32
32
🟩 Convolution layer
🟦 ReLU activation
🟥 2x2 Max Pool
🟪 Sigmoid activation
🟧 Upsample
$\mathcal{L}_T = \sum_i \sum_{t=c-\epsilon}^{c+\epsilon} |I^{(t)}[\rho_i^{(t)}] - I^{(c)}[\rho_i^{(c)}]|$
D. Verification
Manual verification & intervention via GUI
Identify poor tracking results
Identify good tracking results
Manually fix key results
Reanalyze frame to correct nearby results

Figure 2.8: **Overview of ZephIR algorithm A.** Examples of input datasets. ZephIR can track keypoints in various biological systems, including fluorescent cellular nuclei in a tissue and body parts that summarize a posture. Input dimensions can range from 3D (time, XY) to 5D (time, channel, XYZ). Colored dots indicate example keypoints to be tracked. **B.** Frame sorting schemes. A branch defines an ordered queue of frames to be tracked. Each branch begins at a manually annotated reference frame (orange), (cont. on next page)

Figure 2.8: **B.** but subsequent parent (blue) and child (green) frames in a single branch can be sorted either by chronology (top) or by minimizing the similarity distance between each parent-child pair (bottom). **C.** Overview of tracking loss. Tracking loss is comprised of four terms: 1) overlap of local image features around each keypoint, sampled from the current frame and its nearest reference frame, 2) elastic connections between neighboring keypoints with varying stiffnesses based on covariance of the connected keypoints, 3) proximity to features detected by a shallow model selector network that takes in a number of existing feature detection software as input channels, 4) smoothness of temporal dynamics at each keypoint position. **D.** Overview of steps for manual verification and additional supervision. Users can verify tracking results as correct or identify incorrect results. After fixing a few key incorrect results, ZephIR can use those new annotations as well as the verified correct tracking results to improve tracking results for all other keypoints in that frame (and all its child frames).

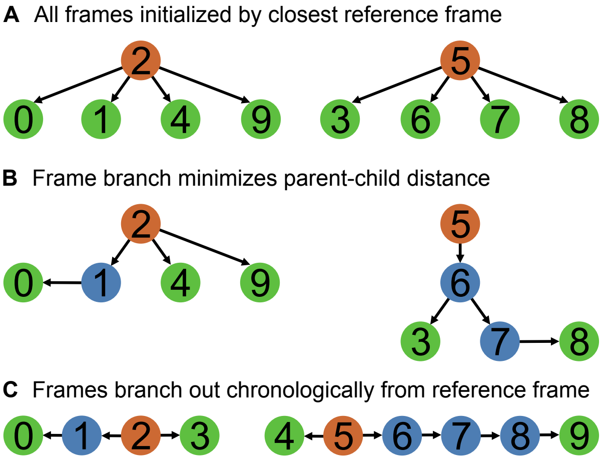
A All frames initialized by closest reference frame
2
5
0
1
4
9
3
6
7
8
B Frame branch minimizes parent-child distance
2
0
1
4
9
5
6
3
7
8
C Frames branch out chronologically from reference frame
0
1
2
3
4
5
6
7
8
9

Figure 2.9: **Overview of frame sorting strategies** Orange indicates fully annotated reference frames, blue indicates parent frames with at least one child frame, and green indicates child frames. **A.** In the simplest strategy, all frames are initialized by the closest reference frame. **B.** Frames are sorted into ordered queues based on similarity. Each of these branches start with a reference frame, and new child frames are added such that the parent-child similarity distance is minimized, naturally clustering similar frames around each reference frame. **C.** Frames are sorted chronologically, branching both forward and backwards from each reference frame.

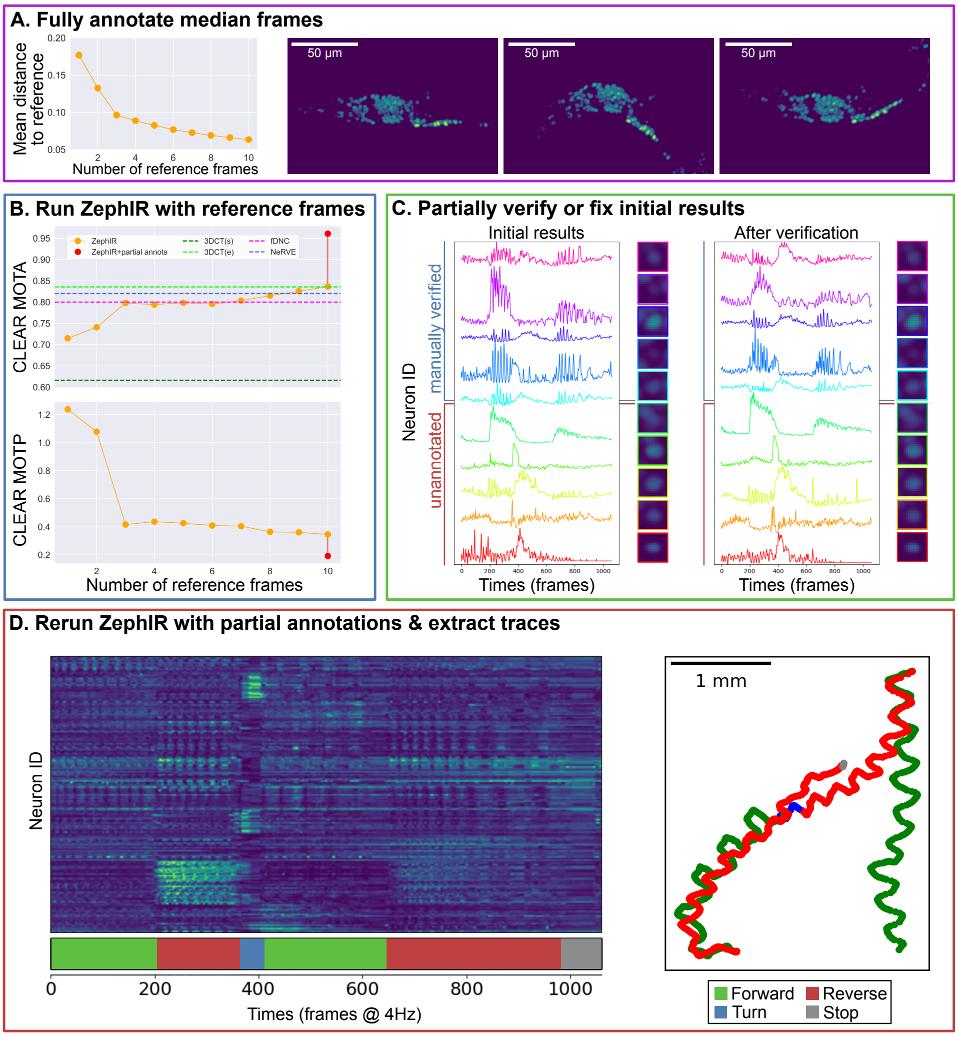

Figure 2.10: **Results for tracking GCamP fluorescent neuron nuclei in 3D volumes of freely behaving *C. elegans* A.** Plot of mean distance to the nearest reference frame vs the number of reference frames (left), and the first three median frames recommended by ZephIR's k-medoids clustering algorithm (right). The first three median frames clearly represent the three main postures that the worm cycles through as it crawls. **B.** MOT accuracy (higher is better) and precision (lower is better) vs the number of reference frames. Note that once the majority of the postures present in the data is well-represented by the first three reference frames, subsequent additions returns diminished improvements. Last data point shows ZephIR's accuracy using 10 reference frames with 10 partial annotations across all frames (panel C). We also compare ZephIR's accuracy with Neuron Registration Vector Encoding (NeRVE) [Nguyen et al., 2017a], fast Deep Neural Correspondence (fDNC) [Yu et al., 2021], and 3DeeCellTracker [Wen et al., 2021] in both single (3DCT(s)) and ensemble (3DCT(e)) modes as reported in their respective publications.

Figure 2.10: (cont. from previous page) Note that the accuracies from 3DeeCellTracker reflects both errors in detection and tracking. **C.** 10 neurons were randomly selected to be verified or corrected to serve as partial annotations. Traces of 5 of these neurons extracted using the initial ZephIR results with 10 reference frames (left), and those using verified true positions (right) are shown, along with 5 other randomly selected neurons. Traces are calculated as fold change over the baseline, where the baseline is defined as the intensity in the first frame. Tracking quality for these 10 neurons can also be seen in individual crops around the neurons averaged across all frames (sharper image of the cell at the center reflects better accuracy and precision in tracking). Note how the five unannotated neurons show improvements in tracking quality after the addition of partial annotations, exemplifying the effects of partial annotations on the unannotated neurons in the same frame. **D.** Neuronal activity traces from 178 neurons, extracted using results from ZephIR with 10 reference frames and 10 partial annotations in all frames. Traces are calculated as fold change over the baseline, where the baseline is defined as the intensity in the first frame. Behavior is shown in the ethogram below the heatmap. Trajectory of the worm (t=0 at bottom right) is also colored with the behavior state at the time. Trajectory of the worm matches changes in behavior over time as expected, and many of the neuronal activity traces show strong correlation with behavior. A video marked with tracking results is available at: `https://github.com/venkatachalamlab/ZephIR/blob/main/docs/examples.md`.

CHAPTER 3

Demixing Signals in Deforming Tissues

3.1 Introduction

Recent advances in imaging techniques have enabled the capture of functional neural ensembles *in vivo* within a wide variety of animal models [Ahrens et al., 2013a; Flusberg et al., 2008; Mann et al., 2017; Prevedel et al., 2014]. Demixing the recorded video signals into estimates of individual neural activity remains a critical bottleneck in the analysis of these large and complex datasets. Previous approaches for extracting individual neural activity traces have involved either region of interest (ROI) methods [Barbera et al., 2016; Dombeck et al., 2007; Göbel et al., 2007; Hofer et al., 2011; Kerlin et al., 2010; Kerr et al., 2005; Nguyen et al., 2016b; Niell and Smith, 2005; Tian et al., 2009; Venkatachalam et al., 2016c] or matrix factorization methods based on principal components analysis (PCA) or independent components analysis (ICA) [Mukamel et al., 2009; Reidl et al., 2007; Siegel et al., 2007; Stetter et al., 2001] or sparse coding [Pachitariu et al., 2013, 2017].

Non-negative matrix factorization (NMF) [Lee and Seung, 1999, 2001; Paatero and Tapper, 1994] based models have been introduced to demix signals from recordings of calcium activity [Andilla and Hamprecht, 2013, 2014; Haeffele et al., 2014; Maruyama et al., 2014; Pachitariu et al., 2017; Pnevmatikakis et al., 2016; Zhou et al., 2018]. A prerequisite for the success of these methods, to permit blind-source separation, is that the imaged ROI remains motionless even when the animal is awake, satisfying the assumption that the spatial footprints of signal sources remain stationary. To facilitate NMF assumptions and remove excess motion variability, a common pre-processing step before NMF is the registration of the imaging volumes to a common template space.

There is a wealth of literature in the medical imaging community regarding the registration of volumetric images to template volumes to account for morphological variability [Klein et al., 2009]. These methods have proven to be very effective in registering images that have similar intensity profiles but they tend to introduce artifacts when the template image and the moving image have different appearances, low signal to noise ratio, or abnormalities [Zeng et al., 2016]. Furthermore, the computational complexity of these methods is a bottleneck since there are potentially tens of thousands of frames in volumetric calcium videos that need to be registered. A number of pipelines [Dubbs et al., 2016; Pachitariu et al., 2017; Pnevmatikakis and Giovannucci, 2017] implement existing sub-pixel registration techniques [Guizar-Sicairos et al., 2008] to enable the rigid and non-rigid registration of calcium videos in a computationally efficient manner. Assuming that the motion does not involve large shifts in the field of view (FOV), these techniques aim to register individual video frames to a template frame through fast patchwise rigid transformations. However, they too are not built to handle severe deformations and large intensity variations.

Recent whole-brain imaging techniques of the model organism *C. elegans* [Kato et al., 2015; Nguyen et al., 2016b; Prevedel et al., 2014; Schrödel et al., 2013; Venkatachalam et al., 2016c] have opened up an exciting new avenue of research, enabling simultaneous recording of neural dynamics and freely-moving behaviors in the same animal. Even during restrained imaging, worms can exhibit highly-nonlinear motion [Girard et al., 2007; Larsch et al., 2013; Voleti et al., 2019], violating the assumptions that enable NMF-based signal separation and overstretching the capabilities of fast piecewise rigid registration techniques. Therefore, common approaches have been to apply motion tracking and simple pixel-averaging around cellular tracking ROIs in two discrete steps, often followed by time-consuming supervision and manual correction of the results [Kato et al., 2015; Nguyen et al., 2016b; Venkatachalam et al., 2016b]. One way to perform motion tracking is to use a second imaging channel to record a temporally-invariant fluorescent marker (such as RFP) which is insensitive to calcium activity. By using such cellular motion tracking markers, calcium activity can then be extracted by averaging the pixel values in the ROI that overlap with the marker. However, this approach is flawed for at least two reasons: 1) ROI averaging in densely-packed cell regions is prone to mixing signal between different neurons, due to limitations in optical resolution, and 2) introducing a second imaging channel effectively requires experimenters to reduce the frame rate and/or spatial resolution by at least half in order to acquire this channel or add an additional optical path and camera. On the other hand, if tracking is performed only on the calcium imaging channel, due to the low signal-to-noise regime and calcium signal fluctuations, tracking approaches may miss cellular markers at time points when the cells become dim, creating downstream errors in tracking and demixing.

In general, tracking cells in moving animals (and even restrained animals with restricted mobility), has proven to be a challenging machine vision problem [Hirose et al., 2017]. Cell nuclei have similar shapes, thus providing only a limited set of unique features to facilitate their tracking. Spatial noise represents a further, inherent limitation, due to the microscopic size of the objects under investigation. Most available microscopy approaches scan the animal in both space and time to achieve volumetric video recordings. Therefore, there are fundamental limits in reaching the high spatiotemporal resolution necessary to resolve unique cell identities and extract their calcium signals through tracking techniques.

Even if high accuracy cell tracking can be achieved, another issue with extracting calcium signal around tracked ROIs is that many existing volumetric optical imaging setups have a relatively poor resolution in the depth axis, characterized by an elongated point-spread-function [Yang and Yuste, 2017]. This phenomenon causes the calcium signals of nearby cells to be mixed, which in turn causes the pixel-wise signal read-out to be an inaccurate portrayal of actual neural activity.

Orthogonally, there have been NMF techniques that are invariant to signal shift, such as convolutive NMF [O'grady and Pearlmutter, 2006; Smaragdis, 2006]. However, these techniques model discrete translation based shifts and are not suitable for modeling the complex deformable motion exhibited across biological volumetric recordings.

In the case of *C. elegans* imaging, worms can exhibit nonlinear motion (even when immobilized using popular paralytics [Larsch et al., 2013; Venkatachalam et al., 2016b; Voleti et al., 2019]) and variability in their neural firing patterns over time, making the application of previous techniques such as Normcorre [Pnevmatikakis and Giovannucci, 2017] or convolutive NMF ineffective. To surmount these issues, we introduce deformable non-negative matrix factorization (dNMF) to jointly model the motion, spatial shapes, and temporal traces of the observed neurons in a tri-factorization framework. Instead of the two-step approach of sequentially tracking then demixing calcium signals, we update motion parameters together with updates in the spatial and temporal matrices. To ensure that our model is not overfitting and picking up spurious motion and signal, we use regularized models for cell shapes, temporal fluctuations, and deformations. The model parameters capture the worm's motion corresponding to a fixed, spatial representation of the video, enabling the deformation terms to match the worm's posture at each time frame. Our framework is general and is suitable for decomposing videos into a set of motion parameters, fixed spatial representations for image components, and temporally varying signals with underlying linear and/or nonlinear motion. This approach can be considered a generalization of the model developed in [Peng et al., 2012] (applied to calcium imaging data by [Poole et al., 2015]), which restricts attention to affine transformations.

We validate our method on an intensity-varying particle-tracking simulation and compare it to state-of-the-art calcium-imaging motion-correction techniques [Pnevmatikakis and Giovannucci, 2017] followed by NMF [Pnevmatikakis et al., 2016]. We then demonstrate the ability of our framework to extract calcium traces from all neurons in the head and tail of semi-immobilized *C. elegans* exhibiting nonlinear motion. We use a dataset of 42 animals, 21 worm heads and 21 worm tails, recorded for 4 minutes each while presenting three stimuli, a repulsive concentration of salt and two attractive odors. We find that the proposed approach outperforms both ROI averaging and standard NMF, delivering more accurate tracking and demixing than either of these methods in this dataset.

Finally, after accurate extraction of neural activity signals from each animal, a post-processing normalization step is still required in order to compare neurons of the same type, across a population of animals. This is because factors such as variable illumination, anisotropy associated with animal orientation, and a lack of stereotypy in fluorescence expression across animals introduce substantial variability into the baseline and amplitude of the extracted neural activity signals. Standard post-processing approaches based on estimating $\Delta F/F_0$, do not resolve this variability, which if uncorrected will confound any group-type neural comparisons across animals. Even worse, outlier signals that arise due to mistracking and demixing can considerably warp the mean signal measured across a population of animals, especially when the neuron type of interest is dim and is present next to neuron types with brighter signal.

To reduce this excess variability across animals, we introduce a time-series normalization approach, termed quantile regression. This approach optimizes for a linear transformation of time-series intensities in a group of samples (e.g., all traces extracted from a given cell type over all animals), transforming the time-series samples to have matched histograms. We compare this approach with z-scoring and advocate its adoption for population-based time-series analysis due to several desirable properties. In particular, our approach retains the approximate baseline and magnitude across a population of neurons of the same type, while maintaining robustness against outlier signals. Lastly, we introduce an option for ensuring the non-negativity of the normalized signals, when appropriate for the biological measurements being performed.

3.2 Deformable Non-negative Matrix Factorization

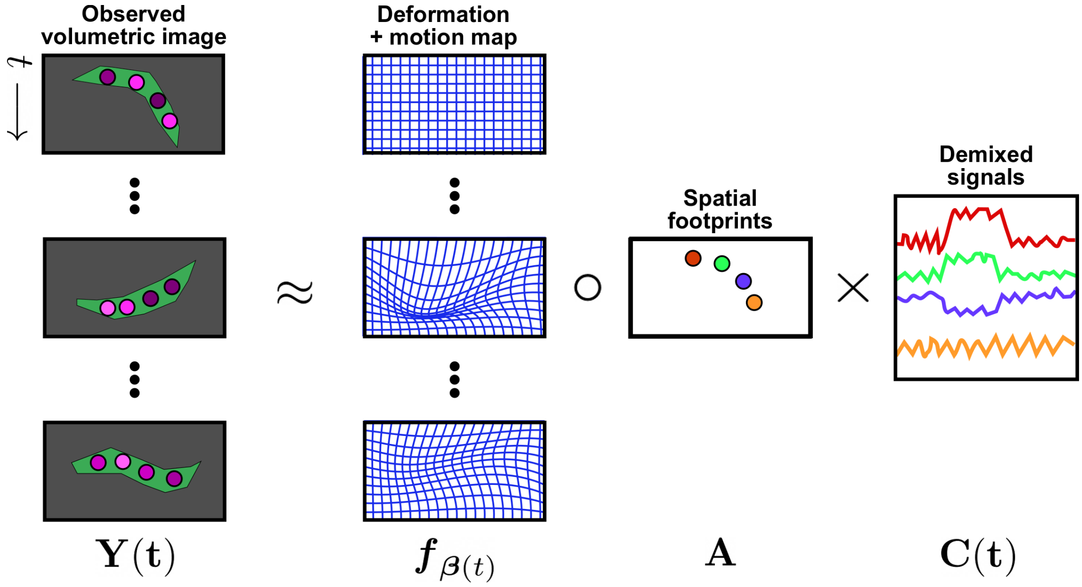

Figure 3.1: **Schematic of the deformable non-negative matrix factorization model** The volumetric time series data $\boldsymbol{Y}(t)$ is factorized into time-varying deformation + motion maps $\boldsymbol{f}_{\boldsymbol{\beta}(t)}$ which transform the factorized signal (with spatial footprints $\boldsymbol{A}$ multiplied by time-varying intensity coefficients, $\boldsymbol{C}(t)$) onto the observed data volumes.

The joint motion correction and signal extraction framework proposed here involves several steps illustrated in Figure 4.8. First, the volumes undergo several pre-processing steps that involve coarse tracking, background subtraction and smoothing, details of which are discussed in the "Pre-processing steps" subsubsection below. The pre-processed volumes are then subjected to simultaneous deformation compensation and signal demixing using a matrix tri-factorization model.

First, we introduce notation. Let $\boldsymbol{Y}_t \in \mathbb{R}^d$ denote the d-pixel vectorized volumetric image at time $t = 1, \ldots, T$. We seek to decompose the observations, $\boldsymbol{Y}_t$, into a factorization involving a time-varying deformation term, $\boldsymbol{f}_{\beta_t}$ that acts on a time-invariant canonical representation of k object shapes encoded by $\boldsymbol{A}$. The time-varying spatial signatures, $\boldsymbol{f}_{\beta_t}(\boldsymbol{A}) \in \mathbb{R}^{d \times k}$, are then multiplied by signal carrying coefficients $\boldsymbol{C}_t \in \mathbb{R}^k$. We also encourage model parameters to be "well-behaved" using regularization functions, $\mathscr{R}$ (details of which will be outlined later). The resulting objective function is:

$$\min_{\boldsymbol{A},\boldsymbol{C},\boldsymbol{\beta}} \sum_{t=1}^{T} \left\| \boldsymbol{Y}_t - \boldsymbol{f}_{\boldsymbol{\beta}_t}(\boldsymbol{A})\boldsymbol{C}_t \right\|_2^2 + \mathscr{R}(\boldsymbol{A}, \boldsymbol{C}, \boldsymbol{\beta}) \quad (3.1)$$

$$\text{s.t. } \boldsymbol{A}, \boldsymbol{C}_{1:T} \geq 0.$$

This formulation differs from standard NMF techniques [Lee and Seung, 2001] in that the spatial footprint term consists of a time invariant term, $\boldsymbol{A}$ and a time varying term, $\boldsymbol{f}_{\boldsymbol{\beta}_t}$, which is a differentiable transformation parametrized by $\boldsymbol{\beta}_t$, that deforms the canonical representation into the t-th time frame. $\boldsymbol{\beta}_t$ encapsulates the motion parameters and is usually low dimensional to avoid over-parameterization and overfitting. The regularization $\mathscr{R}(\cdot)$ further constrains the possible choice of spatial footprints, signal coefficients, and spatial deformations. Figure 4.8 illustrates the model. Next, we detail two possible parameterizations of the spatial terms, $\boldsymbol{A}$ and $\boldsymbol{f}$.

3.2.0.1 Spatial component: non-parametric model

Similar to the standard NMF models, we can parameterize $\boldsymbol{A}$ using a d-by-k matrix, where d is the number of pixels of one time frame of the video and k is the number of objects that are present. We use a Gaussian interpolant, $\boldsymbol{T}_t$, to transform these spatial footprints to arbitrary locations such that $\boldsymbol{f}_{\boldsymbol{\beta}_t}(\boldsymbol{A}) = \boldsymbol{T}_t\boldsymbol{A}$, where $\boldsymbol{T}_t$:$\mathbb{R}^{d\times d}$ and

$$\boldsymbol{T}_t[i,j] = \exp\left(\frac{\left\|\boldsymbol{\beta}_t\Psi(\boldsymbol{x}_j) - \boldsymbol{x}_i\right\|_2^2}{2\sigma^2}\right). \tag{3.2}$$

Here, $\boldsymbol{x_i},\boldsymbol{x_j} \in \mathbb{R}^3$ denote the coordinates of two arbitrary pixels in the volume. Ψ:$\mathbb{R}^3 \to \mathbb{R}^p$ denotes a basis mapping of coordinates to enable non-linear deformations and $\boldsymbol{\beta}_t$ is a 3-by-p matrix that parametrize the deformations. For example, in the case of a quadratic polynomial basis, $\boldsymbol{\beta}_t$ would be a 3-by-10 matrix, and Ψ:$\mathbb{R}^3 \to \mathbb{R}^{10}$ would be the quadratic basis function $\Psi([x,y,z]^T) = [1,x,y,z,x^2,y^2,z^2,xy,yz,xz]^T$. The choice of σ controls the amount of the spread of the mass of a pixel into nearby pixels.

3.2.0.2 Spatial component parametrization: Gaussian functions

When we have strong prior information about the component shapes we can incorporate that into the model using an appropriate parameterization for the spatial footprints. Neural activity is most commonly imaged using cytosolic or nuclear-localized calcium indicators; nuclear-localized indicators can be reasonably modeled using ellipsoidally-symmetric shape models. Specifically, we observed that the spatial component of the neurons in the videos analyzed here, of *C. elegans* imaged using nuclear-localized calcium indicators, can be well approximated using three-dimensional Gaussian functions. By taking advantage of this observation we can reduce the number of parameters in $\boldsymbol{A}$ from one parameter per pixel per component, to k 3D centers (3 parameters per each neuron) and k covariance matrices (6 parameters per each neuron using the Cholesky parameterization). Formally, we model the footprint of component k using a 3-dimensional Gaussian function with location parameters $\boldsymbol{\mu}_k \in \mathbb{R}^3$ and shape parameters $\boldsymbol{\Sigma}_k \in \mathbb{R}^{3\times 3}$.

Under this new spatial model for $\boldsymbol{A} = \{\boldsymbol{\mu}_{1:K}, \boldsymbol{\Sigma}_{1:K}\}$, we modify the $\boldsymbol{f}_{\boldsymbol{\beta}_t}$ function to match this parameterization to have $\boldsymbol{f}_{\boldsymbol{\beta}_t}(\boldsymbol{A}) \in \mathbb{R}^{d\times k}$:

$$\boldsymbol{f}_{\boldsymbol{\beta}_t}(\boldsymbol{A})[i,k] \approx \exp\left([\boldsymbol{p}_i - \boldsymbol{\beta}_t\Psi(\boldsymbol{\mu}_k)]^T\boldsymbol{\Sigma}_k^{-1}[\boldsymbol{p}_i - \boldsymbol{\beta}_t\Psi(\boldsymbol{\mu}_k)]\right), \tag{3.3}$$

where $\boldsymbol{p}_i$ is the 3D coordinate of the i-th pixel in the image. (Note that non-negativity of the spatial components is enforced automatically here.) Due to the differentiability of $\boldsymbol{f}_{\boldsymbol{\beta}_t}$, it is straightforward to compute gradients with respect to $\boldsymbol{\beta}_t$ and $\boldsymbol{\Sigma}_k$.

3.2.0.3 Regularization: temporal continuity

To enforce smoothness of the temporal traces and motion trajectories in time we add a regularizer that penalizes discontinuities in the neural trajectories and signal coefficients. Specifically, we encourage the neural centers and signal coefficients at neighboring time points to be close. The regularizer for this purpose is:

$$\mathscr{R}_T(\boldsymbol{C},\boldsymbol{\beta}) = \lambda_\beta \sum_{t=0}^{T-1} \left\|\psi(\boldsymbol{\mu}_{1:K})\boldsymbol{\beta}_{t-1} - \psi(\boldsymbol{\mu}_{1:K})\boldsymbol{\beta}_t\right\|_F^2 \tag{3.4}$$

$$+ \lambda_C \sum_{t=0}^{T-1} \|\boldsymbol{C}_{t-1} - \boldsymbol{C}_t\|_F^2. \tag{3.5}$$

In this formulation $\psi(\boldsymbol{\mu}_{1:K})$ is the quadratic transformation of the canonical neural centers. When multiplied by $\boldsymbol{\beta}_{t-1}$ and $\boldsymbol{\beta}_t$ the result will be the neural centers at time $t-1$ and t respectively.

3.2.0.4 Regularization: Jacobian constraints for plausible deformations

The term $\boldsymbol{f}_{\boldsymbol{\beta}_t}$ induces a deformable transformation of the pixel correspondences between time t and the canonical representation $\boldsymbol{A}$. In order to constrain this transformation to yield physically realistic deformations that respect volumetric changes, we regularize the cost function using the determinant of the Jacobian of the transformation term to encourage the Jacobian to be close to 1 and prevent the deformation from contracting or expanding unrealistically. The Jacobian can be represented as:

$$\mathscr{J}_{\boldsymbol{\beta}}(x_1,x_2,x_3) \quad \text{with} \quad \mathscr{J}_{ij} = \frac{\partial(\boldsymbol{f}_{\boldsymbol{\beta}})_i}{\partial x_j}.$$

Using the Jacobian, the regularizer is:

$$\mathscr{R}_{\mathscr{J}}(\boldsymbol{\beta}) = \lambda_{\mathscr{J}} \sum_{t=1}^{T} \sum_{i=1}^{j} (\det \boldsymbol{J}_{\boldsymbol{\beta}_t}(x_i, y_i, z_i) - 1)^2, \tag{3.6}$$

where the Jacobian is evaluated on a grid where we want to ensure its proximity to one.

3.2.0.5 Optimization

All the variations of the dNMF cost function are optimized in the following way. To update $\boldsymbol{\beta}$ and $\boldsymbol{A}$ we use the `autograd` tool and `PyTorch` library to automatically compute gradients of the cost function and `Adam` optimizer to back-propagate the gradients. A forward pass of computation is evaluating the cost function with $\boldsymbol{\beta}_{1:T}$ and $\boldsymbol{A}$ (in the fully parametric case, or $\boldsymbol{\beta}_{1:T}$ (in the Gaussian case) as parameters. Note that for a fixed $\boldsymbol{C}$, all compartments of the cost function are differentiable with respect to the parameters.

To update $\boldsymbol{C}$ we use multiplicative updates as described in [Taslaman and Nilsson, 2012]:

$$\boldsymbol{C}_t \leftarrow \boldsymbol{C}_t \odot \frac{\boldsymbol{f}_{\boldsymbol{\beta}_t}^T \boldsymbol{Y}_t + \lambda_C(\boldsymbol{C}_{t-1} + \boldsymbol{C}_{t+1})}{\boldsymbol{f}_{\boldsymbol{\beta}_t}^T \boldsymbol{f}_{\boldsymbol{\beta}_t} \boldsymbol{C}_t + 2\lambda_C \boldsymbol{C}_t}. \tag{3.7}$$

The key difference between these multiplicative updates from those found in [Lee and Seung, 2001] is that the parts of the derivatives of the temporal smoothness regularization terms $2\lambda_C \boldsymbol{C}_t$ and $\lambda_C(\boldsymbol{C}_{t-1} + \boldsymbol{C}_{t+1})$ appear in the denominator and numerator to promote smoothly varying signal.

3.2.0.6 Initialization

One key advantage of the *C. elegans* datasets considered here is that we can reliably identify the locations of all cells in the field of view, using methods developed in [Yemini et al., 2019b]. Using the location of cells in the initial frame (for example) can tremendously aid the optimization of the objective 3.1 for two main reasons. First, it serves as a very good initializer for the μ_k parameters for cell spatial footprints mentioned in subsection 3.2.0.2. Second, we know a priori the correct number of cells to be demixed in the FOV. These two factors enable our framework to operate in a **semi-blind** manner towards the deconvolution of neural signals of *C. elegans*, unlike fully blind deconvolution techniques such as e.g. PCA-ICA [Mukamel et al., 2009] or CNMF [Pnevmatikakis et al., 2016].

3.2.0.7 Using dNMF for image registration

The transformation terms $\boldsymbol{f}_{\boldsymbol{\beta}_t}$, learned using dNMF, can be used to obtain a pixel-level transformation of the video frames to a reference frame in order to yield a registered video; in the ideal case this registered video would remove all the motion from the video, leaving each neuron to flicker in place as its internal activity modulates its fluorescence level. In the current formulation, $\boldsymbol{f}_{\boldsymbol{\beta}_t}$ represent push-forward mappings of a reference frame to all the frames in the video. However, to obtain a registration, we need to recover the inverse mappings from all of the video frames to the reference frame. We solve this inverse transform, $\boldsymbol{\beta}_t^i$, by optimizing the following objective:

$$\min_{\boldsymbol{\beta}_t^i} \sum_{t=1}^{T} \|\boldsymbol{\mu}_{1:K} - \boldsymbol{\beta}_t^i \boldsymbol{\psi}(\boldsymbol{\beta}_t^* \boldsymbol{\psi}(\boldsymbol{\mu}_{1:K}))\|_F^2 + \mathscr{R}_{\mathscr{J}}(\boldsymbol{\beta}^{\boldsymbol{i}}) \tag{3.8}$$

where $\boldsymbol{\mu}_{1:K} \in \mathbb{Z}_+^{K\times 3}$ indicates the set of neuron coordinates at the reference frame and $\boldsymbol{\beta}_t^* \boldsymbol{\psi}(\boldsymbol{\mu}_{1:K})$ indicates the forward polynomial mapping of these neurons in the t-th video frame after optimization, with $\boldsymbol{\beta}_t^*$ indicating the transformation optimized through (3.1). Lastly, $\mathscr{R}_{\mathscr{J}}(\boldsymbol{\beta})$ indicates the same Jacobian regularizer as in (3.6). In the simplest case that $\boldsymbol{\beta}_t^* \boldsymbol{\psi}$ is restricted to be affine, and the regularizer weight λ_J in $\mathscr{R}_{\mathscr{J}}(\boldsymbol{\beta}_{\boldsymbol{t}}^{\boldsymbol{i}})$ is negligible, then $\boldsymbol{\beta}_t^i \boldsymbol{\psi}$ simply implements the shift and matrix inversion of $\boldsymbol{\beta}_t^* \boldsymbol{\psi}$. More generally, the exact inverse mapping may not exist or may be unstable; in this more general setting

(3.8) will output a smooth approximation to the inverse mapping.

Note that Eq. (3.8) solves a labeled point-set registration problem (since it operates on the neuron centers $\boldsymbol{\mu}_{1:K}$), not an image registration problem per se. Next we use the recovered inverse mapping $\boldsymbol{\beta}_t^i$ to perform image registration, using pixel-wise interpolation:

$$\boldsymbol{p}_t \mapsto \text{Interp.}[\boldsymbol{\beta}_t^i \boldsymbol{\psi}(\boldsymbol{p}_t)]. \tag{3.9}$$

Here, $\boldsymbol{p}_t \in \mathbb{Z}_+^{d\times 3}$ denotes the mesh of pixel coordinates that span the entire volume of the image and Interp. refers to an interpolation function such as linear, nearest neighbor, or bicubic, that can be used to convert non-integer values of pixel coordinates to map to discrete pixels. In practice, we set the reference frame to be the first frame in the video series and use linear interpolation. Note that this way of performing registration differs from traditional registration techniques such as Normcorre [Pnevmatikakis and Giovannucci, 2017] in a critical way: the deformation terms that are used to drive the registration are informed by the neural activity and are decoupled from the inferred activity in the joint objective function (3.1). Thus, in theory, large fluctuations in neural activity from frame to frame should not affect the deformation terms. In contrast, pure registration techniques on functional neural data may be driven to poor local optima if the neural activity in a particular frame differs strongly from the reference frame.

3.2.0.8 Population neural analysis

After we have extracted activity traces from each neuron in a single field of view, a typical next step is to compile and analyze a collection of extracted traces across multiple imaged animals. The traces exhibit variability due to both methodological variability (e.g., variability inherent in imaging equipment) and biological variability (e.g., variability inherent in the levels of fluorescent-protein expression across neurons of the same type). These “extra" sources of variability can obscure the changes in neural activity that we wish to extract and analyze here. Consequently, a neuron's calcium trace, measured across multiple animals, can exhibit differences in overall intensity that require correction to obtain valid comparisons across animals. As a simple example, many neuron classes are composed of a symmetric left and right pair that often show identical calcium activity. With most imaging equipment, when the left neuron is near the lens, the corresponding right neuron is far away, leading to a false differential reading of brightness. Thus, even within a single animal, symmetric neurons can require corrections to be comparable.

The commonly used technique of converting neural traces to $\Delta F/F_0$ aims to correct these issues in mismatched fluorescence intensity profiles but is often insufficient (see Results subsection below and Figure 3.2). One way to further normalize time-series data is through z-scoring the signal such that the mean and variance across time is zero and one, respectively. However, in practice, simply mean-shifting to zero often misrepresents the neuron's baseline signal. Similarly, scaling to unit variance will scale unresponsive and responsive neurons to the same magnitude, thus inflating instead of suppressing measurement noise in unresponsive cells.

A method that employs a more robust view of the distribution of neural signal would provide a more accurate normalization. Here we generalize the concept of z-scoring time-series by first observing that z-scoring is a linear transform that matches the histogram of the time series to a standard Gaussian distribution with zero mean and unit variance. We then cast histogram normalization in a way such that the transformation is constrained to be a linear transform that minimizes distance to the distribution as a whole, leading to more robust results compared to z-scoring, which restricts attention to two non-robust summary statistics of the histogram (the mean and variance). Lastly, we provide a strategy for normalizing a population of time series data by transforming to the *medoid* of these time series (i.e., the time series which is on average closest to all the others in the population). Empirically, the resulting approach preserves signal while reducing variability across the population.

Quantile regression Let $\boldsymbol{C}^i \in \mathbb{R}^T$ denote the time series of a neuron in the ith animal over T time steps. Suppose we want to match the neural time-series of the ith animal to the jth animal using a linear transform. One possible strategy to match two time-series signals to one another is to match their baselines and match their peaks. This corresponds to transforming the minimum and the maximum of one time series such that they match the minimum and maximum of the other time series. This is equivalent to matching their minimal and maximal quantiles through a transformation term involving scaling and shifting.

We can generalize this procedure with more quantiles to yield a transformation estimate that is more robust to noise. Matching multiple quantiles using a linear transformation term can be represented by the following linear model:

$$F^{-1}_{\boldsymbol{C}^j}(a) = F^{-1}_{\boldsymbol{C}^i}(a)v + v_0 + \epsilon \tag{3.10}$$

where F^{-1} denotes the inverse cumulative distribution function and v, v_0 denote the scaling and magnitude shift of the time-series, respectively, and ϵ represents an error term. This model posits that each time series signal consists of a baseline and several peaks which can be represented as quantiles of histograms that require matching; baselines and peaks of the same neuron, across different animals, should roughly have similar values.

We can then estimate v and v_0 by solving the following least squares problem:

$$W_{2,L}(\boldsymbol{C}^i, \boldsymbol{C}^j) = \min_{v, v_0} \int_0^1 \|F^{-1}_{\boldsymbol{C}^j}(a) - F^{-1}_{\boldsymbol{C}^i}(a)v - v_0\|_2^2 da. \tag{3.11}$$

The arg-min of (3.11) yields the linear estimates v, v_0 that can be used to transform the time series $\boldsymbol{C}^i$ to match the time series $\boldsymbol{C}^j$, where $\hat{\boldsymbol{C}}^{i,j} = \boldsymbol{C}^i v^* + v_0^*$. We term this regression model, quantile regression (QR), since the predictors and responses are quantiles of time-series data. If only two quantiles are used i.e. the bottom and topmost quantiles, this procedure is equivalent to matching the minimum/maximum of the two time-series.

Optimizing (3.11) yields the transformation that best matches the histogram of the ith time series $\boldsymbol{C}^i$ with that of the jth time series $\boldsymbol{C}^j$. The residual discrepancy between the transformed ith time series $\hat{\boldsymbol{C}}^{i,j}$ and the jth time series $\boldsymbol{C}^j$ can be thought of a distance between these time series. In fact, the minimum of (3.11) is a linear approximation of a *bona fide* distance metric, termed the Wasserstein metric [Peyré et al., 2019], that is a distance between probability distributions.

Using this notion of proximity between time series, if we have a population of N samples $\boldsymbol{C}^1,\ldots,\boldsymbol{C}^N$, the strategy for normalizing the time series we advocate here is to compute pairwise Wasserstein distance approximations between all time series and choose the medoid time series to normalize to:

$$\boldsymbol{C}^0 \leftarrow \arg\min_{\boldsymbol{C}^\ell} \sum_{i=1}^{N} W_{2,L}(\boldsymbol{C}^\ell, \boldsymbol{C}^i) \tag{3.12}$$

In other words, we can find the best fit of each time series through quantile regression to all other time series, and set as a reference the time series that has the minimal average distance to all the other time series. Once the reference is set, all the samples are transformed to match the reference quantiles using (3.11). See Figure 3.2 for an illustration.

Lastly, if the time series all capture non-negative signal (as is often encountered in calcium imaging) the regression in (3.11) can be constrained to be non-negative to ensure the transformed time series maintains its positivity. This yields the non-negative linear estimate of the Wasserstein metric. We term this variant of the quantile regression model as non-negative quantile regression (NQR):

$$W_{2,N}(\boldsymbol{C}^i, \boldsymbol{C}^j) = \min_{v, v_0 \geq 0} \int_0^1 \| F_{\boldsymbol{C}^j}^{-1}(a) - F_{\boldsymbol{C}^i}^{-1}(a) v - v_0 \|_2^2 da \tag{3.13}$$

3.2.0.9 Evaluation metrics

To evaluate the performance of the proposed method as well as the compared methods, we focus on several metrics that shed light both on the signal demixing capabilities of the methods as well as their ability to track objects in time. Namely we focus on two major metrics: **trajectory correlation**, which measures the ability of the deformation model to keep track of the observed motion, and **signal correlation**, which measures the demixing performance by comparing the correlation of demixed signal intensities relative to the ground truth. Specifically, these metrics can be expressed as

Trajectory correlation:

$$\rho(\hat{\boldsymbol{\beta}}, \boldsymbol{\beta}) = \frac{\sum_{i,j,t} (\hat{\beta}_t^{ij} - \bar{\hat{\beta}})(\beta_t^{ij} - \bar{\beta})}{\sqrt{\sum_{i,j,t} (\hat{\beta}_t^{ij} - \bar{\hat{\beta}})^2} \sqrt{\sum_{i,j,t} (\beta_t^{ij} - \bar{\beta})^2}}$$

Signal correlation:

$$\rho(\hat{\boldsymbol{C}}, \boldsymbol{C}) = \frac{\sum_{kt} (\hat{C}_{kt} - \bar{\hat{C}})(C_{kt} - \bar{C})}{\sqrt{\sum_{kt} (\hat{C}_{kt} - \bar{\hat{C}})^2} \sqrt{\sum_{kt} (C_{kt} - \bar{C})^2}}.$$

The above metrics are applicable when the ground truth motion trajectories and the signal coefficients are known. To evaluate the performance of the methods using unsupervised registration heuristics, we focus on the **correlation of registered frames** to the average frame (after registration). Heuristically, this measure has been demonstrated to be an effective indicator of successful registration [Pnevmatikakis and Giovannucci, 2017]. Furthermore, in the real data experiments, we also evaluate the average **spread of cell locations** before and after registration. This is computed by taking the average distance of the cells to the average cell location. Similar to the frame correlation measure,

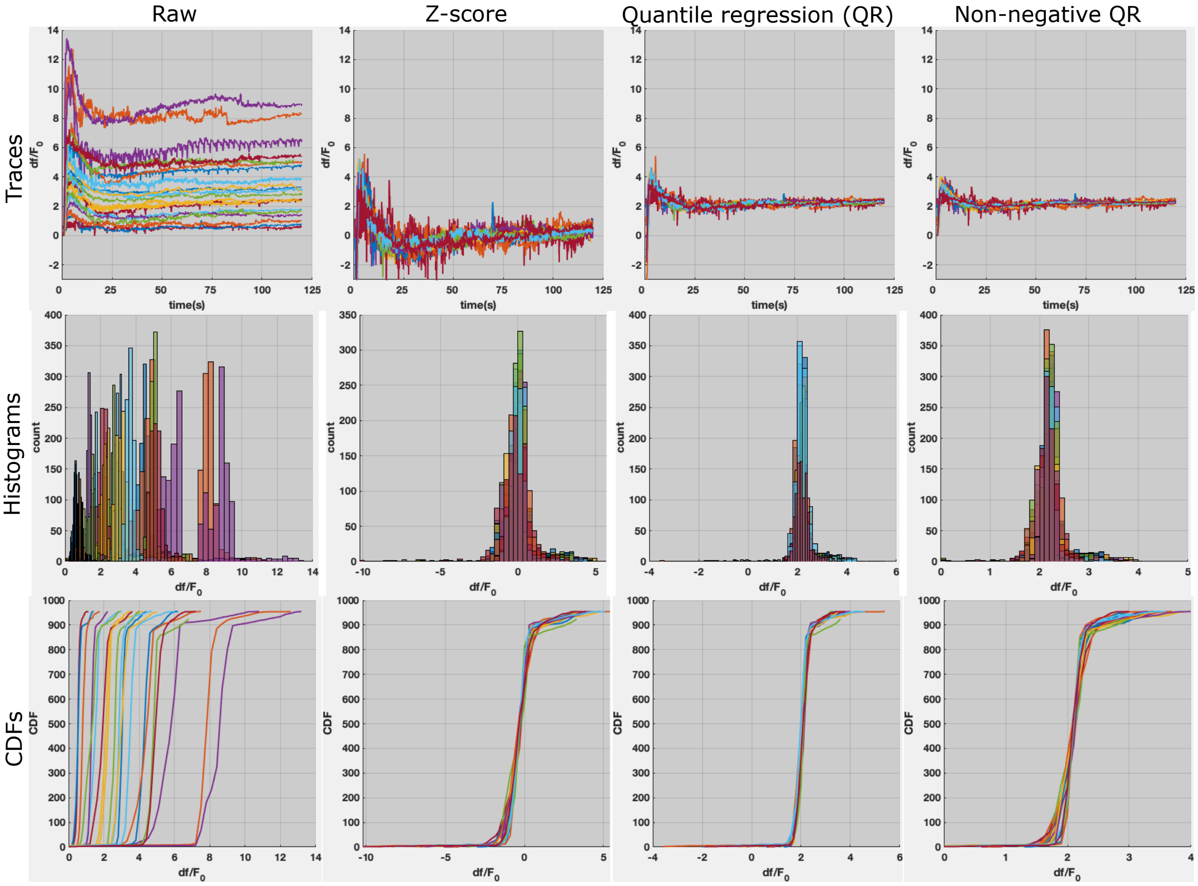

Figure 3.2: **A demonstration of quantile regression for the tail neuron LUAL in *C. elegans*, across 21 animals** Different colors indicate different animals. First column: The raw traces superimposed exhibit variability in intensity profiles due to imaging and biological differences (top). The histograms and cumulative distribution functions (CDFs) of the time-series signals display the differing distributions representing these traces (middle and bottom). **Second column:** Z-scored traces exhibit tighter grouping than raw traces (top) further shown in their CDFs (bottom). However, these z-scored traces are shifted towards zero mean (top) which is misrepresentative of the signal magnitude and also exhibit significant remaining variability. **Third and fourth columns:** After quantile regression (QR) and non-negative quantile regression (NQR) to the medoid of the traces, we see that the normalized traces retain their shape (top), exhibiting even tighter grouping than after z-scoring. In comparison with the z-scored traces, both QR and NQR preserve the median signal magnitude, $\Delta F/F_0$=~2 (top and middle) with smaller tails in their histogram (middle), implying a better fit across the population of animals.

this metric allows us to diagnose whether certain cells are registered better than others. While the average correlation of frames is a high-level measure of registration performance, the measure of cellular spread is a localized metric, indicating whether certain regions of the volume are registered better than others. In other words, the former metric measures global sharpness while the latter measures local sharpness.

The added benefit of the latter two evaluation metrics is that since they do not require any ground truth, they can be used for hyperparameter selection; i.e., we can select regularization parameters that yield the sharpest registration results. Furthermore, we can use the sharpness criteria to evaluate the goodness of fit for different deformation models such as quadratic polynomials (as used here), b-splines [Rueckert et al., 1999], or higher order polynomials.

3.2.0.10 Compared methods

We argue in this paper that jointly optimizing for deformable registration and time-series signal extraction has the potential to improve the quality of both the registration and signal extraction. Therefore, we compare the registration performance of dNMF against the state-of-the-art method for calcium video motion registration, named Normcorre [Pnevmatikakis and Giovannucci, 2017]. Normcorre does not explicitly model the presence of independent signal carrying units in the FOV and instead performs piecewise-rigid transformations on overlapping sub-blocks of the volume using a fast fourier transform based technique [Guizar-Sicairos et al., 2008]. Furthermore, Normcorre uses a normalized cross-correlation registration loss function that is less prone to intensity variations across time-frames.

Next, we also evaluate the signal extraction performance of dNMF against two standard routines in calcium imaging. First, we compare against region of interest (ROI) tracking and pixel averaging within the ROI [Venkatachalam et al., 2016b]. This method tracks the positions of cells across time and extracts signal by taking the average pixel intensity value in a pre-defined radial region around the tracking marker. We also compare against the routine of performing motion correction first and then signal extraction through NMF [Pnevmatikakis et al., 2016]. To replicate this routine in our experiments, we motion correct using Normcorre and then use the Gaussian cell shape parametrization version of NMF that is described in subsection 3.2.0.2. We use this variant of NMF rather than non-parametric variants such as CNMF [Pnevmatikakis et al., 2016] to bring the comparison against dNMF to an equal footing since dNMF already uses this parametrization that tends to model nuclear shapes well.

3.2.0.11 Implementation details

All the optimization codes are implemented in `Python 3.7.3` using the `autograd` tool and the `PyTorch 1.5` package. We used the `Adam` optimizer with learning rate 0.001 for the simulations and 0.00001 for the worm experiments. Large learning rates lead to jumps in the tracks and lower quality traces, while small learning rates need more iterations to converge. The experiments are run on a Lenovo X1 laptop with Microsoft Windows operating system using 64 GB RAM and Intel(R) Core(TM) i7-8850H CPU @ 2.60GHz, 2592 Mhz, 6 Core(s), 12 Logical Processor(s). We further implemented a sequential optimizer for the demixing of an online stream of videos where each batch of data consisting of a few time frames of the video is processed with parameters initialized using the previous batch. In addition, to improve the memory and time efficiency of our algorithms we also introduced a stochastic variant of dNMF, where in each iteration, to compute the loss and its gradient, we randomly subsample the pixels both in the spatial and temporal domains and update the parameters based on those samples.

3.2.0.12 *C. elegans* video description

Videos of calcium activity in *C. elegans* were captured via a spinning-disk confocal microscope with resolution (x,y,z)=(0.27,0.27,1.5) microns. Whole-brain calcium activity was measured using the fluorescent sensor GCaMP6s in animals expressing a stereotyped fluorescent color map that permitted class-type identification of every neuron in the worm's brain (NeuroPAL strain OH16230) [Yemini et al., 2019b]. Each video was 4 minutes long and was acquired at approximately 4Hz. Worms were paralytically immobilized (using tertramisole) in a microfluidic chip capable of delivering chemosensory stimuli (salt and two odors) [Chronis et al., 2007; Si et al., 2019]. This setup allows for the controlled delivery of multiple soluble stimuli to the animal with high-temporal precision. See [Yemini et al., 2019b] for full experimental details.

Despite paralytic immobilization, we still observed some motion of the worm within the chip, primarily over small distances of several microns and over slow, multi-second time scales. Some of this motion was driven by the animal, while some was the result of the animal drifting passively due to minute pressure differences in the chip. This motion was strongest in the tail, which, due to its taper, was not well secured by the channel walls of the microfluidic chip. Despite the smaller scale of this motion (as compared to freely-moving behavior such as crawling), motion artifacts could strongly confound traces, particularly in the head of the animal where the neurons are very tightly packed. Thus, these motion artifacts required algorithmic correction.

Each dataset from this collection is a video in the form of a 4D tensor $W \times H \times D \times T$ (approximately $256 \times 128 \times 21 \times 960$) where the value of the tensor at (x, y, z, t) corresponds to the activity of a neuron located near the point (x, y, z) at time t. To extract the neural activity from the videos we first reformat the data into a $d \times T$ matrix where $d = WHD$ that is called the data matrix $\boldsymbol{Y}$. We then run the Gaussian dNMF with cell centers initialized using the cell locations in the initial frame, determined using the semi-automated methods described in [Yemini et al., 2019b]. Since the cells are approximately spherical in this video we used a fixed spherical covariance matrix for all the cells with squared root diagonal entries equal to 0.57μm (roughly a third of the minimal diameter of adult worm neurons).

3.2.0.13 Pre-processing steps

Neuron centers were first tracked using a local image registration approach throughout the time series, using the approach in [Venkatachalam et al., 2016b]. After identifying each neuron center in the first frame, every subsequent frame was registered to this first frame. The registration was performed on x-, y-, and z- maximum-intensity projections of a small volume around the neuron center using the `imregister` function in MATLAB. The volume was chosen to be small enough that nonrigid deformations could generally be neglected, so we used a rigid registration model (translation and rotation only). Because motion is continuous between frames, the initial guess for the transformation was taken to be the calculated transformation from the previous time frame.

We use the initial trajectories of the neurons to initialize our motion parameters $\boldsymbol{\beta}_t$ by solving $\boldsymbol{\beta}_t = \arg\min_\beta \left\|\boldsymbol{\beta}\psi(P_1) - P_t\right\|_2^2$ where P_t contains the locations of neurons in time t tracked using local image registration techniques. For computational efficiency, we also mask out pixels that are outside of the circles with radii $3\mu m$ from the location of all neurons in all time points.

3.2.1 Results

3.2.1.1 Simulation experiments

To evaluate the effectiveness of our algorithm in capturing motion and demixing time-series traces, we simulated the trajectory of 10 neurons, with a time-specific trace

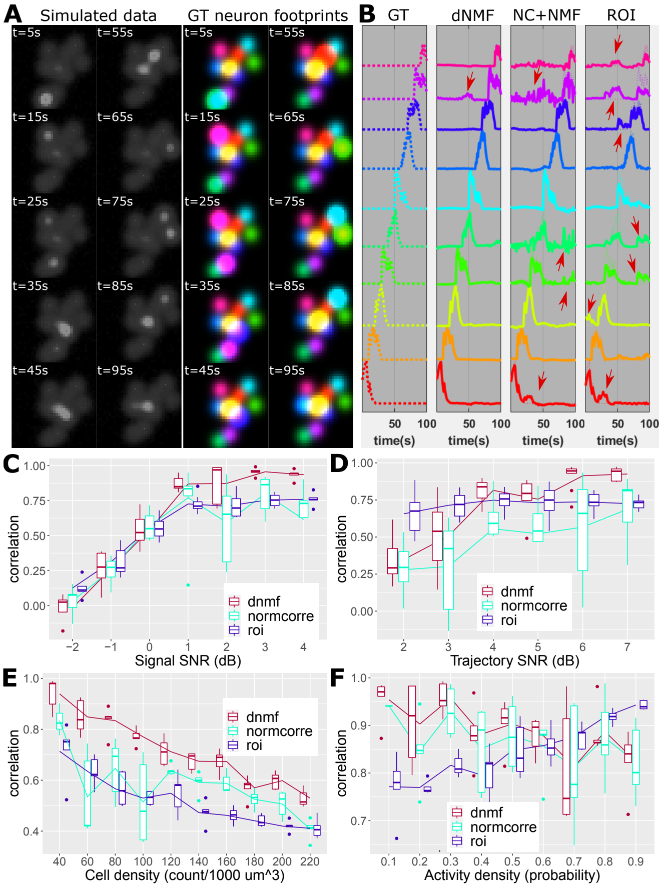

Figure 3.3: **Demixing calcium signals in simulated videos A:** Neurons are generated as Gaussian shapes and undergo motion and simulated calcium activity in a 100-second long video. Static snapshots of the video are shown (left) and spatial footprints for each cell are assigned unique colors with intensities proportional to calcium activity (right). Note that the spatial footprints of cells are also in motion, tracking the position of the cells.

assigned to each (Fig. 3.3A-B). The signal for each neuron is modeled as a binary vector with length T and probability p of observing a unit spike, convolved with a decaying exponential kernel. Each trajectory was generated using quadratic transformations of the point cloud in its previous time point, starting from a random initial point cloud. (Note that the composition of many such quadratic mappings is non-quadratic, and therefore the generative model here does not perfectly match the model dNMF uses to fit the data, where a quadratic transformation maps the spatial components A to match the observed data at each frame; nonetheless, despite this model mismatch, dNMF

Figure 3.3: (cont. from previous page) **B:** The ground truth calcium activity for each cell (left) is compared with the neural activity extracted using dNMF (second column), Normcorre [Pnevmatikakis and Giovannucci, 2017]+NMF (third column) and ROI tracking and pixel averaging (fourth column). dNMF recovers the ground truth signal well whereas Normcorre+NMF and ROI methods yield significantly more mixed signals (indicated by red arrows) due to the proximity of the cells and the tendency of the spatial footprints of mobile cells to overlap. **C:** The correlation of the recovered signal to the ground truth signal as a function of the image signal-to-noise ratio (SNR). **D:** The correlation of the recovered cell movement trajectories to the ground truth trajectories as a function of trajectory SNR. **E:** The correlation of the recovered signals to the ground truth as a function of the density of independent objects in the FOV. **F:** The correlation of the recovered signals to the ground truth as a function of the density of signaling events (simulating neural excitation) exhibited by the cells. Note that we provided ROI tracking here with access to the ground truth cell centers at all times (explaining why ROI averaging correlation values remain high even in the limit of very high activity density); nonetheless, even with artificially perfect tracking accuracy, mixing of nearby signals remains a significant issue. See MOVIE LINK for further details.

achieves accurate results here, as discussed below.) The trajectory of each neuron was then convolved with a fixed 3D Gaussian filter that represents the shape of that neuron and then multiplied with the time course assigned to that neuron. The simulated video is the result of the superposition of these moving Gaussian functions.

We compare the performance of dNMF, Normcorre+NMF, and ROI pixel averaging in a variety of confounding scenarios using the metrics defined in subsection 3.2.0.9. In all simulation experiments, the ROI averaging method is provided with the ground truth cell positions — i.e., we examine the accuracy of this method under the (unrealistically optimistic) assumption that neurons are tracked perfectly, to evaluate the demixing performance of ROI signal extraction without the additional confound of tracking performance.

In Fig. 3.3C-F, we explore the performance limits of dNMF, Normcorre+NMF, and ROI pixel averaging as a function of imaging noise and motion variability. Signal SNR is defined by the peak-to-trough difference between the neural activity signals during

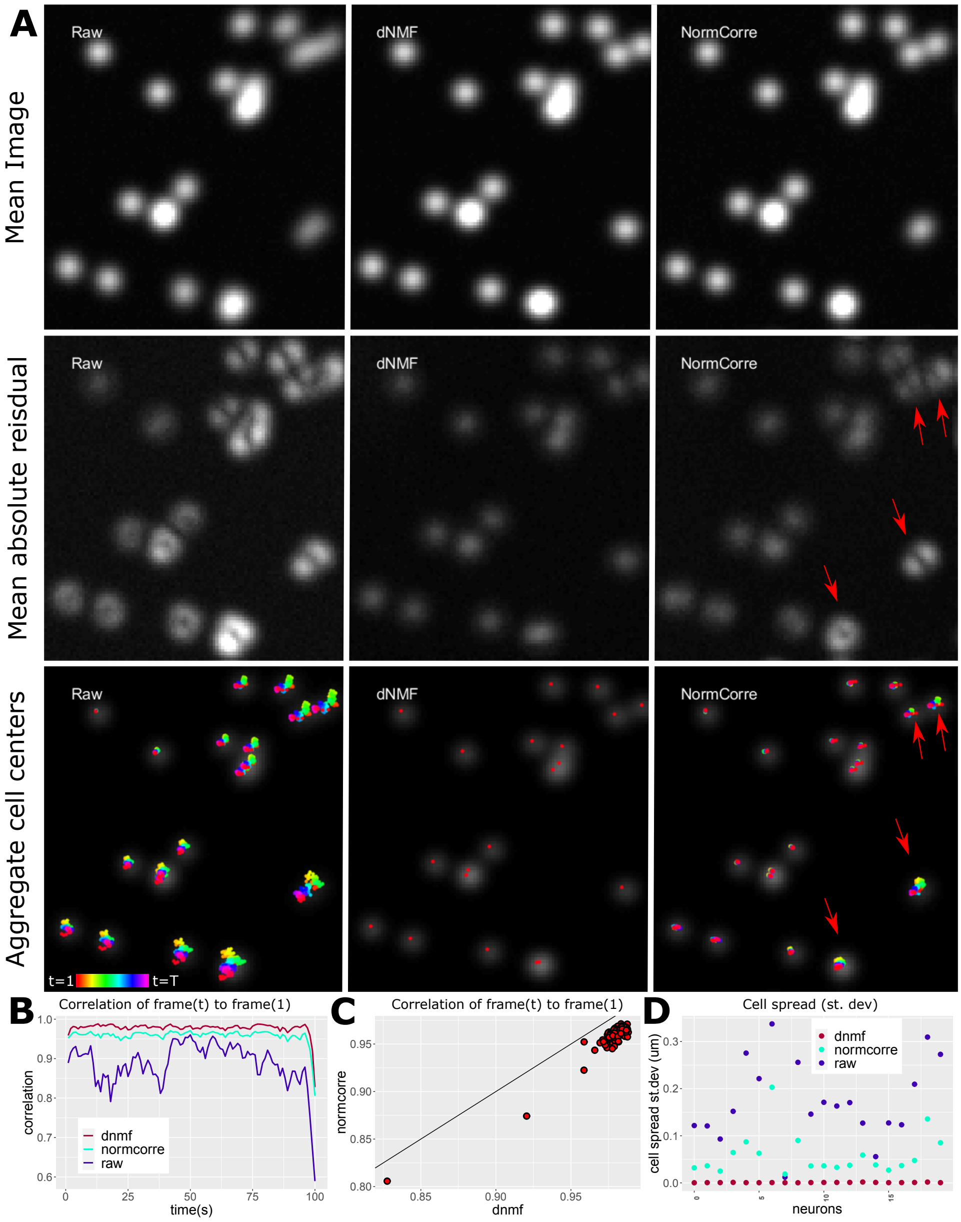

Figure 3.4: **Simulated data registration results A: Top row:** The mean video frame prior to registration (left), after dNMF based registration (middle), and after Normcorre [Pnevmatikakis and Giovannucci, 2017] registration. **Middle Row:** Mean of the absolute value of the video frames subtracted from the first frame prior to registration (left), after dNMF based registration (middle) and after Normcorre (right). If registration is perfect, this image will look like a weighted sum of Gaussian shapes, one for each cell (corresponding to the cell dimming and brightening, but remaining in place); imperfect registrations are indicated by “spreading" or “doubling" of the cell shapes, as indicated by red arrows.

Figure 3.4: (cont. from previous page) **Bottom row:** The locations of the cells across time (colors denote different times) superimposed on the first frame prior to registration (left), after dNMF based registration (middle) and after Normcorre (right). Red arrows indicate cells with imperfect registration, with significant remaining movement of the cells across frames. **B:** The correlation of the video frames to the mean video frame before registration (blue), after dNMF based registration (red), and after Normcorre (cyan); higher values indicate better performance here. **C:** The correlation of individual registered frames to the mean video frame after dNMF registration (x-values) and Normcorre (y-values). The straight line indicates $x = y$; points below this line indicate the higher correlation of dNMF registered frames to the mean frame. **D:** The spread of the cell position centers, relative to their average. in the unregistered video (blue), after dNMF-based registration (red), and after Normcorre (cyan). A lower standard deviation for cell spread indicates better performance for local registration of cell shapes. See MOVIE LINK for further details.

times of activity. Trajectory SNR is quantified by how well the cells adhere to the motion of all other cells; high trajectory SNR indicates all cells move in unison, resembling a deformable medium, and low trajectory SNR indicates each cell is moving like independent particles. Mathematically, this is proportional to the log ratio of the variance of the average location of the cells versus the variance of the time differences of these locations. It can be seen in Fig. 3.3D that dNMF is robust to noise but ultimately may introduce errors to demixing and trajectory tracking if the signal and trajectory SNR (Fig. 3.3D) are too low. Normcorre+NMF does relatively worse than dNMF as a function of signal SNR and trajectory SNR. ROI pixel averaging has the poorest signal recovery performance of the three compared methods as a function of signal SNR. (Note that ROI pixel averaging enjoys a constant trajectory estimation rate in Fig. 3.3D, since it has access to ground truth cell locations, as discussed above.)

Next, we evaluated the signal extraction performance as a function of the cell density in the FOV. Increased cell density indicates an increased superpositioning of independent signals and therefore a higher degree of signal mixing. dNMF demixing performance degrades linearly as the density of independent objects within the FOV increases (Fig. 3.3E) but enjoys higher rates of recovery than both Normcorre+NMF and ROI pixel averaging.

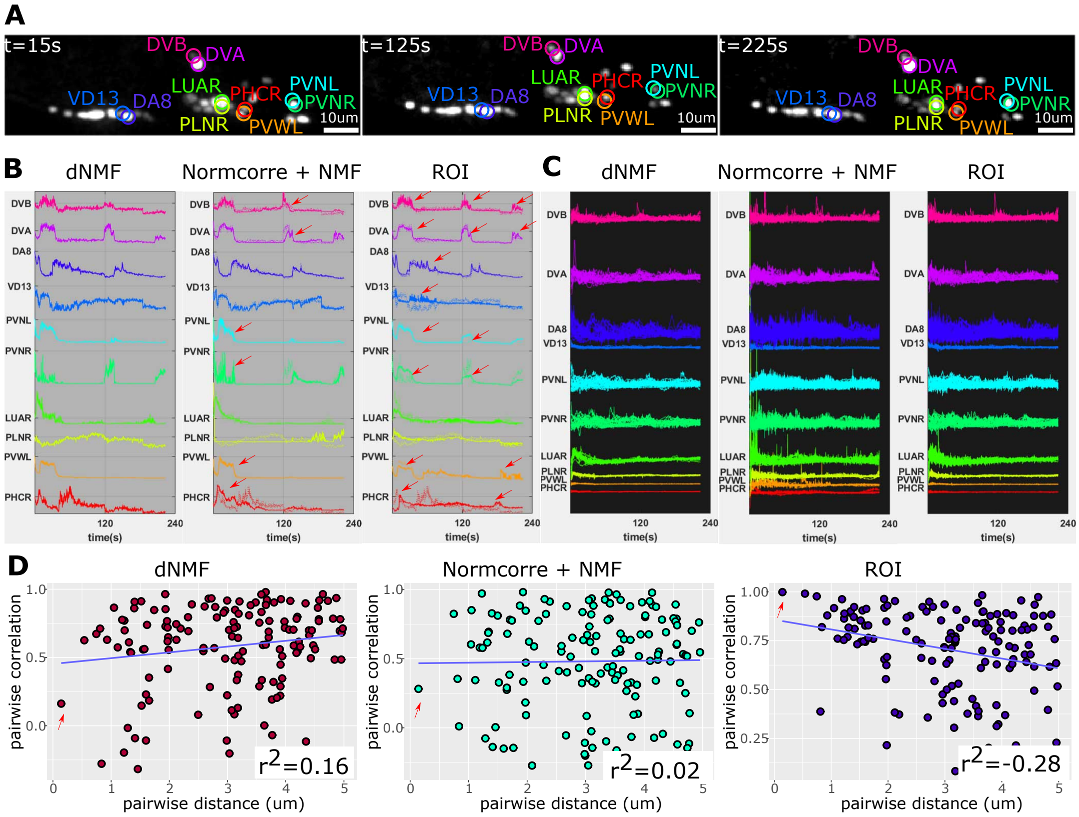

Figure 3.5: **Demixing neural calcium signal in semi-immobilized *C. elegans* videos A:** Three static, z-axis maximum projected frames from a representative 4-minute long video of GCaMP6s neural activity. We focus on the signal from five pairs of spatially-neighboring neurons in the tail: DVA/DVB, PVNR/PVNL, PVWL/PHCR, PLNR/LUAR, and VD13/DA8.

Lastly, we observe that the density of signaling events changes the demixing performance for the three compared methods. In particular, low signal densities (simulating weak excitation) make it harder to track individual cells, which may be dim and therefore hard to detect and track in many frames.

In Fig. 3.4, we qualitatively demonstrate the registration performance of dNMF versus Normcorre [Pnevmatikakis and Giovannucci, 2017]. We see that the average frame, after registering with dNMF, is sharper than the non-registered average frame, with better-localized and less-variable cell center locations. In comparison, Normcorre yields a higher spread of cells, even after registration, which may lead to erroneous signal

Figure 3.5: (cont. from previous page) **B:** Calcium signals extracted by dNMF (left), Normcorre [Pnevmatikakis and Giovannucci, 2017] + NMF (middle), and ROI tracking and averaging (right). dNMF extracts uncoupled signals that demonstrate independent neural activity. The selected cells were chosen such that the signal recovered by ROI averaging is inconsistent with dNMF (quantified by having correlation smaller than 0.4). Normcorre + NMF partially mixes signals between both PHCR/PVWL and PVNL/PVNR around the 30-second mark and DVB/DVA around the 120-second mark (red arrows), and loses nearly all signal from PLNR, due to motion exhibited by the semi-immobilized animal. ROI averaging produces completely correlated signal (red arrows) between all of the labeled neurons, and loses most of the signal from LUAR and PLNR, due to overlap in their spatial footprints. **C:** Calcium activity traces, of the labeled tail neurons, extracted from a population of 21 worms. The unique colors label traces from the same neurons, across different animals. Here, the dNMF traces are tightly grouped, exhibiting minimal variability between animals. Normcorre+NMF traces exhibit mixed-signal and mistracked neurons. ROI traces exhibit wider variability than dNMF, due to mixed signals and, potentially, noise common to ROI averaging. **D:** Pairwise neuron distances versus pairwise correlation of neural signals for all three methods. Note that signal mixing tends to occur when the signal sources are close to one another, necessitating techniques such as NMF to disentangle independent signals. For this reason, dNMF is well suited to demix spatially-close neuron pairs. Normcorre+NMF experiences mixing effects due to motion for which it fails to account (seen in the supplementary movie linked below). ROI averaging does mix traces and thus shows increasingly correlated signals between neuron pairs as they get nearer to each other (indicated by the red arrow). See MOVIE LINK for further details.

recovery. Both of these global and local sharpness metrics are quantified in Fig. 3.4C-D.

3.2.1.2 Demonstration of demixing in real *C. elegans* data

In the simulated data analyzed above, dNMF exhibits superior registration performance due to its ability to decouple the intensity signal from the motion of objects. Conversely, coupling registration with signal extraction enables dNMF to capture the neural signal and demix it from nearby cells more accurately.

We extend this demonstration further with a real data example. The worm's tail contains several ganglia, with densely-packed neurons, whose spatial footprints often overlap due to insufficient spatial resolution. Additionally, even neurons in separate ganglia can end up in sufficient proximity, due to microfluidic confinement or other imaging-setup induced deformations, such that their spatial footprints overlap. The spatial overlap represents a significant challenge, both for tracking individual neurons and demixing their signal. Figure 3.5 shows an example of the difficulty present when tracking and demixing neural activity signals from animals with spatially overlapping neural footprints in their recorded images. In this example, ROI tracking loses most of the signal from the LUAR and PLNR neurons and further mixes signals between the DA8/VD13, DVA/DVB, PHCR/PVWL, and PVNL/PVNR neurons. Normcorre+NMF performs better but loses nearly all signals from PLNR while also still mixing signals between the DVA/DVB, PHCR/PVWL, and PVNL/PVNR neurons. In comparison, dNMF recovers strong, independent signals from all ten neurons. Thus dNMF can track and differentiate signals from neurons, even within areas containing multiple spatially-overlapping neural footprints where other comparable algorithms fail.

Additionally, in figure 3.5, we quantify the demixing performance by computing the pairwise correlation of nearby neurons as a function of the distance between these neurons. Signal mixing is expected to occur when the spatial footprints of nearby neurons, blurred by the point spread function and/or insufficient spatial resolution, overlap with another. Therefore, one heuristic to determine how well demixing was performed is the correlation of pairwise distances of neurons to the pairwise correlations of their activity. Indeed, both matrix factorization methods, dNMF and Normcorre+NMF, yield uncorrelated trends between neuron pair distance and the correlation of their respective trace activity. On the other hand, simple ROI averaging tends to visibly mix signals in closely-neighboring neurons, resulting in unrealistically high correlation values near 1 for the closest neighbors.

3.2.1.3 Worm registration

After optimizing dNMF, we can obtain registered videos of worms to evaluate performance and compare against Normcorre (Figure 3.6A). Similar to simulated data, we can once again observe that the mean video frame after registering with dNMF is sharper when compared to both the raw average frame and this average after applying Normcorre. Furthermore, the mean of the absolute difference between video frames and the first frame shows that the dNMF registration has fewer distorted toroidal shapes than Normcorre, indicating better registration of cell shapes. Lastly, we can also see that after registering with dNMF the cell centers have a tighter grouping than Normcorre; this is another indication of better registration performance.

Figure 3.6B-D evaluates these observations quantitatively. The subfigures B and C indicate that the frames registered via dNMF tend to have a higher correlation to the mean registered frame than Normcorre, which indicates the quality of registration. Furthermore, the dNMF and Normcorre registration results diverge most in the initial frames, where the majority of deformable motions are observed. Since Normcorre is a piecewise rigid registration technique, its deformation model may be misspecified to capture such motions, whereas the dNMF motion model is more accurate. Figure 3.6D demonstrates that the cell grouping after dNMF is indeed tighter than Normcorre.

3.2.1.4 Population study of *C. elegans*

Using the neural traces extracted with dNMF (converted to $\Delta F/F_0$), we demonstrate the time-series histogram-normalization technique of quantile regression (QR), show its non-negative regression variant (NQR), and compare these with z-score normalization. The time-series data we used is the brain-wide neural GCaMP6s intensity extracted from 21 worm heads (up to 189 neurons in each head) and 21 worm tails (up to 42 neurons in each tail). In these animals, neurons with the same identity often exhibited very different intensity distributions across individual animals. In the course of a time series, neuron intensities change to reflect the underlying activities but, given a sufficiently long recording, after proper alignment, the probability density function (PDF) should be roughly equivalent for neurons of the same class type.

Differences in the intensity PDFs of neurons with identical class types are due to variability in imaging conditions, anisotropy due to random animal orientations, and biological variability in fluorescence expression. To properly compare one animal to another, class-specific neural intensity distributions must be corrected so that they match each other appropriately (e.g., all LUA neurons should exhibit similar PDFs); otherwise, this variability will distort population representations of the signal. In figure 3.2, we explore these population representations of signal by focusing on a single neuron, LUAL, to compare raw, z-scored, QR, and NQR normalized neural traces. Although the LUAL neurons should preserve similar PDFs, instead they exhibit high variability in both their signal magnitude and baseline activity in their traces, histograms, and cumulative distribution functions (CDFs). Z-scoring partially corrects this variability but retains long tails in the PDFs (histograms), while shifting them to zero mean, which is far less than the median signal observed in the raw traces (a median $\Delta F/F_0$ of approximately 2). In comparison, QR and NQR reduce LUAL neural variability substantially, when compared to z-scoring. Moreover, both QR and NQR preserve the median exhibited by the raw traces and, thereby, retain a better approximation of the neural baseline, whereas z-scoring distorts this baseline.

In figure 3.7, we extend our demonstration to all head and tail neurons. In this broader representation of neural activity, one can see that the raw traces and even the z-scored traces distort the neural signal, exhibiting a flat appearance with outliers flanking this flattened signal. In contrast, the QR and NQR traces exhibit strong signals without obvious outliers. Thus both QR and NQR can correct variability in neural intensities to help compare signals from neurons with identical types, recorded from a population of animals.

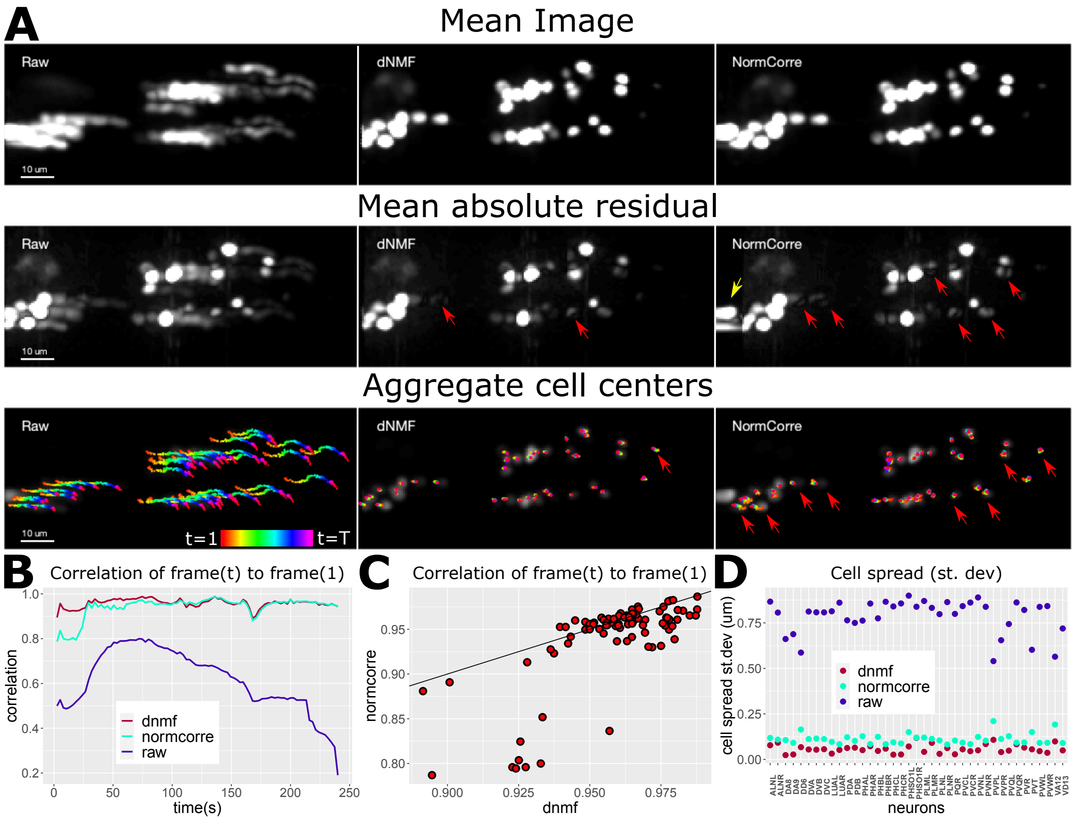

Figure 3.6: ***C. elegans* neural activity video registration results A: Top row:** The mean video frame prior to registration (left), after dNMF based registration (middle), and after Normcorre (right). Overall conventions are similar to Fig. 3.4. **Middle row:** The mean, of the absolute value, of the difference between the first video frame (prior to registration) and subsequent video frames (left). We show these results for dNMF-base registration (middle) and Normcorre (right). Distorted toroidal shapes (indicated by red arrows) denote the superposition of mismatched spatial footprints, indicating a misestimation of deformation. The yellow arrow indicates Normcorre's boundary pixel extrapolation, which introduces blocky artifacts. **Bottom row:** The positions of cells over time superimposed on the first video frame (left), after dNMF-based registration (middle), and after Normcorre (right). Tighter grouping of cell centers indicates a good correction of motion. Spread groupings of cells indicate poor registration (indicated by red arrows). **B:** Correlation of the video frames to the mean frame, across time, for the unregistered video (blue), after dNMF-based registration (red), and after Normcorre (cyan). dNMF slightly outperforms Normcorre here. **C:** Correlation of the individual video frames, to the mean video frame, after registering with dNMF (x-values) and after Normcorre (y-values). The solid line denotes $x = y$. Points below this line (which indicate a higher correlation of registered frames to the average frame) represent better performance for dNMF, and points above this line represent better performance for Normcorre. **D:** The spread of the cell position centers, relative to their average. in the unregistered video (blue), after dNMF-based registration (red), and after Normcorre (cyan). Again, a lower standard deviation for cell spread indicates better performance for local registration of cell shapes. See MOVIE LINK for further details.

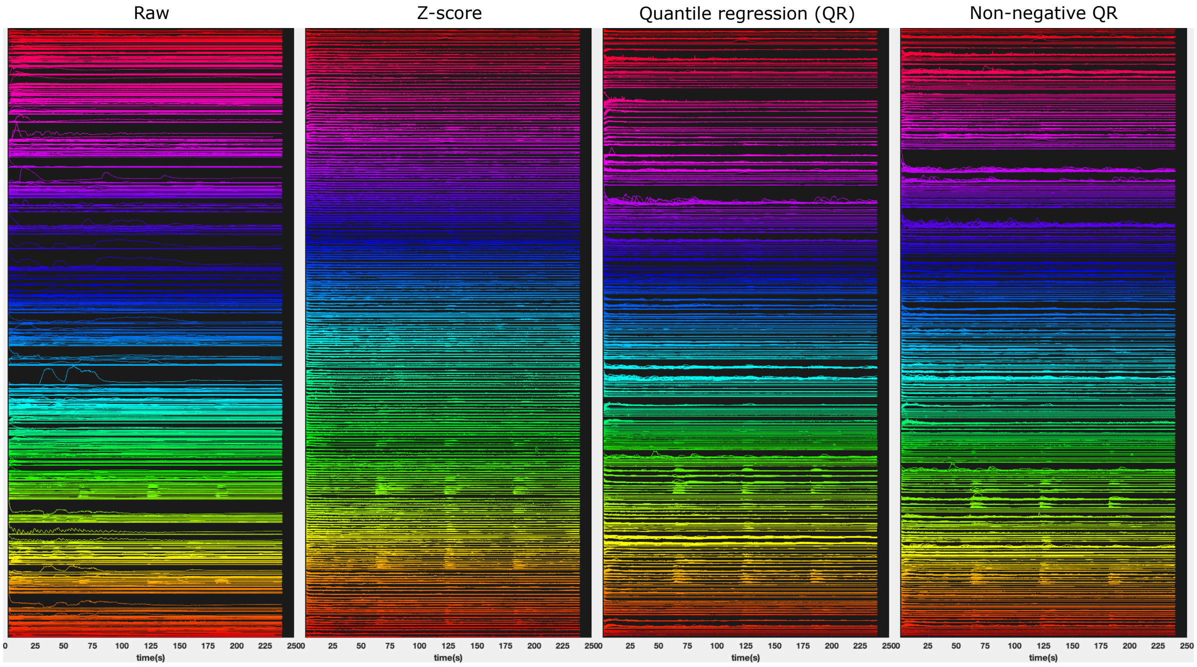

Figure 3.7: **dNMF was used to motion correct, extract, and demix calcium traces of *C. elegans* neurons from 21 animal heads and 21 animal tails** We demonstrate four strategies for superimposing multi-animal traces for different neuron types. **First column:** Raw traces superimposed, colors indicate different neuron types. Within the same color, different traces indicate different animals. (Y-axis: neuron types), (X-axis) time (s). **Second column:** Z-scored neuron time series. **Third column:** Quantile regression (QR) normalized time series. **Fourth column:** Non-negative quantile regression (NQR). Z-scores use only two summary statistics (mean and variance) for normalization. Z-score scaling to unit variance is strongly influenced by any large-magnitude fluctuations in the signal. Consequently, in a mixture of responsive and unresponsive traces from the same neuron, across multiple animals, z-scored traces with a response will be scaled to match their unresponsive counterparts, thus muting signal in these traces. This is exhibited by the compressed appearance of the z-scored traces in the second column. In contrast, quantile regression uses a more robust and rich set of summary statistics to determine an appropriate scaling. As such responsive and unresponsive neural traces retain appropriate differential scales. This is exhibited by the quantile regression methods shown in third and fourth columns which show better preservation of neuron responses when compared to their z-scored equivalents. Additionally, the z-score translates to zero mean and thus can misrepresent the signal baseline. In contrast, both QR methods preserve the correct signal baseline and, when appropriate, the NQR method can be used to further maintain non-negativity of signals (see **Fig. 3.2**).

CHAPTER 4

Causal Models of Neural Data

The fundamental goal of machine learning (ML) has been to build intelligent models that can learn from data and bypass requiring detailed knowledge of the physical processes involved in the underlying system. Ultimately, these models are applied to downstream decision-making or policy-making processes, or they are interrogated to achieve a causal understanding of the underlying system. However, the change in the environment and distributions that generate data are often ignored when using the models for downstream applications. Specifically, in biological settings, the collected data rarely exhibits iid assumptions, and systems often involve complex nonlinear dynamic interactions and history dependence. Formalized by J. Pearl, causal inference describes a causal model by defining a hierarchy consisting of observational, interventional, and counterfactual layers [Pearl, 2009]. This hierarchy is motivated by intuition from human reasoning and thinking in a changing environment which consists of seeing, interacting with, and imagining the counterfactual effect of potential outcomes. The notation $p(y|do(x))$ defines the interventional distribution, modeling the density of variable y if a variable x is intervened on. This is very different from the conditional distribution $p(y|x)$ because it involves a conditional distribution in a modified graphical model where the causal arrows into the variable x are removed. This distinction allows us to model the changing and fixed aspects of the environment, and find conditions under which we can use a combination of observational and interventional data and answer causal queries in a changing environment.

4.1 Introduction to Causal Inference

The focus of ML in the prior decade has been on dataset where samples are iid. The main underlying assumption in this setting is that the data used for training the models exhibits similar statistical patterns to the data that the model will be tested on. This assumption is not valid in most applications due to constant changes in the environment that the data is collected from. The classical ML paradigm assumes the existence of an unknown true model $\mathscr{M}$ that generates sample of the data $\boldsymbol{x}$ and some dependent aspect of $\boldsymbol{x}$ denoted by $\boldsymbol{f}_{\mathscr{M}}(\boldsymbol{x})$. In this setting, training ML models amounts to finding a function $\boldsymbol{h} \in \mathscr{H}$ that best approximates $\boldsymbol{f}_{\mathscr{M}}(\boldsymbol{x})$. The set $\mathscr{H}$ is called the hypothesis class and contains the functions that can be achieved using the training process. In the **supervised** case, the data consists of pairs $\{(\boldsymbol{x}, \boldsymbol{y})\}$ and the goal is to model $p(\boldsymbol{y}|\boldsymbol{x})$ whereas in the **unsupervised** case we are given vectors $\{(\boldsymbol{x})\}$ and we want find patterns by means of modeling $p(\boldsymbol{x})$. Examples of supervised learning are classification and regression and examples of unsupervised learning are clustering and dimensionality reduction.

How should we move away from the iid assumption and what is missing in the classical ML paradigm? The case is made by several examples popularized as Simpson's paradox (see [Pearl, 2009] for reference). The intuition comes from the observation that different generative processes of the data can give rise to identical joint distributions. Therefore solely acquiring the information of associations between the random variables is not sufficient for making causal conclusions. To cure this, the framework of structural causal models (SCM) is introduced by Judea Pearl incorporating causal statements in the form of functional arguments. Below we will introduce this framework and we reference it throughout this chapter.

Definition 4.1.1 (Structural Causal Models)**.** SCMs consist of two sets of variables, U and V, and a set of functions F. Variable X is cause of Y if it appears as an argument in the function that assigns Y values. U is called exogenous and V is called endogenous variables.

Definition 4.1.2 (Causal Diagrams)**.** Each SCM induces a causal diagram where nodes represent variables and arrows correspond to causal influences.

Remark. Notice that causal diagrams are different from probabilistic graphical models (PGM) in that an arrow in PGM is merely an association between variables but in SCM arrows entail causal relations which is a stronger form of association. Of note, each SCM induces a joint distribution on its variables.

Classical statistics manoeuvres around joints, marginals, and conditionals but equipped with SCMs we are now able to define distributions that are non-associative. In Pearl's words, if $p(y|x)$ corresponds to the observational distribution and can be achieved by means of "seeing" the world, $p(y|do(x))$ is defined as the interventional distribution and is achieved by acting in the world. Below we have a formal definition of interventional distributions through the language of SCMs.

Definition 4.1.3 (Interventional Distribution)**.** For an SCM with joint distribution p and causal diagram $\mathscr{G}$, the interventional distribution $p(y|do(x))$ is defined as a conditional distribution in a modified causal diagram $p_{\mathscr{G}_m}(y|x)$ where the incoming arrows to x are removed.

Table 4.1: Table summarizing the effect of the drug on men, women, and the combined population with a certain disease (taken from [Pearl, 2009]).

Recovery table	Drug	No Drug
Men	81/87 (93%)	234/270 (87%)
Women	192/263 (73%)	55/80 (69%)
Combined	273/350 (78%)	289/350 (83%)

It is exactly this property of SCMs that allow us to make causal conclusions in an ever-changing environment by the virtue of modifying the causal diagrams and incorporating the changes by adding and removing causal arrows. Consider the following example known in the literature as Simpson's Paradox. The Table. 4.1 shows the number and percentage of people recovered by using a certain type of drug when diagnosed with a particular disease. The table shows that if we do take into account gender (male or female) we should prescribe the drug, but if we do not know a patient's gender we should not! It is a paradox.

We can resolve this contradition by considering a causal story behind the creation of the disease.

- **Case 1:** Estrogen has a negative effect on recovery. Consider a scenario in which women are less likely to recover from the disease (regardless of taking the drug), also women are more likely to take the drug. In this case the reason for the combined effect of being a woman is a common cause of both drug-taking and failure to recovery. Notice that if we believe that this is the undelrying causal story, then we should consult the separated table since difference in recovery rates is not ascribed to estrogen.
- **Case 2:** Drug affects recovery by lowering blood pressure (BP), but also has a toxic effect. In this case if we separate the data by post-treatment BP, in the combined table we see improved recovery rate due to the effect on BP but in the separated

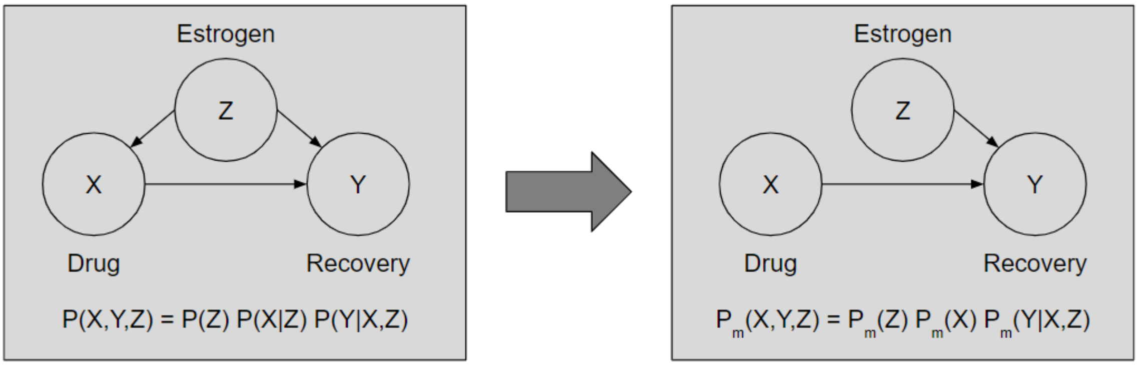
Estrogen
Z
X
Y
Drug
Recovery
P(X,Y,Z) = P(Z) P(X|Z) P(Y|X,Z)
Estrogen
Z
X
Y
Drug
Recovery
$P_m(X,Y,Z) = P_m(Z) P_m(X) P_m(Y|X,Z)$

Figure 4.1: **Causal graphical model for the first scenario of Simpson's paradox** Left: the underlying causal graph for case 1 in Simpson's paradox; Right: the modified graph used for computing the interventional distribution.

> tables we only see the toxic effect. Therefore if we believe that this is the underlying causal story, then we should consult the segregated tables.

We illustrate the first scenario by the causal graph in Fig. 4.1. In this case we can write the joint distribution of Drug denoted by X, Recovery denoted by Y, and Estrogen denoted by Z as $P(X,Y,Z) = P(Z)P(X|Z)P(Y|X,Z)$. To decide whether to consult the separated or combined table, we need to compute the interventional distribution $P(Y|do(X))$ which corresponds to the causal effect of Drug on Discovery when controlling for all other causal factors in play such as Estrogen.

To compute this we need to do a graph surgery and remove the incoming edges to X

and compute the conditional $P_m(Y|X)$ in the modified graph.

$$\begin{aligned}
P(Y|do(X)) &= P_m(Y|X) && \text{Definition}\\
&= \sum_Z P_m(Y|X,Z)P(Z|X) && \text{Probability Rules}\\
&= \sum_Z P_m(Y|X,Z)P_m(Z) && \text{Independece of } X, Z \text{ in } G_m\\
&= \sum_Z P(Y|X,Z)P_m(Z) && P_m(Y|X,Z) = P(Y|X,Z)\\
&= \sum_Z P(Y|X,Z)P(Z) && P_m(Z) = P(Z),
\end{aligned}$$

This is an example where we can compute the interventional distribution using observational data. This process is called *Causal Identification* and the equations used above are examples of *do-calculus*. Going back to our original motivation, making causal conclusions in a changing environment, one can observe that $P(Y|do(X))$ can adapt itself to changes in Z if we assume the graph in Fig. 4.1.

So far the examples described above only considered the case in which the underlying causal structure is known. This is unrealistic in a real world application because often determining the causal variables in play is not possible, let alone having access to the underlying SCM. The field of Causal Inference has recently focused on discovering the causal structure from single or multi-environment observational data or both observational and interventional data. This process is referred to as *Causal Discovery* in the literature. Since obtaining interventional data is an expensive process, there is an interest in active learning setting, where the interventions are sequentially designed to provide the most information about the target causal query. Finding the most informative intervention is referred to as *Active Causal Discovery*.

4.1.1 Identifiable Models

A closely related topic in statistics is identifiable models which refer to statistical models where datasets can only be uniquely generated by the parameters of the model (up to a certain known family of transformations such as affine). These models are of particular interest by statisticians because they can lead to interpretations that are not consistent across different datasets (say coming from different labs or environments). Therefore if the ranodm variables of the model have causal interpretations, then inference in these models using observational data is equivalent to performing causal discovery.

4.1.2 Causal Inference in Time Series

For time series, existing methods mainly use the potential outcome framework. *Synthetic Controls (SC)* aim to identify the counterfactual effect of an intervention at a given time point using autoregressive models [Abadie et al., 2010; Alberto Abadie, 2021]. Under the SC framework, the intervened signal is first estimated as a convex combination of non-intervened signals, referred to as the donor pool. This estimation only uses the time series before the intervention. The estimated coefficients are then used to extrapolate the value of the intervened time series after the time of the intervention, had it not been intervened. The effect of intervention is measured as the mean difference between intervened and not intervened signal denoted by *Average Treatment Effect (ATE)*. More recently the framework of *Synthetic Intervention (SI)* was introduced [Agarwal et al., 2020]. The main benefit of SI is that instead of estimating the counterfactual effect, it allows for estimating the intervention effect in time, but it requires the donor pool to exist both in the observational and interventional layers.

To account for temporal dependencies in the data, differential equations are shown to be simple and powerful tools. Instead of modeling the input-output relationship, differential equations model the changes in the data from each time point to the next [Peters et al., 2017]. Often the observations of a time-varying system are accompanied by some noise, either introduced by the measurement device, or during the information transmission between different components of the system. Having a model with stochastic and deterministic separate components allows us to capture the noise and isolate the time dynamics for further interpretability [Roweis and Ghahramani, 1999].

4.1.3 Causal Inference in Neuroscience

Interventional studies are becoming a quintessential part of systems neuroscience studies. When a circuit mechanism is hypothesized based on observational data, the next step is to confirm the correlational hypothesis using a causal intervention. Various tools in experimental neuroscience is developed for this purpose ranging from intervening on specific neurons using optogenetic stimulations to intervening on groups of neurons using electrophysiological microcircuit stimulations or intervening on brain regions using lesioning studies.

In addition to the above, another popular development in the field is to infer the causal influence of one variable (neuron, brain region, etc.) onto the other, or infer the hierarchy of influence in a network of interacting units. This is referred to as causal functional connectivity with a long-standing literature of developed methods and their interpretations [Edinburgh et al., 2021].

This chapter is organized as following. In the first section I define the notion of *interventional connectivity (IC)*, summarizing the treatment effect of a multivariate time series with N nodes in an $N \times N$ matrix. I then investigate which of the developed causal connectivity metrics can best recover IC for simulated chaotic network dynamics and monkey electrophysiology data from prefrontal cortex [Nejatbakhsh et al., 2020a]. In the second chapter I build on the literature of model identification and active causal discovery and develop a time series model for improving the fitting and generalization properties of switching models (unpublished).

4.2 Functional Causal Flow

4.2.1 Introduction

Complex cognition in humans and other primates is an emergent property of the collective interactions of large networks of cortical and subcortical neurons. Targeted manipulation of the brain to alter cognitive behavior will be greatly facilitated by understanding the causal interactions within ensembles. Examples of such manipulations include altering the perceptual judgement of motion direction in area MT [Salzman et al., 1990, 1992], biasing object classification towards faces in the inferior temporal cortex [Afraz et al., 2006; Moeller et al., 2017; Parvizi et al., 2012], changing the value of an associated stimulus in the anterior caudate [Santacruz et al., 2017], and controlling movements and postures in motor and premotor cortex [Graziano et al., 2002]. Recent experimental [Chettih and Harvey, 2019] and modeling studies [Sadeh and Clopath, 2020] in mice showed that targeted optogenetic manipulations of single neuron activity evoking just a few spikes can reveal the local functional structure of cortical circuits. Although opto- and chemo-genetic manipulations have become recently available for monkeys, the prevalent technique for perturbation in both human and non-human primates is targeted electrical stimulation. Here, we focus on electrical stimulation as other alternatives remain infrequently used in humans. Perturbations of neural circuits represent a promising avenue for ameliorating cognitive dysfunction in the human brain, as well as development of future brain-machine interfaces.

A crucial challenge in targeted perturbation is to identify perturbation sites, satisfying at least two requirements. The first is selectivity: the local neural population around the site should exhibit specific selectivity properties for the desired perturbation effect, e.g., motion direction selectivity in area MT [Salzman et al., 1990, 1992], face selectivity in face patches of inferotemporal cortex [Afraz et al., 2006; Moeller et al., 2017; Parvizi et al., 2012], or the locus of seizures in epilepsy [Fisher and Velasco, 2014]. The second is efficacy: stimulation of the local population should exert some significant effect on the activity of the rest of the brain, and consequently on behavior. While selectivity of sensory and motor neurons may be estimated by recording neural activity in simple and well-defined tasks, selectivity tends to be quite complex or variable across tasks in many regions of the association cortex. Further, discovering efficacy is currently achieved by trial-and-error: many perturbations are performed until a site whose stimulation leads to a significant change in activity is located. As a result, current methods for targeted perturbations are labor intensive, time consuming, and often unable to generalize beyond the limited task set they are optimized for.

A promising avenue for predicting the efficacy of a potential perturbation site is to examine its functional connectivity within a local neural circuit. Intuitively, one expects that perturbing a node with strong functional connectivity to other nodes within a circuit may exert stronger effects than perturbing the nodes that are functionally isolated. Estimating the functional connectivity in cortical circuits is a central open problem in neuroscience [Marinescu et al., 2018]. Existing methods for estimating functional interactions between multi-dimensional time series are challenged by the properties of neural activity in the cortex. Cortical circuits comprise highly recurrent neural networks [Binzegger et al., 2004; Braitenberg and Schüz, 2013; Lefort et al., 2009; Thomson and Lamy, 2007], where the notion of directed functional couplings is not obvious. Correlation-based methods [Cocco et al., 2009] lack sufficient power when correlations are weak, as in most cortical circuits [Cohen and Kohn, 2011]. Granger causality, a widely used method [Dhamala et al., 2008; Faes et al., 2011; Granger, 1969; Sheikhattar et al., 2018], relies on the assumption of linear dynamics and thus it is challenged when the circuit's dynamical properties are not well known. Transfer entropy [Schreiber, 2000a], applicable to non-linear systems, requires large datasets hard to acquire in conventional experiments. Critically, both Granger- and entropy-based methods require stochasticity and are challenged in the presence of self-predictability in deterministic dynamics such as nonlinear couplings between variables [Sugihara et al., 2012]. Moreover, commonly encountered confounding effects such as phase delay [Vakorin et al., 2013] or common inputs [Sugihara et al., 2012] render these methods unreliable. It is thus of paramount importance to develop new theoretical tools. These new tools should be able to estimate functional connectivity in the presence of common inputs using extremely sparse recordings, typical of cortical recordings in humans and monkeys. A promising approach is offered by delay embedding methods, in particular, convergent cross-mapping, which is capable of reconstructing nonlinear dynamical systems from their time series data [Sugihara et al., 2012]. These methods were expressly developed to work precisely in the sparse recording regime [Sauer et al., 1991; Takens, 1981] and in the presence of noise, of common inputs, and of nonlinear couplings between variables [Sugihara et al., 2012], all of which are hallmarks of cortical dynamics and explicitly violate the assumptions underlying Granger-based methods. While this powerful framework, rigorously articulated in [Cummins et al., 2015], has been successfully applied in ecology [Sugihara et al., 2012], and *in vitro* [Sugihara et al., 2012; Tajima et al., 2017] and ECoG neural activity [Tajima et al., 2015], it has never been adapted to spiking activity *in vivo*.

Here, we build on the delay embedding method to develop a statistical approach for predicting the effects of stimulation using causal functional connectivity ("functional causal flow," FCF) based on spiking activity of a simultaneously recorded ensemble in the cortex of awake monkeys (Fig 4.2). We characterize the effects of perturbation by introducing the concept of interventional connectivity, an observable that is agnostic to the underlying structural connectivity and only depends on responses to perturbations; and show that one can efficiently predict interventional connectivity solely based on FCF inferred from small snippets of data collected during resting blocks. We first performed a series of simulated experiments to precisely quantify regime of applicability of FCF and compare its performance to existing methods such as Granger causality (GC) and other traditional methods for estimating functional connectivity. Our simulated experiments are designed to present strong challenges for the inference methods, such as noise and common inputs, yet retaining features of biological plausibility from known cortical circuit dynamics. These validation studies revealed that FCF can predict interventional connectivity accurately even in the extremely sparse recording regime, solely based on short snippets of resting data. We then demonstrate that our method infers the causal flow of ensembles of neurons from sparse recordings of spiking activity, obtained from chronically implanted prefrontal multi-electrode arrays in awake, resting monkeys. Using the causal flow inferred during resting activity, we successfully predict the effect of electrical microstimulations of single electrodes on the rest of the circuit. A critical comparison of FCF with GC and other alternative methods demonstrates the superior performance of FCF both in the simulated as well as in the empirical data. This highlights the advantages of deploying causal flow to guide perturbation experiments compared to traditional methods, opening the way for much more efficient protocols for targeted manipulations of cortical ensembles in primates and humans.

4.2.2 Results

4.2.2.1 Uncovering the functional causal flow with delay embedding

To illustrate the concept and methods of functional causal flow (FCF) and establish its regime of validity, we performed a series of simulated experiments using ground truth data from recurrent network simulations. We chose two sets of neural networks (Fig. 4.2). In the first experiment, we simulated a continuous rate network comprising both feedforward and recurrent features in its structural connectivity, where we arbitrarily varied the noise levels and features to assess FCF robustness against different signal-

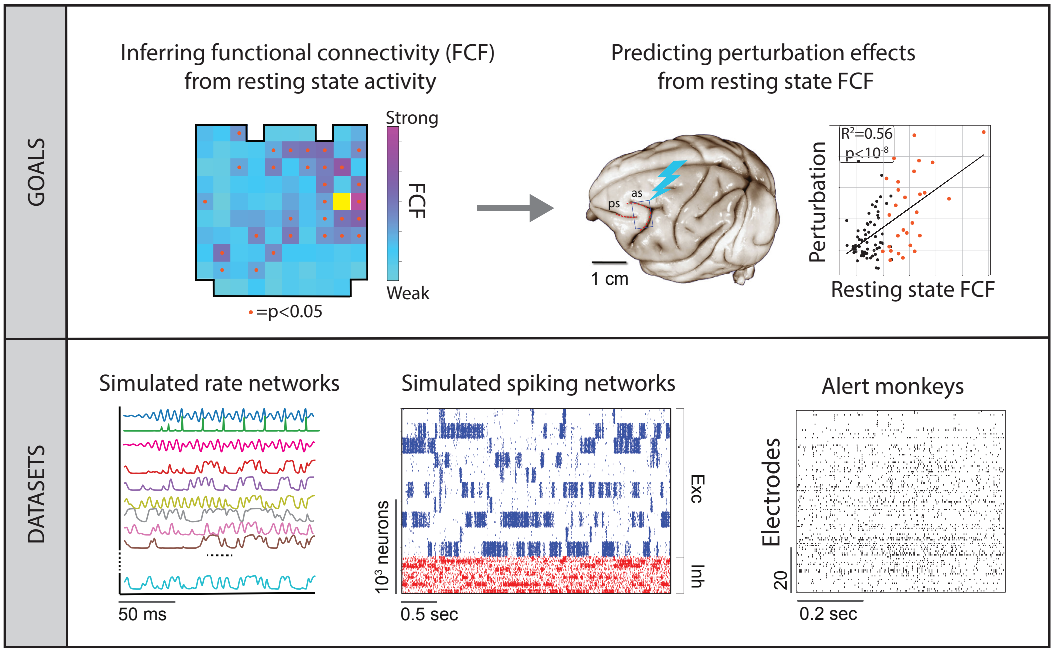

Figure 4.2: **Conceptual summary of Functional Causal Flow** Top left: Functional causal flow (FCF) map inferred from the prefrontal cortex activity of a resting alert monkey. Each square represents an electrode of a 96-electrode Utah array (yellow square: stimulated electrode; orange dots: electrodes with significant FCF and functionally downstream to the stimulated one). Top right: Schematics of an electrical microstimulation experiment and prediction of stimulation effects from FCF. The scatter plot shows the correlations of resting state FCF vs. perturbation effects on efferents with significant (orange dots) and non-significant FCF (black dots). Bottom: We validated our method for predicting perturbation effects from resting state FCF in three different datasets: a chaotic rate network, a spiking network with cell-type specific connectivity, and a prefrontal cortical circuit in alert monkeys.

to-noise ratios and common inputs. In the second experiment, we simulated a cortical circuit model based on a spiking network with cell-type specific connectivity endowed with functional assemblies. This class of spiking models captures the intrinsic neural variability observed in various cortical areas across different tasks and behavioral states[Deco and Hugues, 2012; Litwin-Kumar and Doiron, 2012; Mazzucato et al., 2015, 2019; Wyrick and Mazzucato, 2020]. Calibrating our FCF inference on this spiking model served as a guide for our experimental design in the case of alert monkey.

We first examined a deterministic network Z, comprising N units $z_i = x_i, y_i$ arranged in two subnetworks X and Y, each endowed with their own local recurrent connectivity and, crucially, directed projections from X to Y with coupling strength g; but no feedback couplings from Y to X. Units in X represent a chaotic Rossler attractor and have strong all-to-all recurrent couplings, while units in Y have only sparse and weak recurrent couplings. We aimed to capture the intuitive idea that the upstream subnetwork X drives the activity of the downstream subnetwork Y (Fig. 4.3). It is well known from the theory of deterministic dynamical systems that one can (at least partially) reconstruct the N-dimensional attractor topology of a network of coupled units, represented by the vector time series of the activity of all units $\{\vec{z}(t)\}_{t=1:T}$, by using only the information encoded in the temporal trajectory of a single unit $\{z_i(t)\}_{t=1:T}$. From the mapping between the activity of the full network and the activity of a single unit, one can derive a map between the activity of the units themselves and (at least partially) reconstruct the activity of one unit $\{z_i(t)\}_{t=1:T}$ from the activity of a different unit $\{z_j(t)\}_{t=1:T}$, for $i \neq j$. The reconstruction is possible whenever the two units are functionally coupled. This general property of dynamical systems is known as "delay embedding" [Sauer et al., 1991; Takens, 1981] and relies on a representation of network dynamics using "delay coordinates" (see Fig. 4.3A for details). This reconstruction has also been shown to be robust to noise in driven dynamical systems [Casdagli et al., 1991].

We used convergent cross-mapping based on delay embedding to infer the FCF between all pairs of network units. We first considered the FCF between a unit y_i in the downstream subnetwork Y and a unit x_j in the upstream subnetwork X. The activity of unit x_j only depends on the other units in X, to which it is recurrently connected, but not on the units in Y, as there are no feedback couplings from Y to X. On the other hand, the activity of unit y_i depends both on the units in X, from which it receives direct projections, and on the other units in Y to which it is recurrently connected. In other words, $y_i(t)$ is causally influenced by units in both Y and X, whereas $x_j(t)$ depends only on other units in X. Thus, we expect that the reconstruction of $x_j(t)$ from $y_i(t)$ will be more accurate than the reconstruction of $y_i(t)$ from $x_j(t)$, because in the latter case the causal influence on y_i from the other recurrently connected units in Y is being neglected.

We tested our prediction by reconstructing the temporal series of unit $x_j(t)$ given $y_i(t)$ from the corresponding delay vectors $[x(t), x(t-\tau), \ldots, x(t-d\tau+\tau)]$ and $[y(t), y(t-\tau), \ldots, y(t-d\tau+\tau)]$ of dimension d with a step τ. Reconstruction accuracy was quantified as the Fisher z-transform $z[\rho(y_i|x_j)]$ of the Pearson correlation $\rho(x_j|y_i)$ between the *empirical* activity of the delay vector of unit x_j and its *predicted* activity obtained from the delay vector of unit y_i. Whereas the Pearson correlation is bounded between -1 and 1, its Fisher z-transform is approximately normally distributed thus facilitating statistical comparisons [Fisher, 1925]. The process was cross-validated to avoid overfitting (see Methods for details). Similarly, we estimated cross-validated reconstruction accuracy $z[\rho(y_i|x_j)]$ of the temporal series of unit $y_i(t)$ given $x_j(t)$. As expected, reconstruction accuracy increased as a function of the dimensionality of the delay coordinate vector (i.e., how many time steps back we utilize for the reconstruction, Fig. 4.3A). The accuracy plateaued beyond a certain dimensionality (related to the complexity of the time series [Tajima et al., 2017]), whose value we fixed for our subsequent analyses. We define the functional causal flow (FCF) from x_j to y_i as $F_{ij} = z[\rho(x_j|y_i)]$ (see Methods and table 4.2 for a summary of our conventions). Columns of the FCF represent the *afferent* units, whose activity is being reconstructed, and rows represent the *efferent* units, whose activity is used for the reconstruction. As explained above, the FCF was estimated using a cross-validation procedure to avoid overfitting. Model selection for the FCF hyperparameters d (delay dimension) and τ (time step) depended on the specific datasets. For the continuous rate network considered here model selection yielded optimal values of $d = 7$ and $\tau = 4$ ms. For delay dimensions $d \geq 5$, the sample size dependence of FCF

	FCF	Feature
Upstream	$F_{ij} > 0$ & sig; F_{ji} non-sig	j is causally upstream of i.
Downstream	F_{ij} non-sig; $F_{ji} > 0$ & sig	j is causally downstream from i.
Reciprocal	$F_{ij} \sim F_{ji}$ & both sig	i and j are reciprocally functionally connected.
Independent	F_{ij}, F_{ji} both non-sig	i and j are causally independent.

Table 4.2: Definitions and notations for the functional causal flow (FCF).

plateaued at about 1500 ms.

We established statistical significance by comparing the FCF estimated from the empirical data with that estimated from surrogate datasets carefully designed to preserve the temporal statistics of the network activity while destroying its causal structure (see [Thiel et al., 2006] for details). Surrogates are produced in three stages: first, nearest neighbors of a state are identified in the delay-embedding space, and "twin" states are constructed for each state including all neighbors within a small distance; finally, surrogate trajectories are generated by temporally concatenating states from the same coarse-grained sets of twins, allowing for jumps backward or forward in time while preserving all large-scale nonlinear properties of the system.

Unlike the Pearson correlation r_{ij}, which is a symmetric quantity, the FCF is a directed measure of causality. By comparing the value and significance of F_{ij} with F_{ji}, we can establish the *directionality* of the functional relationship between y_i and x_j, uncovering several qualitatively different cases which we proceed to illustrate (table 4.2).

In the example above, the reconstruction accuracy of x_j given y_i was significant and large, while that of y_i given x_j was not significant. In other words, while one can significantly reconstruct x_j with high accuracy from y_i, because the latter receives information from the former, the opposite is not possible, matching predictions based on the simulated network architecture. We refer to x_j as being *causally upstream* to y_i in the network functional causal flow.

4.2.2.2 Functional causal flow uncovers hierarchical structures

The notion of being causally upstream or downstream is an entirely *functional* relation and *a priori* different from the underlying structural/anatomical coupling between units. We illustrate here two more examples from the network in Fig. 4.3 to reveal the variety of the relationships encoded in the FCF. We considered the FCF between $x_1(t)$ and $x_3(t)$ within the subnetwork X, whose units are part of a Rössler attractor, a well studied dynamical system (see Fig. 4.3B and Methods). Because the X subnetwork does not receive inputs from other network units in Y, it is causally isolated (i.e., its activity is conditionally independent from Y). Hence one can reconstruct the activity of one x_i unit from another with high accuracy, yielding large and significant values for any pair of X units. Fig. 4.3B shows large values for both $\rho(x_1|x_3)$ and $\rho(x_3|x_1)$. This is a classic demonstration of the embedding theorem [Takens, 1981], ensuring accurate bidirectional reconstruction of variables mapping a chaotic attractor. The large and significant $F_{x_1x_3}$ and $F_{x_3x_1}$ reveal that the unit pair has a strong reciprocal functional coupling, and the two units lie at the same level of the functional hierarchy. This is unlike the case of pairs x_i, y_j described above, where a significant F_{ij} but a non-significant F_{ji} showed a strong directional coupling and a functional hierarchy [Stark et al., 1997]. As another qualitatively different pair, we considered two units y_4 and y_5 within the subnetwork Y, whose units are only sparsely recurrently coupled. The FCFs were not significant for this pair (Fig. 4.3C), suggesting that the two units are functionally independent, namely, their activities do not influence each other significantly. The taxonomy of causal flows are summarized in table 4.2 and Fig. 4.3D.

The variety of FCF features discussed so far suggests that, even if the FCF is a measure of pairwise causal interactions, it may reveal a network's global causal structure.

We thus analyzed the N-dimensional *causal vectors* $\mathbf{f}^{(j)} = \{F_{ij}\}_{i=1}^{N}$, representing the reconstruction accuracy of unit j given the activity of each one of the efferents i. The causal vector $\mathbf{f}^{(j)}$ encodes the FCF from unit j to the rest of the network. For example, a significant positive entry i of the causal vector implies that the afferent unit j has a strong functional coupling with efferent i. A Principal Component analysis of the causal vectors from a sparse subsample of the network units (10 out of 103) revealed a clear hierarchical structure present in the network dynamics showing two separate clusters corresponding to the subnetworks X and Y (Fig. 4.3E-F). Thus, causal vectors revealed the global network functional hierarchy from sparse recordings of the activity.

We further quantified the hierarchical functional structure of causal vectors, measured by their Gini coefficients (Fig. 4.3F), which estimates the degree of inequality in a distribution. For example, a delta function, where all points have the same value, has zero Gini coefficient, while an exponential distribution has a Gini coefficient equal to 0.5. In the absence of hierarchies, one would expect all efferents from a given afferent unit to have comparable values, namely, yielding a low Gini coefficient. Alternatively, heterogeneity of FCFs across efferents for a given afferent would suggest a network hierarchy with a gradient of functional connectivities, yielding a large Gini coefficient. For our simulated network, we found a large heterogeneity in the distribution of causal vectors Gini coefficients, capturing the functional hierarchy in the network. For comparison, when restricting the causal vectors to afferents in either X or Y (green and brown bars in Fig. 4.3F, respectively), we found a clear separation with larger Gini coefficients for X afferents and lower Gini coefficients for Y afferents. This result shows that the feedforward structural couplings from X to Y introduce a hierarchy in the full network Z, encoded in the network causal vectors. Importantly, inferring this structure does not require observing the full network and can be achieved by recording from a small subset of the network units.

4.2.2.3 Inferred causal flow predicts the effects of perturbation

Can we predict the effects of perturbations on network activity from the causal flow inferred in the unperturbed system? We hypothesized that the effects of stimulating a specific node on the rest of the network can be predicted by the causal flow inferred during the resting periods.

We simulated a perturbation protocol where we artificially imposed an external input on one afferent network unit for a brief duration, mimicking electrical or optical stimulation protocols to cortical circuits (Fig. 4.4A). We estimated the stimulation effect on each efferent unit, by comparing the distribution of binned activity in each efferent in intervals preceding the stimulation onset and following its offset (Fig. 4.4B). We found that stimulation exerted complex spatiotemporal patterns of response across efferent units, which we captured in the *perturbation vector*: $\mathbf{I}^{(j)} = \{I_{kj}\}_{k=1}^{N}$, where I_{kj} is the interventional connectivity matrix (Fig. 4.4B). The entries in the perturbation vector represent the Kolmogorov-Smirnov test statistics between pre- and post-stimulation spiking activity aggregated over several stimulation trials of the same neural cluster. Stimulation effects across efferents k strongly depended on the afferent unit j that was stimulated. Perturbation effects increased with stimulation strength for afferent-efferent pairs in $X \to X$, $X \to Y$ and $Y \to Y$, but did not depend on stimulation strengths for pairs $Y \to X$, consistent with the underlying structural connectivity lacking feedback couplings $Y \to X$ (Fig. 4.4C). Can one predict the complex spatiotemporal effects of stimulation solely based on the FCF inferred during resting activity?

We hypothesized that, when manipulating afferent unit i, its effect on efferent unit k could be predicted by the FCF estimated in the absence of perturbation (Fig. 4.4D). Specifically, we tested whether: stimulation of afferent unit i would exert effects only on those efferent units k that have significant FCFs, F_{ki}; but no effects on units whose FCFs were not significant; and that stimulation of "downstream" units in Y would not exert any effect on "upstream" units in X. We found a strong predictive power of FCF regarding perturbation effects (Fig. 4.4D). More specifically, we found that the perturbation effects on the efferent units were localized on units with significant FCFs (dots in Fig. 4.4D); no effects were detected on pairs with non-significant FCF. In particular, we found that pairs where the stimulated afferent was in Y and the efferent in X did not show any significant effects of perturbations (Fig. 4.4E); this was expected given the absence of feedback couplings $Y \rightarrow X$. Two crucial features of the FCF, underlying its predictive power, were its directed structure and its causal properties.

We thus conclude that the causal effect of perturbations on network units can be reliably and robustly predicted by the FCF inferred during the resting periods (i.e., in the absence of the perturbation).

4.2.2.4 Inferring the causal flow from alert monkeys during resting periods

To test our theory, we performed an experiment comprising recording and stimulation of spiking activity in alert monkey prefrontal cortex (pre-arcuate gyrus, area 8Ar) during a period of quiet wakefulness (resting) while the animal was sitting awake in the dark. The experiment had two phases (Fig. 4.2 and 4.5). In the first phase, we recorded population neural activity from a multi-electrode array (96-channel Utah array, with roughly one electrode in each cortical column in a $4 \times 4\text{mm}^2$ area of the cortex), estimating the FCF between pairs of neural clusters (multiunit activities collected by each recording electrode). In the second phase, we perturbed cortical responses by delivering a train of biphasic microstimulating pulses (15 μA, 200 Hz) to one of the clusters for a brief period (120ms), recording population neural activity across the array before and after the stimulation.

We first examined whether causal flow could be estimated reliably for the recorded population, which constituted a small fraction of neurons in the circuit. Following previous experimental evidence supporting the existence of assemblies in monkey prearcuate gyrus [Kiani et al., 2015], we reasoned that the activity of neural clusters around each electrode may represent sparse samples from a local cortical assembly. In Fig. 4.5A we show four representative 96-dimensional causal vectors representing the FCF for each of four different afferent clusters recorded in two different sessions (channels 14 and 56 from session 1 and channels 42 and 29 from session 2). We overlaid the causal vectors onto the array geometry (location of recording electrodes in the array) for illustration.

Comparison of the causal vectors across afferents revealed remarkable features about the structure of the functional connectivity. First, FCF is channel-specific, namely, it depends on the afferent clusters whose activity is being reconstructed. Second, each causal vector shows a hierarchical structure, with significant FCF in a subset of downstream efferents, while most efferents cannot reconstruct the afferent activity (Fig. 4.5A). This result is qualitatively consistent with the FCF obtained from our model in Fig. 4.4, supporting the hypothesis of functional hierarchies embedded within prefrontal cortical circuits [Kiani et al., 2015].

Is the FCF of an afferent cluster to different efferent clusters uniformly distributed across the array, or is there a preferential spatial footprint of FCF? We found a spatial gradient whereby FCF was largest in the efferent clusters immediately surrounding the afferent cluster, while FCF for distant efferents typically plateaued at low but nonzero values (Fig. 4.5C).

We thus concluded that FCF inferred during the resting periods was cluster-specific and revealed a hierarchy of functional connectivity where functionally downstream neural clusters are spatially localized around the afferent cluster. These results extend previous correlation analyses of spatial clusters in alert monkeys [Kiani et al., 2015] highlighting a spatial gradient of directed functional couplings at the mesoscale level.

4.2.2.5 Perturbation effects on cortical circuits in alert monkeys

We next proceeded to examine the effect of microstimulation on the cortical activity in alert monkeys. We estimated perturbation effects by comparing the activity of neural clusters in the intervals preceding the onset and following the offset of the stimulation of the afferent, for each pair of stimulated afferent and recorded efferent (see Fig. 4.5B). We focused on the activity after offset as opposed to during the stimulation period to minimize the effects potential stimulation artifacts on the recording apparatus. Perturbation effects were quantified via a Kolmogorov-Smirnov test statistics aggregated over all stimulations of a specific neural cluster (comparison between the pre- and post-perturbation distributions of activity, see Methods).

We first examined the spatiotemporal features of stimulation effects. We found that perturbations exerted a strong effect on ensemble activity, and that these effects where specific to which afferent channel was stimulated (Fig. 4.5B; perturbation effects for each stimulated afferent j are visualized as a perturbation vector $\mathbf{I}^{(j)}$ overlaid on the array geometry). By comparing the effects of perturbing a single cluster across all efferents, we found a hierarchical structure with strong effects elicited in specific subsets of efferent clusters. The identity of strongly modulated efferents was specific to the stimulated channel. We found a spatial gradient in perturbation effects, whereby distant efferents were less affected by perturbation, though the effects were nonzero even far away from the stimulated afferent (Fig. 4.5D).

4.2.2.6 Predicting perturbation effects from resting activity in alert monkeys

Our theory posits that the effects of stimulation of afferent cluster j on the efferent neural clusters can be predicted by the corresponding causal vector $\mathbf{f}^{(j)}$ inferred at rest (i.e., a column of the FCF matrix; four representative causal vectors are overlaid on the array geometry in Fig. 4.5A). Specifically, our theory predicts that perturbing an afferent cluster exerts a strong effect on those efferent clusters which have a strong functional connectivity to the afferent, identified by a significant resting state FCF as read out from the afferent causal vector. Moreover, perturbation effects on efferents with significant FCF should be stronger compared to efferents with non-significant FCF. Visual inspection of the resting state FCF causal vectors (Fig. 4.5A) and comparison to the map of perturbation effects (Fig. 4.5B, perturbation vectors) suggest that the FCF and perturbations are strikingly similar for a given afferent. We confirmed this intuition quantitatively and found that the FCF inferred at rest was indeed predictive of perturbation effects at the level of single stimulated afferent (Fig. 4.5E, Pearson correlations between causal vectors and perturbation vectors). In particular, we found that for all stimulated clusters, the effect of a perturbation was significantly stronger on efferents with strong functional connectivity to the stimulated cluster compared to efferents with weak functional connectivity, as predicted by our theory (Fig. 4.5F). The predictive power of FCF held at the level of single stimulated afferents, thus achieving a high level of granularity in prediction.

These results demonstrate that the causal flow estimated from sparse recordings during the resting periods accurately predicts the effects of perturbation on the neural ensemble at the single channel level, thus establishing the validity of our theory in cortical circuits of alert primates.

4.2.2.7 Comparison to Other Causality Indices: Results

A recent paper [Edinburgh et al., 2021] investigates how different causality indices recover the direction of causation in simulations where there is a clear unidirectional influence from one variable onto the other. Here we apply those indices to our simulations and real data to address whether or not other indices can recover the direction of influence in the presence of recurrence and network dynamics. Below we first briefly explain each causality index. Then, we present results on the simulated rate network from Figs. 4.3-4.4 and monkey prefrontal data from Fig. 4.5 to investigate which indices can predict perturbation effects measured by interventional connectivity.

A general principle underlying all alternative causality indices is that causality is defined by the precedence of influence in time. If the past of variable X contains information about or allows the prediction of the future of variable Y then there is causal influence from X to Y. This is precisely the idea behind the definition of *Granger Causality (GC)* and its variants. If we assume that two signals evolve jointly according to an autoregressive model, then GC measures if the past of X,Y together helps predicting the future of Y better than the past of Y alone. The significance test is performed using F-test as commonly done in the GC literature.

Transfer Entropy (TE) is defined similarly, but TE relaxes the autoregressive assumption to arbitrary rules for the stochastic evolution of time series, computing the conditional mutual information between the future of Y and past of X conditioned on the past of Y [Marschinski and Kantz, 2002; Schreiber, 2000b]. It is worth noting that if the data follows an autoregressive model, GC and TE become equivalent. Although TE is nonparametric, its estimation is a challenging statistical task often requiring large amounts of data. For TE, here we use an estimator developed by [Kraskov et al., 2004] and employed by [Edinburgh et al., 2021] which is based on nearest neighbor methods.

Although GC is originally developed for univariate and autoregressive signals, one can generalize it to multivariate and nonlinear counterparts. *Multivariate GC* (MGC) computes the same criterion as GC with the difference that the conditioning is done on all the other variables in the multivariate time series. This allows for measuring the unique predictability of future of Y from past of X when we control for other intermediate signals in the network. *Nonlinear GC* (NGC) performs the autoregression using radial basis functions [Ancona et al., 2004]. *Extended GC* (EGC) provides another generalization to GC based on locally linear approximation [Chen et al., 2004].

Notice that in contrast to these methods which rely on the stochastic fluctuations of signals, FCF is based on the deterministic aspects of a dynamical system and instead of measuring noise statistics it uses nearest neighbors in the state space of a stationary dynamical system to predict one signal from the other. Moreover, in the limit of large datasets, FCF is less affected by the unobserved nodes due to the topological correspondence between the time-lagged history of each variable and the high-dimensional data generating system which includes the unobserved nodes.

We summarize the results obtained from different causality indices in Figs. 4.6 and 4.7. In the simulated rate network, FCF and GC show significant positive correlations with IC, and FCF performs best. For the monkey data, GC and FCF show positive correlations with IC but FCF is more robust and outperforms other indices. TE fails perhaps due to the small sample size or the presence of observed and unobserved nodes in the network which are not accounted for. A summary of the hyperparameters used for the simulations and calculating different causality indices are included in the corresponding `config` file on the code repository released with this paper:
`https://github.com/amin-nejat/CCM/tree/master/example_configs`.

4.3 Controlled Switching Linear Dynamical Systems

4.3.1 Introduction

Time Series The literature on causal inference has mainly focused on directed acyclic graphs (DAG). This is because the notion of causal interaction is substantiated in terms of function arguments and it is not possible to have functions f, g such that $x = f(y)$ and $y = g(x)$. However bidirectional interactions are fundamental to biological systems and modeling them is essential in most applications. This limitation of CI can be bypassed using time series models where signals are modeled as sequences of random variables.

For time series, existing methods mainly use the potential outcome framework. Synthetic controls (SC) aim to identify the counterfactual effect of an intervention at a given time point using autoregressive models [Abadie et al., 2010; Alberto Abadie, 2021]. Under the SC framework, the intervened signal is first estimated as a convex combination of non-intervened signals, referred to as the donor pool. This estimation only uses the time series before the intervention. The estimated coefficients are then used to extrapolate the value of the intervened time series after the time of the intervention, had it not been intervened. More recently the framework of synthetic intervention (SI) was introduced [Agarwal et al., 2020]. The main benefit of SI is that instead of estimating the counterfactual effect, it allows for estimating the intervention effect in time, but it requires the donor pool to exist both in the observational and interventional layers.

To account for temporal dependencies in the data, differential equations are shown to be simple and powerful tools. Instead of modeling the input-output relationship, differential equations model the changes in the data from each time point to the next [Peters et al., 2017]. Often the observations of a time-varying system are accompanied by some noise, either introduced by the measurement device, or during the information transmission between different components of the system. Having a model with stochastic and deterministic separate components allows us to capture the noise and isolate the time dynamics for further interpretability [Roweis and Ghahramani, 1999].

State Space Models There is a long-standing literature on statistical inference in noisy dynamic systems. In the simplest case, a Markov model consists of a discrete set of states and stochastic transitions between them where a transition matrix encodes the probability of transitioning between pairs of states. In most applications, the states and transitions between them are unobserved latent variables, instead a temporal rule determines the dynamic regime of a different set of observed variables that evolve according to the state rules. For example, in a Hidden Markov Model, the observed variables can follow linear dynamics where the linear matrix is indexed by the discrete state. Instead of discrete latent dynamics, we can consider cases where latent dynamics follow low-dimensional continuous dynamics and the observations are noisy projections of the latent space. If both latent dynamics and observed projections are parameterized by linear functions, the model is called a Linear Dynamical System [Roweis and Ghahramani, 1999]. Combining the two ideas, we can build hybrid models where a set of latent discrete states inform the dynamics of a low-dimensional continuous latent dynamics, and the observations are generated by the noisy projection of continuous latents [Linderman et al., 2017]. This family of models is called Switching Linear Dynamical Systems [Fox et al., 2009]. Depending on the problem at hand, more complex structures and parameterizations can be incorporated into these models. For example, discrete state transitions can be made a function of observations and the observations can evolve according to nonlinear dynamics parameterized by a neural network [Gao et al., 2016].

The graphical model shown in Fig. 4.8 is of special interest, where the observations follow piecewise linear dynamics, with transition boundaries defined by hyperplanes crossing observational space. This provides a flexible function family as all stochastic dynamical systems can be approximated up to arbitrary precision with this class of model with an increasing number of pieces, much like piecewise linear function can globally approximate smooth functions as the number of pieces grows larger. The functional space given by this model is equivalent to that of ReLU networks if there is no observational noise. Although SLDS models have shown to be successful in fitting complex biological data and their fitted parameters are associated with biological interpretations, there is no guarantee that the fitted SLDS has any causal correspondence to the underlying dynamics if we only use observational data for fitting, even if the underlying system is governed by switching dynamics [Linderman et al., 2019].

Recently, SLDS models are used as simpler models to describe and understand the dynamics of trained complex nonlinear RNNs. RNNs trained for a motor reaching task show switching dynamics between linear pieces with each piece corresponding to a reach direction [Smith et al., 2021].

4.3.2 Methods

This has motivated the recent line of research on performing inference using both interventional and observational data. Intuitively, we expect the real-world interventions in the system to influence a small subset of variables, locally or sparsely. This principle which is referred to as Sparse Mechanism Shift (SMS) by Scholkopf can be used to distinguish between causal and non-causal representations or can be applied in the form of regularization to encourage disentangled and causal representations [Scholkopf et al., 2021]. The significance of causal representations is that not only can lead to a stronger form of understanding of the underlying system, but also they can provide models that are robust to distribution shifts and enjoy out-of-distribution generalization properties. [Peters et al., 2016] shows that given interventional data (x_e, y_e), treating each intervention as an environment we can use a form of invariant risk minimization to identify variables that are causally downstream of x_e. [Brehmer et al., 2022] shows that if we have access to pairs of intervened and non-intervened variables under the same realization of exogenous variables, we can learn a latent space that is identifiable to the data generating latent space up to component-wise transformations. Following this line of work, here we take a step towards the identification of switching systems and extending them to reflect some causal characteristics of the underlying system.

Assuming that the underlying process follows a set of differential equations denoted by $\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t)$, traditional fitting uses the observation from this system to fit a simpler model, say a ARHMM, parameterized by $\boldsymbol{\theta}$ and described by

$$z_{t+1}|z_t \sim \text{Categorical}(\sigma(\boldsymbol{V}\boldsymbol{x}_t))$$

$$\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, z_t \sim \mathcal{N}(\boldsymbol{A}_{z_t}\boldsymbol{x}_t + \boldsymbol{b}_{z_t}, \boldsymbol{Q})$$

where $\boldsymbol{\theta} = \{\boldsymbol{A}_{1:K}, \boldsymbol{b}_{1:K}, Q, \boldsymbol{V}\}$ such that the data generated from the fitted system is close to that of generated from the original system in a probabilistic sense, namely they aim to solve $\max_{\boldsymbol{\theta}} \log P(\boldsymbol{\theta}|\boldsymbol{x}_{1:T})$. The data generating system can be a RNN trained for solving a task, the firing rate of a set of neurons in the brain, or trajectories simulated from a set of differential equations. Alternative to the traditional approach, if we have access to the components of the underlying system in an interventional level, we would like to apply informative dynamic interventions to augment the collected data with interventional data. Formally, we introduce a matrix $\boldsymbol{B}$ and control inputs $\boldsymbol{u}_1{:}T$ and intervene the system by injecting the inputs in the following way $\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t) + \boldsymbol{B}\boldsymbol{u}_t$. Now instead of the original optimization problem, we replace it with the following optimization that involves both observational and interventional data with the knowledge of matrix $\boldsymbol{B}$: $\max_{\boldsymbol{\theta}} \log P(\boldsymbol{\theta}|\boldsymbol{x}_{1:T}, \boldsymbol{u}_{1:T})$. We choose the following class of probabilistic models for fitting purposes.

$$z_{t+1}|z_t \sim \text{Categorical}(\sigma(\boldsymbol{V}\boldsymbol{x}_t))$$

$$\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, z_t \sim \mathcal{N}(\boldsymbol{A}_{z_t}\boldsymbol{x}_t + \boldsymbol{b}_{z_t} + \boldsymbol{B}\boldsymbol{u}_t, \boldsymbol{Q})$$

Since we are fixing the matrix $\boldsymbol{B}$, the controlled model does not include new parameters, but the space of possible parameters describing the data is now further constrained by the interventional data $\boldsymbol{u}_{1:T}$. Notice that we are not the first to propose this model and it is referred to as Input-Output Hidden Markov Model or IO-HMM in the statistics literature [Bengio]. However, in IO-HMM we assume that the control inputs are observed, without having a rule to determine the informative interventions; whereas here we use insights from computational neuroscience literature to propose a method for generating the control inputs given some knowledge about the underlying system.

Linear Quadratic Regulators (LQR) provide the analytical treatment of controlling linear dynamical systems. Given a LDS described by $\boldsymbol{x}_{t+1} = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{b} + \boldsymbol{B}\boldsymbol{u}_t$ the inputs $\boldsymbol{u}_{1:T}$ to control the state of the system $\boldsymbol{x}_T$ towards a pre-determined point $\boldsymbol{x}_f$ is given by the solution to the following optimization problem [Chow and others, 1975].

$$\mathscr{L}(\boldsymbol{u}_{1:T}) = \min_{\boldsymbol{u}_{1:T}} \sum_{t=1}^{T} \left(\boldsymbol{x}_t - \boldsymbol{x}_f\right)^T \boldsymbol{R} \left(\boldsymbol{x}_t - \boldsymbol{x}_f\right) + \gamma \boldsymbol{u}_t^T \boldsymbol{Q} \boldsymbol{u}_t$$

where the hyperparameter γ controls how the total energy of the input. If we assume that the underlying system is confined to evolve on a low-dimensional manifold in the high-dimensional space, locally there are only a few directions that can be explored by the system trajectory. Although we do not have access to those directions, but we can use a locally linear approximation of the dynamics to estimate those directions. To formalize this intuition and develop a concrete control rule accordingly, we consider an isotropic sphere with radius r around the initial point of the trajectory $\boldsymbol{x}_0$ denoted by $\mathscr{S}_r(\boldsymbol{x}_0)$, and aim to find the point that traverses the maximum distance after a fixed duration T_1. We call this point maximally deviant or `max-dev` and provide a derivation in the supplementary for linear systems. The extension of `max-dev` from LDS to SLDS is straightforward as shown in the supplementary. Controlling the state of the system towards `max-dev` point has two advantages, 1) it allows the system to explore a small neighborhood around the spontaneous manifold defined by the radius r and 2) the explored regions correspond to the "natural" directions of expansion in the original system. The former results in improving the switching fit up to certain pre-defined radius and the latter enables energy efficiency, meaning that instead of wasting the energy of control inputs along irrelevant dimensions, it focuses the energy of the input along the relevant directions. This is specifically important in high-dimensional systems where computational efficiency becomes a bottleneck. We summarize our proposed algorithm in 4.8.

4.3.3 Results

We applied our proposed CSLDS fitting algorithm to multiple systems, each of which is detailed below. In each case, we apply 3 strategies for generating a sequence of control inputs and appending the controlled data to the observations. The first strategy is using the LQR framework as demonstrated in Fig. 4.8, the second strategy is to inject random noise input with a standard deviation equal to r, and the third strategy is to use zero control input and continue appending spontaneous data to the observations. In each case we follow training log likelihood and test flow error defined by the difference between the ground truth and fitted flow fields in a neighbourhood around some randomly sampled trajectory from the system.

Ground Truth SLDS To investigate whether the proposed sequence of interventions and our CSLDS fitting procedure helps with identification, we generated signals from a 4-state SLDS described by the following parameters with $K = 4$.

$$\boldsymbol{A}_k = 0.99 \times \boldsymbol{R}(0.01\pi + \epsilon_k), \epsilon_k \sim \mathcal{N}(0, 0.003\pi) \quad \boldsymbol{b}_k = \left[\cos\left(\frac{k\pi}{K}\right), \sin\left(\frac{k\pi}{K}\right)\right]^T$$

$$\boldsymbol{B} = \boldsymbol{I}_2 \quad \boldsymbol{Q} = 10^{-5} \times \boldsymbol{I}_2 \quad \boldsymbol{V} = 10^6 \times \boldsymbol{b}_{1:K}$$

In Fig. 4.10 we show the difference in the flow fields fitted by LQR and other control strategies in the CSLDS framework.

Trained RNN for Cycling Task SLDS models provide approximations to more complex nonlinear systems, and provide a mechanistic understanding of the underlying system. Different linear pieces correspond to regions of the state space where the dynamics either demonstrate a limit cycle, stable, or unstable fixed points. Depending on the organization of the linear pieces, the fixed points can either be inside or outside of the corresponding region, nevertheless the state of the system will either move towards, repel away from those points in the case of stable and unstable fixed points. These intuitions can be translated into dynamical mechanisms governing the underlying nonlinear system. To investigate whether the CSLDS fitting approach can provide a mechanistic understanding of the underlying mechanism, we trained an RNN to generate a sequence of 3 or 5 sinusoidal cycles when the input in dimension 1 or 2 turns on respectively. This task is motivated by findings in [Russo et al., 2020] where the task solving RNNs are shown to rely on transient dynamics to generate the output. Given two sequences $(\boldsymbol{x}_{1:T}, \boldsymbol{y}_{1:T})$ the RNN is described by the following equations:

$$\boldsymbol{h}_{t+1} = f^{\mathrm{h}}_{\boldsymbol{\phi}}(\boldsymbol{h}_t) + f^{\mathrm{i}}_{\boldsymbol{\phi}}(\boldsymbol{x}_t)$$

where $\boldsymbol{h}$ denotes the hidden state of the RNN, $f^{\mathrm{h}}, f^{\mathrm{i}}, f^{\mathrm{o}}$ denote the hidden, input, and output functions respectively, and $\boldsymbol{\pi}$ denotes all the parameters of the RNN. Training the RNN is performed via Backprop Through Time (BPTT) minimizing the following loss:

$$\mathscr{L}(\boldsymbol{\phi}) = \sum_{i=1}^{T} \left\| \boldsymbol{y}_t - f^{\mathrm{o}}_{\boldsymbol{\phi}}(\boldsymbol{h}_t) \right\|_2^2$$

In Fig. 4.11 we show examples of inputs and outputs, test performance, the dynamics of the hidden state, and the result of the CSLDS fitting. We used a 20-dimensional hidden space for training the RNN. In this high dimensional setting, MaxDeviant outperforms two other strategies by large margins. We hypothesize that this happens because the dynamics in high dimensions evolve on a lower dimensional space and MaxDeviant strategy allows for exploring only those dimensions which leads to faster convergence of the test error in the vicinity of the observed trajectories.

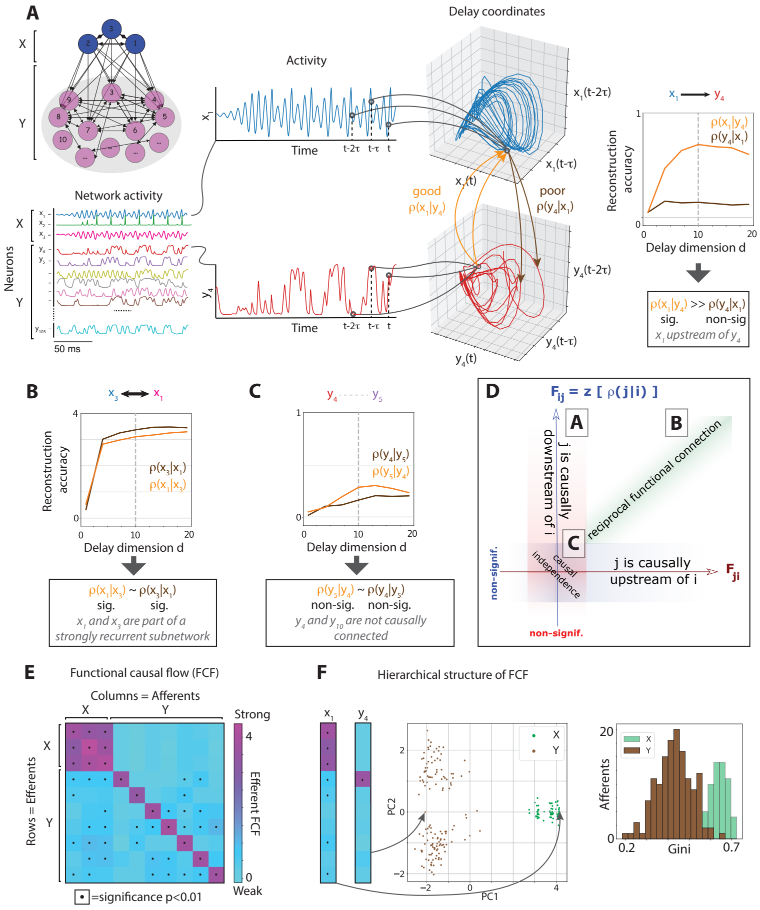

Figure 4.3: **Functional causal flow of a simulated rate network** A) Left: Schematic of network architecture Z: two subnetworks X (blue nodes) and Y (pink nodes) comprising strong and weak recurrent couplings, respectively, are connected via feedforward couplings from X to Y (thickness of black arrows represents the strength of directed structural couplings). Center: Activity of units $y_4(t)$ (orange, bottom) and $x_1(t)$ (in blue, top) are mapped to the delay coordinate space $X_1 = [x_1(t), x_1(t-\tau), \ldots, x_1(t-(d-1)\tau)]$and Y_4 (right, $\tau = 4$ms, confirmed by model selection. Reconstruction accuracy increases with delay vector dimension d before plateauing. The reconstruction accuracy $\rho(x_1|y_4)$ of upstream unit x_1 given the downstream unit y_4 is significant and larger than the reconstruction accuracy $\rho(y_4|x_1)$ of y_4 given x_1 (non-significant). The FCF value F_{41} reveals a strong and significant functional connectivity from upstream node x_1 to downstream node y_4.

Figure 4.3: (cont. from previous page) B) The significant FCF between two units x_1 and x_3 within the strongly coupled subnetwork X reveal strong and significant causal flow between them, but no preferred directionality of causal flow. C) The non-significant FCF between two units y_4 and y_5 in the weakly coupled subnetwork Y suggests the absence of a causal relationship. D) Summary of the FCF cases in panels A, B, C (see table 4.2). E) The FCF between 10 representative units sparsely sampled from the network (columns and rows represent afferent and efferent units, respectively; columns are sorted from functionally upstream to downstream units). F) The functional hierarchy in the network structure is encoded in the causal vectors (Left: PCA of columns of the FCF matrix, each dot represents one afferent, see Methods; Right: Gini coefficient of causal vectors).

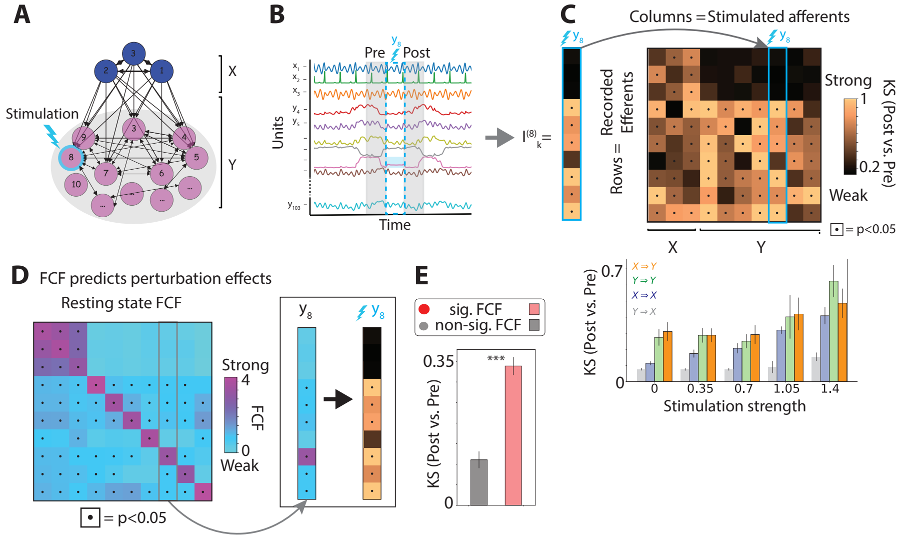

Figure 4.4: **Causal flow predicts perturbation effects** A) Perturbation protocol: single nodes are stimulated with a pulse of strength S lasting for 100ms (representative trials with stimulation of unit y_8). B) Perturbation effects on efferent units are estimated by comparing the activity immediately preceding onset and following offset of the perturbation (Kolmogorov-Smirnov test statistics, black dot represents significant effect, $p < 0.05$). The effects of stimulating one afferent i on all efferents k is encoded in the perturbation vector $\mathbf{I}^{(i)}$. C) Perturbation effects increase with the stimulation strength S for afferent-efferent pairs in populations $X \to X$, $Y \to Y$, and $X \to Y$, but not $Y \to X$, reflecting the absence of feedback structural couplings from Y to X (mean±s.e.m. across stimulations). D) For each afferent, its causal vector (column of the resting state FCF matrix representing unit y_4) is compared with the perturbation vector (columns of the interventional connectivity matrix), revealing that FCF predicts perturbation effects. E) Efferent units with significant resting state FCF had a larger response to perturbation, compared to pairs with non-significant FCF (red and gray bars, respectively; t-test, $*** = p < 10^{-6}$).

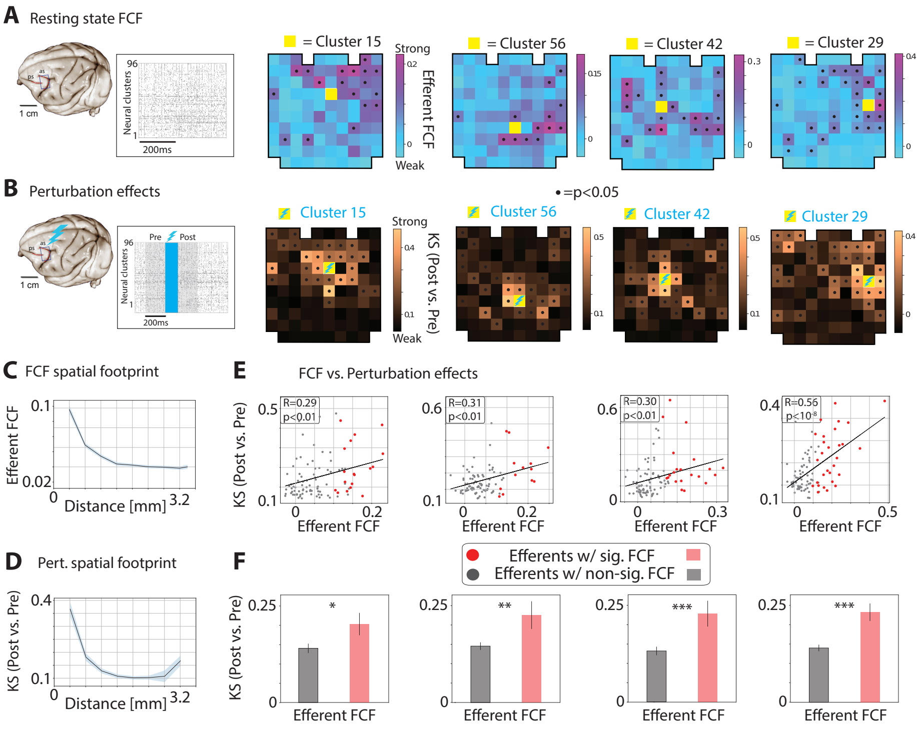

Figure 4.5: **Causal flow predicts perturbation effects in alert monkeys** A) Left: Ensemble spiking activity in representative session from multi-electrode array activity in the pre-arcuate gyrus during quiet wakefulness (black tick marks are spikes from each neural cluster, defined as the aggregated spiking activity around each recording electrode). Right: FCF inferred from resting periods for four representative afferent clusters (clusters 15 and 56 from session 1 and clusters 42 and 29 from session 2; yellow squares represent the reconstructed afferent cluster for each causal vector). FCF causal vectors for each afferent are overlaid to the array geometry (black dots represent significant FCF values, established by comparison with surrogate datasets, $p < 0.05$). B) Left: The perturbation effect from electrical microstimulation of cluster 15 (120ms stimulation train, blue shaded area) was estimated by comparing the activity in the 200ms intervals immediately preceding and following the perturbation (grey shaded areas). Right: Perturbation effects from four stimulated clusters (same clusters as in A) overlaid on the array geometry (Kolmogorov-Smirnov test statistics between post- vs. pre-perturbation activity distribution; black dots represent a significant difference, $p < 0.05$).

Figure 4.5: (cont. from previous page) C) The spatial footprint of resting state FCF decays with increasing distance of the efferent from the afferent cluster (mean±s.e.m. across 96 clusters in two sessions). D) Spatial footprint of perturbation effects for the four stimulated clusters decays with increasing distance from the stimulated cluster (mean±s.e.m. across four stimulated afferents). E) Resting state FCF predicts perturbation effects. For each stimulated afferent, the perturbation effects on all efferent clusters are shown (Kolmogorov-Smirnov test statistics between post- and pre-stimulation activity) as functions of the corresponding resting state FCF (gray and red dots represent efferents with non-significant and significant FCF, respectively, $p < 0.05$; black line: linear regression, R^2 and p-value reported). F) For each stimulated afferent in panel E, aggregated perturbation effects are larger over efferents with significant resting state FCF vs. efferents with non-significant FCF (mean±s.e.m. across gray and red-circled dots from panel E; t-test, $*, **, *** = p < 0.05, 0.01, 0.001$). G) After removing the spatial distance effects from the FCF and perturbation effects, the residual aggregated perturbation effects are larger over efferents with significant residual resting state FCF vs. efferents with non-significant residual FCF.

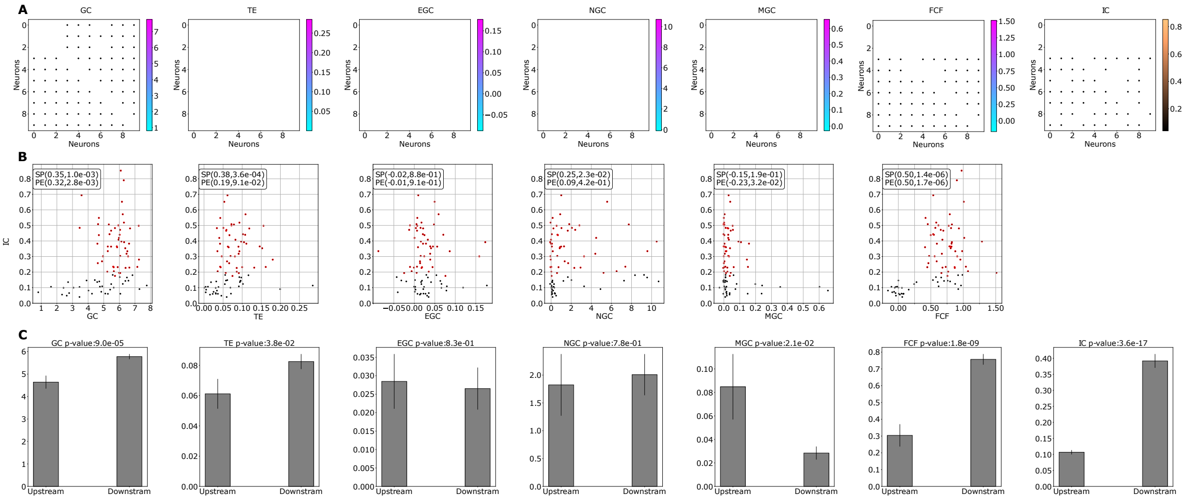

Figure 4.6: **Comparison of causality indices on the simulated rate network** A) Different causality indices applied to the simulated rate network of Fig. 4.3-4.4; from left: Granger Causality (GC), Transfer Entropy (TE), Extended Granger Causality (EGC), Nonlinear Granger Causality (NGC), Multivariate Granger Causality (MGC), Functional Causal Flow (FCF), Interventional Connectivity (IC). B) Scatter plots of correlations between measured causality index on the x-axis and interventional connectivity values on the y-axis (SP, PE represent Spearman and Pearson correlation coefficients and p-values, respectively). In this simulation FCF can best predict IC among the indices. C) Each bar plot corresponds to the causality index values separated according to the significance of IC matrix (t-test with respective p-values reported); FCF best reflects upstream vs. downstream as defined by the significant elements of IC. The rightmost bar plot corresponds to the IC values separated by their significance, providing a ceiling for the causality indices.

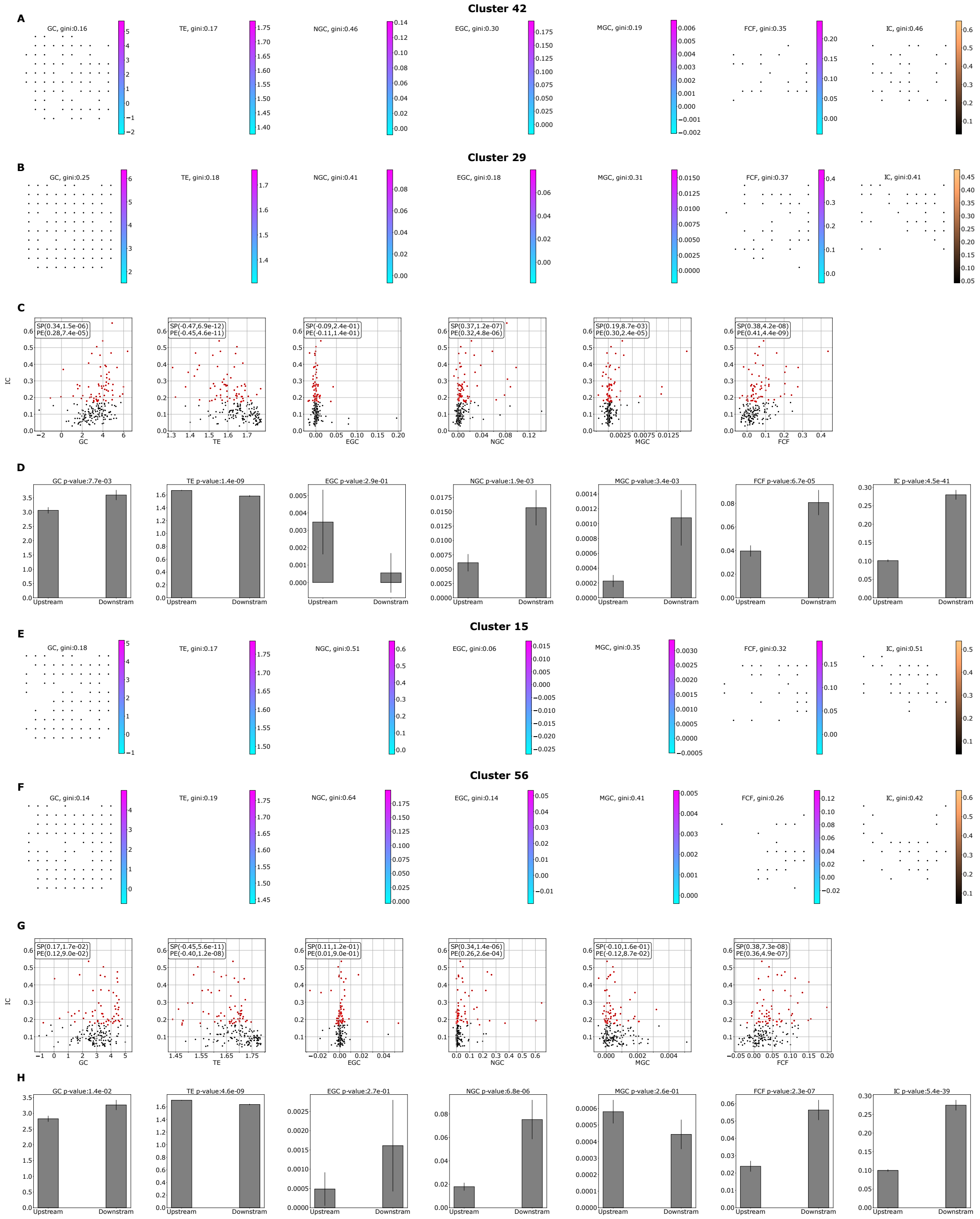

Figure 4.7: **Comparison of causality indices on the monkey data:** A, B, E, F) Causality indices computed on two stimulated channels during the resting activity (see Fig. 4.6 for notations);

Figure 4.7: (cont. from previous page) each image corresponds to the causality between the stimulated channel and all other channels organized in the physical layout of the electrode array (same as in Fig. 4.5; Gini index of causal vector reported on top). C, G) Scatter plots of correlations between measured causality index on the x-axis and interventional connectivity values on the y-axis (SP, PE represent Spearman and Pearson correlation coefficients and p-values, respectively), in this dataset FCF can best predict IC among the indices. D, H) Each bar plot corresponds to the causality index values separated according to the significance of IC matrix; FCF best reflects upstream vs. downstream as defined by the significant elements of IC. A-D and E-H correspond to the two different recording sessions, respectively.

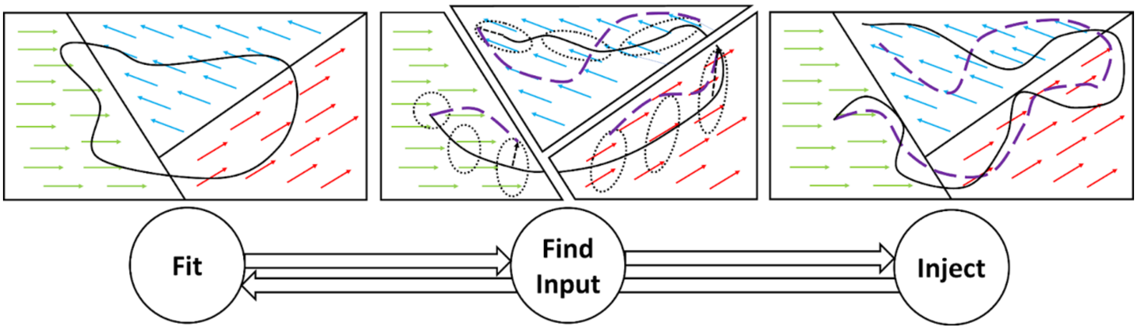

Figure 4.8: **Schematic of proposed CSLDS** Starting from a small sphere around the initial points, we analytically derive the shape of the deformed ellipsoid after time T_1 and use LQR to control the state of the system towards the `max-dev` point shown in green trajectory; red trajectory is non-driven trajectory of the initial point.

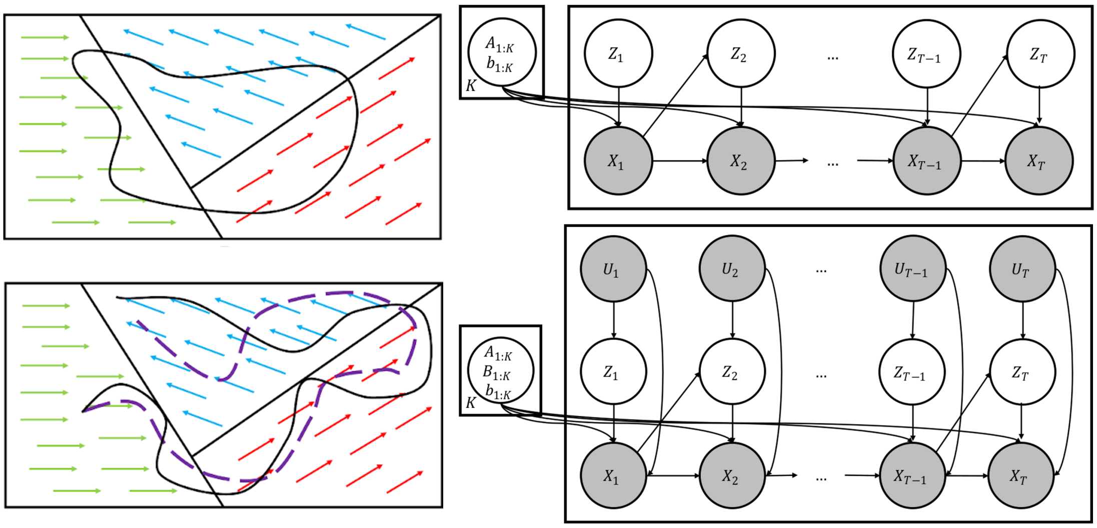

Figure 4.9: **Recurrent ARHMM and Recurrent IO-ARHMM Graphical Models** Observed (designed) input is injected to the model and system to tune the fit in the neighborhood of the observed trajectory.

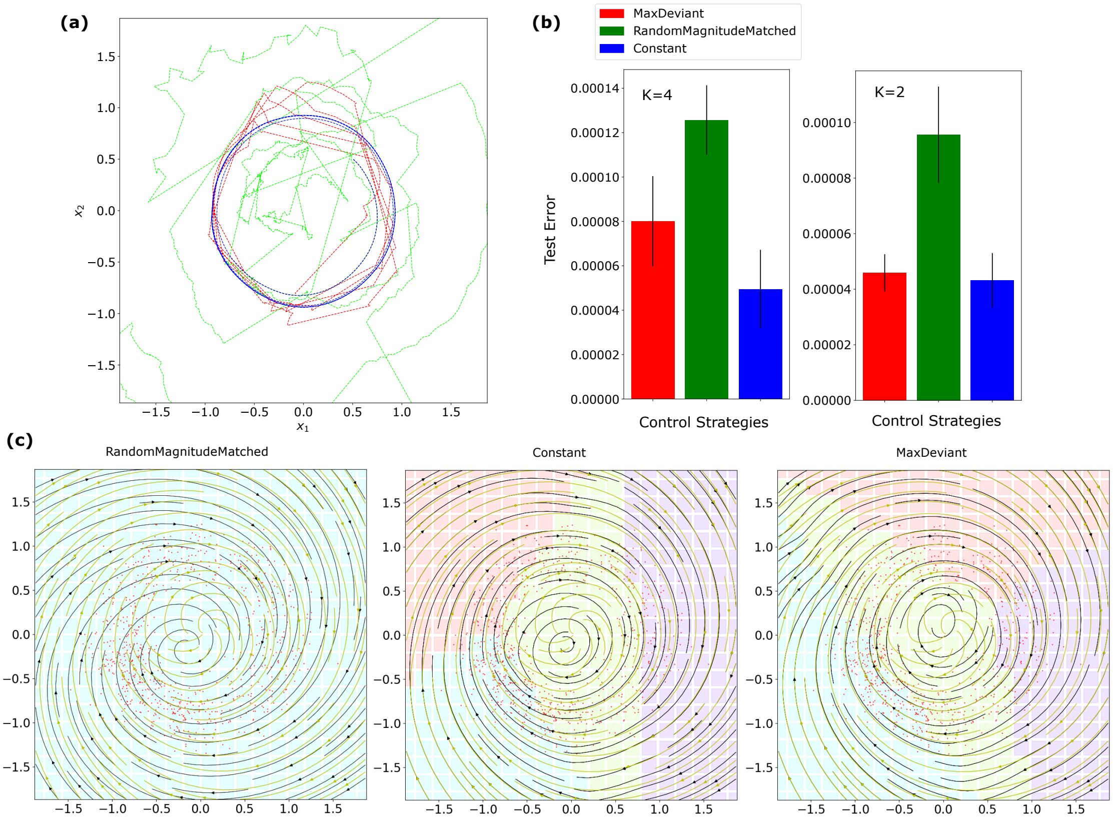

Figure 4.10: **Results on SLDS System** (a) Controlled trajectories generated by 3 control strategies, RandomMagnitudeMatched strategy explores parts of the state space that are far from the observed trajectory and "wastes" the pieces of SLDS in irrelevant regions whereas MaxDeviant leads to efficient exploration of the vicinity of the observed trajectory (b) Test flow error for three control strategies showing that MaxDeviant (proposed) and Constant input strategies achieve comparable performance whereas RandomMagnitudeMatched cannot estimate the underlying flow field. (c) Demonstration of fitted flow fields using 3 control strategies, black is the fitted flow field and yellow is the true flow field; colors correspond to the partitioning of the space given by the CSLDS model.

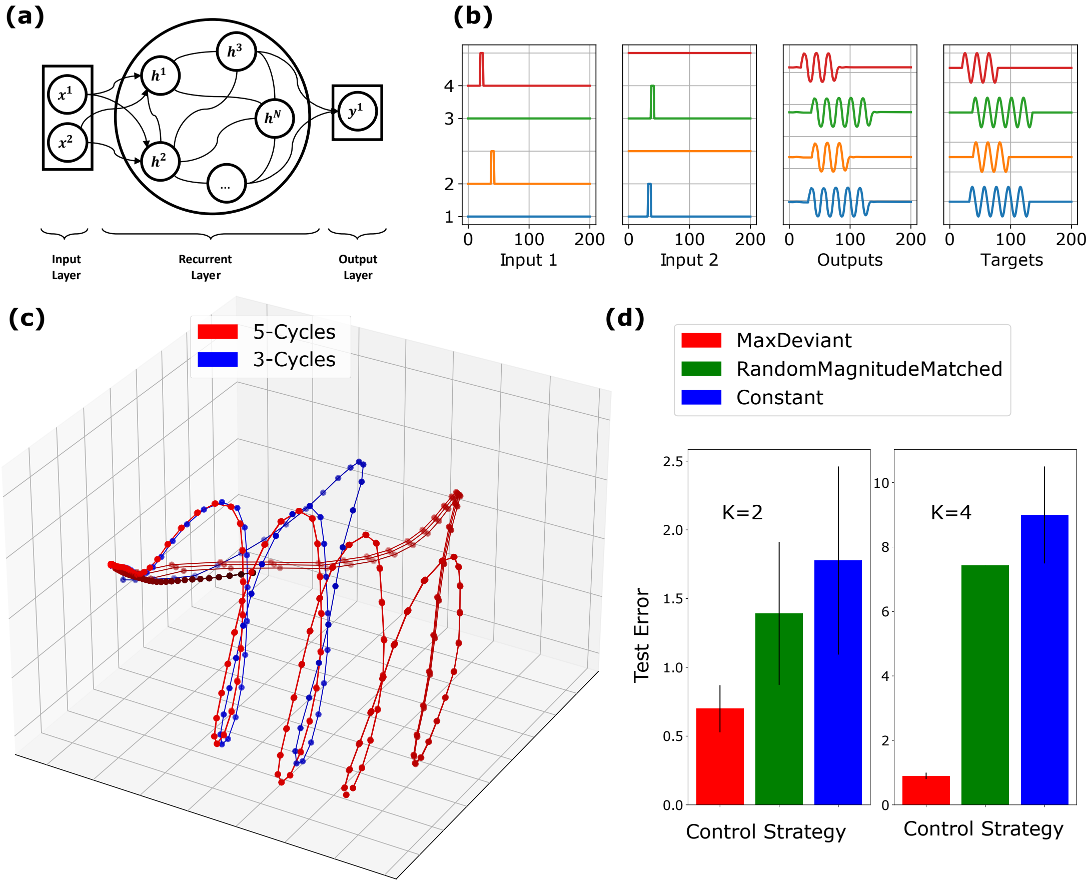

Figure 4.11: **Results on Trained RNN on Cycling Task** (a) The schematic of the architecture of RNNs used for training, the input is a 2-dimensional signal, we chose the hidden layer dimension to be 20 (N=20), the output is a sequence of 3 or 5 cycles generated when the input in dimension one or two turns on (b) A few examples of task inputs and targets and predictions made by the trained RNN, the RNN achieves test MSE near zero (c) PCA of the trajectories of hidden units; blue and red trajectories corresponds to 3 and 5 cycles; the trained RNN reuses the first 3 cycles for generating 5 cycles, consistent with [Russo et al., 2020] (d) Test flow error of fitted CSLDS to hidden dynamics decreases faster for MaxDeviant (proposed) through iterations of fitting compared to two other strategies.

CHAPTER 5

Measuring the Unique Information of Neurons

5.1 Introduction and Background

In neural systems, often multiple neurons are driven by one external event or stimulus; conversely multiple neural inputs can converge onto a single neuron. A natural question in both cases is how multiple variables hold information about the singleton variable. In their seminal work [Williams and Beer, 2010], Williams and Beer proposed an axiomatic extension of classic information theory to decompose the mutual information between multiple source variables and a single target variable in a meaningful way. For the case of two sources X_1, X_2, their partial information decomposition (PID) amounts to expressing the mutual information of X_1, X_2 with a target Y as a sum of four non-negative terms,

$$I(Y:(X_1,X_2)) = U(Y:X_1\backslash X_2) + U(Y:X_2\backslash X_1) + R(Y:(X_1,X_2)) + S(Y:(X_1,X_2)), \tag{5.1}$$

corresponding to unique (U_1, U_2), redundant (R) and synergistic (S) contributions, respectively. These terms should also obey the consistency equations

$$I(Y:X_1) = R(Y:(X_1,X_2)) + U(Y:X_1\backslash X_2), \tag{5.2}$$

$$I(Y:X_2) = R(Y:(X_1,X_2)) + U(Y:X_2\backslash X_1). \tag{5.3}$$

The PID has proved useful in understanding information processing by distributed systems in a diverse array of fields including machine learning [Tax et al., 2017; Wollstadt et al., 2021], earth science [Goodwell et al., 2020] and cellular automata [Flecker et al., 2011], and particularly in neuroscience [Kay et al., 2019; Pica et al., 2017; Timme et al., 2016; Wibral et al., 2015, 2017b], where notions of synergy and redundancy, traditionally considered mutually exclusive and distinguished by the sign of

$$\begin{aligned} \Delta &= I(Y:(X_1,X_2)) - I(Y:X_1) - I(Y:X_2), \\ &= S(Y:(X_1,X_2)) - R(Y:(X_1,X_2)), \end{aligned} \tag{5.4}$$

have long played a central role in the quest to understand how neural circuits integrate information from multiple sources [Brenner et al., 2000; Gat and Tishby, 1999; Quiroga and Panzeri, 2009; Schneidman et al., 2003]. The novelty of the PID framework here is in separating the measures of synergy and redundancy in (5.4).

The above abstract formulation of PID provides three equations for four unknowns, and only becomes operational once one of U_1, U_2, R, or S is defined. This has been done in [Bertschinger et al., 2014] via a definition of the unique information:

Definition 5.1.1 (BROJA [Bertschinger et al., 2014]). Given three random variables (Y,X_1,X_2) with joint probability density $p(y,x_1,x_2)$, the unique information U_1 of X_1 with respect to Y is

$$U(Y:X_1\backslash X_2) = \min_{q\in Q} I_q(Y:X_1|X_2), \tag{5.5}$$

$$= \min_{q\in Q} \int dy dx_1 dx_2\, q(y,x_1,x_2) \log\left(\frac{q(y,x_1|x_2)}{q(y|x_2)q(x_1|x_2)}\right), \tag{5.6}$$

where

$$Q = \{q(y,x_1,x_2) \,|\, q(y,x_i) = p(y,x_i), i = 1,2\}\,. \tag{5.7}$$

In words, we minimize the conditional mutual information $I(Y:X_1|X_2)$ over the space of density functions that preserve the marginal densities $p(y,x_1)$ and $p(y,x_2)$. The above definition implies, along with (5.2)-(5.3), that the unique and redundant information only depend on the marginals $p(y,x_1), p(y,x_2)$, and that the synergy can only be estimated from the full $p(y,x_1,x_2)$.

The original definition in [Bertschinger et al., 2014] was limited to discrete random variables. Here, we show that the extension to continuous variables is well-defined and can be practically estimated.

Motivation from decision theory [Bertschinger et al., 2014]. Consider for simplicity discrete variables. A decision maker DM$_1$ can choose an action a from a finite set $\mathscr{A}$, and receives a reward $u(a,y)$ based on the selected action and the state y, which occurs with probability $p(y)$. Notably, DM$_1$ has no knowledge of y, but observes instead a random signal x_1 sampled from $p(x_1|y)$. Choosing the action maximizing the expected reward for each x_1, his maximal expected reward is

$$R_1 = \sum_{x_1} p(x_1) \max_{a|x_1} \sum_{y} p(y|x_1) u(a,y)\,. \tag{5.8}$$

DM$_1$ is said to have no unique information about y w.r.t. another decision maker DM$_2$ that observes $x_2 \sim p(x_2|y)$ – if $R_2 \geq R_1$ for any set $\mathscr{A}$, any distribution $p(y)$, and any reward function $u(a,y)$. A celebrated theorem by Blackwell [Blackwell, 1951; Leshno and Spector, 1992] states that such a generic advantage by DM$_2$ occurs iff there exist a stochastic matrix $q(x_1|x_2)$ which satisfies

$$p(x_1|y) = \sum_{x_2} p(x_2|y) q(x_1|x_2)\,. \tag{5.9}$$

But this occurs precisely when the unique information (5.5) vanishes, since then there exists a joint distribution $q(y,x_1,x_2)$ in Q for which $y \perp x_1|x_2$, which implies $q(x_1|x_2,y) = q(x_1|x_2)$, and thus (5.9) holds. Similar results exist for continuous variables [Le Cam, 1996; Torgersen, 1991]. Thus the unique information from Definition 5.1.1 quantifies a departure from Blackwell's relation (5.9).

In this work we present a definition and a method to estimate the BROJA unique information for generic continuous probability densities. Our approach is based on the observation that the constraints (5.7) can be satisfied with an appropriate copula parametrization, and makes use of techniques developed to optimize variational autoencoders. We only consider one-dimensional Y,X_1,X_2 for simplicity, but the method can be naturally extended to higher dimensional cases. In Section 5.2 we review related works, in Section 5.3 we present our method and Section 5.4 contains several illustrative examples.

5.2 Related Work

Partial information decomposition offers a solution to a repeated question that was not addressed by 'classical' information theory regarding the relations between two sources and a target [Williams and Beer, 2010]. From a mathematical perspective a 'functional definition' has to be made, meaning that such a definition should align with our intuitive notions. Yet, as shown in [Bertschinger et al., 2013], not all intuitively desirable properties of a PID can be realized simultaneously. Thus, different desirable properties are chosen for distinct application scenarios. Thus, various proposals for decomposition measures are not seen as conflicting but as having different operational interpretations. For example, the BROJA approach used here builds on desiderata from decision theory, while other approaches appeal to game theory [Ince, 2017] or the framework of Kelly gambling [Finn and Lizier, 2018]. Yet other approaches use arguments from information geometry [Harder et al., 2013]. Other approaches assume agents receiving potentially conflicting or incomplete information about the source variables for the purpose of inference or decryption (see e.g. [Makkeh et al., 2021; Rauh, 2017]). In [Gutknecht et al., 2021] the authors separate the specific operational interpretations of PID measures from the general structure of information decomposition.

The actual computation of the BROJA unique information is non-trivial, even for discrete variables. Optimization methods exist for the latter case [Banerjee et al., 2018; Makkeh et al., 2017, 2018], and analytic solutions are only known when all the variables are univariate binary [Rauh et al., 2019]. For continuous probability densities, an earlier definition aligned with the BROJA measure was made by Barret [Barrett, 2015], but only applies to Gaussian variables. For Barret's measure, an analytic solution is known when $p(y,x_1,x_2)$ is a three-dimensional Gaussian density [Barrett, 2015], but does not generalize to higher dimensional Gaussians [Schamberg and Venkatesh, 2021].

5.3 Bounding and Estimating the Unique Information

We proceed in two steps. We first introduce a parametrization of the optimization space Q in (5.7) and then introduce and optimize an upper bound on the unique information.

5.3.1 Parametrizing the Optimization Space with Copulas

To characterize the optimization space Q in (5.5)-(5.7), it is convenient to recall that according to Sklar's theorem [Sklar, 1959], any n-variate probability density can be expressed as

$$p(x_1 \dots x_n) = p(x_1) \dots p(x_n) c(u_1 \dots u_n), \tag{5.10}$$

where $p(x_i)$ is the marginal and $u_i = F(x_i)$ is the CDF of each variable. The dependency structure among the variables is encoded in the function $c{:}[0,1]^n \to [0,1]$. This is a *copula* density, a probability density on the unit hypercube with uniform marginals [Joe, 1997],

$$\int_{[0,1]^{n-1}} \prod_{j=1, j\neq i}^{n} du_j \, c(u_1 \dots u_n) = 1 \quad \forall i \,. \tag{5.11}$$

Note that under univariate reparametrizations $z_i' = g(z_i)$, the u_i's and the copula c remain invariant. For an overview of copulas in machine learning, see [Elidan, 2013].

Proposition 4. Under the BROJA Definition 5.1.1 of unique information, all the terms of the partial information decomposition in (5.1)-(5.3) are independent of the univariate marginals $p(x_1), p(x_2), p(y)$, and only depend on the copula $c(u_y, u_1, u_2)$.

Proof. Expressing $q(y,x_1,x_2), q(x_1,x_2), q(y,x_2)$ via copula decompositions (5.10), and changing variables as $du_y = q(y)dy$, etc., the objective function in (5.6) becomes

$$I_q(Y : X_1 | X_2) \;=\; \int_{[0,1]^3} du_y du_1 du_2 \, c(u_y,u_1,u_2) \log\left(\frac{c(u_y,u_1,u_2)}{c(u_y,u_2)c(u_1,u_2)}\right). \tag{5.12}$$

Note that the copula of any marginal distribution is the marginal of the copula:

$$c(u_y,u_2) = \int_{[0,1]} du_1 \, c(u_y,u_1,u_2), \qquad c(u_1,u_2) = \int_{[0,1]} du_y \, c(u_y,u_1,u_2). \tag{5.13}$$

Thus the optimization objective and the unique information are independent of the univariate marginals. A similar result holds for the mutual information terms in the l.h.s. of (5.1)-(5.3).[1] It follows that none of the PID terms in (5.1)-(5.3) depend on the univariate marginals, and therefore all the PID terms are invariant under univariate reparametrizations of (y, x_1, x_2). ■

In order to parametrize the optimization space Q in (5.7) using copulas, consider the factorization

$$p(y,x_1,x_2) = p(x_1)p(y|x_1)p(x_2|y,x_1) \,. \tag{5.14}$$

[1] The connection between mutual information and copulas was discussed in [Calsaverini and Vicente, 2009; Ma and Sun, 2011].

Using the copula decomposition (5.10) for $n = 2$, the last two factors in (5.14) can be expressed as

$$p(y|x_1) = \frac{p(y,x_1)}{p(x_1)} = \frac{p(y)p(x_1)c(y,x_1)}{p(x_1)} = c(u_y,u_1)p(y), \tag{5.15}$$

and similarly

$$\begin{aligned} p(x_2|y,x_1) &= \frac{p(x_1,x_2|y)}{p(x_1|y)}, && (5.16) \\ &= c_{1,2|y}(u_{1|y},u_{2|y})p(x_2|y), && (5.17) \\ &= c_{1,2|y}(u_{1|y},u_{2|y})c(u_y,x_2)p(x_2), && (5.18) \end{aligned}$$

where we defined the conditional CDFs,

$$u_{i|y} = F(u_i|u_y) = \frac{\partial C(u_y,u_i)}{\partial u_y} \qquad i = 1,2 \tag{5.19}$$

and $C(u_y,u_i)$ is the CDF of $c(u_y,u_i)$. Note that the function $c_{1,2|y}(u_{1|y},u_{2|y})$ in (5.17) is not the conditional copula $c(u_1,u_2|u_y)$, but rather the copula of the conditional $p(x_1,x_2|y)$. Using expressions (5.15) and (5.18), the full density (5.14) becomes

$$p(y,x_1,x_2) = p(y)p(x_1)p(x_2)c(u_y,u_1,u_2), \tag{5.20}$$

where

$$c(u_y,u_1,u_2) = c(u_y,u_1)c(u_y,u_2)c_{1,2|y}(u_{1|y},u_{2|y}). \tag{5.21}$$

This is a simple case of the pair-copula construction of multivariate distributions [Aas et al., 2009; Bedford and Cooke, 2001; Czado, 2010], which allows to expand any n-variate copula as a product of (conditional) bivariate copulas.

Proposition 5. The copula of the conditional, $c_{1,2|y}(\cdot,\cdot)$, parametrizes the space Q in (5.7).

Proof. Since $q(y,x_i) = p(y,x_i)$ $(i = 1,2)$, the copula factors in

$$p(y,x_i) = p(y)p(x_i)c(u_y,u_i), \qquad i = 1,2 \tag{5.22}$$

are fixed in Q. Therefore, in the copula decomposition (5.21) for $q(y,x_1,x_2) \in Q$, only the last factor can vary in Q. Let us denote by θ the parameters of a generic parametrization for the copula $c_{1,2|y}(u_{1|y},u_{2|y})$. Since the latter is conditioned on u_y, the parameters can be taken as a function $\theta(u_y)$. It follows that the copula of q necessarily has the form

$$c_\theta(u_y,u_1,u_2) = c(u_y,u_1)\,c(u_y,u_2)\,c_{1,2|\theta(u_y)}(u_{1|y},u_{2|y}), \tag{5.23}$$

and the parameters of the function $\theta(u_y)$ are the optimization variables.[2] ■

5.3.2 Optimizing an Upper Bound

Inserting now the expression (5.23) into the objective function (5.12) we get

$$I[\theta] = \mathbb{E}_{c_\theta(u_y,u_1,u_2)} \log\left[c(u_y,u_1)c_{1,2|\theta(u_y)}(u_{1|y},u_{2|y})\right] - \mathbb{E}_{c_\theta(u_1,u_2)} \log c_\theta(u_1,u_2), \tag{5.24}$$

which is our objective function and satisfies the marginal constraints (5.7). Note that apart from the optimization parameters θ, it depends on the bivariate copulas $c(u_y,u_1)$ and $c(u_y,u_2)$ which should be estimated from the observed data. Given D observations $\{y^{(i)},x_1^{(i)},x_2^{(i)})\}_{i=1}^D$, we map each value to $[0,1]$ via the empirical CDFs of each coordinate (y,x_1,x_2). Computing the latter has a $O(D\log D)$ cost from sorting each coordinate and yields a data set $\{u_y^{(i)},u_1^{(i)},u_2^{(i)})\}_{i=1}^D$. The latter set is used to estimate copula densities $c(u_y,u_1)$ and $c(u_y,u_2)$ by fitting several parametric and non-parametric copula models [Nelsen, 2007], and choosing the best pair of models using the AIC criterion.[3] From the learned copulas we also get the conditional CDF functions $u_{i|y} = F(u_i|u_y)$ that appear in the arguments of the first term in (5.24).

A variational upper bound. Minimizing (5.24) directly w.r.t. θ is challenging because the second term depends on the copula marginal $c_\theta(u_1,u_2)$ which has no closed

[2] We note that in multivariate pair-copula expansions it is common to assume constant conditioning parameters θ [Nagler and Czado, 2016], but we do not make such a simplifying assumption.

[3] For this fitting/model selection step, we used the `pyvinecopulib` python package [Nagler and Vatter, 2020].

form, as it requires integrating (5.23) w.r.t. u_y. We introduce instead an inference distribution $r_\phi(u_y|u_1,u_2)$, with parameters ϕ, that approximates the conditional copula $c_\theta(u_y|u_1,u_2)$, and consider the bound

$$\log c_\theta(u_1,u_2) = \log \int du'_y\, c_\theta(u'_y,u_1,u_2) \geq \int du'_y\, r_\phi(u'_y|u_1,u_2) \log \frac{c_\theta(u'_y,u_1,u_2)}{r_\phi(u'_y|u_1,u_2)}, \quad (5.25)$$

which follows from Jensen's inequality and is tight when $r_\phi(u'_y|u_1,u_2) = c_\theta(u'_y|u_1,u_2)$. This expression gives an upper bound on $I_q[\theta]$, which can be minimized jointly w.r.t. (θ,ϕ).

A disadvantage of the bound (5.25) is that its tightness depends strongly on the expressiveness of the inference distribution $r_\phi(u'_y|u_1,u_2)$. This situation can be improved by considering a multiple-sample generalization proposed by [Burda et al., 2016],

$$\log c_\theta(u_1,u_2) \geq D_{A,\theta,\phi}(u_1,u_2) \equiv \mathbb{E}_{p(u_y^{(1)}\ldots u_y^{(A)})} \log \left[\frac{1}{A} \sum_{a=1}^{A} \frac{c_\theta(u_y^{(a)},u_1,u_2)}{r_\phi(u_y^{(a)}|u_1,u_2)} \right], \quad (5.26)$$

where the expectation is w.r.t. A independent samples of $r_\phi(u'_y|u_1,u_2)$. $D_{A,\theta,\phi}(u_1,u_2)$ coincides with the lower bound in (5.25) for $A=1$ and satisfies [Burda et al., 2016]

$$\begin{aligned} D_{A+1,\theta,\phi}(u_1,u_2) &\geq D_{A,\theta,\phi}(u_1,u_2), & (5.27)\\ \lim_{A\to\infty} D_{A,\theta,\phi}(u_1,u_2) &= \log c_\theta(u_1,u_2). & (5.28) \end{aligned}$$

Thus, even when $r_\phi(u'_y|u_1,u_2) \neq c_\theta(u'_y|u_1,u_2)$, the bound can be made arbitrarily tight for large enough A. Inserting (5.26) in (5.24), we get finally

$$I_q[\theta] \leq B_1[\theta] + B_2[\theta,\phi], \quad (5.29)$$

where

$$\begin{aligned} B_1[\theta] &= \mathbb{E}_{c_\theta(u_y,u_1,u_2)} \log\left[c(u_y,u_1) c_{1,2|\theta(u_y)}(u_{1|y},u_{2|y})\right], & (5.30)\\ B_2[\theta,\phi] &= -\mathbb{E}_{c_\theta(u_1,u_2)} D_{A,\theta,\phi}(u_1,u_2), & (5.31) \end{aligned}$$

and we minimize the r.h.s. of (5.29) w.r.t. (θ,ϕ). Low-variance estimates of the gradients to perform the minimization can be obtained with the reparametrization trick [Kingma and

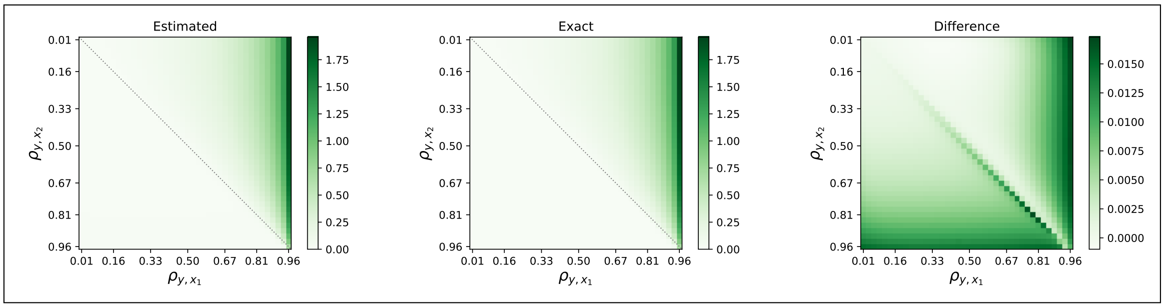

Figure 5.1: **Estimated vs. exact values of unique information for Gaussians** For a three-dimensional Gaussian, we show estimates of $U(Y:X_1\backslash X_2)$ as a function of the correlations $\rho_{y,x_i}(i=1,2)$, compared with the exact results from [Barrett, 2015]. Only for Gaussian distributions are exact results known for continuous variables.

Welling; Tucker et al., 2019]. In our examples below we use for $c_{1,2|\theta(u_y)}$ a bivariate Gaussian copula. Such a copula has just one parameter $\theta \in [-1,+1]$, and thus the optimization is done over the space of functions $\theta(u_y)$:$[0,1] \to [-1,+1]$, which we parametrize with a two-layer neural network. Similarly, we parametrize $r_\phi(u_y|u_1,u_2)$ with a two-layer neural network.

While the term B_2 in our bound is similar to the negative of the ELBO bound in importance weighted autoencoders (IWAEs) [Burda et al., 2016], there are some differences between the two settings, the most important being that we are interested in the precise value of the bound at the minimum, rather than the learned functions c_θ, r_ϕ. Note also that our latent variables $u_y^{(k)}$ are one-dimensional, as opposed to the usual higher dimensional latent distributions of variational autoencoders, and that the empirical expectation over data observations in IWAEs is replaced in B_2 by the expectation over $c_\theta(u_1,u_2)$, whose parameters are also optimized.

Estimating the other PID terms In the following we adopt the minimal value taken by the upper bound (5.29) as our estimate of U_1. The other terms in the partial information decomposition are obtained from the consistency relations (5.1)-(5.3), after estimating the mutual informations $I(Y:(X_1,X_2)), I(Y:X_1), I(Y:X_2)$. There are several methods for the latter. In our examples, we use the observed data to fit additional copulas $c(u_1,u_2)$ and $c_{12|\theta(u_y)}$ and estimate $I(Y:X_1) \simeq \frac{1}{D}\sum_{i=1}^{D} \log c(u_y^{(i)}, u_1^{(i)})$ and similarly for the other terms. Note that all our estimates have sources of potential bias. Firstly, the estimation of the parametric copulas is subject to model or parameter misspecification, which can be ameliorated by more refined model selection strategies. Secondly, the optimized bound might not saturate, biasing the estimate upwards. This can be improved using higher A values and improving the gradient-based optimizer used.

5.4 Examples

Comparison with exact results for Gaussians. Consider a three-dimensional Gaussian with correlations ρ_{y,x_i} between y, x_i for $i = 1,2$. The exact solution to (5.5) in this case is [Barrett, 2015]

$$U(Y:X_1 \backslash X_2) = \frac{1}{2}\log\left(\frac{1-\rho_{y,x_2}^2}{1-\rho_{y,x_1}^2}\right)\mathbb{1}\left[\rho_{y,x_2} < \rho_{y,x_1}\right]. \tag{5.32}$$

Fig. 5.1 compares the above expression with estimates from our method. Here we know that $c_{y,1}$ and $c_{y,2}$ are Gaussian copulas, with parameters $\rho_{y,x_1}, \rho_{y,x_2}$, and we assumed a Gaussian copula for $c_{1,2|y,\theta}(u_{1|y}, u_{2|y})$ as well. For each pair of values $\rho_{y,x_1}, \rho_{y,x_2}$. In this and the rest of the experiments, we optimized the parameters (θ, ϕ) using the ADAM algorithm [Kingma and Ba] with a fixed learning rate 10^{-2} during 1200 iterations, and using $A = 50$. The results reported correspond to the mean of the bound in the last 100 iterations. The comparison in Fig. 5.1 shows excellent agreement.

Model systems of three neurons. The nature of information processing of neural systems is a prominent area of application of the PID framework, since synergy has been proposed as natural measure of information modification [Lizier et al., 2013; Timme

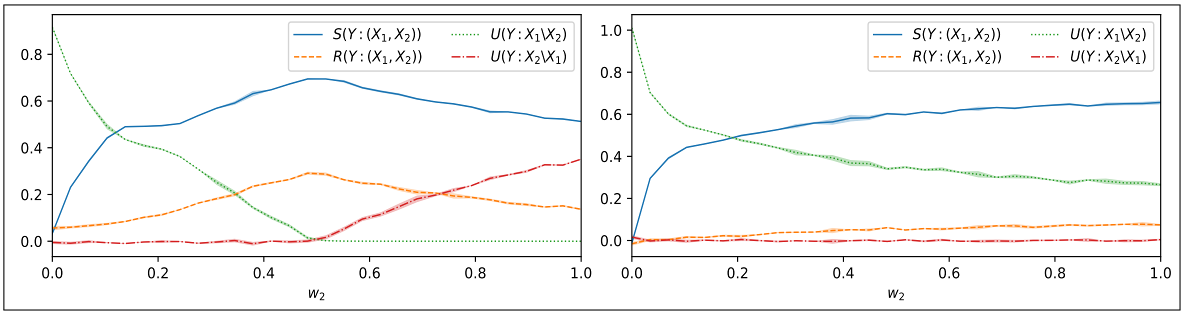

Figure 5.2: **Partial information decomposition for two neural network models.** In both models (5.33) we fixed $w_1 = 0.5, \rho_{12} = 0.3$, and show the PID terms as a function of the synaptic strength w_2, normalized by $I(Y : (X_1, X_2))$. We show mean (lines) and standard deviations (shaded area around each line) from 3 runs. *Left:* Model 1: The input of greatest weight conveys all the unique information, and synergy and redundancy both peak as $w_1 = w_2$. *Right:* Model 2: The second input X_2 has negligible unique information contribution, but its synaptic strength w_2 modulates the synergistic term, associated to the modification of information the neuron performs [Lizier et al., 2013].

et al., 2016]. We consider two models:

$$\begin{array}{cc} \boldsymbol{M}1 & \boldsymbol{M}2 \\ (X_1, X_2) \sim \mathcal{N}(0, \rho_{12}^2), & (X_1, X_2) \sim \mathcal{N}(0, \rho_{12}^2), \\ Y = \tanh(w_1 X_1 + w_2 X_2). & Y = X_1^2 / \left(0.1 + w_1 X_1^2 + w_2 X_2^2\right). \end{array} \tag{5.33}$$

Both models are parameterized by the correlation ρ_{12} and weights w_1, w_2. Model 1 is a particularly simple neural network. The tanh activation does not affect its copula, and even for a linear activation function the variables are not jointly Gaussian since Y is deterministic on (X_1, X_2). Model 2 is inspired by a normalization operation widely believed to be canonical in neural systems [Carandini and Heeger, 2011] and plays a role in common learned image compression methods [Ballé et al., 2017]. The results, presented in Figure 5.2. are obtained from 3000 samples from each model

Computational aspects of connectivity in recurrent neural circuits. We apply our continuous variable PID to understand computational aspects of the information processing between recurrently coupled neurons (Fig. 5.3). A large amount of work has been devoted to applying information theoretic measures for quantifying directed pairwise information transfer between nodes in dynamic networks and neural circuits [Reid et al., 2019]. However, classical information theory only allows for the quantification of information transfer, whereas the framework of PID enables further decomposition of information processing into transfer, storage, and modification, providing further insights into the computation within a recurrent system [Wibral et al., 2017a]. Transfer entropy (TE) [Schreiber, 2000b] is a popular measure to estimate the directed transfer of information between pairs of neurons [Novelli and Lizier, 2021; Vicente et al., 2011], and is sometimes approximated by linear Granger causality. Intuitively, TE between a process X and a process Y measures how much the past of X, X^-, can help to predict the future of Y, Y^+, accounting for its past Y^-. Although TE quantifies how much information is transferred between neurons, it does not shed light on the computation emerging from the interaction of X^- and Y^-. Simply put, the information transferred from X^- could enter Y^+, independently of the past state Y^-, or it could be fused in a non-trivial way with the information in the state in Y^-[Wibral et al., 2017a; Williams and Beer, 2011]. PID decomposes the TE into **modified transfer** (quantified by $S(Y^+{:}X^-,Y^-)$) and **unique transfer** (quantified by $U(Y^+{:}X^- \setminus Y^-)$) terms (see the Appendix for a proof):

$$TE(X \to Y) = I(Y^+{:}X^-|Y^-) = U(Y^+{:}X^- \setminus Y^-) + S(Y^+{:}X^-,Y^-).$$

Furthermore, the information kept by the system through time can be quantified by the **unique storage** (given by $U(Y^+{:}Y^- \setminus X^-)$) and **redundant storage** (given by $R(Y^+{:}X^-,Y^-)$) in PID [Lizier et al., 2013]. This perspective is a new step towards understanding how the information is processed in recurrent systems beyond merely detecting the direction functional interactions estimated by traditional TE methods. To explore these ideas, we simulated chaotic networks of rate neurons with an a-priori causal structure consisting of two sub-networks **X** and **Y** (Fig. 5.3a, see [Nejatbakhsh et al., 2020a] for more details on causal analyses of this network model). The sub-network

$\mathbf{X}$ is a Rossler attractor of three neurons obeying the dynamical equations:

$$\begin{cases} \dot{X}_1 = -X_2 - X_3 \\ \dot{X}_2 = X_1 + \alpha X_2 \\ \dot{X}_3 = \beta + X_3(X_1 - \gamma) \end{cases} \tag{5.34}$$

where $\{\alpha, \beta, \gamma\} = \{0.2, 0.2, 5.7\}$. There are 100 neurons in the sub-network $\mathbf{Y}$ from which we chose the first three, $Y_{1:3}$, to simulate the effect of unobserved nodes. Neurons within the sub-network Y obey the dynamical equations

$$\dot{Y} = -\lambda Y + 10 \tanh(J_{YX} X + J_{YY} Y) \tag{5.35}$$

where $J_{YX} \in \mathbb{R}^{100\times 3}$ has all its entries equal to 0.1, and J_{YY} is the recurrent weight matrix of the Y sub-network, sampled as zero-mean, independent Gaussian variables with standard deviation $g = 4$. No projections exist from the downstream sub-network $\mathbf{Y}$ to the upstream sub-network $\mathbf{X}$. We simulated time series from this network (exhibiting chaotic dynamics, see Fig. 5.3a) and estimated the PID as unique, redundant, and synergistic contribution of neuron i and neuron j at time t in shaping the future of neuron j at time $t+1$. For each pair of neurons $Z_i, Z_j \in \{X_{1:3}, Y_{1:3}\}$ we treated $(Z_i^t, Z_j^t, Z_j^{t+1})_{t=1}^T$ as iid samples[4] and ran PID on these triplets (i, j represent rows and columns in Fig. 5.3b-d). The PID uncovered the functional architecture of the network and further revealed non-trivial interactions between neurons belonging to the different sub-networks, encoded in four matrices: modified transfer S, unique transfer U_1, redundant storage R, and unique storage U_2 (details in Fig. 5.3d). The sum of the modified and unique transfer terms was found to be consistent with the TE (Fig. 5.3c, TE equal to $S + U_1$, up to estimation bias). The TE itself captured the network effective connectivity, consistent with previous results [Nejatbakhsh et al., 2020a; Novelli and Lizier, 2021].

[4] Note that the estimation of the PID from many samples of the triplets $(Z_i^t, Z_j^t, Z_j^{t+1})$ is operationally the same whether such triplets are iid or, as in our case, temporally correlated. This is similar to estimating expectations w.r.t. the equilibrium distribution of a Markov chain by using temporally correlated successive values of the chain. In both cases, the temporal correlations do not introduce bias in the estimator but can increase the variance.

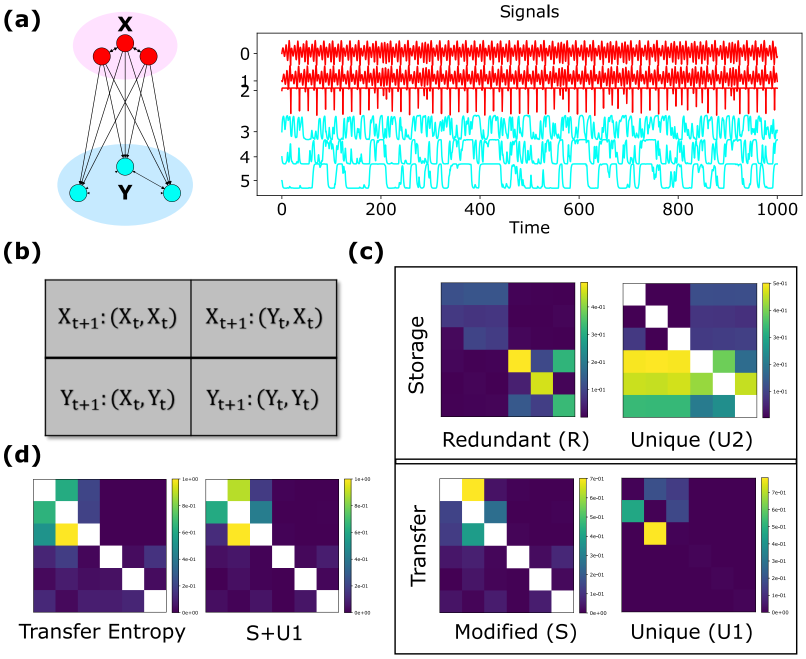

Figure 5.3: **PID uncovers the effective connectivity and allows for the quantification of storage, modification, and transfer of information in a chaotic network of rate neurons.** **a**: Schematics of recurrent network architecture (left) and representative activity (right). **b**: Schematic of the PID triplets for each 3×3 block of the matrices in c, d. **c:** PID decomposition into modified transfer S, unique transfer U_1, redundant storage R, and unique storage U_2 for the rate network. The future of X neurons only depends on unique information in the past of X neurons and their synergistic interactions. The interactions between the X and Y sub-networks only contain synergistic information regarding the future of Y but no redundant information; the latter is only present in the interactions confined within each sub-network. **d**: The transfer entropy (TE), estimated via IDTxl [Wollstadt et al., 2019], recovers the sum of modified and unique transfer terms $S + U_1$.

Uncovering a plurality of computational strategies in RNNs trained to solve complex tasks. A fundamental goal in neuroscience is to understand the computational mechanisms emerging from the collective interactions of recurrent neural circuits leading to cognitive function and behavior. Here, we show that PID opens a new window for assessing how specific computations arise from recurrent neural interactions. Unlike MI or TE, the PID quantifies the alternative ways in which a neuron determines the information in its output from its inputs, and thus can be a sensitive marker of

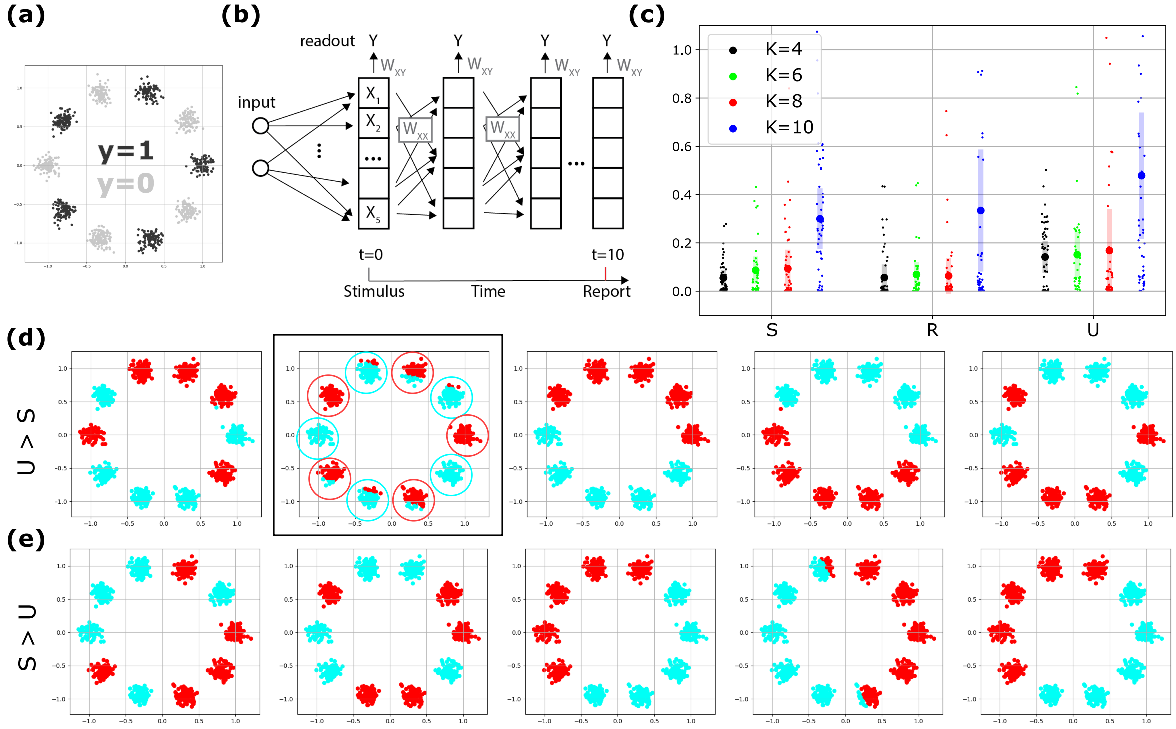

Figure 5.4: **PID of RNNs trained to solve generalized XOR problem a**: Input data drawn from a 2D Gaussian Mixture Model with K mixture components $X \sim \sum_{k=1}^{K} \frac{1}{K} \mathcal{N}(X|\mu_k, \sigma I)$ with means lying on the unit circle (grey and black dots represent the two class labels). **b**: Two layer network with 2D input layer, 5 recurrently connected hidden neurons X and one readout neuron Y; RNN activity unfolds in time (horizontal axis). The input is presented at time $t = 0$, then withdrawn, and the RNN is trained with BPTT to report the decision at $t = 10$. In this representation, layers correspond to time-steps and weights W_{XX} are shared between layers. **c**: PID between output $Y(t)$ and pairs of hidden neurons $X_i(t-1), X_j(t-1)$ for $t = 10$ yielding S, R, U_1, U_2 (distribution over 1000 input samples for each task K; 20 networks per task). Harder tasks led to an increase in PID measures. **d**: Example receptive fields for a network with $U > S$ shows emergence of grand-mother cells in the hidden layer (red and blue colors represent hidden neurons outputs; grandmother cell, second from left). **e**: Example receptive fields for a network with $S > U$, relying on higher synergy between neurons to solve the task.

different computational strategies. We here trained RNNs as models of cortical circuits [Mante et al., 2013] and used the PID to elucidate how the computations emerging from recurrent neural interactions contribute to task performance. We trained RNNs to solve a generalized version of the classic XOR classification problem with target labels corresponding to odd vs. even mixture components (Fig. 5.4a). Stimuli were presented for one time step ($t = 0$) and the network was trained to report the decision at $t = 10$. By tracking the temporal trajectories of the hidden layer activity we found that the network recurrent dynamics (represented as unfolded in time in Fig. 5.4b) progressively pulls the two input classes in opposite directions along the output weights (see Appendix). We used PID to dissect how a plurality of different strategies emerge from recurrent neural interactions in RNNs trained for solving a classification task. The computation emerged from the recurrent interaction between hidden neurons at different time steps. Do all successfully trained networks have a similar profile in terms of the PID terms? If so, this hints at a single computational strategy across these networks. If not, it is safe to assume that task performance is reached via different mechanisms, despite identical network architecture and training algorithm.

We found that on average across multiple networks S, R, and U rose with task difficulty (Fig. 5.4c), yet at all difficulties, individual networks differed strongly with respect to the ratio S/U, i.e. there were networks with larger average synergy across neuron pairs compared to the average unique information, and vice versa. For simple networks like the ones used here, one can inspect receptive fields to understand the reason for this differential behaviour (Fig. 5.4d-e). Indeed, networks with high average unique information displayed 'grandmother-cell'-like neurons, that would alone classify a large parts of the sample space, while in networks with higher average synergy such cells were absent (Fig. 5.4d). The emergence of these 'grandmother-cell'-like receptive fields is due to the recurrent dynamics. While in a feedforward architecture ($W_{XX} = 0$) hidden layer receptive fields are captured by hyperplanes in input space, in the RNN the receptive fields are time dependent, where later times are interpreted as deeper layers (Fig. 5.4b) and thus can capture highly non-linear features in input space. The advantage of PID versus a manual inspection of receptive fields is twofold: First, the PID framework abstracts and generalizes descriptions of receptive fields as being e.g. 'grandmother-cell'-like; thus the concept of unique information stays relevant even in scenarios where the concept of a receptive field becomes meaningless, or inaccessible. Second, the quantitative outcomes of a PID rest only on information theory, not specfic assumptions about neural coding or computational strategies, and can be obtained for large numbers of neurons.

Comparison of our PID-based approach with the concept of neuronal selectivity used in neuroscience highlights interesting similarities and differences. Several kinds of selectivity (pure, mixed linear, and mixed non-linear) can be identified by performing regression analysis of neural responses vs. task variables [Rigotti et al., 2013]. In this framework, our grand-mother cells correspond to neurons with pure selectivity to the input class labels (a.k.a. "choice-selective" neurons). In the XOR task, [Rigotti et al., 2013] showed that non-linear mixed selectivity of neurons to the class labels is beneficial when solving the XOR task, by leading to a high-dimensional representation of the task variables. While selectivity profiles are a property of single neuron responses to task variables, our PID measures are a property of the combined activity of triplets of neurons and thus reveal emerging functional interactions between units and their computational algorithms (see also [Timme et al., 2016] and [Wibral et al., 2017a]). This allowed us to characterize a functional property of neural systems less studied than task variable selectivity: the computations that require functional mixing of the information from multiple units (measured by the average synergistic information) vs. the computations that rely on the output of individual neurons (measured by the unique information and described as grandmother cells). Concretely, by comparing PID and receptive fields we found that that in networks with high unique information, neurons typically have receptive fields with pure selectivity (grandmother cells, with large unique information to the class labels). In networks with high synergy, neurons show complex mixed selectivity to class labels.

Appendix A

Conclusions

Large scale experimentation and massive datasets in neuroscience present challenges and opportunities to the computational neuroscience community. We need to build models that make use of the right information, but also leave room for new discoveries. This is the case both for models that extract signals from raw data and models that aim to make sense of those signals. In this thesis I presented a number of ideas from computer vision, statistics, machine learning, and dynamical systems aiming at automating the repeating steps of data analysis that do not require expert supervision.

In chapter 1, we introduced a general probabilistic framework to compute statistical atlases in novel imaging datasets in model organisms such as *C. elegans* and fruit flies. As new imaging modalities emerge to capture different views of nervous systems of model animals, we expect that the flexibility of our framework will be valuable in generating common coordinate spaces for downstream analyses. Involving more complex motion models, neural network architectures and different loss functions are important future directions of our framework.

In chapter 2, we expanded on the linear regression without correspondence model [Abid et al., 2017; Hsu et al., 2017; Pananjady et al., 2016; Unnikrishnan et al., 2018] to account for missing data and outliers. Furthermore, we provided several exact and approximate algorithms for the recovery of regression coefficients under noiseless and noisy regimes. The proposed algorithms are combinatorial at worst with variable dimensions. However, randomization procedures make the average-case complexity in constant dimension tractable given enough tolerance for failure. We provided several theoretical guarantees for exact recovery and running time complexity. A future algorithmic direction is to employ branch and bound techniques found in [Tsakiris and Peng, 2019] to reduce the computational complexity of the brute force nature of the algorithms. We then introduced algorithms for segmentation and tracking neurons in images and videos of worms. A notable limitation of our approach is that at least one annotated frame is required. We hope to mitigate this issue through future key upgrades. For example, we hope to use an object detection algorithm to automatically annotate the images, where identity-classification is not necessary [Meijering, 2012; Schneider et al., 2012; Spilger et al., 2020; Weigert et al., 2020; Wu et al., 2021].

In chapter 3, we considered the problem of extracting and demixing calcium signals from microscopy videos of *C. elegans*. We developed an extension of NMF, with a nonlinear motion model applied to the spatial cellular footprints, to deform the static image of these cells, modeling the worm's posture at each time frame. We provided different parameterizations for the spatial footprints and described regularizations that can help in finding smooth trajectories and signals. We further showed that our method outperforms state-of-the-art models that use a two-step process for motion stabilization/tracking and signal extraction. Finally, we demonstrated the effectiveness of our model by extracting calcium signals from videos of semi-immobilized *C. elegans*. In this chapter we focused on nuclear-localized calcium imaging in semi-immobilized *C. elegans*. We believe that a similar approach will be useful with other indicators [Chen et al., 2020] and in other preparations, e.g. larval zebrafish [Vanwalleghem et al., 2018], Drosophila [Schaffer et al., 2020],and Hydra [Szymanski and Yuste, 2019]; see, in particular, the preprint by [Lagache et al., 2020a], who develop improved tracking methods that may nicely complement the dNMF approach. We look forward to exploring these directions further in future work.

In chapter 4, we first introduced interventional connectivity and functional causal flow as its functional counterpart and demonstrated a new framework for predicting the effect of perturbations to a cortical circuit based solely on the causal interactions within a circuit inferred from sparsely recorded spiking activity at rest. A limitation of FCF stems from the fact that neuronal activity in frontal areas likely receives time-varying input from several other cortical and subcortical areas. These contextual effects might present a potential challenge when generalizing FCF predictions across different conditions (such as resting vs. task engaged sessions). It is an interesting open question to estimate how FCF may generalize across different behavioral conditions and we hope to report on this in the future. Next in chapter 4 we introduced controlled switching linear dynamical systems as a method to causally interrogate RNNs. While we focused on toy examples in this thesis extending the methods to larger systems as well as real biological circuits are interesting future works to explore.

Finally in chapter 5, we presented a partial information decomposition measure for continuous variables with arbitrary probability densities, thereby extending the popular BROJA PID measure for discrete variables. Extending PID measures to continuous variables drastically broadens the possible applications of the PID framework. This is important as the latter provides key insights into the way a complex system represents and modifies information in a computation – via asking which variables carry information about a target uniquely (such that it can only be obtained from that variable), redundantly, or only synergistically with other variables. Answering these questions is pivotal to understanding distributed computation in complex systems in general, and neural coding in particular. We believe that the methods presented here will allow PIDs to be extended efficiently in neuroscience for multiple continuous sources with potentially complex dependency structures, as would be common in cellular imaging data or activation properties of brain modules or areas in functional imaging.

A.1 Contributions

The contributions to specific papers are clarified in the following tables.

Table A.1: NeuroPAL [Yemini et al., 2021] Contributions

	EY	AL	AN	EV	RS	GM	AS	LP	VV	OH
Conceptualizatio	■	■	■	■	■	■	■	■	■	■
Data Collection	■	■							■	
Data Analysis			■	■	■	■				
Code Development			■	■	■	■				
Writing	■						■	■	■	■
Editing		■					■	■	■	■

Table A.2: StatAtlas [Varol et al., 2020] Contributions

	EV	AN	RS	GM	EY	OH	LP
Conceptualizatio	■	■			■		■
Data Collection					■		
Data Analysis	■	■	■	■			
Code Development	■	■					
Writing	■				■	■	■
Editing	■	■	■	■	■	■	■

Table A.3: dNMF [Nejatbakhsh et al., 2020c] Contributions

	AN	EV	EY	VV	AS	LP
Conceptualizatio	■	■	■			■
Data Collection			■	■		
Data Analysis	■					
Code Development	■					
Writing	■	■		■	■	■
Editing	■	■		■	■	■

Table A.4: MaleAtlas [Tekieli et al., 2021] Contributions

	TT	EY	AN	CW	EV	RF	NM	LP	OH
Conceptualizatio	■	■	■		■			■	■
Data Collection	■	■							
Data Analysis	■		■				■		
Code Development			■						
Writing									■
Editing	■	■		■		■			

Table A.5: DeformableAtlas Contributions

	AN	ND	EV	EY	OH	LP
Conceptualizatio	■		■	■		■
Data Collection				■		
Data Analysis	■	■				
Code Development	■					
Writing	■		■			■
Editing	■	■	■	■	■	■

Table A.6: RRWOC [Nejatbakhsh and Varol, 2021] Contributions

	AN	EV
Conceptualizatio	■	■
Data Collection		
Data Analysis	■	■
Code Development	■	■
Writing		■
Editing	■	

Table A.7: SinkhornEM [Nejatbakhsh et al., 2020b] Contributions

	AN	**EV**	**EY**	**OH**	**LP**
Conceptualizatio	■	■			■
Data Collection			■	■	
Data Analysis	■				
Code Development	■				
Writing	■	■	■		■
Editing	■	■	■	■	■

Table A.8: FCF [Nejatbakhsh et al., 2020a] Contributions

	AN	**FF**	**SE**	**TT**	**RK**	**LM**
Conceptualizatio	■	■		■	■	■
Data Collection			■		■	
Data Analysis	■	■				
Code Development	■	■				
Writing	■	■				■
Editing	■	■	■	■	■	■

Table A.9: CSLDS Contributions

	AN	**MB**	**LL**
Conceptualizatio	■		■
Data Collection			
Data Analysis	■		
Code Development	■		
Writing	■	■	■
Editing	■	■	■

Table A.10: Zephir [Yu et al., 2022] Contributions

	JY	**AN**	**MT**	**SG**	**MS**	**JK**	**LP**	**VV**
Conceptualizatio	■	■					■	■
Data Collection			■		■			■
Data Analysis	■							
Code Development	■							
Writing	■	■					■	■
Editing	■	■					■	■

Table A.11: PID [Pakman et al., 2021] Contributions

	AP	**AN**	**DG**	**AM**	**LM**	**MW**	**ES**
Conceptualizatio	■					■	■
Data Collection	■						
Data Analysis	■	■					
Code Development	■	■					
Writing	■	■			■	■	■
Editing	■	■	■	■	■	■	■

Bibliography

K. Aas, C. Czado, A. Frigessi, and H. Bakken.
Pair-copula constructions of multiple dependence.
Insurance: Mathematics and economics, 44(2):182–198, 2009.

A. Abadie, A. Diamond, and H. Jens.
Synthetic control methods for comparative case studies: Estimating the effect of California's Tobacco control program.
Journal of the American Statistical Association, 105(490):493–505, 2010.
ISSN 01621459.
doi: 10.1198/jasa.2009.ap08746.

A. Abid, A. Poon, and J. Zou.
Linear regression with shuffled labels.
arXiv preprint arXiv:1705.01342, 2017.

S. J. Aerni, X. Liu, C. B. Do, S. S. Gross, A. Nguyen, S. D. Guo, F. Long, H. Peng, S. S. Kim, and S. Batzoglou.
Automated cellular annotation for high-resolution images of adult caenorhabditis elegans.
Bioinformatics (Oxford, England), 29(13):i18–i26, Jul 2013.
ISSN 1367-4811.
doi: 10.1093/bioinformatics/btt223.
URL `https://pubmed.ncbi.nlm.nih.gov/23812982`.
23812982[pmid].

S.-R. Afraz, R. Kiani, and H. Esteky.
Microstimulation of inferotemporal cortex influences face categorization.
Nature, 442(7103):692–695, 2006.

A. Agarwal, D. Shah, and D. Shen.
Synthetic Interventions.

2020.
URL `http://arxiv.org/abs/2006.07691`.

M. B. Ahrens, M. B. Orger, D. N. Robson, J. M. Li, and P. J. Keller.
Whole-brain functional imaging at cellular resolution using light-sheet microscopy.
Nature Methods, 10(5):413–420, 2013a.
ISSN 1548-7105.
doi: 10.1038/nmeth.2434.

M. B. Ahrens, M. B. Orger, D. N. Robson, J. M. Li, and P. J. Keller.
Whole-brain functional imaging at cellular resolution using light-sheet microscopy.
Nature methods, 10(5):413–420, 2013b.

D. Aiger, N. J. Mitra, and D. Cohen-Or.
4-points congruent sets for robust pairwise surface registration.
In *ACM transactions on graphics (TOG)*, volume 27, page 85. Acm, 2008.

Alberto Abadie.
Using Synthetic Controls: Feasibility, Data Requirements, and Methodological Aspects.
Journal of Economic Literature, 59(2):391–425, 2021.

N. Ancona, D. Marinazzo, and S. Stramaglia.
Radial basis function approach to nonlinear granger causality of time series.
Physical Review E, 70(5):056221, 2004.

F. D. Andilla and F. A. Hamprecht.
Learning multi-level sparse representations.
In *Advances in Neural Information Processing Systems*, pages 818–826, 2013.

F. D. Andilla and F. A. Hamprecht.
Sparse space-time deconvolution for calcium image analysis.
In *Advances in neural information processing systems*, pages 64–72, 2014.

C. Archambeau, J. A. Lee, M. Verleysen, et al.
On convergence problems of the em algorithm for finite gaussian mixtures.
In *ESANN*, volume 3, pages 99–106, 2003.

J. Ashburner and K. J. Friston.
Voxel-based morphometry—the methods.
Neuroimage, 11(6):805–821, 2000.

M. A. Audette, F. P. Ferrie, and T. M. Peters.
An algorithmic overview of surface registration techniques for medical imaging.
Medical image analysis, 4(3):201–217, 2000.

J. Ballé, V. Laparra, and E. P. Simoncelli.
End-to-end optimized image compression.
ICLR, 2017.

P. K. Banerjee, J. Rauh, and G. Montúfar.
Computing the unique information.
In *ISIT*, 2018.

G. Barbera, B. Liang, L. Zhang, C. R. Gerfen, E. Culurciello, R. Chen, Y. Li, and D.-T. Lin.
Spatially compact neural clusters in the dorsal striatum encode locomotion relevant information.
Neuron, 92(1):202–213, 2016.

M. M. Barr, L. R. García, and D. S. Portman.
Sexual dimorphism and sex differences in caenorhabditis elegans neuronal development and behavior.
Genetics, 208(3):909–935, 2018.

A. B. Barrett.
Exploration of synergistic and redundant information sharing in static and dynamical Gaussian systems.
Physical Review E, 91(5), 2015.

J.-C. Bazin, Y. Seo, R. Hartley, and M. Pollefeys.
Globally optimal inlier set maximization with unknown rotation and focal length.
In *European Conference on Computer Vision*, pages 803–817. Springer, 2014.

E. F. Beckenbach and R. Bellman.
Inequalities, volume 30.
Springer Science & Business Media, 2012.

T. Bedford and R. M. Cooke.
Probability density decomposition for conditionally dependent random variables modeled by vines.
Annals of Mathematics and Artificial intelligence, 32(1-4):245–268, 2001.

Y. Bengio.
An Input Output HMM Architecture.

N. Bertschinger, J. Rauh, E. Olbrich, and J. Jost.
Shared information—new insights and problems in decomposing information in complex systems.
In T. Gilbert, M. Kirkilionis, and G. Nicolis, editors, *Proceedings of the European Conference on Complex Systems 2012*, pages 251–269, Cham, 2013. Springer International Publishing.
ISBN 978-3-319-00395-5.

N. Bertschinger, J. Rauh, E. Olbrich, J. Jost, and N. Ay.
Quantifying unique information.
Entropy, 16(4):2161–2183, 2014.

P. J. Besl and N. D. McKay.
Method for registration of 3-d shapes.
In *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, pages 586–607. International Society for Optics and Photonics, 1992.

S. Beucher and C. Lantuejoul.
Use of Watersheds in Contour Detection, 1979.
URL `http://www.citeulike.org/group/7252/article/4083187`.

T. Binzegger, R. J. Douglas, and K. A. Martin.
A quantitative map of the circuit of cat primary visual cortex.
Journal of Neuroscience, 24(39):8441–8453, 2004.

D. Blackwell.
Comparison of experiments.
Proc. 2nd Berkeley Symp. Math. Stats. and Probability, 1951.

V. Braitenberg and A. Schüz.
Anatomy of the cortex: statistics and geometry, volume 18.
Springer Science & Business Media, 2013.

J. Brehmer, P. de Haan, P. Lippe, and T. Cohen.
Weakly supervised causal representation learning.
pages 1–20, 2022.
URL `http://arxiv.org/abs/2203.16437`.

N. Brenner, S. P. Strong, R. Koberle, W. Bialek, and R. R. d. R. v. Steveninck.
Synergy in a neural code.
Neural computation, 12(7):1531–1552, 2000.

G. Bubnis, S. Ban, M. D. DiFranco, and S. Kato.
A probabilistic atlas for cell identification, 2019.

Y. Burda, R. Grosse, and R. Salakhutdinov.
Importance weighted autoencoders.
ICLR 2016, 2016.

A. P. Bustos, T.-J. Chin, F. Neumann, T. Friedrich, and M. Katzmann.
A practical maximum clique algorithm for matching with pairwise constraints.
arXiv preprint arXiv:1902.01534, 2019.

M. Cabezas, A. Oliver, X. Lladó, J. Freixenet, and M. B. Cuadra.
A review of atlas-based segmentation for magnetic resonance brain images.
Computer methods and programs in biomedicine, 104(3):e158–e177, 2011.

R. S. Calsaverini and R. Vicente.
An information-theoretic approach to statistical dependence: Copula information.
EPL (Europhysics Letters), 88(6):68003, 2009.

J. Cao, J. S. Packer, V. Ramani, D. A. Cusanovich, C. Huynh, R. Daza, X. Qiu, C. Lee, S. N. Furlan, F. J. Steemers, et al.
Comprehensive single-cell transcriptional profiling of a multicellular organism.
Science, 357(6352):661–667, 2017.

M. Carandini and D. J. Heeger.
Normalization as a canonical neural computation.
Nat. Rev. Neurosci., 13(1):51–62, Nov. 2011.

M. Casdagli, S. Eubank, J. D. Farmer, and J. Gibson.
State space reconstruction in the presence of noise.
Physica D: Nonlinear Phenomena, 51(1-3):52–98, 1991.

S. Chaudhary, S. A. Lee, Y. Li, D. S. Patel, and H. Lu.
Automated annotation of cell identities in dense cellular images.
bioRxiv, 2020.
doi: 10.1101/2020.03.10.986356.

URL https://www.biorxiv.org/content/early/2020/03/11/2020.03.10.986356.

S. Chaudhary, S. A. Lee, Y. Li, D. S. Patel, and H. Lu.
Graphical-model framework for automated annotation of cell identities in dense cellular images.
eLife, 10:1–108, 2021.
ISSN 2050084X.
doi: 10.7554/eLife.60321.

Y. Chen, G. Rangarajan, J. Feng, and M. Ding.
Analyzing multiple nonlinear time series with extended granger causality.
Physics letters A, 324(1):26–35, 2004.

Y. Chen, H. Jang, P. W. Spratt, S. Kosar, D. E. Taylor, R. A. Essner, L. Bai, D. E. Leib, T.-W. Kuo, Y.-C. Lin, et al.
Soma-targeted imaging of neural circuits by ribosome tethering.
Neuron, 2020.

N. Chenouard, I. Smal, F. De Chaumont, M. Maška, I. F. Sbalzarini, Y. Gong, J. Cardinale, C. Carthel, S. Coraluppi, M. Winter, A. R. Cohen, W. J. Godinez, K. Rohr, Y. Kalaidzidis, L. Liang, J. Duncan, H. Shen, Y. Xu, K. E. Magnusson, J. Jaldén, H. M. Blau, P. Paul-Gilloteaux, P. Roudot, C. Kervrann, F. Waharte, J. Y. Tinevez, S. L. Shorte, J. Willemse, K. Celler, G. P. Van Wezel, H. W. Dan, Y. S. Tsai, C. O. De Solórzano, J. C. Olivo-Marin, and E. Meijering.
Objective comparison of particle tracking methods.
Nat. Methods, 11(3):281–289, 3 2014.
ISSN 15487091.
doi: 10.1038/nmeth.2808.

S. N. Chettih and C. D. Harvey.
Single-neuron perturbations reveal feature-specific competition in v1.
Nature, 567(7748):334–340, 2019.

D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek.
The trimmed iterative closest point algorithm.
In *Object recognition supported by user interaction for service robots*, volume 3, pages 545–548. IEEE, 2002.

K. P. Choe and K. Strange.
Molecular and genetic characterization of osmosensing and signal transduction in the nematode caenorhabditis elegans.
The FEBS Journal, 274(22):5782–5789, 2007.

G. C. Chow and others.
Analysis and control of dynamic economic systems.
Wiley, 1975.

N. Chronis, M. Zimmer, and C. I. Bargmann.
Microfluidics for in vivo imaging of neuronal and behavioral activity in Caenorhabditis elegans.
Nature Methods, 4(9):727–731, sep 2007.
ISSN 15487091.
doi: 10.1038/nmeth1075.

J. H. Clark, B. Engineering, and S. Ca.
Automated analysis of cellular signals from large-scale calcium imaging data.
Neuron, 63(6):747–760, 2009.
doi: 10.1016/j.neuron.2009.08.009.Automated.

S. Cocco, S. Leibler, and R. Monasson.
Neuronal couplings between retinal ganglion cells inferred by efficient inverse statistical physics methods, 2009.

M. R. Cohen and A. Kohn.
Measuring and interpreting neuronal correlations.
Nature neuroscience, 14(7):811, 2011.

L. Cong, Z. Wang, Y. Chai, W. Hang, C. Shang, W. Yang, L. Bai, J. Du, K. Wang, and Q. Wen.
Rapid whole brain imaging of neural activity in freely behaving larval zebrafish (danio rerio).
Elife, 6, 2017.

S. J. Cook, T. A. Jarrell, C. A. Brittin, Y. Wang, A. E. Bloniarz, M. A. Yakovlev, K. C. Nguyen, L. T.-H. Tang, E. A. Bayer, J. S. Duerr, et al.
Whole-animal connectomes of both caenorhabditis elegans sexes.
Nature, 571(7763):63–71, 2019a.

S. J. Cook et al.
Whole-animal connectomes of both caenorhabditis elegans sexes.
Nature, 571(7763):63–71, 2019b.

B. Cummins, T. Gedeon, and K. Spendlove.
On the efficacy of state space reconstruction methods in determining causality.
SIAM Journal on Applied Dynamical Systems, 14(1):335–381, 2015.

C. Czado.
Pair-copula constructions of multivariate copulas.
In *Copula theory and its applications*, pages 93–109. Springer, 2010.

G. Deco and E. Hugues.
Neural network mechanisms underlying stimulus driven variability reduction.
PLoS Comput. Biol., 8(3):e1002395, Mar. 2012.

A. P. Dempster, N. M. Laird, and D. B. Rubin.
Maximum likelihood from incomplete data via the em algorithm.
Journal of the Royal Statistical Society: Series B (Methodological), 39(1):1–22, 1977.

N. S. Detlefsen, O. Freifeld, and S. Hauberg.
Deep Diffeomorphic Transformer Networks.
Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 4403–4412, 2018.
ISSN 10636919.
doi: 10.1109/CVPR.2018.00463.

M. Dhamala, G. Rangarajan, and M. Ding.
Estimating granger causality from fourier and wavelet transforms of time series data, 2008.

D. A. Dickie, S. D. Shenkin, et al.
Whole brain magnetic resonance image atlases: a systematic review of existing atlases and caveats for use in population imaging.
Frontiers in neuroinformatics, 11:1, 2017.

D. A. Dombeck, A. N. Khabbaz, F. Collman, T. L. Adelman, and D. W. Tank.
Imaging large-scale neural activity with cellular resolution in awake, mobile mice.
Neuron, 56(1):43–57, 2007.

D. L. Donoho et al.
Compressed sensing.
IEEE Transactions on information theory, 52(4):1289–1306, 2006.

B. Drost, M. Ulrich, N. Navab, and S. Ilic.
Model globally, match locally: Efficient and robust 3d object recognition.
In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 998–1005. Ieee, 2010.

A. Dubbs, J. Guevara, and R. Yuste.
moco: Fast motion correction for calcium imaging.
Frontiers in neuroinformatics, 10:6, 2016.

A. Dufour, T. Y. Liu, C. Ducroz, R. Tournemenne, B. Cummings, R. Thibeaux, N. Guillen, A. Hero, and J. C. Olivo-Marin.
Signal processing challenges in quantitative 3-D cell morphology: more than meets the eye.
IEEE Signal Process. Mag., 32(1):30–40, 1 2015.
ISSN 10535888.
doi: 10.1109/msp.2014.2359131.

I. Dworkin and G. Gibson.
Epidermal growth factor receptor and transforming growth factor-β signaling contributes to variation for wing shape in drosophila melanogaster.
Genetics, 173(3):1417–1431, 2006.

T. Edinburgh, S. J. Eglen, and A. Ercole.
Causality indices for bivariate time series data: a comparative review of performance.
Chaos: An Interdisciplinary Journal of Nonlinear Science, 31(8):083111, 2021.

G. Elidan.
Copulas in machine learning.
In *Copulae in mathematical and quantitative finance*, pages 39–60. Springer, 2013.

S. W. Emmons.
The development of sexual dimorphism: studies of the caenorhabditis elegans male.
Wiley Interdisciplinary Reviews: Developmental Biology, 3(4):239–262, 2014.

S. W. Emmons.

Neural circuits of sexual behavior in caenorhabditis elegans.
Annual Review of Neuroscience, 41:349–369, 2018.

S. W. Emmons and P. W. Sternberg.
Male development and mating behavior.
2011.

O. Enqvist, K. Josephson, and F. Kahl.
Optimal correspondences from pairwise constraints.
In *2009 IEEE 12th international conference on computer vision*, pages 1295–1302. IEEE, 2009.

G. D. Evangelidis and R. Horaud.
Joint alignment of multiple point sets with batch and incremental expectation-maximization.
IEEE transactions on pattern analysis and machine intelligence, 40(6):1397–1410, 2018.

L. Faes, G. Nollo, and A. Porta.
Information-based detection of nonlinear granger causality in multivariate processes via a nonuniform embedding technique, 2011.

C. Finn and J. T. Lizier.
Pointwise partial information decomposition using the specificity and ambiguity lattices.
Entropy, 20(4):297, 2018.

M. A. Fischler and R. C. Bolles.
Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography.
Communications of the ACM, 24(6):381–395, 1981.

R. A. Fisher.
Statistical Methods for Research Workers.
Oliver and Boyd, Edinburgh, 1925.

R. S. Fisher and A. L. Velasco.
Electrical brain stimulation for epilepsy.
Nature Reviews Neurology, 10(5):261–270, 2014.

B. Flecker, W. Alford, J. M. Beggs, P. L. Williams, and R. D. Beer.
Partial information decomposition as a spatiotemporal filter.
Chaos, 21(3):037104, 2011.

B. A. Flusberg, A. Nimmerjahn, E. D. Cocker, E. A. Mukamel, R. P. Barretto, T. H. Ko, L. D. Burns, J. C. Jung, and M. J. Schnitzer.
High-speed, miniaturized fluorescence microscopy in freely moving mice.
Nature methods, 5(11):935–938, 2008.

E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky.
Nonparametric Bayesian learning of switching linear dynamical systems.
Advances in Neural Information Processing Systems 21 - Proceedings of the 2008 Conference, pages 457–464, 2009.

O. Freifeld, S. Hauberg, K. Batmanghelich, and J. W. Fisher.
Highly-expressive spaces of well-behaved transformations: Keeping it simple.
Proceedings of the IEEE International Conference on Computer Vision, 2015 Inter: 2911–2919, 2015.
ISSN 15505499.
doi: 10.1109/ICCV.2015.333.

O. Freifeld, S. Hauberg, and K. Batmanghelich.
Transformations Based on Continuous Piecewise-Affine Velocity Fields.
Ieeexplore.Ieee.Org, 8828(c):2496–2509, 2016.
URL `https://ieeexplore.ieee.org/abstract/document/7814343/`.

Y. Gao, E. Archer, L. Paninski, and J. P. Cunningham.
Linear dynamical neural population models through nonlinear embeddings.
Advances in Neural Information Processing Systems, (Nips):163–171, 2016.
ISSN 10495258.

L. R. García and D. S. Portman.
Neural circuits for sexually dimorphic and sexually divergent behaviors in caenorhabditis elegans.
Current opinion in neurobiology, 38:46–52, 2016.

L. R. Garcia, P. Mehta, and P. W. Sternberg.
Regulation of distinct muscle behaviors controls the c. elegans male's copulatory spicules during mating.

Cell, 107(6):777–788, 2001.

I. Gat and N. Tishby.
Synergy and redundancy among brain cells of behaving monkeys.
In *Advances in neural information processing systems*, pages 111–117, 1999.

M. Gendrel, E. G. Atlas, and O. Hobert.
A cellular and regulatory map of the gabaergic nervous system of c. elegans.
Elife, 5:e17686, 2016.

L. R. Girard, T. J. Fiedler, T. W. Harris, F. Carvalho, I. Antoshechkin, M. Han, P. W. Sternberg, L. D. Stein, and M. Chalfie.
Wormbook: the online review of caenorhabditis elegans biology.
Nucleic acids research, 35(suppl_1):D472–D475, 2007.

W. Göbel, B. M. Kampa, and F. Helmchen.
Imaging cellular network dynamics in three dimensions using fast 3d laser scanning.
Nature methods, 4(1):73–79, 2007.

A. E. Goodwell et al.
Debates—does information theory provide a new paradigm for earth science?
Water Resources Research, 56(2):e2019WR024940, 2020.

C. W. J. Granger.
Investigating causal relations by econometric models and cross-spectral methods, 1969.

M. S. Graziano, C. S. Taylor, and T. Moore.
Complex movements evoked by microstimulation of precentral cortex.
Neuron, 34(5):841–851, 2002.

T. Greitz, C. Bohm, S. Holte, and L. Eriksson.
A computerized brain atlas: construction, anatomical content, and some applications.
Journal of computer assisted tomography, 15(1):26–38, 1991.

M. Guizar-Sicairos, S. T. Thurman, and J. R. Fienup.
Efficient subpixel image registration algorithms.
Optics letters, 33(2):156–158, 2008.

A. J. Gutknecht, M. Wibral, and A. Makkeh.
Bits and pieces: Understanding information decomposition from part-whole relationships and formal logic.

Proceedings of the Royal Society A, 477(2251):20210110, 2021.

B. Haeffele, E. Young, and R. Vidal.
Structured low-rank matrix factorization: Optimality, algorithm, and applications to image processing.
In *International conference on machine learning*, pages 2007–2015, 2014.

K. M. Hallinen, R. Dempsey, M. Scholz, X. Yu, A. Linder, F. Randi, A. Sharma, J. W. Shaevitz, and A. M. Leifer.
Decoding locomotion from population neural activity in moving C. Elegans.
eLife, 10, 7 2021.
ISSN 2050084X.
doi: 10.7554/ELIFE.66135.

M. Harder, C. Salge, and D. Polani.
Bivariate measure of redundant information.
Physical Review E, 87(1):012130, 2013.

A. Hast, J. Nysjö, and A. Marchetti.
Optimal ransac-towards a repeatable algorithm for finding the optimal set.
2013.

E. S. Heckscher, F. Long, M. J. Layden, C.-H. Chuang, L. Manning, J. Richart, J. C. Pearson, S. T. Crews, H. Peng, E. Myers, and C. Q. Doe.
Atlas-builder software and the eNeuro atlas: resources for developmental biology and neuroscience.
Development, 141(12):2524–2532, 06 2014.
ISSN 0950-1991.
doi: 10.1242/dev.108720.
URL `https://doi.org/10.1242/dev.108720`.

O. Hirose, S. Kawaguchi, T. Tokunaga, Y. Toyoshima, T. Teramoto, S. Kuge, T. Ishihara, Y. Iino, and R. Yoshida.
Spf-celltracker: Tracking multiple cells with strongly-correlated moves using a spatial particle filter.
IEEE/ACM transactions on computational biology and bioinformatics, 15(6):1822–1831, 2017.

O. Hirose, S. Kawaguchi, T. Tokunaga, Y. Toyoshima, T. Teramoto, S. Kuge, T. Ishihara, Y. Iino, and R. Yoshida.
Spf-celltracker: Tracking multiple cells with strongly-correlated moves using a spatial particle filter.
IEEE/ACM Transactions on Computational Biology and Bioinformatics, 15(6):1822–1831, Nov 2018.
ISSN 2374-0043.
doi: 10.1109/TCBB.2017.2782255.

S. B. Hofer, H. Ko, B. Pichler, J. Vogelstein, H. Ros, H. Zeng, E. Lein, N. A. Lesica, and T. D. Mrsic-Flogel.
Differential connectivity and response dynamics of excitatory and inhibitory neurons in visual cortex.
Nature neuroscience, 14(8):1045, 2011.

D. Houle, D. R. Govindaraju, and S. Omholt.
Phenomics: the next challenge.
Nature reviews genetics, 11(12):855–866, 2010.

D. J. Hsu, K. Shi, and X. Sun.
Linear regression without correspondence.
In *Advances in Neural Information Processing Systems*, pages 1531–1540, 2017.

R. A. Ince.
Measuring multivariate redundant information with pointwise common change in surprisal.
Entropy, 19(7):318, 2017.

P. Indyk, R. Motwani, and S. Venkatasubramanian.
Geometric matching under noise: Combinatorial bounds and algorithms.
In *SODA*, pages 457–465, 1999.

E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele.
Deepercut: A deeper, stronger, and faster multi-person pose estimation model.
Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 9910 LNCS:34–50, 2016.
ISSN 16113349.
doi: 10.1007/978-3-319-46466-4{_}3.

S. Irani and P. Raghavan.
Combinatorial and experimental results for randomized point matching algorithms.
Computational Geometry, 12(1-2):17–31, 1999.

M. Jaderberg, K. Simonyan, A. Zisserman, et al.
Spatial transformer networks.
Advances in neural information processing systems, 28, 2015.

M. Jaritz, R. D. Charette, E. Wirbel, X. Perrotton, and F. Nashashibi.
Sparse and dense data with CNNs: Depth completion and semantic segmentation.
Proceedings - 2018 International Conference on 3D Vision, 3DV 2018, pages 52–60, 2018.
doi: 10.1109/3DV.2018.00017.

T. A. Jarrell, Y. Wang, A. E. Bloniarz, C. A. Brittin, M. Xu, J. N. Thomson, D. G. Albertson, D. H. Hall, and S. W. Emmons.
The connectome of a decision-making neural network.
science, 337(6093):437–444, 2012a.

T. A. Jarrell et al.
The connectome of a decision-making neural network.
Science, 337(6093):437–44, 2012b.

Jia Deng, Wei Dong, R. Socher, Li-Jia Li, Kai Li, and Li Fei-Fei.
ImageNet: A large-scale hierarchical image database.
IEEE Conference on Computer Vision and Pattern Recognition, pages 248–255, 2009.
doi: 10.1109/cvprw.2009.5206848.

H. Joe.
Multivariate Models and Multivariate Dependence Concepts.
CRC Press, May 1997.

A. R. Jones, C. C. Overly, and S. M. Sunkin.
The allen brain atlas: 5 years and beyond.
Nature Reviews Neuroscience, 10(11):821–828, 2009.

R. Jonker and T. Volgenant.
Improving the hungarian assignment algorithm.
Operations Research Letters, 5(4):171–175, 1986.

D. Kainmueller, F. Jug, C. Rother, and G. Myers.
Active graph matching for automatic joint segmentation and annotation of c. elegans. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 81–88. Springer, 2014.

M. Kaiser and C. C. Hilgetag.
Nonoptimal component placement, but short processing paths, due to long-distance projections in neural systems.
PLoS computational biology, 2(7):e95, 2006.

A. K. Kalis, D. U. Kissiov, E. S. Kolenbrander, Z. Palchick, S. Raghavan, B. J. Tetreault, E. Williams, C. M. Loer, and J. R. Wolff.
Patterning of sexually dimorphic neurogenesis in the caenorhabditis elegans ventral cord by hox and tale homeodomain transcription factors.
Developmental Dynamics, 243(1):159–171, 2014.

S. Kato, H. S. Kaplan, T. Schrödel, S. Skora, T. H. Lindsay, E. Yemini, S. Lockery, and M. Zimmer.
Global brain dynamics embed the motor command sequence of caenorhabditis elegans.
Cell, 163(3):656–669, 2015.

J. W. Kay, W. Phillips, J. Aru, B. P. Graham, and M. E. Larkum.
A Bayesian decomposition of BAC firing as a mechanism for apical amplification in neocortical pyramidal neurons.
bioRxiv, page 604066, 2019.

A. M. Kerlin, M. L. Andermann, V. K. Berezovskii, and R. C. Reid.
Broadly tuned response properties of diverse inhibitory neuron subtypes in mouse visual cortex.
Neuron, 67(5):858–871, 2010.

J. N. Kerr, D. Greenberg, and F. Helmchen.
Imaging input and output of neocortical networks in vivo.
Proceedings of the National Academy of Sciences, 102(39):14063–14068, 2005.

N. S. Keskar, J. Nocedal, P. T. P. Tang, D. Mudigere, and M. Smelyanskiy.
On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima.
5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings, 9 2016.

doi: 10.48550/arxiv.1609.04836.
URL https://arxiv.org/abs/1609.04836v2.

R. Kiani, C. J. Cueva, J. B. Reppas, D. Peixoto, S. I. Ryu, and W. T. Newsome.
Natural grouping of neural responses reveals spatially segregated clusters in prearcuate cortex.
Neuron, 85(6):1359–1373, 2015.

D. P. Kingma and J. Ba.
Adam: A method for stochastic optimization.
ICLR 2015.

D. P. Kingma and M. Welling.
Auto-encoding variational bayes.
NIPS 2015.

A. Klein, J. Andersson, B. A. Ardekani, J. Ashburner, B. Avants, M.-C. Chiang, G. E. Christensen, D. L. Collins, J. Gee, P. Hellier, et al.
Evaluation of 14 nonlinear deformation algorithms applied to human brain mri registration.
Neuroimage, 46(3):786–802, 2009.

A. Kraskov, H. Stögbauer, and P. Grassberger.
Estimating mutual information.
Physical review E, 69(6):066138, 2004.

N. Kriegeskorte and X.-X. Wei.
Neural tuning and representational geometry.
Nature Reviews Neuroscience, 22(11):703–718, 2021.

H. W. Kuhn.
The hungarian method for the assignment problem.
Naval research logistics quarterly, 2(1-2):83–97, 1955.

T. Lagache, A. Hanson, A. Fairhall, and R. Yuste.
Robust single neuron tracking of calcium imaging in behaving hydra.
bioRxiv, 2020a.
doi: 10.1101/2020.06.22.165696.
URL https://www.biorxiv.org/content/early/2020/06/23/2020.06.22.165696.

T. Lagache, A. Hanson, A. Fairhall, and R. Yuste.
Robust single neuron tracking of calcium imaging in behaving hydra.
bioRxiv, pages 1–30, 2020b.
ISSN 2692-8205.
doi: 10.1101/2020.06.22.165696.

J. Larsch, D. Ventimiglia, C. I. Bargmann, and D. R. Albrecht.
High-throughput imaging of neuronal activity in caenorhabditis elegans.
Proceedings of the National Academy of Sciences, 110(45):E4266–E4273, 2013.

L. Le Cam.
Comparison of experiments: A short review.
Lecture Notes-Monograph Series, pages 127–138, 1996.

D. D. Lee and H. S. Seung.
Learning the parts of objects by non-negative matrix factorization.
Nature, 401(6755):788–791, 1999.

D. D. Lee and H. S. Seung.
Algorithms for non-negative matrix factorization.
In *Advances in neural information processing systems*, pages 556–562, 2001.

M. C. Lee, O. Oktay, A. Schuh, M. Schaap, and B. Glocker.
Image-and-Spatial Transformer Networks for Structure-Guided Image Registration.
Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 11765 LNCS:337–345, 2019.
ISSN 16113349.
doi: 10.1007/978-3-030-32245-8{_}38.

S. Lefort, C. Tomm, J.-C. F. Sarria, and C. C. Petersen.
The excitatory neuronal network of the c2 barrel column in mouse primary somatosensory cortex.
Neuron, 61(2):301–316, 2009.

A. M. Leifer, C. Fang-Yen, M. Gershow, M. J. Alkema, and A. D. Samuel.
Optogenetic manipulation of neural activity in freely moving Caenorhabditis elegans.
Nature Methods, 8(2):147–152, 2 2011.
ISSN 15487091.
doi: 10.1038/NMETH.1554.

E. S. Lein, M. J. Hawrylycz, et al.
Genome-wide atlas of gene expression in the adult mouse brain.
Nature, 445:168–176, 2007.

M. Leshno and Y. Spector.
An elementary proof of Blackwell's theorem.
Mathematical Social Sciences, 25(1):95–98, 1992.

A. Lin, D. Witvliet, L. Hernandez-Nunez, S. W. Linderman, A. D. Samuel, and V. Venkatachalam.
Imaging whole-brain activity to understand behaviour.
Nature Reviews Physics, 0123456789, 2022.
ISSN 25225820.
doi: 10.1038/s42254-022-00430-w.

S. Linderman, A. Nichols, D. Blei, M. Zimmer, and L. Paninski.
Hierarchical recurrent state space models reveal discrete and continuous dynamics of neural activity in C. elegans.
bioRxiv, pages 1–55, 2019.
ISSN 2692-8205.

S. W. Linderman, M. J. Johnson, A. C. Miller, R. P. Adams, D. M. Blei, and L. Paninski.
Bayesian learning and inference in recurrent switching linear dynamical systems.
Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 54, 2017.

R. Lints and S. W. Emmons.
Patterning of dopaminergic neurotransmitter identity among caenorhabditis elegans ray sensory neurons by a tgfbeta family signaling pathway and a hox gene.
Development, 126(24):5819–5831, 1999.

R. Lints, L. Jia, K. Kim, C. Li, and S. Emmons.
Axial patterning of c. elegans male sensilla identities by selector genes.
Developmental biology, 269(1):137–151, 2004.

A. Litwin-Kumar and B. Doiron.
Slow dynamics and high variability in balanced cortical networks with clustered connections.
Nature neuroscience, 15(11):1498–1505, 2012.

K. S. Liu and P. W. Sternberg.
Sensory regulation of male mating behavior in caenorhabditis elegans.
Neuron, 14(1):79–89, 1995.

J. T. Lizier, B. Flecker, and P. L. Williams.
Towards a synergy-based approach to measuring information modification.
In *2013 IEEE Symposium on Artificial Life (ALIFE)*, pages 43–51. IEEE, 2013.

F. Long, H. Peng, X. Liu, S. K. Kim, and E. Myers.
A 3d digital atlas of c. elegans and its application to single-cell analyses.
Nature Methods, 6(9):667–672, Sep 2009.
ISSN 1548-7105.
doi: 10.1038/nmeth.1366.
URL `https://doi.org/10.1038/nmeth.1366`.

L. B. Lucy.
An iterative technique for the rectification of observed distributions.
The Astronomical Journal, 79(6):745, 1974.
ISSN 00046256.
doi: 10.1086/111605.

W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T. K. Kim.
Multiple object tracking: A literature review.
Artificial Intelligence, 293:1–18, 2021.
ISSN 00043702.
doi: 10.1016/j.artint.2020.103448.

X. Luo and S. M. Bhandarkar.
Multiple object tracking using elastic matching.
IEEE International Conference on Advanced Video and Signal Based Surveillance - Proceedings of AVSS 2005, 2005:123–128, 2005.
doi: 10.1109/AVSS.2005.1577254.

J. Ma and Z. Sun.
Mutual information is copula entropy.
Tsinghua Science & Technology, 16(1):51–54, 2011.

J. Ma, J. Zhao, and A. L. Yuille.
Non-rigid point set registration by preserving global and local structures.

IEEE Transactions on Image Processing, 25(1):53–64, 1 2016.
ISSN 10577149.
doi: 10.1109/TIP.2015.2467217.

K. E. Magnusson, J. Jalden, P. M. Gilbert, and H. M. Blau.
Global linking of cell tracks using the viterbi algorithm.
IEEE Transactions on Medical Imaging, 34(4):911–929, 2015.
ISSN 1558254X.
doi: 10.1109/TMI.2014.2370951.

A. Makkeh, D. O. Theis, and R. Vicente.
Bivariate partial information decomposition: The optimization perspective.
Entropy, 19(10):530, 2017.

A. Makkeh, D. O. Theis, and R. Vicente.
Broja-2pid: A robust estimator for bivariate partial information decomposition.
Entropy, 20(4):271, 2018.

A. Makkeh, A. J. Gutknecht, and M. Wibral.
Introducing a differentiable measure of pointwise shared information.
Physical Review E, 103(3):032149, 2021.

K. Mann, C. L. Gallen, and T. R. Clandinin.
Whole-brain calcium imaging reveals an intrinsic functional network in drosophila.
Current biology : CB, 27(15):2389–2396.e4, Aug 2017.
ISSN 1879-0445.
doi: 10.1016/j.cub.2017.06.076.
28756955[pmid].

V. Mante, D. Sussillo, K. V. Shenoy, and W. T. Newsome.
Context-dependent computation by recurrent dynamics in prefrontal cortex.
nature, 503(7474):78–84, 2013.

I. E. Marinescu, P. N. Lawlor, and K. P. Kording.
Quasi-experimental causality in neuroscience and behavioural research.
Nature human behaviour, 2(12):891–898, 2018.

H. Maron and Y. Lipman.
(probably) concave graph matching.
In *Advances in Neural Information Processing Systems*, pages 408–418, 2018.

R. Marschinski and H. Kantz.
Analysing the information flow between financial time series.
The European Physical Journal B-Condensed Matter and Complex Systems, 30(2): 275–281, 2002.

R. Maruyama, K. Maeda, H. Moroda, I. Kato, M. Inoue, H. Miyakawa, and T. Aonishi.
Detecting cells using non-negative matrix factorization on calcium imaging data.
Neural Networks, 55:11–19, 2014.

M. Maška, V. Ulman, D. Svoboda, P. Matula, P. Matula, C. Ederra, A. Urbiola, T. España, S. Venkatesan, D. M. Balak, P. Karas, T. Bolcková, M. Štreitová, C. Carthel, S. Coraluppi, N. Harder, K. Rohr, K. E. Magnusson, J. Jaldén, H. M. Blau, O. Dzyubachyk, P. Křížek, G. M. Hagen, D. Pastor-Escuredo, D. Jimenez-Carretero, M. J. Ledesma-Carbayo, A. Muñoz-Barrutia, E. Meijering, M. Kozubek, and C. Ortiz-De-Solorzano.
A benchmark for comparison of cell tracking algorithms.
Bioinformatics, 30(11):1609–1617, 6 2014.
ISSN 14602059.
doi: 10.1093/bioinformatics/btu080.

A. Mathis, P. Mamidanna, K. M. Cury, T. Abe, V. N. Murthy, M. W. Mathis, and M. Bethge.
DeepLabCut: markerless pose estimation of user-defined body parts with deep learning.
Nature Neuroscience, 21(9):1281–1289, 2018.
ISSN 15461726.
doi: 10.1038/s41593-018-0209-y.
URL `http://dx.doi.org/10.1038/s41593-018-0209-y`.

P. Matula, M. Maska, D. V. Sorokin, P. Matula, C. Ortiz-De-Solórzano, and M. Kozubek.
Cell tracking accuracy measurement based on comparison of acyclic oriented graphs.
PLoS ONE, 10(12), 2015.
ISSN 19326203.
doi: 10.1371/journal.pone.0144959.

D. Mazza and M. Pagani.
Automatic differentiation in PCF.
Proceedings of the ACM on Programming Languages, 5(POPL):1–4, 2021.
ISSN 24751421.

doi: 10.1145/3434309.

J. Mazziotta, A. Toga, et al.
A probabilistic atlas and reference system for the human brain: International consortium for brain mapping (icbm).
Philosophical transactions of the Royal Society of London. Series B, Biological sciences, 356 1412:1293–322, 2001.

L. Mazzucato, A. Fontanini, and G. La Camera.
Dynamics of multistable states during ongoing and evoked cortical activity.
The Journal of Neuroscience, 35(21):8214–8231, 2015.

L. Mazzucato, G. La Camera, and A. Fontanini.
Expectation-induced modulation of metastable activity underlies faster coding of sensory stimuli.
Nat Neurosci, 22(5):787–796, 05 2019.

E. Meijering.
Cell segmentation: 50 years down the road.
IEEE Signal Process. Mag., 29(5):140–145, 2012.
ISSN 10535888.
doi: 10.1109/msp.2012.2204190.

N. Mellado, D. Aiger, and N. J. Mitra.
Super 4pcs fast global pointcloud registration via smart indexing.
In *Computer Graphics Forum*, volume 33, pages 205–215. Wiley Online Library, 2014.

G. Mena, A. Nejatbakhsh, E. Varol, and J. Niles-Weed.
Sinkhorn em: An expectation-maximization algorithm based on entropic optimal transport, 2020.

S. Moeller, T. Crapse, L. Chang, and D. Y. Tsao.
The effect of face patch microstimulation on perception of faces and objects.
Nat Neurosci, 20(5):743–752, May 2017.

E. Moen, D. Bannon, T. Kudo, W. Graf, M. Covert, and D. Van Valen.
Deep learning for cellular image analysis.
Nature Methods, 16(12):1233–1246, 12 2019.
ISSN 15487105.
doi: 10.1038/S41592-019-0403-1.

L. Molina-García, C. Lloret-Fernández, S. J. Cook, B. Kim, R. C. Bonnington, M. Sammut, J. M. O'Shea, S. P. Gilbert, D. J. Elliott, D. H. Hall, et al.
Direct glia-to-neuron transdifferentiation gives rise to a pair of male-specific neurons that ensure nimble male mating.
Elife, 9:e48361, 2020.

D. M. Mount, N. S. Netanyahu, and J. Le Moigne.
Efficient algorithms for robust feature matching.
Pattern recognition, 32(1):17–38, 1999.

E. A. Mukamel, A. Nimmerjahn, and M. J. Schnitzer.
Automated analysis of cellular signals from large-scale calcium imaging data.
Neuron, 63(6):747–760, 2009.

A. Myronenko and X. Song.
Point set registration: Coherent point drift.
IEEE transactions on pattern analysis and machine intelligence, 32(12):2262–2275, 2010.

T. Nagler and C. Czado.
Evading the curse of dimensionality in nonparametric density estimation with simplified vine copulas.
Journal of Multivariate Analysis, 151:69–89, 2016.

T. Nagler and T. Vatter.
pyvinecopulib, Nov. 2020.
URL `https://doi.org/10.5281/zenodo.4288293`.

A. Nejatbakhsh and E. Varol.
Neuron matching in c. elegans with robust approximate linear regression without correspondence.
In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2837–2846, 2021.

A. Nejatbakhsh, F. Fumarola, S. Esteki, T. Toyoizumi, R. Kiani, and L. Mazzucato.
Predicting perturbation effects from resting state activity using functional causal flow.
bioRxiv, 2020a.

A. Nejatbakhsh, E. Varol, E. Yemini, O. Hobert, and L. Paninski.

Probabilistic joint segmentation and labeling of c. elegans neurons.
In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 130–140. Springer, 2020b.

A. Nejatbakhsh, E. Varol, E. Yemini, V. Venkatachalam, A. Lin, A. D. Samuel, and L. Paninski.
Extracting neural signals from semi-immobilized animals with deformable non-negative matrix factorization.
bioRxiv, 2020c.

R. B. Nelsen.
An Introduction to Copulas.
Springer Science & Business Media, June 2007.

J. P. Nguyen, F. B. Shipley, A. N. Linder, G. S. Plummer, M. Liu, S. U. Setru, J. W. Shaevitz, and A. M. Leifer.
Whole-brain calcium imaging with cellular resolution in freely behaving Caenorhabditis elegans.
Proceedings of the National Academy of Sciences of the United States of America, 113 (8):E1074–E1081, 2 2016a.
ISSN 10916490.
doi: 10.1073/PNAS.1507110112.

J. P. Nguyen, F. B. Shipley, A. N. Linder, G. S. Plummer, M. Liu, S. U. Setru, J. W. Shaevitz, and A. M. Leifer.
Whole-brain calcium imaging with cellular resolution in freely behaving caenorhabditis elegans.
Proceedings of the National Academy of Sciences, 113(8):E1074–E1081, 2016b.

J. P. Nguyen, A. N. Linder, G. S. Plummer, J. W. Shaevitz, and A. M. Leifer.
Automatically tracking neurons in a moving and deforming brain.
PLoS Computational Biology, 13(5):1–19, 2017a.
ISSN 15537358.
doi: 10.1371/journal.pcbi.1005517.

J. P. Nguyen, A. N. Linder, G. S. Plummer, J. W. Shaevitz, and A. M. Leifer.
Automatically tracking neurons in a moving and deforming brain.
PLoS computational biology, 13(5):e1005517, 2017b.

C. M. Niell and S. J. Smith.
Functional imaging reveals rapid development of visual response properties in the zebrafish tectum.
Neuron, 45(6):941–951, 2005.

L. Novelli and J. T. Lizier.
Inferring network properties from time series using transfer entropy and mutual information: validation of multivariate versus bivariate approaches.
Network Neuroscience, 5(2):373–404, 2021.

P. D. O'grady and B. A. Pearlmutter.
Convolutive non-negative matrix factorisation with a sparseness constraint.
In *2006 16th IEEE Signal Processing Society Workshop on Machine Learning for Signal Processing*, pages 427–432. IEEE, 2006.

S. W. Oh et al.
A mesoscale connectome of the mouse brain.
Nature, 508(7495):207–214, 2014.

P. Paatero and U. Tapper.
Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values.
Environmetrics, 5(2):111–126, 1994.

M. Pachitariu, A. M. Packer, N. Pettit, H. Dalgleish, M. Hausser, and M. Sahani.
Extracting regions of interest from biological images with convolutional sparse block coding.
In *Advances in neural information processing systems*, pages 1745–1753, 2013.

M. Pachitariu, C. Stringer, M. Dipoppa, S. Schröder, L. F. Rossi, H. Dalgleish, M. Carandini, and K. D. Harris.
Suite2p: beyond 10,000 neurons with standard two-photon microscopy.
Biorxiv, 2017.

J. S. Packer, Q. Zhu, C. Huynh, P. Sivaramakrishnan, E. Preston, H. Dueck, D. Stefanik, K. Tan, C. Trapnell, J. Kim, et al.
A lineage-resolved molecular atlas of c. elegans embryogenesis at single-cell resolution.
Science, 365(6459):eaax1971, 2019.

A. Pakman, A. Nejatbakhsh, D. Gilboa, A. Makkeh, L. Mazzucato, M. Wibral, and E. Schneidman.
Estimating the unique information of continuous variables.
Advances in Neural Information Processing Systems, 34:20295–20307, 2021.

A. Pananjady, M. J. Wainwright, and T. A. Courtade.
Linear regression with an unknown permutation: Statistical and computational limits.
In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 417–424. IEEE, 2016.

C. F. Park, M. B. Keshteli, K. Korchagina, A. Delrocq, V. Susoy, C. L. Jones, A. D. T. Samuel, and S. J. Rahi.
Automated neuron tracking inside moving and deforming animals using deep learning and targeted augmentation.
bioRxiv, 3 2022.
doi: 10.1101/2022.03.15.484536.
URL `https://www.biorxiv.org/content/10.1101/2022.03.15.484536v1`.

J. Parvizi, C. Jacques, B. L. Foster, N. Withoft, V. Rangarajan, K. S. Weiner, and K. Grill-Spector.
Electrical stimulation of human fusiform face-selective regions distorts face perception.
Journal of Neuroscience, 32(43):14915–14920, 2012.

J. Pearl.
Causal inference in statistics: An overview.
Statistics surveys, 3:96–146, 2009.

Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma.
Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images.
IEEE transactions on pattern analysis and machine intelligence, 34(11):2233–2246, 2012.

L. Pereira, P. Kratsios, E. Serrano-Saiz, H. Sheftel, A. E. Mayo, D. H. Hall, J. G. White, B. LeBoeuf, L. R. Garcia, U. Alon, et al.
A cellular and regulatory map of the cholinergic nervous system of c. elegans.
Elife, 4, 2015.

J. Peters, P. Bühlmann, and N. Meinshausen.

Causal inference by using invariant prediction: identification and confidence intervals.
Journal of the Royal Statistical Society. Series B: Statistical Methodology, 78(5):947–1012, 2016.
ISSN 14679868.
doi: 10.1111/rssb.12167.

J. Peters, D. Janzing, and B. Schölkopf.
Elements of causal inference: foundations and learning algorithms.
The MIT Press, 2017.

G. Peyré, M. Cuturi, et al.
Computational optimal transport.
Foundations and Trends® in Machine Learning, 11(5-6):355–607, 2019.

G. Pica, E. Piasini, H. Safaai, C. Runyan, C. D. Harvey, M. E. Diamond, C. Kayser, T. Fellin, and S. Panzeri.
Quantifying how much sensory information in a neural code is relevant for behavior.
In *NIPS*, 2017.

E. A. Pnevmatikakis and A. Giovannucci.
Normcorre: An online algorithm for piecewise rigid motion correction of calcium imaging data.
Journal of neuroscience methods, 291:83–94, 2017.

E. A. Pnevmatikakis, D. Soudry, Y. Gao, T. A. Machado, J. Merel, D. Pfau, T. Reardon, Y. Mu, C. Lacefield, W. Yang, et al.
Simultaneous denoising, deconvolution, and demixing of calcium imaging data.
Neuron, 89(2):285–299, 2016.

J. Pokrass, A. M. Bronstein, M. M. Bronstein, P. Sprechmann, and G. Sapiro.
Sparse modeling of intrinsic correspondences.
In *Computer Graphics Forum*, volume 32, pages 459–468. Wiley Online Library, 2013.

B. Poole, L. Grosenick, M. Broxton, K. Deisseroth, and S. Ganguli.
Robust non-rigid alignment of volumetric calcium imaging data.
COSYNE.[Google Scholar], 2015.

D. S. Portman.
Sexual modulation of sex-shared neurons and circuits in caenorhabditis elegans.
Journal of neuroscience research, 95(1-2):527–538, 2017.

R. Prevedel, Y.-G. Yoon, M. Hoffmann, N. Pak, G. Wetzstein, S. Kato, T. Schrödel, R. Raskar, M. Zimmer, E. S. Boyden, et al.
Simultaneous whole-animal 3d imaging of neuronal activity using light-field microscopy.
Nature methods, 11(7):727–730, 2014.

L. Qu, F. Long, X. Liu, S. Kim, E. Myers, and H. Peng.
Simultaneous recognition and segmentation of cells: application in c.elegans.
Bioinformatics (Oxford, England), 27(20):2895–2902, Oct 2011.
ISSN 1367-4811.
doi: 10.1093/bioinformatics/btr480.
URL `https://pubmed.ncbi.nlm.nih.gov/21849395`.
21849395[pmid].

R. Q. Quiroga and S. Panzeri.
Extracting information from neuronal populations: information theory and decoding approaches.
Nature Reviews Neuroscience, 10(3):173–185, 2009.

J. Rauh.
Secret sharing and shared information.
Entropy, 19(11):601, 2017.

J. Rauh, M. Schünemann, and J. Jost.
Properties of unique information.
arXiv preprint arXiv:1912.12505, 2019.

A. T. Reid, D. B. Headley, R. D. Mill, R. Sanchez-Romero, L. Q. Uddin, D. Marinazzo, D. J. Lurie, P. A. Valdés-Sosa, S. J. Hanson, B. B. Biswal, et al.
Advancing functional connectivity research from association to causation.
Nature neuroscience, 22(11):1751–1760, 2019.

J. Reidl, J. Starke, D. B. Omer, A. Grinvald, and H. Spors.
Independent component analysis of high-resolution imaging data identifies distinct functional domains.
Neuroimage, 34(1):94–108, 2007.

W. H. Richardson.
Bayesian-Based Iterative Method of Image Restoration.

Journal of the Optical Society of America, 62(1):55, 1972.
ISSN 0030-3941.
doi: 10.1364/josa.62.000055.

M. Rigotti, O. Barak, M. R. Warden, X.-J. Wang, N. D. Daw, E. K. Miller, and S. Fusi.
The importance of mixed selectivity in complex cognitive tasks.
Nature, 497(7451):585–590, 2013.

F. Roch, G. Jiménez, and J. Casanova.
Egfr signalling inhibits capicua-dependent repression during specification of drosophila wing veins.
2002.

P. Roland, C. Graufelds, J. Wǎhlin, L. Ingelman, M. Andersson, A. Ledberg, J. Pedersen, S. Åkerman, A. Dabringhaus, and K. Zilles.
Human brain atlas: for high-resolution functional and anatomical mapping.
Human Brain Mapping, 1(3):173–184, 1994.

S. Roweis and Z. Ghahramani.
A unifying review of linear gaussian models.
Neural Computation, 11(2):305–345, 1999.
ISSN 08997667.
doi: 10.1162/089976699300016674.

S. Ruder.
An overview of gradient descent optimization algorithms.
arXiv:1609.04747, 9 2016.
doi: 10.48550/arxiv.1609.04747.
URL `https://arxiv.org/abs/1609.04747v2`.

D. Rueckert, L. I. Sonoda, C. Hayes, D. L. Hill, M. O. Leach, and D. J. Hawkes.
Nonrigid registration using free-form deformations: application to breast mr images.
IEEE transactions on medical imaging, 18(8):712–721, 1999.

A. A. Russo, R. Khajeh, S. R. Bittner, S. M. Perkins, J. P. Cunningham, L. F. Abbott, and M. M. Churchland.
Neural trajectories in the supplementary motor area and motor cortex exhibit distinct geometries, compatible with different classes of computation.
Neuron, 107(4):745–758, 2020.

S. Sadeh and C. Clopath.
Theory of neuronal perturbome in cortical networks.
Proceedings of the National Academy of Sciences, 117(43):26966–26976, 2020.

C. D. Salzman, K. H. Britten, and W. T. Newsome.
Cortical microstimulation influences perceptual judgements of motion direction.
Nature, 346(6280):174–177, 1990.

C. D. Salzman, C. M. Murasugi, K. H. Britten, and W. T. Newsome.
Microstimulation in visual area mt: effects on direction discrimination performance.
Journal of Neuroscience, 12(6):2331–2355, 1992.

M. Sammut, S. J. Cook, K. C. Nguyen, T. Felton, D. H. Hall, S. W. Emmons, R. J. Poole, and A. Barrios.
Glia-derived neurons are required for sex-specific learning in c. elegans.
Nature, 526(7573):385–390, 2015.

R. Sandkühler, C. Jud, S. Andermatt, and P. C. Cattin.
AirLab: Autograd Image Registration Laboratory.
arXiv:1806.09907, 6 2018.
URL `http://arxiv.org/abs/1806.09907`.

S. R. Santacruz, E. L. Rich, J. D. Wallis, and J. M. Carmena.
Caudate microstimulation increases value of specific choices.
Current Biology, 27(21):3375–3383, 2017.

T. Sauer, J. A. Yorke, and M. Casdagli.
Embedology, 1991.

S. Saxena et al.
Localized semi-nonnegative matrix factorization (locanmf) of widefield calcium imaging data.
bioRxiv, page 650093, 2019.

W. Schafer.
Egg-laying., wormbook: the online review of c. elegans biology.
2005.

E. Schaffer, N. Mishra, W. Li, et al.
flygenvectors: large-scale dynamics of internal and behavioral statesin a small animal.

COSYNE, (III-19), 2020.

G. Schamberg and P. Venkatesh.
Partial Information Decomposition via Deficiency for Multivariate Gaussians.
arXiv preprint arXiv:2105.00769, 2021.

T. Schaul, S. Zhang, and Y. LeCun.
No More Pesky Learning Rates.
30th International Conference on Machine Learning, ICML 2013, (PART 2):1380–1388, 6 2012.
doi: 10.48550/arxiv.1206.1106.
URL `https://arxiv.org/abs/1206.1106v2`.

L. K. Scheffer and I. A. Meinertzhagen.
The fly brain atlas.
Annual review of cell and developmental biology, 35:637–653, 2019.

C. A. Schneider, W. S. Rasband, and K. W. Eliceiri.
NIH Image to ImageJ: 25 years of image analysis.
Nat. Methods, 9(7):671–675, 7 2012.
ISSN 15487091.
doi: 10.1038/nmeth.2089.

E. Schneidman, W. Bialek, and M. J. Berry.
Synergy, redundancy, and independence in population codes.
Journal of Neuroscience, 23(37), 2003.

B. Scholkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio.
Toward Causal Representation Learning.
Proceedings of the IEEE, 109(5):612–634, 2021.
ISSN 15582256.
doi: 10.1109/JPROC.2021.3058954.

T. Schreiber.
Measuring information transfer, 2000a.

T. Schreiber.
Measuring information transfer.
Physical review letters, 85(2):461, 2000b.

T. Schrödel, R. Prevedel, K. Aumayr, M. Zimmer, and A. Vaziri.
Brain-wide 3d imaging of neuronal activity in caenorhabditis elegans with sculpted light.
Nature methods, 10(10):1013, 2013.

S. Schulter, P. Vernaza, W. Choi, and M. Chandraker.
Deep network flow for multi-object tracking.
Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017-Janua:2730–2739, 2017.
doi: 10.1109/CVPR.2017.292.

E. Serrano-Saiz, L. Pereira, M. Gendrel, U. Aghayeva, A. Bhattacharya, K. Howell, L. R. Garcia, and O. Hobert.
A neurotransmitter atlas of the caenorhabditis elegans male nervous system reveals sexually dimorphic neurotransmitter usage.
Genetics, 206(3):1251–1269, 2017.

A. Sheikhattar, S. Miran, J. Liu, J. B. Fritz, S. A. Shamma, P. O. Kanold, and B. Babadi.
Extracting neuronal functional network dynamics via adaptive granger causality analysis.
Proceedings of the National Academy of Sciences, 115(17):E3869–E3878, 2018.

F. B. Shipley, C. M. Clark, M. J. Alkema, and A. M. Leifer.
Simultaneous optogenetic manipulation and calcium imaging in freely moving C. elegans.
Frontiers in Neural Circuits, 8(MAR), 3 2014.
ISSN 16625110.
doi: 10.3389/FNCIR.2014.00028.

G. Si, J. K. Kanwal, Y. Hu, C. J. Tabone, J. Baron, M. Berck, G. Vignoud, and A. D. Samuel.
Structured Odorant Response Patterns across a Complete Olfactory Receptor Neuron Population.
Neuron, 101(5):950–962.e7, mar 2019.
ISSN 10974199.
doi: 10.1016/j.neuron.2018.12.030.

R. M. Siegel, J.-R. Duann, T.-P. Jung, and T. Sejnowski.

Spatiotemporal dynamics of the functional architecture for gain fields in inferior parietal lobule of behaving monkey.
Cerebral cortex, 17(2):378–390, 2007.

R. Sinkhorn and P. Knopp.
Concerning nonnegative matrices and doubly stochastic matrices.
Pacific Journal of Mathematics, 21(2):343–348, 1967.

M. Sklar.
Fonctions de repartition a n dimensions et leurs marges.
Publ. inst. statist. univ. Paris, 8:229–231, 1959.

M. Skuhersky, T. Wu, E. Yemini, E. Boyden, and M. Tegmark.
Toward a more accurate 3d atlas of c. elegans neurons.
bioRxiv, 2021.

P. Smaragdis.
Convolutive speech bases and their application to supervised speech separation.
IEEE Transactions on Audio, Speech, and Language Processing, 15(1):1–12, 2006.

J. T. H. Smith, S. W. Linderman, and D. Sussillo.
Reverse engineering recurrent neural networks with Jacobian switching linear dynamical systems.
(NeurIPS), 2021.

A. Sonnenschein, D. VanderZee, W. R. Pitchers, S. Chari, and I. Dworkin.
An image database of drosophila melanogaster wings for phenomic and biometric analysis.
GigaScience, 4(1):s13742–015, 2015.

R. Spilger, A. Imle, J. Y. Lee, B. Muller, O. T. Fackler, R. Bartenschlager, and K. Rohr.
A Recurrent Neural Network for Particle Tracking in Microscopy Images Using Future Information, Track Hypotheses, and Multiple Detections.
IEEE Transactions on Image Processing, 29:3681–3694, 2020.
ISSN 19410042.
doi: 10.1109/TIP.2020.2964515.

J. Stark, D. Broomhead, M. Davies, and J. Huke.
Takens embedding theorems for forced and stochastic systems.
Nonlinear Analysis: Theory, Methods & Applications, 30(8):5303–5314, 1997.

M. Stetter, H. Greve, C. G. Galizia, and K. Obermayer.
Analysis of calcium imaging signals from the honeybee brain by nonlinear models.
Neuroimage, 13(1):119–128, 2001.

G. Sugihara, R. May, H. Ye, C.-H. Hsieh, E. Deyle, M. Fogarty, and S. Munch.
Detecting causality in complex ecosystems, 2012.

J. E. Sulston and H. R. Horvitz.
Post-embryonic cell lineages of the nematode, caenorhabditis elegans.
Developmental biology, 56(1):110–156, 1977.

J. E. Sulston, D. G. Albertson, and J. N. Thomson.
The caenorhabditis elegans male: postembryonic development of nongonadal structures.
Developmental biology, 78(2):542–576, 1980.

V. Susoy, W. Hung, D. Witvliet, J. E. Whitener, M. Wu, C. F. Park, B. J. Graham, M. Zhen, V. Venkatachalam, and A. D. Samuel.
Natural sensory context drives diverse brain-wide activity during C. elegans mating.
Cell, 184(20):5122–5137, 2021.
ISSN 10974172.
doi: 10.1016/j.cell.2021.08.024.
URL `https://doi.org/10.1016/j.cell.2021.08.024`.

B. Szigeti, P. Gleeson, M. Vella, S. Khayrulin, A. Palyanov, J. Hokanson, M. Currie, M. Cantarelli, G. Idili, and S. Larson.
Openworm: an open-science approach to modeling caenorhabditis elegans.
Frontiers in computational neuroscience, 8:137, 2014.

J. R. Szymanski and R. Yuste.
Mapping the whole-body muscle activity of hydra vulgaris.
Current Biology, 29(11):1807–1817, 2019.

S. Tajima, T. Yanagawa, N. Fujii, and T. Toyoizumi.
Untangling Brain-Wide dynamics in consciousness by Cross-Embedding.
PLoS Comput. Biol., 11(11):e1004537, Nov. 2015.

S. Tajima, T. Mita, D. J. Bakkum, H. Takahashi, and T. Toyoizumi.
Locally embedded presages of global network bursts.
Proc. Natl. Acad. Sci. U. S. A., 114(36):9517–9522, Sept. 2017.

F. Takens.
Detecting strange attractors in turbulence, 1981.

G. K. Tam, Z.-Q. Cheng, Y.-K. Lai, F. C. Langbein, Y. Liu, D. Marshall, R. R. Martin, X.-F. Sun, and P. L. Rosin.
Registration of 3d point clouds and meshes: a survey from rigid to nonrigid.
IEEE transactions on visualization and computer graphics, 19(7):1199–1217, 2013.

L. Taslaman and B. Nilsson.
A framework for regularized non-negative matrix factorization, with application to the analysis of gene expression data.
PLOS ONE, 7(11):1–7, 11 2012.
doi: 10.1371/journal.pone.0046331.

T. Tax, P. A. Mediano, and M. Shanahan.
The partial information decomposition of generative neural network models.
Entropy, 19(9):474, 2017.

S. R. Taylor, G. Santpere, A. Weinreb, A. Barrett, M. B. Reilly, C. Xu, E. Varol, P. Oikonomou, L. Glenwinkel, R. McWhirter, et al.
Molecular topography of an entire nervous system.
Cell, 184(16):4329–4347, 2021.

T. Tekieli, E. Yemini, A. Nejatbakhsh, C. Wang, E. Varol, R. W. Fernandez, N. Masoudi, L. Paninski, and O. Hobert.
Visualizing the organization and differentiation of the male-specific nervous system of c. elegans.
Development, 148(18):dev199687, 2021.

P. Thévenaz, U. E. Ruttimann, and M. Unser.
A pyramid approach to subpixel registration based on intensity.
IEEE Transactions on Image Processing, 7(1):27–41, 1998.
ISSN 10577149.
doi: 10.1109/83.650848.

M. Thiel, M. C. Romano, J. Kurths, M. Rolfs, and R. Kliegl.
Twin surrogates to test for complex synchronisation.
EPL (Europhysics Letters), 75(4):535, 2006.

A. M. Thomson and C. Lamy.
Functional maps of neocortical local circuitry.
Frontiers in neuroscience, 1:2, 2007.

L. Tian, S. A. Hires, T. Mao, D. Huber, M. E. Chiappe, S. H. Chalasani, L. Petreanu, J. Akerboom, S. A. McKinney, E. R. Schreiter, et al.
Imaging neural activity in worms, flies and mice with improved gcamp calcium indicators.
Nature methods, 6(12):875, 2009.

N. M. Timme, S. Ito, M. Myroshnychenko, S. Nigam, M. Shimono, F.-C. Yeh, P. Hottowy, A. M. Litke, and J. M. Beggs.
High-degree neurons feed cortical computations.
PLoS computational biology, 12(5):e1004858, 2016.

J. Y. Tinevez, N. Perry, J. Schindelin, G. M. Hoopes, G. D. Reynolds, E. Laplantine, S. Y. Bednarek, S. L. Shorte, and K. W. Eliceiri.
TrackMate: An open and extensible platform for single-particle tracking.
Methods, 115(2017):80–90, 2017.
ISSN 10959130.
doi: 10.1016/j.ymeth.2016.09.016.
URL `http://dx.doi.org/10.1016/j.ymeth.2016.09.016`.

T. Tokunaga, O. Hirose, S. Kawaguchi, Y. Toyoshima, T. Teramoto, H. Ikebata, S. Kuge, T. Ishihara, Y. Iino, and R. Yoshida.
Automated detection and tracking of many cells by using 4d live-cell imaging data.
Bioinformatics (Oxford, England), 30(12):i43–i51, Jun 2014.
ISSN 1367-4811.
doi: 10.1093/bioinformatics/btu271.
URL `https://pubmed.ncbi.nlm.nih.gov/24932004`.
24932004[pmid].

E. Torgersen.
Comparison of statistical experiments.
Cambridge University Press, 1991.

P. H. Torr and A. Zisserman.
Mlesac: A new robust estimator with application to estimating image geometry.
Computer vision and image understanding, 78(1):138–156, 2000.

Y. Toyoshima, S. Wu, M. Kanamori, H. Sato, M. S. Jang, S. Oe, Y. Murakami, T. Teramoto, C. Park, Y. Iwasaki, T. Ishihara, R. Yoshida, and Y. Iino.
An annotation dataset facilitates automatic annotation of whole-brain activity imaging of c. elegans.
bioRxiv, 2019.
doi: 10.1101/698241.
URL `https://www.biorxiv.org/content/early/2019/07/18/698241`.

Y. Toyoshima, S. Wu, M. Kanamori, H. Sato, M. S. Jang, S. Oe, Y. Murakami, T. Teramoto, C. Park, Y. Iwasaki, et al.
Neuron id dataset facilitates neuronal annotation for whole-brain activity imaging of c. elegans.
BMC biology, 18(1):1–20, 2020.

M. Tsakiris and L. Peng.
Homomorphic sensing.
In *International Conference on Machine Learning*, pages 6335–6344, 2019.

G. Tucker, D. Lawson, S. Gu, and C. J. Maddison.
Doubly reparameterized gradient estimators for Monte Carlo objectives.
ICLR, 2019.

J. Unnikrishnan, S. Haghighatshoar, and M. Vetterli.
Unlabeled sensing: Solving a linear system with unordered measurements.
In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 786–793. IEEE, 2015.

J. Unnikrishnan, S. Haghighatshoar, and M. Vetterli.
Unlabeled sensing with random linear measurements.
IEEE Transactions on Information Theory, 64(5):3237–3253, 2018.

A. E. Urai, B. Doiron, A. M. Leifer, and A. K. Churchland.
Large-scale neural recordings call for new insights to link brain and behavior.
Nature neuroscience, 25(1):11–19, 2022.

V. A. Vakorin, B. Mišić, O. Krakovska, G. Bezgin, and A. R. McIntosh.
Confounding effects of phase delays on causality estimation.
PLoS One, 8(1):e53588, Jan. 2013.

O. Van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or.
A survey on shape correspondence.
In *Computer Graphics Forum*, volume 30, pages 1681–1707. Wiley Online Library, 2011.

G. C. Vanwalleghem, M. B. Ahrens, and E. K. Scott.
Integrative whole-brain neuroscience in larval zebrafish.
Current opinion in neurobiology, 50:136–145, 2018.

E. Varol, A. Nejatbakhsh, R. Sun, G. Mena, E. Yemini, O. Hobert, and L. Paninski.
Statistical atlas of c. elegans neurons.
In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 119–129. Springer, 2020.

V. Venkatachalam, N. Ji, X. Wang, C. Clark, J. K. Mitchell, M. Klein, C. J. Tabone, J. Florman, H. Ji, J. Greenwood, A. D. Chisholm, J. Srinivasan, M. Alkema, M. Zhen, and A. D. Samuel.
Pan-neuronal imaging in roaming Caenorhabditis elegans.
Proceedings of the National Academy of Sciences of the United States of America, 113 (8):E1082–E1088, 2 2016a.
ISSN 10916490.
doi: 10.1073/PNAS.1507109113.

V. Venkatachalam, N. Ji, X. Wang, C. Clark, J. K. Mitchell, M. Klein, C. J. Tabone, J. Florman, H. Ji, J. Greenwood, A. D. Chisholm, J. Srinivasan, M. Alkema, M. Zhen, and A. D. T. Samuel.
Pan-neuronal imaging in roaming Caenorhabditis elegans.
Proceedings of the National Academy of Sciences of the United States of America, 113 (8):E1082–8, feb 2016b.
ISSN 1091-6490.
doi: 10.1073/pnas.1507109113.

V. Venkatachalam, N. Ji, X. Wang, C. Clark, J. K. Mitchell, M. Klein, C. J. Tabone, J. Florman, H. Ji, J. Greenwood, et al.
Pan-neuronal imaging in roaming caenorhabditis elegans.
Proceedings of the National Academy of Sciences, 113(8):E1082–E1088, 2016c.

R. Vicente, M. Wibral, M. Lindner, and G. Pipa.
Transfer entropy—a model-free measure of effective connectivity for the neurosciences.

Journal of computational neuroscience, 30(1):45–67, 2011.

V. Voleti, K. B. Patel, W. Li, C. P. Campos, S. Bharadwaj, H. Yu, C. Ford, M. J. Casper, R. W. Yan, W. Liang, et al.
Real-time volumetric microscopy of in vivo dynamics and large-scale samples with scape 2.0.
Nature methods, 16(10):1054–1062, 2019.

Z. Wang, L. Zhu, H. Zhang, G. Li, C. Yi, Y. Li, Y. Yang, Y. Ding, M. Zhen, S. Gao, T. K. Hsiai, and P. Fei.
Real-time volumetric reconstruction of biological dynamics with light-field microscopy and deep learning.
Nature Methods, 18(5):551–556, 2021.
ISSN 15487105.
doi: 10.1038/s41592-021-01058-x.

M. Weigert, U. Schmidt, R. Haase, K. Sugawara, and G. Myers.
Star-convex polyhedra for 3D object detection and segmentation in microscopy.
Proceedings - 2020 IEEE Winter Conference on Applications of Computer Vision, WACV 2020, pages 3655–3662, 2020.
doi: 10.1109/WACV45572.2020.9093435.

C. Wen, T. Miura, V. Voleti, and K. Yamaguchi.
3DeeCellTracker, a deep learning-based pipeline for segmenting and tracking cells in 3D time lapse images.
eLife, 10:e59187, 2021.

W. Weng and X. Zhu.
U-Net: Convolutional Networks for Biomedical Image Segmentation.
IEEE Access, 9:16591–16603, 5 2015.
ISSN 21693536.
doi: 10.48550/arxiv.1505.04597.
URL `https://arxiv.org/abs/1505.04597v1`.

J. G. White, E. Southgate, J. N. Thomson, S. Brenner, et al.
The structure of the nervous system of the nematode caenorhabditis elegans.
Philos Trans R Soc Lond B Biol Sci, 314(1165):1–340, 1986a.

J. G. White, E. Southgate, N. J. Thomson, and S. Brenner.

The structure of the nervous system of the nematode caenorhabditis elegans.
Philos Trans R Soc Lond B Biol Sci, 314(1165):1–340, 1986b.

M. Wibral, J. T. Lizier, and V. Priesemann.
Bits from brains for biologically inspired computing.
Frontiers in Robotics and AI, 2:5, 2015.

M. Wibral, C. Finn, P. Wollstadt, J. T. Lizier, and V. Priesemann.
Quantifying information modification in developing neural networks via partial information decomposition.
Entropy, 19(9):494, 2017a.

M. Wibral, V. Priesemann, J. W. Kay, J. T. Lizier, and W. A. Phillips.
Partial information decomposition as a unified approach to the specification of neural goal functions.
Brain and cognition, 112:25–38, 2017b.

P. L. Williams and R. D. Beer.
Nonnegative decomposition of multivariate information.
arXiv preprint arXiv:1004.2515, 2010.

P. L. Williams and R. D. Beer.
Generalized measures of information transfer.
arXiv preprint arXiv:1102.1507, 2011.

P. Wollstadt, J. T. Lizier, R. Vicente, C. Finn, M. Martinez-Zarzuela, P. Mediano, L. Novelli, and M. Wibral.
Idtxl: The information dynamics toolkit xl: a python package for the efficient analysis of multivariate information dynamics in networks.
Journal of Open Source Software, 4(34):1081, 2019.

P. Wollstadt, S. Schmitt, and M. Wibral.
A rigorous information-theoretic definition of redundancy and relevancy in feature selection based on (partial) information decomposition.
arXiv preprint arXiv:2105.04187, 2021.

A. Wu, E. Kelly Buchanan, M. R. Whiteway, M. Schartner, G. Meijer, J. P. Noel, E. Rodriguez, C. Everett, A. Norovich, E. Schaffer, N. Mishra, C. Daniel Salzman, D. Angelaki, A. Bendesky, J. Cunningham, and L. Paninski.

Deep graph pose: A semi-supervised deep graphical model for improved animal pose tracking.
Advances in Neural Information Processing Systems, 2020-Decem:1–28, 2020.
ISSN 10495258.

Y. Wu, S. Wu, X. Wang, C. Lang, Q. Zhang, Q. Wen, and T. Xu.
Rapid detection and recognition of whole brain activity in a freely behaving Caenorhabditis elegans.
arXiv:2109.10474v3, 2021.
URL `http://arxiv.org/abs/2109.10474`.

D. Wyrick and L. Mazzucato.
State-dependent control of cortical processing speed via gain modulation.
bioRxiv, 2020.
doi: 10.1101/2020.04.07.030700.

Y. Yan, Y. Mao, and B. Li.
SECOND: Sparsely embedded convolutional detection.
Sensors (Switzerland), 18(10):1–17, 2018.
ISSN 14248220.
doi: 10.3390/s18103337.

H. Yang and L. Carlone.
A polynomial-time solution for robust registration with extreme outlier rates.
arXiv preprint arXiv:1903.08588, 2019.

H. Yang, J. Shi, and L. Carlone.
Teaser: Fast and certifiable point cloud registration.
arXiv preprint arXiv:2001.07715, 2020.

J. Yang, H. Li, D. Campbell, and Y. Jia.
Go-icp: A globally optimal solution to 3d icp point-set registration.
IEEE transactions on pattern analysis and machine intelligence, 38(11):2241–2254, 2016.

W. Yang and R. Yuste.
In vivo imaging of neural activity.
Nature methods, 14(4):349, 2017.

E. Yemini, A. Lin, A. Nejatbakhsh, E. Varol, R. Sun, G. E. Mena, A. D. Samuel, L. Paninski, V. Venkatachalam, and O. Hobert.
Neuropal: A neuronal polychromatic atlas of landmarks for whole-brain imaging in c. elegans.
bioRxiv, 2019a.
doi: 10.1101/676312.
URL `https://www.biorxiv.org/content/early/2019/06/20/676312`.

E. Yemini, A. Lin, A. Nejatbakhsh, E. Varol, R. Sun, G. E. Mena, A. D. Samuel, L. Paninski, V. Venkatachalam, and O. Hobert.
Neuropal: A neuronal polychromatic atlas of landmarks for whole-brain imaging in c. elegans.
BioRxiv, page 676312, 2019b.

E. Yemini, A. Lin, A. Nejatbakhsh, E. Varol, R. Sun, G. E. Mena, A. D. Samuel, L. Paninski, V. Venkatachalam, and O. Hobert.
Neuropal: a multicolor atlas for whole-brain neuronal identification in c. elegans.
Cell, 184(1):272–288, 2021.

J. Yu, A. Nejatbakhsh, M. Torkashvand, S. Gangadharan, M. Seyedolmohadesin, J. Kim, L. Paninski, and V. Venkatachalam.
Versatile multiple object tracking in sparse 2d/3d videos via diffeomorphic image registration.
bioRxiv, 2022.

X. Yu, M. S. Creamer, F. Randi, A. K. Sharma, S. W. Linderman, and A. M. Leifer.
Fast deep neural correspondence for tracking and identifying neurons in c. Elegans using semi-synthetic training.
eLife, 10, 7 2021.
ISSN 2050084X.
doi: 10.7554/eLife.66410.

K. Zeng, G. Erus, A. Sotiras, R. T. Shinohara, and C. Davatzikos.
Abnormality detection via iterative deformable registration and basis-pursuit decomposition.
IEEE transactions on medical imaging, 35(8):1937–1951, 2016.

J. Zhang and S. Singh.
Visual-lidar odometry and mapping: Low-drift, robust, and fast.

In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2174–2181. IEEE, 2015.

P. Zhou, S. L. Resendez, J. Rodriguez-Romaguera, J. C. Jimenez, S. Q. Neufeld, A. Giovannucci, J. Friedrich, E. A. Pnevmatikakis, G. D. Stuber, R. Hen, et al.
Efficient and accurate extraction of in vivo calcium signals from microendoscopic video data.
Elife, 7:e28728, 2018.

Q.-Y. Zhou, J. Park, and V. Koltun.
Fast global registration.
In *European Conference on Computer Vision*, pages 766–782. Springer, 2016.

B. Zitova and J. Flusser.
Image registration methods: a survey.
Image and vision computing, 21(11):977–1000, 2003.