Deep Learning Approaches Towards Semi-Supervised Cell Type Classification in Cerebellar Neuropixels Recordings

Federico D'Agostino[1]

MSc Machine Learning

Supervised by Michael Häusser

Submission date: 12 September 2022

[1]**Disclaimer:** This report is submitted as part requirement for the MSc Machine Learning at UCL. It is substantially the result of my own work except where explicitly indicated in the text. The report may be freely copied and distributed provided the source is explicitly acknowledged.

To Chiara, my parents and my sister,
who love and support me even when I am distant.

Abstract

The advent of revolutionary new multisite silicon probes equips researchers with tools to record simultaneously from an unprecedented number of neurons, opening new avenues for the description of neural circuits. However, numerous molecular, morphological, functional and connectional characteristics distinguish neurons in the brain into different cell types, and reliable identification of such cell types in extracellular recordings is essential to determine their contributions to neural circuit computations. This is difficult in any brain area, but particularly challenging in the cerebellar cortex due to the high density of neurons, their high firing rates, and the elaborately folded cytoarchitecture. Previous studies tried to solve this problem using supervised learning, but lacked rigorousness in their machine learning pipelines and used datasets coming from anaesthetised animals. Here we tackle the problem using a novel dataset coming from high-density Neuropixels recordings of the cerebellar cortex in awake, freely moving mice. Due to the complexity of the experimental protocol for data acquisition, only a small amount of the data is labelled. As a consequence, for the first time in this domain, we adopt a range of modern deep semi-supervised methods to approach the task, making the most efficient possible use of ground-truth information. Results show how our models are able to surpass in accuracy both human experts and a baseline constructed with engineered features from the electrophysiology literature, in some cases using just only a fraction of the total labels available. We further propose concrete steps to bring our model architectures into deployment, to yield a tool that can be reliably incorporated into the analysis pipelines of electrophysiology laboratories across the world. Our broader hope is to inspire researchers in biology to make a more resource-aware use of data, especially when coming from costly and time-consuming experiments.

Contents

Preface

If you try to please all, you please none.

Aesop

As the title page recites, "*this report is submitted as part requirement for the MSc Machine Learning at UCL*". However, given the subject of the matter and my academic background, a substantial amount of this writing will be related to Neuroscience. I recognize that, while I tried to be as concise as possible and at the same time include all necessary relevant facts, the quantity and content of background information could at times seem either excessive or incomplete, depending on which field the reader comes from. Nonetheless, it is my hope that the following work will be both accessible and informative to neuroscientists and computer scientists alike, as I have kept both audiences in mind from the beginning.

Chapter 1

Introduction

It was only less than three centuries ago when scientists started to speculate on the electrical essence of neurophysiology [1], and the first precise and analytical characterisation of the main electrical signature of neurons, the action potential, is merely less than a century old [2]. Since then, systems neuroscience and electrophysiology together have made immense progress in unveiling the fundamental characteristics of communication in neural populations.

A large amount of this progress can be attributed to the constant development of new recording techniques, which, resembling Moore's law, have been doubling the amount of simultaneously recorded neurons approximately every 7 years [3]. However, as neuroscience advances and methods to record neural signals from alive, behaving animals evolve and improve, the data generated by experiments inevitably grows in both magnitude and complexity. As a consequence, practically the totality of data produced with contemporary probes and multi-electrode arrays [4] would be uninterpretable without companion software to aid experimenters.

Take as an example Neuropixels [5, 6], a novel class of high-density silicon probes capable of recording from hundreds of neurons simultaneously. To be understandable, multi-channel neural activity thus recorded needs to be processed with specialised *spike sorting* software (see section 2.2.1; [7]), which attempts to isolate the activity (or *spikes*) of single neurons in a sea of noisy, high-dimensional data.

While isolating the activity of single neurons in population recordings is an extremely important - and difficult - first step in the decoding of neural computation, there is still a crucial missing piece of information that systems neuroscientists need to reach a definitive description of neuronal circuits: which cell types they are recording from. Working on this missing link will be the focus of our work.
But what are neuronal cell types, and why are they important for our understanding of the brain?

1.1 Problem statement and motivation

Unlike artificial neurons in deep neural networks, neurons in the brain come in an astounding variety of cell types, each with their biophysical and morphological characteristics which influence their connectivity and information processing properties within neural circuits [8]. Therefore, beyond the already important yet easier to infer distinction between excitatory and inhibitory neurons, knowing the exact cell type of a neuron is of exceptional value in decoding its contributions to neuronal computation.

Unfortunately, there are several reasons why retrieving the cell type of a neuron from electrophysiological recordings *in vivo* is non-trivial, especially when using contemporary silicon probes such as Neuropixels [5, 6].

First, data from high-density probes is inherently noisy and high-dimensional, generally needing meticulous consideration in any analytical scenario. The reason for this is that any neuron is recorded over multiple channels in space, but every channel will likely receive activity from more than one neuron. Clustering the activity of different neurons from extracellular recordings has been a challenge since the dawn of the field [9, 10], but here we will effectively take this step as solved, and only as a source of noise in our modelling.

Secondly, and most importantly for our task, getting ground-truth data about cell identities when recording them with extracellular probes involves complicated and time-consuming experimental manipulations, especially when performed on live, freely-behaving animals. Even once performed, such experiments have a very low yield of labelled data compared to unlabelled data, calling for the most resourceful possible use of ground-truth information.

Finally, within each cell type, there is not only a significant deal of biological variability influencing the recorded responses, but also an unbounded number of relative arrangements between the silicon probe and the recorded cell, ultimately also affecting variance in the data.
To do well, machine learning methods need to resolve both types of variability.

Previous work in the field of cerebellar cell type classification has focused heavily on fully supervised, feature engineering approaches [11–13], completely disregarding the possibility of using cheaper, and more abundant, unlabeled data to assist the classification process. Moreover, placing themselves in line with a strong tradition of computational modelling and feature extraction from neural activity [14–16], previous studies never completely rely on machine learning methods for feature learning.

Although still appreciative and much indebted to past efforts, here we seek to move past feature engineering approaches for neuronal cell type classification by leveraging achievements in the the rapidly moving subfield of machine learning known as semi-supervised Learning (SSL; [17]). This has no precedent in the cell type classification literature and will allow us to provide new avenues for efficient data use in the task, in alignment with the recent growth of SSL applications in Neuroscience [18–22].

1.2 Aims and research directions

In this study, we will have at our disposal an invaluable dataset of Neuropixels recordings of experimentally identified ground-truth cell types in the cerebellar cortex ($n = 77$), along with the activity of many more unlabelled neurons ($n = 877$), acquired in the Häusser lab over the past 3 years. Importantly, all data comes from live, freely moving animals, an important change of direction from previous research that had limited applicability due to constrained recording conditions.

The current investigation aims to use the dataset to construct a robust machine learning pipeline for semi-supervised cell-type classification in the cerebellum which can be aligned with researchers' needs and readily be adopted by laboratories working with Neuropixels. To accomplish this, two competing yet complementary approaches were followed.

On the one hand, to satisfy a need for interpretable results that reflect the neurons' biological characteristics, we sought to improve existing tree-based methods that work with engineered features. Specifically, the aim behind this first line of research was to unify existing, well-understood engineered features for cell type classification with unsupervised representation learning techniques that could make the most of unlabeled data while still working with relatively interpretable tree-based classifiers.

On the other hand, we sought to abandon a feature engineering approach by adopting contemporary Deep Semi-Supervised methods. These would sacrifice some interpretability at the supposed advantage of improved accuracy, better use of unlabeled data, and efficient compatibility with data augmentation strategies. Applied in conjunction with Bayesian deep learning tools for better uncertainty calibration, deep semi-supervised methods can accurately inform experimenters' decision-making while leveraging the latest developments in Machine Learning research. Moreover, in this case, engineered features could still be used retrospectively on classifiers' results to regain explainability.

1.3 Summary of contributions

Following our two central research aims, the main contributions of the present work toward cell type classification in the cerebellum can be summarised as follows:

1. We let expert electrophysiologists predict the labels of all ground-truth instances in our dataset, recording their performance through a custom web application [1] and setting an expert baseline for the machine learning models.

2. We revised and generalised the pipeline to compute engineered features from Neuropixels data, contributing to the open-source electrophysiology Python package `npyx` [23]. Additionally, we developed a dashboard to visualise the outcomes of the feature extraction process and organise in the same place all sources of information going into the creation of the dataset.[2]

3. A rigorous and reliable modelling pipeline was introduced for the classification task, including hyperparameter optimisation with Bayesian optimisation [24, 25], model selection with stratified cross-validation and performance reporting with leave one out cross-validation.

4. We created a custom, expert-derived and biologically plausible set of data augmentations for neural data, which we used when building all of our deep learning models.

5. For the first time in this domain, we applied representation learning methods to neural responses, demonstrating how meaningful data manifolds that capture essential variability in the data can be learned by variational autoencoders.

6. Again for the first time, we applied deep semi-supervised methods to the cerebellar cell-types classification problem, showing how they can rival the performance of competitor models with vastly more efficient use of labelled data.[3]

1.4 Thesis outline

The contributions just outlined will be presented following a traditional exposition.

First, we will go into the details of the prerequisite knowledge needed to fully understand the directions taken in further chapters.

Following that, we will briefly offer a critical evaluation of past work in the field of cerebellar cell type classification, building up to our own contributions.

In describing our work, we will compare and contrast the performances of different models, deriving some important conclusions about the dataset from the inspection of the most common mistakes.

Finally, future objectives are examined, with particular attention to the specific steps that need to be taken for our models to be adopted by the wider research community.

[1] `https://files.fededagos.me/guess/`

[2] `https://files.fededagos.me/features/`

[3] We release all code and data for the experiments at `https://github.com/fededagos/celltypes-classification`

Chapter 2

Theoretical background

This chapter discusses the fundamental prerequisites needed to understand the cell types classification problem and the rationale behind the proposed solutions.

2.1 The cerebellum

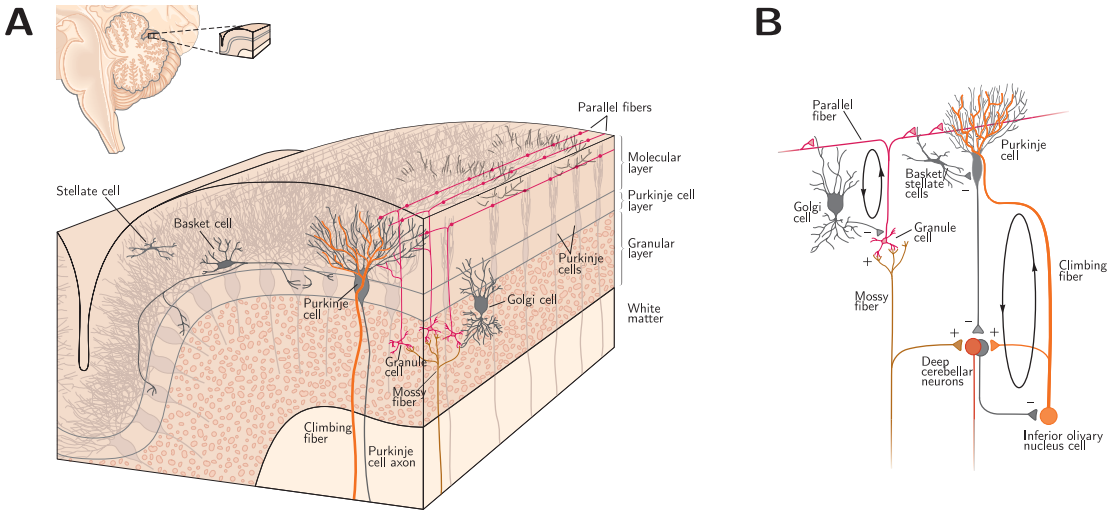


Figure 2.1: Organisation of the cerebellar cortex. **A**. Schematic of a vertical section of a single cerebellar folium. Note how different cell types are found in separate layers. **B**. Diagram illustrating convergent inputs onto the Purkinje cell from parallel and climbing fibres and from local circuit neurons. Recurrent loops involve Golgi cells within the cerebellar cortex and the inferior olive outside the cerebellum. Excitation and inhibition in the microcircuit are indicated by + and − signs. Figures and captions adapted from [26, 27].

2.1.1 Structure and function of the cerebellum

The cerebellum is a structure that lies underneath the occipital and temporal lobes of the cerebral cortex in the rear of the brain. Over 50 % of the central nervous system's total neurons are found in the cerebellum, even though it makes up just about 10 % of the brain's overall volume [26]. Historically thought of as a structure exclusively dedicated to motor control due to the pronounced motor symptoms of cerebellar damage [26, 27], the cerebellum is being recently appreciated for its roles in cognition [28, 29], reward processing [30, 31], and autism [32], among others.

The cerebellum can be divided into different areas according to either anatomical or functional criteria, with each region receiving projections and projecting back to different areas of the central nervous system. In spite of this, all areas share similarities at the cellular level, and, most interestingly, in how the microcircuits are organised. As a consequence, it is widely believed

that different regions of the cerebellum perform essentially similar computations on a variety of different inputs.

The current consensus is that internal models of the body are stored in the arrays of parallel fibre-Purkinje cell synapses in the cerebellar cortex, tuned by the interplay between climbing and parallel fibre firing which engages long-term plasticity mechanisms. These internal models, continuously updated throughout life, use sensory and motor information available in the present (e.g. muscle tension, perceived obstacles) to predict the sensory consequences of motion in the immediate future. It is believed that different cerebellar modules might form similar predictions in different spaces, some directly related to our senses (3D space) and others more abstract (e.g. predicting upcoming social interactions given current facial expressions). The concept of internal models is not exclusive to the cerebellum, but the specificity of cerebellar computations is the time scale at which they operate: on the order of milliseconds [26, 27].

2.1.2 Cell types and cerebellar microcircuitry supporting cerebellar function

The cerebellum is composed of two structures: the cerebellar nuclei, the output stage of the cerebellum, and the cerebellar cortex foliated around the nuclei. The cerebellar cortex projects onto the nuclei via the Purkinje cells (PkCs), its exclusive source of output, and features a layered organisation.

The *granular layer* is the input layer of the cerebellar cortex. It contains a vast number of small, densely packed excitatory granule cells (GrC), accounting for most of the cells in the cerebellum. GrCs fire sparsely in bursts of activity. This layer also contains the terminals of the first source of input of the cerebellum: the excitatory mossy fibres (MFBs), which originate mainly from the pontine nuclei. [26]. MFBs synapse directly onto the cerebellar nuclei as well as onto GrCs. MFBs (and subsequently GrCs) carry sensorimotor information originating from the whole body and every sensory modality. The granular layer also contains Golgi Cells (GoCs), very large interneurons which perform both feedback and feedforward inhibition onto the GrC dendrites and axons, respectively. Finally, this layer also comprises a few other rarer cell types not considered in this work (such as Lugaro cells, chandelier cells, and unipolar brush cells, which are found selectively in some folia).

The *Purkinje cell layer* consists of a single sheet of Purkinje cell (PkC) bodies. As mentioned earlier, they project onto the cerebellar nuclei and thus constitute the sole output of the cerebellar cortex. These are among the largest neurons found in vertebrates, characterised by a dense, intricate and elaborate dendritic arbour which extends into the molecular layer [26]. PkCs are the main hub of the cerebellar cortex, collecting a myriad of information from about 150,000 GrCs each, in a phenomenal example of convergent input.

The second input of the cerebellum, the excitatory climbing fibres (CF), originate from the inferior olive and make a very strong contact on each PkC - in the adult brain [33], each PkC receives inputs from a single CF. They carry a feedback signal teaching the PkCs which subsets of GrC inputs matter: the respective timing of CFs and GrCs rules the synaptic weight updates between GrCs and PkCs [34]. Importantly, CFs and GrCs elicit different types of action potentials in PkCs: while GrC inputs modulate the spontaneous firing of PkCs (called simple spikes), CFs input elicit a massive depolarisation in the whole PkC dendrite, termed complex spike. PkC simple and complex spikes thus reflect the firing of PkCs and CFs, respectively.

The *molecular layer* is an important processing layer of the cerebellar cortex. It is where the PkC dendritic arbours receive both their GrC and CF inputs. They cover extensive regions of space in the anterior-posterior direction but do not spread far in the medial-lateral direction (in other words, they roughly cover a 2D plane in 3D space). The GrC axons run always parallel to the *folia*, the highly convoluted folds on the surface of the cerebellum, in a mediolateral direction. Thus each GrC axon, running perpendicular to the PkC dendritic arbours, has the potential to synapse onto a large number of Purkinje neurons (Figure 2.1A) [26]. The molecular layer also contains the eponymous inhibitory molecular layer interneurons (MLIs), subdivided into the basket and stellate cells. They constitute a functional syncytium, electrically coupled

with each other in the sagittal plane [35], and exert feedforward inhibition between the GrC axons and the PkCs.

It is yet unclear where the predictions about bodily and external events are generated: learning occurs both at the GrC-PkC and the MFB-cerebellar nuclei synapses, and internal models could be stored and used to generate predictions in either of these places [36].

In summary, the diverse cerebellar cortical cell types relay rather different signals and perform very specific types of computations (see figure 2.1B): it is thus crucial to be able to identify them in a given neural recording to enable us to study cerebellar function at all.

2.1.3 Key signals in cerebellar processing

To recapitulate, there are several main signals and/or cell types to be distinguished for a satisfactory understanding of cerebellar processing:

- Purkinje cell complex spikes (`PkC_cs` for our classifier), elicited by climbing fibre inputs, which represent the teaching signal for PkCs.
- Purkinje cell simple spikes (`PkC_ss` for our classifier), which represent the output of PkCs, main hub of the cerebellar cortex, and are modulated by GrC inputs.
- Golgi cells (`GoC` for our classifier), which perform feedback and feedforward inhibition onto GrC dendrites and axons, respectively.
- Molecular layer interneurons (`MLI` for our classifier), which perform feedforward inhibition between GrCs and PkCs.
- Mossy fibres (`MFB` for our classifier), which are the second major source of cerebellar input.
- Granule cells' activity (`GrC` for our classifier), which multiplex MFB activity and relay it to PkCs, GoCs and MLIs.

2.2 Neuronal electrophysiology

Neurons are excitable cells that communicate with one another in terms of action potentials (also referred to as "spikes"), "the signals by which the brain receives, analyzes, and conveys information" [26]. Spikes are highly stereotyped, and the information they convey less determined by the shape of the signal than it is by the pathway they travel in, their frequency, and their timing relative to external events. Even if it does not primarily communicate coding information for the neurons themselves, the shape of action potentials differs considerably among various types of neurons [37]. This can be seen under various recording conditions, including extracellular recordings *in vivo* [37], and will be one source of information to build our classifier system, along with the temporal characteristics of a neuron's spike train. In order to understand the challenges in classifying cell types from neuronal recordings, we need to appreciate some fundamentals of electrophysiology.

The membrane voltage, or potential, at any given time, is defined as the difference between the intracellular and the extracellular potential. In essence, a spike is a quick series of voltage changes across the neuronal cell membrane, which is actively propagated by ions flowing in and out of the cell following a series of fixed phases (Figure 2.2B).

Extracellular recordings in electrophysiology are used to record action potentials through currents that are induced to flow in the extracellular space around an active neuron (Figure 2.2C; [38]).

Volume conductor theory offers a simple method for understanding these current flows. This method imagines the extracellular media around the neuron as a "volume conductor", which has a low uniform Ohmic resistivity ρ. Under these circumstances, the electrical potential in the extracellular space is governed by Laplace's equation [14, 39]:

$$\nabla^2 \Phi = 0, \tag{2.1}$$

where Φ is the extracellular potential. At the boundaries, $(1/\rho)\Phi = \mathbf{J}_m$, where $\mathbf{J}_m$ is the transmembrane current density and ρ is the extracellular resistivity [39]. The issue of point charges in free space from classical physics (Coulomb's law) has a counterpart solution for a single point source of amplitude I in an unbounded isotropic volume conductor [14, 39], which we can express as

$$\Phi = \frac{\rho I}{4\pi r},$$

where I is a point source of current and r is the distance from the source to the measurement. In biological neurons, however, the matter is more complex as membrane currents are dispersed along extended cylindrical processes, whose length is far greater than their width [39].

One may picture a single axon (the output, threadlike part of the neuron that usually conducts impulses away from the cell body) in a saline solution as the most basic scenario. The membrane potential is constant along the length of the axon while it is at rest, and no current is flowing within or outside the cell [38].

The potential difference between the depolarized and resting areas will, however, cause current to flow if the axon becomes depolarized somewhere along the membrane. A current "sink" is the term used to describe the active area [38]. Biologically, this corresponds to the flow of Na^+ ions into the cell, which depolarise the membrane and at the same time go missing in the extracellular space. The depolarization then spreads axially along the axon to neighbouring membrane sections, where capacitive and Ohmic membrane currents serve as a "source" of current for the extracellular space. An electrode that is close to the axonal membrane will record this as a negative deflection, because current flows inward at the active area. A distant electrode will be almost indifferent to the local change.

Mathematically, this can be expressed by the more involved line source approximation (LSA; [14]), which, for a single linear current source (e.g. an axon) of length Δs, gives the potential $\Phi(r, h)$ as

$$\begin{aligned}\Phi(r,h) &= (\rho/4\pi)\int_{-\Delta \mathrm{s}}^{0} I \ \mathrm{d}s/\Delta s\sqrt{r^2+(h-s)^2} \\ &= (\rho I/4\pi\Delta s)\log\left|\left[\sqrt{h^2+r^2}-h\right]/\left[\sqrt{l^2+r^2}-l\right]\right|,\end{aligned}$$

where h is the longitudinal distance from the end of the line, r is the radial distance from the line, and $l = \Delta s + h$ is the distance from the start of the line [39]. As a spike propagates along an axon, different parts of it go from being current sources to sinks, constituting an evolving current dipole which is what ultimately allows extracellular recordings.

Importantly, if we now move away from our idealised "single axon in a saline solution" scenario, and consider that each neurite is modelled by its own LSA, we can appreciate how extracellular spikes will have different shapes depending on the placement of the recording electrode relative to the neural cell body, dendrites and axon (Figure 2.2A), as these will influence the size and magnitude of the dipole.

Moreover, it should now also be clear how the size and morphology of the neurons (i.e. their cell types) also directly influence the current flow around the cell during an action potential [40]. A spike in a tiny cell, for example, will create a smaller total transmembrane current, resulting in a lower extracellular current. Furthermore, cells with extended dendritic trees will generate currents across a larger area than cells with short or thin dendrites.

The above considerations, combined with the fact that current dipoles produce an extracellular signal that decreases as the squared inverse distance to the dipole [40] directly imply that recording from small cells is doubly difficult because their current sources and sinks are both smaller and closer to one another. This has a two-fold practical implication for smaller cell types, such as Granule cells. On the one hand, they will express - on average - more distinctive features in their waveforms across space, such as higher spatial decay of the extracellular peak amplitude (i.e. they will be recorded on less channels). On the other hand, this also means that they will be under-represented in the dataset given the operative difficulties in recording them.

To conclude, consider that for explanatory purposes thus far we have considered the extremely simplified case where we record from a single cell, even if modelled by multiple LSAs. In reality, an electrode placed in the extracellular medium will likely be close to more than one neuron, with possible relative arrangements being uncountable, further complicating the recording landscape.

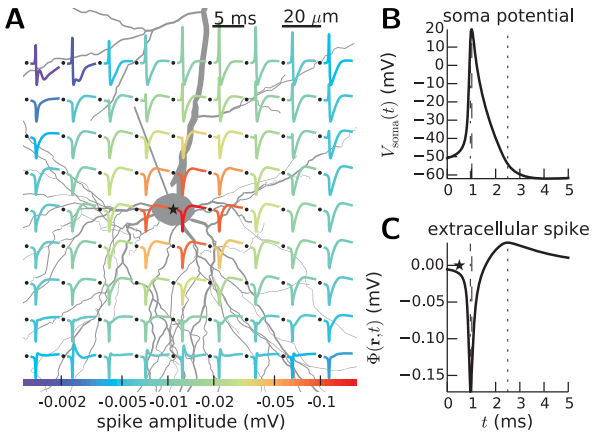


Figure 2.2: **A.** Simulated position-dependent extracellular spike waveforms during an action potential in a rat L5b pyramidal-cell model. Black dots are the (putative, virtual) electrode contact points. **B.** Somatic membrane potential as would have been recorded by an intracellular patch electrode at the point denoted by a star in A. **C.** Corresponding event to B but recorded extracellularly. Dotted lines indicate temporal alignment. Note how the voltage in C on the y axis is also dependent on $\mathbf{r}$, the position of the electrode relative to the cell body, often unknown. Original figure and adapted caption from [41].

Nonetheless, thanks to the condition in equation 2.1, and the purely Ohmic resistivity of the extracellular milieu, multiple current sources combine linearly [14, 39], meaning that discerning different signal sources is "only" a matter of linear source separation. This is what allows "spike sorting" (Section 2.2.1).

2.2.1 Modern techniques in electrophysiology

Techniques to record the activity of neurons in the extracellular medium have seen tremendous developments in the last 50 years. The first tungsten microelectrode to record from single neurons extracellularly was developed only in 1957 [42] to satisfy practical requirements unmet by the glass microelectrodes for intracellular recordings, such as the possibility to record chronically in unrestrained animals.

Today, modern silicon probes such as Neuropixels 1.0 and 2.0 [5, 6], which are the technologies adopted in our research, can record from hundreds of neurons simultaneously.

Neuropixels probes (Figure 2.3) are high-density electrodes with 960 recording sites distributed on the probe shank, which can record extracellular activity at a high frequency (30 kHz) from 384 channels simultaneously over about 4mm of brain tissue [5]. A considerable advantage of the probe design is the possibility of recording from different cortical layers simultaneously, a particularly attractive feature in cerebellar recordings due to the stereotypical nature of the cerebellar circuit. However, given the small size of the contacts ($12 \times 12 \mu$m) and their associated low impedance, Neuropixels probes capture signals that have high noise on single channels. Yet, the fact that each cell is often recorded across multiple channels on the probe offers redundant information that is used to recover the signal-to-noise ratio and identify action potential events.

A question however needs addressing: how does one interpret data coming from high-density recordings of potentially hundreds of cells simultaneously? Surely it is not as straightforward as interpreting recordings from a single extracellular electrode? Indeed, raw recordings from modern probes look almost like meaningless noise if not carefully pre-processed. An essential step in the pre-processing pipeline, which will also be a prerequisite for our classification task, is spike sorting.

Spike sorting

Spike sorting (Figure 2.4) is defined as "the grouping of spikes into clusters based on the similarity of their shapes" [7]. Relying on the rather mild assumption that each neuron will tend to fire

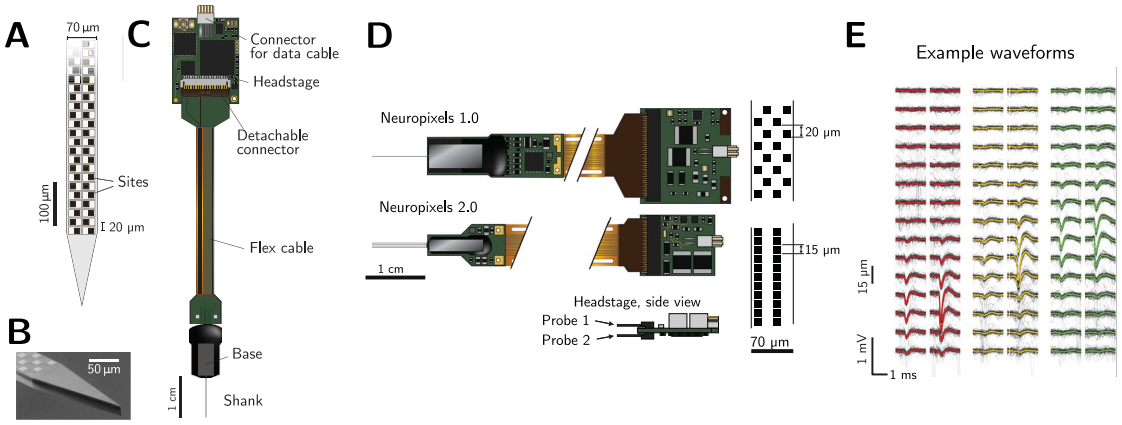


Figure 2.3: Neuropixels 1.0 and 2.0 probes. **A**. Schematic of the probe tip illustrating the characteristic checkerboard layout of the recording sites (dark squares). **B**. Scanning electron microscope view of the probe tip. **C**. Illustration of the full device packaging, including cable and headstage for data transmission. **D**. Comparison between Neuropixels 1.0 and 2.0. The latter have four shanks, a smaller headstage, a more compact channel arrangement and allow attaching two probes to a single headstage. **E**. Example of three spike-sorted waveforms recorded in overlapping channels with 2.0 probes in the olfactory cortex of an awake, head-fixed mouse. Mean waveforms (in colour) are overlayed on 50 random individual traces. Figures and captions adapted from [5, 6].

spikes of a characteristic shape, spike sorting software retrieves the activity of putative neurons (sometimes referred to as "units") based on spike clusters with a similar shape through a process of template matching. The sorting process is an essential step of all extracellular recordings, even if acquired with single electrodes, given that more than a single cell is likely to be in the vicinity of a given electrode at any time [43]. However, the sorting process becomes even more essential and technically difficult in high-density, multi-channel recordings.

Spike sorting is indeed a very active area of research [44–47], showing increasing complexity in the way the newest algorithms attempt to recover unit identities. Some examples include correcting for probe drift, probe shift, adapting the template over time and using multi-channel templates.

Building on such efforts, we wish to further extend the amount of information that can be extracted from multi-electrode recordings with cell type information. In doing so we need to bear in mind some final considerations on cerebellar electrophysiology.

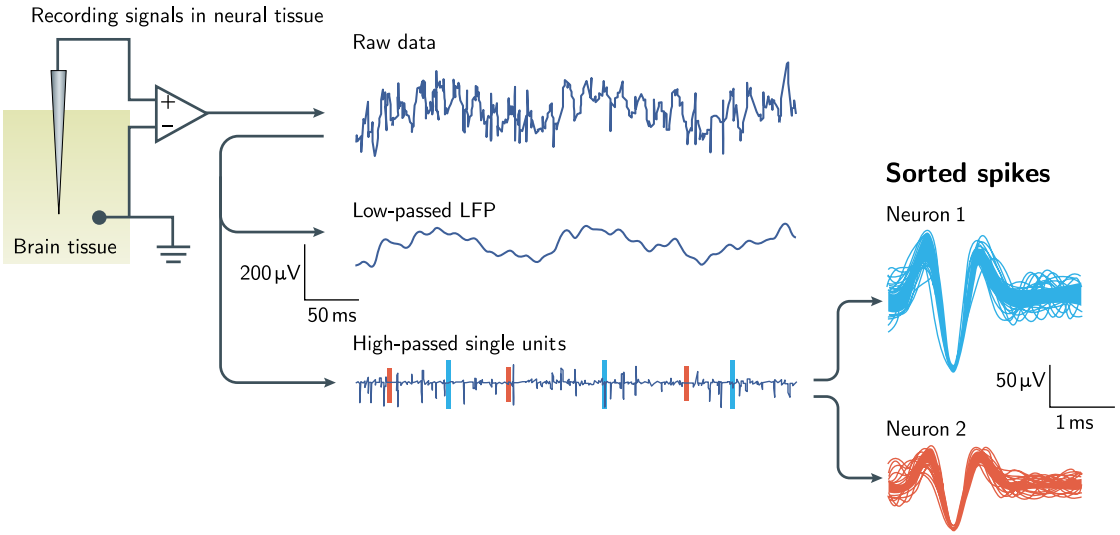


Figure 2.4: Simplified schematic of the spike sorting process. Traces on the right contain 50 overlapping sorted spikes aligned to their peak. Adapted from [4].

2.2.2 Distinctive issues in cerebellar electrophysiology

The cerebellum contains the widest range of neuronal sizes in the brain, from small granule cells to Purkinje cells and Golgi cells [40]. Due to the physical properties of modern electrodes, it is much easier to record from large cells such as Purkinje and Golgi cells than it is to record from interneurons and granule cells. Indeed, to record from such small cells, a set of ideal conditions would need to be satisfied, most notably having a short distance from the probe.

Another issue of cerebellar recordings - which applies also in most brain areas - is the disruption caused by the electrode. Even if modern electrodes reach microscopic sizes, they are still at least 3 or four times larger than neuronal cell bodies, and orders of magnitude larger than dendrites and axons. When a multi-site electrode is inserted in the brain it will therefore inevitably damage the surrounding neuropil. Due to the peculiarly intricate structure of the cerebellar cortex, in particular of the molecular layer, it is safe to assume that there will be a degree of disruption to the local circuit where the electrode is inserted.

Lastly, there are cerebellum-specific precautions to be taken in spike sorting as well, as PkC complex spikes are for example best found in signals filtered with lower frequency bands than for other neurons.

2.3 Temporal properties of neurons: spike statistics

As previously acknowledged, action potentials pass on information through their rate and timing. Calculating spike statistics is thus of paramount importance when characterising the behaviour of a neuron, and, as a consequence, also when trying to determine its identity. Two statistical constructs are often used in theoretical neuroscience to summarise the temporal properties of neurons.

Given a spike train of length n, occurring at times t_i for $i \in [n] \coloneqq \{1, \dots, n\}$ (Figure 2.5A), the inter-spike interval (ISI) distribution is the probability density of time intervals between adjacent spikes, and is a valuable statistics for describing spiking patterns [48]. In practice, this is often represented by the ISI histogram (Figure 2.5B), calculated by taking the difference between all spike times in a recording and plotting their counts with a chosen bin size.

A generalisation of the ISI distribution is the spike-train autocorrelation function or autocorrelogram (ACG) which measures the distribution of times between any two spikes in a given spike train [48]. The ACG is also operatively represented by a histogram (Figure 2.5C) constructed from the data by selecting both a bin size to divide time and a window size to determine the time look-back and look-ahead used to consider surrounding events for each spike. So, for the m-th bin, with $m \in \{-M, \dots, -1, 1, \dots, M\}$, the ACG value for that bin is computed by counting the number of times any two spikes are separated by a time in the interval $(m - 1/2)\Delta t$ and $(m + 1/2)\Delta t$, with Δt the bin size and $1/2M\Delta t$ the window size.
The ACG usually contains more information than the ISI distribution since it reflects temporal relationships not only between adjacent but between all spikes, and is particularly used in detecting oscillations and regularity in spike trains [48].

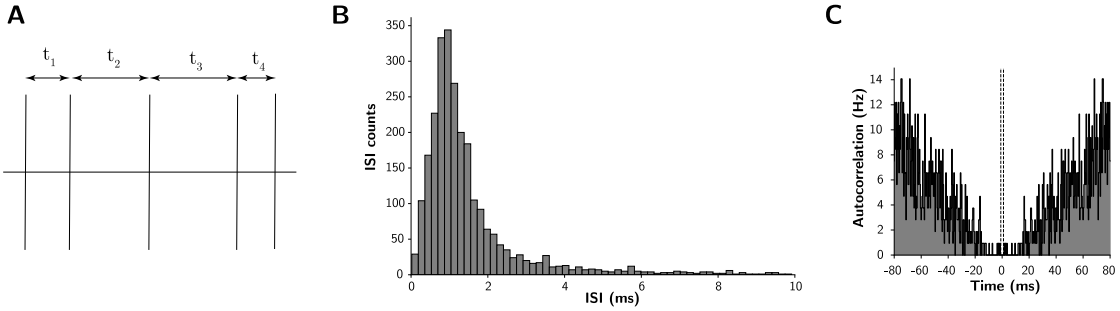


Figure 2.5: Schematic of the interspike interval, related histogram and an example spike train autocorrelation function. **A.** Schematic of interspike intervals for a train of action potentials. **B.** Example interspike interval histogram for a Granule cell in our dataset during the 20 minutes period of spontaneous activity, calculated with bin size ≈ 0.2 ms. **C.** Corresponding autocorrelogram, calculated with a bin size of 0.2 ms and a window size of 160 ms. Dotted lines indicate the refractory period.

Having exhausted all of the Neuroscience-related background material needed to understand our exploration, we are now ready to move onto the Machine Learning concepts that guided and motivated our experiments.

2.4 Introduction to Semi-Supervised Learning [1]

Usually, two different types of tasks are discerned in Machine Learning, based on whether the data at hand is annotated ("labelled") or not.

In Unsupervised Learning, data is not labelled: the learning algorithm receives observations $\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, where feature vectors $\mathbf{x}_i \in \mathcal{X}$ with $i \in [n] \coloneqq \{1, \dots, n\}$ are typically assumed to be independent and identically distributed (i.i.d.) samples of some distribution $\mathcal{X}$. The goal here is to identify the structure in the data, describe its patterns and potentially retrieve a tractable estimate of the probability density $\mathcal{X}$.

In Supervised Learning, the data is labelled: the learner receives a training set of examples $\mathcal{D} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$, where feature $\mathbf{x}_i \in \mathcal{X}$ and label $y_i \in \mathcal{Y}$ pairs are i.i.d. samples from $\mathcal{X} \times \mathcal{Y}$. The goal is to infer a function $f_{\mathcal{D}}(\mathbf{x}_i) \approx y_i$ that approximates the mapping from $\mathcal{X}$ to $\mathcal{Y}$ in the data generating process.

Semi-supervised learning (SSL) can be considered the middle point between supervised and unsupervised learning: the data is partially labelled. In the typical SSL scenario, the learner has access to observations $\mathcal{D}$ divided into $\mathcal{D}_l = \{\mathbf{x}_1, \dots, \mathbf{x}_l\}$ for which targets $\mathcal{Y}_l = \{y_1, \dots, y_l\}$ are provided, and $\mathcal{D}_u = \{\mathbf{x}_{l+1}, \dots, \mathbf{x}_{l+u}\}$ for which labels are unknown, with $|\mathcal{D}_u| \gg |\mathcal{D}_l|$ in most applications. Generally, the goal of SSL algorithms is to use the unlabeled dataset $\mathcal{D}_u$ to obtain a function $f_{\mathcal{D}}(\mathbf{x}_i) \approx y_i$ which is more precise than it would have been if we had only used $\mathcal{D}_l$ for training [50].
However, based on their specific aim, SSL algorithms can be further divided into two subcategories.

In *transductive learning* the aim is to use the trained classifier[2] at test time to infer the classes of the unlabeled instances observed during training. Formally, we only want to learn a function $f : \mathcal{X}^{l+u} \mapsto \mathcal{Y}^{l+u}$ that is expected to be a good predictor of the unlabeled data $\{\mathbf{x}_j\}_{j=l+1}^{l+u}$ [49].
In contrast, *inductive learning* has the goal of outputting a prediction function which can generalize to unseen instances from $\mathcal{X}$ at test time. Formally, this means we want to estimate a function $f : \mathcal{X} \mapsto \mathcal{Y}$ to be a reliable predictor beyond already observed unlabelled points $\{\mathbf{x}_j\}_{j=l+1}^{l+u}$ [49].
While many of the historic SSL methods focus on transductive learning [17], modern Deep SSL methods focus more on the difficult task of inductive learning [50], and will be at the centre of our discussion in subsequent sections.

2.4.1 Motivating Semi-Supervised Learning

Before delving into the details of SSL methods, it is worth stopping to briefly consider why SSL is such a valuable framework in a variety of tasks, including the one we are dealing with.

The most abundant and cheap type of data in the real world is unlabeled data, and the main reason for this is that getting labels is expensive. It could be expensive time-wise, meaning that annotators need to spend long periods of time going through unlabelled instances to create a supervised training set, or because the labelling process itself involves long procedures. It could also be expensive in a monetary sense, for instance, because annotators need to be experts, or more trivially to incentivise what would be an otherwise tedious and unappealing task. In the case of cell-type classification, labelled data is expensive even beyond these reasons, because, as we will see in section 4.1, labelling neurons involves time and resource-consuming experiments.

[1] Most of the content of this section is heavily based on [17] and [49]

[2] Though SSL is also applicable to some regression problems, from here on we will always assume to be in a classification scenario, as it is the most common and active area of SSL development.

To conclude this concise reflection, it should come as no surprise that in many tasks we would want to leverage the power and abundance of unlabeled data over labelled data. The issue is that this is not always straightforward, as a number of assumptions need to hold for SSL to work as intended.

2.4.2 Assumptions

- **Smoothness Assumption** "*If two points that reside in a high-density region are close, then so should be their corresponding outputs*" [17, 50]. This also directly implies that if two points are separated by a low-density region, then their outputs should not be close.
- **Cluster Assumption** "*If two or more points are in the same cluster, they are likely to be of the same class*" [17, 50]. Perhaps this is the most natural assumption, given that in any classification problem, almost by definition, classes are likely to cluster together. A direct implication of this assumption is that any decision boundary should lie in a low-density region.
- **Manifold Assumption** "*High dimensional data should lie (roughly) on a low dimensional manifold*" [17, 50]. When this holds, algorithms can deal more easily with the "curse of dimensionality", ensuring more accurate density estimations and more reliable distance metrics.

Most SSL algorithms either directly or indirectly rely on one or more of these assumptions. Even if they may look relatively innocent, not all tasks will necessarily follow them, a crucial point we will keep in mind when interpreting our results.

Let us now continue this background discussion with some recent research directions in SSL.

2.5 Deep Semi-Supervised methods

In the last decade, an increasing amount of research in the Deep Learning community has been devoted to SSL [50]. Efforts have been broadly directed into different approaches:

- *Consistency regularisation*, where models are trained to give consistent predictions on different, realistic, perturbations of unlabeled data points.
- *Proxy-label methods*, where the trained model on the labelled set is used on the unlabeled set to produce additional labels for subsequent runs, in a form of bootstrapping.
- *Generative models*, used to learn feature representations on labelled and unlabelled data, which can then be transferred to other tasks.
- *Graph-based methods*, similar to more traditional methods such as spectral clustering [51], where data points are considered as nodes on a graph, and the label information is propagated to unlabeled nodes using specific similarity metrics.

Of direct interest to us will be the first three techniques, which we will now survey in a bit more detail.

2.5.1 Generative models and representation learning

Unlike discriminative models, which only aim to learn the most accurate predictor given the data (i.e. $P(Y|X = x)$), generative models deal with the more general task of learning a joint distribution over all the variables [52] (i.e. recovering the distribution $P(X, Y)$ which is most likely to have generated the data).
Due to its inherently probabilistic nature, generative modelling is often more computationally expensive than discriminative modelling and often relies on approximations to surmount analytically intractable operations. Nonetheless, generative models have also many advantages over discriminative models, with the most interesting one being perhaps the ability to do representation learning. Irrespective of the task at hand, a common goal of most scientific pursuits is to identify "disentangled, semantically meaningful, statistically independent and causal factors of

variation in data" [52]. This is precisely what representation learning seeks, motivated by the fact that the performance of machine learning algorithms is powerfully reliant on the choice of features (representations of data) they are applied on [53].
Strictly speaking, representation learning methods are not explicit SSL techniques. However, in situations when labels are scarce and dimensionality of data high, these methods offer a powerful way to learn compact descriptions of the data from unlabelled instances that can then be transferred onto downstream tasks.

The most common models employed for the purpose of representation learning are Variational Autoencoders (VAEs), a class of Deep Neural Networks under constant innovation and refinement in the ML community due to their versatility and performance [52].

Variational Autoencoders [3]

The framework of VAEs (first appeared in [55, 56]) offers a principled approach for learning deep latent-variable models and accompanying inference models simultaneously with stochastic gradient descent.
A deep latent variable model is one where the marginal distribution of the data $\mathbf{x}$ is modelled by deep network parameters $\boldsymbol{\theta}$ through auxiliary, latent variables $\mathbf{z}$, introduced to explain the data generation process, such that:

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \int p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) d\mathbf{z}. \tag{2.2}$$

Commonly, deep latent variable models have the following factorisation

$$p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) = p_{\boldsymbol{\theta}}(\mathbf{z}) p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}), \tag{2.3}$$

which allows specifying a prior distribution over $\mathbf{z}$, our auxiliary variable, influencing the learnt representations. The problem with deep latent variable models which VAEs set to solve is that the integral in 2.2 does not have an analytical solution nor an efficient estimator [52]. Note that this directly implies that the posterior $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) = \frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{x})}$, which expresses the learnt representations, is also intractable.

To mitigate these intractability issues, the VAE framework introduces an inference model $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ which is also a neural network, called an *encoder*, specified by *variational parameters* $\boldsymbol{\phi}$ such that it is an approximation to the intractable posterior $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$. To see how this works and is achieved in practice, let us derive the objective function for the VAE.
The parameters of the inference network are found by minimising the difference between the approximate distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ and $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$. This is represented by the KL divergence

$$D_{KL}\left(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \| p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})\right) = \int q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \log \frac{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})} d\mathbf{z}, \tag{2.4}$$

which, if we isolate the marginal likelihood term, gives:

$$\begin{aligned}
D_{KL}\left(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \| p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})\right) &= \int q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \log \frac{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})} d\mathbf{z} \\
&= \int q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}) q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})} d\mathbf{z} \\
&= \int q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \log \frac{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})} d\mathbf{z} + \int q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \log p_{\boldsymbol{\theta}}(\mathbf{x}) d\mathbf{z} \\
&= \underbrace{-\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\right]}_{\text{ELBO, } \mathcal{L}(\mathbf{x};\phi,\theta)} + \log p_{\boldsymbol{\theta}}(\mathbf{x}).
\end{aligned} \tag{2.5}$$

Re-arranging 2.5 gives the central relationship:

$$\log p_{\boldsymbol{\theta}}\left(\mathbf{x}\right) = \mathcal{L}\left(\mathbf{x}; \phi, \theta\right) + D_{KL}\left(q_{\boldsymbol{\phi}}\left(\mathbf{z}|\mathbf{x}\right) \| p_{\boldsymbol{\theta}}\left(\mathbf{z}|\mathbf{x}\right)\right), \tag{2.6}$$

[3] Most of the derivations in this section are adapted from [54] and are expressed for single data points even where subscripts are omitted for cluttering purposes. For most models, the relation between "per-datapoint" and "per-dataset" terms is rather direct.

where the Evidence Lower Bound (ELBO, sometimes also referred to as Variational Free Energy) is defined as:

$$\mathcal{L}\left(\mathbf{x};\phi,\theta\right) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_{\boldsymbol{\theta}}\left(\mathbf{x},\mathbf{z}\right)}{q_{\boldsymbol{\phi}}\left(\mathbf{z}|\mathbf{x}\right)}\right]. \tag{2.7}$$

A crucial consideration is that due to the, by definition, non-negativity of the KL Divergence, the ELBO is a lower bound on the exact marginal likelihood of the data:

$$\begin{aligned}\mathcal{L}\left(\mathbf{x};\phi,\theta\right) &= \log p_{\boldsymbol{\theta}}(\mathbf{x}) - D_{KL}\left(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\|p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})\right) &(2.8)\\ &\leq \log p_{\boldsymbol{\theta}}(\mathbf{x}). &(2.9)\end{aligned}$$

Therefore, to find a maximum likelihood solution for $\boldsymbol{\theta}$, we can maximise the ELBO with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ as a surrogate for the log marginal likelihood.
Interestingly, this achieves two purposes at once [52]:

- Approximately maximises the marginal distribution, making the generative model $p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})$ better
- Minimises the KL term, therefore bringing our inference model $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ close to the true posterior $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$, improving the quality of the learnt representations.

The graphical model (see [57]) of the VAE framework is presented in Figure 2.6. Details of stochastic gradient-based optimisation of the ELBO will be omitted for brevity, but the interested reader is referred to [52] and the appendix of [55] for an exact derivation. What we will do instead is concisely consider a couple of extensions of the VAE framework of direct use to our work.

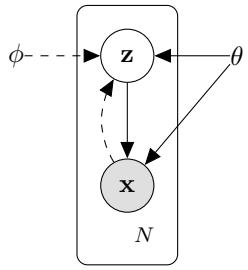


Figure 2.6: VAE graphical model. Solid lines represent the generative model $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{z})$, while dashed lines denote the inference model $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$. Reproduced from [55].

The β-VAE

Expanding the definition of the ELBO in 2.7, we can note it decomposes in two terms:

$$\mathcal{L}(\mathbf{x};\phi,\theta) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\right] = \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})\right]}_{\text{reconstruction term}} - \underbrace{D_{KL}\left(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})\right)}_{\text{regularization term}}. \tag{2.10}$$

The first term determines how well the reconstructions are going to be, whereas the second term indicates how far the decoder is from the prior on the latent variable, penalising taking data points far away from it. In the β-VAE framework an additional hyperparameter β is added to the regularisation term of the ELBO, to obtain the following modified objective:

$$\mathcal{L}(\mathbf{x},\beta;\phi,\theta) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})\right] - \beta D_{KL}\left(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\|p(\mathbf{z})\right). \tag{2.11}$$

The increased pressure on the posterior to match the factorised prior imposes more limits on the capacity of the latent bottleneck, as well as additional pressures for it to be factorised while still being able to reconstruct the data [58, 59]. In practical terms, this modified objective achieves more disentangled learnt representations. A disentangled representation is one in which single latent units are sensitive to changes in single generative factors while being generally

insensitive to changes in other factors [53, 58]. Having close to factorised representations is a very desirable property for representation learning systems, and of critical help when building downstream applications with the learned features. It must also be noted, however, that higher values of β used to encourage disentangling can cause a reduction in reconstruction quality as information needs to pass through a more constrained capacity latent bottleneck [59, 60].

To summarise, there always exists a value of $\beta > 1$ that provides greater disentanglement but produces a higher reconstruction error than a standard VAE [60]. This need not be an issue if we are not interested in the quality of reconstructions, as it will indeed not be the case in our application.

Semi-Supervised Variational Autoencoders

The second and last variation to the VAE framework we are going to consider is the semi-supervised variational autoencoder (SSVAE) introduced in [61]. The idea behind the SSVAE is to extend the standard VAE generative model to also account for a latent, discrete, class variable y in addition to the continuous auxiliary variable $\mathbf{z}$, to have:

$$p(y) = \text{Cat}(y \mid \boldsymbol{\pi}); \quad p(\mathbf{z}) = \mathcal{N}(\mathbf{z} \mid \mathbf{0}, \mathbf{I}); \quad p_{\boldsymbol{\theta}}(\mathbf{x} \mid y, \mathbf{z}) = f(\mathbf{x}; y, \mathbf{z}, \boldsymbol{\theta}).$$

This amounts to splitting the inference network (the encoder in the classical VAE framework, reported below as M1 for comparison) into two models (M2), comprising together a semi-supervised inference model for $\mathbf{z}$ and y, factorised as $q_\phi(\mathbf{z}, y \mid \mathbf{x}) = q_\phi(\mathbf{z} \mid \mathbf{x}) q_\phi(y \mid \mathbf{x})$. These are a latent-feature discriminative model for $\mathbf{z}$ (as in the standard VAE), and a latent-class discriminative model for y, capturing class-specific information [61]. In the case of a Gaussian latent space, we would therefore have:

$$\text{M1: } q_\phi(\mathbf{z} \mid \mathbf{x}) = \mathcal{N}\left(\mathbf{z} \mid \boldsymbol{\mu}_\phi(\mathbf{x}), \text{diag}\left(\boldsymbol{\sigma}^2_\phi(\mathbf{x})\right)\right); \tag{2.12}$$

$$\text{M2: } q_\phi(\mathbf{z} \mid y, \mathbf{x}) = \mathcal{N}\left(\mathbf{z} \mid \boldsymbol{\mu}_\phi(y, \mathbf{x}), \text{diag}\left(\boldsymbol{\sigma}^2_\phi(\mathbf{x})\right)\right); \quad q_\phi(y \mid \mathbf{x}) = \text{Cat}\left(y \mid \boldsymbol{\pi}_\phi(\mathbf{x})\right), \tag{2.13}$$

where $\boldsymbol{\sigma}_\phi(x), \boldsymbol{\mu}_\phi(x), \boldsymbol{\pi}_\phi(x)$ are respectively a mean vector, a vector of standard deviations and a probability vector, all represented as neural networks (commonly simple Multi-Layer Perceptrons, MLPs). Importantly, the network $\boldsymbol{\pi}_\phi(x)$ can be extracted and used to build a classifier after training.
M1 and M2 (i.e. the classical unsupervised VAE inference network and the semi-supervised inference model) can be combined for better outcomes, and result in a generative model with two layers of stochastic variables [61]:

$$p_\theta(\mathbf{x}, y, \mathbf{z}_1, \mathbf{z}_2) = p(y) p(\mathbf{z}_2) p_\theta(\mathbf{z}_1 | y, \mathbf{z}_2) p_\theta(\mathbf{x} | \mathbf{z}_1). \tag{2.14}$$

In this case, first M1 is trained ignoring label information to obtain the latent $\mathbf{z}_1$, then M2 can use these as data representations instead of the raw $\mathbf{x}$.

The complete graphical model for the SSVAE (M2) can be found in figure 2.7. The derivation of the semi-supervised ELBO objective is not trivial and omitted for brevity but can be found in [61].

2.5.2 Consistency training and pseudo-labelling

The second broad class of Deep Semi-Supervised methods we are going to consider rely on consistency training and pseudo-labelling.

As already hinted, consistency regularisation, or consistency training, uses unlabelled data to enforce the cluster assumption, i.e. pushes the model towards outputting similar predictions when fed an instance and its perturbed version [63, 64]. Concretely, given an unlabelled point $x \in \mathcal{D}_u$, a classifier $p_\theta(y|\cdot)$ and a weak transformation $\alpha(\cdot)$, the goal of consistency training is to minimise some form of distance between $p_\theta(y|x)$ and $p_\theta(y|\alpha(x))$. The most popular distance measure is the mean squared error, so that, with C classes, the consistency regularisation loss for a data-point would be:

$$d_{\text{MSE}}\left(p_\theta(y|x), p_\theta(y|\alpha(x))\right) = \frac{1}{C} \sum_{k=1}^{C} \left(p_\theta(y|x_k) - p_\theta(y|\alpha(x_k))\right)^2. \tag{2.15}$$

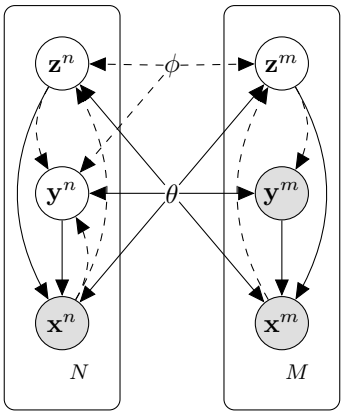


Figure 2.7: SSVAE graphical model for M labeled and N unlabeled data points. Dashed lines connect nodes and parameters of the recognition (or inference) model, solid lines of the generative model. Shaded nodes indicate observed variables, blank nodes latent variables. Adapted from [62].

Pseudo-labelling, on the other hand, has the objective of generating proxy labels to bootstrap the learning process [50, 65]. Specifically, pseudo-labelling only retains artificial labels whose largest class probability outputted by the model is above a predefined threshold τ, according to the loss function:

$$\mathbb{1}\left(\max\left(p_\theta(y|x)\right) \geq \tau\right) \mathrm{H}\left(\arg\max(p_\theta(y|\alpha(x))), p_\theta(y|x)\right), \tag{2.16}$$

where $\mathbb{1}$ is the indicator function and $\mathrm{H}(p, q)$ is the cross-entropy between two probability distributions p and q.

FixMatch

FixMatch [66] is a simple yet powerful SSL approach which combines the ideas of consistency regularisation and pseudo-labelling to outperform more complex deep SSL architectures [67–69]. A diagram of the algorithm is presented in figure 2.8. The algorithm uses two forms of data augmentations, weak augmentations $\alpha(\cdot)$ and strong augmentations $\mathcal{A}(\cdot)$. In image classification, the former are simple transformations such as random flips or shifts, and the latter are based on more complex transformations such as those provided by RandAugment [70]. In our application, these will be data augmentations specifically engineered to be biologically plausible. Precisely, given a batch B of labelled instances and μB unlabelled instances, FixMatch combines two cross-entropy terms:

$$\ell_s = \frac{1}{B}\sum_{b=1}^{B} \mathrm{H}\left(y_b, p_\theta\left(y \mid \alpha\left(x_b^l\right)\right)\right); \tag{2.17}$$

$$\ell_u = \frac{1}{\mu B}\sum_{b=1}^{\mu B} \mathbb{1}\left(\max\left(p_\theta(y|\alpha(x_b^u))\right) \geq \tau\right) \mathrm{H}\left(\arg\max(p_\theta(y|\alpha(x_b^u))), p_\theta\left(y \mid \mathcal{A}\left(x_b^u\right)\right)\right), \tag{2.18}$$

where we used superscripts to denote when a point is l labelled or u unlabelled, with τ being a scalar threshold hyperparameter above which the pseudo-label is retained [66]. The total loss minimised is then simply:

$$\ell = \ell_s + \lambda_u \ell_u, \tag{2.19}$$

with λ_u a scalar hyperparameter denoting the weight of the unlabelled loss.

Putting everything together, FixMatch first obtains an artificial label by computing the class distribution from a weakly-transformed sample. Then, it uses the `argmax` of that as a pseudo-label, enforcing through a second cross-entropy term that this will be the same as the model's output for a strongly-augmented version of that same sample. Despite its seeming simplicity, FixMatch obtains state-of-the-art results in most SSL benchmarks [66] and is compatible with modern Bayesian techniques for improved uncertainty quantification [71, 72].

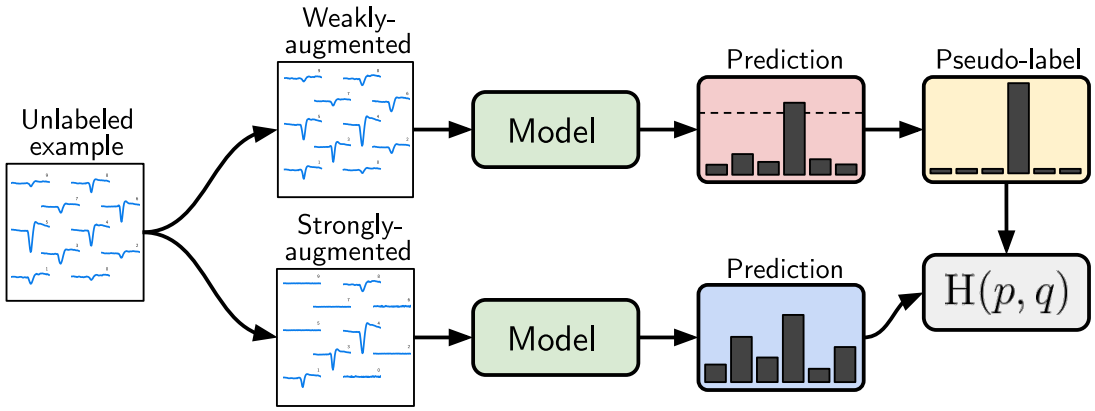


Figure 2.8: FixMatch diagram. The model is fed a weakly enhanced waveform or ACG (top) to provide predictions (red box). The prediction is changed into a one-hot pseudo-label when the model gives a probability to any class that is higher than a threshold (dotted line). The model's forecast for a strong augmentation of the same waveform or ACG is then calculated (bottom). A cross-entropy loss is used to train the model so that its prediction on the strongly-augmented version matches the pseudo-label. Details on the custom augmentations will be in section 4.3. Adapted from [66].

Chapter 3

Related work

3.1 Cell type classification

Classifying neurons into different functional, morphological and physiological types has been a key pursuit in Neuroscience since the dawn of the field [73], given its importance in determining the function of neural circuits, their development, evolution and role in disease [8]. Notable examples include work in the retina and cerebral cortex, where efforts in determining cell identities with a variety of techniques have been and will continue to be central to the success in elucidating the computations performed by neurons in these areas [8, 74–76].

Great efforts within electrophysiology, and recently naturally also within the sub-field of high-density recordings [16], have been devoted to discerning neuronal identities in extracellular recordings ([77, 78] for some early examples). Usually, this does not have to do with specific cell types, but with the distinction of different populations according to some clustering technique [79]. It is in fact very common for researchers using extracellular probes to be interested in the activity of two or more different populations known to be in the anatomical area of the recording [80, 81].
However, we are interested in specifically recovering the identity of the neuron being recorded, a vastly more problematic task with some precedents in the cerebellum, albeit mostly with the use of intracellular recording techniques.

3.1.1 Past work in the cerebellum

With some early attempts [82], work towards cell type classification in the cerebellum has seen rising interest in the last 10 years. Nonetheless, for a variety of reasons which we will now assess, past work will not be directly relevant to many aspects of our exploration.

Ruigrok and colleagues [11] present a first comprehensive attempt at classifying interneurons in the cerebellar cortex using juxtacellularly labelled [83] ground-truth units. Follow-up work by the same group [13] included also Purkinje cells in the classification process. However, sadly, both studies have substantial flaws in their machine learning component. They use a high-variance model (i.e. a decision tree), but do not take any steps to make sure they are not overfitting to the training set. They do not tune hyperparameters, do not perform cross-validation and do not test on a held-out dataset. Unsurprisingly, their results failed to be reproduced by other investigators [12, 84].

Van Dijck et al. [12] trained a Gaussian Process Classifier to identify all major cell types in the cerebellar cortex (PkC, GoC, MFB, MLI and GrC) using only temporal features of the neurons (see next section), with ground-truth labels still obtained via juxtacellular labelling [83]. They also reported remarkably high accuracies, yet they properly validate their results with cross-validation strategies and test sets from independent laboratories.
One drawback of their approach is that it uses different models based on the cerebellar layer in which the neuron is determined to be (by relying on the identification of Purkinje cells complex spikes), making it not trivially generalisable to multi-channel recordings which would frequently

record from more than one *folium* of the cerebellar cortex at the time.

Haar and colleagues [84] did not try to classify cerebellar neurons directly, but acquired ground-truth data (still through juxtacellular labelling methods [83]) to determine if morphologically identified cell types could be clustered through unsupervised methods. Their conclusion is negative, but a number of flaws can be identified in their approach which detracts from their results. First, it is unclear why they decided to cluster distances between inter-spike interval distributions, and did not provide a baseline with commonly used features in the literature. Secondly, they do not take into consideration the shape of the extracellular action potentials, which are now widely acknowledged to contain critical information for cell type classification [16, 81]. Thirdly, they acknowledge explicitly that the measures they use are not *metrics* in the mathematical sense but then do not demonstrate satisfactorily that they corrected for this in subsequent analyses, nor offered a comparison with true distance metrics. Further, they make a strong linearity assumption in the type of methods used and do not explore any nonlinear techniques. Finally, they do not try to fit a classifier to their ground-truth data and base their conclusions only on inspection of clustering results.

To conclude, let us note two major reasons why all past studies just surveyed are of little direct use when tackling our problem.

First, in all instances, recordings were acquired from anaesthetised animals. Given that the cerebellum is an area highly involved in motor control, the activity over time of cerebellar neurons under anaesthesia is known to be very different from that in awake, freely behaving animals [85]. Additionally, on the more practical side, if an experimenter is interested in determining cell types in a behavioural task, it will be completely unrealistic to first put the mouse under anaesthesia, let it recover, and then do the task, just to be able to accurately determine the identities of the neurons recorded.

Finally, most laboratories in systems neuroscience now use modern electrode arrays which yield neuronal waveforms distributed over several recording channels. Previous studies all used single electrodes, which yield 1-dimensional waveforms. However, the spatial footprint of neurons likely contains relevant information to determine their cell type [16, 80, 81]. For example, the amplitude crossed with the spatial decay of a waveform reflects the size of the neuron it originates from. For this reason, we took particular care to extract information from the full spatio-temporal extent of neuronal waveforms in our representation learning experiments.

Ultimately, what we are after is an algorithm that can work on data from freely moving mice, with cell identities computed during spontaneous activity periods, and recorded on high-density electrodes. A classifier with outstanding performance on computed metrics but no practical use is not a good classifier at all.

3.2 Feature Engineering for neuron classification

In spite of the questionable direct applicability of past models to our problem, years of research attempting the classification and clustering of cortical neurons from both extracellular and intracellular recordings have converged on a number of high-quality engineered features to describe neural data. These can be broadly divided into two categories.

Temporal features describe the firing behaviour of the neuron over time and are usually calculated from the inter-spike interval histogram (Figure 2.5). Temporal features have proven to be the most effective in neuron classification in a variety of studies [12, 13] and have the advantage of being blind to the type of probe being used, as they are calculated after the spike sorting process, a universal step in all extracellular recordings. The most complete formulation of the temporal features to be used in neuronal classification is used and summarised by [12], and will also be adopted in our models.

Waveform features describe the shape of the recorded action potential signal. The exact form of the extracellular waveform is closely tied to the type of electrode used, representing the principal obstacle to the generalisation of models that use waveform features. Despite this, waveform features usually add to the capacity of models so there have been valuable attempts

at parameterising waveform characteristics in the literature, mostly on single-channel traces. In recent years, with the advent of high-density, multi-site recordings, the formulation of features that take into account the spatial footprint of the waveform across probe channels has become pressing, with some novel studies trying to address the issue [16, 80, 81], which we will take as the baseline in our modelling.

What seems to be lacking altogether from the literature on neuronal cell type classification is a move past engineered features or an attempt at guiding the feature engineering process through machine learning methods.

3.3 Past efforts in the Häusser lab

The cell types classification project has been an ongoing endeavour in the Häusser lab for around 3 years. During this time, the data collection, pre-processing and curation pipeline has been improved through constant monitoring of experimental results. Compared to the solidity of the experimental paradigm for data collection and curation, machine learning attempts at classifying cerebellar cell types in the lab are more recent.

Previous work in the Hausser lab established that engineered features for neuron classification routinely used in the literature can prove valuable when classifying cerebellar cell types from Neuropixels recordings. However, it was concluded that waveform-based features were not adding value to the classification process.

At the beginning of the current investigation, different flaws were found specifically in the Machine Learning pipeline for cell-type classification:

- The feature engineering process to extract information from multi-dimensional waveforms was not robust to all the possible configurations in which an action potential can be observed in extracellular recordings. This caused the feature extraction process to fail in some instances.
- Test-set leakage was discovered in the trained models, inflating reported performances. Specifically, oversampling techniques such as SMOTE [86] were applied before splitting the dataset.
- Reported results were relative to cross-validation performance, given that data was too limited to create a meaningful test set, but different runs were not satisfactorily averaged to avoid overestimating accuracies.
- Unlabelled data was not considered a potential asset to the classification task, despite its abundance.
- Machine Learning models going beyond engineered features have also not been regarded as potential resources to tackle the problem.

These points will all be addressed this dissertation.

Chapter 4

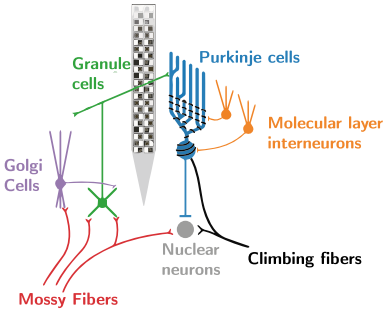
Methods and Experiments

4.1 The data

The data acquisition paradigm for this project has been consolidated in the Häusser lab in the past three years. It involves exquisite experimental manipulations, the details of which would deserve a treatise of their own and are thus well beyond the scope of this dissertation. Accordingly, our discussion will be limited to those aspects of the data that are essential in understanding the problem and its attempted solutions.

4.1.1 Optotagging

Ground-truth labels for the cell types of interest are obtained through optotagging (first termed, rather verbosely, Photostimulation-assisted Identification of Neuronal Populations in [87]). The procedure involves expressing the ion channel channelrhodopsin-2 (ChR2) in specific neuronal subpopulations. Channelrhodopsins are nonspecific cation channels that depolarize when exposed to blue light. These light-gated ion channels were identified from Chlamydomona green microalgae [88], where they control photo-taxis behaviour. A brief flash of blue light causes a reliable, short latency action potential in ChR2-tagged neurons, making them identifiable electrophysiologically in vivo. Getting reliable and specific genetic expression of ChR2 only in certain neuronal subpopulations is a nontrivial task, and involves careful engineering of transgenic mouse lines. The details and references for the expression used for our dataset are summarised in figure 4.1, and benefit from active research in optogenetics [89], a framework now at the core of state-of-the-art neuroscience.



Cell type	Mouse line	Reference for cerebellar expression
Purkinje cells	Pcp2-Cre (Jdhu) x Ai32 (Cre-dep ChR2)	Witter & Regehr (2016) [90]
MLIs	Nos1-Cre x Ai32	Jelitai & Duguid (2016) [91]
Granule cells	Math1-Cre x Ai32	Wagner & Luo (2017) [30]
Golgi cells	Glyt2-Cre x Ai32	Gurnani & Silver (2021) [92]
Mossy fibers	Thy1-ChR2	Hull & Regehr (2012) [93]

Figure 4.1: Details of the mouse lines used in optogenetics manipulations.

4.1.2 Experimental protocol

To ensure that optotagging works as expected, it is not sufficient to deliver light stimulation and include as ground truths all the neurons that respond within a short window of the stimulation. Due to fast monosynaptic and polysynaptic connections highly present in the cerebellar cortex, optical responses need to be compared before and after application of synaptic blockers that target all the major ionotropic receptors. This way, if fast responses persist after drug

application, they can only be caused by the light. One exception to this, however, is *off-target* expression of ChR2 (i.e. in undesired neuronal cell types). To quantify and adjust for it, histological analyses of the neural tissue around the site of recording are performed *post-hoc*, guided by the fact that all electrodes are coated with DiI (a common fluorescent dye for cell membranes).

Details of the experimental paradigm for data acquisition are summarised in figure 4.2. Note how the presence of a period of spontaneous activity, used to compute the temporal and waveform features, is an integral part of the protocol.

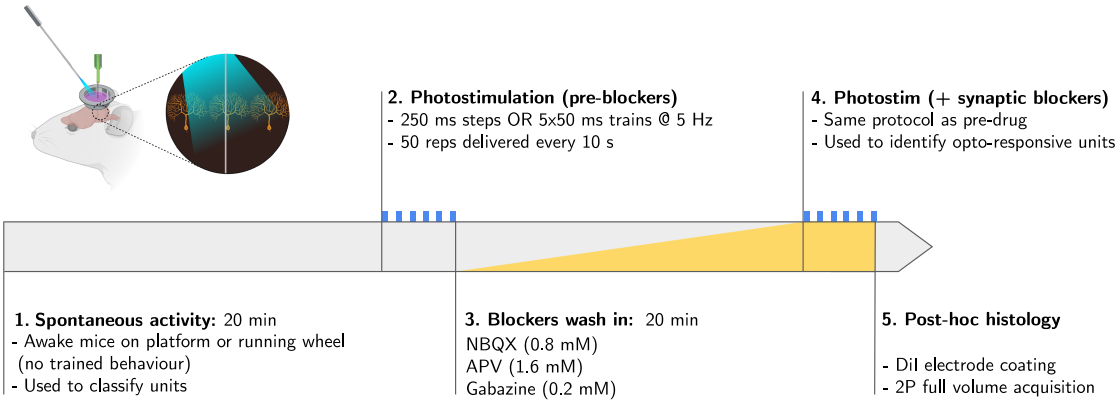


Figure 4.2: Schematic of the optotagging experimental protocol for data acquisition. Adapted from internal presentation.

4.1.3 Data curation and pre-processing

After data acquisition, extensive curation and pre-processing steps go into the creation of the final dataset.

First, spike sorting is run with kilosort [47], followed by manual curation of the sorting output through Phy [46]. At this point the responses to the light of candidate units [1] pre- and post-drugs are assessed, and if there is any evidence of drug efficacy at that depth in the recording (seen as the suppression of synaptic connections observed pre-drug), the unit under examination is called responsive, and included as a candidate neuron in the dataset.

Secondly, all candidate units undergo a series of pre-processing steps to accentuate their salient features. Specifically, spike-sorted waveforms are averaged across high-amplitude portions of the recording and de-noised, to yield a single, high-quality waveform for that neuron. In the process, corrections for averaging artefacts such as drift-matching and shift-matching are applied through the open-source package `npyx` [23].

Finally, before inclusion in the dataset, candidate units undergo a series of quality checks that ensure the spikes found through spike-sorting and manual curation are within certain false-positive and false-negative thresholds, where false positives are clustering errors (i.e. spikes of multiple units clustered into one) and false negatives are missed spikes (as seen by a clipping in the distribution of recorded spike amplitudes over time). As a consequence, the remaining units all consist of spike trains extensively checked for correct attribution to a single neuron, and average waveforms obtained through specialised de-noising mechanisms.

4.1.4 The final cerebellum dataset

The cerebellum dataset for cell types classification, obtained with the methods summarised above, consists of 77 labelled ground-truth neurons (divided into 25 `GoC`, 21 `PkC_ss`, 11 `PkC_cs`, 9 `MFB`, 6 `GrC` and 5 `MLI`), along with 877 high-quality unlabelled units.

[1] As a reminder, clusters of neural activity found after the spike sorting process are called units. Ideally, they coincide with neurons, and we are in fact often calling them neurons when referring to units in the final dataset which underwent several stages of quality checks.

Though this may seem like a remarkably small dataset, the reader is encouraged to consider the enormous amount of work that went into gathering this data, which is the fundamental motivator of our need for efficient SSL algorithms: the dataset is extracted from ≈ 150 experiments run on ≈ 140 mice over the course of 3 years by more than 4 different experimentalists. Even if often the animals used in these experiments were also taking part in other investigations, the sheer magnitude of these figures is impressive.

Each neuron in the dataset is represented by a tuple $\left(\mathbf{x}_i^{spk\_t}, \mathbf{X}_i^{wvf}\right)$ consisting of a variable-length array of spike times $\mathbf{x}^{spk\_t}$, indicating the time at which a spike occurred in the 20 min period of spontaneous activity, and a 10×60 $(n\_channels \times samples)$ matrix $\mathbf{X}^{wvf}$ of waveforms across probe channels in space.

For some models, however, each datapoint is represented by a different tuple, $\left(\mathbf{x}_i^{ACG}, \mathbf{X}_i^{wvf}\right)$, where $\mathbf{X}^{wvf}$ still represents the waveform i in space, and $\mathbf{x}^{ACG}$ is a 100-dimensional vector representing the autocorrelogram (see 2.3 and 2.5) for neuron i, calculated with a bin size of 1 ms and a window size of 200 ms.

A visualisation of some labelled examples in the dataset is presented in figure 4.3.

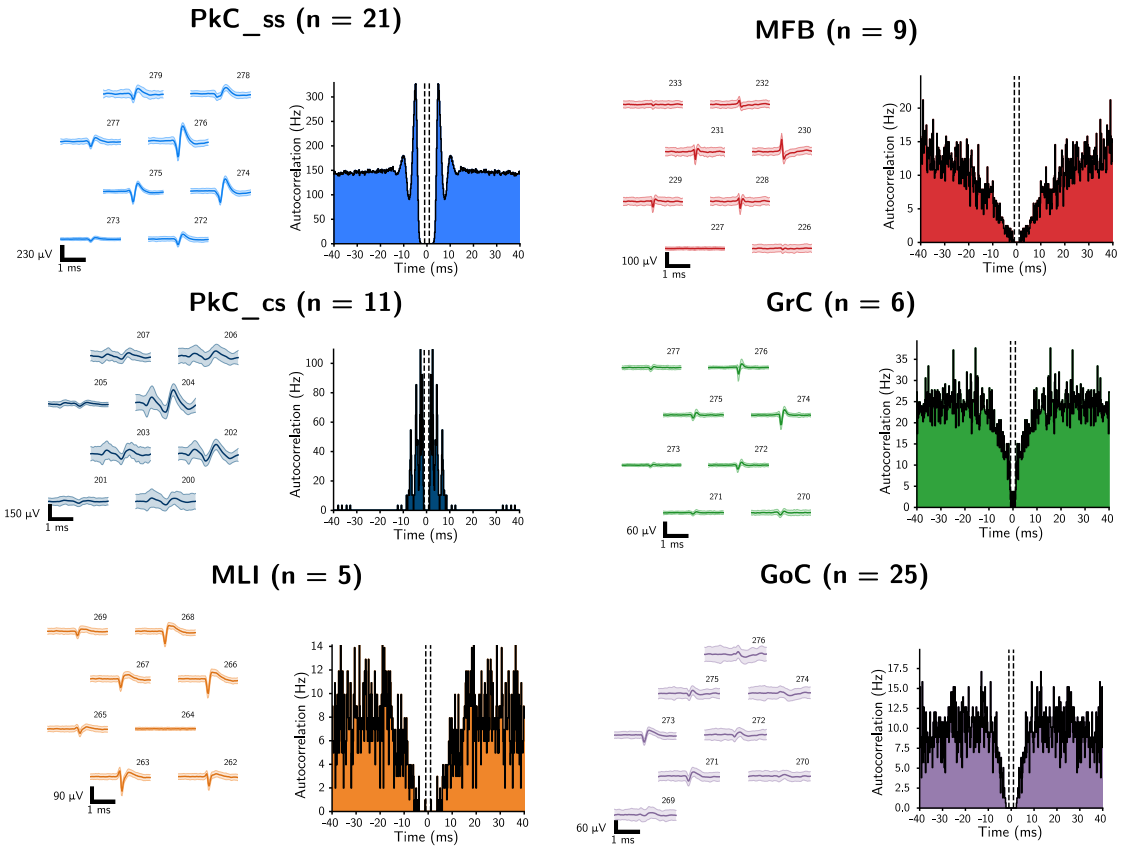


Figure 4.3: Sample waveforms and autocorrelograms of one neuron for each cell type in the dataset. For compactness of representation, only 8 out of the 10 waveforms in $\mathbf{X}^{wvf}$ are plotted.

4.1.5 Dealing with low and imbalanced data

The cerebellum dataset is rather different from the usual machine learning benchmark dataset. The overall number of instances in the dataset is not strikingly high (though it is in a Neuroscience setting), and the number of labelled data is even more problematic. Moreover, there is a great deal of class imbalance in the few ground-truth labels that are present, a situation particularly common in Neuroscience [94]. In dealing with such a troublesome dataset, a number of noteworthy steps were taken to ensure the data was used to its full potential in every model.

1. Stratified cross-validation was always used in hyperparameter tuning unless otherwise specified.

2. Random oversampling of the under-represented classes was performed for every model after cross-validation splits, using the open-source package `imblearn`[2].

3. Given the fundamental lack of labelled data we are facing in this project, a proper test set could not be created to quantify generalisation in a customary way. As a result, the performance of all models proposed is always expressed in terms of leave-one-out cross-validation (LOOCV). While we acknowledge this may lead to overly confident performance estimates, it must also be noted that the LOOCV is a nearly unbiased estimator of the generalisation error [95], with variance comparable to other cross-validation methods [96], and has been used in low-data settings similar as ours [12].

4. To overcome potential issues with the variance of LOOCV and avoid reporting cherry-picked performances due to chance, results are always reported by averaging random runs of LOOCV with different random seeds.

4.2 Baseline models

As detailed in section 3.1.1 and 3.2, past work in cerebellar cell types classification is not directly applicable to our problem, with the exception of the feature engineering process. However, there are no priors on the importance of each feature that can be derived from previous efforts, given the different recording conditions and the probe types used.

As a consequence, a necessary part of our investigation was the development and reproduction of a few baseline models previously adopted in the Häusser lab.

4.2.1 Human experts

The only significant prior that could be used to inform our pursuit comes from the expert opinion of experienced electrophysiologists. However, should we use this type of information? Most electrophysiologists gain experience and insight into the typical behaviour of different cell types by working with intracellular recording methods. However, as we already discussed, extracellular recordings with high-density probes such as Neuropixels generate data that is far more complex and hard to interpret without the aid of software tools and extensive pre-processing. It can thus be argued that, given the novelty of the task and the technology adopted, expert opinion may bias the discovery process by enforcing priors which have not been proven to directly transfer across recording conditions.

Nevertheless, gathering expert opinions can be insightful in establishing a baseline for Machine Learning models to improve upon and also measure the practical significance of our efforts. For these reasons, we developed a web application [3] that allows expert electrophysiologists to predict the labels of ground truth cell types in our dataset. This survey-like application was distributed to three electrophysiologists in the lab who ranged from 4 to 8 years of experience working with both intracellular and extracellular recording methods. Their average performance is reported in table 4.1, along with the performance of a weighted majority classifier constructed by weighting their opinions according to individual accuracies.

4.2.2 Feature extraction

A natural baseline for many machine learning tasks is to construct a simple model with engineered features. Fortunately, the literature on cerebellar classification [12] and cell type clustering from Neuropixels recordings [16, 80] offers an established set of both temporal and waveform features to extract from extracellular recordings.

Some words of caution must be voiced on the process at this point. While temporal features are entirely dependent on the events found by the spike sorting process and are in theory robust to the type of probe being used, waveform features need to be treated with special care. Since the shape of the waveform and its spread across channels can vary due to a large number of factors (including probe type, distance to the probe, morphology of the cell and more. Refer back to figure 2.2), there are some practical decisions to be made when extracting waveform

[2] `https://imbalanced-learn.org/stable/`

[3] Which is accessible here.

features from extracellular recordings. For example, one could decide to only extract features from waveforms on the peak channel, irrespective of their shape. Or, alternatively, to only extract features from the peak channel when they have a certain, common shape (i.e. a somatic spike) which is well amenable to feature calculation. Or, again, one may decide to fit a dipole model to the waveform in space and extract features from the modelled waveform, limiting the variation across waveforms.

However, all of the methods mentioned have a key limitation: they have failure cases. If we always extract from the peak channel, we will inevitably find different types of spikes on it (i.e. somatic, dendritic or axonal), introducing unwanted variation. If we always extract only somatic spikes, there may be types of neurons for which they are not recorded, yielding unusable neurons under this method. Likewise, if we want to fit a dipole model, it is necessary that both the source component and sink component of the waveform are recorded for the same unit across channels, which is often not the case.

The approach taken before the beginning of this study in the Häusser lab was particularly lossy as only somatic features from the peak channel were extracted. We addressed this by modifying the pipeline for waveform features extraction, which now proceeds in different stages. First, peak detection is run to determine if there are any usable somatic spikes in the trace, whether they are on the peak channel or not. If any are found, features are calculated from the highest amplitude one, if not, the highest amplitude non-somatic waveform is flipped in sign and used to extract the features. This process ensures that the waveform features calculation step does not fail for any given $\mathbf{X}^{wvf}$. We call the channel found through this process the *relevant channel*, and extract features from it.

Specifically, the following 15 waveform features were extracted from the relevant channel: peak time, peak voltage, trough time, trough voltage, repolarisation time, depolarisation time, half peak width, half trough width, onset time, onset amplitude, waveform width, peak-to-trough ratio, recovery slope, repolarisation slope and depolarisation slope.

Additionally, two features were calculated using information across channels: the spatial decay of the waveform at 24μm, and the amplitude of the dendritic component.

Furthermore, 11 temporal features were calculated on the ISI histogram (see figure 2.5): median, mode, entropy, 5th percentile, average and median CV2, CV, local variation, revised local variation, rescaled cross-correlation and skewness; together with 4 features calculated on the ISI themselves (mean and mean instantaneous firing rate, instantaneous irregularity and refractory period duration), this provides a total of 15 temporal features.
Our contribution to the feature extraction process have been included in the open-source electrophysiology package `npyx` [23].

To obtain a visual summary of the dataset after feature extraction, an interactive dashboard built in dash[4] was developed. The tool allows us to summarise, compare and inspect all the information in the dataset: the different types of features, both raw and normalised for each feature type, and detailed plots of the optotagging process that resulted in the inclusion of each neuron in the dataset. The application is openly accessible at the following website: `https://files.fededagos.me/features/`.

4.2.3 Random Forest

Using the reviewed feature engineering process, a simple Random Forest classifier (RF)[5] was made to serve as a baseline model. To be competitive with successive models, the hyperparameters of the RF were tuned via Bayesian Optimisation [24] through the open-source package `optuna` [25], with the average stratified 5-fold cross-validation F1-score as the objective. Specifically, the following parameters were tuned: {`n_estimators, criterion, max_features, min_samples_leaves`}.

Results of 50 different runs, each with different random seeds, of LOOCV (leave-one-out cross-validation) on random forests using different subsets of features are reported in table 4.1

[4] `https://dash.plotly.com/`

[5] The popular implementation in `scikit-learn`[97] of the Random Forest classifier was used (`sklearn.ensemble.RandomForestClassifier`).

Model	Accuracy	F1-score	n features
Human Experts	57.3 ± 6.2	45.9 ± 6.5	N.A.
Weighted Majority Experts	57.3 ± 0.0	44.1 ± 0.0	N.A.
Engineered waveform features + RF	58.4 ± 5.9	44.1 ± 3.1	17
Engineered ACG features + RF	59.7 ± 1.2	51.5 ± 2.2	15
All engineered features + RF	71.4 ± 0.5 *	56.8 ± 3.5 *	32

Table 4.1: Baseline model performances. Values indicate means plus or minus standard deviations. Stars indicate the best performances among the ones reported. All results of the RF models are reported after 50 different runs of LOOCV.

Two things are immediate from these results. First, the human baseline achieves very poor and variable performance, which does not improve by taking a weighted majority vote of the responses. Second, the best baseline is the one that uses all available engineered features from the literature. This is in contrast with some previous results internal to the lab and demonstrates the added benefit of including waveform features with our novel, generalised, approach to computing them.

4.3 Data augmentation strategies

Central to the deep learning approaches tried in our further experiments was the development of custom data augmentation strategies for the cerebellum dataset. Given the extremely low data setting we are in, it was essential to come up with augmentations that would mimic the natural variability found in the data and extend the labelled data pool to regularise highly expressive models such as deep networks. Past work has repeatedly shown how, when plausible transformations are known, augmentations in data space are highly superior to synthetic oversampling in feature space [98, 99] in reducing overfitting and models' generalisation capabilities. This is true even in expert domains [100], and for a variety of Deep Learning architectures [101, 102].

In practice, we built a total of 8 custom data augmentations, 4 specific to the waveforms, 4 specific to the spike trains and 2 usable on both. These are, for the waveforms:

1. `SwapChannels`: swaps the indices of even and odd channels in $\mathbf{X}^{wvf}$, mimicking the biological scenario in which the probe was oriented in the same way along the longitudinal axis but in the opposite way along the dorsoventral axis.

2. `VerticalReflection`: reverses the indices of the channels of $\mathbf{X}^{wvf}$, mimicking the scenario in which the probe was oriented in the same way along the dorsoventral axis but in the opposite way along the longitudinal axis.

3. `DeleteChannels`: deletes `n_channels` at random from $\mathbf{X}^{wvf}$, simulating a corrupted recording from one or more channels.

4. `PermuteChannels`: permutes `n_channels` at random in the waveform. This is not a biological transformation and is only used for strong augmentations in FixMatch.

For the ACGs or spike trains:

1. `DeleteSpikes`: each spike in the train is deleted with probability `deletion_prob`. Simulates a scenario in which the neuron was firing more sparsely.

2. `MoveSpikes`: jitters each spike a quantity `max_shift`. Introduces plausible variability in the spike train recordings.

3. `AddSpikes`: adds a random number of spikes `max_addition` $\cdot$ `len(spike_train)`, simulating an increase in firing rate.

4. `NewWindowACG`: re-calculates the ACG from the spike train by multiplying both the window size and bin size by `magnitude_change`. Slightly changes the way the ACG vector is represented, analogously to cropping or re-scaling an image.

And the more generic:

1. `GaussianNoise`: Adds Gaussian noise to both $\mathbf{X}^{wvf}$ and $\mathbf{x}^{ACG}$, with independent standard deviations σ_{wvf} and σ_{ACG} both multiplied by a parameter `eps_multiplier`

2. `ConstantShift`: Randomly compresses or expands the signal by a given scalar amount `scalar` which multiplies both $\mathbf{X}^{wvf}$ and $\mathbf{x}^{ACG}$.

A visual summary of most custom transformations is presented in figure 4.4.

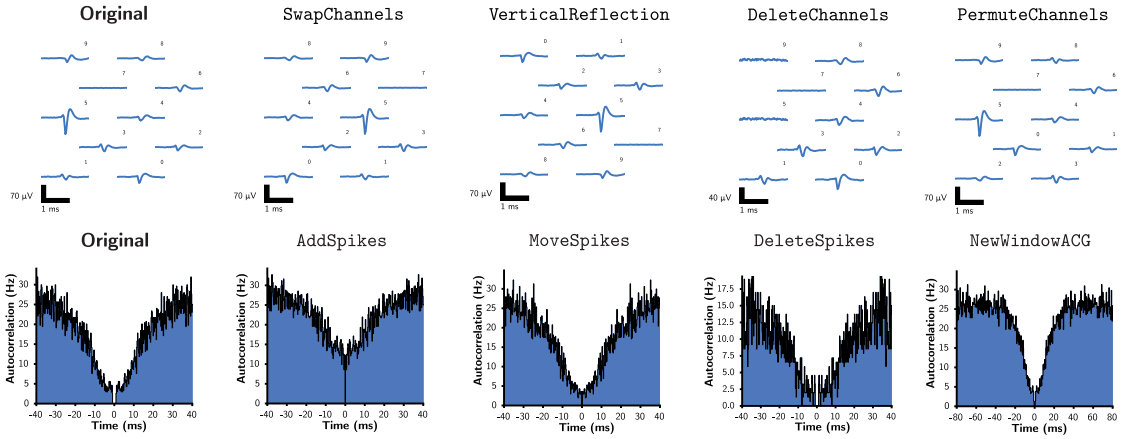


Figure 4.4: Examples of our custom data augmentations for the waveforms and autocorrelograms. `GaussianNoise` and `Constantshift` are not included here as they are more trivial and not specific to the cerebellum dataset.

Furthermore, a custom wrapper inspired by RandAugment [70] was built on top of the augmentations to be used with FixMatch [66], such that random combinations of augmentations of different magnitudes could be applied to any datapoint at training time.

4.4 Representation learning

The first major contribution of this work to the cell types classification problem comes through representation learning techniques. The cerebellum dataset contains 10× more unlabelled than labelled data points, with the potential of increasing the unlabelled pool even more, as it can be collected through much simpler experiments. Moreover, no existing methods in the literature extract features extensively from the waveforms in space, nor from the autocorrelogram (a much richer source of information than the ISI histogram). To resolve such issues, while still retaining the possibility of using engineered features familiar to electrophysiologists in modelling, we decided to train Variational Autoencoders (VAEs) on all data to extract a low-dimensional, compressed and mathematically close to optimal representation of the data (see section 2.5.1).

Specifically, two separate VAEs were trained on the unlabelled data, one to learn representations from $\mathbf{X}^{wvf}$, the waveform in space, and one to extract information from $\mathbf{x}^{ACG}$. In both cases, encoder and decoder networks were symmetrical Multi-Layer Perceptrons (MLPs), the architecture of which was chosen through Bayesian Optimisation. The optimisation objective was the 5-fold cross-validation accuracy of a Random Forest (RF) classifier trained using the latent space encoding as features, along with either temporal features (for the waveform VAE) or waveform features (for the ACG VAE). Almost all the hyperparameters of the MLP were chosen through Bayesian optimisation [24], and included: {`batch_size, n_layers, learning_rate, n_units_layer, dropout, optimizer, d_latent, beta`}. Note how we also hyper-optimised over the dimensionality of the latent space (`d_latent`), and over the coefficient `beta` determining the pressure put on disentanglement of the latent space.

Results of Bayesian Optimisation over the hyperparameters show how higher values of β (i.e. $\beta = 4.63$), corresponding to more disentangled representations, are able to drive better results on downstream models for the waveform VAE. In contrast, the best value for β for the ACG VAE was found to be 1, i.e. the traditional ELBO objective, suggesting how disentanglement may help on a case-by-case basis. Additionally, more latent dimensions were found to be optimal

in compressing the data for the ACG VAE (14) compared to the waveform VAE (10). Further details about the models' architectures are available in Appendix A.

To ensure our trained VAEs were indeed capturing variance as expected, we investigated the structure of the latent spaces by transforming linearly spaced coordinates on the unit square through the inverse cumulative density function of a Gaussian (given that the prior on our latent space was chosen to be Gaussian) to produce a range of values for the latent variable $\mathbf{z}$ (following [55]). However, given that our latent spaces are more than two-dimensional, we actually plot a 2D cross-section of them by transforming those points with the matrix constructed with the first two right singular vectors of the mean latent spaces (which are found by passing all the data through the encoder networks).

The results displayed in figure 4.5 clearly show that our VAEs are working as expected. For example, we can note how the vast majority of the latent space for the waveform VAE is occupied by variations of somatic waveforms (top right of 4.5A), which are indeed the most common in the dataset. However, all other typical spike shapes are also represented, including dendritic waveforms (bottom-left of 4.5A), and both bi-phasic and tri-phasic axonal waveforms (along the diagonal of 4.5A). The same is true for the ACG latent space, which captures anything from bursting activity (top-left 4.5B) to high refractory periods (middle-right 4.5B), oscillations (middle-bottom 4.5B), and both high and low firing rates.

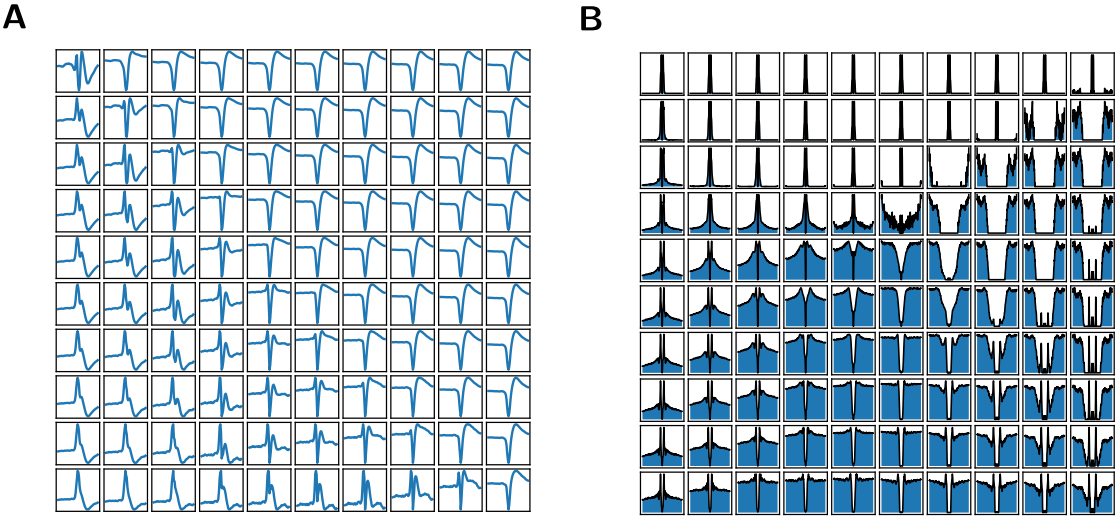


Figure 4.5: Latent space interpolations from our waveform and ACG VAEs trained on the cerebellum dataset. **A.** 2D projection of the 10D latent space learnt by our β-VAE trained by reconstructing the waveforms across channels $\mathbf{X}^{wvf}$. For the sake of visualisation, only the waveform on the peak channel is plotted here. **B.** 2D projection of the 14D latent space learnt by our standard VAE trained on ACGs $\mathbf{x}^{ACG}$.

Having established the expressiveness of the learnt representations, we use the encoder networks from the trained VAEs to compute features to employ in random forest classifiers, which we evaluate again through LOOCV. Results are presented in table 4.2, and clearly demonstrate the positive impact on performance of the representation learning approach. When compared with the baseline, VAE-driven models reach comparable or better levels of accuracy and overall better F1 scores with a reduced number of features. Reducing the number of features used while retaining or improving model performance is especially important in low data regimes where we want to avoid over-specifying the classification problem to help with generalisation, as it is easier to overfit to idiosyncrasies with fewer instances [103].

Notably, VAE waveform features are the ones making the higher impact on the problem, proving how feature engineering approaches are still not successful at extracting all available information from the waveform (in particular its spatial footprint). On the other hand, somewhat surprisingly, the learned ACG features are only a small improvement over the engineered temporal features, corroborating a general consensus in the literature on the robustness of temporal features extracted from the ISI histogram.

Model	Accuracy	F1-score	n features
All engineered features + RF (best baseline)	71.4 ± 0.5	56.8 ± 3.5	32
VAE wvf + engineered temporal features + RF	71.1 ± 0.9	63.8 ± 3.5	25
VAE ACG + engineered waveform features + RF	72.7 ± 0.6*	58.2 ± 2.6	31
VAE wvf + VAE ACG + RF	72.7 ± 0.9*	64.8 ± 3.4*	24

Table 4.2: Performance of Random Forest classifiers trained with different combinations of features, either learned through VAEs or engineered.

On the whole, the representation learning approach resulted fruitful, and, as expected, conveniently complementary to feature engineering, while also diminishing the imbalance present in the models towards the majority classes (as demonstrated through the F1 scores).

While learning representations has the advantage of granting compatibility with any downstream model or other sources of features, it is also possibly not leveraging at its fullest the potential of contemporary semi-supervised methods and is still tied to non-deep models to provide the predictions.

Inspired by the fact that the best-performing model so far is using only features coming from Variational Autoencoders, we expand our exploration first towards explicitly semi-supervised extensions of the VAE framework, and secondly to more general deep semi-supervised methods from the literature [66].

4.5 Deep semi-supervised learning

4.5.1 The semi-supervised Variational Autoencoder

VAEs, as we have seen, are excellent tools to learn rich representations of data in a completely unsupervised manner. However, certain features of the data often covary in a specific way with different classes. Ideally, we would want to provide some partial class information to transform the feature space, so that it can learn to separate inter-class variance from intra-category variabilities and model group membership in a more complete and robust way. This is expressly the objective of the semi-supervised VAE (SSVAE, [61]; see section 2.5.1).

Our SSVAE (model M2 in [61], see 2.5.1) was trained using only 4 labels per class, in order to be able to evaluate it through leave-one-out cross-validation (as only 5 labels were available for the MLIs). To partly compensate for this, all our custom data augmentations were used during training.

To encode and classify neurons with the same architecture, we needed to unify the latent space for waveforms and autocorrelograms. The simplest way to achieve this was to change the representation of our data from a tuple $\left(\mathbf{x}_i^{ACG}, \mathbf{X}_i^{wvf}\right)$ to a single vector, concatenating the two sources of information.

Once more, the specific architecture of the SSVAE was discovered through Bayesian Optimisation [24], using the accuracy on a held-out validation set as the objective. The range of hyperparameters tuned this way included: {`learning_rate, n_layers_autoencoder, n_layers_classifier, batch_size, non_linearity, n_units, optimizer, d_latent`}. Again, the quality of reconstructions for the discovered architecture was inspected to ensure we did not incur in posterior collapse [104], especially since we changed our representation of the data and used a modified version of the ELBO objective [61].

Following hyperparameter optimisation, the final architecture was tested on 11 runs of leave-one-out cross-validation using different random seeds that changed the labels used for each class during training and the initialisation of all networks. Practically, the procedure included first training the SSVAE on all unlabelled data and 24 labelled instances (4 per class) drawn at random from all but one the labelled data points. Then, after training, the classifier network of the SSVAE was used to predict the held-out labelled datapoint. Table 4.3 shows the results of this procedure compared with the performance of previous models.

First of all, it should be stressed how, with only a fraction of the labels of other models, the SSVAE is able to attain comparable performance to the previous best models, if not better when compared to the baseline.

Model	Accuracy	F1-score	n features	n labels
All engineered features + RF	71.4 ± 0.5	56.8 ± 3.5	32	77
VAE wvf + VAE ACG + RF	72.7 ± 0.9	64.8 ± 3.4	24	77
Semi-Supervised VAE	69.3 ± 9.7	65.4 ± 7.8	700	24

Table 4.3: Comparison of the SSVAE performance with previous best performing models. Note the difference in the number of labels used

Secondly, however, we need to acknowledge the greater variance exhibited by the SSVAE between LOOCV trials, which likely comes from multiple sources. The first and most important source of variance is the subset of labels used to train each model. Given that we randomly choose the 4 labelled instances for each class to use in different runs, how representative of its class each example is, and the variance it represents, greatly influences the performance of the model, as has been demonstrated with other SSL methods [66]. The second probable major source of variance is the random initialisation of the different networks. Given known issues with the stability of the ELBO objective during training [105, 106], it is possible that different initialisations, interacting with the variance in label information, might have caused the optimisation procedure to reach, at times, suboptimal solutions.

Overall, results for the SSVAE look satisfactory and promising, with the important caveat that to obtain the best performance from the model the labelled examples need to be chosen to ensure a certain degree of representativeness.

4.5.2 FixMatch

Despite the potential shown by the SSVAE, its reliance on a rather unstable training objective and its rather complicated probabilistic formulation might put off end users. Seeking an alternative highly reliable, simple to understand and hyperparameter-light approach to SSL, we decided to adapt the novel FixMatch [66] algorithm to the cerebellum dataset. FixMatch (see section 2.5.2) is deceptively simple and almost reminiscent of a regularisation procedure, but still leverages the flexibility of arbitrary Deep Learning architectures to deliver state-of-the-art SSL performance. In our case, we used a simple MLP.

First, we optimised over the parameters of the MLP using Bayesian Optimisation with the 5-fold cross-validation accuracy as the objective. The hyperparameters optimised were: {`learning_rate, n_layers, n_units`}. Then, we used this architecture for the FixMatch training procedure.

Only 4 labels per class were used once again, for comparison with the SSVAE. The process involved 64 FixMatch steps per epoch for 32 epochs, giving a total of 2048 FixMatch steps that were found sufficient to attain convergence. FixMatch-specific hyperparameters such as the pseudo-label threshold and temperature were kept at the default values found through the extensive ablation studies by [66].

Regrettably, computational and time constraints did not allow us to benchmark our FixMatch architecture using the same leave-one-out cross-validation procedure as the other models. Nonetheless, table 4.4 reports the validation accuracy and F1-score of 29 FixMatch model runs trained with different subsets of ground-truth labels and parameter initialisations.

Model	Accuracy	F1-score	n features	n labels
FixMatch MLP	48.5 ± 9.3	36.8 ± 7.7	700	24

Table 4.4: Performance of the FixMatch model on the subset of labelled data points not used during training. Mean and standard deviations of 29 runs.

As the validation fold in FixMatch is not used during training at any time and we do not perform any optimisation on the FixMatch-specific hyperparameters, performance on the validation set could still be considered an adequate proxy of generalisation performance. However, due to the heavy class imbalance in the dataset, no real conclusion can be drawn at present from these numbers, and they are only to be considered as a tentative and preliminary evaluation attempt. Future work is needed to properly evaluate the model.

4.6 Error Analysis

Looking only at accuracies and F1 scores for the models examined does not let us satisfactorily evaluate progress, or lack thereof, in our modelling effort. To visually compare the precision and recall of different classes in the dataset under different model architectures, let us examine the respective confusion matrices.

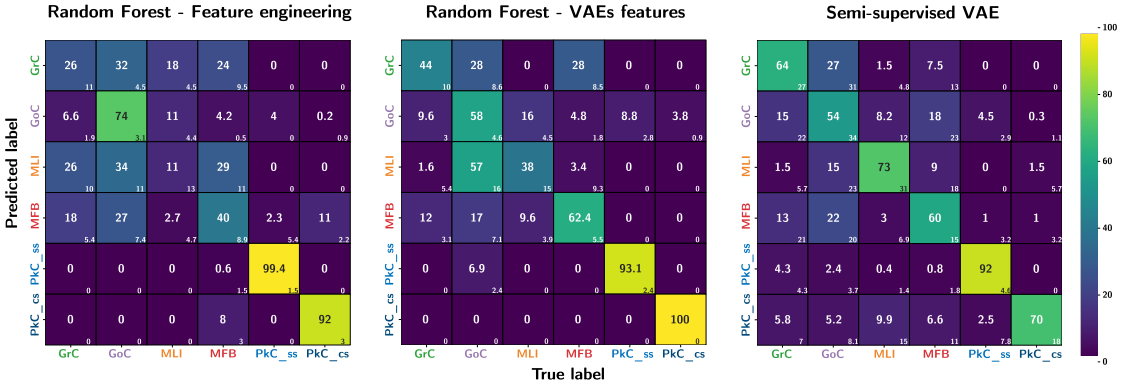


Figure 4.6: Confusion matrices for three of our most representative models. From left to right: random forest model using only engineered features; random forest model using features extracted with two encoder networks, one for the ACGs and one for the waveforms; classifier network from a SSVAE. The first two confusion matrices are averaged over 50 LOOCV trials, while the last is an average of 11 trials. Large numbers at the centres of squares indicate mean values, small numbers in the corners are standard deviations. All confusion matrices show percentage values normalised along the prediction axis.

Figure 4.6 clearly demonstrates how the performance of the feature engineering baseline is strongly determined by the model only learning the majority classes (i.e. `GoC`, `PkC_ss`, and `PkC_cs`), while having very poor performance for all other classes. Using the VAE features compensates for this, achieving a greater deal of class balance in the predictions, with relatively low expenses for the majority classes. However, the SSVAE is perhaps the one achieving a more balanced classification outcome, this time penalising performance on the majority classes.

These figures make sense in light of the fact that the SSVAE has access only to balanced labels during training, and cannot skew the classification performance towards the majority classes if they are not in fact easier to separate.

To further break down and understand the similarities, differences and vulnerabilities of different models, and grasp why some of the majority classes might be penalised in more complex models, it is instructive to look at the most misclassified examples during LOOCV for each algorithm in figure 4.7. They confirm what is already a common denominator in the confusion matrices, which is that most mistakes are either false positive or false negative identifications of Golgi cells.

A possible explanation for this is *off-target* genetic expression of ChR2 (most commonly in Purkinje cells), which is known to be an important confound in optotagging experiments. It can often be resolved via *post-hoc* histology, although this is not always performed after data acquisition. At present, it is not possible for us to determine if Golgi cells are either naturally more variable than other cells, or simply have a few corrupt labels due to off-target expression. Judging by the impact on our classifiers, this should be a priority to be investigated in future work.

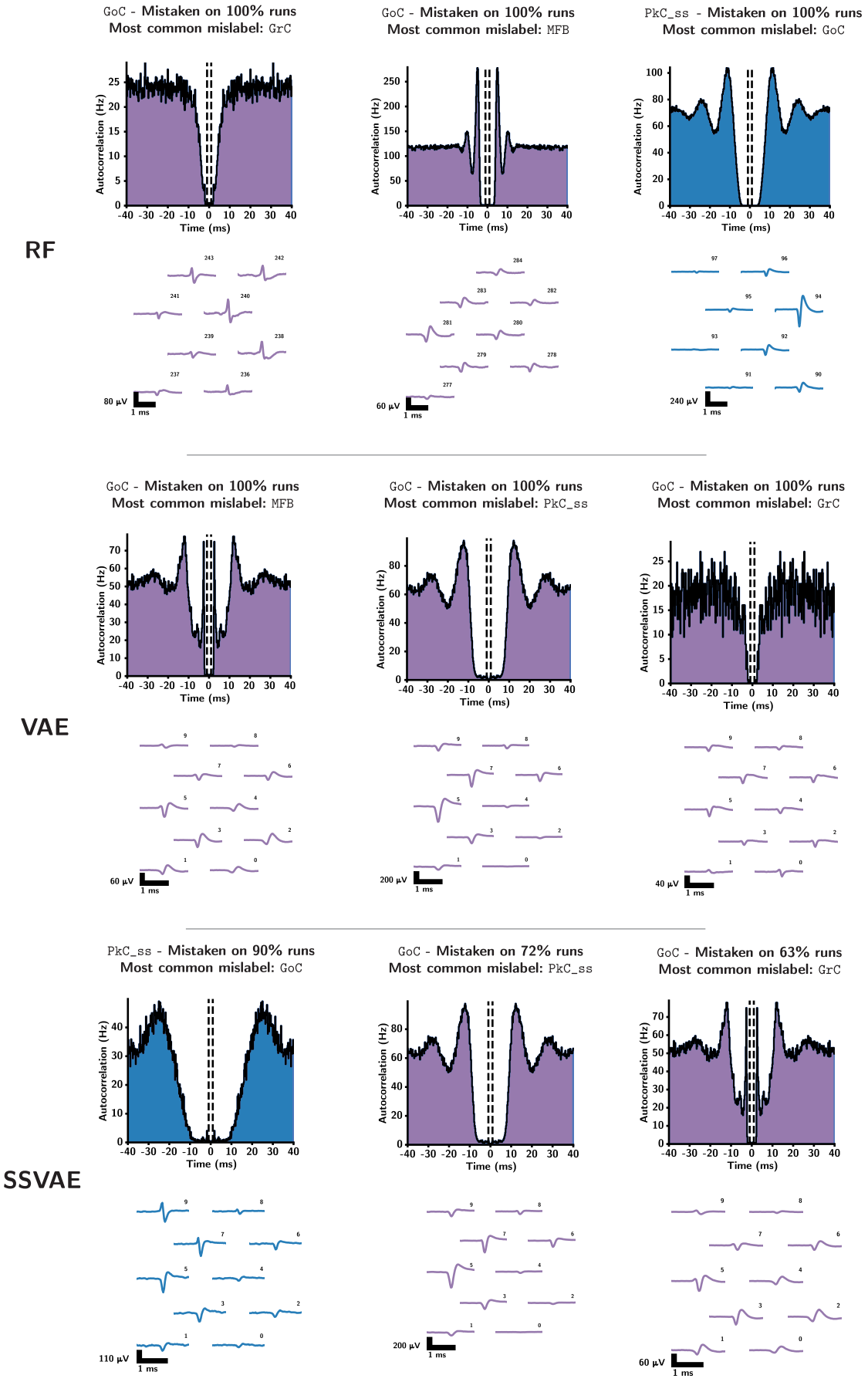


Figure 4.7: Top 3 most common mistakes for each class of models evaluated with LOOCV.

Chapter 5

Discussion and Conclusion

Cell types classification from high-density extracellular recordings *in vivo* is a highly non-trivial task, with no reliable solutions in the literature. As we have shown, even expert electrophysiologists fail at showing reliable performance in classifying units, manifesting how developing machine learning models to solve the issue can not only result in time-saving tools but in the resolution of a hard, ill-specified problem.

Results of our experiments revealed how, as expected, deep semi-supervised methods can be valuable assets in tackling the cell types classification problem. On the one hand, we have shown how using variational autoencoders to produce rich representations of data can improve the performance of models relying on more traditional, literature-grounded, engineered features. On the other hand, we also demonstrated how, using far fewer labels than traditional models, end-to-end deep SSL methods can be of great promise for the task, albeit at the cost of more complex interpretability for researchers having to adopt such a tool out of the box.

Let us now briefly evaluate the impact of our findings in relation to our initial objectives and the broader literature on cell types classification, including some forward directions that will bring our models to distribution.

5.1 Reassessment of research aims

Our first explicit aim was to improve methods that could work with feature engineering approaches so that researchers could work with familiar and understandable constructs while also harnessing the power of machine learning methods. Through our VAEs, we have shown how learnt representation can be used side-by-side with more traditional features to improve the performance of downstream models. Importantly, we also demonstrated how those representations indeed capture most of the variance that an electrophysiologist would see in real data (Figure 4.5). In following this first line of research, we have also established how human experts reach poor performance in the task, confirming our initial doubts about the use of expert-derived priors.

Our second aim of completely abandoning feature engineering approaches in favour of deep semi-supervised methods was also particularly fruitful, setting the stage for future research. The SSVAE model rivalled the performance of our representation learning approaches using only a fraction of all the labels used by other models. Not only that, but it did also show more balance in the predictions.

Moreover, the custom data augmentations and wrappers for deep models that we developed can be readily used in all forthcoming explorations, and serve as an inspiration for other deep learning tasks using Neuropixels data. This will also be directly explored in future work, where the FixMatch model will be repeatedly run to yield meaningful evaluation metrics.

Compared to previous studies in the literature on cell types classification [11–13, 84], the results presented here might look underwhelming at first. However, quite the opposite is the case.

First of all, it should be stressed that there is no precedent in the literature at successfully tackling cell types classification with high-density probes, especially using data coming from awake mice.

Second, unlike some previous studies [11, 13], here we followed a clear machine learning pipeline, taking all necessary precautions to avoid reporting inflated results.

Third, the methods introduced allow us to use labelled data with unprecedented efficiency in the field. As a result, each new data point coming from the complicated optotagging experiments can be used to drive meaningful improvements in the models, lowering overall ground-truth data requirements and saving both time and resources. This can serve of inspiration for future work, outside of the specific models and architectures proposed here.

Finally, it should be noted that while we benchmarked architectures, we did not settle on a single, final, model for the task. This is a venture left for future work, as new data is being acquired in the Häusser lab at the time of writing.

5.2 Future outlook

A few steps still need to be taken before our research can result in a model to be deployed and used by research laboratories working with Neuropixels around the world.

At present, new recordings are undergoing the pre-processing steps to yield novel ground-truth units for under-represented classes in the cerebellum dataset. Moreover, further unlabelled data is also undergoing manual curation to be included in the dataset. Having established the capabilities of the semi-supervised approach, it is clear how adding more unlabeled data can also be of great help during modelling, especially since some small cells, like GrCs and MLIs, will always be naturally under-represented given the experimental difficulties in recording them.

A further pressing direction for future research regarding the dataset is the careful reexamination of the Golgi cell class, to determine if its variability is biological or indeed caused by unwanted sources of variance like off-target ChR2 expression.

More efforts can also be made in the modelling direction. In the present exploration, all our deep architectures were limited to simple MLPs. It is conceivable that the growth of the dataset can be met with the adoption of other architectures, such as convolutional networks, which are compatible both with the VAE and the FixMatch framework.

Given the successes of the β-VAE in our representation learning experiments, an exciting avenue for future explorations would be the adoption of the Factor-VAE [60], an extension of the VAE framework that provides better-disentangled representations without sacrificing the quality of reconstructions.

Further, considering the outcomes of both the representation learning approach and the SS-VAE, an exciting future direction to prioritise is to combine those two models following [61] in a generative model with two layers of stochastic variables (i.e. what they call M1+M2 in the original study, see section 2.5.1), which has been shown to be superior to both the SSVAE on its own and representation learning followed by any other downstream model.

Looking towards deployment, it is our intent to adopt Bayesian methods for improved uncertainty calibration. This would amount to using explicitly probabilistic models such as Gaussian process classifiers in the representation learning setting followed by downstream models. In the end-to-end deep semi-supervised learning scenario, this will practically mean applying *post-hoc* Bayesian approximations to the learned parameters of our networks (for example using Laplace approximations [71]), therefore retaining performance but gaining better uncertainty estimates. Giving reliable estimates of model uncertainty is absolutely crucial to the deployment and subsequent practical adoption of the model, as researchers need to be equipped with the best possible information to guide their decision-making. This tool aims to be something onto which further analyses are solidly built, and as such needs to be as transparent as possible. [1]

As a final aside, it should be mentioned that the relevance of the cell types classification problem is such that an international collaboration is being set up to encourage data sharing and solve the task in a robust and reproducible way. This will be facilitated by some of the steps taken in the present study, including the use of our custom dashboard [2] to transparently

[1] This is also the reason why we release all code, data and optimisation logs from our experiments at `https://github.com/fededagos/celltypes-classification`

[2] Available at `https://files.fededagos.me/features/`

summarise and share all relevant aspects of the dataset across laboratories, assisting biologists in data exploration without requiring them to be fluent in machine learning methods.

5.3 Limitations

As described in section 2.4.2, the adoption of SSL methods is implicitly reliant on a few cardinal assumptions that must be acknowledged when modelling. But what happens if they do not hold?
Quite simply, it would imply that SSL methods are bound to fail on this task, either by not learning any effective decision boundaries or by learning meaningless ones. However, given the apparent reasonableness of the *smoothness, cluster* and *manifold* assumption, a failure of SSL would also be very informative on the nature and solvability of the task. Having models that are limited by such assumptions can be a desirable characteristic in low data settings such as ours, which are more prone to overfitting. Nonetheless, it should be recognised that our explorations were effectively limited by such hypotheses, which need to be more effectively scrutinised in the future.

Additional limitations of our explorations are the lack of a proper test set and the incomplete evaluation of the FixMatch architecture. These will be the object of upcoming work, with the help of incoming ground-truth data and an increase in time and computational resources.

5.4 Conclusion

The ultimate goal of systems neuroscience is to understand the functions and computations performed by neural circuits that mediate complex behaviour in living animals. Recent advancements in electrophysiology equip researchers with tools to record simultaneously from an unprecedented number of cells, opening new avenues for the description of neural computation. In this process, an understanding of how different cells integrate, process and transmit information will be pivotal.

Here we showed how deep semi-supervised learning methods can successfully be applied to novel datasets trying to tackle cell types classification from high-density cerebellar recordings. In doing so, we improved the performance over both the human experts baseline and a feature engineering baseline model, while at the same time showing how data requirements may be lowered if the potential of unlabelled data is correctly used.

We hope new research will stem from our explorations, leading to the first ever machine learning model able to satisfactorily classify non-trivial cell types from high-density extracellular recordings *in vivo*.

Acknowledgments

I want to sincerely thank Maxime Beau and Marlies Oostland for the continuous feedback during the project, and for how they made me appreciate the complexities of day-to-day life as an experimentalist.
I want to also thank Ago Lajko and Gabriela Martinez Lopera for their work on data cleaning, pre-processing and feature extraction as my predecessors on the cell types project. Along with them, I want to thank Dimitar Kostadinov for having kept this project running over the years.
My gratitude goes also to Arnd Roth and Beverley Clark, for their precise and decisive comments on the final draft of this thesis.
Finally, I want to thank Michael Häusser for making all of this possible in the first place.

References

1. Galvani, L., Volta, A., Zambelli, J. & Burndy Library, d. D. *Aloysii Galvani De viribus electricitatis in motu musculari commentarius* lat. http://archive.org/details/AloysiiGalvaniD00Galv (Bononiae : Ex Typographia Instituti Scientiarium, 1791) (cit. on p. 3).
2. Hodgkin, A. L. & Huxley, A. F. Action Potentials Recorded from Inside a Nerve Fibre. en. *Nature* **144.** Number: 3651 Publisher: Nature Publishing Group, 710–711. ISSN: 1476-4687. https://www.nature.com/articles/144710a0 (Oct. 1939) (cit. on p. 3).
3. Stevenson, I. H. & Kording, K. P. How advances in neural recording affect data analysis. en. *Nature Neuroscience* **14.** Number: 2 Publisher: Nature Publishing Group, 139–142. ISSN: 1546-1726. https://www.nature.com/articles/nn.2731 (Feb. 2011) (cit. on p. 3).
4. Hong, G. & Lieber, C. M. Novel electrode technologies for neural recordings. en. *Nature Reviews Neuroscience* **20.** Number: 6 Publisher: Nature Publishing Group, 330–345. ISSN: 1471-0048. https://www.nature.com/articles/s41583-019-0140-6 (June 2019) (cit. on pp. 3, 11).
5. Jun, J. J. *et al.* Fully Integrated Silicon Probes for High-Density Recording of Neural Activity. *Nature* **551,** 232–236. ISSN: 0028-0836. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5955206/ (Nov. 2017) (cit. on pp. 3, 10, 11).
6. Steinmetz, N. A. *et al.* Neuropixels 2.0: A miniaturized high-density probe for stable, long-term brain recordings. *Science* **372.** Publisher: American Association for the Advancement of Science, eabf4588. https://www.science.org/doi/10.1126/science.abf4588 (Apr. 2021) (cit. on pp. 3, 10, 11).
7. Quiroga, R. Q. Spike sorting. en. *Scholarpedia* **2,** 3583. ISSN: 1941-6016. http://www.scholarpedia.org/article/Spike_sorting (Dec. 2007) (cit. on pp. 3, 10).
8. Zeng, H. & Sanes, J. R. Neuronal cell-type classification: challenges, opportunities and the path forward. en. *Nature Reviews Neuroscience* **18.** Number: 9 Publisher: Nature Publishing Group, 530–546. ISSN: 1471-0048. https://www.nature.com/articles/nrn.2017.85 (Sept. 2017) (cit. on pp. 3, 20).
9. Abeles, M. & Goldstein, M. Multispike train analysis. *Proceedings of the IEEE* **65.** Conference Name: Proceedings of the IEEE, 762–773. ISSN: 1558-2256 (May 1977) (cit. on p. 3).
10. Lewicki, M. S. A review of methods for spike sorting: the detection and classification of neural action potentials. eng. *Network (Bristol, England)* **9,** R53–78. ISSN: 0954-898X (Nov. 1998) (cit. on p. 3).
11. Ruigrok, T. J. H., Hensbroek, R. A. & Simpson, J. I. Spontaneous Activity Signatures of Morphologically Identified Interneurons in the Vestibulocerebellum. en. *Journal of Neuroscience* **31.** Publisher: Society for Neuroscience Section: Articles, 712–724. ISSN: 0270-6474, 1529-2401. https://www.jneurosci.org/content/31/2/712 (Jan. 2011) (cit. on pp. 4, 20, 35, 36).
12. Dijck, G. V. *et al.* Probabilistic Identification of Cerebellar Cortical Neurones across Species. en. *PLOS ONE* **8.** Publisher: Public Library of Science, e57669. ISSN: 1932-6203. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0057669 (Mar. 2013) (cit. on pp. 4, 20, 21, 26, 35).

13. Hensbroek, R. A. *et al.* Identifying Purkinje cells using only their spontaneous simple spike activity. eng. *Journal of Neuroscience Methods* **232,** 173–180. ISSN: 1872-678X (July 2014) (cit. on pp. 4, 20, 21, 35, 36).

14. Holt, G. R. & Koch, C. Electrical Interactions via the Extracellular Potential Near Cell Bodies. en. *Journal of Computational Neuroscience* **6,** 169–184. ISSN: 1573-6873. https://doi.org/10.1023/A:1008832702585 (Mar. 1999) (cit. on pp. 4, 8–10).

15. Gold, C., Henze, D. A. & Koch, C. Using extracellular action potential recordings to constrain compartmental models. en. *Journal of Computational Neuroscience* **23,** 39–58. ISSN: 1573-6873. https://doi.org/10.1007/s10827-006-0018-2 (Aug. 2007) (cit. on p. 4).

16. Jia, X. *et al.* High-density extracellular probes reveal dendritic backpropagation and facilitate neuron classification. *Journal of Neurophysiology* **121.** Publisher: American Physiological Society, 1831–1847. ISSN: 0022-3077. https://journals.physiology.org/doi/full/10.1152/jn.00680.2018 (May 2019) (cit. on pp. 4, 20–22, 26).

17. Chapelle, O., Schölkopf, B. & Zien, A. *Semi-Supervised Learning* 1st. ISBN: 978-0-262-51412-5 (The MIT Press, 2010) (cit. on pp. 4, 13, 14).

18. Bzdok, D., Eickenberg, M., Grisel, O., Thirion, B. & Varoquaux, G. *Semi-Supervised Factored Logistic Regression for High-Dimensional Neuroimaging Data* in *Advances in Neural Information Processing Systems* **28** (Curran Associates, Inc., 2015). https://proceedings.neurips.cc/paper/2015/hash/06a15eb1c3836723b53e4abca8d9b879-Abstract.html (cit. on p. 4).

19. Zhang, J., Zhang, C., Yao, L., Zhao, X. & Long, Z. Brain State Decoding Based on fMRI Using Semisupervised Sparse Representation Classifications. en. *Computational Intelligence and Neuroscience* **2018.** Publisher: Hindawi, e3956536. ISSN: 1687-5265. https://www.hindawi.com/journals/cin/2018/3956536/ (Apr. 2018) (cit. on p. 4).

20. Takaya, E., Takeichi, Y., Ozaki, M. & Kurihara, S. Sequential semi-supervised segmentation for serial electron microscopy image with small number of labels. en. *Journal of Neuroscience Methods* **351,** 109066. ISSN: 0165-0270. https://www.sciencedirect.com/science/article/pii/S0165027021000017 (Mar. 2021) (cit. on p. 4).

21. Dan, Y., Tao, J., Fu, J. & Zhou, D. Possibilistic Clustering-Promoting Semi-Supervised Learning for EEG-Based Emotion Recognition. *Frontiers in Neuroscience* **15.** ISSN: 1662-453X. https://www.frontiersin.org/articles/10.3389/fnins.2021.690044 (2021) (cit. on p. 4).

22. Ito, R., Nakae, K., Hata, J., Okano, H. & Ishii, S. Semi-supervised deep learning of brain tissue segmentation. en. *Neural Networks* **116,** 25–34. ISSN: 0893-6080. https://www.sciencedirect.com/science/article/pii/S0893608019300954 (Aug. 2019) (cit. on p. 4).

23. Beau, M., Lajko, A. & Martínez, G. *m-beau/NeuroPyxels: Public release of npyx* Sept. 2021. https://zenodo.org/record/5509776 (cit. on pp. 5, 24, 27).

24. Shahriari, B., Swersky, K., Wang, Z., Adams, R. P. & de Freitas, N. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE* **104.** Conference Name: Proceedings of the IEEE, 148–175. ISSN: 1558-2256 (Jan. 2016) (cit. on pp. 5, 27, 29, 31, 47).

25. Akiba, T., Sano, S., Yanase, T., Ohta, T. & Koyama, M. *Optuna: A Next-generation Hyperparameter Optimization Framework* arXiv:1907.10902 [cs, stat]. July 2019. http://arxiv.org/abs/1907.10902 (cit. on pp. 5, 27, 47).

26. *Principles of neural science* 5th ed (ed Kandel, E. R.) ISBN: 978-0-07-139011-8 (McGraw-Hill, New York, 2013) (cit. on pp. 6–8).

27. *Neuroscience* 5th ed (ed Purves, D.) ISBN: 978-0-87893-695-3 (Sinauer Associates, Sunderland, Mass, 2012) (cit. on pp. 6, 7).

28. Schmahmann, J. D. & Caplan, D. Cognition, emotion and the cerebellum. *Brain* **129,** 290–292. ISSN: 0006-8950. https://doi.org/10.1093/brain/awh729 (Feb. 2006) (cit. on p. 6).

29. Buckner, R. L. The Cerebellum and Cognitive Function: 25 Years of Insight from Anatomy and Neuroimaging. en. *Neuron* **80,** 807–815. ISSN: 0896-6273. https://www.sciencedirect.com/science/article/pii/S0896627313009963 (Oct. 2013) (cit. on p. 6).

30. Wagner, M. J., Kim, T. H., Savall, J., Schnitzer, M. J. & Luo, L. Cerebellar granule cells encode the expectation of reward. en. *Nature* **544.** Number: 7648 Publisher: Nature Publishing Group, 96–100. ISSN: 1476-4687. https://www.nature.com/articles/nature21726 (Apr. 2017) (cit. on pp. 6, 23).

31. Kostadinov, D., Beau, M., Blanco-Pozo, M. & Häusser, M. Predictive and reactive reward signals conveyed by climbing fiber inputs to cerebellar Purkinje cells. en. *Nature Neuroscience* **22.** Number: 6 Publisher: Nature Publishing Group, 950–962. ISSN: 1546-1726. https://www.nature.com/articles/s41593-019-0381-8 (June 2019) (cit. on p. 6).

32. D'Mello, A. M. & Stoodley, C. J. Cerebro-cerebellar circuits in autism spectrum disorder. *Frontiers in Neuroscience* **9.** ISSN: 1662-453X. https://www.frontiersin.org/articles/10.3389/fnins.2015.00408 (2015) (cit. on p. 6).

33. Oostland, M., Buijink, M. R. & van Hooft, J. A. Serotonergic control of Purkinje cell maturation and climbing fibre elimination by 5-HT3 receptors in the juvenile mouse cerebellum. en. *The Journal of Physiology* **591,** 1793–1807. ISSN: 1469-7793. https://onlinelibrary.wiley.com/doi/abs/10.1113/jphysiol.2012.246413 (2013) (cit. on p. 7).

34. Suvrathan, A., Payne, H. L. & Raymond, J. L. Timing Rules for Synaptic Plasticity Matched to Behavioral Function. en. *Neuron* **92,** 959–967. ISSN: 0896-6273. https://www.sciencedirect.com/science/article/pii/S0896627316307231 (Dec. 2016) (cit. on p. 7).

35. Rieubland, S., Roth, A. & Häusser, M. Structured Connectivity in Cerebellar Inhibitory Networks. en. *Neuron* **81,** 913–929. ISSN: 0896-6273. https://www.sciencedirect.com/science/article/pii/S0896627313011902 (Feb. 2014) (cit. on p. 8).

36. De Zeeuw, C. I., Lisberger, S. G. & Raymond, J. L. Diversity and dynamism in the cerebellum. en. *Nature Neuroscience* **24.** Number: 2 Publisher: Nature Publishing Group, 160–167. ISSN: 1546-1726. https://www.nature.com/articles/s41593-020-00754-9 (Feb. 2021) (cit. on p. 8).

37. Bean, B. P. The action potential in mammalian central neurons. en. *Nature Reviews Neuroscience* **8.** Number: 6 Publisher: Nature Publishing Group, 451–465. ISSN: 1471-0048. https://www.nature.com/articles/nrn2148 (June 2007) (cit. on p. 8).

38. Heinricher, M. M. *2 Principles of Extracellular Single-Unit Recording* in (2004) (cit. on pp. 8, 9).

39. Gold, C., Henze, D. A., Koch, C. & Buzsáki, G. On the Origin of the Extracellular Action Potential Waveform: A Modeling Study. *Journal of Neurophysiology* **95.** Publisher: American Physiological Society, 3113–3128. ISSN: 0022-3077. https://journals.physiology.org/doi/full/10.1152/jn.00979.2005 (May 2006) (cit. on pp. 8–10).

40. Tarpin, T. *et al.* en. in *Measuring Cerebellar Function* (ed Sillitoe, R. V.) 187–209 (Springer US, New York, NY, 2022). ISBN: 978-1-07-162026-7. https://doi.org/10.1007/978-1-0716-2026-7_10 (cit. on pp. 9, 12).

41. Lindén, H. *et al.* LFPy: a tool for biophysical simulation of extracellular potentials generated by detailed model neurons. *Frontiers in Neuroinformatics* **7.** ISSN: 1662-5196. https://www.frontiersin.org/articles/10.3389/fninf.2013.00041 (2014) (cit. on p. 10).

42. Hubel, D. H. Tungsten Microelectrode for Recording from Single Units. *Science* **125.** Publisher: American Association for the Advancement of Science, 549–550. https://www.science.org/doi/10.1126/science.125.3247.549 (Mar. 1957) (cit. on p. 10).

43. Rey, H. G., Pedreira, C. & Quian Quiroga, R. Past, present and future of spike sorting techniques. en. *Brain Research Bulletin. Advances in electrophysiological data analysis* **119,** 106–117. ISSN: 0361-9230. https://www.sciencedirect.com/science/article/pii/S0361923015000684 (Oct. 2015) (cit. on p. 11).

44. Buccino, A. P. *et al.* SpikeInterface, a unified framework for spike sorting. *eLife* **9** (eds Colgin, L. L., Grün, S. & Kloosterman, F.) e61834. ISSN: 2050-084X. https://doi.org/10.7554/eLife.61834 (Nov. 2020) (cit. on p. 11).

45. Jun, J. J. *et al. Real-time spike sorting platform for high-density extracellular probes with ground-truth validation and drift correction* en. Pages: 101030 Section: New Results. Jan. 2017. https://www.biorxiv.org/content/10.1101/101030v2 (cit. on p. 11).

46. Rossant, C. *et al.* Spike sorting for large, dense electrode arrays. en. *Nature Neuroscience* **19.** Number: 4 Publisher: Nature Publishing Group, 634–641. ISSN: 1546-1726. https://www.nature.com/articles/nn.4268 (Apr. 2016) (cit. on pp. 11, 24).

47. Pachitariu, M., Steinmetz, N. A., Kadir, S. N., Carandini, M. & Harris, K. D. *Fast and accurate spike sorting of high-channel count probes with KiloSort* in *Advances in Neural Information Processing Systems* **29** (Curran Associates, Inc., 2016). https://proceedings.neurips.cc/paper/2016/hash/1145a30ff80745b56fb0cecf65305017-Abstract.html (cit. on pp. 11, 24).

48. Dayan, P. & Abbott, L. F. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems* ISBN: 0-262-54185-8 (The MIT Press, 2005) (cit. on p. 12).

49. Zhu, X. & Goldberg, A. B. Introduction to Semi-Supervised Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* **3.** Publisher: Morgan & Claypool Publishers, 1–130. ISSN: 1939-4608. https://www.morganclaypool.com/doi/abs/10.2200/S00196ED1V01Y200906AIM006 (Jan. 2009) (cit. on p. 13).

50. Ouali, Y., Hudelot, C. & Tami, M. *An Overview of Deep Semi-Supervised Learning* arXiv:2006.05278 [cs, stat]. July 2020. http://arxiv.org/abs/2006.05278 (cit. on pp. 13, 14, 18).

51. Von Luxburg, U. *A Tutorial on Spectral Clustering* arXiv:0711.0189 [cs]. Nov. 2007. http://arxiv.org/abs/0711.0189 (cit. on p. 14).

52. Kingma, D. P. & Welling, M. An Introduction to Variational Autoencoders. *Foundations and Trends® in Machine Learning* **12.** arXiv:1906.02691 [cs, stat], 307–392. ISSN: 1935-8237, 1935-8245. http://arxiv.org/abs/1906.02691 (2019) (cit. on pp. 14–16).

53. Bengio, Y., Courville, A. & Vincent, P. *Representation Learning: A Review and New Perspectives* arXiv:1206.5538 [cs]. Apr. 2014. http://arxiv.org/abs/1206.5538 (cit. on pp. 15, 17).

54. Paige, B. *Variational autoencoders (and other deep generative models) - UCL COMP0171 Week 8* Handout. 2022 (cit. on p. 15).

55. Kingma, D. P. & Welling, M. *Auto-Encoding Variational Bayes* arXiv:1312.6114 [cs, stat]. May 2014. http://arxiv.org/abs/1312.6114 (cit. on pp. 15, 16, 30).

56. Rezende, D. J., Mohamed, S. & Wierstra, D. *Stochastic Backpropagation and Approximate Inference in Deep Generative Models* arXiv:1401.4082 [cs, stat]. May 2014. http://arxiv.org/abs/1401.4082 (cit. on p. 15).

57. Barber, D. in *Bayesian Reasoning and Machine Learning* (Cambridge University Press, 2012) (cit. on p. 16).

58. Higgins, I. *et al. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework* en. in (July 2022). https://openreview.net/forum?id=Sy2fzU9gl (cit. on pp. 16, 17).

59. Burgess, C. P. *et al. Understanding disentangling in β-VAE* arXiv:1804.03599 [cs, stat]. Apr. 2018. http://arxiv.org/abs/1804.03599 (cit. on pp. 16, 17).

60. Kim, H. & Mnih, A. *Disentangling by Factorising* arXiv:1802.05983 [cs, stat]. July 2019. http://arxiv.org/abs/1802.05983 (cit. on pp. 17, 36).

61. Kingma, D. P., Rezende, D. J., Mohamed, S. & Welling, M. *Semi-Supervised Learning with Deep Generative Models* arXiv:1406.5298 [cs, stat]. Oct. 2014. http://arxiv.org/abs/1406.5298 (cit. on pp. 17, 31, 36).

62. Siddharth, N. *et al. Learning Disentangled Representations with Semi-Supervised Deep Generative Models* arXiv:1706.00400 [cs, stat]. Nov. 2017. http://arxiv.org/abs/1706.00400 (cit. on p. 18).

63. Bachman, P., Alsharif, O. & Precup, D. *Learning with Pseudo-Ensembles* arXiv:1412.4864 [cs, stat]. Dec. 2014. http://arxiv.org/abs/1412.4864 (cit. on p. 17).

64. Sajjadi, M., Javanmardi, M. & Tasdizen, T. *Regularization With Stochastic Transformations and Perturbations for Deep Semi-Supervised Learning* arXiv:1606.04586 [cs]. June 2016. http://arxiv.org/abs/1606.04586 (cit. on p. 17).

65. Lee, D.-H. *Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks* in (2013) (cit. on p. 18).

66. Sohn, K. *et al. FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence* arXiv:2001.07685 [cs, stat]. Nov. 2020. http://arxiv.org/abs/2001.07685 (cit. on pp. 18, 19, 29, 31, 32).

67. Berthelot, D. *et al. ReMixMatch: Semi-Supervised Learning with Distribution Alignment and Augmentation Anchoring* arXiv:1911.09785 [cs, stat]. Feb. 2020. http://arxiv.org/abs/1911.09785 (cit. on p. 18).

68. Berthelot, D. *et al. MixMatch: A Holistic Approach to Semi-Supervised Learning* arXiv:1905.02249 [cs, stat]. Oct. 2019. http://arxiv.org/abs/1905.02249 (cit. on p. 18).

69. Tarvainen, A. & Valpola, H. *Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results* arXiv:1703.01780 [cs, stat]. Apr. 2018. http://arxiv.org/abs/1703.01780 (cit. on p. 18).

70. Cubuk, E. D., Zoph, B., Shlens, J. & Le, Q. V. *RandAugment: Practical automated data augmentation with a reduced search space* arXiv:1909.13719 [cs]. Nov. 2019. http://arxiv.org/abs/1909.13719 (cit. on pp. 18, 29).

71. Daxberger, E. *et al. Laplace Redux – Effortless Bayesian Deep Learning* arXiv:2106.14806 [cs, stat]. Mar. 2022. http://arxiv.org/abs/2106.14806 (cit. on pp. 18, 36).

72. Kristiadi, A., Hein, M. & Hennig, P. *Being Bayesian, Even Just a Bit, Fixes Overconfidence in ReLU Networks* arXiv:2002.10118 [cs, stat]. July 2020. http://arxiv.org/abs/2002.10118 (cit. on p. 18).

73. Ramon y Cajal, S. & Azoulay, L. in *Histologie du systeme nerveux de l'homme et des vertebres* 2v–2v (1955) (cit. on p. 20).

74. Masland, R. H. The neuronal organization of the retina. eng. *Neuron* **76,** 266–280. ISSN: 1097-4199 (Oct. 2012) (cit. on p. 20).

75. Sanes, J. R. & Masland, R. H. The Types of Retinal Ganglion Cells: Current Status and Implications for Neuronal Classification. en. *Annual Review of Neuroscience* **38,** 221–246. ISSN: 0147-006X, 1545-4126. https://www.annualreviews.org/doi/10.1146/annurev-neuro-071714-034120 (July 2015) (cit. on p. 20).

76. Tasic, B. *et al.* Adult mouse cortical cell taxonomy revealed by single cell transcriptomics. en. *Nature Neuroscience* **19.** Number: 2 Publisher: Nature Publishing Group, 335–346. ISSN: 1546-1726. https://www.nature.com/articles/nn.4216 (Feb. 2016) (cit. on p. 20).

77. Wilson, F. A., O'Scalaidhe, S. P. & Goldman-Rakic, P. S. Functional synergism between putative gamma-aminobutyrate-containing neurons and pyramidal neurons in prefrontal cortex. *Proceedings of the National Academy of Sciences* **91.** Publisher: Proceedings of the National Academy of Sciences, 4009–4013. https://www.pnas.org/doi/abs/10.1073/pnas.91.9.4009 (Apr. 1994) (cit. on p. 20).

78. Barthó, P. *et al.* Characterization of Neocortical Principal Cells and Interneurons by Network Interactions and Extracellular Features. *Journal of Neurophysiology* **92.** Publisher: American Physiological Society, 600–608. ISSN: 0022-3077. https://journals.physiology.org/doi/full/10.1152/jn.01170.2003 (July 2004) (cit. on p. 20).

79. Petersen, P. C., Siegle, J. H., Steinmetz, N. A., Mahallati, S. & Buzsáki, G. CellExplorer: A framework for visualizing and characterizing single neurons. en. *Neuron* **109,** 3594–3608.e2. ISSN: 0896-6273. https://www.sciencedirect.com/science/article/pii/S0896627321006565 (Nov. 2021) (cit. on p. 20).

80. Özcan, O. O. *et al.* Differential Coding Strategies in Glutamatergic and GABAergic Neurons in the Medial Cerebellar Nucleus. eng. *The Journal of Neuroscience: The Official Journal of the Society for Neuroscience* **40,** 159–170. ISSN: 1529-2401 (Jan. 2020) (cit. on pp. 20–22, 26).

81. Sibille, J. *et al. Strong and specific connections between retinal axon mosaics and midbrain neurons revealed by large scale paired recordings* en. Pages: 2021.09.09.459396 Section: New Results. Sept. 2021. https://www.biorxiv.org/content/10.1101/2021.09.09.459396v1 (cit. on pp. 20–22).

82. Huang, C. M., Mu, H. & Hsiao, C. F. Identification of cell types from action potential waveforms: cerebellar granule cells. eng. *Brain Research* **619,** 313–318. ISSN: 0006-8993 (Aug. 1993) (cit. on p. 20).

83. Pinault, D. A novel single-cell staining procedure performed in vivo under electrophysiological control: morpho-functional features of juxtacellularly labeled thalamic cells and other central neurons with biocytin or Neurobiotin. eng. *Journal of Neuroscience Methods* **65,** 113–136. ISSN: 0165-0270 (Apr. 1996) (cit. on pp. 20, 21).

84. Haar, S., Givon-Mayo, R., Barmack, N. H., Yakhnitsa, V. & Donchin, O. Spontaneous activity does not predict morphological type in cerebellar interneurons. eng. *The Journal of Neuroscience: The Official Journal of the Society for Neuroscience* **35,** 1432–1442. ISSN: 1529-2401 (Jan. 2015) (cit. on pp. 20, 21, 35).

85. Chen, S., Augustine, G. J. & Chadderton, P. Serial processing of kinematic signals by cerebellar circuitry during voluntary whisking. en. *Nature Communications* **8.** Number: 1 Publisher: Nature Publishing Group, 232. ISSN: 2041-1723. https://www.nature.com/articles/s41467-017-00312-1 (Aug. 2017) (cit. on p. 21).

86. Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* **16.** arXiv:1106.1813 [cs], 321–357. ISSN: 1076-9757. http://arxiv.org/abs/1106.1813 (June 2002) (cit. on p. 22).

87. Lima, S. Q., Hromádka, T., Znamenskiy, P. & Zador, A. M. PINP: A New Method of Tagging Neuronal Populations for Identification during In Vivo Electrophysiological Recording. *PLoS ONE* **4,** e6099. ISSN: 1932-6203. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2702752/ (July 2009) (cit. on p. 23).

88. Tan, N. G. A., Wu, W. & Seifalian, A. M. en. in *Applications of Nanoscience in Photomedicine* (eds Hamblin, M. R. & Avci, P.) 185–203 (Chandos Publishing, Oxford, Jan. 2015). ISBN: 978-1-907568-67-1. https://www.sciencedirect.com/science/article/pii/B9781907568671500101 (cit. on p. 23).

89. Deisseroth, K. Optogenetics: 10 years of microbial opsins in neuroscience. en. *Nature Neuroscience* **18,** 1213–1225. ISSN: 1546-1726. https://www.nature.com/articles/nn.4091 (Sept. 2015) (cit. on p. 23).

90. Witter, L., Rudolph, S., Pressler, R. T., Lahlaf, S. I. & Regehr, W. G. Purkinje Cell Collaterals Enable Output Signals from the Cerebellar Cortex to Feed Back to Purkinje Cells and Interneurons. en. *Neuron* **91,** 312–319. ISSN: 0896-6273. https://www.sciencedirect.com/science/article/pii/S0896627316302483 (July 2016) (cit. on p. 23).

91. Jelitai, M., Puggioni, P., Ishikawa, T., Rinaldi, A. & Duguid, I. Dendritic excitation–inhibition balance shapes cerebellar output during motor behaviour. en. *Nature Communications* **7.** Number: 1 Publisher: Nature Publishing Group, 13722. ISSN: 2041-1723. https://www.nature.com/articles/ncomms13722 (Dec. 2016) (cit. on p. 23).

92. Gurnani, H. & Silver, R. A. Multidimensional population activity in an electrically coupled inhibitory circuit in the cerebellar cortex. en. *Neuron* **109,** 1739–1753.e8. ISSN: 0896-6273. https://www.sciencedirect.com/science/article/pii/S0896627321001975 (May 2021) (cit. on p. 23).

93. Hull, C. & Regehr, W. G. Identification of an Inhibitory Circuit that Regulates Cerebellar Golgi Cell Activity. en. *Neuron* **73,** 149–158. ISSN: 0896-6273. https://www.sciencedirect.com/science/article/pii/S0896627311009949 (Jan. 2012) (cit. on p. 23).

94. Thölke, P. *et al. Class imbalance should not throw you off balance: Choosing the right classifiers and performance metrics for brain decoding with imbalanced data* en. Aug. 2022. https://www.biorxiv.org/content/10.1101/2022.07.18.500262v2 (cit. on p. 25).

95. Wang, B. & Zou, H. Honest leave-one-out cross-validation for estimating post-tuning generalization error. en. *Stat* **10.** _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/sta4.413, e413. ISSN: 2049-1573. https://onlinelibrary.wiley.com/doi/abs/10.1002/sta4.413 (2021) (cit. on p. 26).

96. Zhang, Y. & Yang, Y. Cross-validation for selecting a model selection procedure. en. *Journal of Econometrics* **187,** 95–112. ISSN: 0304-4076. https://www.sciencedirect.com/science/article/pii/S0304407615000305 (July 2015) (cit. on p. 26).

97. Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **12,** 2825–2830 (2011) (cit. on p. 27).

98. Wong, S. C., Gatt, A., Stamatescu, V. & McDonnell, M. D. *Understanding data augmentation for classification: when to warp?* arXiv:1609.08764 [cs]. Nov. 2016. http://arxiv.org/abs/1609.08764 (cit. on p. 28).

99. Perez, L. & Wang, J. *The Effectiveness of Data Augmentation in Image Classification using Deep Learning* arXiv:1712.04621 [cs]. Dec. 2017. http://arxiv.org/abs/1712.04621 (cit. on p. 28).

100. Vasconcelos, C. N. & Vasconcelos, B. N. *Convolutional Neural Network Committees for Melanoma Classification with Classical And Expert Knowledge Based Image Transforms Data Augmentation* Mar. 2017. http://arxiv.org/abs/1702.07025 (cit. on p. 28).

101. Xu, Y. *et al. Improved Relation Classification by Deep Recurrent Neural Networks with Data Augmentation* arXiv:1601.03651 [cs]. Oct. 2016. http://arxiv.org/abs/1601.03651 (cit. on p. 28).

102. Kumar, V., Choudhary, A. & Cho, E. *Data Augmentation using Pre-trained Transformer Models* arXiv:2003.02245 [cs]. Jan. 2021. http://arxiv.org/abs/2003.02245 (cit. on p. 28).

103. Hand, D. J. Classifier Technology and the Illusion of Progress. *Statistical Science* **21.** Publisher: Institute of Mathematical Statistics, 1–14. ISSN: 0883-4237, 2168-8745. https://projecteuclid.org/journals/statistical-science/volume-21/issue-1/Classifier-Technology-and-the-Illusion-of-Progress/10.1214/088342306000000060.full (Feb. 2006) (cit. on p. 30).

104. Lucas, J., Tucker, G., Grosse, R. B. & Norouzi, M. *Don' t Blame the ELBO! A Linear VAE Perspective on Posterior Collapse* in *Advances in Neural Information Processing Systems* **32** (Curran Associates, Inc., 2019). https://proceedings.neurips.cc/paper/2019/hash/7e3315fe390974fcf25e44a9445bd821-Abstract.html (cit. on p. 31).

105. Alemi, A. A. *et al. Fixing a Broken ELBO* arXiv:1711.00464 [cs, stat]. Feb. 2018. http://arxiv.org/abs/1711.00464 (cit. on p. 32).

106. Cremer, C., Li, X. & Duvenaud, D. *Inference Suboptimality in Variational Autoencoders* arXiv:1801.03558 [cs, stat]. May 2018. http://arxiv.org/abs/1801.03558 (cit. on p. 32).

107. Paszke, A. *et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library* arXiv:1912.01703 [cs, stat]. Dec. 2019. http://arxiv.org/abs/1912.01703 (cit. on p. 47).

108. Bingham, E. *et al. Pyro: Deep Universal Probabilistic Programming* arXiv:1810.09538 [cs, stat]. Oct. 2018. http://arxiv.org/abs/1810.09538 (cit. on p. 47).

List of Figures

List of Tables

Appendix A

Details on training procedures

Hardware specifications

All code was run on a custom Linux machine running Ubuntu[1] 20.04 LTS, with the following specifications:

Specification	Details	Specification	Details
Memory	64 GB	OS name	Ubuntu 20.04.4 LTS
Processor	Intel Core™ i7-6700 CPU @ 3.40GHZ × 8	OS type	64-bit
Graphics	NVIDIA Quadro M2000	CUDA version	11.3.1

Table A.1: Relevant specifications for the custom Linux machine on which the experiments were run.

Python environment

All experiments were run using Python[2] version 3.7.13. The virtual environment was managed with Anaconda[3]. Details of the packages used and their versions can be found at `https://github.com/fededagos/celltypes-classification/blob/main/environment.yml`

All deep models were trained using `pytorch`[4] [107], for the exception of the SSVAE which was modelled in `pyro`[5] [108].

Model fitting details

Hyperparameters

As mentioned in the main text, hyperparameter tuning was performed via Bayesian optimisation [24] using the open-source package `optuna` [25].

Here are the details of the hyperparameters for the different models.

Random Forest

```
{'criterion': 'entropy',
'max_features': 'log2',
'min_samples_leaf': 3,
'n_estimators': 328}
```

[1] `https://ubuntu.com/`
[2] `https://www.python.org/`
[3] `https://www.anaconda.com/`
[4] `https://pytorch.org/`
[5] `https://pyro.ai/`

Waveform VAE

```
{'batch_size': 69,
'beta': 4.63,
'd_latent': 10,
'dropout_l0': 0.47,
'lr': 0.00164022900982639,
'n_layers': 1,
'n_units_l0': 86,
'optimizer': 'Adam'}
```

ACG VAE

```
{'acg_d_latent': 14,
'acg_dropout_l0': 0.11,
'lr': 0.0037179027062778855,
'acg_n_layers': 1,
'acg_n_units_l0': 166,
'beta_acg': 1.03,
'optimizer': 'Adam'}
```

SSVAE

```
{'lr': 0.00496835069560119,
'aux_multiplier': 41,
'batch_size': 15,
'd_latent': 7,
'hidden_units_class_l1': 54,
'hidden_units_l1': 65,
'n_layers_classifier': 1,
'n_layers_vae': 1,
'non_linearity': 'tanh',
'optimizer': 'RMSprop'}
```

FixMatch

```
{'dropout_l0': 0.22395782802124392,
'dropout_l1': 0.401563000009004,
'lr': 0.004265516314362439,
'n_layers': 2,
'n_units_l0': 117,
'n_units_l1': 148,
'optimizer': 'RMSprop'}
```

Appendix B

Code and data availability

All code, checkpoints for trained deep models and Bayesian optimisation logs needed to replicate our experiments are released in the `celltypes-classification` repository at `https://github.com/fededagos/celltypes-classification`.

Source code for our custom dashboard to explore the features of the neurons in the dataset is also openly available at `https://github.com/fededagos/features-app`.

The cerebellum dataset, is available for download at `https://files.fededagos.me/datasets/cerebellum_dataset.h5`. It comes in the `hdf5`[1] file format, containing, for each neuron, the waveform, spike train, label and various metadata related to the optotagging experiments.

In order to efficiently work with the dataset, we created custom functions dealing with data extraction, cleaning and pre-processing from the raw `hdf5` files, which can be found on our repository under `utils/h5_utils.py`.

[1] `https://www.hdfgroup.org/solutions/hdf5/`