Alessio Paolo Buccino

A computationally-assisted approach to extracellular neural electrophysiology with multi-electrode arrays

Thesis submitted for the degree of Philosophiae Doctor

Department of Informatics
Faculty of Mathematics and Natural Sciences

Center for Integrative Neuroplasticity (CINPLA)
Faculty of Mathematics and Natural Sciences
University of Oslo
Department of Biomedical Engineering
University of California San Diego

2019
To my girlfriend, family, and friends
Acknowledgements

The work presented in this thesis was conducted from December 2015 to October 2019. I spent the first year at the NANO group, in the Department of Informatics of the University of Oslo. In December 2016, I moved to the Center of Integrative Neuroplasticity (CINPLA) at the Department of Biosciences of the University of Oslo. During my PhD, I spent one year, from August 2017 to August 2018, at the Integrated Systems Neuroengineering Lab of the University of California in San Diego.

I would first like to thank the SUURPh programme and the University of Oslo to have given me the chance to start this wonderful journey. Thank you for believing in me and giving me all the support I needed in these four years. Thank you Andy for your effort in starting this amazing training program (and for the awesome surfing sessions). Thank you Rachel for making my PhD life way easier. Thank you Kim for taking over and keeping this training programme going.

Philipp, you have been a great supervisor. I would really like to thank you for allowing me to follow and pursue my interests, always checking that I was on track.

Gaute, thank you for showing me the wonders of the computational neuroscience world. This thesis would not have been possible without your input and supervision.

Marianne, thank you for welcoming me to CINPLA and for always believing in me. You have built something unique and amazing and it has been a real pleasure to be part of it.

Gert, thank you for making my UCSD time unforgettable. I really enjoyed our science and maths discussions. Thank you for transmitting you vision and passion, and for the great support.

Thank you Torkel for the wonderful times spent together and for your valuable guidance in the realm of neuroscience.

Aslak, the chats we had about science and the future were priceless. Thank you for your trust and support.

I would like to deeply thank all the CINPLA members. Working and spending time with you in these years has been a pure joy. You know you have wonderful colleagues and friends when you never have a day in which you think “I don’t wanna go to work today”.

iii
I would like to thank the OSI H2 soccer team. Playing soccer (and partying) with you guys has represented an important part of this journey. Thank you for making me feel home and for comparing me to Pirlo.

A big thank you to all my "norwegian" friends. Alessio, Antonio, Sunniva, Stephanie, Emil, Ruben, Donato, and the rest of the crew: thank you for the great times spent together.

I would like to thank my mom, my dad, and my sister for the endless support. I know it is hard to have a family member away for such a long time, but I hope that this period will pay off in the future.

Finally, thank you Silvia for your love and immense support. I would not have been able to achieve this without you always by my side.

[Signature]

Oslo, December 2019
Abstract

One of the most used techniques to study neural circuit function is extracellular electrophysiology, which enables one to measure the electrical activity of neuron communication using electrodes placed in the extracellular space of the neural tissue. The power of this technique lies in the opportunity to link single neuron and network activity to complex behaviour and cognition. Recent years have witnessed huge advancements in neurotechnology for extracellular recordings, with the advent of new neural devices capable of recording from hundreds of electrodes simultaneously and to measure the activity of hundreds of neurons with very high spatio-temporal resolution (high-density multi-electrode arrays - HD-MEAs). These novel opportunities raise challenges with regards to data analyses and how to interpret results from recordings.

In this thesis I introduce new tools and analysis methods specifically targeting HD-MEA devices. The first goal of the work was to develop a unified approach using simulations to assist in the method development to address open problems in the analysis of electrophysiological data. These include spike sorting, cell-type classification, neuron localization, and selective electrical stimulation. A secondary goal of this work was to investigate to what extent the current modeling framework is capable of replicating the measured spiking activity.

In Papers I and II I present two tools to improve the process of developing and evaluating methods for spike sorting, introducing a simulator of extracellular spiking activity and a unified framework for spike sorting evaluation.

In Papers III and IV I investigate the use of independent component analysis for spike sorting of high-density multi-electrode arrays, both in an offline and an online setting.

In Paper V I introduce a method for neuron localization and cell-type classification that combines forward modeling and deep learning techniques.

Paper VI presents a model-based optimization framework for targeted extracellular electrical stimulation of single neurons from multi-electrode arrays.

Paper VII is a modeling study which investigates the effect of the neural probe on the recorded potentials.

With the work presented in this thesis, I show that computationally-assisted methods can contribute to the state-of-the-art analysis of extracellular electrophysiological data. In combination with newly developed neurotechnologies, these methods will advance our understanding of neural circuit function.
List of Papers

**Paper I**

Buccino AP, and Einevoll GT.
**MEArec: a fast and customizable testbench simulator for extracellular spiking activity.**
In: *bioRxiv* DOI: [10.1101/691642](https://doi.org/10.1101/691642) Submitted to *Neuroinformatics*.

**Paper II**

**SpikeInterface, a unified framework for spike sorting.**
In: *bioRxiv* DOI: [10.1101/796599](https://doi.org/10.1101/796599) Submitted to *eLIFE*.

**Paper III**

Buccino AP, Hagen E, Einevoll GT, Häfliger PD, and Cauwenberghs G.
**Independent component analysis for fully automated multi-electrode array spike sorting.**

**Paper IV**

Buccino AP, Hsu S-H, and Cauwenberghs, G.
**Real-time spike sorting for multi-electrode arrays with online independent component analysis.**

*These authors contributed equally to this work.*
Paper V

Buccino AP*, Kordovan M*, Ness TV, Merkt B, Häfliger PD, Fyhn M, Cauwenberghs G, Rotter S† and Einevoll GT.†
Combining biophysical modeling and deep learning for multielectrode array neuron localization and classification.
In: *Journal of Neurophysiology* **120-3** (2018), pp. 1212–1232. DOI: [10.1152/jn.00210.2018](https://doi.org/10.1152/jn.00210.2018)

Paper VI

Buccino AP, Stöber T, Naess S, Cauwenberghs G, and Häfliger PD.
Extracellular single neuron stimulation with high-density multielectrode array.
In: *2016 IEEE Biomedical Circuits and Systems Conference (BioCAS)* (2016), pp. 520–523. DOI: [10.1109/BioCAS.2016.7833846](https://doi.org/10.1109/BioCAS.2016.7833846)

Paper VII

Buccino AP, Kuchta M, Jæger KH, Ness TV, Berthet P, Mardal KA, Cauwenberghs G, and Tveito A.
How does the presence of neural probes affect extracellular potentials?
In: *Journal of Neural Engineering* **16–2** (2019), pp. 026030. DOI: [10.1088/1741-2552/ab03a1](https://doi.org/10.1088/1741-2552/ab03a1)

†These authors also contributed equally to this work.
## Contents

**Abstract** v

**List of Papers** vii

**Contents** ix

### 1 Introduction

1.1 Objectives ........................................... 4
1.2 Structure of the Thesis .................................. 5

### 2 The extracellular electrophysiology pipeline

2.1 Spike sorting ........................................... 8
2.1.1 Challenges .......................................... 12
2.2 Cell-type classification .................................. 14
2.3 Cell localization ........................................ 15
2.4 Electrical stimulation .................................... 17

### 3 Computational models

3.1 Multi-compartment models ............................... 19
3.2 Extracellular potential ................................... 21
3.3 Models of electrical stimulation ........................... 24
3.4 Finite element methods .................................. 27
3.4.1 Hybrid approach ...................................... 27
3.4.2 EMI model ........................................... 27

### 4 Engineering solutions

4.1 Independent component analysis .......................... 31
4.2 Machine learning and deep learning ....................... 35
4.2.1 The basics of artificial neural networks ............... 35
4.2.2 Deep Learning ........................................ 38
4.2.3 Convolutional Neural Networks ....................... 39
Chapter 1

Introduction

“I like nonsense; it wakes up the brain cells.”
— Dr. Seuss

The brain is arguably the most fascinating and complicated of all organs. A recent study estimates that the human brain contains around 86,000,000,000, or 8.6x10^10, or, if the reader prefers, eighty six billion neurons. If we consider other animal species, still the numbers are incredible, with chimpanzees having around 28,000,000,000, cats 760,000,000, rats 200,000,000, and honey bees around 960,000 neurons.

I have always had a hard time making a sense of millions and billions. So, the day before a talk that I was preparing, I tried to come up with an analogy, both for me and my audience, to grasp the immensity of the brain:

A tennis ball has an average diameter of 6.7 cm. An soccer pitch is 100 m long and 60 m wide, give or take. Now: imagine covering the entire football field with tennis balls (let’s assume for simplicity that the balls are cubes of side 6.7 cm). Once we are done with the first layer, we go on with the second layer, and so on. In order to place 86,000,000,000 tennis balls, we would have stacked around 64,000 layers, reaching an altitude of 4,300 m. Basically, we are on top of the Alps mountains.

†

Neurons are electrical entities. Their structure enables them to maintain an electric potential across their membrane that is modulated by incoming signals from other neurons. When the membrane potential of a neuron crosses a threshold, an action potential is generated and it is sent to all the neurons to which that neuron’s axon connects to through synapses.

Being electrical entities, one can measure neurons’ activity by inserting electrodes in the brain. When an action potential is generated, electric currents

†when I gave the talk and made this calculation I was in the United States and I ended up on top of the Mount Everest (around 8000 m). Tennis balls are probably bigger in the US.
quickly flow in and out of the neuron for around a millisecond. Extracellularly, if an electrode is close enough to a neuron firing an action potential, we observe a fast transition in the recorded electric potential, that we refer to as spike.

Recent years have witnessed an unprecedented advancement in neurotechnology. Extracellular recordings have been historically performed with single microwires or bundles of microwires (e.g. four microwires in a tetrode), capable of recording a few tens of neurons per experiment. The use of silicon manufacturing processes for neural probes, which can provide higher density and electrode counts than microwire technology, has been investigated for more than two decades. However, the main innovation has come from the integration of electronic circuits in CMOS (Complementary Metal-Oxide-Semiconductor) technology with the neural probes. These embedded electronic circuits can perform amplification and digitization of the neural signals, which results in lower noise levels (thanks to on-site amplification) and the capability of greatly increasing the electrode counts (thanks to on-site digitization, as digital transmission requires less metal wires, which are the main bottleneck for high electrode counts). CMOS-based neural probes have resulted in the prototyping of several high-density custom solutions both for in vitro and in vivo recordings. Leveraging this research, neuroscientists now have access to commercial recording probes with hundreds and thousands of electrodes, for both in vitro and in vivo applications.

High-density multi-electrode arrays, or micro-electrode arrays (MEAs), enable researchers to perform high-yield experiments, in which several hundreds of neurons can be recorded simultaneously. These devices offer many opportunities for next-generation electrophysiology (as well as challenges). Apart from being able to record the activity of many more neurons than previous techniques, the high spatial density of the electrodes enables the observation of the same action potentials on many different recording contacts. The known spatial geometry of the recording sites allows one to study neurons, from extracellular signals, even at the sub-cellular level. Figure 1.1 displays a simulation with a pyramidal cell from layer 5 in the center, a tetrode on the left, and a high-density probe on the right. The tetrode sees the spike from four contacts, with no precise knowledge of the relative electrode locations (at least for self-assembled wire tetrodes). On the high-density MEA, the same spike appears on tens of adjacent electrodes, with precise knowledge of the geometry. The amount of available spatio-temporal information provided by these probes is arguably unprecedented.

How can we exploit this high spatio-temporal information to improve the current methods used in extracellular electrophysiology?

This question is the main thread of my thesis. I have an engineering
Figure 1.1: Illustration of an extracellular recording with two types of probes: a tetrode (left) and a high-density multi-electrode array (right). The traces show a single spike recorded from the pyramidal cell in the center. The amount and richness of spatial information from the MEAs can enable one to observe even sub-cellular aspects of the action potential.\textsuperscript{28}

background and I was naturally attracted to several open questions and problems in the extracellular electrophysiology field. So, during these (almost) four years, I tried, with the help of several colleagues, to tackle them one by one.

I am still missing something: Maths. Engineers see the world as formulas and numbers, equations to solve, parameters to estimate, or, in other words, models. Fortunately, the computational neuroscience field is no disappointment.

Detailed models of single neurons, or multi-compartment models (Section 3.1), have been widely used by the neuroscience community to study various aspects of single-cell dynamics\textsuperscript{29,31}. These models represent a valuable tool for investigating and understanding neural mechanisms, and the international community has invested huge resources in building and sharing a large variety of detailed neuronal models. As an example, the Blue Brain Project has constructed over 30’000 cell models from the somatosensory cortex of juvenile rats, with the final goal to digitally reconstruct the neocortical microcircuit\textsuperscript{32–35}. A similar effort is being

\textsuperscript{*}Yes, engineers do like to solve problems.
1. Introduction

conducted by the Allen Institute for Brain Science, whose cell-type database\cite{36} contains several hundred of cell models from mice and even humans\cite{37}. Moreover, we have a good understanding of how currents generated by neurons translate to recorded extracellular potentials\cite{38–42}.

Great, now everything is in place: an enthusiastic (now) neural engineer, advanced models of neurons, and electrophysiology problems to solve.

1.1 Objectives

The main objective of this work is to combine simulations and engineering methods to tackle several aspects of the electrophysiology pipeline (Chapter 2) for high-density MEAs. Given the availability of biophysically detailed models, that may pass a biological Turing test\cite{43}, the central idea of this project is to use modeling and simulations to assist the development of methods for electrophysiology. These techniques include spike sorting, cell localization and classification from extracellular action potentials, and targeted electrical stimulation from extracellular probes (Figure 2.1 in Chapter 2). Moreover, since a great part of this work uses simulations to drive method development, a secondary goal is to investigate to what extent the conventional modeling framework is accurate and trustworthy and what are its limitations.

While the order of the papers presented in Chapter 5 follows the natural electrophysiology pipeline (Figure 2.1), I present here the objective of the papers in a quasi-chronological order, to highlight why and how the main project evolved and the different sub-projects were conceptualized.

**Paper VI** *How can we exploit the high density of MEAs to make electrical stimulation more selective?*

The goal of this paper was to improve the selectivity of extracellular stimulation leveraging the high density of the electrodes, a model-based approach, and optimization techniques. We assumed that we knew the location of the neuron with respect to the probe.

**Paper V** *Can we accurately estimate the position of a neuron from its extracellular signals?*

The objective of this article was to improve the performance of neuronal localization and classification from extracellular action potentials by combining forward biophysical modeling and deep learning techniques. In this approach, we assumed that extracellular simulations of single spikes are biophysically accurate enough to be used as ground-truth information for training machine learning algorithms. Moreover in this contribution we assumed that inference was performed after spike sorting, on average waveforms.

**Paper VII** *Are our estimates of extracellular signals accurate enough?*

The aim of this article was to investigate to what extent the neural probes we use to record neural activity affect the recorded signals. In order to do so, we
used finite element methods. Moreover, we wanted to implement a more efficient method to include the presence of the probes in the calculation of extracellular potentials.

Paper III  What about spike sorting for high-density MEAs?
In this article we aimed to investigate the use of independent component analysis (ICA) for spike sorting data from high-density MEAs. Moreover, we wanted to build a fully automatic pipeline and compare its performance with state-of-the-art algorithms. In order to benchmark the algorithm, we needed to develop a simulator of ground-truth spiking activity, wrappers for some existing spike sorters, and automated comparison routines.

Paper IV  What about real-time spike sorting for high-density MEAs?
The goal of this paper was to investigate an online version of ICA, namely online recursive ICA – ORICA, for real-time spike sorting. We aimed to benchmark the accuracy of the ORICA model in finding spiking sources within real-time time constraints.

Paper I  Can we develop a simulator of extracellular activity to aid spike sorting development?
This question arose from Paper III and Paper IV, in which I felt the need to include ground-truth simulations in the development of spike sorting algorithms. The objective of this paper was to create a simulator of extracellular activity and make it an accessible and usable software package that could be used by the spike sorting community. We aimed to develop a fast, easy to use, highly controllable, and biophysically detailed simulator. Additionally, we wanted the simulator to be able to reproduce features of extracellular signals that are critical for spike sorting, such as bursting, drifting, and spatio-temporal overlapping spikes.

Paper II  How can we benchmark and compare several spike sorters?
This question as well has its origins in Paper III and Paper IV, from the tedious and time-consuming process of running several spike sorters and comparing their outcome. The goal of this paper was to build a unified software framework for spike sorting. I teamed up with international collaborators in the spike sorting community to create an open-source software to make the spike sorting process easy and accessible, even to those with very little programming background. Moreover, we wanted to include tools for the entire electrophysiology pipeline, including processing, quality control, and curation tools, as well as a comparison framework for both ground-truth and non ground-truth data.

1.2  Structure of the Thesis
The thesis is organized in three background chapters, a summary of the papers (Chapter 5), and a final discussion that contextualizes the work, highlighting its
1. Introduction

limitations and future directions (Chapter 6).

In Chapter 2 I cover the electrophysiology pipeline, with a brief background and state-of-the-art methods for the different steps involved in the interaction between neural tissue and extracellular devices. Chapter 3 covers the basics of the computational models that I used throughout my thesis, from multi-compartment models, to finite element methods. Finally, Chapter 4 introduces the engineering solutions employed in most of the projects, focusing on independent component analysis, machine learning and deep learning, optimization strategies and genetic algorithms, as well as the use of Python for scientific software development.
Chapter 2

The extracellular electrophysiology pipeline

“You are nothing but a pack of neurons.”
— Francis Crick

When we implant a neural probe in the brain, it can be used to bi-directionally interact with the neural tissue that surrounds it. On one hand, electrodes can record the activity of neurons in the neighborhood, on the other end they can be used to inject currents in the tissue and stimulate neurons.

A few steps are required to go from raw recordings to selective stimulation of neurons. I will refer to these steps as the electrophysiology pipeline.

A representation of the electrophysiology pipeline is displayed in Figure 2.1. The neural probe inserted in the brain tissue is surrounded by neurons (A). The electrodes on the probe record the electric potential on their surface. Neural activity produces both low frequency signals, named local field potentials (LFPs), and higher frequency components, which include the spiking activity. In this work, I will focus on the latter, assuming that the recordings (B) are already filtered with a high-pass filter. The signals recorded extracellularly contain a mixture of activity from several neurons. Hence, the first processing step required in the pipeline is to identify, or sort, the activity of individual neurons. This step is called spike sorting (C). After spike sorting, one can focus on single neurons separately and use their extracellular signature, or extracellular action potential (EAP), to extract further information from the recordings. For example, one can classify if the recorded neurons are excitatory or inhibitory (D), which is important when untangling the structure and functions of neural microcircuits. In addition, EAPs can be used to triangulate or localize the 3D position of the neuron with respect to the recording electrodes (E). The locations and type

†not all probes allow for electrical stimulation, as stimulation requires dedicated electronic circuitry.
‡as well as glia cells, but let us focus on neurons for this thesis.
of the recorded neurons can be used to selectively target a specific cell with electrical stimulation (F).

### 2.1 Spike sorting

Spike sorting is the procedure used to identify which spikes belong to the same neuron. The extracellular recordings contain in fact a mixture of activity of several neurons. The relative position between each recorded neuron and the probe, as well as the neuron morphology and electrical properties, results in different spike shapes on the electrodes. Spike sorting attempts to separate different waveforms and it outputs a set of neurons and their spike times.

Historically, spike sorting has been performed by hand. In manual spike sorting, waveforms are first detected by thresholding and aligning the signals; second, waveforms are represented in lower dimensions (either using dimensionality reduction techniques or considering the amplitude of the spikes) and the electrophysiologist manually isolates (or cuts) clusters, i.e., tries to identify separate groups. Manual clustering has two main limitations: first it could undermine the reproducibility of the data, due to subject variability; second, it is not scalable to high-density probes. Manual clustering could be manageable, in terms of time, for up to a few tens of electrodes, but it quickly becomes extremely hard, if not impossible, when the number of electrodes reaches...
larger numbers.

In recent years, several semi-automatic and automatic spike sorters have been developed to alleviate these problems. Spike sorting algorithms attempt to separate spike trains of different neurons (also called units) from the extracellular mixture of signals using a variety of different approaches. Although automatic spike sorting methods were first developed in the 80’s, in recent years, probably due to the above-mentioned limitations of manual spike sorting, there has been a boost in the development and sharing of spike sorting software solutions.

In the following paragraphs, I will briefly introduce the principles of the different approaches to spike sorting. While the most widely used strategies are clustering-based and template-matching, I will also introduce a spike sorting strategy based on independent component analysis (ICA - Section 4.1), because I have used it for Paper III and Paper IV.

**Clustering-based approaches**  Clustering-based approaches are inspired by the manual clustering steps described above (Figure 2.2A). Filtered signals are first thresholded to detect putative spikes. The detected waveforms are aligned and projected on lower dimensional space, using for example principal component analysis (PCA) or wavelet features, and then automatic clustering algorithms are used instead of manual cutting.

While some solutions focus on low-channel count probes, mainly used on epileptic human subjects, other approaches attempt to scale up the algorithms for higher channel counts. Clustering-based approaches can suffer from over-splitting of units, i.e., the cluster corresponding to a single unit is split into multiple clusters. An automatic curation step is therefore applied by some algorithms to merge putative over-split units. In addition, some solutions have mechanisms to correct for drift. Others use an estimate of the spike location, computed for example as the center of mass of the peak amplitude, as a clustering dimension, in order to improve the clustering performance.

One of the main and well-known problem of clustering-based approaches is overlapping spikes. While events that are overlapping in time, but not in space, can be handled by considering the spatial location of the detected waveform or by applying a spatial mask to the feature set, the occurrence of spatio-temporal overlapping events can result in a distortion of the waveforms. This is due to the spatial summation of two (or more) waveforms, and may confuse the clustering algorithm. In order to alleviate this problem, template-matching solutions have been proposed.

**Template-matching approaches**  Although the first template-matching solutions predate clustering-based approaches, these methods have recently undergone a renewed interest, as high-density probes greatly increase the occurrence of spatio-temporal overlapping events.

*drift occurs when there is a relative movement between the recorded neuron and the probe, which causes a change in the spike waveforms over time
2. The extracellular electrophysiology pipeline

**A Clustering-based**

**B Template-matching**

**C ICA-based**

Figure 2.2: Principles of the different spike sorting approaches. The 6-channel recordings are simulated and they contain activity of three neurons. (A) Clustering-based approach. The recordings are thresholded and the detected waveforms are projected onto a lower dimensional space and clustered using automatic algorithms. (B) Template-matching approach. After a pre-clustering step, in which a subset of waveforms is clustered as described in (A), the templates of the different clusters are projected back to the time domain and recursively matched to the recordings (matching) (C) ICA-based approach. The recordings are processed with ICA, and the detection and optionally clustering is performed on the independent sources, which are tuned to separate neurons.

Template-matching approaches in general assume that the recorded signals can be described as:

$$r(t) = \sum_{ij} a_{ij} T_j(t - t_i) + e(t)$$  \hspace{1cm} (2.1)

where $r(t)$ is the set of recorded signals, $T_j(t - t_i)$ is the template associated to a neuron $j$ (firing at times $t_i$), $a_{ij}$ is a scalar that scales the the amplitude of the template for each spike event, and $e(t)$ is additive noise.

Template-matching methods try to identify templates for each neuron, and then reconstruct the signals following Eq. $2.1$ (Figure $2.2$B). To achieve this,
templates are first estimated by running a pre-clustering step, which corresponds to a clustering-based approach on a subset of detected waveforms. The estimated templates are then matched to the recordings $s(t)$ recursively: if a good match is found, the template is subtracted from the recording, in order to disclose underlying overlapping spikes.

Template-matching solutions mainly differ in terms of the clustering step for finding the initial templates, the matching algorithm, and automatic curation steps. Finally, some of the latest algorithms attempt to handle drifting recordings.

**ICA-based approaches** A third and less explored possibility for spike sorting involves the use of ICA (see Section 4.1 for a detailed description of the algorithm). ICA is an unsupervised blind source separation method that aims to find projections that make the signals more statistically independent.

Although ICA has been suggested to spike sort tetrode signals, in combination with clustering to increase the data dimensionality, and dodecatrode recordings (12 microwires), the main limitation of ICA has been its assumption that the number of sources, i.e., neurons, is less than the number of electrodes, which is not true for tetrode and dodecatrode recordings. The development of high-density MEAs, however, could satisfy this assumption. The use of ICA and some of its variants for spike sorting of high-density probes has in fact been suggested more recently.

The principle of an ICA-based spike sorting pipeline is displayed in Figure 2.2C. Instead of detecting spikes or finding templates, the signals are first processed with ICA. The ICA step outputs a set of unmixing matrices. The projections of the initial signals on these matrices are called independent components (IC) or sources. Assuming that the signals from different neurons are fairly independent, each IC source should be tuned to a separate neuron. Detection and optionally clustering can be then performed directly on the IC sources, and this process can also be applied recursively, similarly to template-matching approaches. ICA-based spike sorting approaches, however, have not been fully benchmarked and compared to other available solutions for high-density MEAs, and this is the main motivation for Paper III.

With the assumption of statistical independence of signals coming from different neurons, synchronous activity could be problematic. However, this issue would be an actual problem if the spiking activity of different neurons were perfectly coincident. Due to intrinsic noise and randomness in the spiking activity, this is not fortunately the case. A second potential limitation of ICA-based spike sorting is the assumption of linearity between the sources and the signals (see Section 4.1 and Eq. 4.1), i.e., the recorded signals are assumed to be an instantaneous mix of the sources. Due to the filtering properties of dendrites, there is a phase shift between extracellular spikes recorded in the vicinity of the soma and ones closer to the dendrites. This assumption might result in finding more than one source tuned to the same neuron (duplicate sources). A

*usually defined as an improvement in a cost function
possible solution to this issue is to use convolutive ICA (cICA) instead of the standard ICA model. Using cICA, the recordings are described as a filtered version of the sources. However, cICA algorithms are much more complicated and slower than instantaneous ICA approaches. In order to tackle this problem, in Paper III, we perform a post-processing step on the identified spike trains to find and remove duplicates. Finally, with the increasing number of channels in the recording devices, ICA could be a too computationally intense processing step. Nevertheless, the performance of ICA could be easily improved by applying separate ICA models to subsets of adjacent electrodes in parallel, or to estimate the ICA model using a subsample of the data instead of all the recorded signals, assuming that the statistical properties are the same.

2.1.1 Challenges

Spike sorting is a very important step in extracellular electrophysiology, but despite the huge development over the past years, there are still some open challenges.

**Spike sorting validation** Spike sorting is unsupervised in nature, since when recording from extracellular probes ground-truth information about the underlying spiking activity is not available.

In order to validate the spike sorting output without ground truth, several quality control metrics have been proposed to assess the goodness of sorted units. Some of these metrics quantify biophysical properties of the sorted units, such as refractory violations and waveform amplitude distribution. Others more generally quantify the isolation of the clusters with several indicators, i.e., isolation distance, L-ratio, linear discriminant analysis classification ($d'$), or nearest neighbors. Other metrics related to noise distribution have been proposed, as well as stability measures.

While these metrics are useful to characterize spike sorting performance, they are also unsupervised and they require users to empirically set “good” values. Alternatively, several groups have tried to simultaneously record units both extracellularly and using patch-clamp or juxtacellular recordings. These sets of ground-truth recordings are extremely valuable for validating spike sorting algorithms. However, their main limitation is the low yield of the experiments (only one or a few cells can be patched simultaneously), resulting in a limited validation capability.

A third validation approach relies on the use of simulated data. This approach can range from injecting simulated waveforms to real recordings, a so-called hybrid approach, to the generation of entirely simulated recordings. The latter approach provides full ground-truth information and it can therefore help to benchmark spike sorting algorithms. However, the question of how realistic these simulated recordings is still open.

Paper I and Paper II try to alleviate the problems related to spike sorting validation. In Paper I, we present a Python-based simulator of extracellular spiking activity, called MEArec. MEArec, with its easy-to-use
design, speed, controllability, and capability of reproducing challenging properties of extracellular recordings, can provide high quality ground-truth recordings for spike sorting development and evaluation.

In [Paper II] we introduce SpikeInterface, an open-source and unified framework for spike sorting. SpikeInterface is a powerful and comprehensive software capable of loading several file formats, running a multitude of available spike sorters, and can validate and compare the spike sorting output. Within SpikeInterface, users can compute several quality metrics available in the literature for unsupervised validation, as well as compare spike sorting output with ground-truth recordings, for example generated by MEArec.

**Challenging aspects of extracellular recordings** Some intrinsic aspects of spiking activity and extracellular recordings can be challenging for spike sorting algorithms.

One of the main challenges can be identifying bursting units. When neurons burst (i.e., they fire quick sequences of action potentials), the underlying dynamics of the spike generation change, causing a modulation of the spike shape, with lower amplitudes and wider waveforms. As a result, bursting neurons are harder to correctly identify, and may be over-split. However, bursting neurons can be identified *a posteriori* from their cross-correlograms, and merged using manual curation software. Some algorithms also implement specific steps to identify and automatically merge bursting units.

A second challenge is the occurrence of spatio-temporal overlapping spikes, which can distort the recorded waveforms. However, the use of template-matching approaches and the higher spatial density of probes should alleviate this problem.

A third complicated feature of extracellular recording is drift. Drift occurs when there is a relative movement between the probe and the neural tissue. Drift can result from the slow relaxation of the tissue after a probe insertion (slow drift), or from movement artifacts that cause a quicker shift of the tissue with respect to the probe (fast drift). Some recent algorithms were specifically designed to identify and correct for drifting artifacts.

The presence of noise artifacts can also be challenging for spike sorting algorithms, especially when recording from freely moving animals. Artifacts can be related to shocks, mastication, and grooming and they are usually detected as spikes because of their large amplitude. However, artifact clusters could be automatically rejected based on the above-mentioned quality metrics. Alternatively, due to the large availability of manually curated spike sorting outputs in which artifact-clusters are rejected, another possibility is to build machine learning systems to automatically reject noisy clusters.

In order to provide a faithful validation of spike sorting algorithm, the MEArec simulator [Paper I] is capable of reproducing bursting behavior, drift, and to precisely control the occurrence of spatio-temporal collisions between spikes.
2. The extracellular electrophysiology pipeline

**Performance**  With very high density probes now reaching a channel count of several hundreds/thousands of electrodes\(^{19, 15-20, 24}\), and likely to reach several thousands of simultaneously recorded channels in the near future, scalability is an essential requirement to keep in mind. Strikingly, an algorithm developed in 2016 and classified as for “large and dense” probes (up to 64 channels), has become outdated in less than 4 years\(^{51}\).

Recent methodologically sophisticated and innovative solutions, have been developed for highly parallel clusters\(^{53, 54, 58, 59}\) and for graphical processing unit (GPU) accelerated hardware\(^{52, 57, 60}\).

**Online spike sorting**  While most of the available algorithms are designed for offline use\(^{51-54, 57-59}\), the capability of detecting and sorting spikes online is very powerful for closed-loop intervention with the neural tissue\(^{83}\).

While a few solutions have been suggested for online spike sorting of low-channel probes\(^{63, 84}\), the extension to high-density probes is not trivial.

The matching phase of template-matching approaches, after an offline estimate of the templates, can be applied online\(^{85-87}\). However, the non-stationarity of the signals, due to drifts or non-stationary activity, may require a re-estimation of the templates over the course of an experiment\(^{85}\).

Alternatively, adaptive algorithms could be developed to track non-stationarities of the signals. In Paper IV, we introduce an online spike sorting method based on adaptive ICA\(^{88-90}\), that has the potential to track non-stationarity of the signals in real-time. However, further validation for this approach is required.

2.2 Cell-type classification

In order to understand how the brain works as a whole, we first need to identify the roles of its components. To this end, the characterization of neuronal cell types based on extracellular recordings is essential to study the function of different neurons in computations.

Cell types can be defined by several different aspects, including gene expression, morphology, electrical properties, and connectivity\(^{32, 67, 91, 92}\). However, the main distinction between cell types happens at the connectivity level: neurons that elicit an excitatory post-synaptic potential (EPSP) are said to be **excitatory**; conversely, neurons that cause an inhibitory post-synaptic potential (IPSP) are classified as **inhibitory**. Excitatory and inhibitory cells have very distinctive roles in neural circuits. Excitatory cells mainly provide local recurrent and long-range projections\(^{93}\), and inhibitory cells play a modulatory and balancing role\(^{94}\). It is therefore important to be able to identify these different cell types from their extracellular signatures, in order to better understand the underlying neural mechanisms.

When measuring from extracellular electrodes, one can **putatively** classify cells as being excitatory or inhibitory. Based on the extracellular action potential shape, units are usually classified as regular spiking (RS) or fast spiking (FS)
RS cells exhibit a broader action potential and are regarded as excitatory pyramidal cells; FS cells have a narrower waveform and are generally considered to be inhibitory neurons. The separation of these two classes has historically been performed by extracting waveform features from the spike sorted spikes, and then applying unsupervised clustering techniques. Some of the commonly used features computed from the waveforms include the 1) trough-to-peak width, 2) full-width half maximum, 3) half-amplitude duration, and 4) peak-amplitude asymmetry. Moreover, properties derived from the spike statistics such as firing rate, auto-correlogram shape, inter-spike-interval distributions, and bursting activity can exhibit differences between the two cell-types.

The above-mentioned methods mainly consider the waveform recorded on the electrode with the largest spike amplitude. However, the use of high-density MEAs can provide much more spatial information that can be used to refine and improve the classification. For example, in a recent study, several features were computed from the spatio-temporal signature of the waveforms. The authors showed that high-density silicon probes, in their case Neuropixels, allow for the tracking of backpropagating action potentials, which are unique to pyramidal cells. Moreover, the use of the rich spatial information enabled the authors to identify two separate classes of RS neurons in visual cortex. In Paper V, we show that the use of features extracted from high-density probes improves the classification accuracy over conventional clustering approaches.

The validation of cell-type classification is challenging, but there are a few solution. When measuring the activity of multiple cells from the same region, some of those might be mono-synaptically connected. In that case, the analysis of the cross-correlograms can suggest whether a cell is excitatory, inhibitory, or even if there is reciprocal excitatory-inhibitory interaction between a pair of cells. However, the rate of monosynaptic connections in recorded neurons is usually very low (~0.2%). Another viable alternative for the validation of cell-type classification is the use of optotagging. This technique consists of using optogenetics to target specific sub-populations of neurons. Optogenetics allows one to make neurons express ion channels which can be activated by light (opsins). Using genetic techniques or transgenic lines, these light-sensitive channels can be expressed only in sub-populations of neurons. Shining light with wavelengths matching the activation spectrum of the opsin will activate only neurons belonging to the tagged, or labeled, cell types. However, optogenetics adds another invasive component to the recording approach, as it requires transgenic lines or viral injections to express the light-sensitive channels. Moreover, the presence of these additional ion channels might affect the cell dynamics, hence slightly changing the recorded waveform.

### 2.3 Cell localization

The recorded extracellular potentials can also be used to localize or triangulate the position of a neuron with respect to the probe. The capability to reconstruct
2. The extracellular electrophysiology pipeline

neuron locations can shed light on the microscale organization of neural circuits\textsuperscript{13}. Moreover, precise cell localization could improve targeted stimulation\textsuperscript{102} and automatic positioning of recording electrodes for patch-clamp experiments\textsuperscript{103}.

Localization from extracellular potentials is inherently an ill-posed problem, because of the ambiguity arising from measuring the electric potential in only a limited number of locations when neuronal currents are generated from complex neuronal morphologies. While first approaches for neuron localization used tetrodes\textsuperscript{104,105} or low-density polytrodes\textsuperscript{13,106,107} the development of high-density MEAs can provide much higher spatial resolution, improving localization accuracy\textsuperscript{103} and even enabling tracking sub-cellular mechanisms, such as axonal propagation\textsuperscript{108} and back-propagating action potentials\textsuperscript{99}.

Most of the methods developed for localization attempt to solve the inverse problem. Given the set of recorded potentials $V_r$ (usually observed at the spike peak\textsuperscript{14,109,110}) and a forward model $F$ that describes the spike generation, the solution of the inverse problem tries to estimate the soma position $(x_s, y_s, z_s)$ that minimizes the error between the recorded potentials $V_r$ and the potentials predicted by the forward model $V_f$. Mathematically, the inverse problem can be formulated as:

$$V_f = F(x_s, y_s, z_s, P)$$

$$\arg\min_{x_s, y_s, z_s, P} (V_f - V_r)^2$$

(2.2)

where $P$ is the set of additional parameters to $x_s, y_s, z_s$ of the model $F$\textsuperscript{11}.

The models chosen for the spike generation are usually very simple, in order make the solution unique. Examples of models used in previous studies are monopole current-sources\textsuperscript{13,109,110}, dipole current-sources\textsuperscript{13,104,105}, multi-pole current-sources\textsuperscript{106}, as well as line-source models\textsuperscript{107}, and more recently ball–and–stick model\textsuperscript{111}.

One of the main limitations of solving the inverse problem to tackle neuron localization is that the models chosen to solve the inverse problem are often too simple too grasp complex spike waveforms (e.g. monopolar or bipolar current-source models) or they are tuned to a specific cell types (ball–and–stick model for the pyramidal morphology\textsuperscript{111}). Therefore, in \textsuperscript{Paper V} we suggest a supervised method in which detailed simulations are used as ground truth to train deep learning models.

A challenging aspect of neural localization is arguably its validation. It is in fact experimentally very challenging to accurately measure the correct position of the soma with respect to an extracellular device\textsuperscript{74,75}; therefore, detailed computational neuronal models are usually used and treated as ground truth to evaluate the accuracy of the localization method\textsuperscript{106,107,111}. An alternative mean for validation is to combine extracellular recordings and imaging\textsuperscript{112} to precisely co-localize the cells and validate localization performance.

\textsuperscript{*}for example, for a monopole current source model $P$ corresponds to the value of the monopole current.
Finally, most of the above-mentioned approaches aim to identify the neuronal soma. However, recent experimental and computational findings suggest the axon initial segment (AIS), not the soma, is the main contributor to the extracellular action potential. As the AIS can be tens of µm away from the soma, this discrepancy should be taken into account in developing localization methods.

2.4 Electrical stimulation

Neurons are excitable cells. Almost by definition, this means that they can be excited and one way of doing so is to use electrical stimulation from extracellular electrodes.

Stimulation of neural tissue has been successfully used for decades in several biomedical applications, including cochlear implants for hearing restoration, retinal implants for vision improvements, and deep brain stimulation (DBS) to treat Parkinson’s disease. Moreover, electrical stimulation is used to repair sensory perception in paralyzed patients, and to restore walking in tetraplegic patients.

Another interesting application of electrical stimulation is in closed-loop settings to facilitate neuroplasticity. Following the Hebbian principle “neurons that fire together, wire together”, if a neuron is stimulated whenever another neuron connecting to it, directly or indirectly, fires, their synaptic connection is strengthened. This idea has inspired several studies to, for example, rewire motor neural connections or interact with spatial memories.

The application of extracellular potentials at a neuron’s membrane can depolarize or hyperpolarize the membrane potential. If the depolarization is strong enough, this can trigger action potentials (see Section 3.3 for a modeling perspective). A multitude of experimental studies over the last century have characterized the key aspects of electrical stimulation for neural excitation:

- **Excitable regions**: the most excitable parts of a neuron, and therefore the most likely to be activated by electrical stimulation, are the axon initial segment and the nodes of Ranvier, where the density of sodium channels is the highest.

- **Current-distance relation**: the amount of current required to elicit an action potential (threshold current) in a neuron is proportional to the square of the distance from the electrode tip when simulating with a monophasic constant pulse.

- **Strength-duration curve**: using similar stimulation settings, the threshold current is related to the pulse duration by

\[
I_{th} = I_{rh} \left( 1 + \frac{t_{ch}}{d} \right)
\]
2. The extracellular electrophysiology pipeline

where \( I_{rh} \) is the rheobase current, i.e., the minimum current value capable of eliciting an action potential; \( d \) is the pulse duration; and \( t_{ch} \) is the neuron chronaxie, which is the pulse duration at \( 2I_{rh} \).

- **Charge balance**: electrical stimulation is usually applied by means of charge balanced biphasic current,\(^{129}\) or balanced voltage pulses,\(^{102}\) in order to preserve the safety of the tissue and the electrode-tissue interface.\(^{130}\)

Most of what we know about electrical stimulation comes from experimental studies. More recently, several groups have investigated the effect of electrical stimulation by means of computational studies, both using analytical approaches,\(^{126,131-134}\) and more advanced finite element methods.\(^{135-139}\) Theoretical and numerical simulations can provide a controlled framework to investigate the effects of electrical stimulation.\(^{127}\)

Despite the widespread use of electrical stimulation in the clinical and academic fields, there are still some open challenges and limitations.

The first limitation is physical: when a current is injected in the brain, i.e., a conductive medium, it spreads radially in all direction. This phenomenon can result in the activation of a larger area (and more neurons) than desired.\(^{124,127,140}\) This can clearly be a problem for applications that require targeted stimulation, such as the above-mentioned closed-loop neuroplasticity experiments as well as visual and auditory prostheses applications. The use of high-density probes can in part alleviate this problem. In vitro studies have in fact shown that it is possible to selectively target specific regions of a neuron.\(^{102,141}\) In Paper VI, we exploit the high spatial density of MEAs for optimizing targeted spatial patterns to increase stimulation selectivity.

The second challenge is more related to technology and electronics. In order to apply electrical stimulation from the electrodes, neural probes require specialized stimulation circuits,\(^{18,28,102}\) which occupy space and consume power. Although the integration of such circuits is common for in vitro applications,\(^{15,17,18}\) their inclusion in active CMOS probes designed for in vivo experiments is still limited due to size and tissue heating constraints. Alternatively, extracellular stimulation in vivo is possible using passive electrodes (both silicon probes or microwires) and external systems capable of simultaneous recording and stimulation.\(^\ast\)

\(^\ast\) e.g. [http://intantech.com/stim_record_controller.html](http://intantech.com/stim_record_controller.html)
Chapter 3

Computational models

“All models are wrong, but some are useful.”
—George Box

In this chapter, I will present a brief introduction of the computational models used in this thesis, with their main principles, applications, and limitations.

3.1 Multi-compartment models

Multi-compartment models are representations of a neuron as an electric circuit. Neuron morphologies, extracted from microscopy or built using simplified assumptions, are modeled by many small segments and represented as electrical circuits. Each segment implements a membrane model, which consists of several ion channels represented as conductance-based models.

Conductance-based models describe the biophysical properties of an excitable membrane. In this model, the contribution of an ion channel type is represented as an overall conductance. For example, for an ion channel $x$ (e.g. sodium, potassium, calcium), the current density flowing out of the membrane is:

$$i_x = g_x(V - E_x)$$  \(3.1\)

where $i_x$ is the ionic current density generated by the $x$ ion \([mA cm^{-2}]\), $E_x$ is the reversal potential for the ion, i.e., the potential at which electric and diffusion currents are balanced, $V$ is the membrane potential (the difference between intracellular and extracellular potential: $V = \phi_{in} - \phi_{ex}$), and $g_x [mS cm^{-2}]$ is the ion channel conductance per unit of area.

Conductances can be passive (constant), representing leaky currents that traverse the membrane, or active. The leak current density is usually defined with respect to the resting potential $V_r$, i.e., $i_l = g_l(V - V_r)$. Active channels enable to reproduce the highly non-linear dynamics of neurons, such as the

†Note that in this derivation all spatial units are assumed to be in cm
3. Computational models

generation of an action potential. Famously, Hodgkin and Huxley were able to quantitatively reproduce an action potential by modeling active Na\textsuperscript{+} and K\textsuperscript{+} channels. For active channels, the conductance is dependent on the membrane potential ($g_x = g_x(V)$). The membrane of an excitable cell can be represented as a capacitor, as the lipid bilayer that it is made of keeps ions apart, similarly to a capacitance. The equation governing the current density of a single-compartment can be written using Kirchoff’s current law:

$$c_m \frac{dV}{dt} = - \sum_x g_x(V - E_x) - g_l(V - V_r) + i_{ext} = -i_{ion} - i_{leak} + i_{ext} \quad (3.2)$$

where $c_m \ [mF \ cm^{-2}]$ is the membrane capacitance, $i_{ext} \ [mA \ cm^{-2}]$ is an external current density entering the membrane, for example a synaptic input or a stimulation current. Single-compartment models do not include morphological information as they represent a so-called point neuron (Figure 3.1A). In order to build a morphologically detailed version of a neuron, multi-compartment models can be used.

Multi-compartment models consist of a set of connected single-compartment models. Adjacent compartments, or segments, are connected via internal resistors, which model the resistance of the intracellular cytosol. While single-compartment models are governed by a single equation (Eq. 3.2), when assembling multiple compartments the set of equations become coupled.

Considering a non-branching neuronal segment and assuming a constant cylindrical section (Figure 3.1B), a compartment length of $\delta x$, and an axial conductivity of $\sigma_i \ [mS \ cm^{-1}]$, the current densities between compartments $i - 1$ and $i$, and between compartments $i$ and $i + 1$ can be written as:

$$i_{i-1,i} = \frac{\sigma_i}{\delta x} (\phi_{in,i-1} - \phi_{in,i})$$  
$$i_{i,i+1} = \frac{\sigma_i}{\delta x} (\phi_{in,i} - \phi_{in,i+1}) \quad (3.3)$$

Applying Kirchoff’s current law, we can relate the transmembrane current density flowing out of the $i$-th compartment $i_i$ to the axial currents from adjacent sections (for simplicity the additive external current $i_{ext}$ is neglected):

$$I_i = A_m \left[ c_m \frac{\partial V}{\partial t} + i_{ion} + i_{leak} \right] = A_c (i_{i,i+1} - i_{i-1,i})$$

$$= A_c \left[ \frac{\sigma_i}{\delta x} (\phi_{in,i-1} - \phi_{in,i}) - \frac{\sigma_i}{\delta x} (\phi_{in,i} - \phi_{in,i+1}) \right]$$

$$= A_c \frac{\sigma_i}{\delta x} (\phi_{in,i+1} - 2\phi_{in,i} + \phi_{in,i-1}) \quad (3.4)$$

where $A_c = \pi/4 \ d^2$ is the cross section area and $A_m = \pi d \delta x$ is the membrane area of the compartment, and $d$ is the diameter of the neurite. Differently from Eq. 3.2 the derivative notation is partial because $V$ is now function of time and
space. Let now $C_m = c_m A_m$ be the membrane capacitance, $G_a = A_c \sigma_i/\delta x$ the axial conductance, $I_{ion} = A_m i_{ion}$ the ionic current, and $G_l = A_m g_l$ the leak conductance of the membrane. When $\delta x \to 0$, the right hand-side of Eq. 3.4 becomes the second spatial derivative of $\phi_{in}$. Since $\phi_{in} = V + \phi_{ex}$, we can reformulate as:

$$C_m \frac{\partial V}{\partial t} + I_{ion} + G_l (V - V_r) = G_a \frac{\partial^2 (V + \phi_{ex})}{\partial x^2}$$

(3.5)

where the dependence of $V$, $I_{ion}$ and $\phi_{ex}$ on $(x,t)$ is not shown for clarity. To further simplify, we can rewrite with respect to $V_m = V - V_r$, define $\tau = C_m/G_l = c_m/g_l$ (membrane temporal constant) and $\lambda = \sqrt{G_a/G_l \delta x} = \sqrt{(d\sigma_i)/(4g_l)}$ (membrane spatial constant). Finally, we can assume that the extracellular potential is constant (and therefore its second derivative is null):

$$\tau \frac{\partial V_m}{\partial t} - \lambda^2 \frac{\partial^2 V_m}{\partial x^2} + i_{ion} g_l + V_m = 0$$

(3.6)

This partial differential equation (PDE) is the cable equation and it is a fundamental equation in computational neuroscience. While Eqs. 3.5 and 3.6 assume a homogeneous neurite, axons can be covered by myelin sheets to increase the transmission speed. In this case, the continuous solution can be substituted by a discrete solution, in which the values of the membrane capacitance and resistance are fit to the properties of myelinated sections and nodes of Ranvier.

Importantly, the cable equation assumes the extracellular resistance to be 0 (in Figure 3.1 there is in fact no resistor between the extracellular nodes), i.e., the extracellular potential to be constant. This is done mainly for two reasons: on one hand the intracellular resistance is much larger than the extracellular one; secondly, assuming a null extracellular resistance simplifies extensively the solution of the cable equation. However, when a non-null resistance is incorporated, ephaptic effects, i.e., the contribution of the extracellular potential to neuronal dynamics, can be studied.

Software The solution of the cable equation for complex morphologies and conductances is far from trivial, especially considering the highly non-linear dynamics of ion channels. Therefore, simulators have been developed, such as NEURON and GENESIS, with high-level definition languages and graphical user interfaces. Moreover, NEURON has a Python Application Programming Interface (API). NEURON, through the LFPy environment introduced in the next section, has been used extensively throughout this thesis.

3.2 Extracellular potential

After computing the internal dynamics of the neuron, we need to compute what is actually measured by neural probes, i.e., the extracellular potential.
3. Computational models

Figure 3.1: (A) Point neuron model. The membrane is modeled with a capacitance, a leaky resistor, and active ion channels represented as voltage-dependent resistors. (B) Multi-compartment model. A neuron is split in small compartments. Each compartment implements a membrane model and it is connected to adjacent compartments with resistors that model the intracellular cytosol. (C) Extracellular potentials. The extracellular potential $\phi_{ex}$ is computed as the sum of transmembrane currents’ contributions. (The drawing of the pyramidal cell is by Federico Claudi, and it is available at scidraw.io).

From the solution of the cable equation, which is the set of membrane potentials for each neuronal compartment $V_i(t)$, one can easily obtain the transmembrane currents $I_i$ for each compartment (Eq. 3.4).

One of the most commonly used and computationally efficient way to compute the extracellular potential generated by neurons uses volume conduction theory\[151]\. Considering a quasistatic approximation of Maxwell’s equations and assuming a conductive, isotropic, homogeneous, linear, and infinite medium, the extracellular potential generated by a point current source $I_s(t)$ at position $r_s$ can be computed at any point $r$ (except for $r_s$) as:

$$\phi_{ex}(r,t) = \frac{I_s(t)}{4\pi \sigma |r - r_s|}$$  (3.7)

In this case ground ($\phi_{ex} = 0$) is set far away from the current and $\sigma$ is the conductivity of the medium [$mS cm^{-1}$]. Straightforwardly, one can
then consider each transmembrane current $I_i$ as a point source located at the center of its neuronal compartment $r_i$, and compute the extracellular potential measured by an electrode in position $r_e$ as a linear sum of the individual current’s contributions:

$$\phi_{ex}(r_e, t) = \frac{1}{4\pi\sigma} \sum_i \frac{I_i(t)}{|r_e - r_i|}$$  (3.8)

Eq. 3.8 assumes that transmembrane currents are generated from a single point and it is therefore referred to as point-source approximation (Figure 3.1C). A more realistic approximation treats the neuronal segments as lines rather than points (line-source approximation). In this case the transmembrane currents are evenly distributed along the compartments. Integrating Eq. 3.8 over the segments’ axes, the potential can be computed as:

$$\phi_{ex}(r_e, t) = \frac{1}{4\pi\sigma} \sum_i \frac{I_i}{|r_e - r_i|} \int_{r_i-1/2}^{r_i+1/2} \frac{dr}{|r_e - r|}$$  (3.9)

where $r_{i-1/2}$ and $r_{i+1/2}$ represent the initial and final position of each compartment.

In both Eq. 3.8 and 3.9, the electrode is represented as a three-dimensional point $r_e$. In order to consider the spatial extent of the recording contact, one can compute the potential on several points belonging to the electrode surface and average their values (disk-electrode approximation).

The above-described formulations are based on several assumptions. First of all the conductivity of the medium is assumed to be scalar, hence neglecting capacitive properties of the tissue. This assumption seems however to be well justified for relevant frequencies in extracellular recordings.

Second, the medium is assumed to be isotropic, but this assumption is harder to relax. In the neural tissue, in fact, the presence of oriented pyramidal cells makes conductivity anisotropic. Anisotropy in the tissue can be accounted for with analytical solutions.

Finally, the extracellular milieu is assumed to be homogeneous (without discontinuities) and infinite. This is clearly a stronger assumption, considering that in order to measure the electric potentials generated by the neurons, we insert a probe in their vicinity. In case of in vitro preparations, in which cell cultures or slices lie on a planar MEAs, the effect of the electrode plane and of the discontinuity between the neural tissue and the saline solution can be modeled also analytically, using the method of images. For more complicated cases, one can use numerical solutions, such as finite element methods (FEM) (Section 3.4). In Paper VII, for example, we used FEM to simulate the effect of in vivo neural probes on the recorded potentials.

**Software** There are some available and open-source software to compute the extracellular potentials generated by neural activity. BIONET developed by the Allen Institute for Brain Science, is a software for large-scale simulations.
3. Computational models

It includes a wrapper to the NEURON environment and it enables to compute extracellular potentials arising from network simulations using the line-source approximation (Eq. 3.9). In this thesis, I used LFPy, developed by our group\textsuperscript{152,154} for simulating extracellular potentials. LFPy also provides a NEURON wrapper and it has a simple API for the definition of cells and electrodes. The new version of LFPy\textsuperscript{154} also includes computationally-efficient forward modeling schemes for the calculation of electroencephalography (EEG) and magnetoencephalography (MEG).

3.3 Models of electrical stimulation

In order to model activation of neurons from electrical stimulation, we can combine principles from Section 3.1 and Section 3.2. The models presented in this section have been mainly derived for axons, as axons are the most excitable neuronal parts\textsuperscript{125} (alongside with the axon initial segment - see Section 2.4).

Stimulation can trigger action potentials by imposing an extracellular potential at the neuron’s membrane. In order to compute these potentials from a stimulating electrode at position $r_e$, we can consider, in a first approximation, the electrode as a point current source and use Eq. 3.7.

Let us now get back to the cable equation. During the derivation, we set the extracellular potential to be constant, but this assumption clearly needs to be revisited as we are attempting now to simulate the effect of extracellular potentials on the neuron dynamics. If we relax this assumption, we can rewrite Eq. 3.6 as\textsuperscript{129,133}:

$$\tau \frac{\partial V_m}{\partial t} - \lambda^2 \frac{\partial^2 V_m}{\partial x^2} + \frac{i_{ion}}{g_l} + V_m = \lambda^2 \frac{\partial^2 \phi_{ex}}{\partial x^2}$$

(3.10)

From this formulation, one can see that the source term or force function of the differential equation (the right-hand side) is proportional to the second spatial derivative of the extracellular potential $\phi_{ex}$. The $\tau \frac{\partial V_m}{\partial t}$ term defines the capacitive currents, and $\lambda^2 \frac{\partial^2 V_m}{\partial x^2}$ term represents the longitudinal currents.

While Eq. 3.10 could be solved numerically, several studies tried to propose estimators for neural activation from extracellular stimulation, in order to avoid to solve Eq. 3.10 and still have “first impression of the influence of an applied electric or magnetic field on a target neuron.”\textsuperscript{126}

One of the first estimator proposed in literature is the activating function\textsuperscript{126,132,156}. If we assume that the neuron is at rest, the spatial derivative of $V_m$ (longitudinal currents) can be neglected since $V_m$ is relatively constant. Then, the polarization at the steady state will be proportional to the second spatial derivative of the extracellular potential along the axon:

$$V_m \propto AF = \lambda^2 \frac{\partial^2 \phi_{ex}}{\partial x^2}$$

(3.11)

The activating function $AF$ is convenient because it can estimate neural activation only by knowing the extracellular potential and the neuronal...
morphology. Even neglecting the $\lambda$ term, which might be hard to correctly estimate, the $AF$ can tell whether a region is hyperpolarized ($AF < 0$) or depolarized ($AF > 0$). The $AF$ has been used in Paper VI to predict neural activation in order to optimize electrical stimulation patterns involving multiple electrodes using MEAs.

It is not clear, however, if the longitudinal currents can be safely ignored, as they are scaled by $\lambda^2$, so their contribution depends on this parameter. Secondly, the $AF$ wrongly predicts that an axon is not activated by an applied linear potential. While this might be true for infinitely long axons, in practice neurites have a finite extension and edge effects need to be considered. Another study suggests that when $\lambda$ is large enough, then the longitudinal currents are more prominent than the capacitive currents and cannot be neglected. At the steady state, Eq. 3.10 can then be approximated as:

$$-\lambda^2 \frac{\partial^2 V_m}{\partial x^2} + V_m = \lambda^2 \frac{\partial^2 \phi_{ex}}{\partial x^2}$$

When $\lambda$ is “small enough”, Eq. 3.13 coincides with the $AF$, but when $\lambda$ is large the predictor of $V_m$ becomes:

$$V_m \propto ME = -\phi_{ex} + \langle \phi_{ex} \rangle_L$$

The mirror estimate ($ME$) predicts the depolarization/hyperpolarization of a neurite as the opposite of the applied extracellular field ($mirror$) centered on its mean value ($\langle \phi_{ex} \rangle_L$).

Figure 3.2 shows the extracellular potential ($\phi_{ex}$), $AF$ and $ME$ predictions for a cathodic (current drawn by the electrode - A) and an anodic stimulation (current injected by the electrode - B) for a linear and uniform axon. In this case, the extracellular potential can be computed analytically from Eq. 3.7 for a fiber oriented in the $x$ direction and located at a distance $d$ from the electrode:

$$\phi_{ex}(x) = \frac{I_s(t)}{4\pi \sigma \sqrt{d^2 + x^2}}$$

While $AF$ and $ME$ agree on the position and magnitude of the central depolarization/hyperpolarization, there are some differences between these two estimators in the shape of the depolarization and hyperpolarization regions.

Although estimators can provide a feeling about the effect of extracellular stimulation, one of their main limitation is that they do not take into account the time course of the stimulation, as they predict the neural activation generated by a constant monophasic pulse. However, the effect of the pulse waveform, e.g., biphasic pulses, or of the application of a train of stimulation pulses cannot be estimated. A more powerful, but computationally intense approach consists of solving Eq. 3.10 to simulate the entire dynamics of the neuron in response to an extracellular time-varying stimulation. Nevertheless, even using this approach

*the spatial mean can be computed analytically also for non-uniform fibers, accounting for different diameters ad leaky properties.
3. Computational models

![Diagram of extracellular stimulation and estimators](image)

Figure 3.2: Extracellular stimulation and estimators. Extracellular potential ($\phi_{ex}$), activating function ($AF$) and mirror estimate ($ME$) generated by a cathodic current (A) and by an anodic current (B) on a linear neurite.

some assumptions are made. First of all, since the cable formulation constructs a 1D problem, it assumes that the extracellular potential varies only along the main direction of the fiber, and it is constant in the orthogonal direction. Similarly, it assumes that the membrane potential is constant along the circumferential direction of the neuronal membrane. Finally, it assumes that the presence of the fiber does not affect the extracellular field\textsuperscript{127,139}.

Finite element methods have been used in literature for simulating extracellular potentials in highly non-homogeneous problems and pair them with the cable equation solution (hybrid approach\textsuperscript{42,138,139,158–160}), but the use of the cable framework cannot relax the above-mentioned assumptions. Alternatively, whole-FEM methods, with explicit representation of the extracellular and intracellular spaces, have been suggested and can be used for a detailed study of the effect of extracellular stimulation, at the cost of computational burden\textsuperscript{42,139,161} (see Section 3.4).

Software The NEURON framework can incorporate the effect of a non-constant extracellular potential generated by external currents using the extracellular mechanism, which effectively adds two layers of extracellular potentials outside the membrane and solves Eq. 3.10 instead of Eq. 3.6.

In order to compute extracellular potentials from MEAs, I developed the MEAutility\textsuperscript{*} Python package. MEAutility is convenient to define MEA designs (the user can also use a wide variety of probe designs from the existing probe library), to set static or time-varying currents for each electrode, and to compute

\textsuperscript{*}https://github.com/alejoe91/MEAutility
extracellular fields at neural locations, which can then be paired to NEURON simulations or used to compute estimators. MEAutility is used in Paper I to handle MEA designs, and in Paper VI to compute extracellular potential along the neurons.

3.4 Finite element methods

Finite element methods (FEM) are numerical methods to solve differential equations. The basic idea is to divide the domain of interest (in our case, for example, the extracellular space and optionally the neurons) in small regions (finite elements) and to approximate the unknown function (e.g. the extracellular potentials) as a linear combination of simple functions defined on these regions.

FEM can be used both to compute extracellular potentials generated by the neural activity and to simulate the effect of extracellular stimulation. Another interesting use of FEM is to validate analytical derivations with numerical simulations. For example, in Ness et al., a FEM simulation is used to validate the use of the method of images to analytically compute the extracellular potential for in vitro preparations. In Næss et al., the same approach is used to correct an analytical model of the head conductivity for EEG computation.

3.4.1 Hybrid approach

The hybrid approach combines the FEM and the cable framework. It has been used both for computing extracellular potentials in non-homogeneous extracellular spaces and for computing the effect of stimulation, especially for the spinal cord.

When used for computing extracellular potentials, first neuronal dynamics are solved (for example with NEURON); then, transmembrane currents are used as force functions for a FEM simulation of the extracellular space. Conversely, when simulating the effect of external stimulation, extracellular potentials at the neuron compartments’ location, generated by stimulating electrodes, are first computed using FEM; then the cable equation is solved (using the extracellular mechanism in NEURON).

While the hybrid approach can be useful to model complex geometries, the cable equation framework preserves its assumptions listed in Section 3.1 and Section 3.3.

3.4.2 EMI model

A second and more advanced approach consists of explicitly modeling the extracellular space, the membrane of the neuron, and the intracellular space. This model is also called the EMI model (Extracellular-Membrane-Intracellular).

*an extensive mathematical formulation of finite element methods is beyond the scope of this thesis.
Figure 3.3: Mathematical representation of the EMI model for a simplified neuron and an extracellular probe. Figure from Paper VII.

or whole-FEM model. The EMI model simultaneously solves the following equations:

\[
\nabla \cdot \sigma_i \nabla \phi_{in} = 0 \quad \text{in } \Omega_i, \quad (3.15) \\
\nabla \cdot \sigma_e \nabla \phi_{ex} = 0 \quad \text{in } \Omega_e, \quad (3.16) \\
\phi_{ex} = 0 \quad \text{at } \partial \Omega_e, \quad (3.17) \\
n_e \cdot \sigma_e \nabla \phi_{ex} = -n_i \cdot \sigma_i \nabla \phi_{in} \equiv i_m \quad \text{at } \Gamma, \quad (3.18) \\
\phi_{in} - \phi_{ex} = V \quad \text{at } \Gamma, \quad (3.19) \\
C_m \frac{\partial V}{\partial t} = -I_{ion} - I_{leak} + I_{ext} \quad \text{at } \Gamma, \quad (3.20) \\
\left( \sigma_e \nabla \phi_{ex} \cdot n_e \right) = 0 \quad \text{at } \partial \Omega_p. \quad (3.21)
\]

With reference to Figure 3.3, Eqs. 3.15 and 3.16 are the Poisson equations for the intracellular and extracellular spaces (with conductivities \( \sigma_i \) and \( \sigma_e \)); Eq. 3.17 is the Dirichlet boundary condition to set the potential to zero far away from the neuron; Eq. 3.18 defines the conservation of the transmembrane current density across the membrane; Eq. 3.19 defines the membrane potential as the difference between the intracellular and extracellular potentials; Eq. 3.20 describes the membrane model (see Eq. 3.2); and Eq. 3.21 added when considering an extracellular probe in Paper VII is the Neumann boundary condition that describes the insulating property of the probe.

The EMI model considers the full 3D morphology of the neuron. The EMI solution can in fact reproduce different extracellular potentials along the circumferential direction of the membrane as well as different intracellular potentials in the cross section of the intracellular space. Moreover, the EMI
model does not break down the neuronal morphology into compartments, as it represents the neuron as a *continuum*. Finally, the extracellular potential is not neglected when solving the neuron dynamics, allowing for the study of ephaptic and self-ephaptic effects\textsuperscript{42,146}.

The main limitation of the EMI formulation, however, is that it is much more computationally intense than the cable approach. The hybrid approach lies somewhere in between\textsuperscript{42}.

In [Paper VII] we used both the hybrid and the EMI solutions to investigate the effects of extracellular probes on the recorded potentials. Figure 3.4 shows some examples of the meshes that we used (A - microwire, B-C - MEAs) in combination with a simple ball-and-stick model of a neuron\textsuperscript{142}.

**Software** A very common software used for FEM is COMSOL. However, it is a commercial solution and several groups have contributed with open-source software. **gmsh**\textsuperscript{165} is an open-source and powerful software for generating 3D meshes of complex geometries. **FEniCS**\textsuperscript{166} is a computing platform for solving

---

\*the continuous morphology is discretized when creating the mesh, but increasing the mesh resolution can achieve higher accuracy in representing the morphology.
3. Computational models

partial differential equations in Python. It combines a high-level Python API and C++ API which makes it very accessible, versatile, and easy to use. *gmsh* and *FEniCS* have been used in [Paper VII](#) for generating the meshes and computing the EMI and hybrid solutions.
Chapter 4

Engineering solutions

“Engineers like to solve problems. If there are no problems handily available, they will create their own problems.”

— Scott Adams

In this chapter, I will describe the engineering and implementation solutions used in the presented papers.

4.1 Independent component analysis

Independent Component Analysis (ICA) is a signal processing method used for blind source separation and dimensionality reduction. In the latter case, ICA is used to find a better representation of the data making use of their statistical structure. In Paper III and Paper IV ICA has been used for blind source separation. A classic example of blind source separation is the so-called cocktail party.

Imagine that \( N \) guests are chatting in a bar while sipping their cocktails. Moreover, there are \( M > N \) microphones located at different positions in the bar. Actually, let us simplify the problem and say that we have exactly \( N \) microphones. Each microphone records then a mixture of voices from the different guests. Let us define the speech signals coming from the guests as \( s(t) \in \mathcal{R}^N \) and let us assume that these signals are statistically independent, as the guests are not singing in a choir. Furthermore, we can assume that the sound propagation velocity is negligible. Therefore, we can describe the set of recorded signals \( x(t) \in \mathcal{R}^N \) as a linear combination, or mixture, of \( s(t) \):

\[
x(t) = As(t)
\]

where \( A \) is defined as the mixing matrix and \( s(t) \) are called sources. Solving the ICA problem is far from trivial, as both the sources \( s(t) \) and the matrix \( A \) are unknown. Usually, the ICA problem is formulated as:
4. Engineering solutions

\[ y(t) \approx W x(t) \] (4.2)

where \( y(t) \) is an estimate of \( s(t) \) and \( W \) is the unmixing matrix, which is the pseudo-inverse of the mixing matrix \( A \). There are different approaches for solving this problem. However, the underlying principle is to find an optimal \( W \) that maximizes the statistical independence of the estimated sources \( y(t) \).

In order to better explain the ICA principle, we can consider a neuronal cocktail party. When we insert a probe in the brain (the microphones), we will hopefully pick up the activity of several neurons (the speech signals). Let us create a simplified case to show how ICA transforms the data and how it differs from other techniques, in particular from principal component analysis (PCA). In this example we insert a two-channel probe, which records the activity of two surrounding neurons. We have to assume that the specific spike times of the two neurons are fairly independent. In the recordings, there is also some additive Gaussian background noise. For the sake of clarity, in this example we created the recordings as a linear mix of two spiking sources with some additive noise. Therefore, the recorded signals, at each time point, are instant mixes of the sources, without any phase shift. This assumption for neural signals is discussed in Section 2.1 (ICA-based approach paragraph).

Figure 4.1A shows the simulated recordings, the waveforms of the two neurons, and a scatter plot of the signals on the channel dimensions. The directions of the two neurons can be clearly identified.

When we apply PCA, the signals are projected on the dimensions that maximize the variance of the data. The PCA directions are shown in Figure 4.1A as orange arrows and PCA-transformed signals, waveforms, and the scatter plot of the data on the principal directions are shown in Figure 4.1B. After the PCA transform, the data are decorrelated. Decorrelation implies that the covariance matrix of the variables is diagonal. Related to this, whitening makes the covariance matrix equal to the identity matrix (each direction is normalized). However, each principal component contains activity from both neurons, since the principal axes, displayed in Figure 4.1A as orange arrows, are in the middle between the two neurons’ directions. This makes the PCA-transformed waveforms appear on both PCA channels.

Applying ICA to the data gives us different directions. Statistical independence is in fact different from uncorrelatedness. When two random variables \( x_1 \) and \( x_2 \) are uncorrelated, their covariance is zero:

\[ E[x_1, x_2] - E[x_1]E[x_2] = \text{Cov}[x_1, x_2] = 0 \] (4.3)

When two variables are statistically independent it can be shown that non-linear correlations are also zero:

\[ E[h(x_1), g(x_2)] - E[h(x_1)]E[g(x_2)] = \text{Cov}[h(x_1), g(x_2)] = 0; \forall h(), g() \] (4.4)

*or alternatively singular value decomposition (SVD)
Independent component analysis

Figure 4.1: Illustration of PCA and ICA projections. (A) Top: Simulated recording and waveforms with two channels and two neurons; Bottom: data points on channel 1 - channel 2, with PCA and ICA axis in orange and green, respectively. From the waveforms and the data clouds one can see that the waveform of each unit is present on both recording channels. (B) Top: PCA-transformed signals and waveforms; Bottom: data points in the PCA space. The two units still appear on both axis. (C) Top: ICA-transformed signals and waveforms; Bottom: data points in the ICA space. The ICA axis are selectively tuned to the two units. The waveforms now mainly appear on a single ICA dimension.

Clearly, two statistically independent variables are also uncorrelated \( h(x_1) = x_1, h(x_2) = x_2 \). ICA finds directions that are selectively tuned to the activity of the separate neurons in the recordings. These directions are shown as green arrows in Figure 4.1A. Figure 4.1C shows the ICA-transformed signals, waveforms, and scatter plot. In this case, each IC source is specifically tuned to one of the neuron, so that the waveform mainly appear on a single ICA direction.

There are several approaches to solve the ICA problem, i.e., to maximize the independence of the estimated sources \( y(t) \). ICA is solved iteratively with optimization techniques aiming to maximize a cost function that reflects the status of independence of the sources.

The independence of the sources can be computed in different ways. Independence can for instance be related to non-gaussianity. Due to the Central Limit Theorem, in fact, the sum of two or more independent variables makes the distribution more Gaussian. Hence, projections that maximize non-gaussianity also maximize independence between the variables. Measures of non-gaussianity include kurtosis and negentropy. The famous FastICA
Engineering solutions

The FastICA algorithm maximizes negentropy with a fixed-point iteration scheme, and it has been used in Paper III. Alternatively, the ICA problem can be solved by minimizing the mutual information, a theoretical information measure that indicates dependence between variables. The minimization of mutual information is equivalent to maximizing negentropy.

Another widely used approach for solving the ICA problem is the infomax principle (maximization of information flow). The learning rule for this approach using natural gradient can be written as:

\[ W_{n+1} = W_n + \eta [I - f(y_n)y_n^T]W \]  

where \( \eta \) is the learning rate and \( f(\cdot) \) is a non-linear function. From this formulation, one can see how the infomax principle is also closely related to non-linear decorrelation (Eq. 4.4), as the gradient tends to remove the non-linear correlations given by \( f(y_n)y_n^T \). It can be also proven that maximizing the information flow (in other words the joint entropy) is analogous to minimizing mutual information. In the infomax approach, the choice of the function \( f(\cdot) \) is important: in order to maximize the joint entropy of the sources, hence maximizing independence, the product \( f(y_n)y_n^T \) should result in a uniform distribution. Therefore, the infomax principle has been extended to match different distributions, such as subgaussian and supergaussian ones. However, since the distribution of the sources \( y_n \) is unknown a-priori, one can estimate it by computing the kurtosis of each source and choosing the correct non-linearity \( f(\cdot) \) accordingly.

From the natural gradient infomax learning rule in Eq. 4.5, an online recursive ICA solution – ORICA – has been developed. The adaptive learning rule reads as:

\[ W_{n+1} = \frac{1}{1 - \lambda_n} W_n - \frac{\lambda_n}{1 - \lambda_n} \left( 1 + \frac{y_n f^T(y_n)}{f^T(y_n) y_n - 1} \right) W_n \]  

where \( \lambda_n \) is a time-varying forgetting factor. Using this online formulation, the ICA model can be estimated as the data stream is acquired. The forgetting factor enables the online algorithm to gradually forget about data acquired in the past and adjust the model to newly acquired data. This is of particular interest to track non-stationarity in the data. This solution has been adopted in Paper IV and it is promising for online spike sorting solutions. Neural recordings can in fact exhibit a drift due to relative movement between the tissue and the electrodes. While drift can be corrected for by some automatic offline algorithms, there is no current solution for handling drift online. ORICA could therefore be a valuable solution to tackle this problem for real-time applications.

Software The FastICA algorithm is available directly from the scikit-learn Python package, while other Python implementations can be run with the MNE package. Alternatively, several ICA implementations in MATLAB are available through EEGLAB. For Paper IV, a Python implementation of the ORICA algorithm was developed (https://github.com/alejoe91/spyica).
Machine learning and deep learning

Machine learning can be defined as a set of algorithms that are able to automatically learn from data.

Machine learning algorithms are historically divided in three different classes: supervised learning, unsupervised learning, and reinforcement learning. After a brief description of the three types of learning, I will focus on supervised learning with artificial neural networks and convolutional neural networks, because of their use in Paper V.

Supervised learning aims to learn a relation, or model, between some input and output data. In order to perform this kind of learning, one needs a labeled dataset, i.e., a collection of observations where both the input and corresponding output are known. The output can either be categorical or a real value. The former case is referred to as a classification problem, the latter as a regression problem. In Paper V, we used supervised learning to solve both a classification problem to predict the neuronal cell type, and a regression problem for finding 3D positions of the neurons with respect to the recording probe, using the extracellular spikes as input.

Unsupervised learning aims to learn internal structures of the data, without targeting a particular input-output relation. Unsupervised learning uses unlabeled data. A classical example of unsupervised learning is clustering: given a set of data, clustering algorithms try to find different sub-groups in the data, without knowledge of whether there are any groups and how many groups there are. Clustering is widely used in the spike sorting literature (Section 2.1 - Clustering-based approach) and it has also been used in Paper V for cell-type classification, in order to compare the unsupervised performance to the supervised approach.

Reinforcement learning aims to learn to perform the right action in a certain state to achieve a goal. It is a typical problem for autonomous agents, e.g., robots, and learning is achieved by dispensing a positive or negative reward depending on whether the chosen actions helped the agent to reach its goal.

4.2.1 The basics of artificial neural networks

Among the large variety of machine learning algorithms developed in the past decades for supervised learning, neural networks have arguably become the most popular set of algorithms.

The idea behind artificial neural networks comes from biology, as a simplification of how neurons integrate input signals. An artificial neuron, or perceptron \(^{176}\) (Figure 4.2A), is a simple entity that transforms the sum of its weighted inputs and a bias value:
Figure 4.2: (A) Artificial neuron (or perceptron). The weighted inputs and the bias are transformed with the activation function $f$ to generate the output. (B) Neural network and definitions for the Iris dataset example described in the text.

$y = f \left( \sum_i w_i x_i + b \right)$  \hspace{1cm} (4.7)

where $x_i$ is the $i$-th input to the neuron, $w_i$ is the weight of the input $x_i$, $b$ is a scalar bias, $f(\cdot)$ is the so-called activation function, and $y$ is the output of the neuron. The function $f(\cdot)$ enables the neuron to represent non-linear transformations of the input data. Typical activation functions are sigmoids, hyperbolic tangents, or rectified linear functions. When artificial neurons are combined and connected together, the resulting network is capable of finding very complex and non-linear relations between inputs and outputs.

Let us look at a simple real-world example and see how a neural network can be constructed. A famous dataset to test machine learning algorithms is the Iris dataset. It contains observations of the sepal and petal lengths and widths of three Iris flower species, the *Iris setosa*, *Iris virginica*, and *Iris versicolor*. We would like to find a model that predicts the species from the petal and sepal morphological information. This is a typical example of classification, as the output is categorical.

We can build a three-layer neural network. The first layer is the input layer and it propagates the inputs to the all the nodes of the next layer. In the Iris case, we have four inputs (petal length, petal width, sepal length, and sepal width), so we will have four input neurons. The second layer is also called hidden layer, as it is not seen by the inputs nor the outputs. There can be many hidden layers, but in this case, for simplicity, we will only have one layer with, for example, five neurons. The hidden layers are very important to capture complex and non-linear relations in the input data, which are connected to the output. The last layer is the output layer. In this case there are three outputs (one for
Machine learning and deep learning

each class), hence three output neurons. Figure 4.2B displays a network with these three layers. Nodes from the input layer are referred to as \( I_i \), from the hidden layer as \( H_j \), and from the output layer as \( O_k \). The outputs of neurons belonging to input, hidden, and output layers are defined as \( y_I \), \( y_H \), and \( y_O \), respectively. The activation function \( f(\cdot) \) is a sigmoid function, whose co-domain is 0 to 1. The predicted class for an input \( x = [x_1, x_2, x_3, x_4] \) is the class whose output neuron is closer to 1. The output of a neuron from the output layer can be written as:

\[
y_{O_k} = f \left[ \sum_j w_{jk} y_{H_j} + b_{O_k} \right] = f \left[ \sum_j w_{jk} f \left( \sum_i w_{ij} x_i + b_{H_j} \right) + b_{O_k} \right] \quad (4.8)
\]

Now that we have a network architecture for our classification task we have to train it, i.e., to learn the parameters that better describe the input-output relation. The set of parameter that we want to optimize includes the weights from the input to the hidden layer \( w_{ij} \) (20 parameters), the ones from the hidden to the output layer \( w_{jk} \) (15 parameters), and the bias values of the hidden nodes \( b_{H_j} – 5 \) parameters) and the output nodes \( b_{O_k} – 3 \) parameters). In total, there are 43 parameters to be fitted. Note that input nodes have no weights nor biases, as they propagate the inputs to the next layer without any modification. In order to fit the parameters, we have to define a cost or loss function that tells us how well the network is performing. Given an observation \( n \), its labeled output can be encoded as an array \( t_n \): (1, 0, 0) for Iris setosa, (0, 1, 0) for Iris virginica, and (0, 0, 1) for Iris versicolor. Similarly, the output of the network given the input \( x_n \) can be written as an array \( y_n \), which depends on the values of \( y_{O_1}, y_{O_2}, \) and \( y_{O_3} \). For example, if \( y_{O_2} \) is the output value closest to 1, then \( y_n \) will be (0, 1, 0). A cost or error function \( J \) that describes how well the network is performing over the entire dataset can be the mean of the squared errors (MSE):

\[
J(W) = \frac{1}{N} \sum_n \| t_n - y_n(W) \|^2 = \frac{1}{N} \sum_n \sum_v \left[ t_{nv} - y_{nv}(W) \right]^2 \quad (4.9)
\]

where \( W \) is the set of all parameters and \( v = (1, 2, 3) \) is the output dimension. Note that the dependence of \( y_n \) on \( W \) is explicit.

We can now use an optimization algorithm, for example, the gradient descent, to iteratively update the weights and minimize the cost function \( J \). Using gradient descent, at each iteration a generic weight \( w \) is updated as follows:

\[
w_{t+1} = w_t - \eta \Delta w_t = w_t - \eta \frac{\partial J}{\partial w} \bigg|_{w=w_t} \quad (4.10)
\]

where \( \eta \) is the learning rate. Since the computation of the gradient is effectively propagating the error function back to previous layers, this

*this is a very small network!
optimization strategy is also referred to as *backpropagation* in the neural network literature.

Choosing a differentiable activation function, the partial derivatives of the weights and bias values of the different layers can be computed using the chain rule. For example, the gradient of a weight that connects the hidden and the output layer \((w_{jk})\) can be computed as:

\[
\frac{\partial J}{\partial w_{j=1,k=2}} = \frac{\partial J}{\partial y_n} \frac{\partial y_n}{\partial w_{j=1,k=2}} = \frac{\partial J}{\partial y_n} \frac{\partial y_{O_2}}{\partial w_{j=1,k=2}} = \frac{\partial J}{\partial y_n} \frac{\partial f\left(\sum_j w_{j,k=2} y_{H_j} + b_{O_2}\right)}{\partial w_{j=1,k=2}} = \frac{2}{N} \sum_n \sum_v \left[t_{nv} - y_{nv}(W)\right] f'\left(\sum_j w_{j,k=2} y_{H_j} + b_{O_2}\right) y_{H_j=1}
\]

(4.11)

In the first row of the equation, the only component of \(y_n\) that depends on \(w_{j=1,k=2}\), is indeed \(y_{O_2}\). Similarly, gradients for biases and weights for all layers can be computed.

Here I presented an illustrative implementation of a classification task using a small neural network and some simplifications. For classification, usually the output neurons use a softmax activation function and the cross-entropy as loss function\(^{178}\). Differently from other activation functions, softmax transforms the values of all output neurons in probabilities. Moreover, here we used all data points to compute the gradient (\(\sum_n\)), but in practice, for larger datasets, smaller batches of observations are used at each iteration. This method is referred to as stochastic gradient descent. While this example formalized a classification problem, regression can be achieved with the same principle and a few changes in the output layer. In order to predict one or multiple real values, the output neurons have no activation function, so that the value of \(y_{O_k}\) can range from \(-\infty\) to \(+\infty\). The mean squared error can be used as loss function also for regression problems.

### 4.2.2 Deep Learning

Deep learning refers to the use of much larger architectures with many more parameters, layers, and artificial neurons. Deep learning is now used ubiquitously in many applications, including image and video recognition, natural language processing, and medical diagnosis. Some of these architecture are indeed very deep. For example, the ResNet\(^{179}\) architecture, used for image recognition, can contain up to 1'200 layers.

Deep learning can take several flavours. Different architectures have been in fact developed to tackle diverse classes of problems. Patterson and Gibson\(^{180}\) identify four different classes of deep network architectures: unsupervised

Unsupervised pretrained networks learn compressed representations of the data in an unsupervised manner. In order to do so, these networks are trained to reproduced the input data. In other words, the input is also used as labeled output. Belonging to this class are autoencoders, generative adversarial networks, and deep belief networks.

Recurrent neural networks are different from the type of network shown in Section 4.2.1, which is a feed-forward network, because they can have connections between the same layer (recurrent). An example of this class of architectures is the long short-term memory (LSTM) networks, which are commonly used for temporal sequence learning, such as natural language processing. Thanks to their recurrent nature, these networks are able to keep the memory, or the state, of previous samples.

Recursive neural networks are similar to recurrent networks, but they have a tree structure, which allows them to find hierarchical structures in the data.

The last class of architectures are convolutional neural networks, which have been used in Paper V and are presented in the next section.

4.2.3 Convolutional Neural Networks

Convolutional neural networks (CNNs) are biologically inspired from the information processing of the visual system \[181,182\]. Their architecture is different from other configurations mainly because of the use of convolutional layers. CNNs are typically used in computer vision for image classification.

Figure 4.3 shows a sample architecture of a CNN, similar to the one used in Paper V. The input data are organized in a 2D structure, or an image \(I\), and convolutional kernels \(K\) are convolved with the image to extract feature maps. These are then subsampled using pooling operators. Max pooling, for example, reduces the dimensions of the maps by taking the maximum of adjacent pixels. The convolution-pooling operations can be repeated multiple times (two times in Figure 4.3 and in Paper V). Finally, the outputs of convolutional-pooling layers are connected to one or more fully-connected layers, analogous to the hidden layers shown in Section 4.2.1. The last fully-connected layer is then interfaced to the output layer, which outputs the predictions of the network.

The convolution layers consist of several kernels, or filters. The number of kernels of a convolutional layer is called depth. Each kernel has a certain size (width and height) and it is convolved with the image. For example, if a kernel \(K\) has a 3x3 size, the input to a neuron from the convolutional layer is the convolution between the kernel and the image \(I\):

\[
F(i, j) = f \left[ \sum_k \sum_l K(k, l) I(i - k, l - j) \right]
\]

where \(k = (-1, 0, 1)\) and \(l = (-1, 0, 1)\) as the kernel size is 3x3. \(F(i, j)\) is the value of the feature map at pixel position \(i, j\). The activation function \(f(x)\) is
Figure 4.3: Representation of a Convolutional Neural Network. The 2D input is convolved with 2D filters to generate feature maps. Convolved maps are subsamples using the pooling layers. The last fully connected layer is connected to the output neurons.

usually chosen to be a rectified linear unit (ReLU): the output is zero if \( x \leq 0 \), and it is \( x \) when \( x > 0 \). At the edges of the image \( I \), padding can be applied in order to maintain the same output size. Moreover, the \textit{stride} parameter controls the amount of overlap in pixel convolutions. When the stride is set to 1, there are no gaps between convolutions and \( F \) has the same size of \( I \) (provided that padding is used at the edges).

Training is performed as described in Section 4.2.1 using backpropagation. The convolutional layers, during training, learn filters that extract relevant features from the structured data, which are used by the fully connected layer to improve the prediction accuracy.

In Paper V, we have used CNNs to predict the neuronal location and cell type from extracellular action potentials (EAPs) on dense MEAs. CNNs have been chosen because the configuration of recording sites for recent MEAs is 2D, and the recorded signals can hence be considered as electrical images. Therefore, the task is similar to image classification and CNNs are suited to find 2D structures in the data. However, the EAPs are actually 3D input data, because they evolve over time as well. In order to use CNNs, then, we extracted 2D feature images from the EAPs, including negative and positive peak images, peak-to-peak amplitude, peak-to-valley width, and full-width half maximum. We then fed the 2D feature images as input to the network. Alternatively, we also used 3D kernels that convolved the full 3D EAPs also in the time domain. This approach showed an increase in performance, but the training time was much larger than using 2D convolutional layers.

Due to their very large number of parameters, CNNs (and deep learning
networks in general) can be easily over-trained. When this happens, the model works very well on the data used for training, but it is not capable to generalize over new observations.

4.2.4 Underfitting and overfitting: finding the right model

One of the most important problem in the machine learning literature is to construct models that can provide good predictions of new and unseen data. In order to quantify the performance of the model over new observations, the initial dataset is usually split in two different subsets:

- **Training set**: the training set is used to perform the training and optimize the parameters of the model
- **Test set**: the test set is not used for training (it is also denoted as *holdout set*) and used to evaluate the generalization of the model

The errors on the training and test datasets are defined as training and test errors. Ideally, one wants a machine learning model to 1) have a small error on the training set and 2) to have a small validation gap, i.e., the difference between the training and test errors.

If the model is too weak or simple and it is not able to reach a high performance on the training set in the first place, then the model is **underfitting**. A typical example of underfitting is having data points drawn from a quadratic function and trying to fit it to a straight line. Alternatively, the model may perform very well on the training set, but the performance on the test set is poor. In this case, the model learned aspects of the training data that are not general, such as the noise. The model is **overfitting** the training data. A good model lies in between these two scenarios, with a good performance both on the training and test sets.

In case of deep neural networks, given a large enough architecture, overfitting is definitely more problematic than underfitting. These models have in fact a very large capacity\[^{175}\], that is, the capability of fitting a large variety of functions if properly trained. One way of stemming overfitting is to use regularization techniques, which pose some constraints on the weights. Examples of regularizations are L1 and L2 regularizers, which make the weights sparser (L1) or limit their values (L2). Another popular method against overfitting is dropout\[^{183}\]. When using dropout, during training, a percent of the weights is randomly selected at each iteration and it is *dropped* from updating. Therefore, only a subset of weights is updated at each iteration, effectively increasing the generalization power of the model. Dropout is usually applied layer-wise, and in Paper V it is applied only to the last fully-connected layer.

Another way to improve the model performance is by hyperparameter tuning. Hyperparameters include all the parameters of the model which are not trainable, such as the number of neurons, the learning rate, and the number of training iterations. In order to choose hyperparameters, a validation set can be extracted from the training set and used as a proxy of the test set in order to assess the
4. Engineering solutions

generalization of the model. For example, one can monitor the validation error and stop the training when this starts to diverge from the training error (early stopping).

Software As machine learning and deep learning applications have become ubiquitous, several software solutions exist. For standard machine learning methods (excluding neural networks) the scikit-learn\textsuperscript{173} Python package is a fantastic tool for machine learning. Specifically targeting neural networks and deep learning, both Google and Facebook have developed powerful and open source frameworks: Tensorflow\textsuperscript{184} (Google) and PyTorch\textsuperscript{185} (Facebook). In 2017, Tensorflow incorporated Keras\textsuperscript{186} in their core API, making the creation of deep learning architectures easier and more abstract. Tensorflow has been used in Paper V.

4.3 Optimization and genetic algorithms

In several instances throughout this thesis I had to find a set of parameters to fit some data or to yield a good solution for a given problem. Generally, the problem of finding parameters that optimize a certain cost function is called an optimization problem. Examples of optimization problems that have been introduced in the previous sections include finding the solution of the ICA model and training artificial neural networks.

There is a large variety of strategies to solve optimization problems. Many of them are based on estimating the gradient of the cost function and using it to find the minimum of the function. For example, the gradient descent method follows the steepest path, and other methods add some momentum and adaptive behavior to find the minimum faster and not to get stuck in local minima\textsuperscript{187,188}. One drawback of gradient-based methods is that for some problems the gradient is unknown. In these cases a finite approximation in parameter space is used, but this could be time consuming in some applications.

Alternatively, a second class of optimization tools are evolutionary algorithms. Genetic algorithms are part of the evolutionary algorithms class and they are biologically-inspired, mimicking the evolution process by natural selection\textsuperscript{189}. Genetic algorithms are general-purpose optimization methods that can be applied to any optimization problem. Moreover, as they explore the solution space randomly and they visit several points simultaneously, they can be quite fast, but their meta-heuristic nature does not guarantee convergence to a global minimum.

The fundamental unit of genetic algorithms is a chromosome, or individual, i.e., a representation of a possible solution. At initialization, $N$ chromosomes are randomly created to form a population $P$. Each chromosome’s fitness is evaluated with a fitness function, which indicates how good the solution is. Chromosomes are then selected with a selection process and mated to create a new generation of the population $P$. Usually, part of the individuals with the highest fitness are kept in the new generation (elitism). Additionally, random
mutation is applied to the individuals, to add variability to the search space. The process is repeated until there is a convergent solution or if the solution is in stall.

Genetic algorithms have been used both in Paper V to solve the inverse problem for localization (Eq. 2.2), and Paper VI to optimize the electrical stimulation patterns for improving stimulation selectivity.

Software There are several available solutions for optimization and evolutionary programming in Python. scipy, for example, provides an excellent optimization module (scipy.optimize). For the implementation of genetic algorithms in Paper V and Paper VI the DEAP (Distributed Evolutionary Algorithms in Python) package has been used, because of its simple and flexible API.

4.4 Scientific programming in Python

In the development of the methods presented in this thesis, I had to decide which programming language to use for software development. With my engineering background, I was mainly trained in MATLAB and C++. While C++ is considered too low level for standard scientific programming, despite my familiarity with MATLAB, I opted for Python for the following main reasons:

1. Python is open-source. It is accessible to any individual and, contrary to MATLAB, it is totally free.

2. Python can seamlessly run on several operating systems and platforms.

3. The language has a clean and versatile syntax, allowing for purely procedural coding or object-oriented approaches.

4. Python is an interpreted language (similarly to MATLAB), which allows rapid and interactive prototyping.

5. There is a huge variety of packages for scientific computing and visualization. Among those, the mostly used packages throughout this thesis are numpy (numerical computations), scipy (scientific computing and signal processing), scikit-learn (machine learning), matplotlib and seaborn (visualization), and pandas (data structures and statistics).

6. Sharing of packages in Python is very easy, using the PyPi package manager.

7. Python can be interfaced with several tools for communication and documentation of software, such as jupyter notebooks and sphinx.

---

*www.python.org/pypi
†http://www.sphinx-doc.org/
4. Engineering solutions

In addition to field-agnostic motivations, Python has gained a large popularity in the neuroscientific community over the past few years. Moreover, the field is pushing for a full commitment to open-source. I strongly believe that open science, both in terms of open-source software and hardware resources and open access to research and journals, is truly beneficial for the progress of science.

Here are the source code repositories developed for the papers presented in this thesis:

- **Paper I**: https://github.com/alejoe91/MEArec
- **Paper II**: https://github.com/SpikeInterface/spikeinterface
- **Paper III - Paper IV**: https://github.com/alejoe91/spyica
- **Paper V**: https://github.com/CINPLA/NeuroCNN
- **Paper VI**: https://github.com/alejoe91/MEAutility
- **Paper VII**: https://github.com/MiroK/nEuronMI
Chapter 5

Summary of Papers

**Paper I**

This paper presents MEArec, an open-source Python package for simulating extracellular spiking activity. MEArec is designed to meet the requirements of ease-of-use, speed, controllability, and biophysical detail that we identify as key features of a simulator for development and validation of spike sorting software. The simulation is split in two phases. A templates generation phase builds a template library from biophysically detailed cell models. A recording generation phase selects suitable templates and combines them with random spike trains in customized convolution, capable of reproducing critical aspects of extracellular recordings such as bursting and drifting. Moreover, the user can control the rate of spatio-temporal overlapping events and there are several noise models available to test the robustness of algorithms against noise assumptions. MEArec is fully interfaced with SpikeInterface (Paper II) for benchmarking available software and it is used by SpikeForest\(^{†}\), an interactive website for benchmarking spike sorting algorithms.

**Paper II**

In this article we introduce SpikeInterface, a unified framework for spike sorting. SpikeInterface is designed to tackle the community needs for standardization in spike sorting, which can mine reproducibility in the analysis of extracellular recordings. SpikeInterface is a collection of Python packages for: i) loading and writing spike-sorting-relevant information from several file formats, ii) running numerous available spike sorters with a standardized API (in a single line of code), iii) providing a processing toolkit for pre-, post-processing, validation of results, and automatic/manual curation, iv) performing comparison of spike sorting output with known or unknown ground-truth information; and v) visualizing every step of the electrophysiology pipeline with efficient plotting routines.

\(^{†}\)spikeforest.flatironinstitute.org
5. Summary of Papers

Additionally, we implemented a graphical user interface to build analysis pipelines. In the paper, we present an overview of the framework and sample applications for analyzing experimental Neuropixels data with several sorters (resulting in large disagreements) and for benchmarking spike sorting software on a high-density MEArec [Paper I] recording.

**Paper III**

In this conference contribution we present a fully automated algorithm for spike sorting based on independent component analysis (ICA). While the use of ICA had been investigated before for spike sorting, new recording technologies with a very high electrode density are particularly suitable for ICA processing. In this paper, therefore, we combine ICA with a selection of spiking sources, amplitude clustering in the ICA space, and removal of duplicate spike trains in a fully automatic approach. We compare our approach with two other popular automatic spike sorting algorithms (Mountainsort [54] and SPyKING-CIRCUS [58]) using initial versions of what would become MEArec [Paper I] and SpikeInterface [Paper II].

**Paper IV**

This conference contribution builds upon [Paper III] and it tackles real-time spike sorting. We approach this problem by applying an online version of the ICA method, the Online Recursive Independent Component Analysis (ORICA), which was originally developed for automatic artifact removal in EEG recordings [88–90]. The online pipeline is semi-automatic and it is very similar to the offline method presented in [Paper III]. After an ORICA pre-processing step and an automatic selection of spiking sources, the user needs to set thresholds on single independent components for spike detection. Even in this case, we benchmark the approach on simulated data with known mixing matrices, in order to validate the performance of estimating the ICA model in an online setting.

**Paper V**

In this article we target localization and classification of neurons from extracellular recordings. We present an approach that combines biophysical forward modeling and deep learning. In brief, we simulate a large dataset of extracellular action potentials (EAPs) from around 200 cell models from the Blue Brain Project [35]. Then, knowing the ground-truth position and type of the simulated neurons, we train convolutional neural networks (CNNs) to predict i) the 3D soma locations and ii) the cell type (excitatory-inhibitory) from the EAPs. We investigate the localization and classification performance depending on different probe designs, CNN sizes, and selected features. We compare the performance of our CNN model with state-of-the-art inverse methods for cell localization and clustering techniques for cell classification and show that
our deep learning approach is superior in both cases. Finally, we validate the trained models on simulated EAPs from other databases, e.g., the Allen Brain Institute, and on experimental data that use paired extracellular and patch clamp recordings.

**Paper VI**

In this conference contribution, we investigate a model-based optimization approach for selective electrical stimulation of single cells using high-density MEAs. Assuming that we can extract the position of the soma and the direction of the axon hillock from the extracellular action potentials, we use this information to optimize the spatial patterns of the currents injected by the electrodes in order to increase selectivity of the stimulation. We predict the excitation of a neuron based on the so-called activating function (Eq. 3.11) and we construct a multi-objective optimization problem that aims to i) only stimulate a target neuron (selectivity) and ii) use the least number of electrodes and the lowest stimulation currents (efficiency). We use a genetic algorithm to solve the optimization problem and compute the optimal currents. Using a Monte-Carlo approach, we show that this model-based approach outperforms standard monopolar and bipolar stimulation paradigms.

**Paper VII**

In this paper we investigate the effect of neural probes on the extracellular signals. While the standard modeling framework for computing extracellular potentials from neuronal currents assumes an infinite and homogeneous milieu, here we employ advanced finite element methods (FEM) to study the effect of several kind of probes on the extracellular potentials. We simulate a single action potential from a simple ball–and–stick neuron and its extracellular signature with and without a neural probe in the extracellular space. We show that for microwires/tetrodes the effect of the probe is negligible. Conversely, larger MEAs strongly affect the recorded signals due to their insulating properties. The spike peaks almost double when the probe is explicitly modeled, and the peak ratio between the potential with and without the probe is relatively constant with respect to the distance from the probe. However, the ratio is dependent on the lateral alignment and relative rotation between the probe and the neuron. Finally, we show that the probe effect is electrode-dependent and we suggest an efficient method, namely probe correction method, to correct for the probe effects. This method uses FEM simulations to pre-map the effect each electrode on the extracellular space and then leverages the reciprocity and superposition principles to compute extracellular potentials from neuronal currents.
“Sitting on your shoulders is the most complicated object in the known universe.”
— Michio Kaku

### 6.1 Towards next-generation electrophysiology

In recent years, the opportunity to conduct electrophysiological recordings has been undergoing a revolution. Developments in technology for the fabrication of neural probes have enabled the creation of next-generation devices which can record from hundreds \((in \, viva)\) and thousands \((in \, vitro)\) of channels simultaneously. With these high-density multi-electrode arrays (HD-MEAs) we are now able to measure the activity of hundreds of neurons simultaneously, even at the sub-cellular level. However, next-generation devices introduce novel grand challenges and the need for appropriate tools to handle the rich information that can be recorded. The work presented in this thesis has therefore focused on developing and benchmarking new tools and methods for using such devices.

The literature on electrophysiological analysis targeting spiking signals has mainly focused on extracting individual spike trains from recordings (using spike sorting - [Section 2.1]). However, HD-MEAs pose new challenges for spike sorting and a large part of the presented work aimed to benchmark and compare existing solutions ([Paper I] and [Paper II]), as well as to develop spike sorting methods specifically targeting HD-MEAs ([Paper III] and [Paper IV]).

In addition to the high yield in terms of number of recorded neurons, the abundant spatial information available from HD-MEAs can be exploited for localizing the 3D position of the recorded neurons and accurately classifying their neuronal type ([Paper V]). Extracellular electrophysiology can therefore increasingly be regarded as a *functional electrical imaging* technique, from which one can reconstruct a 3D map of the recorded neuronal microcircuit. Additionally, precise localization and optimized stimulation strategies ([Paper VI]) enable advanced...
6. Discussion

closed-loop experiments, allowing for a precise bi-directional manipulation of neural circuits.

The number of neurons that can be recorded by HD-MEAs is unprecedented. For example, in a recent study\[202\], the authors report data from around 30'000 neurons acquired from 39 sessions in 10 mice. Given this immense amount of information, one question arises: is spike sorting needed at all?

Some novel approaches present spike-sorting-free techniques to infer the dynamics of the entire recorded population\[203–206\]. The idea of these studies is to find neural manifolds, i.e., lower-dimensional representations of the ensemble neural activity, connected to behavior, hence bypassing the need for spike sorting. While the approach is interesting for specific applications, such as motor control\[207\] and brain-machine interfaces\[208\], I believe that the reign of spike sorting will never be overturned. In fact, spike sorting is a required step to study how single neurons respond to stimuli, movements, or encode specific memories. Without spike sorting, we would not know that neurons in visual cortex are orientation-selective\[209\], that some neurons in the hippocampus only fire in a specific place\[210\] while others in the entorhinal cortex in a hexagonal grid pattern\[211,212\], and that some neurons like Jennifer Aniston\[213\].

Another mean for measuring neuronal activity is through imaging, which one might imagine could replace electrophysiology. Instead of inserting a probe in the brain to pick up electrical signals generated by the neurons, one can image fluorescent markers linked to sensors that (directly or indirectly) reflect neural activity. For example, two-photon calcium imaging has been used to image neural activity for decades\[214\]. Alternatively, voltage sensitive dyes\[215\] (VSDs) and, more recently, genetically-encoded voltage indicators\[216–218\] (GEVIs) are proxies for the neuronal membrane potential signals. While the kinetics of calcium indicators and VSDs are considered to be fairly slow with respect to the membrane potential dynamics, recently developed GEVIs can image neuronal spiking activity and sub-threshold dynamics at very high speed. Imaging techniques are capable of recording tens of thousands of neurons simultaneously\[219\], hence falling deservedly into the definition of next-generation tools. In addition, a combination of simultaneous imaging and manipulation of activity using optogenetics and holographic stimulation techniques\[220\] enables a precise and bi-directional manipulation of the neural tissue with unprecedented levels of accuracy, via so-called all-optical investigations of neural circuits\[221,222\]. However, imaging comes with challenges and limitations. Most importantly, current microscopes have a depth-limitation, hence deeper structures cannot be imaged. Nevertheless, multi-photon systems could alleviate this problem\[223\]. The second limitation of imaging techniques comes from the fact that most setups require a bulky and expensive microscope and therefore behavioral studies are mainly restricted head-fixed mice. Despite the possibility of setting up sophisticated behavioral settings, including virtual reality systems\[224\], experiments on freely moving animals arguably represent a more natural setting to observe neuronal correlates of animal behavior. Notably, portable and wireless imaging systems exist, such as the UCLA miniscope\[225\], but their capabilities are not comparable to fixed microscopy systems.
HD-MEA devices, instead, can be used both for head-fixed and freely moving animals, using chronic and recoverable implants. They enable to perform very high-yield experiments, with high spatio-temporal resolution recordings from hundreds of neurons per experimental session from several structures of the brain and they are opening a brand new era for neurophysiology. Moreover, HD-MEA devices are currently being used by large-scale international collaborations and will generate an unprecedented amount of neural data from all known and unknown brain regions, which will greatly increase our understanding of how the brain works.

6.2 Computationally-assisted electrophysiology: benefits and limitations

Modeling and simulations have been central in the presented work to aid the development of methods for extracellular electrophysiology. In recent years, large international consortia have joined forces and dedicated a huge amount of resources to build detailed biophysical models of neurons. In 2015, the Blue Brain Project released the Neocortical Microcircuit Portal with more than 30'000 cortical cell models. In parallel, the Allen Institute for Brain Science is constantly expanding its cell-type database with cell models reconstructed from mice and even from human tissue.

The large and growing availability of such detailed models has created a new opportunity in the development of analysis methods for neuroscience. Simulations can be used to develop model-informed, or computationally-assisted tools to solve specific problems in electrophysiology. In other words, models can provide a-priori and ground-truth knowledge that can be incorporated in the methods themselves. In this thesis, modeling served three main purposes: ground-truth data simulation, validation, and prediction.

In Paper V we used ground-truth simulations in combination with machine learning and showed that supervised localization and classification approaches outperform unsupervised solutions. Simulations can in fact provide a virtually infinite amount labeled datasets, which can be used to train machine learning algorithms.

In Paper I a biophysically detailed simulator for extracellular spiking activity was developed specifically to aid the development and validation of spike sorting algorithms. Although also spike sorting could be tackled in a supervised manner, by learning a model that outputs spike times of different neurons using the extracellular signals as input, this approach would need to learn very complicated spatio-temporal features in the recordings, and at present it has not been attempted.

In Paper VI modeling was used as a predictive tool for the outcome of electrical stimulation. In combination with genetic algorithms used for optimization, we were able to find optimal stimulation patterns that improved the stimulation selectivity.
6. Discussion

Finally, a model-based approach would enable users to tune the methods to diverse applications. Neurons from different brain regions or species can be very different in terms of morphology, electrical properties, and firing patterns. Hence, analysis tools targeting specific applications, for example, hippocampus recordings in mice, could be enhanced by using cell models from the selected brain region and animal species.

There are also some limitations to the modeling framework used throughout this thesis that need to be discussed. First and foremost, simulations are not real. A mathematical representation of a real entity, no matter how complicated, should always be regarded as a simplified and limited representation of the entity. However, recent developments in computational systems and recording technologies make the simulations from cell models used in this work almost indistinguishable from real data. Koch and Buice\(^43\) in a letter accompanying the publication of the Blue Brain Project simulation\(^32\) state that these simulations would pass a biological Turing test, given their ability to accurately replicate neuronal dynamics. Still, these detailed cell models have intrinsic limitations.

Most of the electrophysiological data used to construct biophysically detailed cell models comes from patch-clamp somatic voltage traces. However, neurons are more than just the soma. Neuronal dynamics are greatly affected by non-somatic cell compartments, such as dendrites and axon\(^229\). Modeling studies using data from multiple patch pipettes along the apical dendrites to fit multi-compartment models\(^30,31\) were capable of replicating complex active properties of the dendrites, such as backpropagating action potentials and calcium spikes, which affect the extracellular signals. However, somatic patch-clamp recordings alone are not enough to constrain cell models that replicate these features\(^30\). Moreover, extracellular action potentials are largely influenced by the axonal initial segment (AIS)\(^113,114\), which should therefore be carefully considered in the models. Cell models from both the Blue Brain Project and the Allen Institute, however, mainly use somatic patch-clamp recordings alone to collect data used to construct their cell models. Therefore, active dendritic properties cannot be reproduced. Moreover, in those models the entire axon is replaced by a single stereotyped axonal segment, hence the contribution of the AIS to the extracellular signals might be distorted.

Second, in the calculation of extracellular signals I have mainly used the framework described in Section 3.2 (Paper I, Paper III, Paper IV, Paper V and Paper VI). The volume conduction theory (Eq. 3.9) assumes that the extracellular space is isotropic, homogeneous, linear, and infinite. However, neural tissue is not isotropic\(^153\), mainly due to the preferential orientation of pyramidal cells. Anisotropy could be easily included in the calculation of extracellular potentials with analytical solutions\(^41,154\). Second, the tissue surrounding the neurons is not homogeneous, because of the presence of the neural probes. In Paper VII we therefore investigated how neural probes affect the recordings. We showed that the probe does affect the recorded potentials in a non-trivial way, and proposed an efficient solution that involves a finite element pre-mapping of the effect of the probe in the extracellular space. While this solution is technically feasible and could be incorporated into current simulators\(^151-159\), it would require
one to embed finite element solutions in the software, making it slower and more complicated to handle. Alternatively, the finite element solutions could be approximated by analytical functions and used to efficiently pre-map the effect of different probes in the extracellular space.

Finally, regardless the above-mentioned limitations, I believe that the presented computationally-assisted approach to electrophysiology has the potential to improve the current status of extracellular electrophysiology research. Methodological improvements and the integration of new techniques for acquiring comprehensive data will make computational models more accurate and realistic. As a result, analysis tools built upon simulations will also improve their performance in tackling common problems in electrophysiology.

6.3 Future developments

Part of the work presented in this thesis is fully finalized, documented, and shared with the community. In particular, the software presented in Paper I (MEArec) and in Paper II (SpikeInterface) are already used in other projects. As an example, the SpikeForest project is aiming to benchmark and compare most of the available spike sorters using ground-truth datasets. SpikeInterface is the engine upon which SpikeForest is running, and several MEArec-generated datasets are used as ground-truth recordings.

Paper III and Paper IV are instead at a proof of concept stage. The software, named SpyICA, is dated before the development of MEArec and SpikeInterface, which were specifically designed to ease the development, validation, and comparison of spike sorting algorithm. Therefore, a future effort will tackle the extensive characterization of SpyICA in comparison with state-of-the-art spike sorters, both for offline and online use. Moreover, as mentioned in Section 2.1, the current implementation is not capable of handling spatio-temporal overlapping units. Therefore, a hybrid ICA and template-based approach could be investigated for this purpose. In particular, adding a template-matching step to the ICA-based pipeline would both facilitate the correct identification of spatio-temporal collision and alleviate the problem related to duplicate units in different independent components mentioned in Section 2.1.

In Paper V we performed a thorough analysis on the capabilities of supervised approaches in electrophysiological analysis. Although the source code is open and shared online (see Section 4.4), it consists of a series of scripts rather than a compact package. Moreover, it was developed using an old version of the Tensorflow package, which has been enormously improved and simplified over the last year. Therefore, I plan to make the software more usable and plug-and-play and to update the code to the newest Tensorflow version.

In Paper VII we used a very simplified model of a neuron, a ball–and–stick. While our purpose was to characterize the effect of extracellular probes, in the future, similar detailed finite element simulations could make use of more

*spikeforest.flatironinstitute.org*
6. Discussion

complicated and realistic morphologies, which are widely available on online databases.

Finally, the performance of spike sorting, localization, classification, and stimulation methods has been assessed using simulated data alone. Nevertheless, experimental validation is an important and required step. In this light, validation datasets can be acquired by a combination of multiple recording techniques, for example combining extracellular HD-MEAs, patch-clamp, and imaging techniques. Multi-modal setups with extracellular MEAs and imaging can provide ground-truth information on the neuronal spiking activity, location, cell type, and stimulation outcome.

6.4 Outlook

In conclusion, in this thesis I presented several methods for a computationally-assisted approach targeting neural extracellular electrophysiology for multi-electrode arrays. With the combination of state-of-the-art modeling and engineering tools, this work has introduced a series of next-generation analysis tools to handle newly developed and powerful neural devices, which will contribute to a better understanding of our fascinating brain.


60. Pachitariu, M., Steinmetz, N. A. & Colonell, J. Kilosort2 [https://github.com/MouseLand/Kilosort2](https://github.com/MouseLand/Kilosort2).


82. Rossant, C., Kadir, S., Goodman, D., Hunter, M. & Harris, K. *Phy* [https://github.com/cortex-lab/phy](https://github.com/cortex-lab/phy)


Bibliography


Papers
Combinig biophysical modeling and deep learning for multielectrode array neuron localization and classification
INNOVATIVE METHODOLOGY | Neural Circuits

Combining biophysical modeling and deep learning for multielectrode array neuron localization and classification

Michael Kordovan,1,5* Torbjørn V. Ness,2 Benjamin Merkt,3,4 Philipp D. Häfliger,1 Marianne Fyhnc,1 Gert Cauwenberghs,2 Stefan Rotter,3,4* and Gaute T. Einevoll1,5*

1Center for Integrative Neuroplasticity (CINPLA), Faculty of Mathematics and Natural Sciences, University of Oslo, Oslo, Norway; 2Department of Bioengineering, University of California, San Diego, California; 3Bernstein Center Freiburg, Freiburg, Germany; 4Faculty of Biology, University of Freiburg, Freiburg, Germany; and 5Faculty of Science and Technology, Norwegian University of Life Sciences, Ås, Norway

Submitted 28 March 2018; accepted in final form 29 May 2018

Buccino AP, Kordovan M, Ness TV, Merkt B, Häfliger PD, Fyhnc, Cauwenberghs G, Rotter S, Einevoll GT. Combining biophysical modeling and deep learning for multielectrode array neuron localization and classification. J Neurophysiol 120: 1212–1232, 2018. First published May 30, 2018; doi:10.1152/jn.00210.2018.—Neural circuits typically consist of many different types of neurons, and one faces a challenge in disentangling their individual contributions in measured neural activity. Classification of cells into inhibitory and excitatory neurons and localization of neurons on the basis of extracellular recordings are frequently employed procedures. Current approaches, however, need a lot of human intervention, which makes them slow, biased, and unreliable. In light of recent advances in deep learning techniques and exploiting the availability of neuron models with quasi-realistic three-dimensional morphology and physiological properties, we present a framework for automatized and objective classification and localization of cells based on the spatiotemporal profiles of the extracellular action potentials recorded by multielectrode arrays. We train convolutional neural networks on simulated signals from a large set of cell models and show that our framework can predict the position of neurons with high accuracy, more precisely than current state-of-the-art methods. Our method is also able to classify whether a neuron is excitatory or inhibitory with very high accuracy, substantially improving on commonly used clustering techniques. Furthermore, our new method seems to have the potential to separate certain subtypes of excitatory and inhibitory neurons. The possibility of automatically localizing and classifying all neurons recorded with large high-density extracellular electrodes contributes to a more accurate and more reliable mapping of neural circuits.

NEW & NOTEWORTHY We propose a novel approach to localize and classify neurons from their extracellularly recorded action potentials with a combination of biophysically detailed neuron models and deep learning techniques. Applied to simulated data, this new combination of forward modeling and machine learning yields higher performance compared with state-of-the-art localization and classification methods.

INTRODUCTION

The neural circuits of the brain involve the interplay of many different types of neurons. On the most superficial level, neurons are classified by their effect on the neurons they project to as either excitatory or inhibitory. Extracellular recordings enable us to measure the activity of neurons as electrical potential deflections mainly due to ionic transmembrane currents. In recent years, many groups have been developing high-density multielectrode arrays (MEAs) for in vitro and in vivo applications, which allow measurements of neuronal activity with high spatiotemporal resolution (Berndondini et al. 2014; Müller et al. 2015; Neto et al. 2016; Schröder et al. 2015; Welkenhuysen et al. 2016). These probes provide something close to a functional electrical imaging (Vassanelli 2014) of the neural activity, and this high information density can potentially be exploited to obtain a better understanding of the neural circuits under study. Specifically, it might be possible to extract information that could be used to localize the individual neurons and to classify them into their respective cell types. The latest developments in manufacturing of high-density neural probes call for novel analysis methods to exploit the richness and detail in the recordings.

Neural localization from extracellular action potentials (EAPs) recorded on a MEA is an ill-posed problem, since solutions might not be unique for complex neural morphologies. Current approaches for localization assume simple neuronal models to facilitate the inverse problem and make the solution unique. Examples of models used in previous studies are monopole current-source models (Blanche et al. 2005; Chelaru and Jog 2005; Kubo et al. 2008), dipole current-source models (Blanche et al. 2005; Mechner et al. 2011; Mecherl and Victor 2012), as well as line-source models (Somogyvari et al. 2012). More recently, Delgado Ruz and Schultz (2014) used a modified ball-and-stick model (Pettersen and Einevoll 2008) to predict somatic locations. In these approaches, the soma posi-
tion is estimated by minimizing the error between the electrical potential on the MEA sites predicted by the chosen model and the recorded potential. However, it is experimentally challenging to measure the correct position of the soma (Neto et al. 2016); therefore, detailed computational neuronal models are usually used and treated as simulated ground truth to evaluate the accuracy of the localization methods (Delgado Ruz and Schultz 2014; Somogyvári et al. 2005, 2012). The main limitations regarding neuron localization are that the models chosen to solve the inverse problem are often too simple to grasp complex spike waveforms (e.g., monopolar or bipolar current-source models) or are tuned to specific cell types (ball-and-stick model for pyramidal morphology).

Regarding classification of neurons, unsupervised clustering techniques are commonly applied to differentiate EAP shapes (Barthó et al. 2004; McCormick et al. 1985; Peyrache et al. 2012); narrow waveforms are considered to be fast-spiking inhibitory neurons and broad waveforms excitatory neurons. In this case, it is experimentally challenging, especially in vivo, to validate the classification methods. One approach is to measure a multitude of neurons and find putative monosynaptic connections based on the shape of spike train cross-correlograms. However, the rate of recorded monosynaptic connections is usually very low (~0.2%; Barthó et al. 2004; Peyrache et al. 2012), resulting in a small number of observations useful for validation. In neural classification, the complexity of spike shapes across the multiple recording sites is usually compressed by extracting spike widths (such as peak-to-peak and full-width half-maximum widths; Barthó et al. 2004; Peyrache et al. 2012) only from the electrode with the highest recorded amplitude.

In this study, we apply a powerful machine learning technique, namely, convolutional neural networks (CNNs), to classify excitatory and inhibitory neurons and to localize their somata from simulated EAPs. This approach—being a supervised learning algorithm—demands for a large amount of labeled data, in this case EAPs in combination with soma position and cell type of the neuron evoking the EAPs. The proposed method is schematically depicted in Fig. 1. First, compartmental cell simulations are performed (Fig. 1A) that yield EAP data sets with known simulated ground truth (forward modeling) (Fig. 1B). Relying on the simulations, CNNs are trained (Fig. 1C) on these data sets to predict position and cell type (Fig. 1D) of the neuron generating the simulated EAP. If the method is applied to experimental data (Fig. 1E), a spike-triggered average EAP (average waveform) serves as input to a CNN previously trained on simulated data to predict soma position and cell type. In addition to binary classification, we attempt to distinguish 11 different morphological types (m-type classification). The performance of the CNNs depending on different characteristics extracted from the EAP, different MEA designs, and different CNN configurations is explored. Finally, we evaluate the effect of varying neuron alignments with respect to the recording MEA. To put our approach into context, we compare its performance with established methods of cell localization and classification.

CNNs perform supervised machine learning and require a large data set to be trained. It would be experimentally challenging, if not unfeasible, to gather the required number of recordings of exact known position (used for localization) and morphology (used for classification) information. Therefore, we rely on detailed compartmental cell models to provide detailed simulated recordings and simulated ground-truth information. Forward biophysical modeling of extracellular potentials has been developed and refined over the last 30 years (Gold et al. 2006; Holt and Koch 1999; Lindén et al. 2014; Schultz 2014; Somogyvári et al. 2005, 2012). The main limitations regarding neuron localization are that the models chosen to solve the inverse problem are often too simple to grasp complex spike waveforms (e.g., monopolar or bipolar current-source models) or are tuned to specific cell types (ball-and-stick model for pyramidal morphology).

![Graphical representation of the presented method. Red arrows show our approach for training (dashed lines indicate error backpropagation) and validating the convolutional neural network (CNN) on simulated data. Green arrow shows how the method is applied within an experimental pipeline. Starting with the red path, biophysical simulations (A) are used to generate extracellular action potential (EAP) templates (B), from which features (e.g., amplitude and width; see Classification and Localization Features) are extracted and fed to a CNN (C) to localize and classify excitatory (blue) and inhibitory (red) neurons (D). When applied to experimental data (green arrow), recordings are first spike sorted (E), then features are extracted from spike-triggered average waveforms (B), and finally the CNN trained on simulated data (C) is used to localize and classify spike-sorted neurons (D).](journals.physiology.org/doi/abs/10.1152/jn.00210.2018)

*J Neurophysiol* • doi:10.1152/jn.00210.2018 • www.jn.org

Downloaded from www.physiology.org/journal/jn (005.102.006.141) on July 8, 2019.
In principle, the data set contains 13 types of morphologies (m-types) in L5, in accordance with the NMC Portal. However, because of the limited data variety in the case of bipolar and neuroglial cells (BP and NGC), we excluded them from the analyses unless elsewhere specified. Appendix A describes the data set in detail and explains how we manipulated the original data set to extract unbiased sub-data sets for training and validating the results. A single-cell morphology of each m-type is displayed in Fig. 2, divided into inhibitory and excitatory neurons. Axons are not depicted because only their initial tract is included in the simulations.

Simulated Recordings

Extracellular action potential computation. Each of the compartment neuronal models was simulated separately, and extracellular potentials were calculated in two steps. First, transmembrane currents were computed by solving the cable equation (Holt and Koch 1999) with LFPy (Lindén et al. 2014) running on NEURON 7.4 (Carnevale and Hines 2006; Hines et al. 2009). A constant depolarizing current was applied to the soma to get the neuron firing at least 10 times (and not more than 30 times) in a simulation period of 1.2 s. Multiple spikes were simulated to account for spike-to-spike variations due to electrophysiological properties (e-types). All transmembrane currents for each compartment were stored within a time window \( t = 2 \) ms and \( t = 5 \) ms, where \( t = 0 \) is the time of the intracellular membrane voltage peak considered as spike time. Simulations were run with a time resolution of \( \Delta t = 0.1 \) ms, i.e., with a sampling frequency of 10 kHz, so that a single spike window of 7-ms duration (2 ms + 5 ms) consists of 224 samples.

Second, transmembrane currents were used within LFPy to calculate the extracellular potential at the recording site. Each transmembrane current, including the somatic one, was distributed over a line source with the length of its corresponding neural segment. With quasi-static approximation (Nunez and Srinivasan 2006) and assuming a homogeneous and isotropic neural tissue with conductivity \( \sigma = 0.3 \) S/m (Goto et al. 2010), the contribution of each compartment \( i \) at position \( r_i \) with transmembrane current \( I_i(t) \) to the electric potential \( \phi \) on an electrode at position \( r_e \) reads (Holt and Koch 1999; Lindén et al. 2014; Pettersen and Einevoll 2008):

\[
\phi(r_e, t) = \frac{1}{4\pi\sigma} \int \frac{I_i(t)}{|r_e - r_i|} \, dt \tag{1}
\]

Fig. 2. One representative cell for each different morphology type in layer 5 [data from Neocortical Microcircuit Collaboration Portal (Ramaswamy et al. 2015)]. Top: 9 inhibitory cells (red). Bottom: 4 excitatory cells (blue). Dendrites are colored in lighter shades, and the soma is indicated with a darker circle. The same red/blue convention is used throughout article. For all cells C stands for cell, and for excitatory cells the P represents pyramidal.©
The simulated EAP was obtained by summing up the contributions of all compartments. For each recording site, the electric potential was computed at a single point corresponding to the center of the recording electrode. These EAPs are associated with the templates in Fig. 1B.

Only spiking generates a notable waveform with a peak-to-peak amplitude exceeding 30 µV on at least one of the electrodes were included in the data set. The detection threshold of 30 µV was chosen to mimic experimental settings, where noise in the recordings due to equipment electronics and background neural signals does not allow detection of low-amplitude action potentials.

The coordinate system was fixed in reference to the MEA plane. Each recording site (different MEA designs are explained in MEA designs) lay within the \( \text{yz} \)-plane, and neuron somata were located within the semispace of the positive \( x \)-axis (the \( x \)-coordinate is thus the distance from the MEA). For each neuron, 1,000 EAP recordings above the detection threshold were simulated by randomly choosing one of the generated spikes and by placing the soma at random locations with distances \( x \) between 10 µm and 80 µm. The \( y \) and \( z \) boundaries were different for each MEA, and a neuron could exceed the \( y \) and/or \( z \) boundary of the MEA by a maximum of 30 µm. For the EAP, we considered the contributions of all dendritic compartments, including those crossing the MEA. This was done to force the sum of transmembrane currents to be zero (Pettersen and Einevoll 2008). In other words, we did not clip neurites entering the probe, but we made sure that their position did not coincide with a recording site within \( \text{LIFy} \).

**MEA designs.** We investigated the performance of seven different MEA probes. Five of these were based on the prototype described in Schröder et al. (2015), in which the recording sites are arranged in a 16 \( \times \) 16 matrix on a penetrating shank. Instead of considering a single interelectrode distance, we investigated five different pitches (i.e., distance between the centers of adjacent electrodes), namely, 10, 15, 20, 25, and 30 µm. The probe models were built in a way that they roughly covered the same area on the shank.

Hence, we ended up with the following list of squared probes:

- **SqMEA-10-15:** squared MEA with 10 \( \times \) 10 electrodes and 10-µm pitch;
- **SqMEA-15-10:** squared MEA with 15 \( \times \) 15 electrodes and 10-µm pitch;
- **SqMEA-20-15:** squared MEA with 20 \( \times \) 20 electrodes and 15-µm pitch;
- **SqMEA-30-15:** squared MEA with 30 \( \times \) 30 electrodes and 15-µm pitch;
- **SqMEA-15-30:** squared MEA with 15 \( \times \) 30 electrodes and 30-µm pitch;
- **SqMEA-30-10:** squared MEA with 30 \( \times \) 10 electrodes and 10-µm pitch;
- **SqMEA-30-20:** squared MEA with 30 \( \times \) 20 electrodes and 20-µm pitch;
- **SqMEA-6-25:** squared MEA with 6 \( \times \) 6 electrodes and 25-µm pitch;
- **SqMEA-5-30:** squared MEA with 5 \( \times \) 5 electrodes and 30-µm pitch.

Moreover, we simulated recordings on the Poly3-25s probe (Neuronexus Technologies), which has 32 electrodes in three columns with a hexagonal arrangement, a \( y \)-pitch of 18 µm, and a \( z \)-pitch of 22 µm. Another probe becoming popular is the NeuroPixels probe (Jun et al. 2017), with recording sites arranged in a checkerboard pattern with a \( y \)-pitch of 32 µm and a \( z \)-pitch of 20 µm. Although the probe has 384 recording channels, for convenience we decided to trim it to 128 channels. Finally, we constructed a model of the NeuroSeeker probe (http://www.neuroseeker.eu; used in Neto et al. 2016), a MEA with 128 recording sites arranged in a \( 4 \times 32 \) configuration and a regular interelectrode distance of 22.5 µm. Figure 3 shows all the probes in the \( \text{yz} \)-plane.

The CNNs we used required a rectangular shape of the input data. The two dimensions of the data array correspond to the number of electrode sites \( N_y \) and \( N_z \) in \( y \)- and \( z \)-directions, respectively. Therefore, we cut the NeuroNexus-32 MEA probe to a NeuroNexus-30, which is shown in the fourth position from the left in Fig. 3.

**Neuron-MEA alignment.** We investigated different neuron-MEA alignments (or rotations) of neurons. Some neurons, such as pyramidal cells (PC) or bipolar cells (BP), have morphologies that follow a specific orientation (see Fig. 2) that might affect the classification and localization performance. For this reason, we generated three rotational data sets:

1. **Norot:** The orientation of the cells (e.g., the apical dendrite of PCs) was along the \( z \)-axis (same orientation as in Delgado Ruiz and Schultz 2014 and Somogyvári et al. 2012).
2. **Physrot:** Neurons with a preferential orientation were randomly rotated such that after rotation their axis from white matter toward the pia pointed into a cone around the \( z \)-direction with an opening angle of 15° (the puncture point on the unit sphere is uniformly distributed in this spherical cap). Neurons without a preferential orientation were rotated randomly in the three-dimensional (3D) space. We considered all

![Fig. 3. Multielectrode array models used in the study. Right: plots show an excitatory cell [thick-tufted pyramidal cell (PC) with late bifurcating apical tuft (TTPC1)] and an inhibitory cell [neuroglial cell (NGC)] to compare probe and neuron sizes. PCs are on average much larger than inhibitory cells, and only a portion of the neuron is located directly in front of the probe (the apical dendrite is not fully shown, and it can be seen in Fig. 2).](http://www.physiology.org/journal/jn/005.102.006.141)
neurons apart from nest basket cells (NBC), small basket cells (SBC), and NGC to have a preferential orientation (DeFelipe et al. 2006; Markram et al. 2004; Overstreet-Wadiche and McBain 2015; Wang et al. 2002, 2004; Woodruff and Yuste 2008). NBC, SBC, and NGC were assumed to have no preferential axis.

3drot: Neurons were rotated randomly around all axes.

The soma positions corresponded to the intersection point of rotation axes and were shifted randomly inside the observation volume in all cases. Figure 4 displays a sample orientation with respect to the MEA of a PC [thick-tufted PC with late bifurcating apical tuft (TTPC1)] in each of the three data sets, Norot (Fig. 4A), Physrot (Fig. 4B), and 3drot (Fig. 4C).

Classification and Localization Features

We extracted features from the EAPs as input variables to a CNN for training. Since classification and localization of neurons from extracellular recordings are quite different tasks, we used different sets of features from the simulated spikes.

The pipeline for extracting feature images is described in Fig. 5. First, neurons with known cell type and position were simulated and the spike traces on the MEA probe were obtained. Then, for each spike, a set of features was computed and these features were rearranged in a 2D shape according to the MEA arrangement, i.e., the feature image. In the following paragraphs $N$, $N_z$, and $N_y$ are the total number of electrodes, the number of electrodes in $y$-direction, and the number of electrodes in $z$-direction, respectively.

Localization features. The goal of localization is to estimate the soma position with respect to the probe. Therefore, we considered only the EAP negative peak and the positive peak time points, during which negative and positive transmembrane currents are larger in proximity of the soma (Delgardo Ruiz and Schultz 2014; Gold et al. 2007; Somogyvári et al. 2005, 2012). For simplicity, we refer to the EAP negative peak as Na peak because, close to the soma, it is mainly attributed to the sodium currents flowing into the soma. The positive peak is referred to as Rep because it is associated with the cell repolarization phase. The peak values were computed with respect to a reference of 0 µV, i.e., the baseline, as follows:

Na. For each spike recording on $N$ electrodes, the spike with the largest negative peak amplitude was identified. At the time instant when the minimum peak occurred ($t_{min}$) the recorded potential on all $N$ electrodes was used to build the Na image (the amplitude values are sampled at the same time instant $t_{min}$).

Rep. The time instant of the repolarization peak ($t_{max}$) was extracted from the spike trace with the largest negative trough (same electrode as Na) and a Rep image was built by probing all $N$ electrodes at $t_{max}$.

Overall, the localization-specific ($N_y, N_z$)-dimensional sets of features are Na, Rep, and NaRep, where the last is a stacked version of both features having dimension ($N_y, N_z$), 2.

Classification features. From each spike, we extracted features that are commonly used for cell classification (Barthó et al. 2004; Peyrache et al. 2012): peak-to-peak width ($W$), full-width half-maximum ($F$), and peak-to-peak amplitude ($A$). The peak-to-peak amplitude $A$, despite not being one of the commonly used features for classifying neurons from extracellular recordings, was selected as a feature in combination with $F$ and $W$ because spike widths increase with increasing recording distance (Anastassiou et al. 2015; Hagen et al. 2015; Petersen and Einevoll 2008) and therefore with decreasing amplitude.

The following is a list including a detailed description of the features:

- $A$: peak-to-peak amplitude of the EAP.
- $W$: full-width half-maximum width.
- $F$: peak-to-peak width.
- $Na$: negative peak amplitude.
- $Rep$: repolarization peak amplitude.
A: For each recording site, the peak-to-peak amplitude was extracted as the absolute difference between the negative peak and the following positive repolarization peak. If the amplitude value of a recording site was less than a detection threshold of 5 μV, then the amplitude for that electrode was set to zero.

W: For each electrode site, the width was computed as the time difference between the negative peak and the following positive repolarization peak. If the amplitude of the corresponding electrode was below the detection threshold (i.e., when the amplitude feature was set to 0), the width was set to the duration of the spike window, which was 7 ms.

F: For each electrode site, the full-width half-maximum was computed as the width at 50% of the negative maximum amplitude (refer to Fig. 5 for a graphical visualization and further explanation). In this case, the reference voltage was the initial voltage on the selected electrode site at beginning of the spike window. If the amplitude of the corresponding electrode was below the detection threshold (when the amplitude feature was set to 0), F was set to the duration of the spike window, which was 7 ms.

For classification, we considered the feature combinations AW (Nx, Ny, Nz, 2), FW (Nz, Ny, 2), and AFW (Nz, Nx, Ny, 3), where the shapes of input arrays to the CNN are indicated in parenthesis.

Waveform. We also investigated the performance using the entire spike waveform as input to the CNNs for localization and classification. While localization and classification features focused on amplitudes at specific time points (e.g., Na, Rep) or on extracting significant spike shape parameters (A, F, W), here we took into account the evolution of the action potential in time. As the additional third dimension (2D + time) increased the training time significantly, we downsampled the initial spike waveforms from 224 to 14 samples, i.e., with a downsampling factor of 16. We denoted this feature set, with a shape of (Nx, Ny, 14), as Waveform.

Convolutional Neural Network

CNNs are biologically inspired artificial neural networks, and they differ from standard artificial neural networks mainly by the use of convolutional layers. The biological inspiration originates from the information processing in the visual system (Krizhevsky et al. 2012; Zeiler and Fergus 2014). For our implementation, we used the open-source software TensorFlow to train and evaluate the network (see Abadi et al. 2016; software available from https://www.tensorflow.org). All computations were done on the HPC clusters NEMO in Freiburg and ABEL in Oslo.

Configuration. We investigated the performance of CNNs of different sizes, all having the same underlying configurations (except for Waveform input features, whose CNN morphology is explained at the end of this section). Five CNN sizes (XX, S, M, L, XL) were used, and they differ in the size k of convolutional kernels (the index i ∈ {1, 2} specifies the convolutional layer), convolutional layer depths di, and the number of nodes in the fully connected layer nfc. The values used for different sizes are listed in Table B1 in Appendix B.

Feature images of dimension (Nx, Ny) were input to a d1-deep convolutional layer with rectified-linear units that filter the input image with (k1, k1) kernels and a stride of 1. Then max-pooling was applied, and the image was shrunk to a (n1, n1) = (N/2, N/2) footprint. Another d2-deep convolutional layer with rectified-linear units applied (k2, k2) kernels, and a second max-pooling operation reduced the output image features to a (n2, n2) = (N/4, N/4) size. The (m1, m1) features were input to a fully connected layer with nfc nodes. The fully connected nodes were connected to the nodes in the output layer (see Output layer and optimization).

The Waveform feature set differed from the classification- and localization-specific sets as it included time as a dimension. Although some feature images for localization and classification were concatenated and thus had a 3D shape (for example, NaRep had a shape of (Nx, Ny, 2) and AFW is (Nx, Ny, 3)-dimensional), the optimized kernels were the same for two or three dimensions. For the Waveform feature set, a 3D CNN was used, i.e., convolutional kernels were 3D with a shape of (k1, k1, k1) and (k2, k2, k2). Max-pooling was also applied in all three dimensions. For the Waveform feature set, we used a CNN with k = 4.

Output layer and optimization. The output layer of the network was different depending on whether localization or classification was performed. In case of localization, three output nodes linearly summed the fully connected node inputs and biases to output the x-, y-, and z-coordinates. Optimization minimized the mean squared error between the predicted x-, y-, and z-coordinates and the true soma positions of the training spikes. For classification, there were instead two output nodes in case of excitatory/inhibitory classification. For the m-type classification, we used 11 output nodes, 1 for each cell type under consideration (see Cell Models for a list of m-types). During training, softmax cross entropy was minimized (Goodfellow et al. 2016).

For both localization and classification we used an Adam optimizer (learning rate = 0.0005) (see Kingma and Ba 2014), and we ran 2,000 training epochs. At each iteration, 10% of the training observations were randomly sampled and used to update network weights with backpropagation. To limit overfitting, we used dropout on the fully connected layer (Srivastava et al. 2014) with a dropout rate of 0.3 (during training 30% randomly chosen fully connected nodes were dropped and not considered for updating the network weights).

Validation strategy. To estimate the accuracy of the CNNs, we divided the input data into a training set, used to estimate the CNN parameters, and a validation set, upon which the trained CNN’s accuracy was tested. Before the training-validation set division, we preprocessed the data set so that morphologies in the training and validation sets were unique (Appendix A). Then, we balanced the occurrence of observations of the same cell type (m-type) by random undersampling. For excitatory/inhibitory classification, we further subsampled the inhibitory neuron observations to match the excitatory ones (in the data set, there are 7 inhibitory cell types—not counting BP and NGC—and 4 excitatory types). The class balancing was performed for training and validation sets separately. For localization (and m-type classification), the training and validation data sets consisted of 44,000 and 11,000 instances, respectively. For classification, we used 32,000 observations for training and 8,000 for validation.

Comparison with Other Models

Localization. In previous work on neural localization, the underlying idea has been to solve the inverse problem by choosing a generative model and minimizing the error between the true extracellular potential and the one predicted by the chosen model. The soma position has been among the model parameters that have been optimized. Several models were used in previous studies: monopole and dipole current-source models (Blanche et al. 2005), line-source models (Somogyvári et al. 2012), and ball-and-stick models (Delgado Ruz and Schulz 2014). We compared our localization approach to inverse problem solutions solved with the EAP negative peak (i.e., Na image) with the following models (σ denotes the extracellular conductivity):

- **MONOPOLAR CURRENT SOURCE.** A negative monopolar current-source I placed at position r evokes a potential \( \phi(r) \) at position r according to

  \[
  \phi(r) = \frac{I_s}{4\pi \sigma |r - r_s|}.
  \]

  The somatic current and soma position are the only parameters to be optimized. The predicted soma position is \( r_s \).

- **BIPOLAR CURRENT SOURCE.** Placing a negative current-source \( I_t \) at position \( r_{imp} \) and its positive counterpart (absolute value of \( I_t \)) at \( r_{imp} \), the potential at position \( r \) reads

- **CONVOLUTIONAL NEURAL NETWORK.**
\[
\phi(r) = \frac{I_c}{4\pi r} \left( \frac{1}{|r-r_{\text{seg}}|} - \frac{1}{|r-r_{\text{pool}}|} \right). 
\]

In this case, the estimated soma position corresponds to the negative current-source location, which is \(r_{\text{seg}}\). This model is equivalent to the two-monopole model in Pettersen and Einevoll (2008).

**BALL AND STICK.** The ball-and-stick model combines a somatic point-like constant-current source \(I_c\) at position \(r_c\) with a dendritic stick of length \(d_{\text{seg}}\) pointing in direction \(d\). We used a modified version of the ball-and-stick model described in Pettersen and Einevoll (2008), since we do not assume net currents to be zero (Delgado Ruiz and Schultz 2014). The current along the stick \(I(s)\) is assumed to decay exponentially, as confirmed by experimental data (Foust et al. 2010; Goldberg and Yuste 2005; Golding et al. 2001; Gulleck and Stuart 2003; Migliore et al. 1999; Sasaki et al. 2012; Waters et al. 2005). With initial negative value \(I_{\text{dend}}\) at \(r_c\), the current distribution along the stick reads

\[
I(s) = I_{\text{dend}} \times \exp \left( \frac{s}{d_{\text{seg}}} \right),
\]

where \(d_{\text{seg}}\) denotes the decay constant and \(s \in [0, d_{\text{seg}}]\) is the position along the dendritic segment (discretized in \(N_{\text{seg}} = 50\) uniformly distributed points along the stick of length \(d_{\text{seg}}\)). The predicted soma location corresponds to \(r_c\). The potential at position \(r\) is given by the summation of the somatic and dendritic contributions:

\[
\phi(r) = \phi_{\text{soma}}(r) + \sum_{i=1}^{N_{\text{seg}}} \frac{I_i}{4\pi r_i} \int_{|r-r_i|} \ dt,
\]

where each segment is modeled as a line current source (see Eq. 1).

Table 1 summarizes the parameters to be estimated for each described model.

**GENETIC OPTIMIZATION.** To estimate the model parameters, we minimized the sum of squared errors at each recording site between the extracellular potential predicted by the model and the extracted Na feature image of the true simulated extracellular potential. Optimization was performed with a genetic algorithm implemented with the Distributed Evolutionary Algorithms in Python (DEAP) package (Fortin et al. 2012). We used the \((\mu + \lambda)\) evolution strategy, which selects the next parents from the common set of the current parents (\(\mu\) individuals) and the offspring (\(\lambda\) individuals). More precisely, the algorithm was implemented with the deap.algorithms eaMuPlusLambda function. We used \(\mu = 100, \lambda = 200\), crossover probability \(p_{\text{cx}} = 0.8\), and mutation probability \(p_{\text{mut}} = 0.2\). Furthermore, tournament selection (deap.tools.selTournament) and blend crossover (deap.tools.cxBlend) were used for selecting and mating individuals, respectively. Mutation was performed with a random Gaussian mutation (deap.tools.mutGaussian). When an individual was selected for mutation with probability \(p_{\text{mut}}\), each of its elements was individually mutated with an individual probability of \(p_{\text{mut}} = 0.3\). Gaussian means for all parameters were set to zero, and standard deviations (SDs) were different depending on the parameter. The parameter values are summarized in Table 2, and we constrained the optimization to solutions within biophysically acceptable boundaries (shown in Table 2).

**Classification.** Besides applying a CNN, the problem of classifying neurons according to their EAP can be done by several other methods (Barthó et al. 2004; Delgado Ruiz and Schultz 2014; Peyrache et al. 2012). It is a well-established observation that pyramidal excitatory cells present a broad spike waveform, while inhibitory cells have a narrow one (Barthó et al. 2004). Therefore, a standard way of classifying between the two classes is to plot spike width \(W\) and full-width half-maximum \(F\) (see Classification and Localization Features for feature extraction details) of the EAP with the maximum amplitude and then cluster the data points in this 2D space (Barthó et al. 2004; Peyrache et al. 2012). Once \(W\) and \(F\) were computed for the electrode with the maximum peak-to-peak amplitude, we applied two different clustering techniques to the point cloud: \(k\)-means and a mixture of Gaussians (MoG) clustering (Pedregosa et al. 2011). While \(k\)-means clustering iteratively assigns points to \(K\) clusters based on their distances to the cluster means and then recomputes the cluster means with new assignments until convergence, the MoG estimates \(K\) Gaussians to fit the data and then labels the data points based on the Gaussian shape. In this case, since the goal is excitatory/inhibitory classification, we set \(K = 2\).

### Statistical Analysis

For localization errors, statistical tests were run on the 11,000 validation observations: since all distributions did not satisfy the normality assumption, nonparametric tests were run (1-sided Mann-Whitney U-test; Mann and Whitney 1947). When sample sizes are large, statistical tests are prone to indicate that there is a significant difference (effect) between distributions, resulting in low \(P\) values. To characterize whether such difference is relevant, a measure of its magnitude, or effect size, should be included (Sullivan and Feinn 2012). To quantify the effect size, we used Cohen’s \(d\) coefficient (Cohen 1992), i.e., the difference between population means normalized by the pooled SD. We considered significant differences (low \(P\) values) to be negligible (effect size < 0.2), small (effect size = 0.2), medium (effect size = 0.5), or large (effect size = 0.8). Test results are shown in Appendix B, divided by group (data set rotation, Table B3; CNN size, Table B4; feature type, Table B5; probe type, Table B6; and localization method, Table B7). Each entry of the tables shows the Cohen’s \(d\) coefficient (rounded to 2 decimals) and the significance of the Mann-Whitney U-test with the alternative hypothesis that column group \(<\) row group.

### RESULTS

In the following sections, we show localization and excitatory/inhibitory classification results only on one type of cells included in the training data set. Therefore, unless otherwise specified, BP and NGC are excluded from the analysis. The performance measures were different for localization and clas-

**Table 2. Model parameter summary**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Range</th>
<th>Gaussian σ</th>
</tr>
</thead>
<tbody>
<tr>
<td>(I_c, I_{\text{dend}})</td>
<td>((-100, 0)) nA</td>
<td>1 nA</td>
</tr>
<tr>
<td>(x, y, z) positions</td>
<td>((10, 80)) μm</td>
<td>10 μm</td>
</tr>
<tr>
<td>(d_x, d_y, d_z)</td>
<td>((-1, 1))</td>
<td>0.1</td>
</tr>
<tr>
<td>(d_{\text{seg}})</td>
<td>((1, 500)) μm</td>
<td>50 μm</td>
</tr>
<tr>
<td>(d_c)</td>
<td>((0.1, 500)) μm</td>
<td>20 μm</td>
</tr>
</tbody>
</table>

Range for initialization and constraint and standard deviation \(σ\) for mutation

Different Gaussian for the different parameters. \(I_c\), current source; \(I_{\text{dend}}\), dendritic current; \(d_{\text{seg}}\), dendritic length; \(d_c\), decay constant.

<table>
<thead>
<tr>
<th>Model</th>
<th>Parameters</th>
<th>No. of Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monopolar</td>
<td>(I_c, r_c)</td>
<td>4</td>
</tr>
<tr>
<td>Bipolar</td>
<td>(I_c, r_{\text{seg}}, r_{\text{pool}}, r_{\text{pool}})</td>
<td>7</td>
</tr>
<tr>
<td>Ball and stick</td>
<td>(I_c, r_c, r_{\text{seg}}, r_{\text{pool}}, d_{\text{seg}}, d_{\text{pool}}, d_c)</td>
<td>10</td>
</tr>
</tbody>
</table>

Summary of parameters for the different inverse models involved in the study. \(I_c\), current source; \(r_c\), predicted soma position; \(r_{\text{seg}}, r_{\text{pool}}\), position of positive \(I_c\); \(r_{\text{neg}}\), position of negative \(I_c\); \(d_{\text{seg}}, d_{\text{pool}}\), dendritic current; \(d_c\), decay constant.

Downloaded from www.physiology.org/journal/jn (005.102.006.141) on July 8, 2019.
sification. In case of localization we used the average total error and for classification the average classification accuracy (ratio between correctly classified observations and total number of observations). Moreover, we analyzed the cell-specific accuracy to get more insight on the classification performance. The average localization error in the x-, y-, and z-directions can be interpreted in the following way. Assuming normally distributed errors, the true soma location is with 95% probability inside a box centered at the predicted soma position with edge lengths of twice the average localization error in the corresponding dimension. The probability rises to 87% with a box edge length of four times the average localization error in the corresponding dimension.

Dependence on Neuron-MEA Alignment

The first question we investigate is how the neuron-MEA alignment affects the localization and classification performance. Three data sets were built, Norot, Physrot, and 3drot. To focus on the effects of alignments, we use fixed feature sets (NaRep for localization, FW for classification; see Classification and Localization Features for definitions), MEA probe (SqMEA-15-10), and CNN size L.

Localization. First, we show the mean and SD of the localization errors along the three axes as well as the total error in Table B2. Each row displays the performance of a rotational data set. Average errors and SDs are 7.3 ± 5.7 μm for Norot, 7.8 ± 6.3 μm for Physrot, and 8.9 ± 8.2 μm for 3drot.

The Norot data set results in significantly lower errors with respect to the 3drot data set (effect size = 0.21; Table B3). Negligible differences are found in the comparison between the Norot errors and the Physrot errors (effect size = 0.09) and between the Physrot and 3drot data sets (effect size = 0.13).

Taking into account the finite size of the soma (~10–15 μm of diameter), we consider the resulting error values to be a sufficient performance for most applications. The errors along the three axes appear to be isotropically distributed, as they show similar values in all directions (but the observations in the x-direction are not uniformly distributed—as shown below in Fig. 9C—since we only considered spikes above 30 μV and spike amplitude decreases with distance).

In Fig. 6, A–C, we show the errors along x-, y-, and z-axes with respect to the x-, y-, and z-coordinates for the three rotational data sets. In these plots, we bin the true x, y, and z neuron positions in seven bins and treat them as categorical data (i.e., the positions can have discrete values depending on the bin they belong to). The data points are the mean of the error grouped by bin and rotation type and the error bar is the SD. Figure 6A shows that errors in the distance estimation (x-direction) tend to increase as the distance of the neuron increases for all three data sets, similarly to Delgado Ruz and Schultz (2014). Regarding the y- and z-dimensions (Fig. 6, B and C, respectively), it is interesting to note how the errors have a convex shape, meaning that errors tend to increase when the neuron is at the border of

![Fig. 6. A: x errors with respect to the x-coordinate (distance). B: y errors with respect to the y-coordinate. C: z errors with respect to the z-coordinate. D: x predictions with respect to the x-coordinate (distance). E: absolute value of the y predictions with respect to the true y-coordinate. F: absolute value of the z predictions with respect to the true z-coordinate. All values are in μm. Orange lines indicate the Norot data set, green lines the Physrot data set, and purple lines the 3drot data set. Gray lines correspond to a perfect prediction. Data are binned in 7 bins along x-, y-, and z-directions: points and error bars display the average errors and their SDs for each bin and each data set.](image-url)
the probe, in which case only partial information about the spike is available.

When looking at the distribution of the predicted \(x\)-, \(y\)-, and \(z\)-coordinates with respect to the true coordinates in Fig. 6, \(D-F\), one can note how the errors observed in Fig. 6, \(A-C\), are caused by an underestimation of the soma distance in all dimensions: at large distances (\(x\)-direction) from the probe, neurons are predicted to be closer to the MEA; when they are close to the \(y\) and \(z\) borders of the probe, the predicted position is closer to the center of the MEA.

Next, we consider the variability of localization performance depending on cell types and alignment. In Fig. 7A the bar plots show the average total errors and their SDs grouped by neuron morphology type (11 training morphologies + BP and NGC; see Fig. 2 for representative cells) for the Norot, Physrot, and 3drot data sets. The range and distribution of distances taken into consideration are the same as in Fig. 6. Focusing on the Physrot data set, the minimum error of \(4.7 \pm 3.8 \mu m\) is obtained for the SBC morphology, while the worst performance is \(15.9 \pm 9.1 \mu m\) for slender-tufted PC (STPC) morphology. The difference in prediction performance with respect to cell type does not seem to be depending on excitatory/inhibitory morphologies (i.e., pyramidal and nonpyramidal cells), nor do they look to be clustered depending on morphological subclasses; for instance, among the different basket cells (names ending with BC) there is some variability among large basket cells (LBC), NBC, and SBC, and the same holds for pyramidal cells [STPC, TTPC1, thick-tufted PC with early bifurcating apical tuft (TTPC2), and untufted PC (UTPC)]. The performance of BP and NGC (Fig. 7A), which were not used for training, is in line with other cell types, with errors of \(8.8 \pm 2.2 \mu m\) and \(9.4 \pm 5.1 \mu m\), respectively. This result confirms that the method is capable of dealing with diverse morphologies, as long as the training set contains a large representation of cell types.

**Classification.** The accuracy analysis of excitatory/inhibitory classification is based on the \(FW\) feature set. Table B8 in \textit{APPENDIX B} shows the classification accuracies for each cell morphology plus the average accuracy and the SD for the other cell types. For explicit values see Table B8. C: spike shapes for maximum peak electrode sites are plotted in the \(FW\)-plane. Red dots are inhibitory neurons, and they lie in the lower left part of the plot. Blue dots show excitatory cells, in the upper right part of the plane. The magenta dots are chandelier cells (ChC), and they lie at the intersection between excitatory and inhibitory cells. The opacity represents the number of occurrences; more opaque dots correspond to more frequent observations. The ellipses are built by following the principal axis of each distribution, and the lengths of their major and minor axes are proportional to the eigenvalues of the covariance matrix. FWHM, full-width half-maximum.

The average classification accuracy is equally high for the Norot and Physrot data sets (98.1 \(\pm\) 2.4\% and 98.0 \(\pm\) 3.9\%, respectively) and lower for the 3drot data set (97.6 \(\pm\) 3.9\%). This is because neurons are rotated with more degrees of freedom; nevertheless, on average the accuracy remains very high in all cases. A closer examination of this result reveals that the main reason for the drop in classification accuracy was misclassification of the chandelier cells (ChCs). The lowest value is the ChC accuracy in the 3drot data set (84.1\%). In Fig. 7C, we show the spike shapes in the \(FW\)-plane of the electrode site with largest amplitude. Inhibitory neurons mainly lie in the lower left part (narrow spikes). Excitatory neurons are almost perfectly classified as excitatory cells, as shown in Table B8 and Fig. 7B. The spike shapes of ChC in the \(FW\)-plane mainly lie at the interface with the excitatory neurons. This might explain why they are harder to classify correctly with respect to the other cell types.

**Effect of CNN Size**

We investigated how localization and classification performances vary with network size. The results shown in this section were obtained with the SqMEA-10-15 probe, NaRep features for localization, and \(FW\) for classifications. For the remaining analyses, boxplots and cumulative distribution functions (cdfs) are used to represent the performance of the localization models. In all boxplots, the box is the interquartile range (IQR), i.e., the 25th and 75th percentiles, the horizontal lines inside the box show the medians, and the red diamonds display the means. The whiskers (horizontal black lines) represent the highest and lowest data values within 1.5 times the IQR. Data points outside the whiskers are plotted as black dots and are regarded as outliers. We obtained the cdfs by sorting the sample and pairing each data point with its normalized rank (percentile). Hence, the point where the cdf crosses 0.5 represents the median of the localization error.
**Localization.** Figure 8A shows the localization errors grouped by CNN size (XS, S, M, L, and XL). Increasing the size of the network improves the performance significantly (Table B4), but for sizes L and XL the average localization error is almost the same [7.8 ± 6.3 μm for L and 7.3 ± 5.8 μm for XL (effect size = 0.09)]. If not stated otherwise, networks of size L have been chosen, as they provide a good compromise between performance and time required for training.

**Classification.** Table B9 in **APPENDIX B** shows the performance in classification into excitatory and inhibitory neuron types. The highest accuracy (98.6 ± 1.1%) is reached with a network of size M, while all others show a slightly lower performance. A possible explanation for the lower score of the XL network is overfitting to the training set because of the large number of parameters.

**Feature Selection.**

In the previous sections, we have presented results with fixed feature sets (NaRep for localization and Waveform feature). eliminating the effects caused by other factors, such as alignment, cell type and CNN size. The following results were obtained on SqMEA-10-15 probes using CNNs of size M (because of the long training time required by 3D CNNs for Waveform feature).

Localization. In Fig. 8, C and D, we display the boxplots and cdf of the errors with varying feature sets for localization. In other studies, either the sodium peak is the only feature used (Blanche et al. 2005; Delgado Ruz and Schultz 2014; Mechler et al. 2011; Mechler and Victor 2012; Somogyvári et al. 2005) or the entire spike time course is modeled (Somogyvári et al. 2012). Here we show that all CNNs relying on peak input show roughly the same performance: average errors ± SDs are 8.8 ± 7.1 μm for Na, 8.8 ± 7.9 μm for Rep, and 8.8 ± 7.7 μm for NaRep. Negligible differences are found when comparing Na, Rep, and NaRep, with effect sizes close to zero (Table B5).

The Waveform CNN results in a lower average prediction error of 6.9 ± 6.5 μm, which is significantly better in comparison with Na (effect size = 0.28), Rep (effect size = 0.26), and NaRep (effect size = 0.27; Table B5). We speculate that the performance of the Waveform approach is only slightly increased (by ~2 μm) for the following reason: when considering the peaks only, transmembrane currents are mainly concentrated around the soma (Delgado Ruz and Schultz 2014; Gold et al. 2007; Somogyvári et al. 2005, 2012); therefore, the peak features contain almost all the information the CNN needs for soma location.

Classification. Classification performances are listed in Table B10 in **APPENDIX B**. The AFW feature set, with an accuracy of 98.6%, performs better than AW and FW, with accuracies of 98.1% and 97.0%, respectively. The Waveform feature set, which uses a downsampled version of the entire spike, performs almost perfectly on the classification task (accuracy 99.7%). Given these results, the Waveform feature set is better than the other approaches, at the expense of more computationally demanding training procedures.

**Performance with Different MEA Probes.**

We built simulated spikes using eight different MEA models: five of them are square arrays with varying pitch, and the
other three are the NeuroSeeker (Neto et al. 2016), NeuroPixels (Jun et al. 2017) (trimmed to 128 channels), and Neuronexus-32 (clipped to 30 electrodes to make it rectangular) probes. In the following paragraphs, we present the capabilities in terms of neuron localization and classification for the different probes. All simulations shown in this section make use of CNns of size L, NaRep features for localization, and FW for classification.

Localization. Figure 8, E and F, show localization errors for the eight different probes (boxplots and cdf). Although an error reduction can be observed from square MEA with 30-μm pitch to 10-μm pitch, as expected, even with a relatively low density (30-μm pitch) a CNN can learn localization models with an average error as low as 8.4 ± 6.4 μm for the SqMEA-5-30. As a comparison, the average error for the probe with 10-μm pitch (SqMEA-15-10) is 7.6 ± 6.4 μm. The errors are in the same order also for the Neuronexus probe (mean of 8.5 ± 7.2 μm) and for the NeuroSeeker probe (mean of 9.3 ± 7.7 μm). When evaluating the performance on 128 sites with the arrangement of the NeuroPixels probe, the average error is 10.8 ± 8.5 μm. One may note that even though the Neuroseeker and Neuronexus probes have lower pitch (NeuroSeeker: 22.5 μm in y- and z-axes; Neuronexus: 18 μm in y-axis and 25 μm in z-axis) compared with SqMEA-5-30, their localization error is higher. The reason for this discrepancy might be in the arrangement of the electrodes: while the SqMEA-5-30 has an effective width (considering point electrode contacts) of 120 μm, for the NeuroSeeker the effective width (considering point electrode contacts) is 67.5 μm and for the Neuronexus it is 36 μm. Hence on the NeuroSeeker and Neuronexus probes there is less spatial information in the y-direction, which may explain the reduced localization accuracy.

In general, most comparisons show negligible differences (effect size < 0.2), except for the NeuroSeeker and NeuroPixels probes. The NeuroSeeker probe performs worse than the high-density square MEAs (SqMEA-15-10: effect size = 0.25, SqMEA-10-15: effect size = 0.22), while the NeuroPixels errors show effect sizes above 0.2 in all comparisons (ranging from 0.43 compared with SqMEA-15-10 to 0.3 compared with Neuronexus) except for the comparison with the NeuroSeeker probe (effect size = 0.19). In case of the NeuroPixels probe the checkerboard arrangement might pose additional difficulties, resulting in even lower performance.

Classification. Table B11 in appendix B shows accuracies for classification with different probes. The average accuracies are very high and almost the same for all probes, from a minimum of 96.6% (SqMEA-6-25, SqMEA-15-10) to a maximum of 98.6% (NeuroPixels-128).

Comparison with Other Approaches

In this section, we compare the CNN approach to other state-of-the-art methods. For localization, we used the monopolar, bipolar, and ball-and-stick models described in Comparison with Other Models to solve the inverse problem on our simulated data sets. Hence the results obtained for other methods might be different with respect to ones in the literature because the number of cell models, the utilized probes, and the neuron-MEA alignments vary. For characterization of excitatory and inhibitory neurons, we compared with commonly used clustering techniques.

Localization. For localization, we use the validation data set on theSqMEA-10-15 probe. The CNN errors displayed in the plots are obtained with the NaRep feature set and a network of size L. In Fig. 9, we show the errors of the simplified models described in Comparison with Other Models and for the CNN method.

We found that the CNN performs significantly better than the inverse approach in all cases, with an average error and SD of 7.8 ± 6.3 μm. The large differences between the CNN and the other methods’ error distributions are confirmed by the effect sizes: 0.9 for the monopolar approach, 0.68 for the bipolar approach, and 0.87 compared with the ball-and-stick approach (Table B7). Among the models used to solve the inverse problem, the monopolar has a mean error and SD of 21.7 ± 20.9 μm, the bipolar model of 15.6 ± 15.2 μm, and the ball-and-stick model of 22.6 ± 23.2 μm. The better performance of the bipolar model with respect to the monopolar model (and ball-and-stick model) can be due to the fact that it is the only model capable of predicting negative and positive potential values on the MEA. Dendritic branches act as current sources when the soma is depolarized, causing positive deflections in the extracellular potential (Pettersen and Einevoll 2008).

Studying the probability density function (pdf) of the predicted coordinates by different models in Fig. 9, C–E, the monopolar model tends to underestimate the distance (x-coordinate) from the MEA (note sharp peak in the distribution in Fig. 9C). In the y- and z-axes, instead the predictions are closer to the center of the MEA when observations lie outside the boundary of the probe (note the different steepness and shape of the monopolar pdf with respect to the true pdf in Fig. 9, D and E, close to −100 μm and 100 μm). Similarly, the bipolar model also underestimates distances in the x-direction, but the underestimation is less severe. In the y- and z-directions it nicely follows the true distribution. The ball-and-stick model has distributions very similar to the monopolar model in all three directions. The CNN approach, on the other hand, is the closest match to the true distribution in all three dimensions. Note that the distribution in the x-direction is not as uniform as in the y- and z-directions (the density decreases with increased distances) because we discarded spikes with a peak-to-peak amplitude below 30 μV here.

Classification. For excitatory/inhibitory classification we compared the performance of our CNN approach to standard clustering techniques in the FW space (F: full-width half-maximum, W: width). In Fig. 10A, we show the validation data with the SqMEA-10-15 probe and the excitatory/inhibitory balanced data sets (8,000 observations). Each point is computed from the recording site with largest amplitude. Although it is true that inhibitory cells cover the bottom left part of the cloud (narrower width and full-width half-maximum) and excitatory cells the top right (wider spike shape), we can observe that there is some overlap between the two groups. When we apply k-means clustering (Fig. 10B), the algorithm correctly assigns the bottom left part to inhibitory neurons and the top right part to excitatory neurons, but the overlap is mainly assigned to the excitatory class. This yields an accuracy of 99.9% for the excitatory class but only 60.7% for the inhibitory one, with an average of 80.3%. When it comes to the MoG, the data are fit to two multivariate Gaussians and labels are assigned based on the probability of an observation to belong.
to the two distributions. Figure 10C shows the estimated Gaussians (ellipses) and the labeling of the points. Although the MoG is capable of describing the diagonal shape of the excitatory cloud, the overlap between the observations cannot be untangled, resulting in an accuracy of 98.8% for the excitatory class and 54.2% for the inhibitory one, with an average of 76.5%. The CNN method, instead, is able to discern the overlap in the FW space. This is certainly due to its higher complexity, due both to the method itself and to the use of all electrodes’ information, not only those with highest amplitude. The CNN result shown here (FW feature set, size L) allows us to correctly predict excitatory cells in 98.9% of the cases and inhibitory cells with an accuracy of 97.1%. This makes it the best-performing method among those compared here, with an overall accuracy of 98.0%. It could be argued that the comparison was somewhat unfair, as our CNN approach considers F and W images (computed on all recording sites), while the clustering is performed with values computed from the electrode with highest amplitude only. Nevertheless, it is not common practice to consider waveforms on all electrodes but only on the one with highest amplitude (Barthó et al. 2004; Peyrache et al. 2012).

**m-Type Classification**

In addition to separating excitatory cells from inhibitory ones by trained CNNs, we tried to make a finer subdivision and classify cells into morphology classes (m-type) based on the EAP. The approach is similar to that for excitatory/inhibitory classification, but instead of only 2 output classes we take the 11 m-type classes (cells of m-type BP and NGC are excluded, since only 1 morphology is available in the data set). We use a CNN of size L and consider Waveform features (in this case with a downsampling factor of 8, i.e., a sampling frequency of 4 kHz) on the SqMEA-10-15 with the Physrot data set. The resulting confusion matrix E, in which each entry $E_{ij}$ represents the amount of observations of the true cell type $i$ predicted as cell type $j$, is depicted in Fig. 11. We do not observe a striking diagonal, indicating that full identification of all cell types is not feasible from EAPs. But it is noteworthy that there is some block structure dividing excitatory neurons from inhibitory neurons. This division is learned intrinsically by the network, and inhibitory cells are classified within the inhibitory block in 100.0% of the cases and excitatory cells within the excitatory block in 95.7%. Concerning the mixing of TTPC1 and TTPC2, we do not expect to be able to differentiate between these two types because their only difference is the distance of the bifurcation point of the apical dendrite to the soma. Since the MEA is located close to the somatic region, recordings might not be sensitive to this delicate difference. Disregarding this mixing, the m-type classification performs well (chance would be 9.1%) on excitatory cells and inhibitory Martinotti cells (MC) (80.5%). Note that these well-classified cells make up a large proportion of cortical cells. The overall accuracy of 34.0% illustrates that the morphological details are partially resolved by the CNN. In cases in which the CNN is not able to extract the information about the morphological details, it is unclear whether the information is present at all in the EAP or an increased number of cell models could solve the problem. In
conclusion, the results show promise for a more refined classification than only distinguishing excitatory cells from inhibitory cells.

Validation on Different Models

To investigate how general the trained CNN models are, we tested the performance of localization and classification on simulated EAPs from other neuronal models, namely, the cell model from Hay et al. (2011) and the models from the Allen Brain Institute (ABI) cell type database (Gouwens et al. 2018; http://celltypes.brain-map.org). For the following results, we used the SqMEA-10-15 probe, CNNs of size L, and NaRep and FW feature sets for localization and classification, respectively.

Hay model. The Hay cell models a neocortical pyramidal cell from L5b, and the techniques used to build the models were similar to the models from the NMC Portal. Therefore, we expect a relatively good performance in localization and classification with the CNNs trained on our standard NMC data sets. We built a Physrot data set of Hay cells consisting of 1,000 observations at random locations around the probe as described in Simulated Recordings, and we then evaluated the performance of the CNNs in localization and classification.

For localization, the average error on the Hay data set is $8.7 \pm 6.6 \mu m$, perfectly in line with the average errors of TTPC models in the NMC validation data set (Fig. 7A). The average error over all cell types in the NMC validation data set is $7.8 \pm 6.3 \mu m$. For classification, we obtain an average accuracy of 76.4%, while the accuracy on the NMC validation set is 98.0%. The lower accuracy could be due to the fact that the Hay model includes other types of mechanisms, such as active calcium channels in the apical dendrites, that are not modeled in the NMC cell models.

Allen Brain Institute models. The cell models from the ABI that we selected are quite different from the NMC cell models at least for two reasons. First, the ABI neurons are from mice, whereas the NMC cells are from juvenile rats. Second, they are from visual cortex (19 cells) and postrhinal area (1 cell), whereas the NMC models are from somatosensory cortex. With CNNs trained on NMC data are expected to have lower accuracy when applied to the ABI data. We generated 1,000 EAPs for each of the 20 ABI cell models, according to the description in Simulated Recordings.
5.9 NaRep features, obtaining a localization average error of 19.3 ± 11.5 μm, larger than the 8.9 ± 8.2 μm obtained on the NMC validation set as expected. For classification, we distinguished excitatory and inhibitory cells in the ABI data set based on mouse transgenic lines (details in Appendix A). With the CNN for classification trained on NMC models, the average accuracy is 76.9%, while it is 97.6% on the NMC validation data set.

Since the cell models of the ABI come from a different species and are from a different cortical region, we trained a CNN on this data set only—16 models are used for training and 4 for validation (Appendix A). We used a CNN of size L and NaRep features, obtaining a localization average error of 5.9 ± 4.5 μm, which is in line with the performance we obtained on NMC models only. We did not run classification with so few models (only 20 cell models in total), because the CNNs need a larger diversity to find general features related to excitatory-inhibitory types (using the NMC data we trained on 192 cell models).

Test on Experimental Data

Although the method proposed in this report is at a proof-of-concept stage, we tested some CNNs trained with simulated data on experiments at least for plausibility. We decided to use data (publicly available at http://www.kampff-lab.org/validating-electrodes) from paired juxtacellular and extracellular recordings (Neto et al. 2016) where, to a certain extent, the ground-truth location is known. The extracellular signals are measured with either the Neuronexus or the NeuroSeeker probe. Taking the amplitude threshold of our CNN training simulation (peak-to-peak amplitude of 30 μV on at least 1 electrode) into account and considering only cells in front of the MEA, we were left with 10 data sets (see Appendix A for further details). After performing juxtacellular-triggered averaging, we fed the average EAP waveform into the CNN and predicted the soma position. The CNNs were trained with simulated data having the appropriate geometric alignment (MEA probes are rotated by –48.2° along the y-axis). On average the prediction error is 42.2 ± 16.8 μm, assuming the true soma position is the tip of the juxtacellular probe. The experimentally determined positions for the x-, y-, and z-coordinates range from 27 μm to 129 μm, –48 μm to 6 μm, and –121 μm to 21 μm, respectively. Neto et al. (2016) report a distance uncertainty of 10.5 ± 5.2 μm. This uncertainty only applies to the tip position of the juxtacellular probe, but considering the correct soma position would be the center of the soma (which are not quantified) are not considered.

DISCUSSION

This work provides a deep learning approach for neuron localization and classification based on MEA recordings. We simulated in vivo-equivalent EAPs and built data sets for various probe designs, using a multitude of cell models from the NMC Portal (205 cell models from Ramaswamy et al. 2015). CNN models trained on these simulated spikes predict the soma position of the neuron and characterize whether it is excitatory or inhibitory. The accuracy depends on the neuron-MEA alignment, the specific cell types, the CNN size itself, and the input feature sets. For completeness, we compared the proposed method with existing strategies regarding both localization and classification of recorded spikes, we validated on cell models from other databases, and we tested the models on publicly available experimental data.

Localization

We showed that the CNN method is robust and accurate in predicting the 3D soma location from spikes generated by neurons with a physiological neuron-MEA alignment (Physrot, defined in Neuron-MEA alignment). The average errors are on the order of 7.6–11.7 μm for all probes involved in the study (Fig. 8E). We demonstrated the CNN approach to be robust with different cell models and to be able to generalize among cell types not used for training (BP and NGC). Finally, local-
Fig. 12. Soma position predictions of a convolutional neural network (CNN) based on experimental extracellular action potential (EAP) recordings. Experimental data are from paired juxtacellular-extracellular recordings (Neto et al. 2016) where the position of the soma is associated with the tip position of the juxtacellular probe. The CNN (size L, NaRep feature) is trained on simulated (3drot) EAP signals. Error bars are CNN prediction uncertainties for the predicted coordinates and 4.2 μm, 2.8 μm, and 8.5 μm (misalignment uncertainties reported by the experimenters of Neto et al. 2016) for true x-, y-, and z-coordinates, respectively.

Generalization performances achieved with our approach are significantly better than solving the inverse problem with various generative models. With a SqMEA-10-15 data set the total error in three dimensions was 21.7 ± 20.9 μm for the monopolar current source, 15.6 ± 15.2 μm for the bipolar current source, and 22.6 ± 23.2 μm for the ball-and-stick model, whereas with our CNN approach we obtained an error of 7.8 ± 6.3 μm (as shown in Fig. 9).

In a recent study (Delgado Ruz and Schultz 2014), a Neuronexus-32 probe was used (shown in Fig. 3) with a modified ball-and-stick model to solve the inverse problem. For the five cell types considered in Delgado Ruz and Schultz (2014), they reached average errors of 6.26 ± 6.10 μm, 6.03 ± 7.68 μm, and 2.58 ± 4.75 μm along the x-, y-, and z-axes, respectively. Using the same probe on our Physrot data set, we obtained with CNNs average errors of 4.1 ± 4.5 μm, 4.3 ± 4.7 μm, and 4.3 ± 5.3 μm for the x-, y-, and z-axes, respectively (with NaRep feature set and size L).

**Classification**

The deep learning method was applied to excitatory/inhibitory classification with accuracies above 96.6% for all employed MEA models using the FW feature set and a CNN size L. An almost perfect outcome of 99.7% was obtained with the Waveform features on the SqMEA-10-15 probe. Compared with standard strategies using spike widths extracted from the spike shape, it showed a significant improvement (k-means clustering: 80.3%, MoG: 76.5%; Comparison with Other Approaches). We also attempted to distinguish among 11 cell morphologies (m-type classification). The overall accuracy of 34.0% is substantially better than the chance level of 9.1%. It is interesting to see that m-type classification performs a sort of unsupervised learning, as inhibitory cells were classified as inhibitory in 100.0% of the cases and excitatory cells as excitatory 95.7% of the time.

**Overfitting and Stability**

When evaluating the predictions of our CNNs on the validation data set, we observed a drop in accuracy compared with training accuracy. The drop is in an acceptable range for excitatory/inhibitory classification (0–3% with respect to the training accuracy) and localization (up to 3.7 μm prediction error increase). In case of m-type classification, the validation accuracy drops ~65% compared with training accuracy, clearly indicating overfitting. Since we do not have enough diversity in cell model data to build a third data set for implementing early-stopping regularization (i.e., stop training as soon as the generalization error increases), we tracked the evaluation accuracies depending on the number of training epochs. In most cases, they reached a plateau after roughly 2,000 training epochs and did not decrease significantly afterwards, while training accuracies still increased. Therefore, we decided to stop training after 2,000 training epochs, assuming that the CNN has extracted most of the generalizable information provided by the EAPs at that point. Moreover, we tried to quantify the stability of the performance depending on different initial weights before the optimization process. To do so we ran the CNN training for localization and classification (on the SqMEA-10-15 probe, CNN size L, and with NaRep and FW features, respectively) six times with different random seeds. We obtained an average mean error of 7.6 ± 0.1 μm with an average SD of 6.3 ± 0.2 μm for localization (including the BP and NGC models) and an average mean accuracy of 97.9% with a SD of 0.2% for classification, indicating that performance is not dependent on the initial conditions of network weights and the convergence is robust.

**Model-Based Approach**

The findings presented in this study are based on simulations. Although this might be regarded as a limitation, we want to stress that the proposed method makes use of highly detailed cell models (Markram et al. 2015) and the complexity of such models is maintained and learned by CNNs. Previous approaches to localization and/or classification relied on simple forward models to solve the inverse models—monopolar, bipolar, ball-and-stick models, etc. (Blanche et al. 2005; Delgado Ruz and Schultz 2014; Somogyvári et al. 2012). We showed that CNNs outperform these models in estimating the soma positions. Another point that plays in favor of the use of neural simulations is the difficulty in gathering ground-truth data experimentally. Localizing and classifying neurons in real recordings requires advanced and highly accurate equipment, and the recorded labeled data would most likely still not be sufficient to train data-hungry machine learning algorithms such as CNNs. Nevertheless, validation on experimental data is definitely a required step and will be based on combined approaches with paired electrophysiological recordings and standard microscopy (Neto et al. 2016), or even involving more sophisticated and precise imaging techniques, such as two-photon imaging (Göbel and Helmchen 2007), which was paired with electrophysiological recordings in vivo in Shew et al. (2010). Paired electrophysiology and two-photon microscopy data, possibly in combination with intracellular voltage monitoring through patch clamping or voltage-sensitive dyes, could also represent a valuable tool to further validate and
improve the forward modeling schemes, providing morphological, intracellular, and extracellular recordings simultaneously.

Another advantage of using forward modeling is that the performance of the machine learning algorithm could be improved by building case-specific data sets that better match the real experimental scenarios. In this work, we assumed that the simulated probe was inserted in L5 of somatosensory cortex of a juvenile rat with a vertical insertion angle. However, somatosensory cortex can present large differences with respect to other brain regions (e.g., hippocampus, cerebellum, or other cortical regions) but also among animal species. Therefore, we do not envision a single universal model to localize and classify neurons but species- and brain area-specific CNNs to accurately deal with variability in neuronal types and functions. For example, when we fed mouse data from the ABI database, the localization CNN trained on rat data performed relatively poorly (19.3 ± 11.5 μm; Allen Brain Institute models), but trained on mouse models the performance is in line with what we obtained on rat data (5.9 ± 4.5 μm).

**Effect of Probe Design**

Regarding neural probes, a forward modeling-based approach can give important insights for the design and manufacturing of next-generation probes. For example, our results showed that even relatively low-density probes, such as the SqMEA-5-30, despite performing slightly worse than higher-density probes, still yield high accuracy in localizing and classifying neurons. Potentially, the pursuit of extremely high-density probes, which makes the design complicated and the data throughput very high, is not required for classification and localization tasks [although it might still be important for spike sorting (Franke et al. 2012; Rossant et al. 2016)]. However, for such simulation-driven MEA design, the simulations lack a more accurate electrode model considering finite size recording sites (in this work we used an ideal point electrode), electrode impedances, and transfer functions.

**Future Extensions**

The generative model for spike simulations could be improved in various ways. A straightforward improvement to obtain more accurate simulations could be including the MEA scar in the data generation, by clipping or bending neuronal branches in the proximity of the probe before simulating the recordings. Another refinement might be to take into account the finite size effects of the electrode contacts by means of the disk-electrode approximation (Lindén et al. 2014), which was shown to be appropriate for current sources positioned at distances larger than the contact radius (Ness et al. 2015). Moreover, here we assumed a tissue with homogeneous and isotropic electrical properties, but experimental findings suggest that in the cortex the conductive properties of the extracellular space are anisotropic (Goto et al. 2010). Anisotropy could be easily taken into account for the simulation of spikes (Ness et al. 2015; Pettersen et al. 2012). As the proposed approach strongly relies on high-fidelity simulations that reliably describe the neuron dynamics and volume conduction, another strategy could be using finite element method-based models, as in Agudelo-Toro and Neef (2013), Pods et al. (2013), and Tveito et al. (2017), which would result in more detailed simulations at the cost of a much higher computational cost for data generation. Another layer of modeling is the electrode-tissue interface. The generated data should include electrical properties of the electrodes, such as the impedance, and account for their variability in experimental scenarios. In this work, we used polytrodes with a relevant size with respect to the neuron: although we assumed a homogeneous medium, the presence of the probe itself represents an obstacle for electrical signal propagation and can be modeled with either finite element method or analytical simplifications, such as the method of images (Ness et al. 2015).

In this work, we did not include any noise in the simulated recordings. The rationale behind this choice is that sorted spikes can be cleaned by applying spike-triggered averaging. With spike-triggered averaging, additive random noise is reduced by a factor of \(\sqrt{N} \), where \(N\) is the number of occurrences of the sorted unit. Moreover, a common problem in spike sorting is electrode drift, in which the relative position between a neuron and the recording electrodes changes during the experiment. If drifting is detected from the spike sorting algorithm, one could feed different averaged EAPs computed in separate time windows and evaluate the drift over time, similarly to Delgado Ruz and Schulze (2014), in which windows of 5 min were used to compute the mean EAP.

Furthermore, the recording site area affects the amount of noise in the recordings, as the recording area is related to the impedance of the electrode. Here we assumed perfectly sorted spikes, from which a clean EAP can be computed. Clearly, with experimental data errors in spike sorting would affect the performance of localization and classification due to distorted waveforms from wrong assignments.

**Outlook**

Precise neural localization and classification from in vivo extracellular recordings has the potential of making electrophysiology an even more powerful technique to interact with neural tissue. Rather than only extracting spike trains, we could build a 3D representation of the recorded units and perform functional electrical imaging to study the spatial interactions among different cell types in neural microcircuits. On top of this, a precise localization of neuronal somata might enable the use of highly selective electrical stimulation patterns (Buccino et al. 2016) and represent an advancement in single-neuron stimulation from extracellular probes.

We strongly believe that computational approaches must go hand in hand with experimental ones, and an extension of this work might include the simulations of the entire pipeline from simulated MEA recordings, for example, with VISAPy (Hagen et al. 2015), to electrical stimulation including spike sorting, localization, classification, electrical stimulation, and evaluation of its effect on detailed neural morphologies.

**APPENDIX A: DATA SELECTION**

**Neocortical Microcircuit Collaboration Portal Data Set**

In this appendix we discuss the data set and the modifications that we applied to make sure that that training and validation set are completely disjointed.

In the original data set (https://bbp.epfl.ch/ncmc-portal/welcome; L5 cells) there are nine inhibitory neuron types: BP, bitufted cells (BTC), ChC, double bouquet cells (DBC), LBC, MC, NBC, NGC, and SBC.
The four excitatory types, i.e., the PCs, are grouped into STPC, TTPC1, TTPC2, and UTPC. While belonging to the same m-type, neurons can have different electrophysiology properties (e-type) based on their firing patterns (Markram et al. 2015). In L5 the e-types are categorized into continuous accommodating (cAC), continuous stuttering (cSTUT), burst accommodating (bAC), burst stuttering (bSTUT), continuous nonaccommodating (cNAC), delayed stuttering (dSTUT), burst nonaccommodating (bNAC), continuous irregular (cIR), delayed nonaccommodating (dNAC), burst irregular (bIR), and continuous adapting (cAD). Since not all m-types express all e-types, the combination of morphological and electrical type gives rise to 52 morpho-electrical types (me-types) in L5. For each me-type, the NMC database contains five cell models; therefore, there are a total of 260 cell models in the data set.

In Markram et al. (2015), to extend the number of reconstructed models, an algorithm is used to clone morphologies: neural compartments are randomly scaled and rotated with respect to each other. Moreover, morphologies are also stretched and shrunk to make up new morphologies. We identified 54 different morphologies in the data set, listed in the Supplemental Material for this article. Although the cloned and/or scaled morphologies are indeed different than the original ones, their shape is quite similar.

The use of CNNs, which are among the most powerful machine learning algorithms, pushed us to pay particular attention in the training-validation splitting so that no information of the validation set is present in the training set (leakage). Hence, the presence of a cloned/scaled version of the same morphology in both training and validation has been avoided. We selected training and validation cell models so that all morphologies in the validation set are unique. In doing so, we had to remove all instances of BP and NGC from the training set, as all the models are derived from the same reconstructed morphology. For localisation and excitatory/inhibitory classification, we kept a BP and an NGC model in the additional validation set.

After the manipulation, the training set consists of 192 cell models, while the validation set only contains 11 cell models, one for each m-type. Moreover, we use one BP and one NGC model, not used for training, as further validation. In total, we included 205 neuronal models out of the available 260. The cell models are listed in the Supplemental Material.

Allen Brain Institute Data Set

From the Allen Brain Institute cell type portal (http://celltypes.brain-map.org/data), we selected cell models according to three criteria: 1) cells were from mice, 2) cells were from L5 (to maintain consistency with the data from the NMC Portal), and 3) cells had an all-active model. This search reduced the number of cell models available to 42. During the simulation process, we further discarded 22 models based on two extra rules: 1) if adjusting the current-clamp amplitude to the soma could not induce a number of spikes between 10 and 30 in 10 iterations (in which the weight was multiplied by 0.75 if the number of intracellular spikes was >30 and by 1.25 when <10 spikes were detected) and 2) if <5 EAP peaks had a peak-to-peak threshold of 30 μV in 500 random positioning of the neuron around the probe (meaning that the EAPs were mainly below the defined detection threshold). After this pruning, 20 cell models are left. To distinguish between excitatory and inhibitory cells, we used the transgenic line information: Pvalb, Sst, Htr3, and Gad2 lines were considered inhibitory; Rbp4, Scnn, and Rorb were considered excitatory (Gouwen et al. 2018). After this division there were 11 inhibitory and 9 excitatory cell types.

To avoid overfitting, we randomly selected 4 models, 2 excitatory and 2 inhibitory, and we set them aside for validation, while we used the remaining 16 neuronal models for training.

The cell models are listed in the Supplemental Material.

Kampff Laboratory Data Set

Accompanying their article on paired juxtacellular-extracellular recordings (Neto et al. 2016), the laboratory of Adam Kampff publicly offers the data on http://www.kampff-lab.org/validating-electrodes. To extract an averaged extracellular waveform for each cell that can be fed into a trained CNN, some data processing was necessary. First, we detected spikes in the juxtacellular probe by thresholding the signal to get the cell’s spike times. Second, we high-pass filtered the extracellular MEA recording with a third-order Butterworth filter in forward-backward mode with a band pass of 100–14,250 Hz. Afterwards, we averaged the EAP in windows of 7 ms around the spike times (2 ms preceding and 5 ms after the peak). This average waveform was then referred to as the juxtacellular-triggered average and was used as input for CNN predictions. After this preprocessing, 10 of the 29 available data sets fulfilled the criteria of having a peak-to-peak amplitude of 30 μV on at least one electrode and being in front of the extracellular probe (2014.03.26: Pair 2.0, 2014.03.20: Pair 3.0, 2014.03.26: Pair 2.1, 2014.10.17: Pair 1.1, 2014.10.17: Pair 1.0, 2014.11.25: Pair 3.0, 2014.11.25: Pair 1.0, 2014.11.25: Pair 2.0, 2015.09.04: Pair 5.0, 2015.09.03: Pair 6.0). These 10 data sets were used in Test on Experimental Data to test our deep learning approach.

APPENDIX B: ADDITIONAL INFORMATION

This appendix contains additional information on parameters and results. Table B1 shows the specific CNN parameters for different network sizes. The average localization errors and SDs for different rotational data sets are contained in Table B2, and the corresponding statistical analysis is depicted in Table B3.

Further results on significant differences and effect sizes of localization performances for different CNN sizes, features, MEA probes, and localization methods are listed in Tables B4, B5, B6, and B7, respectively.

The excitatory/inhibitory classification accuracies grouped by cell type for different rotational data sets, CNN sizes, feature sets, and MEA probes are shown in Tables B8, B9, B10, and B11, respectively.

ACKNOWLEDGMENTS

We thank Joana Neto and colleagues from the Kampff laboratory for helping with analysis of the experimental data.

GRANTS

The authors acknowledge support by the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through Grant INST 39/963-1 FUGG. Furthermore, the research leading to these results has received funding from the European Union’s Seventh Framework Programme (FP7/2007-2013) under Grant Agreement 600925 (NeuroSeeker) and the European Union Horizon 2020 Research and Innovation Programme under European Union's Seventh Framework Programme (FP7/2007-2013) under Grant Agreement 600925 (NeuroSeeker).

Table B1. CNN sizes

<table>
<thead>
<tr>
<th>Size</th>
<th>k1</th>
<th>d1</th>
<th>k2</th>
<th>d2</th>
<th>nEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>XS</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>8</td>
<td>128</td>
</tr>
<tr>
<td>S</td>
<td>2</td>
<td>8</td>
<td>2</td>
<td>16</td>
<td>256</td>
</tr>
<tr>
<td>M</td>
<td>3</td>
<td>16</td>
<td>3</td>
<td>32</td>
<td>512</td>
</tr>
<tr>
<td>L</td>
<td>3</td>
<td>32</td>
<td>3</td>
<td>64</td>
<td>1,024</td>
</tr>
<tr>
<td>XL</td>
<td>3</td>
<td>64</td>
<td>3</td>
<td>128</td>
<td>2,048</td>
</tr>
</tbody>
</table>

Convolutional neural network (CNN) parameters of the different network sizes: layer 1 convolutional kernel size k1, layer 1 convolutional kernel depth d1, layer 2 convolutional kernel size k2, layer 2 convolutional kernel depth d2, and nodes in fully connected layer nEC.
Table B2. Localization errors grouped by rotational data set

<table>
<thead>
<tr>
<th>Data Set</th>
<th>x Error</th>
<th>y Error</th>
<th>z Error</th>
<th>Total Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Norot</td>
<td>3.9 ± 3.9</td>
<td>3.0 ± 3.2</td>
<td>3.9 ± 4.6</td>
<td>7.3 ± 5.7</td>
</tr>
<tr>
<td>Physrot</td>
<td>3.8 ± 3.9</td>
<td>3.5 ± 3.8</td>
<td>4.3 ± 5.1</td>
<td>7.8 ± 6.3</td>
</tr>
<tr>
<td>3drot</td>
<td>4.4 ± 4.9</td>
<td>4.5 ± 5.7</td>
<td>4.4 ± 5.6</td>
<td>9.8 ± 8.2</td>
</tr>
</tbody>
</table>

Values (in μm) are average ± SD errors along x, y, and z dimensions and total errors grouped by rotational data sets. The average of total error is computed over the 3-dimensional distances and is not derived from the mean x, y, and z errors.

Table B4. Localization by CNN size: statistical analysis

<table>
<thead>
<tr>
<th>XS</th>
<th>S</th>
<th>M</th>
<th>L</th>
<th>XL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XS</td>
<td>0.26***</td>
<td>0.49***</td>
<td>0.63***</td>
<td>0.71***</td>
</tr>
<tr>
<td>S</td>
<td>ns</td>
<td>0.23***</td>
<td>0.38***</td>
<td>0.47***</td>
</tr>
<tr>
<td>M</td>
<td>ns</td>
<td>ns</td>
<td>0.14***</td>
<td>0.22***</td>
</tr>
<tr>
<td>L</td>
<td>ns</td>
<td>ns</td>
<td>ns</td>
<td>0.09***</td>
</tr>
<tr>
<td>XL</td>
<td>ns</td>
<td>ns</td>
<td>ns</td>
<td>ns</td>
</tr>
</tbody>
</table>

Statistical analysis for localization errors grouped by convolutional neural network (CNN) size. Each entry shows Cohen’s d and significance of the test column group < row group. **P < 0.01, Not significant.

DISCLOSURES
No conflicts of interest, financial or otherwise, are declared by the authors.

AUTHOR CONTRIBUTIONS

REFERENCES


Table B3. Localization by data set rotation: statistical analysis

<table>
<thead>
<tr>
<th></th>
<th>Norot</th>
<th>Physrot</th>
<th>3drot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Na</td>
<td>0.0***</td>
<td>ns</td>
<td>ns</td>
</tr>
<tr>
<td>Rep</td>
<td>ns</td>
<td>0.26***</td>
<td>ns</td>
</tr>
<tr>
<td>NaRep</td>
<td>ns</td>
<td>0.0***</td>
<td>0.27***</td>
</tr>
<tr>
<td>Waveform</td>
<td>ns</td>
<td>ns</td>
<td>ns</td>
</tr>
</tbody>
</table>

Statistical analysis for localization errors grouped by rotational data set. Each entry shows Cohen’s d and significance of the test column group < row group. Na, extracellular action potential (EAP) negative peak (mainly attributed to sodium currents flowing into soma); Rep, EAP positive peak (associated with cell repolarization phase); NaRep, stacked version of Na and Rep; Waveform, downsampled EAP waveforms. **P < 0.001, ***P < 0.01, Not significant.
Table B6. Localization by MEA probe: statistical analysis

<table>
<thead>
<tr>
<th></th>
<th>Sq15-10</th>
<th>Sq10-15</th>
<th>Sq7-20</th>
<th>Sq6-25</th>
<th>Sq5-30</th>
<th>Neuronexus</th>
<th>NeuroSeeker</th>
<th>NeuroPixels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sq15-10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sq10-15</td>
<td>0.04***</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sq7-20</td>
<td>0.08***</td>
<td>0.04**</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sq6-25</td>
<td>0.1***</td>
<td>0.06***</td>
<td>0.02***</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sq5-30</td>
<td>0.13***</td>
<td>0.1***</td>
<td>0.06***</td>
<td>0.04***</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Neuronexus</td>
<td>0.13***</td>
<td>0.1***</td>
<td>0.06**</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NeuroSeeker</td>
<td>0.25***</td>
<td>0.22***</td>
<td>0.18**</td>
<td>0.16***</td>
<td>0.13***</td>
<td>0.12***</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NeuroPixels</td>
<td>0.43***</td>
<td>0.41***</td>
<td>0.37***</td>
<td>0.36***</td>
<td>0.32***</td>
<td>0.3***</td>
<td>0.19***</td>
<td></td>
</tr>
</tbody>
</table>

Table B7. Localization with different methods: statistical analysis

<table>
<thead>
<tr>
<th></th>
<th>Monopolar</th>
<th>Bipolar</th>
<th>B-A-S</th>
<th>CNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monopolar</td>
<td>0.33***</td>
<td>ns</td>
<td>0.9***</td>
<td></td>
</tr>
<tr>
<td>Bipolar</td>
<td>ns</td>
<td></td>
<td>0.68*</td>
<td></td>
</tr>
<tr>
<td>B-A-S</td>
<td>0.04*</td>
<td>0.36**</td>
<td>0.87***</td>
<td></td>
</tr>
<tr>
<td>CNN</td>
<td>ns</td>
<td>ns</td>
<td>ns</td>
<td></td>
</tr>
</tbody>
</table>

Table B8. Classification accuracies by rotation

<table>
<thead>
<tr>
<th></th>
<th>Norot</th>
<th>Phystot</th>
<th>3drot</th>
</tr>
</thead>
<tbody>
<tr>
<td>BTC</td>
<td>98.4</td>
<td>99.3</td>
<td>99.5</td>
</tr>
<tr>
<td>ChC</td>
<td>96.0</td>
<td>84.8</td>
<td>84.1</td>
</tr>
<tr>
<td>DBC</td>
<td>100.0</td>
<td>99.3</td>
<td>99.1</td>
</tr>
<tr>
<td>LBC</td>
<td>100.0</td>
<td>99.5</td>
<td>97.9</td>
</tr>
<tr>
<td>MC</td>
<td>100.0</td>
<td>100.0</td>
<td>100.0</td>
</tr>
<tr>
<td>NBC</td>
<td>99.8</td>
<td>98.1</td>
<td>98.2</td>
</tr>
<tr>
<td>SBC</td>
<td>100.0</td>
<td>98.9</td>
<td>98.4</td>
</tr>
<tr>
<td>STPC</td>
<td>97.4</td>
<td>96.0</td>
<td>96.5</td>
</tr>
<tr>
<td>TTPC1</td>
<td>92.6</td>
<td>100.0</td>
<td>99.5</td>
</tr>
<tr>
<td>TTPC2</td>
<td>100.0</td>
<td>100.0</td>
<td>99.1</td>
</tr>
<tr>
<td>UTPC</td>
<td>98.7</td>
<td>99.8</td>
<td>98.7</td>
</tr>
<tr>
<td>Average</td>
<td>98.1</td>
<td>98.0</td>
<td>97.6</td>
</tr>
<tr>
<td>SD</td>
<td>2.4</td>
<td>3.9</td>
<td>3.9</td>
</tr>
</tbody>
</table>

Statistical analysis for localization errors grouped by probe type. Each entry shows Cohen’s d and significance of the test column group < row group. MEA, multielectrode array. ***p < 0.001, **p < 0.01, ns, Not significant.
Table B9. **CNN size classification performance**

<table>
<thead>
<tr>
<th>XS</th>
<th>S</th>
<th>M</th>
<th>L</th>
<th>XL</th>
</tr>
</thead>
<tbody>
<tr>
<td>BTC</td>
<td>96.8</td>
<td>98.4</td>
<td>99.3</td>
<td>99.3</td>
</tr>
<tr>
<td>ChC</td>
<td>73.6</td>
<td>79.5</td>
<td>97.0</td>
<td>84.8</td>
</tr>
<tr>
<td>DBC</td>
<td>97.4</td>
<td>99.1</td>
<td>98.9</td>
<td>99.3</td>
</tr>
<tr>
<td>LBC</td>
<td>96.1</td>
<td>97.4</td>
<td>99.6</td>
<td>99.5</td>
</tr>
<tr>
<td>MC</td>
<td>99.5</td>
<td>100.0</td>
<td>99.8</td>
<td>100.0</td>
</tr>
<tr>
<td>NBC</td>
<td>94.2</td>
<td>96.5</td>
<td>96.1</td>
<td>98.1</td>
</tr>
<tr>
<td>SBC</td>
<td>96.7</td>
<td>98.2</td>
<td>97.9</td>
<td>98.9</td>
</tr>
<tr>
<td>STPC</td>
<td>100.0</td>
<td>100.0</td>
<td>98.7</td>
<td>96.0</td>
</tr>
<tr>
<td>TTPC1</td>
<td>100.0</td>
<td>99.8</td>
<td>99.4</td>
<td>100.0</td>
</tr>
<tr>
<td>TTPC2</td>
<td>99.8</td>
<td>99.4</td>
<td>99.6</td>
<td>100.0</td>
</tr>
<tr>
<td>UTPC</td>
<td>97.6</td>
<td>97.9</td>
<td>97.8</td>
<td>99.8</td>
</tr>
<tr>
<td>Average</td>
<td>96.4</td>
<td>97.4</td>
<td>98.6</td>
<td>98.0</td>
</tr>
<tr>
<td>SD</td>
<td>6.6</td>
<td>5.1</td>
<td>1.1</td>
<td>3.9</td>
</tr>
</tbody>
</table>

Values are excitatory/inhibitory classification accuracy (in %) grouped by size and cell type. BTC, btufted cells; ChC, chandelier cells; DBC, double bouquet cells; LBC, large basket cells; MC, Martinotti cells; NBC, nest basket cells; SBC, small basket cells; STPC, slender-tufted pyramidal cells (PC); TTPC1, thick-tufted PC with late bifurcating apical tuft; TTPC2, thick-tufted PC with early bifurcating apical tuft; UTPC, untufted PC.

Table B10. **Feature classification performance**

<table>
<thead>
<tr>
<th>AW</th>
<th>FW</th>
<th>AFW</th>
<th>Waveform</th>
</tr>
</thead>
<tbody>
<tr>
<td>BTC</td>
<td>99.8</td>
<td>98.4</td>
<td>99.3</td>
</tr>
<tr>
<td>ChC</td>
<td>91.6</td>
<td>79.3</td>
<td>97.0</td>
</tr>
<tr>
<td>DBC</td>
<td>97.7</td>
<td>98.4</td>
<td>98.9</td>
</tr>
<tr>
<td>LBC</td>
<td>99.3</td>
<td>97.7</td>
<td>99.6</td>
</tr>
<tr>
<td>MC</td>
<td>100.0</td>
<td>100.0</td>
<td>99.8</td>
</tr>
<tr>
<td>NBC</td>
<td>91.9</td>
<td>93.7</td>
<td>96.1</td>
</tr>
<tr>
<td>SBC</td>
<td>99.6</td>
<td>98.4</td>
<td>97.9</td>
</tr>
<tr>
<td>STPC</td>
<td>99.9</td>
<td>95.6</td>
<td>98.7</td>
</tr>
<tr>
<td>TTPC1</td>
<td>99.9</td>
<td>100.0</td>
<td>99.9</td>
</tr>
<tr>
<td>TTPC2</td>
<td>99.7</td>
<td>99.9</td>
<td>99.6</td>
</tr>
<tr>
<td>UTPC</td>
<td>97.0</td>
<td>99.8</td>
<td>97.8</td>
</tr>
<tr>
<td>Average</td>
<td>98.1</td>
<td>97.0</td>
<td>98.6</td>
</tr>
<tr>
<td>SD</td>
<td>2.8</td>
<td>5.3</td>
<td>1.1</td>
</tr>
</tbody>
</table>

Values are excitatory/inhibitory classification accuracy (in %) grouped by feature and cell type. W, peak-to-peak width; F, full-width at half-maximum; A, peak-to-peak amplitude; BTC, btufted cells; ChC, chandelier cells; DBC, double bouquet cells; LBC, large basket cells; MC, Martinotti cells; NBC, nest basket cells; SBC, small basket cells; STPC, slender-tufted pyramidal cells (PC); TTPC1, thick-tufted PC with late bifurcating apical tuft; TTPC2, thick-tufted PC with early bifurcating apical tuft; UTPC, untufted PC.
Table B11. MEA probe classification performance

<table>
<thead>
<tr>
<th>Probe</th>
<th>BTC</th>
<th>ChC</th>
<th>DBC</th>
<th>LBC</th>
<th>MC</th>
<th>NBC</th>
<th>SBC</th>
<th>STPC</th>
<th>TTPC1</th>
<th>TTPC2</th>
<th>UTPC</th>
<th>Average</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEA</td>
<td>98.9</td>
<td>99.3</td>
<td>98.4</td>
<td>98.9</td>
<td>99.1</td>
<td>99.1</td>
<td>99.1</td>
<td>99.1</td>
<td>99.1</td>
<td>99.1</td>
<td>99.6</td>
<td>98.9</td>
<td>4.8</td>
</tr>
<tr>
<td>NeuroSeeker</td>
<td>99.1</td>
<td>99.3</td>
<td>98.4</td>
<td>98.9</td>
<td>99.1</td>
<td>99.1</td>
<td>99.1</td>
<td>99.1</td>
<td>99.1</td>
<td>99.1</td>
<td>99.6</td>
<td>98.9</td>
<td>4.8</td>
</tr>
<tr>
<td>NeuroPixels</td>
<td>99.1</td>
<td>99.3</td>
<td>98.4</td>
<td>98.9</td>
<td>99.1</td>
<td>99.1</td>
<td>99.1</td>
<td>99.1</td>
<td>99.1</td>
<td>99.1</td>
<td>99.6</td>
<td>98.9</td>
<td>4.8</td>
</tr>
</tbody>
</table>

Values are excitatory/inhibitory classification accuracy (in %) grouped by probe and cell type. MEA, multielectrode array; BTC, bitufted cells; ChC, chandelier cells; DBC, double bouquet cells; LBC, large basket cells; MC, Martinotti cells; NBC, nest basket cells; SBC, small basket cells; STPC, slender-tufted pyramidal cells (PC); TTPC1, thick-tufted PC with late bifurcating apical tuft; TTPC2, thick-tufted PC with early bifurcating apical tuft; UTPC, untufted PC.


V. Combining biophysical modeling and deep learning for multielectrode array neuron localization and classification

Errata corrige Paper V

In Figure 7C the INHIB and EXCIT labels are reversed.
Paper VII

How does the presence of neural probes affect extracellular potentials?
How does the presence of neural probes affect extracellular potentials?

Alessio Paolo Buccino1,2, Miroslav Kuchta3, Karoline Horgmo Jæger4, Torbjørn Vefferstad Ness1,5, Pierre Berthet1, Kent-Andre Mardal3,4, Gert Cauwenberghs2, and Aslak Tveito4

1 Center for Integrative Neuroplasticity (CINPLA), Faculty of Mathematics and Natural Sciences, University of Oslo, Oslo, Norway
2 Department of Bioengineering, University of California San Diego, San Diego, CA, United States of America
3 Department of Mathematics, University of Oslo, Oslo, Norway
4 Simula Research Laboratory, Oslo, Norway
5 Faculty of Science and Technology, Norwegian University of Life Sciences, Ås, Norway

E-mail: alessiob@ifi.uio.no

Received 27 March 2018, revised 24 January 2019
Accepted for publication 31 January 2019
Published 26 February 2019

Abstract

Objective. Mechanistic modeling of neurons is an essential component of computational neuroscience that enables scientists to simulate, explain, and explore neural activity. The conventional approach to simulation of extracellular neural recordings first computes transmembrane currents using the cable equation and then sums their contribution to model the extracellular potential. This two-step approach relies on the assumption that the extracellular space is an infinite and homogeneous conductive medium, while measurements are performed using neural probes. The main purpose of this paper is to assess to what extent the presence of the neural probes of varying shape and size impacts the extracellular field and how to correct for them. Approach. We apply a detailed modeling framework allowing explicit representation of the neuron and the probe to study the effect of the probes and thereby estimate the effect of ignoring it. We use meshes with simplified neurons and different types of probe and compare the extracellular action potentials with and without the probe in the extracellular space. We then compare various solutions to account for the probes’ presence and introduce an efficient probe correction method to include the probe effect in modeling of extracellular potentials. Main results. Our computations show that microwires hardly influence the extracellular electric field and their effect can therefore be ignored. In contrast, multi-electrode arrays (MEAs) significantly affect the extracellular field by magnifying the recorded potential. While MEAs behave similarly to infinite insulated planes, we find that their effect strongly depends on the neuron-probe alignment and probe orientation. Significance. Ignoring the probe effect might be deleterious in some applications, such as neural localization and parameterization of neural models from extracellular recordings. Moreover, the presence of the probe can improve the interpretation of extracellular recordings, by providing a more accurate estimation of the extracellular potential generated by neuronal models.

Original content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
1. Introduction

Huge efforts have been invested in computational modeling of neurophysiology over the last decades. This has led to the development and public distribution of a large array of realistic neuron models, for example from the Blue Brain Project (bbp.epfl.ch, 1, 2), the Allen-Brain Institute brain cell database (celltypes.brain-map.org 3), and the Neuromorpho database (neuromorpho.org 4, 5). As experimental data become available, these models become both more elaborate and more accurate. However, some of the assumptions underlying the most commonly used models may not allow the accuracy necessary to obtain good agreements between models and experiments. For instance, it was pointed out in Tveito et al 6 that assumptions underlying the classical cable equation and the associated method for computing the extracellular potential, lead to significant errors both in the membrane potential and the extracellular potential. In the present paper we investigate whether the classical modeling techniques used in computational neurophysiology are sufficiently accurate to reflect measurements obtained by different types of probes, such as microwires/tetrodes, and larger silicon multi-electrode arrays (MEAs). Traditionally, these devices are not represented in the models describing the extracellular field, and our aim is to see if this omission introduces significant errors and how this mismatch could be accounted for in modeling of extracellular activity.

The most widely accepted and used modeling framework for computing the electrophysiology of neurons is the cable equation 7–12, which is used to find current and membrane potentials at different segments of a neuron. One straightforward and computationally convenient way to model the extracellular electric potential generated by neural activity is to sum the individual contributions of the transmembrane currents (computed for each segment) considering them as point current sources or line current sources 7, 11 using volume conductor theory. Although this approach represents the gold standard in computational neuroscience, there are some essential assumptions that need to be discussed. First, (i) the neuron is represented as a cable of discrete nodes and the continuous nature of its membrane is not preserved. Second, (ii) when solving the cable equation, the extracellular potential is neglected, but the extracellular potential is computed a posteriori. Third, and foremost, (iii) when computing extracellular potentials, the tissue in which the neuron lies is modeled as an infinite medium with homogeneous properties. The validity of these assumptions must be addressed in light of the specific application under consideration. The first assumption (i) can be justified by increasing the number of nodes in the model, but assumption (ii) is harder to relax since it means that the model ignores ephaptic effects. Therefore, this assumption has gained considerable attention 6, 13–18. However, the main focus of the present paper is assumption (iii). More specifically our aim is to study the effect of the physical presence of a neural probe on the extracellular signals. Can it be neglected in the mathematical model, or should it be included as a restriction on the extracellular domain? Specifically, is the conventional modeling framework, ignoring the effect of the probes, sufficient to yield reliable prediction of extracellular potentials? Finally, what can modelers do in order to represent and include the effect of recording probes?

In order to investigate this question, we have used the extracellular-membrane-intracellular (EMI) model 6, 19, 20. The EMI model allows for explicit representation of both the intracellular space of the neuron, the cell membrane and the extracellular space surrounding the neuron. Therefore, the geometry of neural probes can be represented accurately in the model. We have run finite element simulations of simplified pyramidal cells combined with different types of probes, such as microwires/tetrodes, and larger silicon multi-electrode arrays (MEAs).

Our computations strongly indicate that the effect of the probe depends on several factors; small probes (microwires) have little effect on the extracellular potential, whereas larger devices (such as multi-electrode arrays, MEAs) change the extracellular potential quite dramatically, resembling the effect of a non-conductive infinite plane in the proximity of the neuron. The effect, however, depends on the neuron-probe alignment and orientation. We then compare the EMI results with conventional cable equation-based techniques, such as the current summation approach 11, 20, the hybrid solution 20–23, and the method of images 24, 25 and introduce the probe correction method, which allows to reach a hybrid solution accuracy leveraging on a pre-mapping of the probe-specific effect and the reciprocity principle.

The results may aid in understanding experimental data recorded with MEAs, it may improve accuracy when extracellular potentials are used to parameterize membrane models as advocated in 26, and to localize and classify neurons from MEA recordings 27, 28.

The rest of the article is organized as follows: in section 2 we describe the methods used throughout the paper, with particular focus on the EMI model (section 2.1), the meshes (section 2.2), the finite element framework (section 2.3), and modeling approaches used for comparison (section 2.4). In section 3 we present our findings related to the effect of probes of different geometry on the extracellular recordings (section 3.1), the variability of our simulations depending on geometrical parameters of the mesh (section 3.2), before comparing them with results obtained from other computational...
approaches (section 3.3) and the relative computational costs of these methods (section 3.4). Finally, we discuss and contextualize the work in section 4.

2. Methods

In this section we introduce the modeling frameworks used to investigate the effect of the probes on the extracellular potential. In particular we first describe the EMI model, the meshes, and the membrane and finite element modeling. Then, we describe the conventional modeling based on the cable equation solution: the current summation approach (CS), the hybrid solution (HS) and the method of images (MoI). Finally, we introduce the probe correction method (PC), which reaches the hybrid solution accuracy in a more efficient and computationally-cheap way.

2.1. The extracellular-membrane-intracellular model

The purpose of the present report is to estimate the effect of introducing a probe in the extracellular domain on the extracellular potential. This can be done using a model discussed in [6, 19, 29–31] referred to as the EMI model. In the EMI model the extracellular space surrounding the neuron, the membrane of the neuron and the intracellular space of the neuron are all explicitly represented in the model. The model takes the form

\[ \nabla \cdot \sigma_e \nabla u_e = 0 \quad \text{in } \Omega_e, \]  
\[ \nabla \cdot \sigma_m \nabla u_m = 0 \quad \text{in } \Omega_m, \]  
\[ u_e = 0 \quad \text{at } \partial \Omega_e, \]  
\[ \sigma_e \nabla u_e \cdot n_e = 0 \quad \text{at } \partial \Omega_p, \]  
\[ n_e \cdot \nabla u_e = -n_i \cdot \sigma_i \nabla u_i \text{def } I_m \quad \text{at } \Gamma_i, \]  
\[ u_i - u_e = v \quad \text{at } \Gamma_i, \]  
\[ \frac{\partial v}{\partial t} = \frac{1}{C_m}(I_m - I_{ion}) \quad \text{at } \Gamma_i. \]

In the simplified geometry sketched in figure 1, \( \Omega \) denotes the total computational domain consisting of the extracellular domain \( \Omega_e \) and the intracellular domain \( \Omega_m \), and the cell membrane is denoted by \( \Gamma \). \( n_i \) and \( n_e \) are the vectors normal to \( \Gamma \) pointing out of the intra- and extracellular domains, respectively. \( u_i \) and \( u_e \) denote the intra- and extracellular potentials, and \( v = u_i - u_e \) denotes the membrane potential defined at the membrane \( \Gamma \). The intra- and extracellular conductivities are given respectively by \( \sigma_i \) and \( \sigma_e \) and in this work we assume that the quantities are constant scalars. The cell membrane capacitance is given by \( C_m \), and the ion current density is given by \( I_{ion} \). \( I_m \) is the total current current escaping through the membrane.

The EMI model is here considered with grounding (Dirichlet) boundary conditions, i.e. \( u_e = 0 \), on the boundary of the extracellular domain (\( \partial \Omega_e \)) while insulating (Neumann) boundary conditions, i.e. \( \sigma_e \nabla u_e \cdot n_e = 0 \), were prescribed at the surface of the probe (\( \partial \Omega_p \)). Note that the latter is a suitable boundary condition also for the conducting surfaces of the probe [25, 32]. The resting potential (see table 1) is used as initial condition for \( v \).

2.2. Meshes

In order to implement the EMI model described above, the computational domain was discretized by unstructured tetrahedral meshes generated by \textsc{gmsh} [33]. We used a simplified neuron model similar to a ball-and-stick model [34, 35], with a spherical soma with 20 \( \mu \)m diameter—whose center is in the origin of the axis—an apical dendrite of length \( L_a = 400 \) \( \mu \)m and diameter \( D_a = 5 \mu \)m in the positive \( z \) direction and an axon of length \( L_d = 200 \mu \)m and diameter \( D_d = 2 \mu \)m in the negative \( z \) direction. Both the axon and the dendrites are connected to the soma via a tapering in the geometry. On the dendritic side, the diameter at the soma is 8 \( \mu \)m and it linearly reduces to 5 \( \mu \)m in a 20 \( \mu \)m portion. On the axonial side, the axon hillock has a diameter of 4 \( \mu \)m at the soma and it is tapered to 2 \( \mu \)m in 10 \( \mu \)m.

The neuron was placed in a box with and without neural probes to study the effect of the recording device on the simulated signals. We used three different types of probes:

- **Microwire**: the first type of probe represents a microwire type of probe (or tetrode). For this kind of probes we used a cylindrical insulated model with 30 \( \mu \)m diameter. The extracellular potential, after the simulations, was estimated as the average of the electric potential measured at the tip of the cylinder. The microwire probe is shown in figure 2(A) alongside with the simplified neuron.

- **Neuronexus (MEA)**: the second type of probe model represents a commercially available silicon MEA (A1x32-Poly3-5mm-25s-177-CM32 probe from Neuronexus Technologies), which has 32 electrodes in three columns (the central column has 12 recording sites and first and third columns have 10) with hexagonal arrangement, a \( y \)-pitch of 18 \( \mu \)m, and a \( z \)-pitch of 22 \( \mu \)m. The electrode radius is 7.5 \( \mu \)m. This probe has a thickness of 15 \( \mu \)m and a maximum width of 114 \( \mu \)m, and it is shown in figure 2(B).

- **Neuropixels (MEA)**: the third type of probe model represents the Neuropixels silicon MEA [36]. The original probe has more than 900 electrodes over a 1 cm shank, it is 70 \( \mu \)m wide and 20 \( \mu \)m thick. In our mesh, shown in figure 2(C) we used 24 \( 12 \times 12 \) \( \mu \)m recording sites arranged in the chessboard configuration with an inter-electrode-distance of 25 \( \mu \)m [36].

In order to evaluate the effect of the described probes depending on the relative distance to the neuron (\( x \) direction), we generated several meshes in which the distance between the contact sites and the center of the neuron was 17.5, 22.5, 27.5, 37.5, 47.5, and 77.5 \( \mu \)m. Note that these distances refer to the beginning of the microwire tip (which extends in the \( x \) direction for 30 \( \mu \)m) and to the MEA \( y - z \) plane (for the MEA...
probes the recording sites do not extend in the x direction). When not specified, instead, the distance for the microwire probe was 25 µm, 32.5 µm for the Neuronexus MEA probe, and 30 µm for the Neuropixels probe (center of the probe tip at 40 µm).

To investigate if and how the bounding box size affects the simulation, since the electric potential is set to zero at its boundary, instead, the distance for the microwire and for the bounding box surface, respectively, the mesh element sizes for the neuron volume and membrane, for each of the computational domain.

Figure 1. Sketch of the simplified neuron geometry and its surroundings. The intracellular domain is denoted by \( \Omega_i \), the cell membrane is denoted by \( \Gamma \), and the extracellular domain is denoted by \( \Omega_e \). The boundary of the probe is denoted by \( \partial\Omega_p \) and the remaining boundary of the extracellular domain is denoted by \( \partial\Omega_e \). The normal vector pointing out of \( \Omega_i \) is denoted by \( n_i \), and \( n_e \) denotes the normal vector pointing out of \( \Omega_e \). L and D are the length and diameter of neural segments, respectively, and \( D_t \) is the diameter of the hillocks in correspondence of the soma. In our simulations, we consider three types of probe geometry (see figure 2).

Moreover, we evaluated the solution convergence depending on the resolution by generating meshes with four different resolutions. Defining \( r_n \), \( r_p \), and \( r_{ext} \) as the resolutions/typical mesh element sizes for the neuron volume and membrane, for the probe, and for the bounding box surface, respectively, the four degrees of coarseness were:

- coarse 0: \( r_n = 2 \mu m, r_p = 5 \mu m, \) and \( r_{ext} = 7.5 \mu m \)
- coarse 1: \( r_n = 3 \mu m, r_p = 6 \mu m, \) and \( r_{ext} = 9 \mu m \)
- coarse 2: \( r_n = 4 \mu m, r_p = 8 \mu m, \) and \( r_{ext} = 12 \mu m \)
- coarse 3: \( r_n = 4 \mu m, r_p = 10 \mu m, \) and \( r_{ext} = 15 \mu m \)

At the interface between two resolutions, the mesh size was determined as their minimum. Further, having instructed gmsh to not allow hanging nodes the mesh in the surroundings of the neuron and probe is gradually coarsened to \( r_{ext} \) resolution.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_m )</td>
<td>1 µF cm(^{-2} )</td>
<td>( g_{syn} )</td>
<td>10 mS cm(^{-2} )</td>
</tr>
<tr>
<td>( \sigma_i )</td>
<td>7 mS cm(^{-1} )</td>
<td>( v_{eq} )</td>
<td>0 mV</td>
</tr>
<tr>
<td>( \sigma_e )</td>
<td>3 mS cm(^{-1} )</td>
<td>( I_0 )</td>
<td>0.01 ms</td>
</tr>
<tr>
<td>( g_L )</td>
<td>0.06 mS cm(^{-2} )</td>
<td>( \alpha )</td>
<td>2 ms</td>
</tr>
<tr>
<td>( v_{rest} )</td>
<td>−75 mV</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For each of the mesh configuration with varying probe model, box size, and coarseness we simulated the extracellular signals with and without the probe in the extracellular space and sampled the electric potential at the recording site locations (even when the probe is absent).

2.3. Membrane model and finite element implementation

On the membrane of the soma and the axon, the ionic current density, \( I_{ion} \), is computed by the Hodgkin–Huxley model with standard parameters as given in [37]. On the membrane of the dendrite, we apply a passive membrane model with a synaptic input current of the form

\[
I_{ion} = I_{\text{leak}} + I_{\text{syn}},
\]

\[
I_{\text{leak}} = g_{L}(v - v_{\text{rest}}),
\]

\[
I_{\text{syn}} = g_{s}(x)e^{-\frac{v-v_{eq}}{\alpha}}, \quad (v - v_{eq}),
\]

where

\[
g_s(x) = \begin{cases} 
    g_{syn}, & \text{for } x \text{ in the synaptic input area}, \\
    0, & \text{elsewhere}.
\end{cases}
\]
The parameters of the dendrite model are given in table 1, and the synaptic input area is defined as a section of the dendrite of length 20 \( \mu \text{m} \) located 350 \( \mu \text{m} \) from the soma, as illustrated in figure 1.

The EMI model (1)–(7) is solved by the operator splitting scheme and the \( \mathcal{H}(\text{div}) \) discretization proposed in [20]. In this scheme a single step of the EMI model consists of two sub-steps. First, assuming the current membrane potential \( v \) is known, the ordinary differential equations (ODE) of the membrane model are solved yielding a new membrane state and the value of \( v \). Next, equation (7), discretized in time with \( I_{\text{ion}} \) set to zero, is solved together with equations (1)–(6) using the computed value of \( v \) as input. This step yields the new values of intra/extra-cellular potentials \( u_i, u_e \) and the transmembrane potential \( v \). The \( \mathcal{H}(\text{div}) \) approach then means that the EMI model is transformed by introducing unknown electrical fields \( \sigma_i \nabla u_i \) and \( \sigma_e \nabla u_e \) in addition to the potentials \( u_i, u_e \) and \( v \). Thus more unknowns are involved, however, the formulation leads to more accurate solutions, see [20, section 3].

In our implementation the ODE solver for the first step of the operator splitting scheme is implemented on top of the computational cardiac electrophysiology framework \texttt{cbc.} beat [38]. For the second step, the \( \mathcal{H}(\text{div}) \) formulation of the EMI model, see [20, section 2.3.3], is discretized by the finite element method (FEM) using the \texttt{FEniCS} library [39]. More specifically, the electrical fields are discretized by the lowest order Raviart–Thomas elements [40] while the potentials use piecewise constant elements. The linear system due to implicit/backward-Euler temporal discretization in equation (7) and FEM is finally solved with the direct solver \texttt{MUMPS} [41] which is interfaced with \texttt{FEniCS} via the \texttt{PETSc} [42] linear algebra library.

**Figure 2.** Visualization of simplified neuron and probe meshes. (A) Microwire: the probe has a 15 \( \mu \text{m} \) radius and it is aligned to the neuronal axis (\( z \) direction) and the center of the probe tip is at (40, 0, 0) \( \mu \text{m} \) (the soma center is at (0, 0, 0) \( \mu \text{m} \)). The axon and soma of the neuron are depicted in yellow, the dendrite is orange, and the axon and dendritic hillock are in cyan. (B) Neuronexus MEA: the probe represents a Neuronexus A1x32-Poly3-5mm-25s-177-CM32 with recording sites facing the neuron. The MEA is 15 \( \mu \text{m} \) thick and the center of the bottom vertex is at (40, 0, −100) \( \mu \text{m} \). The maximum width of the probe is 114 \( \mu \text{m} \), which makes it almost four times larger than the microwire probe. (C) Neuropixels MEA: this probe [36] has a width of 70 \( \mu \text{m} \), a thickness of 20 \( \mu \text{m} \), and the center of the bottom vertex is at (40, 0, −100) \( \mu \text{m} \). All meshes represented here are built with the finest coarseness described in the text (coarse 0).
2.4. Other modeling approaches

2.4.1. Current summation (CS), method of images (MoI), and scaled current summation (SCS). The cable equation [43–45] is of great importance in computational neuroscience, and it reads,

\[ C_m \frac{\partial V}{\partial t} + I_{ion} = \eta \frac{\partial^2 V}{\partial x^2}, \]

(12)

where \( V \) is the membrane potential of the neuron, \( C_m \) is the membrane capacitance, \( I_{ion} \) is the ion current density and \( \eta = \frac{\sigma m}{4} \), where \( h \) is the diameter of the neuron, and \( \sigma \) denotes the intracellular conductivity of the neuron [43].

This equation is used to compute the membrane potential of a neuron and the solution is commonly obtained by dividing the neuron into compartments and replacing the continuous model (12) by a discrete model [43]. In order to compute the associated extracellular potential, it is common to use the solution of the cable equation to compute the transmembrane currents densities in every compartment, and then invoke the classical summation formula,

\[ u_e(x, y, z) = \frac{1}{4\pi \sigma_e} \sum_k \frac{I_k}{|r - r_k|}. \]

(13)

Here, \( \sigma_e \) is the constant extracellular conductivity (in all the implemented models, the milieu is assumed to be linear by using a constant \( \sigma_e \)), \( r_k \) is the center of the \( k \)th compartment of the neuron, \( |r - r_k| \) denotes the Euclidean distance from \( r = (x, y, z) \) to the point \( r_k \), and \( I_k \) denotes the transmembrane current of each compartment. This solution assumes that the extracellular milieu is purely conductive, infinite, and homogeneous. We denote this method as current summation approach (CS) [6].

As the silicon probes are made of insulated material, they could be approximated with the method of images (MoI) [12, 24, 25]. With the MoI the probe is assumed to be an infinite insulating plane, effectively increasing the extracellular potential by a factor of 2. Using the MoI, the factor 2 can be explained as follows: for each current source, an image current source is introduced in the mirror position with respect to the insulating plane, effectively doubling the potential in proximity of the plane and canceling current densities normal to the plane. For the MoI, the summation formula (equation 13) reads:

\[ u_e(x, y, z) = \frac{1}{2\pi \sigma_e} \sum_k \frac{I_k}{|r - r_k|}. \]

(14)

As will be shown section 3.1, the peak scaling factor (1 and 2 for the CS and MoI solutions, respectively) of the modeled probes is modulated by the neuron-probe alignment, rotation, and by the probe type and it can be a value between 0 and 2 depending on these factors. Therefore, we also propose and compare a third current summation-based approach, namely scaled current summation (SCS), in which the scale factor is set to match the peak ratio between the hybrid solution (section 2.4.2) and the CS solution on the electrode with largest amplitude (e.g. 1.65 is used in section 3.3.1).

We implemented the same simulations presented in section 2.1 using the conventional modeling approach described above (CS) to compare them with the EMI simulations. We used LFPy [11, 12], running upon NEURON 7.5 [9, 10], to solve the cable equation and compute extracellular potentials using equation (13). As morphology, we used a ball-and-stick model with an axon with the same geometrical properties described in section 2.2. Similarly to the EMI simulations, we used a synaptic input in the middle of the dendritic region activated in the EMI simulation (\( z = 360 \mu m \)) to induce a single spike and we observed the extracellular potentials on the recording sites. The synaptic weight was adjusted so that the extracellular largest peak was coincident in time with the one from the EMI simulation. To model the spatial extent of the electrodes, we randomly drew 50 points within a recording site and we averaged the extracellular potential computed at these points [11]. We used the same parameters shown in table 1 (note that in NEURON conductances are defined in S cm\(^{-2}\) so we set \( g_L = g_{pas} = 0.06 \cdot 10^{-3} \) S cm\(^{-2}\)) and we used an axial resistance \( R_a \) of 150 \( \Omega \) cm\(^{-1}\). The fixed_length method was used as discretization method with a fixed length of 1 \( \mu m \), yielding 658 segments (23 somatic, 422 dendritic, and 213 axonal). Transmembrane currents were considered as current point sources in their contributions to the extracellular potential, following equation (13) (using LFPy point-source argument of the RecExtElectrode class). The MoI and SCS solutions were calculated by multiplying the CS solution by a factor 2 and 1.65 (optimized scale factor using the hybrid solution).

2.4.2. Hybrid solution (HS). The hybrid solution (HS) [21–23] combines the transmembrane currents for each neural segments computed with the cable equation and a finite element modeling for the extracellular space. The transmembrane currents are used as source terms in a finite element solution of the Poisson Equation in the extracellular space (equation (2), using an iterative solver for the Poisson problem, specifically, preconditioned conjugate gradients with algebraic multigrid preconditioning). With this approach, the probe can be explicitly modeled using insulating (Neumann) boundary conditions at the surface of the probe (equation (5)) and the differences between the HS and the EMI solution lie in differences regarding the modeling of the neuron dynamics, such as the self-ephaptic effect. The HS requires that a FEM simulation is run for each timestep of the transmembrane currents, each time setting the source terms with the currents at the specific timestep. This makes it computationally expensive, especially, for long simulations. Alternatively, one could run a single FEM simulation for each neural segment with a unitary test current and then use the potentials computed at the recording sites as a static map for summing the contribution of all currents at each timestep. The latter approach can be also computationally complex, as the number of segments in the multi-compartment simulation can be quite high and it would require to store in memory a large number of finite element solutions.
2.4.3. Probe correction (PC). The hybrid solution is a good and widely used approach to model a non-homogeneous extracellular space, especially in the peripheral nervous system literature [21–23]. However, it requires to run a finite element simulation for every neuron simulation, as transmembrane currents are located in different positions for different neurons.

In order to overcome this issue, we designed the probe correction method (PC) that relies on the reciprocity principle [46] and the principle of superimposition (given the assumption of linearity of the milieu expressed in section 2.4.1). The reciprocity principle states that if a current \( I_1 \) in a position \((x_1, y_1, z_1)\) generates a potential \( u_1 \) in a second position \((x_2, y_2, z_2)\), then the same current \( I_1 \) placed in \((x_2, y_2, z_2)\) will result in a potential \( u_1 \) in \((x_1, y_1, z_1)\). Using this principle, we first simulated with a finite element method the extracellular potential generated by a test current (1 nA) from each electrode \( i \) of a specific probe (e.g. Neuronexus) in any point of the extracellular space and define it as \( u_i(x_i, y_i, z_i) \), where \((x_i, y_i, z_i)\) is the relative position with respect to the electrode \( i \). Also in this case we used an iterative solver for the Poisson problem (preconditioned conjugate gradients with algebraic multigrid preconditioning). Then, leveraging on the reciprocity and superimposition principles, we mapped the contribution of each transmembrane current to the potential at each electrode \( i \) as: \( u_i = \sum_k I_k u_k (x_k, y_k, z_k) \), where \((x_k, y_k, z_k)\) is now the relative position between the \( k \)th neural segment and the electrode \( i \), and \( I_k \) is the transmembrane current for the \( k \)th neural segment. The potential at each electrode \( i \) can be computed as:

\[
    u_i = \sum_k u_k = \sum_k I_k u_k (x_k, y_k, z_k).
\]

The PC method allows to pre-compute the effect of a probe in the extracellular space and then use this mapping for any neural model, without the need to run a full FEM simulation. The number of FEM solutions that need to be computed and stored during the pre-mapping is equal to the number of electrodes in the probe.

3. Results

In this section we present results of numerical simulations which quantify the effect of introducing probes in the extracellular domain on the extracellular potential. We show how this effect depends on the distance between the neuron and the probe, their lateral alignment, and the probe rotation. The evaluation of the probe effect (section 3.1) is carried out using the EMI simulation framework. Furthermore, we evaluate the numerical variability of the EMI solutions (section 3.2), we compare with other modeling schemes (section 3.3), and finally report CPU-efforts for the simulations (section 3.4).

3.1. The probe effect

3.1.1. The geometry of the probe affects the recorded signals. The first question that we investigated is whether the probes have an effect and, if so, how substantial this effect is and if it depends on the probe geometry. In order to do so we analyzed the extracellular action potential (EAP) traces with and without placing the probe in the mesh.

In figure 3 we show the EAP with and without the microwire probe (A), the Neuronexus probe (B), and the Neuropixels probe (C). The blue traces are the extracellular potentials computed at the recording sites when the probe was removed, while the orange traces show the potential when the probe is present in the extracellular space. In this case the probe tip was placed 40 µm from the soma center, we used a box of size 2 and coarse 2 resolution. It is clear that the probe effect...
is more prevalent for the MEA probes than for the microwire, suggesting that the physical size and geometry of the probe play an important role. In particular, for the Neuronexus probe the minimum peak without the probe is $-21.09 \, \mu V$ and with the probe it is $-41.26 \, \mu V$: the difference is $20.17 \, \mu V$. For the Neuropixels probe the peak with no probe is $-21.2 \, \mu V$, with the probe it is $-44.36 \, \mu V$ and the difference is $23.16 \, \mu V$. In case of the microwire type of probe, the effect is minimal: the minimum peak without the probe is $-16.85 \, \mu V$, with the probe it is $-15.82 \, \mu V$, and the difference is about $1.03 \, \mu V$. In the case of the microwire type of probe, the effect is minimal: the minimum peak without the probe is $-16.85 \, \mu V$, with the probe it is $-15.82 \, \mu V$, and the difference is about $1.03 \, \mu V$. The currents are deflected due to the presence of the probe, and this causes an increase (in absolute value) in the extracellular potential between the neuron and the probe, as shown in panel C, where the difference of the extracellular potential with and without probe is depicted. The substantial effect using the MEA probe probably also depends on the arrangement of the recording sites: while for the MEAs, the electrodes face the neuron (they lie on the $y – z$ plane) and currents emitted by the membrane cannot flow in the $x$ direction, yielding a little effect (figure 4(C) shows that the effect at the tip of the MEA probe is almost null).

Figure 4. Extracellular potential distribution on the $x – z$ plane with the Neuronexus MEA probe (A) without the probe (B), and their difference (C). The images were smoothed with a gaussian filter with standard deviation of $4 \, \mu m$. The color code for panel A and B is the same. The isopotential lines show the potential in $\mu V$. The probe (white area) acts as an insulator, effectively increasing the extracellular potential (in absolute value) in the area between the neuron and the probe (panel C, blue colors close to the soma and red close to the dendrite) and decreasing it behind the probe of several $\mu V$. The effect is smaller at the tip of the probe (the green color represents a $0 \, \mu V$ difference).
3.1.2. The amplitude ratio is constant with probe distance. In this section we analyze the trend of the probe-induced error depending on the vicinity of the probe. We swept the extracellular space from a closest distance between the probe and the somatic membrane of 7.5 µm to a maximum distance of 67.5 µm.

In figures 5(A)–(C) we plot the absolute peak values with (orange) and without probe (blue), as well as their difference (green) for the microwire (A), Neuronexus (B) and Neuropixels (C) probes. For the microwire (A), as observed in the previous section, the probe effect is small and the maximum difference is 1.97 µV, which is 10.1% of the amplitude without probe, when the probe is closest. For the Neuronexus MEA probe (B), at short distances the difference is large—40.5 µV (88.8% of the amplitude without probe) at 7.5 µm probe-membrane distance—and it decreases as the probe distance increases. At the farthest distance, where the probe tip is at 75 µm from the neuron. (D) Ratio between peak with and without probe for the Neuronexus (red), the Neuropixels (blue) and the microwire probe (green). The ratio is almost constant at different distances and the average value is 1.9 for the Neuronexus, 1.91 for the Neuropixels, and 1.05 for the microwire probe.

Figure 5. Differences in EAP maximum absolute value peak with and without probe depending on distance. (A) Microwire probe: maximum peak without probe (blue), with probe (orange), and their difference (green). The difference is small even when the probe is close to the neuron. (B) Neuronexus MEA probe: maximum peak without probe (blue), with probe (orange), and their difference (green). The difference is large at short distances and it decays at larger distances. (C) Neuropixels MEA probe: maximum peak without probe (blue), with probe (orange), and their difference (green). Also for this probe the difference is large at short distances and it reduces at further away from the neuron. (D) Ratio between peak with and without probe for the Neuronexus (red), the Neuropixels (blue) and the microwire probe (green). The ratio is almost constant at different distances and the average value is 1.9 for the Neuronexus, 1.91 for the Neuropixels, and 1.05 for the microwire probe.
somatic membrane, the difference is 4.38 µV, which is 90.2% of the amplitude without probe. For the Neuropixels MEA probe (C) the effect is in line with the Neuronexus probe, with a maximum difference of 41.07 µV (95.9% of the amplitude without probe) when the probe is closest and a minimum of 5.08 µV, which is still 116.1% of the amplitude without probe, when the probe is located at the maximum distance. Note that the peak amplitudes on the microwire probe are smaller than the one measured on the MEAs at a similar distances. At the closest distance, for example, the Neuronexus MEA electrodes lie on the y – z plane exactly at 7.5 µm from the somatic membrane. For the microwire, instead, 7.5 µm is the distance to the beginning of the cylindrical probe, whose tip extends in the x direction for 30 µm. The simulated electric potential is the average of the electric potential computed on the microwire tip and it results in a much lower amplitude due to the fast decay of the extracellular potential with distance (see equation (13)).

In panel (D) of figure 5 we show the ratio between the peak with probe and without probe depending on the probe distance for the Neuronexus (red), Neuropixels (blue), and the microwire (grey) probes. The ratio for the microwire probe varies around 1 (average = 1.05), confirming that the probe effect can be neglected for microwire-like types of probe, due to their size and geometry. Instead, when a MEA probe is used, the average ratio is around 1.9 and its effect on the recordings cannot be neglected.

3.1.3. The probe effect is reduced when neuron and probe are not aligned. So far, we have shown results in which the neuron and the probe are perfectly aligned in the y direction, but the probe effect is likely to be affected by the neuron-probe alignment, since the area of the MEA probe (we focus here on the Neuronexus and Neuropixels MEA probes as the effect using the microwire is negligible) facing the neuron changes depending on the lateral shift in the y direction and probe rotation.

To quantify the trend of the probe effect depending on the y shift, we ran simulations moving the probes at different y locations (10, 20, 30, 40, 50, 60, 80, and 100 µm) and computed the ratios between the maximum peak with and without the MEA in the extracellular space. The simulations were run with coarse 2 resolution and boxsize 5 and the probe tip was at 40 µm from the center of the neuron. In figure 6(A) we show the peak ratios depending on lateral y shifts. The ratio appears to decrease almost linearly with the shifts, from a value of around 1.8–1.9 when the probe is centered (note that the peak ratio slightly varies depending on resolution and size, as covered in section 3.2) to a value of around 1.2 when the shift is 100 µm (the half width of the probe is 57 µm for Neuronexus and 35 µm for Neuropixels).

In order to evaluate the effect of rotating the probes, we ran simulations with the probe at 70 µm distance (to accommodate for different rotations), coarse 2 resolution, boxsize 4, and rotations of 0, 30, 60, 90, 120, 150, and 180°. In figure 6(B) the peak ratios depending on the rotation angle are shown. For small or no rotations (0, 30°) the value is around 1.7 (note that we always selected the electrode with the largest amplitude, which might not be the same electrode for all rotations). For a rotation of 90° the peak ratio is around 1 (the probe exposes its thinnest side to the neuron) and for further rotations the probe’s shadowing effect makes the peak with the probe smaller (as observed in figure 4(C)), yielding peak ratio values below 1. These results demonstrate that the
The relative arrangement between the neuron and the probe play an important role in affecting the recorded signals.

3.2. EMI solution dependence on domain size and resolution

We generated meshes of four different resolutions and five different box sizes, as described in section 2.2, in order to investigate how the resolution and the domain size affect the finite element solutions. Since we are mainly interested in how the probe affects the extracellular potential and we showed that only for MEA probes this effect is large, we focus on the extracellular potential at the recording site with the maximum negative peak. We used the Neuronexus MEA probe for this analysis and the distance of the tip of the probe was 40 µm (the recording sites plane is at 32.5 µm from the somatic center). The recording site which experienced the largest potential deflection was at position (32.5, 0, −13) µm, i.e. the closest to the neuron soma in the axon direction. For a deeper examination of convergence of the EMI model refer to [6]. For resolutions coarse 0 and coarse 1 the box of size 4 and 5, and of size 5, respectively, were too large to be simulated.

In table 2 we show the values of the minimum EAP peak with and without the Neuronexus probe, their difference, and their ratio grouped by the domain (box) size and averaged over resolution. Despite some variability due to the numerical solution of the problem, there is a common trend in the peak values as the domain size increases: the minimum peaks tend to be larger in absolute values, both when the probe is in the extracellular space (from −40.12 µV for box size 1 to −43.09 µV for box size 5) and when it is not (from −20.64 µV for box size 1 to −23.71 µV for box size 5). This can be explained by the boundary conditions that we defined for the bounding box (equation (3)), which forces the electric potential at the boundaries to be 0. For this reason, a smaller domain size causes a steeper reduction of the extracellular potential from the neuron to the bounding box, making the peak amplitude, in absolute terms, smaller. The peak difference with and without the MEA probe appears to be relatively constant, but the peak ratio tends to slightly decrease with increasing domain size for the same reason expressed before (from 1.95 for box size 1 to 1.82 for box size 5). The solutions appear to be converging for box sizes 4 and 5, but the relative error (difference between box 1 and box 5 values divided by the value of box 5) is moderate (6.89% for the peak with probe, 12.95% for the peak without probe, and 4.14% for the peak ratio). Nevertheless, the 1.8–1.85 peak ratio values obtained with larger domain sizes should be a closer estimate of the true value.

Table 3 displays the same values of table 2, but with a fixed box size of 2 and varying resolution (Coarseness). The relative error (maximum difference across resolutions divided by the average values among resolutions) of the peak with the MEA is 3.3%, without the probe it is 6.65%, and for the peak ratio it is 3.53%.

Because the main purpose of this work was to qualitatively investigate the effect of various probe designs and the effect of distance, alignment, and rotation on the measurements, we used resolution coarse 2 and box size 2, which represented an acceptable compromise between accuracy and simulation time. For investigating the effect of probe rotation and side shift we increased the box size to 4 and 5, respectively, to accommodate the position of the neural probe. Finally, in section 3.3 we increased the resolution to coarse 0 and used box size 3 to obtain more accurate results for the comparison with the cable equation simulations.

3.3. Comparison with other approaches

After having investigated how an extracellular probe affects the amplitude of the recorded potentials and how this amplitude is modulated with distance, alignment, and rotation between the neuron and the probe, we now compare the EMI solution to other modeling approaches. We first analyze the differences between the EMI solution without the probe and the cable equation / current summation approach (CS) and between the EMI solution with the probe and the hybrid solution (HS). Then we focus on the HS, which combines a cable equation solution and an explicit model of the extracellular space, including the probe, in a FEM framework, and compare its solution to three correction strategies: the method of images (MoI), the scaled current summation (SCS), and the probe correction (PC).

In all the following simulations we used a mesh with coarse 0 resolution and box size 3. The distance between the neuron soma center and the probe tip was 40 µm, resulting in recording sites on the x = 32.5 µm plane.

3.3.1. EMI, CS, and HS comparison. In order to compare the EMI simulations to conventional modeling, we built the
same scenario shown in figure 2(B) (Neuronexus probe) using Neuron and LFPy, as described in section 2.4. As conventional modeling assumes an infinite and homogeneous medium, we compared the EAPs obtained by combining the cable equation solution (equation (12)) and the current summation formula (equation (13)) with the EMI simulations without the probe. The extracellular traces for the current summation approach (CS, red) and the EMI model (blue) are shown in figure 7(A). The EAPs almost overlap for every recording site, despite some differences in amplitude. On the electrode with the largest peak, the value for the EMI solution is $-23.03 \, \mu V$, while the value for the CS is $-27.95 \, \mu V$ (the difference is $4.91 \, \mu V$). This difference, which has been previously observed, is intrinsic to the EMI model [6], and can be due to self-ephaptic effects [6, 13–18]. Note also that the condition that forces the extracellular potential to zero at the boundary of the domain causes a steeper descent in the extracellular amplitudes, as discussed in section 3.2.

The hybrid solution (HS) uses currents computed with the cable equation and runs a FEM simulation of the extracellular space, including the probe. In figure 7(B) we show the extracellular potential of the EMI simulation with probe (orange) and the HS (green). Also in this case we observe that the EMI solution yields slightly smaller amplitudes with respect to the HS (EMI peak: $-42.6 \, \mu V$; HS peak: $-46.15 \, \mu V$; difference: $3.55 \, \mu V$) and these differences can be once again traced back to underlying differences of the neural solver.

3.3.2. HS, Mol, SCS, and PC comparison. After having shown that there are intrinsic differences between the EMI model and solutions based on the cable equation (CS, HS), we now compare two computationally less expensive strategies that could be used to account for the probe effect in modeling of extracellular potentials.

The Mol and SCS are attractive candidates due to their almost null computational cost, as they only multiply all values by a constant factor. The factor for infinite insulated planes, as described in section 2.4.1, is 2, but as shown in figures 5 and 6, for MEA probes it is somewhere between 0 and 2 depending on the neuron-probe lateral shift and rotation. In this scenario, the neuron is perfectly aligned with the probe and there is no rotation. The peak ratio for the SCS was computed by dividing the largest peaks of the HS and CS solutions and it was set to 1.65. In figure 8(A) the EAP from the HS (green), from the Mol (pink), and from the SCS with factor 1.65 (grey) are displayed. The Mol (pink) overshoots the estimation of the extracellular amplitudes (Mol peak: $-55.89 \, \mu V$; HS $-46.15 \, \mu V$; difference: $9.74 \, \mu V$). The SCS solution, expectedly, results in the same amplitude as the HS on the electrode with the largest peak, as the scaling factor was computed using the actual peak ratio between the HS and the CS solution. However, there are some discrepancies between HS and SCS. Figure 8(B) shows the distribution of peak ratios of all the 32 electrodes with respect to the HS peaks. The CS, Mol, and SCS solutions display a range of values in the peak ratios, showing that the amplitude modulation of the electrodes is not a constant value. This can be traced back to the fact that a lateral shift of the neuron reduces the peak ratio (figure 6(A)): electrodes on the side of the probe yield a lower effect than the ones at the center of the probe. Due to this variability, a correction strategy based on a constant scaling will not be able to accommodate for this effect.
The probe correction (PC) solution, based on the reciprocity principle (section 2.4.3), results in a solution perfectly coincident to the HS, at a much smaller computational cost (see table 5). In figure 8(B) the PC ratios are depicted as a vertical line at 1 because the peak amplitudes are exactly the same as the HS. The PC approach, in fact, pre-maps the effect of each electrode on the extracellular domain, effectively modeling in an efficient way the distribution of peak ratios observed when using the CS, MoI, and SCS methods.

In table 4 we summarize the comparison results, showing maximum, minimum, average peak ratios and the peak ratio distribution standard deviation for all the pairwise comparisons analyzed in this section.

### 3.4. CPU requirements

Whereas the EMI formulation represents a powerful and more detailed computational framework for neurophysiology simulations, it is associated with a much larger computational load. The simulations were performed on an Intel(R) Xeon(R) CPU E5-2623 v4 @ 2.60 GHz machine with 16 cores and 377 GB RAM running Ubuntu 16.04.3 LTS.

Table 5 contains the coarseness, domain size, number of tetrahedral cells, number of mesh vertices, total number of triangular cells (facets), facets on the surface of the neuron, the system size for the FEM problem, and the time in second (CPU time) to compute the solution for meshes without the probe in the extracellular domain. We show the results without probes in the extracellular domain, as they are they are computationally more intense due to the fact that the volume inside the probe is not meshed (although the resolution on the probe surface is finer, the resulting system size without the probe is larger than with the probe). The CPU requirements and the time needed to run the simulation strongly depend on the resolution of the mesh: the problem with coarseness 3 and box size 3 takes around 1 h and 20 min (system size = 745 789), while for the same box size and coarseness 0, the time required is around...
The last four rows show the CPU requirements for the HS model type, FEM system size, resolution (Coarseness), box size, mesh parameters (number of cells, number of facets, number of neuron facets, and vertices), and CPU time to run the simulations. Note that for coarse 2 and coarse 3 the resolution of the neuron ($r_0 = 4 \mu m$) is the same.

<table>
<thead>
<tr>
<th>Model</th>
<th>System size</th>
<th>Coarse</th>
<th>Box size</th>
<th>Mesh Cells</th>
<th>Total facets</th>
<th>Neuron facets</th>
<th>Vertices</th>
<th>T (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMI</td>
<td>337515</td>
<td>3</td>
<td>1</td>
<td>66171</td>
<td>135672</td>
<td>2552</td>
<td>12400</td>
<td>1414.24</td>
</tr>
<tr>
<td>EMI</td>
<td>516079</td>
<td>3</td>
<td>2</td>
<td>101443</td>
<td>207318</td>
<td>2552</td>
<td>18628</td>
<td>2813.22</td>
</tr>
<tr>
<td>EMI</td>
<td>562137</td>
<td>2</td>
<td>1</td>
<td>110363</td>
<td>225887</td>
<td>2480</td>
<td>20420</td>
<td>2589.83</td>
</tr>
<tr>
<td>EMI</td>
<td>745789</td>
<td>3</td>
<td>3</td>
<td>146905</td>
<td>299442</td>
<td>2552</td>
<td>26540</td>
<td>4569.11</td>
</tr>
<tr>
<td>EMI</td>
<td>835365</td>
<td>2</td>
<td>2</td>
<td>164331</td>
<td>335517</td>
<td>2480</td>
<td>29940</td>
<td>4753.39</td>
</tr>
<tr>
<td>EMI</td>
<td>1204001</td>
<td>2</td>
<td>3</td>
<td>237259</td>
<td>483371</td>
<td>2480</td>
<td>42660</td>
<td>10797.78</td>
</tr>
<tr>
<td>EMI</td>
<td>1225082</td>
<td>1</td>
<td>1</td>
<td>241402</td>
<td>491840</td>
<td>3888</td>
<td>43373</td>
<td>9593.98</td>
</tr>
<tr>
<td>EMI</td>
<td>1254096</td>
<td>3</td>
<td>4</td>
<td>247514</td>
<td>503291</td>
<td>2552</td>
<td>44013</td>
<td>10756.46</td>
</tr>
<tr>
<td>EMI</td>
<td>1881777</td>
<td>1</td>
<td>2</td>
<td>371471</td>
<td>755153</td>
<td>3888</td>
<td>65867</td>
<td>18880.78</td>
</tr>
<tr>
<td>EMI</td>
<td>1983058</td>
<td>3</td>
<td>5</td>
<td>391986</td>
<td>795536</td>
<td>2552</td>
<td>68875</td>
<td>23756.09</td>
</tr>
<tr>
<td>EMI</td>
<td>2110421</td>
<td>2</td>
<td>4</td>
<td>416949</td>
<td>846736</td>
<td>2480</td>
<td>73535</td>
<td>21582.90</td>
</tr>
<tr>
<td>EMI</td>
<td>2532813</td>
<td>0</td>
<td>1</td>
<td>501235</td>
<td>1015789</td>
<td>8376</td>
<td>87535</td>
<td>27676.91</td>
</tr>
<tr>
<td>EMI</td>
<td>2728288</td>
<td>1</td>
<td>3</td>
<td>539518</td>
<td>1094385</td>
<td>3888</td>
<td>94417</td>
<td>45430.64</td>
</tr>
<tr>
<td>EMI</td>
<td>3486058</td>
<td>2</td>
<td>5</td>
<td>689996</td>
<td>1398031</td>
<td>2480</td>
<td>119968</td>
<td>53132.75</td>
</tr>
<tr>
<td>EMI</td>
<td>3810512</td>
<td>0</td>
<td>2</td>
<td>755076</td>
<td>1527718</td>
<td>8376</td>
<td>130389</td>
<td>52495.76</td>
</tr>
<tr>
<td>EMI</td>
<td>4802239</td>
<td>1</td>
<td>4</td>
<td>951245</td>
<td>1925497</td>
<td>3888</td>
<td>164359</td>
<td>68474.42</td>
</tr>
<tr>
<td>EMI</td>
<td>5271370</td>
<td>0</td>
<td>3</td>
<td>1045440</td>
<td>2112965</td>
<td>8376</td>
<td>179195</td>
<td>82601.74</td>
</tr>
<tr>
<td>HS</td>
<td>403085</td>
<td>0</td>
<td>3</td>
<td>2299046</td>
<td>4665105</td>
<td>—</td>
<td>403085</td>
<td>3572.91</td>
</tr>
<tr>
<td>PC (map)</td>
<td>403085</td>
<td>0</td>
<td>3</td>
<td>2299046</td>
<td>4665105</td>
<td>—</td>
<td>403085</td>
<td>2015.02</td>
</tr>
<tr>
<td>PC (load)</td>
<td>403085</td>
<td>0</td>
<td>3</td>
<td>2299046</td>
<td>4665105</td>
<td>—</td>
<td>403085</td>
<td>409.91</td>
</tr>
<tr>
<td>PC (run)</td>
<td>403085</td>
<td>0</td>
<td>3</td>
<td>2299046</td>
<td>4665105</td>
<td>—</td>
<td>403085</td>
<td>3.51</td>
</tr>
</tbody>
</table>

### 22 h (system size = 5.271 370). The domain size also strongly affects the mesh size and computation time. For example, for the coarse 2 resolution, with respect to box 1, box 2 is 1.83× slower, box 3 4.16×, box 4 8.33×, box 5 20.51×.

In this article, we have used a detailed modeling framework—the extracellular-membrane-intracellular (EMI) model [6, 20]—to evaluate the effect of placing an extracellular recording device (neural probe) on the measured signals. We used meshes representing a simplified neuron and two different kind of probes: a microwire (a cylindrical probe with diameter of 30 µm) and multi-electrode arrays (MEAs), modeling a Neuropixels probe [36]. We quantified the probe effect by simulating the domain with and without the probe in the extracellular domain and showed that the effect is substantial for the MEA probes (figures 3(B) and (C)), while it is negligible for microwires (figure 3(A)). The amplitude of the largest peak using the MEA probes is almost twice as large (~1.9 times) compared to the case with no probe, and this factor is relatively independent of the probe distance (figure 5(D)), but it is reduced when the neuron and the probe are shifted laterally (figure 6(A)) or when the probe is rotated (figure 6(B)). Moreover, we discussed the effect of varying the mesh resolution and of the size of the computational domain. We also compared our finite element solutions to solutions obtained by solving the conventional cable equation, and found that the latter gave result very similar to the finite element solution when the probe was removed from the extracellular space (figure 7(A)). Therefore, we suggest that the probe effect can be a key element in modeling experimental data obtained with MEA probes. However, clearly further analysis is needed to...
clarify this matter. At present the computational cost of the EMI model prevents simulations of neurons represented using realistic geometries. Thus, in an effort to offer less computationally expensive solutions to include the probe effect in simulations, we investigated various correction methods resulting in more accurate predictions and we proposed the probe correction method, which allows to obtain accurate solutions with reasonable computational cost and resources.

4.1. Comparison with previous work

In this work we used a finite element approach [20] to simulate the dynamics of a simplified neuron and to compute extracellular potentials using the EMI model. The use of FEM modeling for neural simulations has been performed before [19, 29, 30, 47, 48], but mainly as an advanced tool to study neural dynamics and ephaptic effects. In Moffit et al [47], the authors simulated, using the cable equation approach, a neuron at 65 µm from a shank microelectrode with a single recording site, and then used the currents in a finite element implementation of the extracellular domain, including the shank microelectrode. They found that the amplitude of the recorded potential with the shank was 77-100% larger than the analytical solution, but the spike shape was similar to the analytical solution (equation (13)), in accordance with our results (figures 7(A) and (B)). The effects using MEA probes and varying distances, lateral shifts, and probe rotations were not investigated. In Ness et al [25], an analytical framework for in vitro planar MEA using the method of images [24] was developed. A detailed neural model was simulated using the cable equation and transmembrane currents were used as forcing functions for a finite element simulation to validate the analytical solutions. In the in vitro case, in which the MEA is assumed to be an infinite insulating plane, the authors showed that the insulating MEA layer affects the amplitudes of the recorded potentials, effectively increasing it by a maximum factor of 2, which can be analytically predicted by the method of images (MoI).

In this study, we investigated how large the effect of commonly used in vivo probes is using the advanced EMI modeling framework. Our results are in line with these previous findings and we also show that the geometry, in terms of size and alignment of the probe, plays a very important role. We show that large silicon probes can be almost regarded as insulated planes when the neuron is aligned to them (potential increased by factor ∼1.9) for large ranges of distances (figure 5(D)). An interesting effect following the reduction of the amplitude factor with lateral shifts (figure 6(A)) is that neurons not aligned with the probe will be recorded with a lower signal-to-noise ratio (SNR) due to the smaller amplitude increase, assuming that other sources of noise are invariant with respect to the probe location (such as electronic noise and biological noise from far neurons). This might bias neural recordings towards identifying neurons that are closer to the center of the probe, rather than the ones lying at the probes’ sides. However, this conclusion is speculative and might be affected by other factors, such as the distribution of neurons around the probe and their morphology (which contributes to the EAP). Therefore, ground truth information about the position of the recorded neurons and their reconstructed morphologies are needed for a quantitative evaluation of this phenomenon.

4.2. Limitations and extensions

4.2.1. Mesh improvements. The EMI model is, in principle, able to accurately represent the neuron and the neural probe. However, the accuracy of the model comes at the cost of computational resources. In order to be able to run simulations in a reasonable amount of time, the geometry of the neuron needed to be simplified considerably. First, we used a simple neuron in terms of a ball-and-stick with axon. This model is able to describe certain aspects of the neuronal dynamics [35], but it clearly cannot reach a level of detail of some more realistic morphologies, such as the reconstructed models made available by various initiatives [1–5]. We quantified the amplitude shift due to the probe in the extracellular domain (∼1.9 on average for the MEA probes when neuron and probe are aligned), but this factor most likely also depend on the specific cell morphology that we used, and not only on the probe design and geometry. Therefore, we aim at extending the framework [49] for generating finite element meshes from publicly available realistic morphologies [5], allowing us to explore the probe effect for more complex morphologies.

Furthermore, we assumed ideal recording sites with an infinite input impedance which does not allow any current to flow in. In reality, recording electrodes have a high, but not infinite impedance that could be modeled by considering electrodes as an additional domain with very low conductance, even if it has been shown that for normal electrodes’ impedance the effect of conductive and equipotential recording sites is negligible [32].

4.2.2. Computational costs. In section 3.4 we showed that the EMI model is much more computationally demanding than conventional modeling using cable and volume conduction theory. For the simplest simulation performed in this study (course 3 and box size 1), a system with 337 515 unknowns was solved in about 40 min. The Neuron simulations described in section 2.4 took ∼0.59 s to run, about 2400 times faster than the simplest EMI simulation performed here. However, because of our implementation and solution strategy for FEM, this factor should be considered as a rather pessimistic upper bound. In particular, the employed version of FEnICS (2017.2.0) does not allow for finite element spaces with upper bound. In particular, the employed version of FEnICS (2017.2.0) does not allow for finite element spaces with
are naturally higher. The number of unphysical unknowns can be seen in table 5 as a difference between total number of facets in the mesh and the number of facets on the surface of the neuron. For example, in the largest system considered here, avoiding the unphysical unknowns would reduce the system size by about 2 million.

In addition to assembling the linear system with only the physical unknowns, a potential speed up could be achieved by employing iterative solvers with suitable preconditioners. That is, fast PDE solvers for diffusion equations typically use around 1s per million degrees of freedom. As we here employ a H(div) formulation, we expect the solution to be computed in around 5 s per million degrees with multilevel methods. As shown in table 5, 500 timesteps of solving systems with around one million degrees of freedom takes 82,600 s, which means 165 s per time steps. Hence, we may expect to speed up the solving procedure by around a factor 30 with better solvers. If further speed-up is required then finite element based reduced basis function method provides an attractive approach that should be addressed in future research.

4.2.3. Finite element methods are not alternatives to the conventional cable equation. The EMI framework, due to its computational requirements, is presently not an alternative to conventional modeling involving the cable equation (equation (12)) and the current summation formula (equation (13)). However, for specific applications, it can provide interesting insights. The hybrid solution combines the cable equation solution to finite element modeling, in practice solving the FEM problem only for the extracellular space and using the transmembrane currents computed by the cable equation as forcing functions [21–23, 25, 47]. However, the HS is also computationally expensive and it increases in complexity with longer simulation durations. Similar considerations can be made if Boundary Element Methods (BEM) [50] are employed instead of FEM ones, even though they are less computationally intense than the current FEM formulation. One possible drawback of BEM solvers is that they could not accommodate for anisotropic conductivity, while FEM solvers could in principle solve meshes with non-homogeneous conductivity between surfaces [51].

Another much faster option could be using approaches based on constant scaling, such as MoI and SCS. However, even correcting with a right factor smaller than 2, the these methods cannot account for the variability of peak ratios among the electrodes (figure 8(B)). Therefore, we suggested here the probe correction (PC) method, which combines a one-time finite element simulation to model how each electrode of a specific probe affects the extracellular domain, and then uses the reciprocity principle to compute the potential on the recording sites arising from transmembrane currents. We showed that this method is able to reach the HS accuracy at a much smaller computational time (table 5), which is also not strongly dependent on the simulation duration. Moreover, the time required to compute the probe specific mapping (PC (map)) and loading the FEM solutions in memory (PC (load)) could be further reduced by decreasing the mesh resolution. This possibility should be further investigated with a convergence analysis, similar to section 3.2 for the EMI model.

4.3. Significance of the probe effect

The effect of the recording device has not been fully taken into consideration in mathematical models of the extracellular field surrounding neurons. The probe effect needs to be considered when modeling silicon MEA, whose sizes are significantly larger than the recorded neurons. The assumption of an infinite and homogeneous medium is in fact largely violated when such bulky probes are in the extracellular space in the proximity of the cells. Although the tissue can be regarded as purely conductive and with a constant conductivity [52], these probes represent clear discontinuities in the extracellular conductivity, which strongly affect the measured potential due to their insulating properties. While the probe effect is large for MEAs, we found that it was negligible for microwire-type of probes, mainly for two reasons: first, the microwire is thinner and overall smaller than the MEA; second, the electric potential is sampled at the tip of the probe and in the entire semi-space below the microwire currents are free to flow without any obstacle.

When dealing with silicon MEAs, though, this effect could be crucial for certain applications that require to realistically describe recordings. For example, Gold et al [26] used, in simulation, extracellular action potentials (EAP) to constrain conductances of neuronal models. Clearly, neglecting the probe effect would result in an incorrect parameterization of the models in this case.

Another example in which including this effect could be beneficial is when EAP are used to localize the somata position with respect to the probe. This is traditionally done by solving the inverse problem: a simple model, such as a monopolar current source [53–55], a dipolar-current source [53, 56, 57], line-source models [58, 59], or a ball-and-stick model [60], is moved around the extracellular space to minimize the error between the recorded potential and the one generated by the model. Ignoring the probe might result in larger localization errors.

Recently, we used simulated EAP on MEA as ground truth data, from which features were extracted to train machine-learning methods to localize neurons [27, 28] and recognize their cell type from EAPs [28]. When training such machine-learning models on simulated data and applying them to experimental data, neglecting the probe effect could confound the trained model and yield prediction errors.

Moreover, explaining experimental recordings on MEA without considering the probe might cause discrepancies between the modeling and experimental results hard to reconcile. On the other hand, in order to fully explain and validate our findings, an experiment with accurate co-location of extracellular recordings and cell position (and ideally morphology) is required. For example, an experimental setup in which a planar MEA is combined with two-photon calcium imaging [61] could provide an accurate estimate of the relative position between the neurons and the MEA.
In conclusion, we presented numerical evidence that suggests that the probe effect, especially when using multi-electrode silicon probes, affects the way we model extracellular neural activity and interpret experimental data and cannot be neglected for specific applications.

Acknowledgments

APB and KJH are part of the Simula-UCSD-University of Oslo Research and PhD training (SUURPh) program, an international collaboration in computational biology and medicine funded by the Norwegian Ministry of Education and Research. TVN would like to acknowledge the European Union Horizon 2020 Research and Innovation Programme under Grant Agreement No. 720270 and 785907 (Human Brain Project (HBP) SGA1 and SGA2). PB would like to acknowledge the Defense Advanced Research Project Agency (DARPA), Contract No. N66001-17-C-4002. The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policy of the Department of Defense of the U S Government.

ORCID IDs

Alessio Paolo Buccino https://orcid.org/0000-0003-3661-527X
Miroslav Kuchta https://orcid.org/0000-0002-3832-0988
Karoline Horgmo Jæger https://orcid.org/0000-0003-4234-9094
Torbjørn Vefferstad Ness https://orcid.org/0000-0001-9080-8502
Pierre Berthet https://orcid.org/0000-0002-6878-6842
Kent-Andre Mardal https://orcid.org/0000-0002-4946-1110
Gert Cauwenberghs https://orcid.org/0000-0002-3166-5529
Aslak Tveito https://orcid.org/0000-0003-2362-146X

References

[10] Hines M L, Davison A P and Muller E 2009 NEURON and Python Front. Neuroinform. 3 1
Appendices
Appendix A

Other projects

Open source modules for tracking animal behavior and closed-loop stimulation based on Open Ephys and Bonsai

In this journal contribution to the Journal of Neural Engineering, we developed a system for tracking and closed-loop stimulation of rodents in open-field experiments. The system uses open-source systems for image analysis, Bonsai and electrophysiology, Open Ephys. We contributed featured plugins for tracking the animal position in real-time and setting up position-dependent closed-loop stimulations. We showed the reliability of the system, its capability of recording place and grid cells, and a sample use case to optogenetically stimulate a grid-cell within its grid field. More recently, we have been using the system also to selectively rewire place cells by stimulating them electrically outside of their place field.

I am the first and corresponding author for this paper, contributing in designing and conceiving the project, implementing the software, running tests and experiments, and writing most of the paper.

Scalable Spike Source Localization in Extracellular Recordings using Amortized Variational Inference

In this conference paper, recently accepted to NeurIPS 2019, we presented a spike localization method for multi-electrode array recordings based on a variational autoencoder (VAE). The main idea is to use a deep network encoder and a model-based decoder to predict the 3D location of spike waveforms. We showed that the use of a VAE improves the performance of spike localization both in terms of accuracy with respect to center-of-mass estimates and it is much more computationally efficient than Markov Chain Monte Carlo (MCMC) approaches. The results were obtained both for a dense square MEA simulated with MEArec and for an experimental Neuropixels dataset. A precise localization of individual spike waveforms could be beneficial for spike sorting.
A. Other projects

methods that use this feature for clustering\cite{52,53}.

My contribution to this article has been mainly in simulating the extracellular recordings, participating in discussions about the methods and results, and revising and approving the final manuscript.