Title

RT-Sort: An action potential propagation-based algorithm for real time spike detection and sorting with millisecond latencies.

Journal
PLoS ONE, 19(12)

Authors
van der Molen, Tjitse
Lim, Max
Bartram, Julian
et al.

Publication Date
2024

DOI
10.1371/journal.pone.0312438

Peer reviewed

PLOS ONE

RESEARCH ARTICLE

RT-Sort: An action potential propagation-based algorithm for real time spike detection and sorting with millisecond latencies

Tjitse van der Molen[1,2]*, Max Lim[1,2], Julian Bartram[3], Zhuowei Cheng[4], Ash Robbins[5], David F. Parks[6], Linda R. Petzold[4], Andreas Hierlemann[3], David Haussler[6,7,8], Paul K. Hansma[1,9], Kenneth R. Tovar[1], Kenneth S. Kosik[1,2]

1 Neuroscience Research Institute, University of California Santa Barbara, Santa Barbara, California, United States of America, **2** Department of Molecular, Cellular and Developmental Biology, University of California Santa Barbara, Santa Barbara, California, United States of America, **3** Department of Biosystems Science and Engineering, ETH Zürich, Basel, Switzerland, **4** Department of Computer Science, University of California Santa Barbara, Santa Barbara, California, United States of America, **5** Department of Electrical and Computer Engineering, University of California Santa Cruz, Santa Cruz, California, United States of America, **6** Department of Biomolecular Engineering, University of California Santa Cruz, Santa Cruz, California, United States of America, **7** Howard Hughes Medical Institute, University of California Santa Cruz, Santa Cruz, California, United States of America, **8** UC Santa Cruz Genomics Institute, University of California Santa Cruz, Santa Cruz, California, United States of America, **9** Department of Physics, University of California Santa Barbara, Santa Barbara, California, United States of America

* tjitse@ucsb.edu

OPEN ACCESS

Citation: van der Molen T, Lim M, Bartram J, Cheng Z, Robbins A, Parks DF, et al. (2024) RT-Sort: An action potential propagation-based algorithm for real time spike detection and sorting with millisecond latencies. PLoS ONE 19(12): e0312438. https://doi.org/10.1371/journal.pone.0312438

Editor: Nicholas V. Swindale, University of British Columbia, CANADA

Received: June 11, 2024

Accepted: October 7, 2024

Published: December 5, 2024

Data Availability Statement: All data used in this manuscript come from publicly available source and can be found here https://datadryad.org/stash/dataset/doi:10.25349/D9031Z and here https://buzsakilab.nyumc.org/datasets/SiegleJ/AllenInstitute_744912849/session_766640955/. All the code used for this manuscript is publicly available in the following Github repository: https://github.com/KosikLabUCSB/RT-Sort/tree/main. A more user friendly library that provides access to the full spike sorting algorithm and related

Abstract

With the use of high-density multi-electrode recording devices, electrophysiological signals resulting from action potentials of individual neurons can now be reliably detected on multiple adjacent recording electrodes. Spike sorting assigns these signals to putative neural sources. However, until now, spike sorting can only be performed after completion of the recording, preventing true real time usage of spike sorting algorithms. Utilizing the unique propagation patterns of action potentials along axons detected as high-fidelity sequential activations on adjacent electrodes, together with a convolutional neural network-based spike detection algorithm, we introduce RT-Sort (Real Time Sorting), a spike sorting algorithm that enables the sorted detection of action potentials within 7.5ms±1.5ms (mean ±STD) after the waveform trough while the recording remains ongoing. RT-Sort's true real-time spike sorting capabilities enable closed loop experiments with latencies comparable to synaptic delay times. We show RT-Sort's performance on both Multi-Electrode Arrays as well as Neuropixels probes to exemplify RT-Sort's functionality on different types of recording hardware and electrode configurations.

Introduction

Over the past decade, significant improvements in extracellular neural recording techniques have resulted in the development of dense multi electrode recording devices for both *in vivo* [1, 2] and *in vitro* [3] recordings. These advances have significantly enhanced our

functionalities can be accessed here: github.com/braingeneers/braindance.

Funding: The author(s) received no specific funding for this work.

Competing interests: The authors have declared that no competing interests exist.

understanding of neural circuits and their activity. With interelectrode pitches smaller than 20μm, these devices allow for the recording of the activity of a single neuron by multiple adjacent electrodes. Still, in contrast to intracellular single cell recordings, individual extracellular electrodes regularly detect the activity of multiple adjacent neurons. To address this problem, spike sorting algorithms have been developed for the assignment of detected spikes to specific units that represent putative neurons [4–9].

However, current conventional spike sorting algorithms are far from ideal [10, 11]. The sorting algorithms are usually based on spike waveform template matching. This means that the detected spikes are clustered into groups with similar waveform shapes, which then determines the units to which these spikes are assigned. This method does not account for time variant biological events or experimentally induced phenomena that alter the shape of the waveform. For example, extensive preceding activity of a bursty neuron can significantly change the shape of the waveform [12–14]. This is especially problematic when studying neural activity that occurs in occasional barrages like hippocampal sharp wave ripples or population bursts of *in vitro* cultures. Furthermore, pharmacological treatments affecting the ion channel dynamics and/or differences in temperature can lead to large differences in the detected waveform shape as well [11, 12]. As a result, spikes from a single neuron can often be assigned to multiple differently detected units or even completely missed. This causes the most commonly used spike sorting algorithms to be in stark disagreement with each other [15]. Notably, Buccino et al. showed that when six different, regularly used, spike sorting algorithms [4–9] were applied to the same Neuropixels recording [16], out of the total 1356 uniquely detected units, all sorting algorithms were in agreement with the detected spike times for only 33 units while 1093 units were detected by only one of any of the sorting algorithms.

Furthermore, when two neurons closely located to the same electrode fire an action potential at the same time, it will lead to an overlap of the waveforms in the recorded signal. This overlapping signal will not match any of the templates and will therefore often be missed by the sorter [17]. Finally, template matching based algorithms and other automated post-hoc clustering-based methods [18, 19] require all spikes in the recording to be pooled together before performing the clustering. As a result, spike sorting can only be performed after a recording is finished, leaving it impossible to know to which neuron a detected spike belongs while an experiment is ongoing.

This issue will impact anyone interested in performing closed loop experiments in which a neural system receives external inputs based on preceding activity recorded from the same neurons. Such closed loop experiments have been performed in multiple forms, ranging from controlling the bursting activity of *in vitro* cultures [20] to embedding a neural culture in a virtual environment [21, 22] or letting a neural culture control a robot in the real world while providing information about the robot's environment [23]. Similar issues are faced when reading out signals of brain machine interfaces in real time [24]. A more in depth review of various types of closed loop experiments and the difficulties with real time spike sorting can be found in [25] and [26] respectively.

As a result, various groups have attempted to develop algorithms that can sort detected action potentials within tens of milliseconds after their occurrence, in the range of signal transmission delay times of single synapses [27]. However, so far no algorithm has yielded good results. For example, Franke et al. suggested a method akin to template matching, using linear filters and Independent Component Analysis (ICA) for the real time detection and sorting of action potentials [28]. However, in subsequent work co-authored by Franke, it was noted that: "The analysis of extracellular neuronal signals, recorded at high spatiotemporal resolution, reveals that the recorded data cannot be modeled as a purely linear mixture. As a consequence, ICA fails to separate completely the neuronal signals and cannot be used as a stand-alone method for spike sorting in HD-MEA recordings.", concluding that post processing techniques have to be used to overcome the most severe limitations of ICA [29]. This makes the ICA based method useful as a preprocessing step to spike sorting but defeats its purpose as a real time spike sorter.

More recently, machine learning based solutions have been proposed. For example, Li et al. utilized a convolution neural network (CNN) trained with manually labeled spikes to sort spikes detected on a single electrode [30]. They claimed that their method could classify spikes in one millisecond; however, the work did not include results of the actual latency between the occurrence of the spike and their classification which creates uncertainty about the actual sorting latencies in real time. More importantly, this method entailed training a model for each individual electrode, requiring manually labeling spikes for each electrode before being able to run a real time experiment. In addition, Li et al. reported on lower performances when only several hundred manually labeled spikes were included for the training and also noted that a spike waveform may be recorded by multiple electrodes at the same time, which still requires a solution for distinguishing spikes that are simultaneously recorded by different electrodes. Therefore, it can be concluded that although highlighting the potential of CNN's for real time spike sorting, the algorithm developed by Li et al. cannot readily be applied to experiments using conventional recording devices containing hundreds of closely located electrodes [1, 3].

Also utilizing CNNs, Racz et al. took advantage of the fact that single spikes can be detected on multiple electrodes [31]. They trained their model using data from a silicon probe, resulting in a data matrix representing the signal of 32x4 equally distanced electrodes, to better distinguish waveforms from closely located but distinct neurons. They applied Kilosort, an offline, template matching based spike sorting algorithm [32] to a pre-recording made by the probe to obtain a "ground truth" for training their model. As a result, a new model would have to be trained every time a new probe was inserted. Furthermore, by optimizing the models to an electrode configuration of 32x4 electrodes, this algorithm doesn't transfer well to other related recording devices with variable electrode configurations [1, 3]. Most importantly, the authors noted that the model was directly trained on the outputs of Kilosort and so any mistakes resulting from the previously listed shortcomings of template matching based sorting algorithms are necessarily inherited by this algorithm.

We set out to develop RT-Sort (Real Time Sorting), a spike detection and sorting algorithm that overcomes the issues related to overlapping spikes, changes in waveform shapes and the inability to perform spike sorting in real time. Instead of relying on waveform shape similarity, we based our sorting method on a more biologically plausible phenomenon. Namely, the fact that an action potential propagates along a neural axon and thus gets detected with very high fidelity by a sequence of electrodes, in the same order, with sub millisecond but nonzero time delays [33]. Interestingly, all real time spike sorting methods described above still apply some form of template matching, where the shape of the spike waveform recorded on one or multiple electrodes directly determines the unit cluster to which the spike is assigned. For the first time, we used action potential propagation detection instead of spike waveform template matching to identify individual neurons in our recordings. This idea has already been acknowledged and utilized in the classification of neuron types based on high density recordings [34] but has not yet explicitly been applied to enhance spike sorting routines.

In this work, we present RT-Sort, an algorithm that can sort spikes in real time based on their action potential propagation trajectory. In doing so, we show that RT-Sort performs at the level of a broad series of conventionally used template matching based spike sorting algorithms [4–9], while reporting latencies for detection and sorting of 7.5ms±1.5ms (mean±STD) from the trough of the waveform on replayed recordings. Furthermore, RT-Sort can be applied to any system with sufficiently high electrode density (≤50μm pitch) as exemplified by our implementations on some of the most commonly used recording devices for both *in vivo* [1] and *in vitro* [3] experiments.

Because our method relies on the sequential detection of a rapidly decaying propagating action potential on multiple electrodes, our spike detection sensitivity is an important limiting factor as to whether we can sort a spike. Relying on conventional RMS based threshold crossings does not provide the desired signal to noise ratio. Therefore, similar to Racz et al. [31], we utilize the power of CNNs for the detection of spikes. However, we limit our model to detecting waveform shapes in noisy signals of single electrodes which means that our model does not require training on individual recording or electrode configurations. Furthermore, thanks to our methods of artificial data amplification, inspired by the work from Parks et al. [35], we are able to generate very large training samples, which enables the development of a more powerful and generalizable model. As a result, a single spike detection model trained for a specific model type–recording device combination can be applied to any future experiment on that model with the specific recording device but a different electrode configuration. Only the action potential propagation sequences will need to be detected using a pre-recording, which is based completely on the detections from the spike detection model and does not require sorting the recording using an external spike sorting algorithm. This feature minimizes the duration of the pre-recording and its processing time and reduces susceptibility to errors that could be inherited from the implementation of other spike sorting algorithms. We have trained readily available spike detection models for *in vitro* human brain organoid neurons recorded with MaxWell Biosystems HD-MEAs and *in vivo* mouse neurons recorded with Neuropixels. Furthermore, the open-source Python library that allows users to apply RT-Sort (github.com/braingeneers/braindance) also contains the functionality to train specific detection models to match user needs with only a single line of code.

Results

Real time spike detection on injected spikes

A CNN with 4 layers (Fig 1A) was trained to detect spikes in recording device specific noise. Separate models were trained for MaxWell MEA recordings performed on human brain organoids and Neuropixels 1.0 recordings performed on mice (S1 Fig for Neuropixels). The training was performed using averaged waveform shapes of electrodes selected along the whole recorded axon, the trajectory along which the action potential propagates (Fig 1B). These averaged waveform shapes were then pasted into recording device specific noise fragments which enabled artificial amplification of the training dataset (Fig 1C) and the generation of training cases of overlapping waveform shapes with ground truth (Fig 1D). The small differences between the training loss and testing loss indicated that our models are not overfitting nor underfitting the data (Fig 1E). See the corresponding sections in Methods for a more detailed description of the model training and evaluation.

When testing the trained models on the artificially generated data from the held-out organoid recordings, using the detection threshold yielding an optimal F1 score, we obtained a recall score of 89.1%±5.59% and a precision score of 91.5%±3.23% (mean±STD, Fig 1F, 1G). Similarly, the mouse Neuropixels recordings yielded a recall score of 84.7%±5.81% and a precision score of 86.4%±7.90% (mean±STD, S1F, S1G Fig). In both cases, the obtained recall was significantly larger than applying a 5RMS threshold to the injected spikes (organoids: 5RMS recall mean±STD = 63.6%±7.36%, two-sided paired t-test P = 1.25*10–7, n = 6, Fig 1F; mouse: 5RMS recall mean±STD = 65.1%±3.37%, two-sided paired t-test P = 5.19*10–5, n = 6, S1F Fig). Meanwhile, the precision was not significantly different between the CNN detection model and the 5RMS threshold (organoids: 5RMS precision mean±STD = 95.0%±3.36%, two-

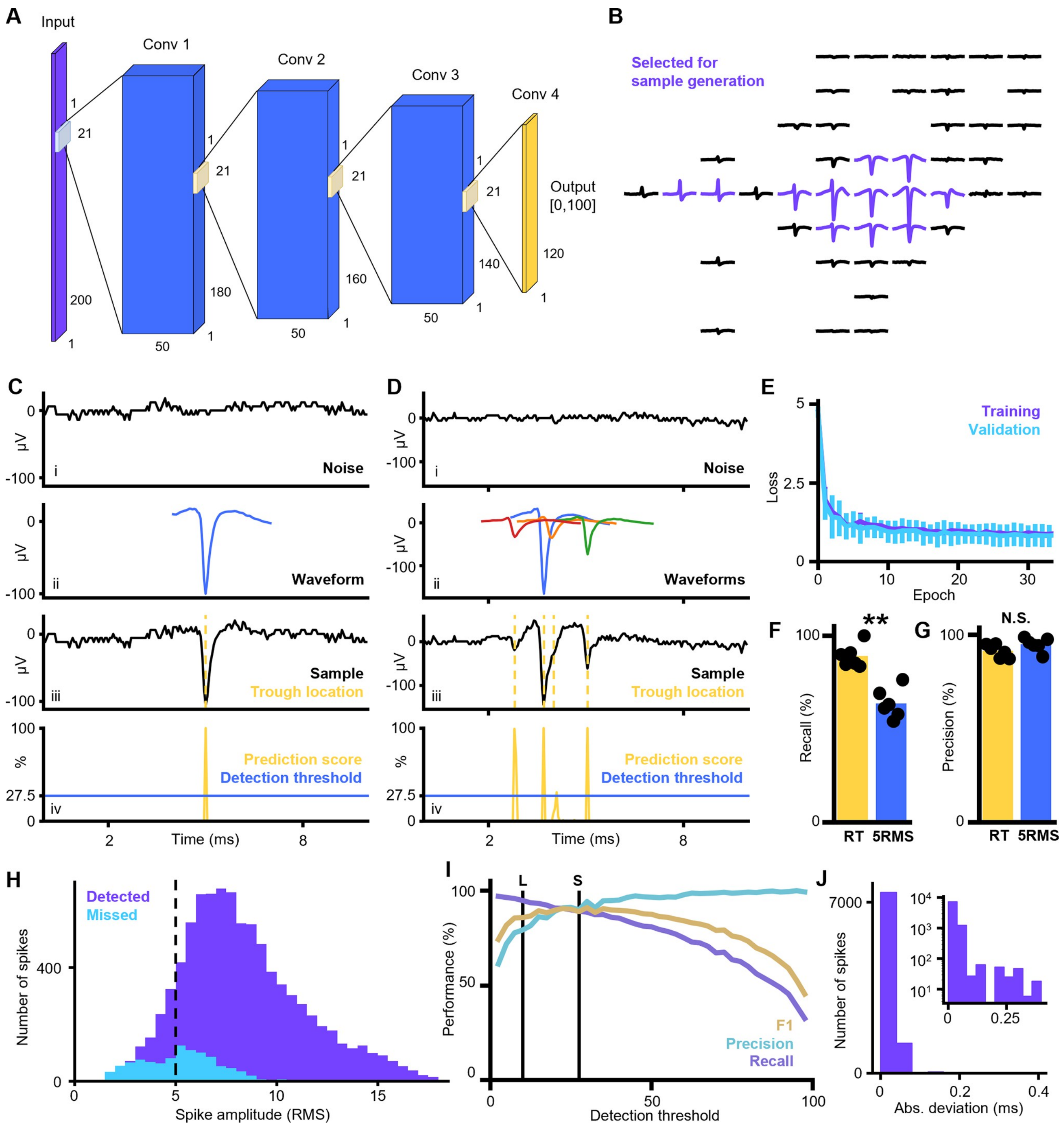

Fig 1. Action potential detection with high temporal accuracy using a convolutional neural network (CNN). A) Architecture of the CNN used for spike detection in 20kHz MEA recordings. The input layer consists of 200 nodes, corresponding to a 10ms window. The output layer consists of 120 nodes corresponding to 2-8ms in the input window. Each output node provides a score between 0 and 100 indicating the likelihood that the corresponding input frame contains a waveform trough. The 4 convolutional layers have a kernel size of 21 frames and a stride of 1 frame so that each output node makes a prediction based on the signal in 2ms before until 2ms after the corresponding frame in the input. **B)** Example of averaged waveform footprint detected by Kilosort2. The purple traces were selected for training/validating the CNN. Note the variety in the selected waveform shapes from this single unit. This ensures that the detection model can detect the propagating action potential along different parts of the neuron. **C)** Example of training/validating sample creation and model prediction. i: A 10ms sample of recording specific noise is taken. ii: A waveform shape is selected from the training or validating pool. iii: The waveform is pasted into the recording device specific noise with ground truth certainty about the trough location (marked with yellow dotted line). iv: CNN detection model predictions for the frames in 2-8ms of the input window show a narrow detection peak at the waveform trough. All figures share the same x-axis. **D)** Same as C but with multiple overlapping waveforms in the same sample. **E)** Training (purple) and validating (cyan) loss as a function

of training epoch. Error bars indicate the STD over the different cross-validation folds. The small differences between the training loss and validating loss indicate that the models is not overfitting nor underfitting the data. **F)** Recall when validating the detection model on samples generated from the held-out recording and when applying a 5RMS threshold to the same samples. The markers indicate the results for each of the 6 held-out datasets and the bar reflects the mean over all held-out datasets. Mean±STD for CNN = 89.1%±5.59% and for 5RMS = 63.6%±7.36%. The detection model has a significantly higher recall (P = 1.25*10^{-7}, two-sided paired t-test, n = 6). **G)** Precision when validating the detection model on samples generated from the held-out recording and when applying a 5RMS threshold to the same samples. The markers indicate the results for each of the 6 held-out datasets and the bar reflects the mean over all held-out datasets. Mean±STD for CNN = 91.5%±3.23% and for 5RMS = 95.0%±3.36%. The difference in precision is not significant (P = 0.13, two-sided paired t-test, n = 6). **H)** Amplitude distribution of detected (purple) and missed (cyan) spikes by the detection model shows a good detection performance for spikes below 5RMS. Amplitudes are expressed as RMS relative to the surrounding 50ms of signal. **I)** F1 score, precision and recall of the detection model as a function of the detection threshold. The loose and stringent detection thresholds are marked with L and S respectively. **J)** The absolute deviation between the model detections and the ground truth trough times shows a high temporal accuracy with on average a deviation of 13μs (0.26 frames). Inset contains the same distribution on a log scale.

https://doi.org/10.1371/journal.pone.0312438.g001

sided paired t-test P = 0.13, n = 6, Fig 1F; mouse: mean±STD = 79.7%±4.60% (smaller than RT-Sort CNN model), two-sided paired t-test P = 0.15, n = 6, S1G Fig). The significantly higher recall scores were due to the fact that the model could detect spikes with amplitudes below 5RMS, with the 5th percentile of the amplitude distribution over all detected waveforms at 4.08RMS for the MEA model and 3.67RMS for the Neuropixels model. Meanwhile, the models hardly missed spikes with amplitudes above 5RMS (Fig 1H, S1H Fig).

Adjusting the detection threshold for the organoid model from its optimal value allowed for enhanced precision scores at the cost of a slightly lower recall score. This enabled the use of both a stringent detection score at the optimal F1 score, obtained with a detection threshold of 27.5%, as well as a loose detection threshold at 10% (Fig 1I, 17.5% and 7.5% for the Neuropixels model, S1I Fig). Even though we considered a detection to be a true positive if it was detected within 400μs of the actual waveform trough, the average absolute deviation from the correct trough time was only 13μs and 14μs for the organoid and mouse recordings respectively (Fig 1J, S1J Fig), reflecting the high temporal sensitivity of the model detections. When probing the model to update an injected piece of noise signal until it reaches an optimal detection threshold value, a classical waveform shape emerges (S2 Fig) which further highlights that the model indeed is sensitive to waveform shapes in noisy signals.

Real time spike detection on replayed recordings

We subsequently tested our model on actual spikes detected by Kilosort2 in the held-out recordings. When only considering the largest waveform amplitude electrodes, we observed precision and recall scores of 26.0%±29.4% and 83.1%±23.4% for the MEA model applied to organoid recordings and 7.47%±12.7% and 77.6%±25.8% for the Neuropixels model applied to intact mouse brain recordings (mean±STD, S3A, S3B Fig). When interpreting these precision scores, it has to be considered that we are comparing the spike detection model against sorted Kilosort2 detections. A sorted detection only gets detected on a single electrode while our spike detection model can detect a spike on multiple electrodes. As such, detections from nearby units might wrongfully be considered false positive spikes leading to low precision scores. In addition, Kilosort2 is no ground truth and might miss spikes or falsely detect spikes due to various possible reasons like overlapping waveforms, noise, low amplitude or waveform shape changes. This all will negatively impact the obtained performance scores.

When we considered the 4th largest waveform amplitude electrodes per unit, the precision and recall were 10.6%±20.6% and 25.3%±24.5% for the MEA model applied to organoid recordings and 5.61%±10.8% and 53.1%±33.0% for the Neuropixels model applied to intact mouse brain recordings (mean±STD, S3A, S3B Fig). The reduced recall reflects the fact that some of the detected units will not be detectable on 4 individual recording electrodes, even though this is a requirement for spikes to be sorted by RT-Sort. However, the bimodality of the distribution of recall scores per unit indicates that there is a subset of units that can reliably get detected (S3C, S3D Fig) on at least 4 individual electrodes which makes them strong candidates for detection by RT-Sort.

Real time spike sorting on ground truth datasets

Subsequently, we utilized the CNN outputs to perform spike sorting in real time using two different ground truth datasets. A detailed explanation of the spike sorting algorithm can be found in Methods: Propagation sequence detection and following sections. In summary, within a short pre-recording made right before the experiment, recurring sequential detections on multiple electrodes with short but non-zero temporal interelectrode intervals were detected, reflecting the propagation of an action potential. We refer to these detections as propagation signals. Detections were based on high fidelity interelectrode intervals and compared to the highest amplitude electrode in the sequence. Sequences that were highly similar in their propagation sequence were merged into single propagation signals. This then yields the final selection of propagation signal sequences that were detected. In the subsequent experiment, sequential detections were compared to the pre-detected propagation signal sequences and if there was sufficient overlap, the sequential detection was counted as a spike for the propagation signal sequence.

The first ground truth dataset consisted of recordings that were made of rat primary neurons plated on MaxWell HD-MEAs while one of the neurons was simultaneously recorded by patch clamp (Methods: Patch-MEA ground truth recording). Applying RT-Sort to this recording yielded the detection of 49 different units, including the neuron that was recorded using patch clamp (Fig 2A–2C).

Because our CNN model was trained to detect general waveform shapes in sample specific noise which are then sorted based on their action potential propagation patterns, RT-Sort is well capable of detecting all spikes from the same neuron despite changes in waveform shape (Fig 2D, 2E, S4 Fig for additional MEA example and Neuropixels example). Furthermore, since our CNN was trained on multiple waveforms injected into the same detection window, we find that RT-Sort is very suited to detect overlapping waveforms in both the injected (Fig 1D) and actual recordings (Fig 2F, S5 Fig for Neuropixels examples and additional MEA examples). Finally, with the enhanced sensitivity of our spike detections, RT-Sort can detect units that are fully below 5RMS (S6 Fig). Doing the ground truth tests on n = 4 neurons from 2 different MEAs using RT-Sort in offline mode, resulted in a precision of 97.5%±4.4% and a recall of 90.9%±5.57% (mean±STD, Fig 2G). For the array with a pre-recording, RT-Sort was also applied in online mode on the n = 2 neurons on this array, yielding a precision of 98.2% ±1.78% and a recall of 97.8%±1.60% (mean±STD, Fig 2H).

We further tested RT-Sort on a different ground truth dataset consisting of a simulated mouse *in vivo* Neuropixels dataset used in [15]. This simulated dataset consisted of 250 biophysically detailed neurons that each exhibit independent Poisson firing of spikes with added Gaussian noise. To first test the ability of RT-Sort to assign spikes to the correct sequences, sequence metrics were initially obtained using the ground truth spike times from the modeled recording. Subsequently assigning spikes to these sequences yielded 142 units that were detected on the minimally required 4 electrodes to measure the propagating action potential. Over these 142 units, a precision of 98.8%±12.4% (mean±STD) was obtained, reflecting a low false positive detection rate on spike detection (Fig 2I). Subsequently, RT-Sort was run in offline mode on the first 5 minutes of the simulated dataset to detect propagation sequences, yielding 120 detectable action potential propagation sequences. Over these 120 units, a

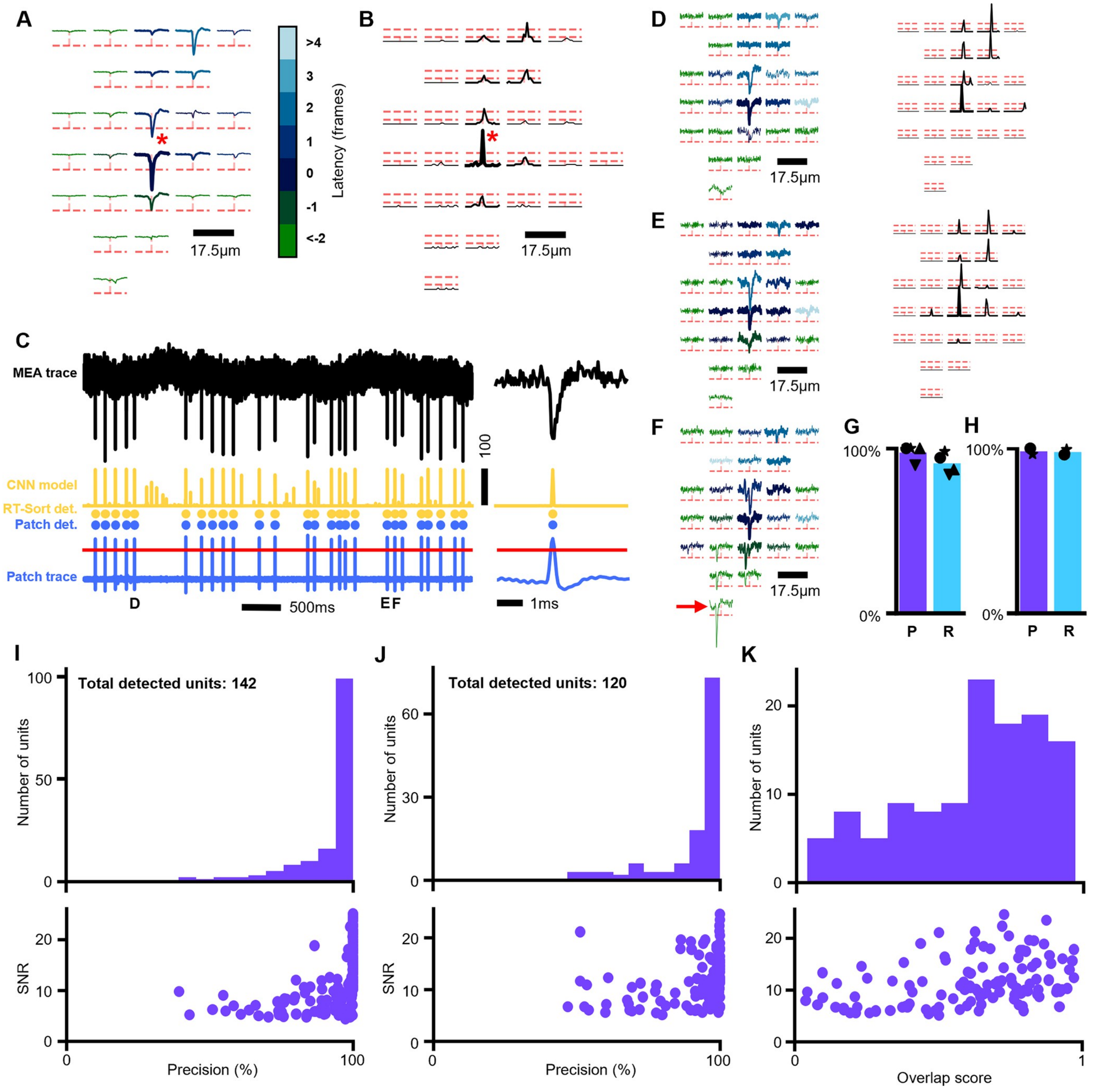

Fig 2. RT-Sort spike sorting performance on ground truth datasets. A) Averaged waveform footprint from a neuron recorded using a high-density multi-electrode array (MEA). Each trace represents the signal from a single electrode averaged over all action potential detections of the recorded neuron. 5 times the signal to noise ratio of the signal measured on each electrode is indicated with dotted red lines. The root electrode detected by RT-Sort and used for the spike triggered averaging is marked with the red star. The trace on each electrode ranges from 2ms before until 2ms after the waveform trough on the root electrode. All detected loose electrodes by RT-Sort are marked in bold. For each electrode, the color of the trace represents the average detection interval relative to the root electrode. **B)** The average spike detection model scores over all detected action potentials for the same neuron and electrodes as A. The loose and stringent detection threshold on each electrode is indicated with dotted red lines. All detected loose electrodes by RT-Sort are marked in bold. To better reflect the high temporal precision of the detection model, the time range for each trace is magnified 4 times relative to the traces in A, resulting in the detection score trace ranging from 0.5ms before until 0.5ms after the waveform trough on the root electrode (marked with red star). **C)** From top to bottom: Raw unfiltered MEA trace of the root electrode marked in A and B for part of the recording (black). This signal is used as input for the detection model. Detection model scores for the same period (yellow). RT-Sort spike sorted detections (yellow dots). Spike detections in simultaneously made patch recording of the same neuron (blue dots). Patch trace recorded in cell-attach mode (blue line) with spike detection threshold marked in red. Note that the detection model sometimes detects spikes from an adjacent neuron (see panel F) although RT-Sort correctly assigns these detections to a different neuron. Right: zoomed in signals for a single spike marked with D. **D)** Same as A and B but for a single spike from the same neuron, detected by RT-Sort in online mode.

The spike is marked with D in panel C. The colors for each trace correspond to the same latencies as A. **E)** Another example of a single spike footprint from the same neuron, marked with E in panel C. Note the difference in the waveform shape compared to D on the electrode above the root electrode. **F)** Single spike footprint from the same neuron that overlaps with a spike from a different neuron (root electrode marked with red arrow). Despite the waveform overlap, the spike is still correctly detected as marked in panel C. **G)** Precision and recall for spikes detected by RT-Sort from 4 different neurons on 2 different MEAs with simultaneous patch-clamp ground truth recording. Precision = 97.5%±4.4%, Recall = 90.9%±5.57% (mean±STD). The example neuron in A is marked with the circle. **H)** Precision and recall for spikes detected by RT-Sort from 2 different neurons on an MEA with simultaneous patch-clamp ground truth recording. The detections were made using sequences detected in a pre-recording, similar to how RT-Sort would be applied in real time. Precision = 98.2%±1.78%, Recall = 97.8%±1.60% (mean±STD). **I)** Top: for each unit detected in the simulated ground truth recording using sequence metrics generated based on the ground truth spike locations, the precision over all detected spikes compared to the most similar ground truth neuron. Mean ±STD = 98.8%±12.4%. Bottom: for each unit shown in the histogram at the top of the panel, the corresponding average waveform amplitude expressed in SNR. Top and bottom share the same x-axis. **J)** Top: for each unit detected in the simulated ground truth recording, the precision over all detected spikes compared to the most similar ground truth neuron. Mean±STD = 97.3%±13.4%. Bottom: for each unit shown in the histogram at the top of the panel, the corresponding average waveform amplitude expressed in SNR. Top and bottom share the same x-axis. **K)** Top: for each unit detected in the simulated ground truth recording, the overlap score for the most similar ground truth neuron. Mean±STD = 0.651±0.242. Bottom: for each unit shown in the histogram at the top of the panel, the corresponding average waveform amplitude expressed in SNR. Top and bottom share the same x-axis.

https://doi.org/10.1371/journal.pone.0312438.g002

precision of 97.3%±13.4% (mean±STD) was obtained, again reflecting a low false positive rate (Fig 2J), in accordance with the patch-MEA results.

These results indicate that RT-Sort is capable of detecting a subset of units with low false positive rates while being unable to detect other units. This is likely due to the relatively stringent detection constraints that are imposed on RT-Sort, in order to reliably detect propagating action potentials. Unlike the patch-MEA recordings, RT-Sort did miss some of the true positive spikes in the simulated dataset (S7A, S7B Fig). This resulted in an overlap score per unit of 0.65±0.24 (mean±STD, Fig 2K). The missed true positive spikes could partially be due to the fact that the spike detection model was trained on real waveforms while the biophysically detailed neurons did not always yield realistically looking waveforms (S7C Fig). RT-Sort is more sensitive to not detecting unrealistically looking waveform shapes than template matching algorithms, which makes it less likely for RT-Sort to detect false positive spikes.

Detection latencies

For RT-Sort to be used in real time experiments, it is important to minimize the latency between the occurrence of the action potential and its detection. In our case, the latency between the waveform trough and its sorted detection can be ascribed to three different factors: biological constraints, detection speed and sorting speed (Fig 3A). The biological constraints consist of the duration of the remainder of the spike waveform and propagation to occur after the waveform trough. For the spike detection, using a single NVIDIA-RTX-A5000 GPU allowed us to perform a forward pass through the MaxWell MEA spike detection model for 1020 electrodes simultaneously (total number of available recording electrodes for MaxWell Biosystems HD-MEAs) in 1.12ms and a forward pass through the Neuropixels spike detection model for 384 electrodes simultaneously (total number of available recording electrodes on a Neuropixels probe) in 0.92ms. Adding additional electrodes from there only led to a linear increase in the required computation times (Fig 3B). A first order linear regression model fitted to the forward pass duration yielded: duration (ms) = 0.155 + 0.001 * #elec (R2 = 0.997, P = $4.14*10^{-15}$) for the MEA model and duration (ms) = 0.155 + 0.002 * #elec (R2 = 0.999, P = $4.09*10^{-19}$) for the Neuropixels model. The steeper slope for the Neuropixels model is due to the higher sampling rate of 30kHz compared to 20kHz.

Based on these results, we conclude that our model is very well suited for detecting action potentials within milliseconds of their occurrence, at a higher sensitivity than thresholding the signal and all while being robust against overlapping waveforms as well as changing waveform shapes.

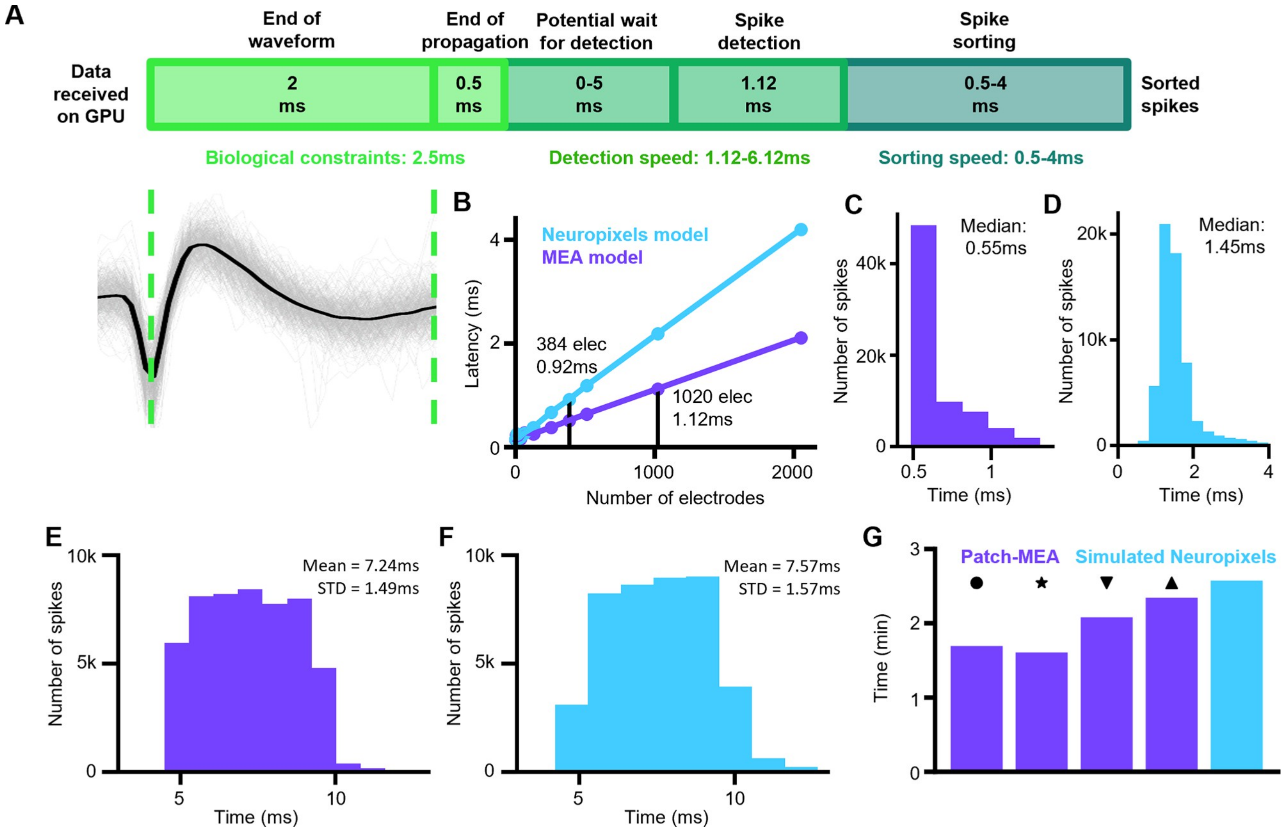

Fig 3. Real time spike sorting with latencies in the range of synaptic transmission. A) Schematic representation of the durations of the different steps to go from action potential trough to sorted spike. The duration is split up into 3 categories: Biological constraints refer to biologically intrinsic waiting times since the action potential trough occurs at the soma in order to measure all the data required for the detection. This consists of measuring the action potential until the end of the waveform (bottom left) plus an additional 0.5ms for the action potential to propagate. Detection speed refers to the duration for the spike detection model to detect spikes on all the electrodes used in the recording. Sorting speed refers to the duration to take the spike detection probability outputs and use those to assign spikes to the correct units. **B)** Computation time to perform a forward pass through the spike detection model as a function of the number of electrodes in the recording. A first order linear regression model fitted to the forward pass duration yielded: duration = 0.155 + 0.001 * #elec (R2 = 0.997, P = $4.14*10^{-15}$) for the MEA model in purple and duration = 0.155 + 0.002 * #elec (R2 = 0.999, P = $4.09*10^{-19}$) for the Neuropixels model in cyan. **C)** Distribution of spike sorting duration on the detection model outputs for the two patch-MEA with real time sorting from Fig 2H. **D)** Distribution of spike sorting duration on the detection model outputs for the real time detections on the simulated Neuropixels recording from Fig 2I–2K. **E)** Distribution of spike detection and sorting durations for the real time sorting on the two patch-MEA recordings from Fig 2H. Time reflects duration from waveform trough until sorted detection (mean±STD over all detected spikes: 7.24ms ±1.49ms). **F)** Distribution of spike detection and sorting durations for the real time sorting on the simulated ground truth dataset from Fig 2I–2K. Time reflects duration from waveform trough until sorted detection (mean±STD over all detected spikes: 7.57ms±1.57ms). **G)** Durations for detecting propagation sequences and sorting all spikes in the different offline patch-MEA recordings from Fig 2G (purple, markers same as Fig 2G) and in the pre-recording for the simulated Neuropixels recording made before running RT-Sort in online mode for Fig 2I–2K (cyan). Time is expressed as number of minutes per minute of recording time.

https://doi.org/10.1371/journal.pone.0312438.g003

Subsequently processing the model detections into sorted spikes only required an additional 0.5-4ms with a distribution median of 0.55ms (Fig 3C) for the patch-MEA recordings from Fig 2H and a distribution median of 1.55ms (Fig 3D) for the simulated Neuropixels recording from Fig 2I–2K. Altogether, this yielded a total latency from waveform trough to sorted detection of 7.24ms±1.49ms (Fig 3E) for the patch-MEA recordings and 7.57ms ±1.57ms (Fig 3F) for the simulated Neuropixels recording (mean±STD) when applying RT-Sort in online mode. These results clearly illustrate the detection and sorting speed enabled by RT-Sort. Running RT-Sort in online mode requires detecting sequences in offline mode first in a pre-recording. Detecting sequences in offline mode took 1.93±0.30 minutes (mean ±STD) per minute of recording for the patch-MEA recordings and 2.58 minutes per minute of recording for the simulated Neuropixels recording (Fig 3G). Therefore, RT-Sort can be used in online mode with sequences recorded only a few minutes in advance.

RT-Sort compared to other sorters

We next compared RT-Sort to 6 different conventionally used sorting algorithms (SpyKING Circus, HDSort, HerdingSpikes2, IronClust, Kilosort2, IronClust and Tridesclous [4–9]) that were all applied to the same mouse Neuropixels recording [16]. RT-Sort detected a total of 168 units in the Neuropixel recording, which is fewer than any of the other sorters, in agreement with the results observed in the simulated dataset (S8A Fig). The units consisted of detected propagations of 40μm to 150μm in length (S8B Fig) and up to 500μs in duration (S8C Fig).

We sought to quantify the quality of the detected units by comparing them to the other sorters. A previous comparison between the six other sorters on this dataset showed only a very limited overlap in the detected units between the different sorters [15] (see Methods: Overlap score for how the overlap was assessed). We hypothesize that the low overlap between units could be due to false positive spike detections of different sorters which could make the spike trains of different sorters so different that they are not recognized as coming from the same unit. This would mean that good performance of a certain sorter could potentially be obscured by bad performance of a different sorter (Fig 4A). As a result, for comparing detections between sorters, we quantified how similar the spikes that were detected by either RT-Sort or a comparison sorter alone are compared to the spikes that both sorters agreed on.

Firstly, the distribution of the interelectrode intervals between each loose electrode in the sequence and the root electrode was constructed for "matched" spikes, "RT-Sort only" spikes and "Other sorter only" spikes (Fig 4B). For every spike in each of these three groups, the interelectrode interval difference between the spike and the distribution mean of the matched spikes was used as a measure for how different the propagation detected on this loose electrode was compared to the agreed upon spikes by RT-Sort and the other sorter, with absolute differences clipped at a maximum of 7 frames (Fig 4C). Next, an interelectrode interval difference score per unit was defined as the average absolute difference over all spikes on all loose electrodes. To account for variability intrinsic to the variable action potential propagation speed, the average absolute difference over all matched spikes was subtracted from the average absolute difference of both the "RT-Sort only" and the "Other sorter only" spikes to get the final interelectrode interval difference score for both groups.

A paired one-sided t-test showed that the interelectrode interval difference score was significantly larger for each of the other sorters compared to the RT-Sort only spikes, taken over all units with an overlap score of $\geq$0.1 where both sorters detected at least 10 unique spikes (Fig 4D, 4E, SpyKing Circus: $P = 3.50*10^{-7}$, n = 115 units; Kilosort2: $P = 1.49*10^{-8}$, n = 119 units; Herdingspikes2: $P = 9.76*10^{-5}$, n = 78 units; Tridesclous: $P = 5.87*10^{-10}$, n = 104 units; HD-Sort: $P = 2.36*10^{-13}$, n = 108 units; IronClust: $P = 2.43*10^{-4}$, n = 106 units). Note that n is notably lower than the number of units detected by these sorters which is due to the fact that RT-Sort has a lower recall on unit detection due to more stringent detection thresholds (S8A Fig).

We applied the same procedure on detected spike amplitude differences instead of interelectrode interval differences. Also here, we noted a significantly larger difference score for the "Other sorter only" spikes compared to the "RT-Sort only" spikes, except for HDSort (Fig 4F, 4G, SpyKing Circus: $P = 1.15*10^{-8}$, Kilosort2: $P = 2.49*10^{-8}$, Herdingspikes2: $P = 5.83*10^{-4}$, Tridesclous: $P = 5.55*10^{-7}$, HD-Sort: P = 0.0789, IronClust: $P = 8.24*10^{-6}$). However, HDSort had the largest increase in the interelectrode interval difference score compared to RT-Sort of all other sorters.

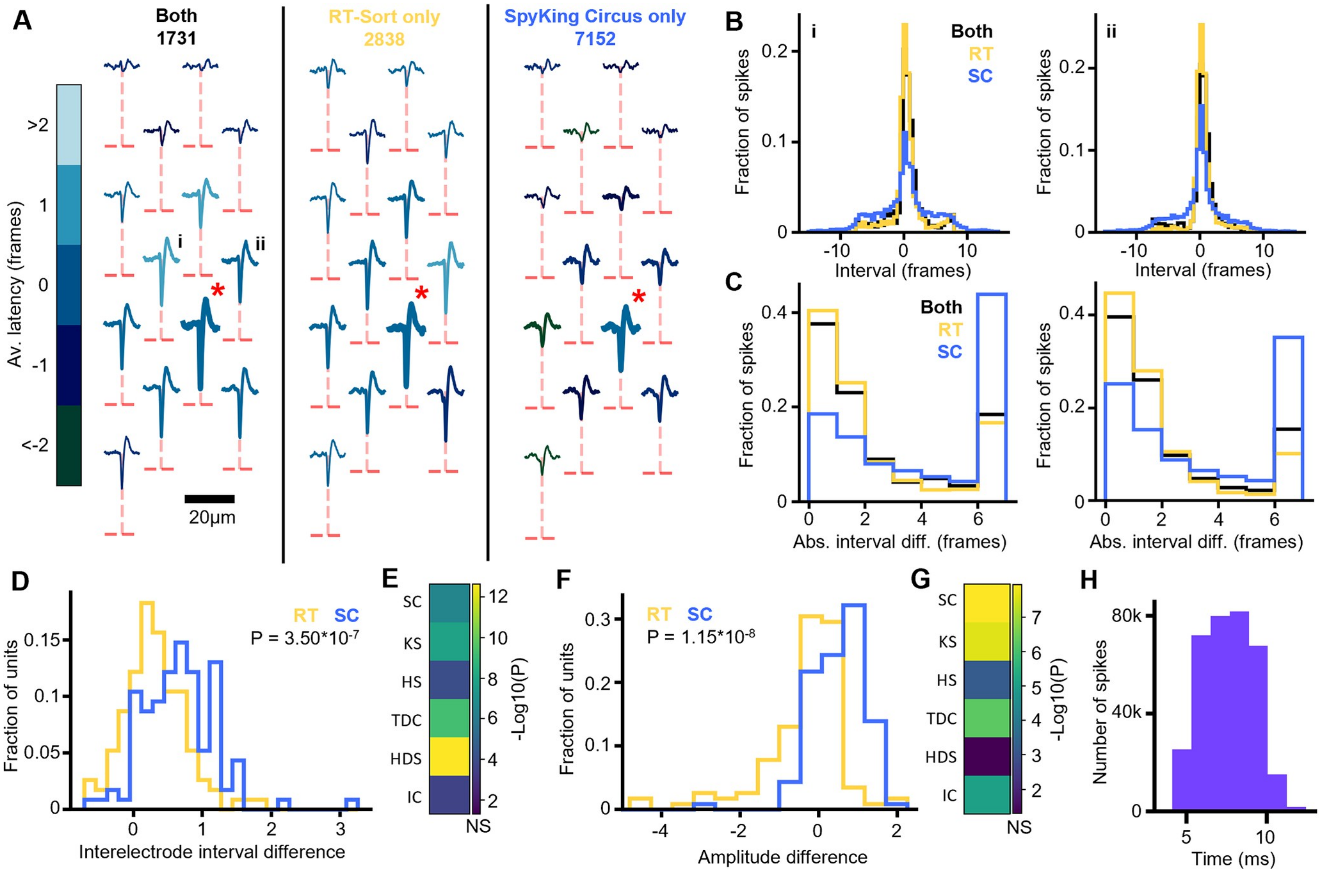

Fig 4. RT-Sort detects spikes with consistent propagations and amplitudes. A) Waveform footprint of a unit detected by both RT-Sort and SpyKing Circus. Left: averaged footprint over all spikes detected by both RT-Sort and SpyKing Circus. Middle: averaged footprint over all spikes only detected by RT-Sort. Right: averaged footprint over all spikes only detected by SpyKing Circus. 5 times the signal to noise ratio of the signal measured on each electrode is indicated with dotted red lines. The root electrode detected by RT-Sort and used for the spike triggered averaging is marked with the red star. All detected loose electrodes by RT-Sort are marked in bold. For each electrode, the color of the trace represents the average detection interval relative to the root electrode. **B)** The distribution of interelectrode intervals per spike relative to the root electrode for the comparison electrodes marked in A. Each spike group in A (both, RT-Sort only and SpyKing Circus only) is plotted as a separate distribution. The wider base of the SpyKing Circus distribution indicates spike contamination from different units for the SpyKing Circus only spikes, which is not detected by RT-Sort. **C)** The distribution of interelectrode interval differences compared to the mean of all spikes detected by both RT-Sort and SpyKing Circus for the comparison electrodes in B. Differences are clipped at 7 frames. **D)** Interelectrode interval difference scores for all matching RT-Sort and SpyKing Circus units. SpyKing Circus only spikes are significantly more different from the matched spikes in their interelectrode intervals compared to the RT-Sort only spikes ($P = 3.50*10^{-7}$, one-sided paired t-test). **E)** -Log10 (P) values for inter-electrode interval differences comparing RT-Sort only spikes to other sorter only spikes relative to matched spikes. All other sorters are significantly more different ($P<0.05$, one-sided paired t-test). Abbreviations: SC = SpyKing Circus, KS = Kilosort2, HS = Herdingspikes2, TDC = Tridesclous, HDS = HD-Sort, IC = IronClust. **F)** Amplitude difference scores for all overlapping RT-Sort and SpyKing Circus units. SpyKing Circus only spikes are significantly more different from the matched spikes in their amplitudes compared to the RT-Sort only spikes ($P = 1.15*10^{-8}$, one-sided paired t-test). **G)** -Log10(P) values for amplitude differences comparing RT-Sort only spikes to other sorter only spikes relative to matched spikes. All other sorters are significantly more different ($P<0.05$, one-sided paired t-test) except HD-Sort ($P = 0.0789$, one-sided paired t-test). Abbreviations same as E. **H)** Distribution of spike detection and sorting durations for the real time sorting on the Neuropixels dataset from Fig 4. Time reflects duration from waveform trough until sorted detection (mean±STD over all detected spikes: 7.62ms±1.58ms).

https://doi.org/10.1371/journal.pone.0312438.g004

Despite there not being a ground truth for these recordings, these results indicate that the RT-Sort only detections were more similar to the matched detections than the detections that were only made by the other sorter. This makes it more likely that the RT-Sort only detections reflect false negatives of the other sorter while the other sorter only detections could reflect false positives of that sorter. And so we conclude that RT-Sort is able to detect a lower number of units but that the units that are detected are of relatively high quality. Most importantly, these results were obtained by detecting spikes in online mode where it took 7.62ms±1.58ms (mean±STD) between the trough of the waveform and its sorted detection (Fig 4H).

Altogether, our results indicate that RT-Sort performs at least at a comparable level as the currently available spike sorting algorithms, although with a lower unit recall. However, unlike all of the other spike sorting algorithms, RT-Sort is able to achieve this performance in real time, with sorting occurring within milliseconds after the spike occurred, while the experiment is still ongoing. Thus it is the first algorithm that allows for closed loop feedback experiments with single neuron resolution and mono-synaptic delay times.

Discussion

The technological advancements in hardware used for extracellular neural recordings have yielded dense multi electrode recording devices for both *in vivo* [1, 2] and *in vitro* [3] recordings with interelectrode pitches smaller than 20μm. These devices allow for the recording of the activity of a single neuron by multiple adjacent electrodes. With the high temporal resolution of these recording devices in the range of 20–30 thousand frames per second, consistent time delays can be observed among electrodes that detect the action potential of a single neuron resulting from the time it takes for this action potential to propagate along the axon [33]. This action potential propagation detection has been used to enhance the classification of neuron types based on high density recordings [34]. In this work, we presented RT-Sort (Real Time Sorting) that for the first time utilizes action potential propagation detection instead of spike waveform template matching as a spike sorting methodology to identify individual neurons in high-density extracellular field recordings, enabling millisecond latencies from action potential occurrence to sorted detection.

Spike sorting based on action potential propagation detection overcomes the issues related to changes in waveform shapes and overlapping spikes (Fig 2, S4, S5 Figs). Simultaneously, action potential propagation-based spike sorting reduces the rate of false positive spike detections (Figs 2 and 4). Importantly, all of this can be achieved while able to detect and sort spikes within less than 10ms after their trough time (Fig 3), as opposed to waiting until completing a recording before the spike sorting procedure can even be initiated. This puts the sorting latencies of RT-Sort in the range of signal transmission delay times of single synapses [27].

To obtain an optimal sorting performance, it is important to detect the propagating action potential on as many electrodes as possible, with high temporal precision. Therefore, we developed a convolutional neural network (CNN) that detects waveform shapes in noisy traces on single electrodes (Fig 1A–1D, S1A–S1D Fig), with a significantly higher recall than 5RMS based spike detection (Fig 1G, S1G Fig). When training our model, we ensured that it is sensitive to the wide variety of waveform shapes that can occur while the action potential propagates along various parts of a neuron. As a result, even the *in vivo* recordings from Neuropixels probes yielded propagating action potentials over 150μm, taking up to 500μs for their propagation (S8B, S8C Fig).

The single electrode CNN detections form the inputs for the subsequent spike sorting algorithm. In offline mode, this algorithm takes the spike detections from a recording and finds recurring sequential activations of adjacent electrodes in sub-millisecond time scales. Using the interelectrode intervals of these observed sequential detections, newly occurring action potentials in live ongoing recordings can directly be matched to the previously detected propagation pathways, resulting in real time spike sorting, which is the basis for designating the algorithm RT-Sort (Real Time Sorting). The only requirement for applying RT-Sort in online mode is to make a short pe-recording of up to 5 minutes, using the same electrode configuration as will be used in the actual experiment. Subsequently processing this recording in offline mode to detect the sequences that can be used in online mode takes about 1.93±0.30 minutes (mean±STD) per minute of recording (Fig 3G).

In this work, we presented a version of RT-Sort optimized for organoid recordings from HD-MEAs from MaxWell Biosystems (Figs 1 and 2) [3] and a separate version of RT-Sort optimized for mouse recordings from Neuropixels 1.0 (S1 Fig, Fig 4) [1]. However, these are just examples of commonly used recording devices for both *in vitro* and *in vivo* applications. RT-Sort was developed to easily produce optimized versions for any dense multi electrode recording device with interelectrode spacing of ≤50μm. The main aspect of recording device and biological sample specific optimizations for RT-Sort resides in the single electrode spike detection model. Here we trained a CNN to optimally detect waveforms from a specific biological model type in recording device specific noise (See Methods for more details). If another biological sample type and/or recording device were to be used, the currently presented versions of RT-Sort will likely still yield good performance. For example, we successfully applied our human organoid MEA model on rat primary cultures plated on these same MEAs (Fig 3A–3H), and we got good results for a subset of units when we applied our mouse Neuropixels model on biophysically simulated waveforms in random noise (Fig 3I, 3J). However, when applying our MEA model to the biophysically simulated Neuropixels dataset, the results were notably worse (S7D Fig). Therefore, for optimal results, we recommend training a model that is specific to the biological sample type and recording device. This can easily be done through the publicly available RT-Sort Python library and only requires a few previous recordings from the same biological sample type and recording device combination.

RT-Sort has been tested on a simulated ground truth recording as well as simultaneous intracellular and extracellular field ground truth recordings. Furthermore, its detection and sorting performance has been compared on real *in vivo* recordings [16] against a variety of commonly used spike sorters [4–9]. Our results indicate that RT-Sort has a high precision, meaning that it detects very few false positive units and spikes (Fig 2). In the recording without ground truth, this is reflected in the fact that the spikes that RT-Sort detects but that are not detected by the other sorter are more similar to the spikes that both sorters agree on compared to the spikes that the other sorter detects but that RT-Sort doesn't (Fig 4A–4G). This makes it more likely that the RT-Sort only detections reflect false negatives of the other sorter while detections from other sorters only could reflect false positives of the other sorter. These results come with the price of a lower recall on unit detection, meaning that RT-Sort misses units that other sorters might detect. Since our recall of single spike detections was significantly higher than 5RMS (Fig 1G, S1G Fig), we suspect that this reduced recall on unit detection is due to the more stringent unit detection bounds imposed on RT-Sort, requiring the detection of a spike on at least 4 different recording electrodes. These bounds are applied to force detections to resemble propagating action potentials which leads to the improved precision.

Given that the main differentiator of RT-Sort is its short latency spike sorting in the millisecond range, we anticipate that the lower unit recall does not form a significant issue for the experimenter as long as the detected units are sorted with the high accuracy that we report. This would still enable the experimenter to design a wide range of closed loop experiments with tens to hundreds of units that can be recorded in real time (RT-Sort detected 168 units in the Neuropixels recording in Fig 4). Nonetheless, we anticipate that future iterations of action potential propagation-based spike sorting algorithms will further improve on RT-Sort's performance, both when it comes to precision and recall as well as the sorting speed.

Areas where we see further space for improvement include the architecture and training procedure of the single spike detection model. For example, we have observed slightly better performance for deeper detection models (consisting of more layers) but this also notably increased detection latencies. A better detection performance might also be obtained by including a larger number of averaged waveform shapes used for generating semi-artificial training data to get an even more complete picture of all the different possible waveform shapes that might be present in a recording. Furthermore, there will likely be room for further optimizations in the sorting algorithm and chosen parameters for detecting propagation sequences, merging them and assigning spikes. For example, to better handle electrode drift in long duration recordings, as this can alter the interelectrode latencies and relative amplitudes of the propagating action potential. We anticipate a better performance in handling electrode drift by integrating dedicated motion correction algorithms [36] and by dynamically updating the selection of detected propagation signals and their corresponding parameters while the experiment that applies RT-Sort in real time is ongoing. This is something that we are keen to implement in the next iteration of RT-Sort. Finally, through clever multithreading and enhanced computational processing power, we anticipate the sorting latencies of future propagation-based spike sorting routines to approach latencies below 5ms.

In making improvements, one will have to carefully balance sorting accuracy with detection latencies. With the continuous improvement of processing capabilities, we expect this balance to increasingly favor sorting accuracy without having to give in on detection latencies. Note that all the reported results were obtained using a single NVIDIA-RTX-A5000 card, which we anticipate to soon become outdated and easy to outperform. In presenting RT-Sort, we hope to promote a paradigm shift in the field of spike sorting towards propagation-based methods over template matching methods.

Despite leaving space for improvements, we think that in its current form, RT-Sort will already provide significant value to the researchers that conduct neural extracellular field recordings with dense multi electrode recording devices, both *in vivo* and *in vitro*. RT-Sort enables, for the first time, an easily implementable spike sorting routine with time latencies for sorted spike detection that are comparable to single synaptic delay times. This important feature of RT-Sort opens up interesting opportunities for closed loop experiments, where a neural system receives external inputs based on preceding activity recorded from the same system. These types of experiments have been performed in various forms but always utilize multi-unit activity as read-out of the neural system [20–23]. With RT-Sort, closed loop experiments can now be conducted with real time single unit read out accuracy, which enables studying neural circuit dynamics in response to external inputs with a greatly enhanced granularity. Similarly, we anticipate that this single unit real time detection granularity for large populations of recorded neurons can greatly enhance the performance of brain machine interfaces utilized in real time.

In summary, RT-Sort exemplifies a novel approach to spike sorting based on action potential propagation which comprises high-fidelity sequential detection of a propagating action potential on multiple adjacent electrodes with sub millisecond but nonzero time delays. This approach lies closer to the biology that underlies action potentials than template matching, the current norm in spike sorting algorithms. As a result, RT-Sort is more robust against waveform shape changes and overlapping waveforms than conventional methods, yielding a reduced false positive rate in detected units and spikes. Importantly, RT-Sort is able to detect and sort spikes within less than 10ms after their occurrence, on the range of single synaptic delay times. This makes it a useful tool for closed loop experiments and brain machine interfaces as well as various other implementations. We hope that the broader community will take interest in action potential propagation-based spike sorting and apply this new paradigm to continuously iterate on better performing real time spike detection algorithms.

Materials and methods

Ethics statement

For the newly generated patch-MEA dataset, the experimental protocols involving animal tissue harvesting were approved by the veterinary office of the Canton Basel-Stadt (license

#2358) according to Swiss federal laws on animal welfare and were carried out in accordance with the approved guidelines. Pregnant Wistar rats (obtained from Janvier Labs) were anesthetized with isoflurane and decapitated, followed by harvesting of the brains of E-18 embryos. All other data comes from previously published sources and is accessible in publicly available repositories (see: Data and materials accessibility).

Real time spike detection

RT-Sort uses action potential propagation detection to identify individual neurons in recordings with dense electrode configurations of ≤50μm interelectrode pitch. This requires a sensitive detection of the action potential on multiple electrodes followed by a sorting procedure that assigns the action potentials to the right neurons. All of this needs to happen in millisecond timescales to be able to use RT-Sort in real time. For this, we developed a spike detector that can simultaneously detect spikes on over a thousand different electrodes within millisecond time scales, while being more sensitive than 5RMS threshold crossings and able to detect overlapping waveforms.

A convolutional neural network (CNN) was trained to detect neural waveform shapes in a noisy signal. Due to variability in the type of noise produced by different recording devices, it is important to train a specific model per recording device type. In this work, separate models are trained for human brain organoids recorded with MaxWell Biosystems multi-electrode arrays [3] and for intact mouse brains recorded with Neuropixels probes [1]. The data used for the training and validating process comes from [16, 37] respectively and can be found here https://datadryad.org/stash/dataset/doi:10.25349/D9031Z and here https://buzsakilab.nyumc.org/datasets/SiegleJ/AllenInstitute_744912849/session_766640955/. All training and validating is done using an NVIDIA-RTX-A5000 GPU.

CNN architecture

The CNN for spike detection consists of 4 convolutional layers. The first three convolution layers have 50 filters with a kernel size of the number of frames in 1 millisecond of raw data + 1 (21 for MEAs with sampling rate of 20kHz, 31 for Neuropixels with sampling rate of 30kHz), stride of 1, and ReLU nonlinearity. The last layer (the output layer) has 1 filter with the same kernel size and stride as the previous layers and followed by a Sigmoid nonlinearity (Fig 1A, S1A Fig for Neuropixels model with slightly altered parameters to account for different sampling rate). The CNN takes in a 10ms snippet of raw, unfiltered data. The output layer contains a node for each frame in the 2-8ms period within the input data. The kernel size was chosen so that each node in the output layer receives information from all frames in the input layer in the period of 2ms before and after the frame that the node makes the prediction for.

With this architecture, the model is trained so that for the provided snippet of 10ms of data, it returns a likelihood of a spike waveform trough to be present for each recording frame in the period of 2-8ms within this window. The outer boundaries of the window are not considered since the full shape of the waveform would not be present within the window, leading to a worse detection performance. Each frame receives a score between 0 and 100 indicating the likelihood of there being a waveform trough in this frame. Before each 10ms sample is fed into the CNN for training or inference, it is converted from int16 format to float16. To account for low frequency, high amplitude LFPs impacting the signal, all samples are standardized by subtracting the median of the sample.

Spike detection semi-artificial data generation

Note that it is difficult to obtain sufficient ground truth spike detections to effectively train a neural network. The availability of high electrode density electrophysiology recordings with corresponding ground truth is limited, and it is difficult to generate this data [36]. Alternatively, ground truth training and validation datasets are generated through artificial data amplification, inspired by the work from Parks et al. [35]. First, a selection of recordings from the recording device and sample type of interest are made. In our case, 6 recordings from different human brain organoid slices are selected where 5 recordings are used to generate training data, and 1 recording is used to generate validation data in a 6-fold cross validation manner as described next. A same selection is made for intact mouse brain recordings made *in vivo*. Note that during validation, only data generated from a different replica of the recording device is used (although from the same type), interfacing with a different biological sample (organoid slice or mouse brain). In this way, the model is trained for recording device type and sample type specificity while allowing for generalizability among different replicates of recording devices of the same type and generalizability among different biological replicates.

The selected recordings are bandpass filtered between 300Hz and 6000Hz and subsequently spike sorted using Kilosort2 and curated. In the initial curation round, all units with at least 30 detected spikes, less than 1% of spikes violating a <1.5ms inter spike interval and a signal to noise ratio larger than 5 on the main electrode are selected. The spikes of these units are considered to be all spikes in the recording. From the selected units, a smaller selection is made for training our spike detection model by applying a second round of curation. Here all units with at least 50 detected spikes are selected and the filtered signal on each electrode is averaged around the maximum absolute waveform peak from the detections on the main electrode.

In order to detect a propagating action potential along the whole axon, it is important for the spike detection algorithm to be able to detect spikes that are not just from the highest amplitude electrode. Therefore, waveforms were selected from all electrodes where the detection trough of the averaged waveforms cross -19μV (-36μV for the Neuropixels data). Alternatively, if the electrode detected a waveform with a peak at least twice as large as the through, this electrode was selected if the peak crossed 19μV (36μV for the Neuropixels data). Also, the standard deviation in the peak amplitude divided by the absolute value of the amplitude had to be less than or equal to 0.6. The averaged waveforms for these electrodes are stored for generating training and testing samples (Fig 1B, S1B Fig for Neuropixels example). This yields 82.83 ±74.62 averaged waveforms from single electrodes per recording (1327.33±427.00 for the Neuropixels). Note that the selection includes averaged waveforms of a broad variety of shapes from electrodes that recorded a signal from any part of a neuron, which allows us to detect a propagating action potential along different parts of the neuron.

From the same recordings as used to select the average waveforms, 10ms periods with no detected spikes on any electrode are extracted to obtain recording device specific noise. To ensure that there is no presence of waveforms in these 10ms periods, only periods where there were no spike troughs detected in the 3ms before the start of the window until 3ms until after the end of the window were used. Spikes after the initial curation round were used to find these no-activity periods. Next, an averaged waveform is pasted into a noise snippet to create an artificial spike event that can be used for training and validation (Fig 1C, S1C Fig for Neuropixels example). In this way, the average waveform shapes pasted into the recording and background noise allowed us to generate near infinite training samples that we could use to train our CNN to be able to detect waveform shapes of any part of a neuron in noise intrinsic to the sample preparation and recording device. Furthermore, multiple waveforms can be pasted into the same noise snippet to artificially create overlapping waveforms. When doing so, the number of pasted waveforms was randomly determined for each generated sample, following the probability distribution 50%, 30%, 12%, 6%, 2% for 0, 1, 2, 3, or 4 waveforms respectively. The distance between waveform troughs was randomly determined within a range of 0.2ms to

6ms apart (Fig 1D, S1D Fig for Neuropixels example). In cases where multiple waveforms are added to the same noise snippet, no sample has multiple copies of the same waveform shape.

During the 6-fold cross-validation training, one of the 6 recordings is used as the validation dataset while the remaining 5 are used as the training dataset. For the MEA model, one validation and training epoch consisted of 2 and 20 samples per waveform in the dataset, respectively. For the Neuropixels, only 1 sample per waveform was used since recordings yielded more different waveform shapes for the training and validation datasets. In the training dataset, a waveform from any of the 5 recordings could be pasted into a 10ms noise sample from any of the other 5 recordings.

CNN training

The generated training samples are used to train the aforementioned CNN. The batch size is 1. Stochastic gradient descent with an initial learning rate of 0.000776 (0.0002 for the Neuropixels model) and momentum factor of 0.85 is used. If the validation does not decrease by at least 0.0001 (0.01 for the Neuropixels model) after 5 epochs, the learning rate decreases by a factor of 0.4. These optimal hyperparameters were obtained through 6-fold cross-validation with Bayesian tuning using CometML. The model is trained until the validation loss does not decrease by at least 0.01 after 10 epochs. After training is completed, the weights that give the best validation loss are used. This yields 33.5±7.1 epochs (mean±STD) for the MEA model and 18.7±5.0 epochs for the Neuropixels model. The small differences between the training loss and testing loss indicated that our models are not overfitting nor underfitting the data (Fig 1E, S1E Fig for Neuropixels results).

Validating and detection threshold

Model detections within 0.4ms of the actual absolute waveform peaks from the samples pasted into the snippet are considered to be true positives. For a range of thresholds for the detection score, the precision and recall are computed as a function of the detection threshold. The optimal detection threshold of each model trained on one of the 6 folds was chosen by optimizing the F1 score. Using this threshold, the performance of the model detections compared to 5RMS based detection was assessed by computing the precision and recall.

The training and validation were performed using semi-artificially generated samples as described above. The MEA model trained on the fold with the largest number of waveforms in the training dataset and the Neuropixels model trained on the fold where the validation dataset consisted of the recording described in Results: RT-Sort compared to other sorters were then applied in our real time spike sorting algorithm as described below. The results after sorting were further tested against actual ground truth datasets as will be described in Methods: Real time spike sorting.

Since for the spike sorting algorithm, we require sequential detections on a specific set of electrodes in a specific order, detecting false positives on single electrodes is less problematic than detecting false negatives. A false positive detection on a single electrode will not yield an actual detected spike since it will not meet the sorting thresholds. As a result, we will apply two model detection thresholds: the stringent threshold (which is at the optimal F1 score, Fig 1I, S1I Fig) and the loose threshold (below the optimal F1 score). Specifically, we will use a stringent threshold of 27.5% (17.5% for the Neuropixels model) and a loose threshold of 10% (7.5% for the Neuropixels model). The loose threshold was chosen such that the recall on the validation dataset was about 15% larger than the precision, in order to prioritize a low false negative rate of single electrode spike detection. These thresholds can be adjusted by the user based on their sample and their experimental needs.

CNN inference

The trained model is applied to recordings with varying degrees of noise. To ensure that the model's inputs are on the same amplitude scales as it encountered during training, the mean amplitude deviation of the 5 recordings in its training dataset (training amplitude deviation) is calculated. The amplitude deviation of a recording is calculated as follows: for the first 50ms on each electrode, the interquartile range of the recording traces is calculated. The median of these ranges is the amplitude deviation.

To scale a recording to the model's expected amplitude range, the amplitude deviation of the recording is calculated, and the recording's traces are multiplied by the training amplitude deviation divided by the recording's training amplitude deviation.

From detected spikes to propagation sequences

The sorting algorithm takes the raw recording traces and real time detection scores as inputs and finds propagating action potentials based on recurring consecutive detection patterns on adjacent electrodes (S9 Fig). The algorithm can be used to process previously made recordings (offline mode), and it can also be applied in real time (online mode). Using RT-Sort for real time sorting requires a pre-recording of the same sample with the same electrode configuration as the actual experiment. In this pre-recording, action potential propagations are detected as described in Methods: Propagation sequence detection. Afterwards, action potentials detected in real time are assigned to these sequences that are detected in the pre-recording. We recommend a pre-recording of at least 40 seconds, although 5 minutes is ideal for optimal sequence detection of units with lower firing rates (S10 Fig), as propagation detection requires at least 10 spikes in the pre-recording. The length of the pre-recording mainly affects the number of different units that can be tracked in real-time, whereas the average precision and recall over the detected units remains mostly consistent.

For the description of the real time spike sorting algorithm below, the stringent and loose detection thresholds will be used as defined previously. Furthermore, two types of distance thresholds will be used. The "inner" electrodes will be within 50μm of the electrode of interest whereas the "outer" electrodes will be within 100μm of the electrode of interest. These distance metrics were chosen for optimal performance with a checkerboard Neuropixels configuration as well as MaxWell MEA electrode configurations that use every other electrode resulting from a sparse 7x activity scan, yielding a 35μm pitch (S11 Fig). More information about the different parameters used for RT-Sort can be found in Methods: RT-Sort parameters.

Propagation sequence detection

The first phase of the sorting algorithm consists of propagation sequence detection. For each recorded electrode, which will be referred to as the root electrode of the sequence (Fig 2A, S9B–S9D Fig marked with red star), all stringent model detections are obtained. Only the stringent model detections that have their largest standardized amplitude (defined below) on the root electrode and no higher standardized amplitude detections within 0.5ms on any of its outer electrodes are considered.

All amplitudes are detected in the raw, unfiltered data. To account for LFP impacting the amplitude, amplitudes are standardized according to the calculation below. In the remainder of the text, any reference to amplitudes pertains to amplitudes standardized in this manner:

The voltage at the model's predicted spike location ("absolute amplitude") on the electrode is extracted. For the preceding 50ms ("window"), the following values are found:

A = median(window)

B = median(|window—A|) / 0.6745

The standardized amplitude is then equal to (absolute amplitude—A) / max(0.5, B). Where B is an estimate of the standard deviation of the background noise [38, 39].

The selected model detections on the root electrode that have a stringent detection on at least one inner electrode within 0.1ms are selected. These will be referred to as the root detections (S9B–S9D, number of detections on root electrode). All root electrodes with an average firing rate of at least 0.05Hz over the whole recording are selected for further processing. For each root electrode, all root detections are compared to all stringent detections on the inner electrodes of the root electrode in a pairwise manner. In this comparison, the root detections might be split into multiple clusters as follows:

During the comparison to each inner electrode ("the splitting electrode") the time intervals between the root detections and the detections on the splitting electrode that occur within 0.5ms of any of the root detections ("splitting codetections", see detections on all electrodes besides the root electrode in S9B–S9D) are obtained. To account for the fact that action potential propagations from different neurons may be detected by a single root electrode–splitting electrode pair (S9D Fig), 1-component, 2-component, 3-component, and 4-component Gaussian mixture models are fit to the codetection interval distribution. The model with the lowest Bayesian information criterion is used to determine whether there are a single or multiple interelectrode propagations present in the interval distribution (S12A Fig). To prevent overfitting the distribution, a Gaussian mixture model is ignored if it has a negative Bayesian information criterion (BIC). To prevent underfitting the distribution, a Gaussian mixture is ignored if the fitting of the model yields a FloatingPointError due to encountered underflow. If all Gaussian mixtures are ignored, the 1-component Gaussian mixture model is used.

Finally, any cluster that has fewer detections than max(10, 10% of the root detections) are discarded and their splitting codetections can be used to form other clusters (see below). If all clusters do not meet this threshold, then the root detections remain together as a group and are not split into clusters.

The root electrode's inner electrodes are sorted in ascending order of distance. Let the ith electrode refer to the ith electrode in this sorted order. All root detections are first compared to the 1st electrode (i = 1, S9B Fig, electrode next to root electrode). Every cluster that is formed is then compared to the i+1th electrode (i = 2,3,. . ., S9B Fig, subsequent electrodes in order of distance from root), treating each cluster the same as a set of root detections as described previously. This process repeats until no more electrodes remain, resulting in a branching process of the root detections splitting into clusters (S9B Fig, the total detections of the whole cluster decrease, while the growing propagation sequence is split on every subsequent splitting electrode) which then split into more clusters based on differences in interelectrode intervals. Each root detection can only join one cluster. After the clusters starting on the splitting with the ith electrode have been fully formed, the remaining root detections not assigned to a cluster are then compared to the i+1th electrode (S9C Fig, two different sequences can form from the same root electrode. S9D Fig, two different sequences can also diverge on a splitting electrode rather than a root electrode). This process repeats until there are no more root detections or no more inner electrodes for direct comparison with the root electrode.

To accommodate long neurons that could extend farther than the root electrode's inner electrodes, the list of splitting electrodes for splitting a cluster can expand. If at least 50% of the root detections form a single cluster, then the inner electrodes of the most recent splitting electrodes are added to the list of splitting electrodes in ascending order of distance with the most recent splitting electrode (S9B Fig, electrodes far removed from the root can still be included in the propagation sequence). To prevent a single cluster from traversing the entire array, a patience counter is used. The patience counter starts at 0. If a cluster is not split (i.e. the root times stay together) after all the splitting electrodes with the same distance are considered, then the patience counter increases by 1. If it reaches 6, there are no more attempts to split the cluster, regardless of the potentially remaining splitting electrodes. If the cluster is split, the patience counter resets to 0. The clusters formed in the previous steps are each considered to be preliminary propagation sequences.

It is possible for one preliminary propagation sequence to contain the spikes from multiple neurons if the spikes have very similar intervals on multiple electrodes. To account for this, a distribution is constructed of the amplitudes on the root electrode. If a Hartigan's dip test indicates that this amplitude distribution is not unimodal (P<0.1), the preliminary propagation sequence is split into multiple sequences. To do this, 2-component, 3-component, and 4-component Gaussian mixture models are fit to the amplitude distribution (using the GaussianMixture object from the Python scikit-learn library). Only the model with the lowest BIC is used to split the amplitude distribution (S12B Fig). See S12C Fig for the distribution of all P-values resulting from applying the Hartigan's dip test on the root electrode amplitude distributions. After splitting, sequences with less than 10 spikes are discarded.

Extracting the detections, intervals, and amplitudes

When all preliminary propagation sequences are formed, the median of the detection scores for all detections in a preliminary propagation sequence are computed for each electrode that is part of the preliminary propagation sequence or that falls within 50μm of any electrode in the preliminary propagation sequence. Based on the median detection scores, each sequence will have three groups of electrodes:

1. Loose electrodes: electrodes that have a median detection above the loose threshold and that are within 50μm of any of the preliminary propagation sequence's splitting electrodes (including the root electrode)
2. Inner loose electrodes: loose electrodes that are within 50μm of the root electrode
3. Footprint electrodes: loose electrodes and any electrode within 50μm of a loose electrode

Any sequence that has fewer than 4 loose electrodes or fewer than 3 inner loose electrodes is discarded to meet spike assignment thresholds (see Methods: Spike reassignment).

For all footprint electrodes, 0.5ms of the signal before and after the root electrode detection time point is extracted for each of the action potential propagation sequence detections in the preliminary propagation sequence ("propagation period"). The model's detection outputs are obtained for this propagation period. Iterating over all detections in the preliminary propagation, the maximum value in the detection output during the propagation period is obtained for each footprint electrode. The time delay between this maximum detection and the maximum detection time on the root electrode is considered to be the electrode's interval for that detection (S9A Fig). The amplitude at the electrode's maximum detection is considered to be the electrode's amplitude for that detection. The median detection score, mean interval, and mean amplitude across all detections are found individually for each footprint electrode (S9A Fig, Fig 2A, waveform colors reflect latencies). If an electrode's median detection score is less than 3%, it is set to 0. Finally, the standard deviation in the root electrode's amplitudes of all detections are extracted.

To remove sequences that are the result of noise spikes across all electrodes, any sequence that has more than max(100, 10% of the sampled electrodes in the recording) electrodes with a median detection score above the loose threshold is discarded.

Spike reassignment

The preliminary propagation sequences might only contain a low number of detections due to the strict detection thresholds. Therefore, spikes are reassigned to the preliminary propagation sequences using the same pre-recording that was used to detect the preliminary propagation sequences, in order to make the extracted median detections, mean interelectrode intervals, and mean amplitudes more robust and representative of the actual neuron that underlies the preliminary propagation sequence.

A spike may be assigned to a preliminary propagation sequence if all the following conditions are met:

1. A stringent detection must occur on the root electrode.
2. The amplitude of this detection on the root electrode must fall within 2.5 standard deviations of the mean amplitude on the root electrode.
3. At least 3 of the inner loose electrodes and max(4, 33% of the number of loose electrodes for the sequence) of the loose electrodes need to detect a spike above the loose threshold within the propagation period.
4. For all footprint electrodes, both the weighted interelectrode interval difference (relative to the root electrode) and the weighted amplitude percent difference (relative to the sequence's mean amplitudes) cannot be greater than the interelectrode interval and amplitude thresholds, respectively. For MEAs, the thresholds are 3.5 frames and 65% respectively (Fig 2D–2F for 3 single spike detections). For Neuropixels, the thresholds are 2.5 frames and 45%, respectively (S4–S6 Figs for examples of single spike detections). See Table 1 for all hardware-specific parameter differences.

The "weight" of each footprint electrode is calculated as its median detection score divided by the sum of all the footprint electrode's median detection scores. Subsequently, for each footprint electrode, the difference (or percent difference) between the sequence's mean interelectrode interval (or amplitude) and the spike's interval (or amplitude) on that electrode is found. Differences are clipped to be at most twice the interelectrode interval and amplitude thresholds. Then, the weighted sum of the differences is calculated using each electrode's weight.

When multiple sequences with the same root electrode detect a spike at the same time, the spike's difference score with each sequence is found. The difference score is calculated as the sum of:

1. The interelectrode interval difference divided by the interval threshold
2. The amplitude difference divided by the amplitude threshold
3. -0.5 times the number of the sequence's loose electrodes that detect the spike divided by the number of electrodes that have a loose threshold crossing

The spike is only assigned to the sequence with the lowest score. This process will be referred to as "repeated detection removal". After reassigning the spikes, new median detection scores, mean interelectrode intervals and mean amplitudes are computed for each preliminary propagation sequence. Preliminary propagation sequences that, after spike reassignment, have less than 10 spikes or a firing rate less than 0.05Hz are discarded.

Merging

Detecting the preliminary propagation sequences in this manner allows for the possibility of detecting the same axon multiple times. For example, the first electrode in the preliminary

Table 1. RT-Sort parameters.

Parameter	MEA	Neuropixels
General		
Stringent CNN spike detection threshold	27.5%	17.5%
Loose CNN spike detection threshold	10%	7.5%
Inference scaling numerator	12.6	15.4
Maximum propagation time window	0.5ms before and after root detection	
Amplitude normalization window	50ms	
Inner electrode radius	50 μm	
Outer electrode radius	100 μm	
Set up before detecting preliminary propagation sequences		
min_{elec} noise threshold = max(N_{elec}, M_{elec} * the total number of electrodes)	N_{elec} = 100 M_{elec} = 10%	
min_{seq} noise threshold = max(N_{seq}, M_{seq} * the total number of electrodes)	N_{seq} = 50 M_{seq} = 5%	
Stringent codetection threshold	2	
Detecting preliminary propagation sequences (splitting clusters of codetections)		
Maximum Gaussian mixture model components	4	
Minimum codetection cluster size = max(N_{split}, M_{split} * the total number of codetections before splitting)	N_{split} = 10 M_{split} = 10%	
Splitting electrode selection update threshold	50%	
Splitting patience counter	6	
Hartigan's P threshold	0.1	
Assigning spikes / merging sequences		
Minimum average detection score	3%	
Minimum inner loose electrode detections / overlaps	3	
Minimum loose electrode detections / overlaps	4	
Minimum percent loose electrode detections / overlaps	33%	
Maximum weighted interval difference (frames)	3.5	2.5
Interval clip (frames)	7	5
Maximum weighted amplitude percent difference	65%	45%
Amplitude percent difference clip	130%	90%
Maximum z-score for root electrode's amplitude	2.5	
Repeated detection removal time window	0.2ms	
Updating and filtering sequences		
Minimum percent amplitude for root electrode relocation	80%	
Minimum interval for root electrode relocation	-2 frames	
Minimum number of spikes	10	
Minimum spike frequency over entire pre-recording	0.05Hz	

All parameter values used by RT-Sort for spike detection and spike sorting. In cases where parameter values are hardware dependent, values are provided for both MaxWell MEAs and Neuropixels.

https://doi.org/10.1371/journal.pone.0312438.t001

propagation sequence that detects a signal after the root electrode might also be detected as a separate root electrode for the same axon. Therefore, the next phase of the algorithm consists of merging the detected preliminary propagation sequences into the final propagation sequences. For the first merging round, only preliminary propagation sequences that share the same root electrode are considered. For all footprint electrodes, the interelectrode intervals and amplitudes compared to the root electrode, weighted by their average detection score, are compared. The following conditions must be met for two sequences to be merged into one:

1. At least 3 electrodes must be in both sequence's inner loose electrodes.
2. The number of electrodes in both sequence's loose electrodes must be greater than max(4, 33% of the total number of electrodes that are loose electrodes for at least one of the sequences).
3. The weighted interelectrode interval difference for both sequence's footprint electrodes is at most 3.5 frames for MEAs and 2.5 frames for Neuropixels. The same clipping as in spike reassignment is used.
4. The weighted amplitude percent difference for both sequence's footprint electrodes is at most 65% for MEAs and 45% for Neuropixels. The same clipping as in spike reassignment is used.
5. The absolute difference between the sequences' mean amplitudes on the root electrode is calculated, and the largest standard deviation in the amplitudes between the sequences is found. The difference divided by this standard deviation must be at most 2.5. Due to higher neuron density in the *in vivo* recordings, this condition is implemented for Neuropixels only. The user can choose whether to use this requirement or not.

If the two sequences are merged into one, the median detections, mean interelectrode intervals, and mean amplitudes are updated accordingly.

The weights and differences for the root electrode are not considered for the weighted interelectrode interval difference since the intervals are relative to the root electrode, making the interval always 0. The midpoint method is used for the amplitude percent difference. The merge score is the sum of the interelectrode interval difference divided by the interval threshold and the amplitude percent difference divided by the amplitude threshold. The pair of preliminary propagation sequences with the lowest score is merged first until no more pairs can be merged or there is only one sequence remaining.

To better asses sequences for merging in the subsequent merging round, the root electrode of each sequence is replaced by an electrode with a mean amplitude of at least 80% of the current root electrode's mean amplitude, a median detection score above the stringent threshold, and a interelectrode interval less than -2 frames. If multiple electrodes meet these criteria, the electrode with the most negative interelectrode interval becomes the root electrode. If no electrodes meet these criteria, the current root electrode remains as the root electrode. The interelectrode intervals on other electrodes are updated accordingly.

In the subsequent merging round, root electrodes of a preliminary propagation sequence that are inner electrodes of a root electrode from a different preliminary propagation sequence are assessed for merging in the same way as described above. In this case, the root electrode with the highest mean amplitude is used as a reference for the interelectrode interval difference, and for Neuropixels, the root electrodes of both sequences must pass condition #5 described above. This final round of merging yields the actual propagation sequences. For each of these sequences, the electrode with the highest median detection score becomes the root electrode, and the intervals on the other electrodes are updated accordingly.

To conclude the pre-experiment preparations, spikes are reassigned to sequences in the same way as described in Methods: Spike reassignment but this time with applying repeated detection removal to all groups of root electrodes that detect a spike within 0.2ms and that are inner electrodes to each other. In the case of using RT-Sort for offline spike sorting, these spike times are the final results.

Real time spike sorting

In the case of using RT-Sort on online mode for real time spike sorting, the same process as spike reassignment is used to spike sort in real time using the updated median detection scores, mean interelectrode interval and mean amplitudes. Also here, repeated detection removal is applied to all groups of root electrodes that detect a spike within 0.2ms and that are inner electrodes to each other. Using the streamed data, a 10ms sample is processed every 5ms to account for the spikes in the first and last 2.5ms parts of the sample that can't be detected due to biological constraints (Fig 3A).

RT-Sort parameters

An overview of all parameters used for RT-Sort is presented in Table 1, followed by a brief explanation of each parameter underneath. Note that some parameters are hardware dependent and may need to be tuned when different hardware is used. Specifically, we noted that spike detections from the MEA recordings yielded an optimal F1 score at higher detection scores than spike detections from the Neuropixels recordings (Fig 1I, S1I Fig). For more information, see Methods: Validating and detection threshold. Due to the more conservative spike detection thresholds for the MEA recordings, we experienced that we could set parameters for the downstream assignment of spikes to propagation sequences and for the merging of propagation sequences to be more lenient for the MEA recordings than for the Neuropixels recordings. The remaining parameters were set to be the same for both MEAs and Neuropixels.

Stringent CNN spike detection threshold. Detection threshold for CNN spike detection model scores used for spike detections on the root electrode.

Loose CNN spike detection threshold. Detection threshold for CNN spike detection model scores used for spike detections on all electrodes besides the root electrode.

Inference scaling numerator. Used for scaling traces to roughly the same range as seen during training before feeding them into the CNN spike detection model.

Maximum propagation time window. Time before and after root electrode detection to consider for detections on the loose electrodes that constitute the action potential propagation.

Amplitude normalization window. Time before the spike that is used for normalizing the amplitude.

Inner electrode radius. Distance to the root electrode within which electrodes are considered to be inner electrodes for sequence detection and spike detection purposes.

Outer electrode radius. Distance to the root electrode within which electrodes are considered to be outer electrodes for sequence detection and spike detection purposes.

$\mathrm{min}_{\mathrm{elec}}$ noise threshold. Events that lead to stringent spike detections on at least $\mathrm{min}_{\mathrm{elec}}$ electrodes on the same frame are considered noise and discarded.

$\mathrm{min}_{\mathrm{seq}}$ noise threshold. Action potential propagation sequences with at least $\mathrm{min}_{\mathrm{seq}}$ loose electrodes are considered noise and discarded.

Stringent codetection threshold. Minimum number of stringent spike detections on inner electrodes within the maximum propagation time window that cause a stringent spike detection on a root electrode to be counted as a stringent codetection.

Maximum Gaussian mixture model components. Maximum number of components used when fitting a Gaussian mixture model to amplitude and interelectrode interval distributions.

Minimum codetection cluster size. After splitting a cluster of codetections, a cluster is discarded if the number of remaining codetections is smaller than the minimum codetection cluster size.

Splitting electrode selection update threshold. The required percentage of codetections before splitting that is preserved after the split in order for the inner electrodes of the current splitting electrode to be added to the total list of electrodes used to further split the cluster.

Splitting patience counter. Number of electrodes considered for splitting that do not lead to a split before terminating the splitting process.

Hartigan's P threshold. The minimum Hartigan's dip test P-value for a distribution to be considered unimodal.

Minimum average detection score. CNN spike detection model scores smaller than the threshold are considered noise and set to 0.

Minimum inner loose electrode detections / overlaps. Minimum number of inner loose electrode detections within the maximum propagation time window required for a spike to be considered to possibly be part of a propagation sequence during spike assignment and for two sequences to be considered as possibly undermerged sequences during merging.

Minimum loose electrode detections / overlaps. Minimum number of loose electrode detections within the maximum propagation time window required for a spike to be considered to possibly be part of a propagation sequence during spike assignment and for two sequences to be considered as possibly undermerged sequences during merging.

Minimum percent loose electrode detections / overlaps. Minimum percentage of the total loose electrodes in a propagation sequence that require a detection within the maximum propagation time window for a spike to be considered to possibly be part of a propagation sequence during spike assignment and the minimum percentage of total loose electrodes of two propagation sequences form them to be considered as possibly undermerged sequences during merging.

Maximum weighted interval difference. Maximum summed difference in the weighted interelectrode intervals between a spike and a detected propagation sequence for the spike to be considered part of the propagation sequence or between two sequences for the sequences to be considered as possibly undermerged.

Interval clip. Interelectrode interval differences larger than the interval clip are set to be the interval clip.

Maximum weighted amplitude percent difference. Maximum summed percent difference in the weighted amplitudes between a spike and a detected propagation sequence for the spike to be considered part of the propagation sequence or between two sequences for the sequences to be considered as possibly undermerged.

Amplitude percent difference clip. Amplitude percent differences larger than the amplitude percent difference clip are set to be the amplitude percent difference clip.

Maximum z-score for root electrode's amplitude. Maximum number of standard deviations that the root electrode amplitude of a spike can be removed from the mean of the distribution over all pre-recording spike detections of a detected propagation sequence for the spike to be assigned to the propagation sequence. Similarly, the maximum number of standard deviations that the average root electrode amplitude of a detected propagation sequence can be removed from the average root electrode amplitude of another detected propagation sequence for them to be considered as possibly undermerged.

Repeated detection removal time window. Spikes from the same propagation sequence that occur within the repeated detection removal time window are considered duplicate detections and only the spike with the highest similarity to the detected propagation sequence is kept.

Minimum percent amplitude for root electrode relocation. Minimum percent difference of the average amplitude of a loose electrode relative to the average amplitude of the root electrode for that loose electrode to become the new root electrode, assuming the minimum interval for root electrode relocation requirement is met.

Minimum interval for root electrode relocation. Minimum average interelectrode interval difference of a loose electrode relative to the root electrode for that loose electrode to become the new root electrode, assuming the minimum percent amplitude for root electrode relocation requirement is met.

Minimum number of spikes. Minimum number of spikes detected in the pre-recording for a propagation sequence to be detected.

Minimum spike frequency over entire pre-recording. Minimum spike frequency in the pre-recording for a propagation sequence to be detected.

The default parameters were set to provide optimal performance in a broad range of applications. However, if an RT-Sort user wants to apply RT-Sort to recordings made with different hardware and/or is unsatisfied with the performance under the default parameters, they may want to consider tuning some of the parameters. Below follow descriptions of different scenarios and our suggestions for parameters to tune:

A low number of detected preliminary propagation sequences. Consider adjusting your electrode configuration to increase the number of electrodes that can detect a signal from a single neuron and increasing the duration of your pre-recording if this was below 5 minutes. If no improvements can be made here, you can lower the stringent and/or loose spike detection threshold or lower the minimum inner loose electrode detections and loose electrode detections. However, this makes RT-Sort more susceptible to detecting noise. When working with shank probes, we recommend adjusting the minimum inner loose electrode detections and loose electrode detections first. Overall, we recommend trying to lower the loose spike detection threshold before lowering the stringent spike detection threshold as this maintains the requirement of at least one high confidence spike detection.

The detected sequences are undermerged/overmerged. Make the parameters for merging sequences more/less stringent. Specifically, for shank probes, consider tuning the minimum inner loose electrode overlaps and minimum loose electrode overlaps first. Meanwhile for MEAs, consider tuning the maximum weighted interval and amplitude differences first.

Spikes are incorrectly (not) assigned to a detected propagation sequence. If spikes are not assigned to a propagation sequence even though the correct propagation sequence was detected, consider making the parameters for assigning spikes less stringent. The tuning process may also include lowering the stringent and loose thresholds for spike detection on individual electrodes. On the other hand, if spikes are assigned to detected propagation sequences even though the spikes come from neurons of which the propagation sequence is not detected, consider making the parameters for assigning spikes more stringent. Note that even though the parameters for merging and spike assignment are set to be the same for the current configuration, this is not a requirement. Alternatively, you can try to tune the parameters for the detection of preliminary propagation sequences (see scenario: A low number of detected preliminary propagation sequences).

Noise events are detected as spikes. If noise events that are clearly not actual action potentials are incorrectly detected as spikes, you can lower the threshold for the number of stringent spike detections that occur on the same frame that lead to considering the detection as noise. You may also consider increasing the stringent and loose CNN spike detection thresholds.

Processing of the pre-recording takes too long. Detection sequences is a computationally intensive process that is mostly run on CPUs. Consider upgrading the CPU cores and or RAM on your data processing device for enhanced speed. If this is not an option, you can consider limiting the number of electrodes that you include in your configuration. Alternatively, you can make the parameters for detecting preliminary propagation sequences more stringent.

This reduces the total number of detected units, leaving only the most reliable units. As a result, the processing time will decrease. Specifically, we recommend to first try lowering the number of electrodes considered for splitting that do not lead to a split before terminating the splitting process.

Apply RT-Sort on recordings with lower electrode density. Increase inner and outer electrode radius. Note that lower electrode densities will make it increasingly more unlikely for an action potential to be detected on a sufficient number of electrodes. The user might consider lowering the minimum number of inner loose electrode detections and loose electrode detections but this makes RT-Sort more susceptible to noise detections.

Apply RT-Sort on recordings with higher electrode density. Continuous improvements are made with regards to fitting an increasing number of electrodes onto small recording devices at high electrode densities [2]. When using hardware with a substantially higher electrode density that the MaxWell MEAs and the Neuropixels used as examples in this work, decreasing the inner and outer electrode radius and increasing the minimum number of inner loose electrode detections and loose electrode detections will likely lead to a further increase in spike sorting performance by RT-Sort.

Besides the scenarios described above, sub-optimal performance might be a result of utilizing a spike detection model that was not specifically trained for your biological sample type—recording device combination (S7D Fig). If so, you can train a new spike detection model using the public RT-Sort Python library which only requires a single previously made recording (multiple recordings is recommended) from the same biological sample type and recording.

The subsequent sections provide a description of the detection and merging of action potential propagation sequences and the sorting of detected spikes, for added context about the meaning of the different parameters.

Patch-MEA ground truth recordings

To test the performance of RT-Sort, a ground truth dataset was generated by recording from a primary rat culture plated on a MaxWell MEA while simultaneously, a patch clamp recording was obtained from a single neuron. Spiking activity of the patched neuron was recorded in cell-attached mode or whole-cell current clamp and was also detected by the MEA. The patch-MEA dataset from [40] was used, as well as an additional dataset, generated largely according to the same protocol using only the whole-cell current-clamp mode and internal solution for recording. Cell culturing protocols are identical to those described in [40]. Both datasets contained 2 individually patched neurons. One of the two datasets contained an additional 5-minute pre-recording that was made using the MEA only, which allowed for the detection of propagation sequences in order for RT-Sort to be run in online mode on the patch-MEA recording. This pre-recording occurred within 60 minutes before the simultaneous patch-MEA recordings.

The simultaneous patch-MEA recordings have durations ranging from 40 seconds to over 3 minutes. By replaying these recordings, a real time application of RT-Sort was mimicked. Intracellular voltage peaks above 4x RMS of the patch recordings are considered to be spike events. RT-Sort spike detections are compared to these patch spike events by computing an overlap score for each of the 4 patched neurons.

Overlap score

Following [15], when two spike trains (A and B) are compared to see if they detect a signal from the same neuron, an overlap score is computed as follows:

Overlap = #matches / (#A + #B—#matches)

Here #matches are the total number of spikes from the two spike trains that are detected within 0.4ms of each other, #A are the total number of detected spikes in spike train A and #B are the total number of detected spikes in spike train B. In the case of the patch-MEA ground truth recordings as described in Methods: Patch-MEA ground truth recordings, the patch spikes are considered to be spike train A while the RT-Sort spikes are considered to be spike train B. In the case of the sorter comparison on the Neuropixels recording as described in Methods: Sorter comparison on *in vivo* Neuropixels recording, the overlap score is used for all pairwise comparisons between sorters.

Simulated Neuropixels ground truth recording

Genuine ground truth recordings obtained by combining extracellular field recordings and patch clamp recordings are hard to obtain *in vitro*, and even harder *in vivo* [11, 40]. This yields a low number of ground truth units while these units are often biased to be larger in size and amplitude and higher in their firing rates. As a result, these units are relatively easy to sort and might not form a representative sample of all neurons that are part of a large-scale recording.

As a result, we also tested RT-Sort on a simulated Neuropixels recording. This simulated ground truth dataset consisted of 250 biophysically detailed neurons that each exhibit independent Poisson firing of spikes with added Gaussian noise. See [15] for a full description of the generation of the dataset. RT-Sort was first tested on this simulated recording using sequence metrics generated from the known ground-truth spikes times to focus the performance assessment on spike assignment to propagation sequences in online mode. Next, RT-Sort was used in offline mode on the first 5 minutes of the simulated recording to also detect the propagation sequences which were then used to test RT-Sort again in online mode on the second 5 minutes of the simulated recording.

Sorter comparison on *in vivo* Neuropixels recording

In addition to a simulated Neuropixels recording, RT-Sort is tested on a genuine Neuropixels recording without ground truth. The recording was part of a larger dataset made from a head-fixed mouse by the Allen Institute for Brain Science ([16]; Allen Institute for Brain Science, 2019; dataset ID: 766640955; probe ID: 77359232). We use the first 5 minutes of the recording to detect sequences as described in Methods: Propagation sequence detection and following sections. We then assign spikes to these detected sequences using the subsequent 5 minutes of the recording as described in Methods: Real time spike sorting. Other recordings from the same dataset are used to generate training data for our detection model as described in Methods: Spike detection semi-artificial data generation.

Neuropixels have a sampling frequency of 30kHz and can record from up to 384 channels simultaneously. This specific recording used a checkerboard configuration (S9 Fig) and yielded 246 active recording channels (the remainder of the channels were either not inserted in the brain tissue or had a firing rate below 0.1Hz). The probe records from part of the cortex (V1), the hippocampus (CA1), the dentate gyrus and the thalamus (LP). During the experiment, the mouse was presented with a variety of visual stimuli while freely running on a rotating disk (for more details see ref. [16]).

As shown in the work from Buccino et al. [15], applying 6 different conventionally used spike sorters (HDSort, HerdingSpikes2, IronClust, Kilosort2, IronClust, SpyKING Circus and Tridesclous [4–9]) to this recording yielded wildly varying results with limited overlap between the different sorters. However, without a ground truth, it is not possible to say which results are correct. We hypothesize that the low overlap between units could be due to false spike detections of different sorters which could make the spike trains of different sorters so different that they are not recognized as coming from the same unit. This would mean that good performance of a certain sorter could potentially be obscured by bad performance of a different sorter.

For that reason, in comparing RT-Sort to each of the 6 other spike sorters, we focused on the units detected by both RT-Sort and the other sorter (overlap score $\geq$0.1). For each of these units, a fraction of the spikes was detected by both RT-Sort and the other sorter while both RT-Sort and the other sorter also detected spikes that were not co-detected. For each unit that had at least 10 spikes in each of these 3 groups, the interelectrode intervals and amplitudes of the not co-detected spikes from each sorter were compared to the mean of the interelectrode interval and amplitude distributions over all co-detected spikes.

Specifically, for each of the loose electrodes except the root electrode, the interelectrode intervals relative to the root electrode and the amplitudes were obtained over every spike in each of the three groups. Next, the absolute difference in the interelectrode interval and the percent difference in the amplitude were computed for each of the spikes, in comparison to the distribution mean over all spikes that were detected by both sorters. The differences were then pooled over all loose electrodes which yielded three absolute difference distributions for the interelectrode intervals and three percent difference distributions for the amplitudes per unit. 1 distribution consisted of the differences for the co-detected spikes, 1 distribution consisted of the differences for the RT-Sort only spikes and 1 distribution consisted of the differences for the other sorter only spikes.

The mean of the RT-Sort only distribution and the mean of the other sorter only distribution were taken as a measure for how different these spikes were relative to the co-detected spikes. The mean of the distribution over all spikes that were detected by both sorters was taken as a measure for the intrinsic variability within all co-detected spikes. The mean over all co-detected spikes was then subtracted from both the mean over all RT-Sort only spikes and the mean over all other sorter only spikes to get a difference score for the RT-Sort only spikes and a difference score for the other sorter only spikes per unit. An individual score was computed for the interelectrode interval absolute differences and for the amplitude percent differences. Due to the normalization achieved by subtracting the mean of the differences of the co-detected spikes, a difference score of 0 indicates that the spikes in the distribution of either RT-Sort only spikes or other sorter only spikes were equally variable in their interelectrode intervals or amplitudes as the co-detected spikes. Meanwhile, difference scores larger than 0 indicate that the spikes in the distribution of either RT-Sort only spikes or other sorter only spikes were more variable in their interelectrode intervals or amplitudes as the co-detected spikes.

As a final comparison between RT-Sort and the other sorter, a paired t-test was performed to see whether either RT-Sort only spikes or other sorter only spikes were more different from the co-detected spikes, using the difference scores per unit. This t-test was computed separately for both interelectrode interval differences and amplitude differences.

Supporting information

S1 Fig. Action potential detection with high temporal accuracy using a convolutional neural network (CNN). A) Architecture of the CNN used for spike detection in 30kHz Neuropixels recordings. The input layer consists of 300 nodes, corresponding to a 10ms window. The output layer consists of 180 nodes corresponding to 2-8ms in the input window. Each output node provides a score between 0 and 100 indicating the likelihood that the corresponding input frame contains a waveform trough. The 4 convolutional layers have a kernel size of 31 frames and a stride of 1 frame so that each output node makes a prediction based on the signal

in 2ms before until 2ms after the corresponding frame in the input. **B)** Example of averaged waveform footprint detected by Kilosort2. The purple traces were selected for training/validating the CNN. **C)** Example of training/validating sample creation and model prediction. i: A 10ms sample of recording specific noise is taken. ii: A waveform shape is selected from the training or validating pool. iii: The waveform is pasted into the recording device specific noise with ground truth certainty about the trough location (marked with yellow dotted line). iv: CNN detection model predictions for the frames in 2-8ms of the input window show a narrow detection peak at the waveform trough. All figures share the same x-axis. **D)** Same as C but with multiple overlapping waveforms in the same sample. **E)** Training (purple) and validating (cyan) loss as a function of training epoch. Error bars indicate the STD over the different cross-validation folds. The small differences between the training loss and validating loss indicate that the models is not overfitting nor underfitting the data. **F)** Recall when validating the detection model on samples generated from the held-out recording and when applying a 5RMS threshold to the same samples. The markers indicate the results for each of the 6 held-out datasets and the bar reflects the mean over all held-out datasets. Mean±STD for CNN = 84.7%±5.81% and for 5RMS = 65.1%±3.37%. The detection model has a significantly higher recall (P = 5.19*10–5, two-sided paired t-test, n = 6). **G)** Precision when validating the detection model on samples generated from the held-out recording and when applying a 5RMS threshold to the same samples. The markers indicate the results for each of the 6 held-out datasets and the bar reflects the mean over all held-out datasets. Mean±STD for CNN = 86.4%±7.90% and for 5RMS = 79.7%±4.60%. The difference in precision is not significant (P = 0.15, two-sided paired t-test, n = 6). **H)** Amplitude distribution of detected (purple) and missed (cyan) spikes by the detection model shows a good detection performance for spikes below 5RMS. Amplitudes are expressed as RMS relative to the surrounding 50ms of signal. **I)** F1 score, precision and recall of the detection model as a function of the detection threshold. The loose and stringent detection thresholds are marked with L and S respectively. **J)** The absolute deviation between the model detections and the ground truth trough times shows a high temporal accuracy with on average a deviation of 14μs (0.28 frames). Inset contains the same distribution on a log scale.
(TIF)

S2 Fig. Gradient ascent forms genuine looking waveform shapes from recording noise. A) (top) Intrinsic noise from an MEA electrode. (middle) Same piece of intrinsic noise after using the MEA spike detection model to perform gradient ascent on the noise to generate a waveform shape that leads to a detection with 100% certainty at the center of the noise sample. (bottom) Prediction scores for the noise sample after completing gradient ascent. **B)** Same as A but for Neuropixels noise using the Neuropixels spike detection model.
(TIF)

S3 Fig. Precision and recall for Kilosort2 detections. A) Precision and recall of CNN spike detection model using the ith highest amplitude electrode for each Kilosort2 detection as ground truth. The Kilosort2 detections for 6 organoid MEA recordings are grouped together and the spike detection model is used to detect spikes on the highest amplitude electrode (1 on x-axis) down to the 6th highest amplitude electrode (6 on x-axis). The line indicates the mean and the error bars the standard deviation over all units from all different recordings. **B)** Same as A but for the 6 mouse in vivo Neuropixels recordings. **C)** Recall distribution over all highest amplitude electrodes from the Neuropixels recordings. **D)** Recall distribution over all the 4th highest amplitude electrodes from the Neuropixels recordings.
(TIF)

S4 Fig. Examples of detected units with varying spike waveforms. A) (left) Averaged waveform footprint of unit recorded with patch-clamp and MEA and detected by RT-Sort. (right) Corresponding averaged spike detection model footprint. Lines, colors and markers have the same meaning as Fig 2A, 2B. **B)** Examples of single spikes from the detected unit in A using RT-Sort in online mode. Color scale is the same as in A. Lines, colors and markers have the same meaning as Fig 2D. **C)** (left) Waveform footprint of unit recorded with Neuropixels probe and detected by RT-Sort. (right) corresponding averaged spike detection footprint over all detected action potentials. Lines, colors and markers have the same meaning as Fig 2A, 2B. **D)** Examples of single spikes from the detected unit in C. Color scale and markers are the same as in panel C.
(TIF)

S5 Fig. Examples of detected spikes with overlapping waveforms. A) Additional examples of overlapping waveform spikes from the unit in Fig 2A that are correctly detected by RT-Sort. Lines, colors and markers have the same meaning as in Fig 2D. **B)** (left) Averaged waveform footprint of unit recorded with Neuropixels probe and detected by RT-Sort. (right) Corresponding averaged spike detection model footprint. Lines, colors and markers have the same meaning as Fig 2A, 2B. **C)** Examples of individual overlapping waveform spikes from the unit in panel B. Color scale and markers are the same as in B.
(TIF)

S6 Fig. Examples of detected low amplitude units. A) (top) Averaged waveform footprint of unit recorded with MEA and detected by RT-Sort. All electrodes record the unit below 5 times the signal to noise ratio, marked per electrode with the dotted red line. (bottom) Corresponding averaged CNN detection footprint. 4 electrodes detect the action potential above the loose detection threshold marked per electrode with the bottom dotted red line. Lines, colors and markers have the same meaning as Fig 2A, 2B. **B)** (left) Averaged waveform footprint of unit recorded with Neuropixels and detected by RT-Sort. All electrodes record the unit below 5 times the signal to noise ratio, marked per electrode with the dotted red line. (right) Corresponding averaged spike detection model footprint. 10 electrodes detect the action potential above the loose detection threshold marked per electrode with the bottom dotted red line. Lines, colors and markers have the same meaning as Fig 2A, 2B.
(TIF)

S7 Fig. Recall on simulated ground truth recording. A) Top: for each unit detected in the simulated ground truth recording using sequence metrics generated based on the ground truth spike locations, the recall over all detected spikes compared to the most similar ground truth neuron. Mean±STD = 70.9%±25.0%. Bottom: for each unit shown in the histogram at the top of the panel, the corresponding average waveform amplitude expressed in SNR. Top and bottom share the same x-axis. **B)** Top: for each unit detected in the simulated ground truth recording, the recall over all detected spikes compared to the most similar ground truth neuron. Mean±STD = 60.4%±25.9%. Bottom: for each unit shown in the histogram at the top of the panel, the corresponding average waveform amplitude expressed in SNR. Top and bottom share the same x-axis. **C)** Examples of averaged waveform footprints from units from the simulated ground truth recording with unrealistic waveform shapes that did not get detected by RT-Sort (overlap score <0.01). Lines, colors and markers have the same meaning as Fig 2D. **D)** Precision and recall when applying the MEA spike detection model on the simulated Neuropixels recordings using either the Neuropixels parameters (middle) or the MEA parameters (right). For comparison, the results of applying the Neuropixels model as presented in Fig 2I and S7A Fig are included on the left.
(TIF)

S8 Fig. Detection statistics for a mouse *in vivo* Neuropixels recording. A) Number of units detected by different spike sorting algorithms in the recording for Fig 4. Abbreviations: RT = RT-Sort, HS = Herdingspikes2, HDS = HD-Sort, IC = IronClust, KS = Kilosort2, SC = SpyKing Circus, TDC = Tridesclous. **B)** Distribution of propagation lengths for all detected RT-Sort units in the same recording as panel A. **C)** Distribution of all propagation durations for all detected RT-Sort units in the same recording as panel A.
(TIF)

S9 Fig. Schematic illustration of spike detection and sequence detection. A) An action potential traveling down an axon and recorded by different electrodes (represented as gray squares) can yield varying detected waveform shapes on each electrode. The CNN spike detection model can identify these waveforms as spikes and provides a highly sensitive estimation of the waveform trough detection time on each electrode (the detection model output on the right covers 0.5ms whereas the waveform shapes in the middle cover 5ms). Using these CNN spike detection model outputs, the action potential propagation duration can be measured for the propagating action potential. **B)** Example of an action potential propagation measured by 4 electrodes. In this example, the root electrode, marked with a red star, detects 1000 spikes. Subsequent electrodes that co-detect a spike within 0.5ms of a spike detected on the root electrode are used as splitting electrodes in order of increasing distance. The nearest electrode is used as the first splitting electrode to assess whether the current subset of root detections are composed of spikes from multiple neurons; in this example, they are not. The subset of root detections that coincide with spikes on the splitting electrode are considered to be the new selection of root detections when the next closest electrode is assessed. **C)** Similar as B but in this example, two different neurons share the same root electrode. Root detections that are considered to be part of the propagation sequence of one neuron can not also be part of the propagation sequence of the other neuron. **D)** Similar as C but in this case the neurons diverge only after having shared a first splitting electrode after the root electrode. For the subsequent electrodes, detections that are considered to be part of the propagation sequence of one neuron can not also be part of the propagation sequence of the other neuron.
(TIF)

S10 Fig. Spike sorting performance dependence on pre-recording duration. Bottom: The number of detected sequences and the minimum firing rate over all detected sequences as a function of the duration of the pre-recording in which the sequences were detected. Note that the x-axis uses a non-linear scale. Middle: For each pre-recording duration plotted in the bottom (shared x-axis), the distribution of the recall scores of the detected units when tested on a separate 2-minute part of the recording, replayed as to mimic a real time sorting. The red line reflects the average recall over all sequences detected in the pre-recording. Top: Same as middle but for precision instead of recall. Overall, the number of detected sequences increases with longer pre-recording durations. This is due to more low firing rate units being detected. Some sequences with below 1Hz firing rates can already be detected in pre-recordings of 40s. Additional units are being detected with longer pre-recording durations and this plateaus around pre-recordings of 5min. Note that the average precision and recall over all detected sequences remains consistent, independent of the pre-recording duration.
(TIF)

S11 Fig. Electrode locations justify the 50μm distance threshold. A) Schematic representation of electrode locations on an MEA where every other electrode is selected in the configuration (resulting in 35μm pitch, grey electrodes are not included in the configuration). Within a 50μm radius of the electrode in red, the closest electrodes directly above and besides the red

electrode are selected, as well as the closest electrodes diagonally spaced relative to the red electrode. **B)** Schematic representation of electrode locations on a Neuropixels probe with checkerboard configuration (grey electrodes are not included in the configuration). Within a 50μm radius of the electrode in red, the closest electrodes above and besides the red electrode are selected, as well as the closest electrodes diagonally spaced relative to the red electrode.
(TIF)

S12 Fig. Spike cluster splitting based on amplitude and interelectrode interval distribution. A) Example of an interelectrode interval distribution for all splitting codetections of a contaminated loose electrode on the Neuropixels recording of Fig 4. The bimodal distribution reflects a propagation from two different axons. The decision boundary of the 2-component Gaussian mixture model splits the interelectrode intervals into the two separate preliminary propagations sequences. The blue and orange curves show the probability density function of the two clusters. **B)** Example of an amplitude distribution for all amplitudes on the root electrode of a contaminated preliminary propagation sequence on the Neuropixels recording of Fig 4. The bimodal distribution (Hartigan's dip test, P = 0.0005) reflects a propagation from two different axons. The decision boundary of the 2-component Gaussian mixture model splits the interelectrode intervals into the two separate preliminary propagations sequences. The blue and orange curves show the probability density function of the two clusters. **C)** Distribution of the Hartigan's dip test P-values used for splitting the preliminary propagation sequences based on the amplitude of the root electrode in the Neuropixels recording of Fig 4. A distribution is considered multimodal if P<0.1.
(TIF)

Acknowledgments

The authors would like to thank Alessio Buccino for sharing the results from applying the 6 different spike sorters to the Neuropixels recording used in Fig 4 and for answering various of our questions throughout the development of RT-Sort. We would also like to thank Samantha Hermans for their work on related analyses and sharing their results.

Author Contributions

Conceptualization: Tjitse van der Molen, Kenneth R. Tovar, Kenneth S. Kosik.

Data curation: Tjitse van der Molen, Max Lim, Julian Bartram.

Formal analysis: Tjitse van der Molen, Max Lim.

Methodology: Tjitse van der Molen, Max Lim, Julian Bartram, Zhuowei Cheng, Ash Robbins, David F. Parks, Paul K. Hansma, Kenneth R. Tovar, Kenneth S. Kosik.

Project administration: Tjitse van der Molen.

Resources: Tjitse van der Molen.

Software: Tjitse van der Molen, Max Lim, Zhuowei Cheng.

Supervision: Tjitse van der Molen, Linda R. Petzold, Andreas Hierlemann, David Haussler, Paul K. Hansma, Kenneth R. Tovar, Kenneth S. Kosik.

Validation: Tjitse van der Molen.

Visualization: Tjitse van der Molen, Max Lim.

Writing – original draft: Tjitse van der Molen.

Writing – review & editing: Ash Robbins, David F. Parks, Linda R. Petzold, Andreas Hierlemann, David Haussler, Paul K. Hansma, Kenneth R. Tovar, Kenneth S. Kosik.

References

1. Jun J. J., Steinmetz N. A., Siegle J. H., Denman D. J., Bauza M., Barbarits B., et al. (2017). Fully integrated silicon probes for high-density recording of neural activity. Nature, 551(7679), 232–236. https://doi.org/10.1038/nature24636 PMID: 29120427
2. Ye Z., Shelton A. M., Shaker J. R., Boussard J., Colonell J., Manavi S., et al. (2023). Ultra-high density electrodes improve detection, yield, and cell type specificity of brain recordings. bioRxiv.
3. Ballini M., Müller J., Livi P., Chen Y., Frey U., Stettler A., et al. (2014). A 1024-channel CMOS microelectrode array with 26,400 electrodes for recording and stimulation of electrogenic cells in vitro. IEEE journal of solid-state circuits, 49(11), 2705–2719. https://doi.org/10.1109/JSSC.2014.2359219 PMID: 28502989
4. Hilgen G., Sorbaro M., Pirmoradian S., Muthmann J. O., Kepiro I. E., Ullo S., et al. (2017). Unsupervised spike sorting for large-scale, high-density multielectrode arrays. Cell reports, 18(10), 2521–2532. https://doi.org/10.1016/j.celrep.2017.02.038 PMID: 28273464
5. Pachitariu, M., Steinmetz, N., Colonell, J., (2018) Kilosort2. https://github.com/MouseLand/Kilosort2.
6. Jun J. J., Mitelut C., Lai C., Gratiy S. L., Anastassiou C. A., & Harris T. D. (2017b). Real-time spike sorting platform for high-density extracellular probes with ground-truth validation and drift correction. BioRxiv, 101030.
7. Yger P., Spampinato G. L., Esposito E., Lefebvre B., Deny S., Gardella C., et al. (2018). A spike sorting toolbox for up to thousands of electrodes validated with ground truth recordings in vitro and in vivo. Elife, 7, e34518. https://doi.org/10.7554/eLife.34518 PMID: 29557782
8. Garcia S. & Pouzat C. (2015) Tridesclous. https://github.com/tridesclous/tridesclous
9. Diggelmann R., Fiscella M., Hierlemann A., & Franke F. (2018). Automatic spike sorting for high-density microelectrode arrays. Journal of neurophysiology, 120(6), 3155–3171. https://doi.org/10.1152/jn.00803.2017 PMID: 30207864
10. Buccino A. P., Garcia S., & Yger P. (2022). Spike sorting: new trends and challenges of the era of high-density probes. Progress in Biomedical Engineering, 4(2), 022005.
11. Steinmetz N. A., Koch C., Harris K. D., & Carandini M. (2018). Challenges and opportunities for large-scale electrophysiology with Neuropixels probes. Current opinion in neurobiology, 50, 92–100. https://doi.org/10.1016/j.conb.2018.01.009 PMID: 29444488
12. Erickson C. A., Jung M. W., McNaughton B. L., & Barnes C. A. (1996). Contribution of single-unit spike waveform changes to temperature-induced alterations in hippocampal population spikes. Experimental brain research, 107, 348–360. https://doi.org/10.1007/BF00230417 PMID: 8821377
13. Bean B. P. (2007). The action potential in mammalian central neurons. Nature Reviews Neuroscience, 8(6), 451–465. https://doi.org/10.1038/nrn2148 PMID: 17514198
14. Stratton P., Cheung A., Wiles J., Kiyatkin E., Sah P., & Windels F. (2012). Action potential waveform variability limits multi-unit separation in freely behaving rats. PloS one, 7(6), e38482. https://doi.org/10.1371/journal.pone.0038482 PMID: 22719894
15. Buccino A. P., Hurwitz C. L., Garcia S., Magland J., Siegle J. H., Hurwitz R., et al. (2020). SpikeInterface, a unified framework for spike sorting. Elife, 9, e61834. https://doi.org/10.7554/eLife.61834 PMID: 33170122
16. Siegle J. H., Jia X., Durand S., Gale S., Bennett C., Graddis N., et al. (2021). Survey of spiking in the mouse visual system reveals functional hierarchy. Nature, 592(7852), 86–92. https://doi.org/10.1038/s41586-020-03171-x PMID: 33473216
17. Garcia S., Buccino A. P., & Yger P. (2022). How do spike collisions affect spike sorting performance?. Eneuro, 9(5). https://doi.org/10.1523/ENEURO.0105-22.2022 PMID: 36171060
18. Chung J. E., Magland J. F., Barnett A. H., Tolosa V. M., Tooker A. C., Lee K. Y., et al. (2017). A fully automated approach to spike sorting. Neuron, 95(6), 1381–1394. https://doi.org/10.1016/j.neuron.2017.08.030 PMID: 28910621
19. Rodriguez A., & Laio A. (2014). Clustering by fast search and find of density peaks. science, 344 (6191), 1492–1496.
20. Wagenaar D. A., Madhavan R., Pine J., & Potter S. M. (2005). Controlling bursting in cortical cultures with closed-loop multi-electrode stimulation. Journal of Neuroscience, 25(3), 680–688. https://doi.org/10.1523/JNEUROSCI.4209-04.2005 PMID: 15659605

21. DeMarse T. B., Wagenaar D. A., Blau A. W., & Potter S. M. (2001). The neurally controlled animat: biological brains acting with simulated bodies. Autonomous robots, 11, 305–310. https://doi.org/10.1023/a:1012407611130 PMID: 18584059
22. Kagan B. J., Kitchen A. C., Tran N. T., Habibollahi F., Khajehnejad M., Parker B. J., et al. (2022). In vitro neurons learn and exhibit sentience when embodied in a simulated game-world. Neuron, 110(23), 3952–3969.
23. Bakkum, D. J., Chao, Z. C., & Potter, S. M. (2007). Adaptive goal-directed behavior in embodied cultured networks: living neuronal networks and a simulated model. In 2007 3rd International IEEE/EMBS Conference on Neural Engineering (pp. 46–49). IEEE.
24. Patil P. G., & Turner D. A. (2008). The development of brain-machine interface neuroprosthetic devices. Neurotherapeutics, 5, 137–146. https://doi.org/10.1016/j.nurt.2007.11.002 PMID: 18164493
25. Arsiero M., Lüscher H. R., & Giugliano M. (2007). Real-time closed-loop electrophysiology: towards new frontiers in in vitro investigations in the neurosciences. Archives italiennes de biologie, 145(3), 193–209. PMID: 18075116
26. Franke F., Jäckel D., Dragas J., Müller J., Radivojevic M., Bakkum D.,et al. (2012). High-density microelectrode array recordings and real-time spike sorting for closed-loop experiments: an emerging technology to study neural plasticity. Frontiers in neural circuits, 6, 105. https://doi.org/10.3389/fncir.2012.00105 PMID: 23267316
27. Koch C., Rapp M., & Segev I. (1996). A brief history of time (constants). Cerebral cortex, 6(2), 93–101. https://doi.org/10.1093/cercor/6.2.93 PMID: 8670642
28. Franke F., Natora M., Boucsein C., Munk M. H., & Obermayer K. (2010). An online spike detection and spike classification algorithm capable of instantaneous resolution of overlapping spikes. Journal of computational neuroscience, 29, 127–148. https://doi.org/10.1007/s10827-009-0163-5 PMID: 19499318
29. Jäckel D., Frey U., Fiscella M., Franke F., & Hierlemann A. (2012). Applicability of independent component analysis on high-density microelectrode array recordings. Journal of neurophysiology, 108(1), 334–348. https://doi.org/10.1152/jn.01106.2011 PMID: 22490552
30. Li Z., Wang Y., Zhang N., & Li X. (2020). An accurate and robust method for spike sorting based on convolutional neural networks. Brain Sciences, 10(11), 835. https://doi.org/10.3390/brainsci10110835 PMID: 33187098
31. Rácz M., Liber C., Németh E., Fiáth R., Rokai J., Harmati I., et al. (2020). Spike detection and sorting with deep learning. Journal of neural engineering, 17(1), 016038. https://doi.org/10.1088/1741-2552/ab4896 PMID: 31561235
32. Pachitariu M., Steinmetz N., Kadir S., Carandini M., & Kenneth D H. (2016). Kilosort: realtime spike-sorting for extracellular electrophysiology with hundreds of channels. BioRxiv, 061481.
33. Tovar K. R., Bridges D. C., Wu B., Randall C., Audouard M., Jang J., et al. (2018). Action potential propagation recorded from single axonal arbors using multielectrode arrays. Journal of neurophysiology, 120(7), 306–320. https://doi.org/10.1152/jn.00659.2017 PMID: 29641308
34. Jia X., Siegle J. H., Bennett C., Gale S. D., Denman D. J., Koch C., et al. (2019). High-density extracellular probes reveal dendritic backpropagation and facilitate neuron classification. Journal of neurophysiology, 121(5), 1831–1847. https://doi.org/10.1152/jn.00680.2018 PMID: 30840526
35. Parks D., Prochaska J. X., Dong S., & Cai Z. (2018). Deep learning of quasar spectra to discover and characterize damped Lyα systems. Monthly Notices of the Royal Astronomical Society, 476(1), 1151–1168.
36. Windolf C., Yu H., Paulk A. C., Meszéna D., Muñoz W., Boussard J., et al (2023). DREDge: robust motion correction for high-density extracellular recordings across species. BioRxiv. https://doi.org/10.1101/2023.10.24.563768 PMID: 37961359
37. Sharf T., van der Molen T., Glasauer S. M., Guzman E., Buccino A. P., Luna G., et al. (2022). Functional neuronal circuitry and oscillatory dynamics in human brain organoids. Nature communications, 13(1), 4403. https://doi.org/10.1038/s41467-022-32115-4 PMID: 35906223
38. Donoho D. L., & Johnstone I. M. (1994). Ideal spatial adaptation by wavelet shrinkage. biometrika, 81 (3), 425–455.
39. Quiroga R. Q., Nadasdy Z., & Ben-Shaul Y. (2004). Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. Neural computation, 16(8), 1661–1687. https://doi.org/10.1162/089976604774201631 PMID: 15228749
40. Bartram J., Franke F., Kumar S. S., Buccino A. P., Xue X., Gänswein T., et al. (2023). Parallel reconstruction of the excitatory and inhibitory inputs received by single neurons reveals the synaptic basis of recurrent spiking. bioRxiv, 2023–01.