A Scalable 1024-Channel Ultra-Low-Power Spike Sorting Chip With Event-Driven Detection and Spatial Clustering

Akhoundi, Arash; Yan, Pumiao; Landbrug, Yawende; Hays, Madeline; Murmann, Boris; Chichilnisky, E. J.; Muratore, Dante G.

DOI
10.1109/JSSC.2025.3611139

Publication date
2025

Document Version
Final published version

Published in
IEEE Journal of Solid-State Circuits

A Scalable 1024-Channel Ultra-Low-Power Spike Sorting Chip With Event-Driven Detection and Spatial Clustering

Arash Akhoundi, *Graduate Student Member, IEEE*, Pumiao Yan, *Member, IEEE*, Yawende Landbrug, *Student Member, IEEE*, Madeline Hays, Boris Murmann, *Fellow, IEEE*, E. J. Chichilnisky, and Dante G. Muratore, *Senior Member, IEEE*

***Abstract*—This article presents a 1024-channel ultra-low-power spike sorting chip featuring event-driven spike detection and spatial clustering for large-scale neural recording. To address power and scalability constraints in brain–computer interfaces (BCIs), the design integrates a compressive analog-to-digital converter (ADC) with a two-stage spike detector that significantly reduces memory and processing activity. Spatial features derived from high-density micro-electrode array (MEA) enhance cluster separability, enabling robust performance even under neural signal distortion or probe drift, particularly when recordings are obtained using planar MEAs. A modified self-organizing map (SOM) algorithm clusters spikes in the spatial domain with minimal memory access, supporting on-chip training and real-time operation with low latency. Fabricated in 40-nm CMOS, the chip achieves 0.00029-mm^2/channel area and 74-nW/channel power consumption, with over 1000× data compression. Performance is validated across synthetic and ex vivo datasets containing up to 500 neurons, demonstrating competitive accuracy and robust drift tracking compared to state-of-the-art solutions with much lower data bandwidth, processing, and power demands.**

***Index Terms*—Brain–computer interfaces (BCIs), event-driven spike detection, high-density neural interface, neural signal compression, neural signal processor (NSP), on-chip spike sorting.**

I. Introduction

BRAIN–COMPUTER interfaces (BCIs) that can record neural activity (*spikes*) have shown the ability to decode motor and speech intentions in humans for several therapeutic applications [1], [2], [3], [4], [5]. There is an ongoing demand in the field for larger channel count neural interfaces to further improve the performance and reliability of these devices, both for clinical and research applications [6], [7], [8], [9], [10], [11]. Furthermore, there is a need for these implantable devices to be wireless to reduce infection risks associated with tethered systems. At the same time, their power and area budget are extremely limited. Typically, wireless data transmission is the most power-hungry component and limits the channel scalability [12]. Hence, there is a need to reduce the amount of data to be transmitted. Several on-chip compression techniques that exploit the sparsity in the neural signal have been proposed [13], [14], [15], [16], [17], [18], but the compression rate is limited to 10–100×, which is insufficient for the massive bandwidth demands of future BCIs. Further data reduction can be obtained by integrating part of the signal processing chain on-chip, but careful consideration must be given to the power trade-off. In other words, the power savings obtained in the wireless data transmission need to be larger than the power consumption of the on-chip signal processing.

Spike sorting is a common processing step in neural interfaces, where detected spikes are assigned to putative neurons (Fig. 1). This step is required to achieve single-cell resolution because one electrode can record spikes from multiple nearby neurons. Spike sorting is an effective method for achieving a

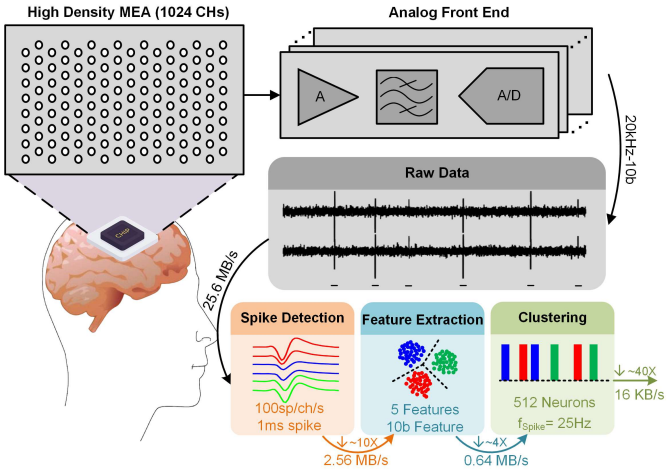


Fig. 1. Overview of the spike sorting pipeline for a 1024-channel MEA. An example of data reduction is shown at each step, demonstrating the progressive compression of neural data from raw recordings to final neuron indices.

Received 2 May 2025; revised 12 July 2025 and 6 September 2025; accepted 9 September 2025. Date of publication 1 October 2025; date of current version 29 October 2025. This article was approved by Associate Editor Kea-Tiong Tang. This work was supported in part by the Dutch Brain Interface Initiative (DBI2), part of the Gravitation Research Program, financed by the Dutch Ministry of Education, Culture and Science (OCW) via the Dutch Research Council (NWO), under Project 024.005.022; in part by the National Institutes of Health (NIH) Blueprint for Neuroscience Research and by the National Eye Institute and the National Institute of Biomedical Imaging and Bioengineering under Grant 3U54EB033650-02S1; and in part by the Wu Tsai Neurosciences Institute and NIH under Grant EY032900 and Grant EY021271. *(Corresponding author: Arash Akhoundi.)*

Arash Akhoundi, Yawende Landbrug, and Dante G. Muratore are with the Microelectronics Department, Delft University of Technology, CD 2628 Delft, The Netherlands (e-mail: a.akhoundi@tudelft.nl).

Pumiao Yan is with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA.

Madeline Hays is with the Department of Bioengineering, Stanford University, Stanford, CA 94305 USA.

Boris Murmann is with the Department of Electrical and Computer Engineering, University of Hawai'i at Manoa, Honolulu, HI 96822 USA.

E. J. Chichilnisky is with the Hansen Experimental Physics Laboratory, Department of Neurosurgery and Ophthalmology, Stanford University, Stanford, CA 94305 USA.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JSSC.2025.3611139.

Digital Object Identifier 10.1109/JSSC.2025.3611139

high data rate reduction since only the neuron ID needs to be transmitted when a spike occurs (which happens very rarely), if it can be implemented efficiently on-chip. In addition to data reduction, spike sorting can improve BCI decoding performance by isolating the activity of individual neurons, which may become necessary in the future for more demanding tasks such as full-body movement. Moreover, it supports drift compensation by tracking gradual changes in spike waveform or spatial footprint caused by electrode–tissue movement, thereby improving the long-term stability of neural recordings. Spike sorting and high-density recordings also enable cell-type identification [19], which can help tailor decoding strategies or support neuroscience studies targeting specific neuronal subtypes. The spike sorting process generally involves three main steps: spike detection, feature extraction, and clustering.

First, spikes are detected in the raw neural recordings using a threshold-based detector. To enhance the detectability of the spike signal-to-noise ratio (SNR), a pre-emphasizer is often applied before thresholding. Common pre-emphasizer methods include the nonlinear energy operator (NEO) [20], discrete wavelet transform (DWT) algorithms [21], and integer coefficient filters [22]. Second, after the spike is detected, important features are extracted from the spike data before clustering. This step is used to reduce the dimensionality of the input data and project it into a space in which spikes from different neurons are easier to differentiate. Typically, spike sorting relies on the distinct spike waveforms recorded from different neurons. Commonly used features extracted from these waveforms include principal components [23], DWT coefficients [24], first and second derivative extrema (FSDE) [25], zero crossing [26], *salient* features [27], and integer coefficient filters' response [22]. The final step defines clusters in the feature space to assign spikes to putative neurons. This process typically utilizes either supervised or unsupervised clustering algorithms. Supervised methods such as k-means [22], support vector machines [28], and template matching [29], [30] achieve high clustering accuracy but require prior information such as the number of neurons, firing rates, or labeled recordings. However, in online spike sorting applications where such prior information is unavailable, unsupervised algorithms are preferred, such as OSort [31], self-organizing map (SOM) [32], and hierarchical adaptive means [33].

In addition to conventional three-step spike sorting algorithms described above, neural networks (NNs) have also been investigated in recent years. The work in [34] uses a convolutional autoencoder and a modified version of k-means to improve clustering accuracy. Binarized NNs have also been explored in [35]. A Δ-based spike sorting system was proposed in [36], featuring an analog computing-in-memory (CIM) binary autoencoder NN (B-AENN) for feature extraction, combined with a digital implementation of *k*-means for clustering. While NN-based methods achieve high accuracy, the increased complexity and power consumption make them unsuitable for wireless implantable devices. Additionally, a significant latency is introduced by the necessary resource sharing in large channel count implementations, which is impractical for real-time applications.

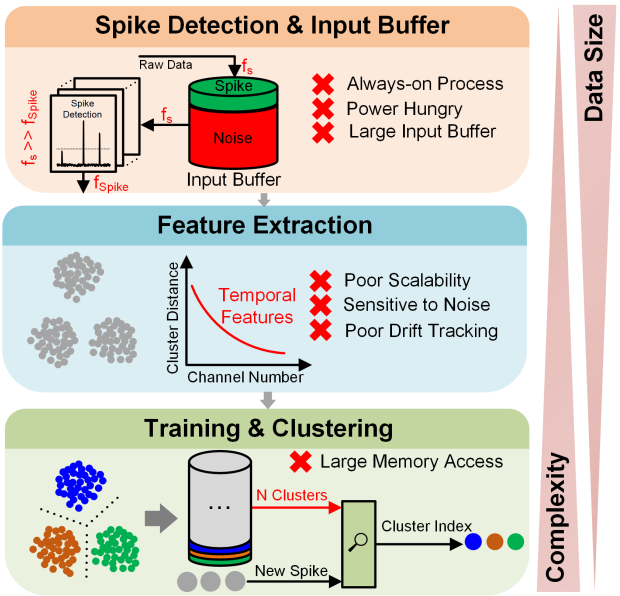


Fig. 2. Overview of the three primary challenges in on-chip spike sorting.

Despite recent advances, on-chip spike sorting continues to face fundamental challenges in scaling to tens of thousands of channels, particularly in implantable devices, where strict thermal constraints apply (see Fig. 2). A local heat-flux limit of approximately 40 mW/cm^2 corresponds to a total power budget of less than 4 mW for a cortical IC occupying no more than 0.1 cm^2 [37]. This constraint implies that the signal processing circuitry must consume less than 0.1 μW per channel in order to scale to large channel counts without exceeding safety limits. Substantial progress in each of these areas is necessary for spike sorting to become a viable solution in future neural interface technologies.

First, although spike detection is the simplest step in spike sorting, it typically consumes most of the system's power and area because it operates at the raw data rate (e.g., 200 Mbps for 1024 10-bit channels sampled at 20 kHz), while the rest of the system operates at the spike rate (e.g., 25 kHz for 1024 channels with an average spike rate per channel of 25 Hz [38]). For example, in [22], [39] and [40], the spike detector consumes significant power due to its continuous operation at the raw data, limiting its scalability to larger channel counts. This step also requires a large memory to buffer the raw input data and an always-on detection circuit, which impacts area and power efficiency.

Second, most prior on-chip spike sorters rely solely on temporal features for classification [22], [41], [42], [43], which do not capture useful spatial information in high-density micro-electrode arrays (MEAs) and do not scale well to large channel counts [23]. Unfortunately, because single-channel datasets [44] and datasets with few neurons [45] were used for benchmarking, the impact of discarding spatial information is not apparent in the prior art. In [47], the authors show that incorporating the waveforms from neighboring electrodes can improve clustering accuracy. However, such approaches increase dimensionality and memory requirements, which affect power and scalability. The geometry-aware OSort in [39] and [40] partially addresses spatial redundancy by discarding

duplicate spikes, but it still uses only temporal features from the primary electrode and misses valuable spatial context.

Third, typically, the proposed clustering algorithms such as k-means [22] and template matching [42] need to retrieve the entire large template or cluster memories to sort the incoming spikes, which reduce scalability to a large number of channels. Indeed, as the number of clusters and detected spikes scales with the number of recording channels, the complexity of the clustering step increases approximately with $O(n^2)$, where n is the number of recording channels.

To address these challenges and enable effective low-power, scalable on-chip spike sorting, this work is based on three key ideas.

1) A compressive analog-to-digital converter (ADC) [14] and a spike pre-detector move from always-on to event-driven spike detection and reduce the input memory size. This approach leverages the sparse spiking activity and eliminates useless noise samples directly during quantization, resulting in over an order of magnitude power savings in the spike detector.
2) Spatial features enhance cluster separability in large-scale, high-density MEAs, which record signals from hundreds of neurons, particularly in planar MEA recordings. These features are also more resilient to waveform distortions introduced by compressive ADCs, making them well-suited for modern MEA architectures.
3) A modified SOM clustering algorithm operates in the spatial feature space and forms a geometry-based cluster map. This approach minimizes memory read operations during clustering and facilitates stable cluster tracking despite MEA drift relative to the tissue.

This article extends our work in [47] and is organized as follows: Section II provides a detailed description of the three main components of the spatial spike sorting algorithm. The design rationale is presented with a focus on balancing computational complexity, power consumption, and silicon area. Section III presents a detailed description of the hardware development, covering the architecture, design methodology, and implementation details, along with timing diagrams to illustrate the system's operation. Section IV presents comprehensive simulation results, including accuracy benchmarking against existing spike sorting methods, as well as evaluations of compression rate, power consumption, electrical image (EI) extraction, and drift compensation performance. Section V discusses the trade-offs between spatial and temporal feature spaces and analyzes the scalability of the proposed chip. Section VI offers the conclusions. Finally, Appendix provides detailed information about the five datasets—both synthetic and real—used for evaluation.

II. System Innovations and Design Methodology

A. Compressive ADC and Spike Detection

Spike detection is the simplest step in the processing pipeline (e.g., NEO-based spike detection only requires two multiplications, one addition, and a small buffer to store two samples); however, it typically consumes the most power and area because it operates at the input sampling frequency and

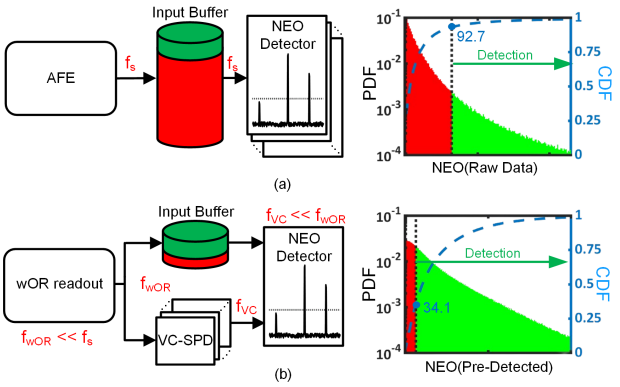


Fig. 3. Comparison of (a) conventional spike detector structure with all raw data stored in the input buffer and processed by NEO-based detector and (b) proposed event-driven spike detector structure with only wired-OR outputs stored in the input buffer and processed by the VC-SPD before the NEO-based detector. Probability density function (pdf) and cumulative density function (cdf) of the input to the NEO-based detector are reported on the right—subthreshold samples in red and suprathreshold samples in green.

is active across all available channels. Furthermore, due to latency in spike detection or the need to store spike samples for the feature extraction unit, a large input memory is required to buffer the incoming data from all channels. However, the typical firing rate of neurons is relatively low (generally 0.05–5 Hz and rarely exceeding 100 Hz) [48], [49], and even under strong or temporally modulated inputs, it rarely reaches 200 Hz [50], which is still much lower than the 20-kHz sampling frequency, resulting in sparse activity in the input signal. As a result, the input memory is mostly filled with noise samples, and the spike detector rarely processes spike samples. For example, in the ExVivo-1 dataset over a 5-s segment, only 7.3% of the input memory has NEO values higher than the detection threshold, indicating that they are related to spike samples [see Fig. 3(a)]. Consequently, most of the power is consumed in processing and storing noise samples. To make this analysis independent of the detection threshold calculation method, the NEO threshold is swept across a range of values, and the threshold yielding the highest detection accuracy is selected as the detection threshold.

To make spike detection more efficient, we employ a compressive ADC known as wired-OR readout [6], [14], [51] rather than using an independent ADC for each recording channel. This approach takes advantage of the sparsity of spikes within neural recordings, at the analog–digital interface, to eliminate noise samples during the ADC stage and largely reduce activity in the spike detector.

The working principle of this compressive ADC is that the digitized voltage of a given electrode is retained only if it differs from the values recorded on other electrodes. This is efficiently achieved using a ramp ADC combined with a wired-OR readout and a unique-signal decoder [Fig. 4(a)]. During each sampling period, electrode values are compared with a ramp with N quantization levels (Q-levels) to perform pulse-position modulation (PPM) on each pixel. The PPM outputs are combined with wired-OR logic across the row and columns of the MEA and read at each ramp step. For electrodes capturing a spike sample, the quantized value is typically unique, enabling the corresponding channel location

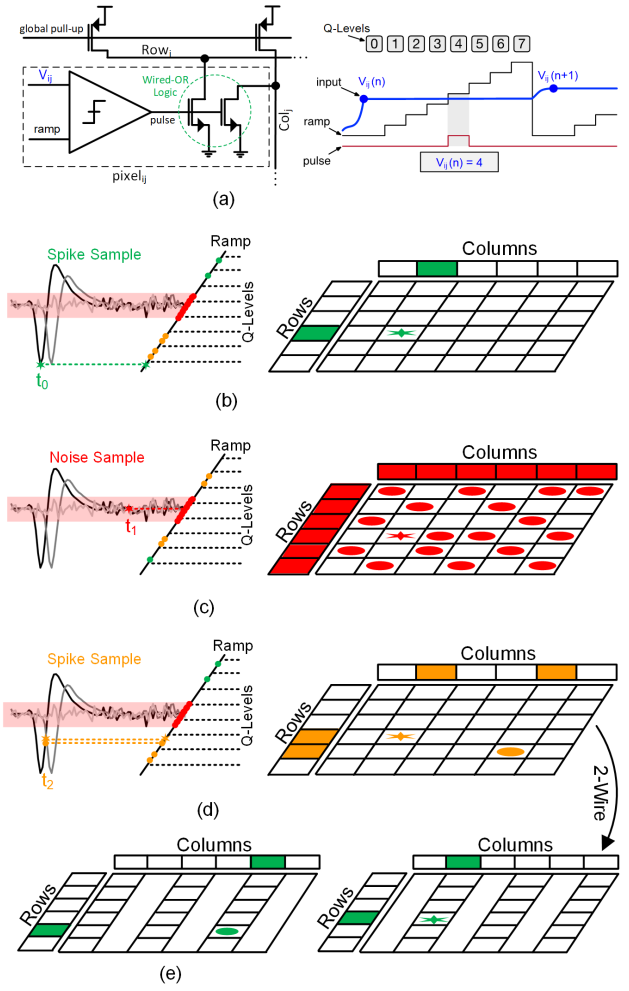


Fig. 4. Wired-OR concept. (a) Pixel schematic. (b) Collision-free scenario for a spike sample. (c) Massive collision scenario caused by noise samples. (d) Small collision scenario for spike samples, which can be resolved by using (e) two-wire configuration.

to be identified through a uniquely decoded row and column [collision-free sample, see Fig. 4(b)]. Collision-free samples are decoded and stored for further processing. Conversely, electrodes recording noise samples often produce redundant quantized values, activating multiple rows and/or columns at the same time without indicating a unique channel [massive collision, see Fig. 4(c)]. Collision samples are discarded, which leads to the desired compression.

Notably, the wired-OR readout is inherently a lossy compression method. While it effectively discards noise samples, it may occasionally result in collisions among spike samples, distorting the recorded signal. This distortion occurs when multiple spike events produce identical quantized values, preventing accurate decoding [small collision, see Fig. 4(d)]. To mitigate this distortion, the recording array can be divided into two, four, or more subarrays, reducing the likelihood of collisions at the cost of a lower compression rate. As illustrated in Fig. 4(e), the small collision scenario is resolved using a two-wire configuration, in which the array is divided into two subarrays.

In this work, we used a wired-OR readout with 8-bit amplitude resolution (256 ramp quantization levels) and a four-wire configuration, unless stated otherwise. In contrast to prior works, this design is the first to implement a fully integrated on-chip spike sorting pipeline using the wired-OR data-compressive readout in [6]. While [14] introduced the theoretical foundation of wired-OR readout, and [51] evaluated its compressive performance in software, neither included a hardware implementation or spike processing. Similarly, [6] implemented the wired-OR architecture in a 1024-channel neural recording IC but did not incorporate any on-chip spike processing.

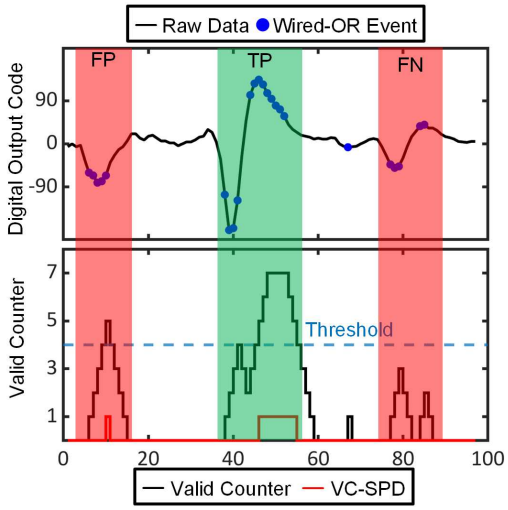


Fig. 5. Illustration of the VC-SPD algorithm.

Because the wired-OR readout does not generate a valid output for collision events, which are mostly related to noise samples, it is assumed that its output is valid only when a spike is being recorded. The valid counter spike pre detector (VC-SPD) leverages this to define a spike event as a time window containing more than a predetermined number of valid events in the output of the wired-OR readout. To evaluate the performance of the VC-SPD, accuracy and sensitivity are calculated using the following equations:

$$\text{Accuracy} = \frac{\text{TP}}{\text{TP} + \text{FN} + \text{FP}} \tag{1}$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{2}$$

Here, true positives (TPs) represent spikes correctly detected as spikes, false negatives (FNs) represent spikes that are not detected, and false positives (FPs) represent non-spikes detected as spikes. Fig. 5 illustrates various scenarios in VC-SPD, in which high-level noise is detected as a spike FP, a true spike is correctly detected as a spike TP, and a low-level spike is missed leading to an FN.

There is a trade-off between sensitivity and accuracy in VC-SPD. Increasing the number of wires in the wired-OR readout reduces collisions and increases sensitivity by decreasing the number of FNs, but it lowers detection accuracy due to a higher number of FPs. This trade-off is illustrated in Fig. 6 for five different datasets. The VC-SPD has a sensitivity near 100% for a higher number of wires but suffers from very low accuracy. However, the low accuracy is mostly due to false positives; hence, the critical information (TPs) is still

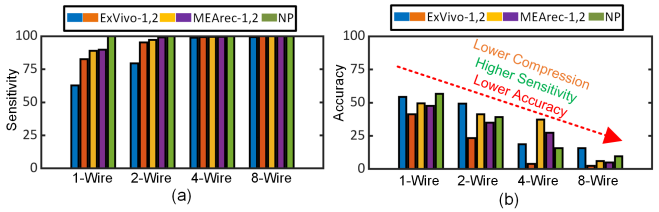


Fig. 6. Trade-off between (a) sensitivity and (b) accuracy for different wire configurations in wired-OR.

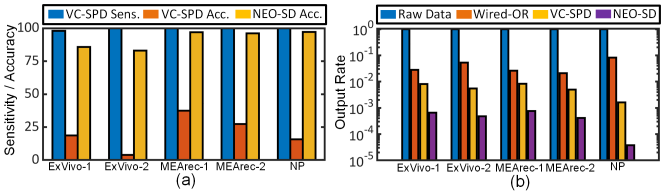


Fig. 7. (a) Increasing accuracy by removing false positive events using an NEO spike detector in a wired-OR four-wire configuration. (b) Reduction in input memory and NEO spike detector activity rate using wired-OR and VC-SPD.

preserved. The proposed system combines the VC-SPD with an NEO-based spike detector to reduce the false positive rate and obtain high sensitivity and high accuracy [Fig. 7(a)]. The wired-OR readout effectively removes most noise samples, making the VC-SPD and input memory event-driven processes that are active only when there is a non-collision sample (f_{wor}). Furthermore, employing VC-SPD reduces the proportion of noise samples processed by the NEO detector (f_{VC}), thereby increasing the likelihood that the NEO values exceed the detection threshold [Fig. 3(b)].

Implementing this strategy decreases the activity rate of all spike detection components, resulting in significant power savings. As shown in Fig. 7(b), the activity rates of the input memory and the VC-SPD are reduced by more than one order of magnitude due to wired-OR compression, while the activity rate of the NEO spike detector is decreased by approximately two orders of magnitude as a result of using the VC-SPD output.

B. Feature Extraction

After spike detection, features are extracted from the spike data to differentiate spikes originating from different neurons. In a single-channel recording system, the electrode primarily captures signals from a few nearby neurons. Typically, the waveforms of spikes recorded from each neuron reflect its unique biophysical properties and distance to the electrode. Consequently, a temporal feature space based on the spike waveforms serves as an effective feature space for spike sorting. However, with the increasing number of recording channels in modern high-density MEAs, the likelihood of a unique distance between electrodes and neurons decreases exponentially. At the same time, the probability of recording from neurons with similar biophysical properties, such as morphology and ion channel distributions, also increases. This directly affects the separability of clusters in a shared temporal feature space constructed across multiple electrodes, which is

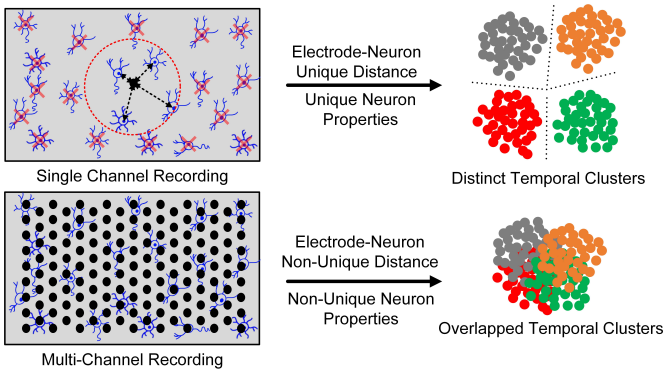


Fig. 8. Effect of increasing the number of recording channels on the separability of temporal clusters. A higher channel count increases the likelihood of recording from neurons with similar electrode-neuron distances and similar biophysical properties, both of which contribute to overlapping waveform shapes and reduced cluster separability.

commonly used in on-chip spike sorters, making it unsuitable for spike sorting in large-scale MEAs (see Fig. 8). This critical issue has not been addressed in previous studies, as they either use single-channel recording datasets or multi-channel datasets with a low number of neurons [22], [39], [40], [42], [43].

Another important challenge in high-density MEAs is that each spike from an individual neuron is typically recorded on multiple nearby channels, leading to redundant detections that require an additional post-processing stage to remove duplicates [39], [40]. This redundancy complicates spike sorting and can degrade accuracy if not handled properly. In contrast, this work uses spatial features that leverage redundant spikes recorded on multiple electrodes to extract the location of neurons for spike sorting, resulting in better cluster separation and classification accuracy. The basic principle of spatial feature extraction is detailed in Fig. 9, which provides an example with six neighboring channels and two neurons. Due to the different locations of neurons relative to the channels, each neuron generates a distinct spike pattern across the channels [Fig. 9(a)], even though they share the same central channel (the channel with the highest recorded amplitude). For each detected spike, the central channel, and a number N of neighboring channels that depends on the geometry of the MEA are determined. Then, the direction of the neuron's location relative to the central channel ($\angle\Delta_{c,s}$) [Fig. 9(b)] and the distance between the neuron's location and the central channel ($|\Delta_{c,s}|$) [Fig. 9(c)] are calculated for each central channel spike (s) using the following equations:

$$\angle\Delta_{c,s} = \angle \sum_{i=1}^{N} A_{i,s} \times (x_i, y_i)_s \tag{3}$$

$$\left|\Delta_{c,s}\right| = \frac{\max_{i=1}^{N} A_{i,s}}{A_{c,s}}. \tag{4}$$

Here, $A_{c,s}$ is the amplitude of the spike in the central channel, $A_{i,s}$ is the largest sample observed on the neighboring channel i (captured from the wired-OR around the central channel's detection time), and $(x_i, y_i)_s$ is the relative coordinates of the neighboring channels with respect to the central channel. In cases where no samples are captured

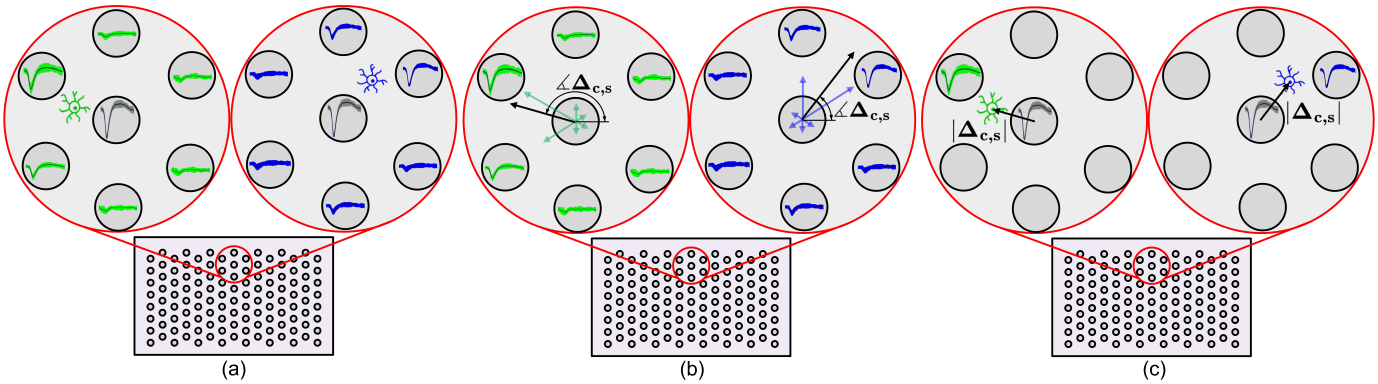


Fig. 9. Overview of spatial feature extraction, illustrated with an example of two neurons and six neighboring channels. (a) Each neuron generates a unique pattern across the neighboring channels. Based on these unique patterns, (b) direction of each neuron's location relative to the central channel and (c) distance from the central channel are calculated.

from the neighboring channels, a less compressive wired-OR configuration can be used to reliably obtain the required data. Equations (3) and (4) calculate the location of the neuron with respect to the central channel. To ensure a uniform feature space across all channels, the estimated location of the neuron is calculated in terms of general coordinates as follows:

$$(x, y)_{\mathrm{FE},s} = (x_c, y_c)_s + \Delta_{c,s} \tag{5}$$

where $(x_c, y_c)_s$ is the coordinates of the central channel. This work is implemented for the staggered electrode layout shown in Fig. 9. However, the proposed algorithm can be adapted to other electrode configurations by updating the relative coordinates $(x_i, y_i)_s$ of the neighboring channels with respect to the central channel in (3).

To evaluate the robustness of spatial features in high-density, large-scale MEA, we compare its feature importance score against common temporal features as a function of the number of recording channels. For this analysis, we used the "ground truth" (Kilosort results) from the ExVivo-1 dataset to identify the waveforms of all spikes associated with each neuron. Each spike waveform was projected into the feature space, and the mean of the features for each neuron was calculated to determine the centroid. The dataset was then re-clustered using these computed centroids. This approach leverages the ground truth to achieve 100% spike detection accuracy, performs all calculations in floating-point format (avoiding fixed-point precision limitations), and establishes cluster centroids directly from the ground truth, eliminating training-phase errors. The analysis was conducted for various sub-array sizes (e.g., 1×1, 2×2, and 3×3) using both our spatial feature and multiple temporal features, including the first three principal components of principal component analysis (PCA), FSDE, peak-FSDE, integer coefficients, and full waveforms. For each feature, we calculated the feature score using $f_{\mathrm{score}} = (N_{\mathrm{Match}}/N_{\mathrm{GT}})$, where N_{Match} represents the number of spikes correctly re-clustered compared to the ground truth, and N_{GT} denotes the total number of ground truth spikes.

The results indicate that for single-channel recordings, temporal features outperform spatial features, as spatial features cannot be fully computed and are limited to detecting only the central channel. The spatial feature score of 0.5 in the single-channel case reflects that half of the recording channels serve as the central channels for only one neuron. For recordings with a low number of channels, all feature spaces yield high feature scores. This is because a small electrode subset typically captures only one or two neurons, simplifying the clustering task due to reduced cluster complexity. However, temporal feature spaces yield low feature scores when applied to high-density MEAs recording activity from a large number of neurons. Specifically, waveforms recorded from different neurons in high-density MEAs are often not sufficiently distinct for effective differentiation. This finding is particularly important because many power-constrained, on-chip spike-sorting architectures utilize a single shared clustering memory and consequently perform spike sorting in a common temporal feature space across all recording channels. Even [40] that uses a central channel to localize spike sorting within a neighborhood must still process groups of channels together to track electrode–tissue drift and ensure long-term stability. Consequently, single-channel datasets, such as those from Quiroga's dataset [44], and multi-channel datasets like the Neuropixel (NP) dataset [45], which, despite having many channels, contain relatively few neurons (eight neurons), are inadequate for assessing the accuracy of a spike sorter designed for high-density recordings.

In contrast, the spatial feature space maintains a nearly constant feature score as the number of recording channels increases, making it a more suitable choice for on-chip spike sorting in high-channel-count MEAs [see Fig. 10(a)]. Moreover, since the spatial feature is mainly dependent on the spike amplitude, which is generally preserved after wired-OR compression, its performance degrades only slightly when compressed data is used. However, temporal features, such as full waveforms where all samples carry important information, are more vulnerable to distortion caused by wired-OR compression, resulting in a more significant drop in performance [see Fig. 10(b)].

It is important to note that this analysis is based on recordings from a macaque retina, which has a 2-D neural structure. Spatial features generally provide better spike-sorting performance in 2-D neural structures recorded with planar MEAs. At

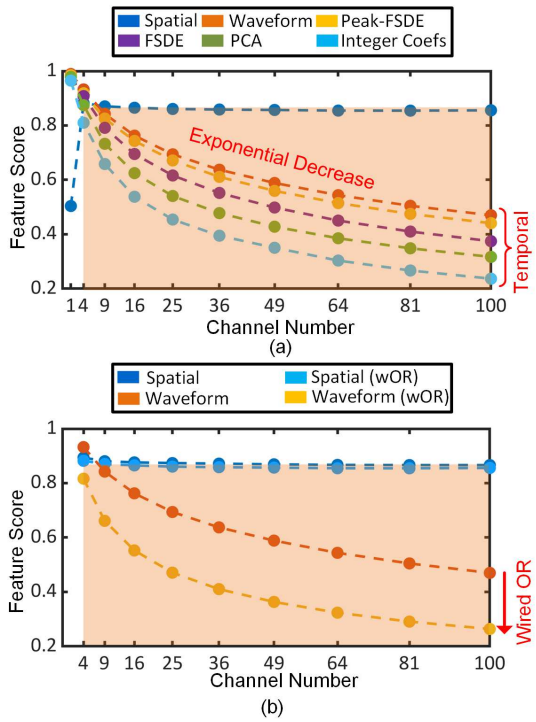


Fig. 10. (a) Effect of increasing the number of recording channels on different features. (b) Impact of wired-OR compression on spatial and waveform features.

the same time, in 3-D neural structures, the effective recording volume of a planar MEA is strongly limited by the rapid decay of spike amplitude with distance from the electrodes. As a result, the neurons that can be reliably recorded are effectively confined to a thin (effectively 2-D) layer close to the MEA surface, where spatial features remain highly informative for spike sorting.

C. Modified SOM Clustering

The proposed clustering method is inspired by the SOM artificial neural network (ANN) to balance the trade-off between accuracy and complexity during the clustering stage. This method utilizes a spatial feature space to create a structural map of neuron locations through an unsupervised learning approach, eliminating the need to retrieve all cluster centroids from memory when calculating similarities and assigning proper cluster indices to new spikes, thereby reducing power consumption during the clustering phase.

During the training phase, a grid is established where elements represent locations in the MEA. The grid is initialized to elements for all electrode positions and increased by elements positioned between the electrodes, adjusted based on the MEA density [see Fig. 11(a)]. Arrays with lower density have more elements between channels. In online spike sorting, where the number of clusters is not predefined, these grid elements serve as potential cluster centroids. During training, the goal is to reposition these grid elements to match the locations of the neurons and eliminate superfluous elements so that there is one element per neuron in the grid. During clustering, when a new spike is detected, the algorithm finds the element in the grid (i.e., the neuron's location) that better matches the spatial feature calculated for that specific spike. The details of the algorithm are outlined below [see also Fig. 12(a)].

The algorithm in the training mode operates using training batches, typically set to six seconds according to the firing rates in the tested datasets. For each spike, the algorithm calculates the Euclidean distance between the spatial feature of the detected spike and the grid elements. The grid element with the smallest distance is identified as the best matching unit (BMU) and its location $(x, y)_{\mathrm{BMU}}$ is adjusted toward the spike's spatial feature $(x, y)_{\mathrm{FE}}$ using a weighted average [Fig. 11(b)]

$$(x, y)_{\mathrm{BMU_{new}}} = \frac{(\alpha - 1)\,(x, y)_{\mathrm{BMU_{old}}} + (x, y)_{\mathrm{FE}}}{\alpha} \tag{6}$$

where α is a constant equal to 16 in this work.

Each grid element tracks the number of times it is identified as the BMU within a training batch, thereby creating a frequency map for the grid. At the end of each training batch, based on the calculated frequency map, grid elements selected as a BMU fewer times than a specified threshold (T_{Batch}) are removed from the grid. This threshold is adaptively determined based on the number of detected spikes in one training batch (N_{spike}) and the number of remaining active elements (N_{element}) as follows:

$$T_{\mathrm{Batch}} = \frac{N_{\mathrm{spike}}}{N_{\mathrm{element}} \times \beta}. \tag{7}$$

Here, β is a constant equal to 4. This approach assumes that the firing rates among neurons within a specific brain region in a training batch are relatively consistent [52], allowing for the removal of elements that are infrequently selected as a BMU. Furthermore, grid elements in close proximity are merged, and a new element, whose location is the average of the merged ones, is created [Fig. 11(c)]. The training phase ends when there are two consecutive training batches without any elements being removed or merged, and the remaining active elements represent the cluster centroids. Fig. 12(b) illustrates how the number of active elements changes with each training batch, highlighting the convergence time of the training phase.

During clustering, for each detected spike, a BMU is identified from the cluster centroids, and the cluster index of the BMU is transmitted. Additionally, the location of the BMU is updated based on (6) to track potential MEA drift on the tissue.

III. System Architecture and Implementation

A. Spatial Spike Sorter Architecture

The spatial spike sorting chip is designed to process wired-OR output in real time from a 1024-channel array [6], [53]—see Fig. 13 for the block diagram and the timing diagram. The chip operates in two separate modes: spike detection and post-processing, which involves feature extraction and clustering. The wired-OR output is event-based and the number of collision-free channels varies from sample to sample. Here, it is assumed that the maximum number of collision-free channels from the wired-OR in one sampling period is 256 (i.e., at least 4× compression on every sampling period [14]). Hence, the chip requires at most 256 clock cycles

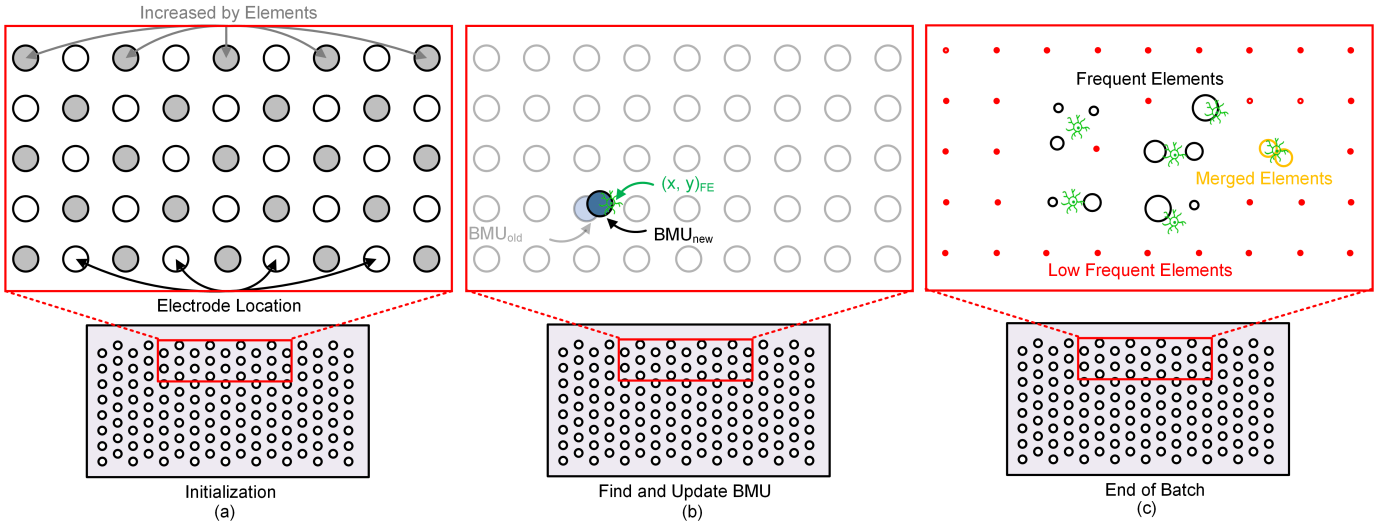


Fig. 11. Overview of the main steps in the SOM clustering algorithm, including (a) grid initialization, (b) finding and updating the BMU, and (c) removing low-frequency elements and merging close elements at the end of each batch. The sizes of the circles correspond to the number of times each element is selected as the BMU.

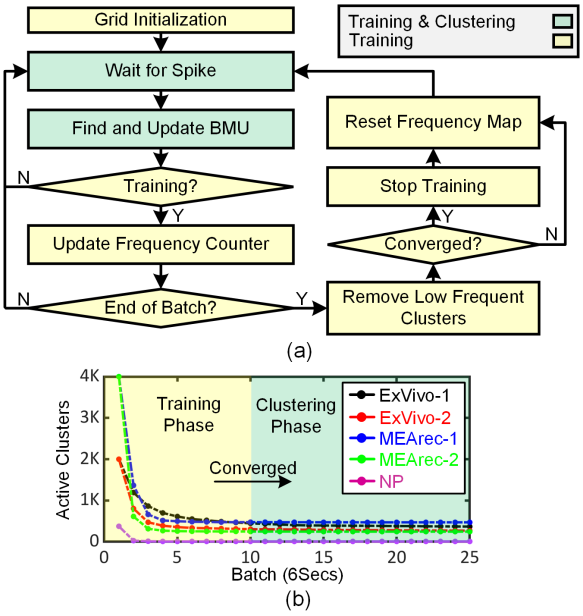


Fig. 12. (a) Flowchart of the proposed SOM clustering algorithm. (b) Convergence time of the training phase for five datasets.

to receive collision-free samples and perform spike detection. Furthermore, it is assumed that at most 16 neurons are firing a spike simultaneously (320 spike/s/ch), which requires another 256 clock cycles to post-process (16 clock cycles per spike). Therefore, operating at a sampling frequency of 20 kHz requires a chip clock of 10.24 MHz.

In spike detection mode, collision-free samples from the wired-OR array are stored in the input memory while simultaneously being processed by the spike detection unit. The channel addresses of detected spikes are stored in the rate adapter FIFO for post-processing. Importantly, during the spike detection phase, the post-processing components are clock-gated to reduce power consumption. Vice versa, spike detection blocks are clock-gated during post-processing.

In post-processing mode, the channel addresses of the detected spikes in the rate adapter FIFO are read one at a time every 16 clock cycles. If the detected channel has the highest amplitude, it is assigned as the central channel and processed. Otherwise, it is discarded, and the next address in the FIFO is processed. Central channels are sent to the feature extractor and, subsequently, to the clustering processor to perform spike sorting.

B. Spike Detection and Input Memory Implementation

The proposed design features a two-stage spike detection process designed to reduce power consumption by moving from an always-on process to an event-driven process with a low activity rate. The chip has a dedicated VC-SPD per channel and one shared NEO-based spike detection for all channels. Since there are 1024 VC-SPD units, it is crucial to implement this process using minimal logic to reduce both power consumption and silicon area.

The VC-SPD utilizes a 3-bit counter with overflow and underflow protection circuitry, optimized using the Karnaugh map method, to track the number of recent wired-OR collision-free samples for each channel. When the counter value for a channel exceeds a predefined threshold (fixed at four for implementation simplicity), a pre-detection event is triggered, activating the NEO spike detection unit. Since it is not determined whether each channel has a collision-free sample until the end of each sampling period, the valid counter can be updated only at that time (EoS). To facilitate this, the state of each channel—whether it had a collision-free sample or not during each sampling period—is saved using an asynchronous SR flip-flop.

The wired-OR collision-free samples are delivered to the chip sequentially. Consequently, at every clock cycle, only one spike pre-detection flag can be activated. Once a pre-detection event is triggered for a channel, the spike detection unit retrieves the amplitude of the last four collision-free samples

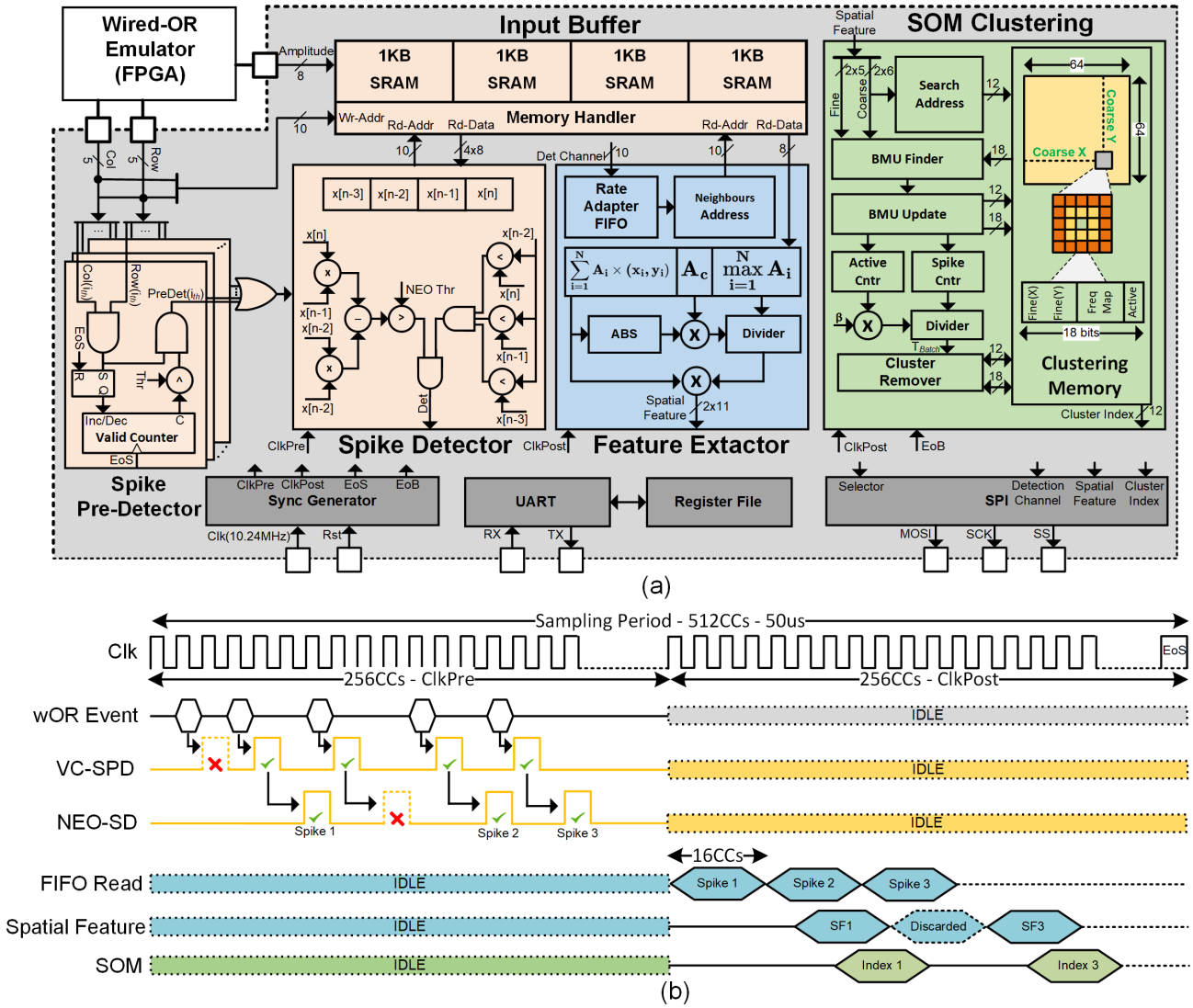


Fig. 13. (a) System architecture and (b) timing diagram of the spatial spike sorting chip.

for that channel from the input memory and identifies the location of the maximum value ($x[n-3:n]$). When the maximum amplitude sample is at $x[n-2]$, the NEO value is computed. This prevents the spike from being detected multiple times and ensures that enough samples are stored in memory to calculate the NEO and extract spatial features. The spike detection flag is activated if the NEO value surpasses the spike detection threshold. This NEO threshold is programmable, providing flexibility to tune detection sensitivity according to varying experimental conditions.

Since the wired-OR collision-free samples are delivered to the chip sequentially, only one sample can be written to the input buffer per clock cycle. However, during spike detection, all four previous samples need to be accessed simultaneously. Additionally, the memory write rate is higher than the memory read rate, as memory is only read during spike pre-detection events. To optimize power consumption and enable reading all four samples in a single clock cycle, the design employs four 1-KB memory units (one for each sampling period) instead of a single 4-KB memory unit. This approach reduces the power consumption of write operations by targeting smaller memory blocks while maintaining fast read access.

C. Feature Extraction Implementation

The feature extraction is performed on the detected spikes stored in the rate adapter FIFO if they belong to a central channel. For each detected spike, the collision-free samples from the central channel and its six neighboring channels are read from the input memories to compute their amplitudes and determine whether the detected spike has the largest amplitude and can be assigned to the central channel or should be discarded. In this chip, retrieving the samples from memory takes seven clock cycles (one for the central channel and six for its neighbors). The amplitude of the central channel is referred as $A_{c,s}$, while the amplitude on neighboring channels is referred as $A_{i,s}$. It should be noted that since we use a high-density MEA and the maximum amplitude sample of the detected channel is aligned to $x[n-2]$, we can expect the maximum amplitude sample of the neighboring channels within $x[n-3:n]$, which are available in the input memory.

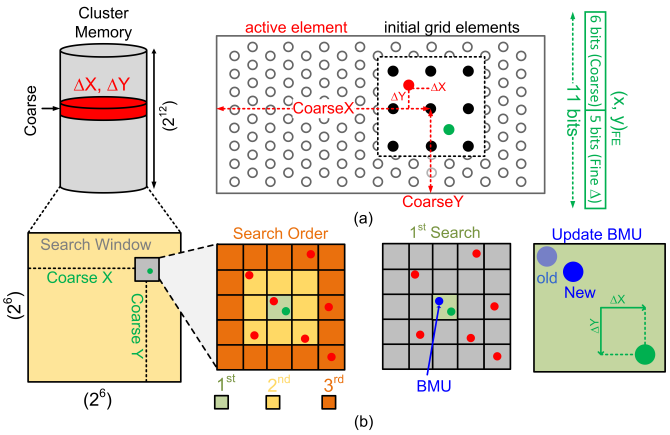


Fig. 14. (a) Coarse and fine representation of the spatial feature space. (b) Sequential search for the BMU in the cluster memory.

Pre-calculations to implement (3) and (4) are made in parallel with the central channel check to accelerate the overall feature extraction. Specifically, the following values are computed: $\sum_{i=1}^{N} A_{i,s} \times (x_i, y_i)_s$, and $\max_{i=1}^{N} A_{i,s}$. When the detected channel is the central channel, feature extraction proceeds, and the magnitude of $\sum_{i=1}^{N} A_{i,s} \times (x_i, y_i)_s$ vector is calculated using the alpha max plus beta min algorithm [54]. Finally, the spatial feature is calculated using the following equations, which are equivalent to (3) and (4) but result in a simpler hardware implementation:

$$\beta_s = \frac{\max_{i=1}^{N} A_{i,s}}{A_{c,s} \times \left| \sum_{i=1}^{N} A_{i,s} \times (x_i, y_i)_s \right|} \tag{8}$$

$$\Delta_{c,s} = \beta_s \times \sum_{i=1}^{N} A_{i,s} \times (x_i, y_i)_s \,. \tag{9}$$

The result of the division in (8) is an unsigned variable less than 1, and it is calculated using a serial restoring division algorithm for power efficiency. The outcome (β_s) is stored in a 12-bit unsigned variable to ensure adequate accuracy. Additionally, after calculating the multiplication in (9), we retained only the six most significant bits (MSBs) to represent $\Delta_{c,s}$. This implies that the spatial feature resolution is pitchsize $\times 2^{-6}$. The spatial feature $(x, y)_{\mathrm{FE},s}$ is represented using 11 bits for each direction (x,y), with 5 bits allocated for the central channel address $(x_c, y_c)_s$ and 6 bits to represent $\Delta_{c,s}$.

D. Modified SOM Clustering Implementation

The proposed SOM clustering algorithm takes advantage of spatial features to create a geometrically mapped clustering memory. This approach aims to reduce the number of read operations from the clustering memory to find the BMU and allows for tracking clusters as the MEA drifts on the tissue. The 11 bits representing the spatial feature in each direction (x,y) can also be interpreted as a coarse and fine representation of the grid element (i.e., the neuron location after training). The coarse representation corresponds to the initial grid elements in the SOM algorithm, while the fine representation denotes the deviations from these initial grid locations based on the spatial features [see Fig. 14(a)]. The coarse representation uses 6 bits, which results in 2^{12} initial grid elements, while the fine representation uses 5 bits. The elements in the map are stored in a clustering memory with 2^{12} words of 18 bits that represent the fine location for (x,y) directions (10 bits), the frequency counter for each element (7 bits), and a flag to indicate whether the element is active (1 bit).

At the start of the training phase, the clustering memory is initialized by setting the fine representations and frequency counters to zero and setting the active flag of all elements high. The six MSBs of the spatial feature in each direction (x,y) are used as the coarse address for reading the clustering memory, pointing to the grid element nearest to the detected spike location. However, during the training phase, some grid elements may become deactivated. If the nearest grid element is inactive, a small finite-state machine (FSM) sequentially scans the neighboring addresses until an active BMU is found [see Fig. 14(b)]. In hardware, the clustering memory is implemented as a dual-port SRAM to accelerate the BMU search procedure. This is necessary because, in the worst case, up to 25 surrounding elements must be retrieved while the rate adapter FIFO provides only 16 cycles to process each spike. The memory stores 4096 words of 18 bits each, allowing the spike sorter to support up to 4096 clusters (neurons), and uses 8× column multiplexing (M8). This reduces the number of physical bitlines, shortening wire lengths and lowering capacitance, which improves both access time and energy efficiency. This is particularly beneficial in this algorithm, where only a local portion of the spatial feature needs to be searched to find the BMU, since most of the time, only the column decoder changes while accessing the memory. With this memory organization, along with direct address decoding from the spatial feature and sequential BMU search, the average access rate to the clustering memory is less than 0.1% per detected spike. This avoids wide associative searches and significantly reduces both dynamic power and memory bandwidth.

After finding the BMU, its frequency counter is increased by one, and its location is updated using (6). The BMU update pipeline consists of three stages: 1) sequential scan and retrieval of the BMU entry; 2) computation of the new fine coordinates and frequency counter; and 3) writing the updated entry back into memory, which can be performed sequentially in 16 clock cycles. If the coarse representation of the BMU changes after the update, two scenarios are possible. In the first, the new coarse address points to an active entry, in which case the two elements are merged. In the second, the new coarse address corresponds to an inactive or deactivated entry, indicating a large shift due to electrode drift. In this case, the BMU entry is copied to the new coarse address and its active flag is set in a single cycle, while the original entry is deactivated by resetting its flag. This lightweight mechanism enables the algorithm and hardware to track MEA drift in real time during the clustering process.

During each training batch, a counter keeps track of the number of active elements in the map (N_{element}). At the end of the batch, the frequency map threshold in (7) is calculated

based on N_{element}. Since this calculation is required only once per batch, it is implemented using a lightweight serial multiplier and divider, minimizing area overhead. Elements with a frequency counter below the threshold are deactivated, and training concludes when no element is deactivated for two consecutive batches.

During the clustering phase, the process is similar to the training phase, except that the frequency map is not checked, and elements are not deactivated. The memory address retrieved from the BMU is used as the cluster index for each detected spike.

IV. Results

A. Accuracy and Compression Performance

To evaluate the performance of our spike sorter, we developed a fixed-point MATLAB model (available online [55]) that implements the components of the spike sorter. This model produces outputs that are fully aligned with both the register transfer level (RTL) design and the post-layout netlist generated by Cadence© Innovus software. Similar to other state-of-the-art methods, the clustering accuracy in this work is calculated using the following equation:

$$\text{Clustering Accuracy} = \frac{N_{\text{Match}}}{N_{\text{GT}}} \tag{10}$$

where N_{Match} represents the number of spikes correctly clustered compared to the ground truth, and N_{GT} denotes the total number of ground truth spikes.

Because state-of-the-art online spike sorters typically rely on single-channel datasets, such as the Quiroga datasets [44], a direct comparison with our spike sorter, which utilizes spatial information from spikes captured on multiple channels, is not possible. Therefore, we benchmarked our spike sorter using three different approaches [see Fig. 15(a)].

1) Comparison with Kilosort [56] on ex vivo primate retina recordings where ground truth is not available, showing over 80% similarity.
2) Evaluation on artificial datasets with available ground truth (MEArec [57], [58]) demonstrating competitive accuracy compared to software-based algorithms (Kilosort, IronClust, and Herdingspikes [59]).
3) Comparison with the only available on-chip spike sorter [39], [40] using a multi-channel dataset (NP [45]), achieving slightly better performance.

Since the chip is implemented for the staggered MEA configuration, the results in this section for ex vivo and MEArec datasets are based on the actual chip output, whereas the NP dataset results are based on the fixed-point MATLAB model. The compression rate shown in Fig. 15(b) for both the wired-OR and spike sorter depends on the spike rate and the number of neurons. The results demonstrate an additional compression of over 40× on top of the wired-OR compression, resulting in an overall compression of more than 1000× across all datasets.

B. Electrical Image

The EI represents the average spatiotemporal voltage footprint of a neuron across the MEA during its spike events.

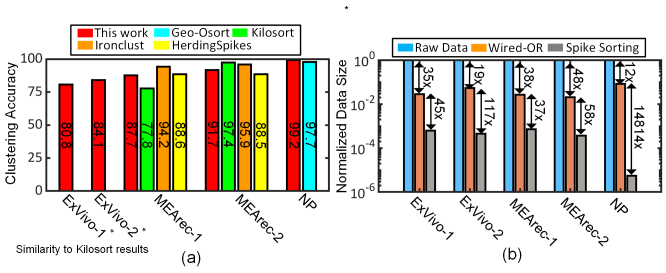


Fig. 15. (a) Clustering accuracy and (b) average data rate reduction performance of the proposed spike sorter. See Appendix for dataset details.

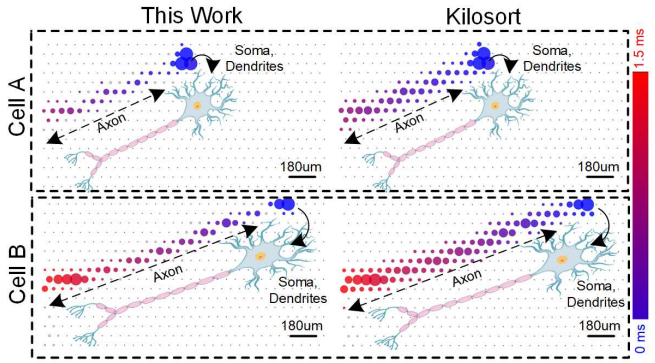


Fig. 16. Comparison of EIs obtained from Kilosort spike sorting on raw data and from the proposed spike sorting on wired-OR processed data.

It captures the characteristic activity pattern of each cell and provides valuable information for cell-type classification [60]. Once spike sorting has identified individual neurons, the EI for each cell can be computed by averaging the recorded waveforms across all electrodes within a defined time window centered on the spike times.

For comparison, Fig. 16 presents simplified EIs for two neurons, computed from the ExVivo-1 dataset based on spike sorting results from both Kilosort and our method. The EIs are collapsed over time to emphasize spatial structure, revealing strong similarity between Kilosort's output on raw data and the real-time results of the proposed approach applied to wired-OR processed data. In both cases, the EI aligns well with the underlying neuronal morphology: the region of highest spike amplitude typically corresponds to the soma, where action potentials are initiated. A visible trace extending from the soma likely represents the axon. In this dataset, recorded from the peripheral retina, the axonal projections of the two neurons appear approximately parallel, consistent with known retinal ganglion cell organization [61].

To evaluate the effect of higher compression in the wired-OR configuration, we applied the proposed spike sorter to a dataset recorded directly with the wired-OR chip [6] from the peripheral retina in a one-wire configuration and extracted EIs from sorted cells. As shown in Fig. 17, even under these highly compressed conditions (163× average compression rate), the extracted EIs present consistent retinal ganglion cell organization, demonstrating that the entire processing pipeline performs well also in real-world scenarios.

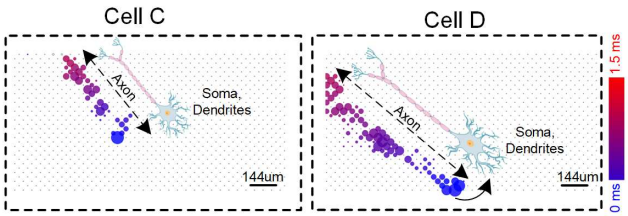


Fig. 17. EIs obtained from a real wired-OR recording under high compression settings (one-wire configuration). The EIs remain consistent with retinal ganglion cell organization, although their quality is reduced due to the high compression.

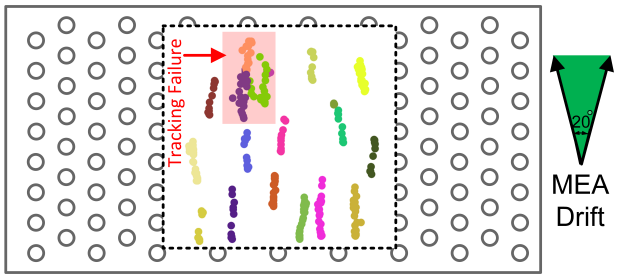


Fig. 18. Cluster centroids over 20 s of drift, showing successful tracking of 14 neurons, with two merged due to spatial proximity.

C. Drift Compensation

Stability is a critical requirement for neural signal processors (NSPs), particularly in implantable devices where relative motion between the MEA and the surrounding neural tissue, commonly referred to as drift, degrades the output accuracy. This degradation often necessitates frequent retraining of the spike sorting algorithm to maintain accurate unit identification and tracking. The proposed spike sorter addresses this challenge by leveraging a spatial feature space, which allows it to inherently compensate for drift without additional computational overhead or reconfiguration. To assess the robustness of this approach, we used MEArec [57] to generate a synthetic extracellular dataset containing 16 neurons. In this dataset, drift is enabled after the training phase, with a speed of 300 μm/min and an angular tolerance of 20°, applied for 20 s. Fig. 18 illustrates the centroid of each cluster at 1-s intervals throughout the drifting period. The results show that the spike sorter successfully tracks 14 out of 16 neurons, demonstrating its ability to handle drift. The remaining two neurons are incorrectly merged into a single cluster, primarily due to their close spatial proximity, which makes them difficult to resolve in the feature space.

D. Chip Measurements

The described online spike sorting chip has been fabricated in a 40-nm CMOS process, occupying a total core area of 0.3 mm^2 (i.e., 0.00029 mm^2 per channel). The die photo and core area breakdown are shown in Fig. 19, highlighting that the design is dominated by the input memory and SOM clustering unit, which account for 59% of the total area.

A Xilinx Zynq UltraScale+ MPSoC FPGA kit is used to evaluate the chip. Since the wired-OR readout is not implemented in this chip, raw data from all 1024 channels

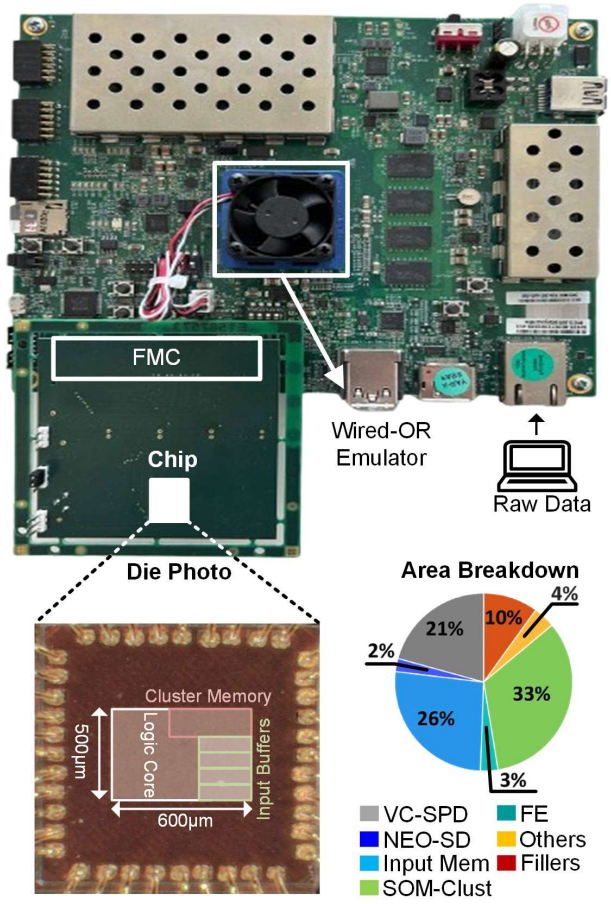


Fig. 19. Test setup illustration and core area breakdown of the proposed spike sorting chip.

at a 20-kHz sampling rate is streamed to the FPGA. The FPGA then emulates the wired-OR readout. The wired-OR event amplitudes and corresponding channel addresses are transmitted to the chip using 19 input pads: ten pads for the channel address, eight pads for the amplitude, and one pad for the event-valid signal within the spike detection time interval.

The chip's output can be configured to provide one of the three data types: spike detection results, extracted features, or cluster index outputs. Depending on the selected mode, the corresponding data is transmitted from the chip to the FPGA via the SPI protocol and subsequently sent to a PC for validation.

To measure the power consumption of the chip, we used the MEArec-1 dataset, which simulates 500 neurons with an average firing rate of 20 Hz. This dataset is the most demanding among those used in our study, containing significantly more neurons than the NPs dataset (eight neurons) and the single-channel datasets (three neurons) employed in other works. The chip operates reliably at a minimum supply voltage (V_{DD}) of 0.72 V. For V_{DD} values lower than this threshold, faults occur in the input memory, leading to malfunctions in spike detection.

At V_{DD} = 0.72 V, the total power consumption of the chip is 76 μW (i.e., 74 nW per channel or 7.8 nJ per spike), as shown in Fig. 20(a). The power consumption breakdown,

TABLE I
COMPARISON WITH PRIOR STATE-OF-THE-ART ON-CHIP SPIKE SORTERS

Design		TVLSI 2019 [22]	TBCAS 2021 [41]	TVLSI 2022 [42]	TBCAS 2022 [43]	JSSC 2023 [40]	This work
Algorithm	Spike Detection	Integer Coefficients	Absolute Thresholding	Absolute Thresholding	NEO	NEO	VC-SPD NEO
	Feature Extraction	Integer Coefficients	FSDE	Waveform	Adaptive Filter	Peak-FSDE	Spatial Features
	Clustering	Modified K-means	Perturbed K-means	Cross Correlation	Configurable	Geo-Osort	Modified SOM
Number of Channels		128	4	64	16	384	1024
Training		On-chip	On-chip	Offline	On-chip	On-chip	On-chip
Feature Space		Temporal	Temporal	Temporal	Temporal	Temporal Central Ch.	Spatial
Dataset	Name	Quiroga	Quiroga	Quiroga	Quiroga	Quiroga, NP	NP, MEArec, ExVivo
	Type	Synthetic	Synthetic	Synthetic	Synthetic	Synthetic	Synthetic, Real
	#Cells	<5	<5	<5	<5	<10	<10, 250∼500, 146∼381
	#Channels	1	1	1	1	1, 128	128, 1024, 512
Clustering Latency		>1.84 ms	1 ms	>1.36 ms	>66.7 µs	<401 µs	<50 µs
Clustering Accuracy(%)	Quiroga	72%	93.2%	85%	94.1%	89.5%	-
	NP	-	-	-	-	97.7%	99.2% [a]
	MEArec	-	-	-	-	-	87.7%, 91.7%
	ExVivo	-	-	-	-	-	80.8%, 84.1%
Technology		65 nm	180 nm	180 nm	22 nm	22 nm	40 nm
Core Voltage		0.54 V	1.5 V	1.8 V	0.5 V	0.59 V	0.72 V
Sampling Frequency		25 kHz	24 kHz	25 kHz	25 kHz	30 kHz	20 kHz
Input Resolution		9 bits	Analog	8 bits	9 bits	12 bits	8 bits
Power per Channel		0.175 µW	4.68 µW	1.74 µW	2.79 µW	1.78 µW	0.074 µW
Energy per Spike		N/A	N/A	N/A	N/A	37.8 µJ [b]	7.8 nJ [c]
Area per Channel		0.003 mm^2	1.032 mm^2	0.047 mm^2	0.014 mm^2	0.0013 mm^2	0.00029 mm^2

[a] Calculated using the MATLAB model.
[b] Calculated using the NP dataset.
[c] Measured using the MEArec-1 dataset.

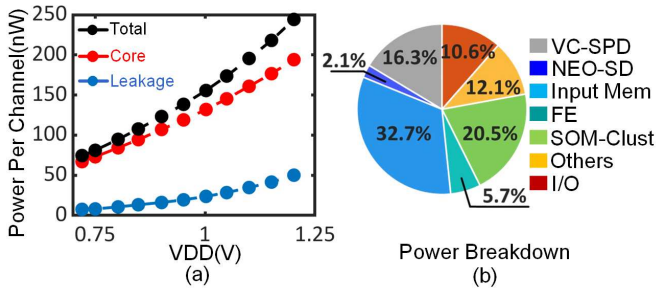


Fig. 20. (a) Power consumption versus V_{DD} of the fabricated chip. (b) Power breakdown of the spike sorting chip.

shown in Fig. 20(b), indicates that the input memory and the SOM clustering are the dominant units. Additionally, the power analysis shows that switching power is the dominant component compared to leakage power.

Table I compares our spike sorter with other state-of-the-art solutions. Although this work is the only online spike sorter that has been validated using both real and synthetic datasets with hundreds of cells, it achieves better accuracy compared to other online spike sorters and competitive accuracy with software-based spike sorters, while performing the training phase entirely unsupervised and on chip. Additionally, this work improves power efficiency by over 23× and area efficiency by over 3× compared to designs with high sorting accuracy [40], [41], [42], [43]. In terms of clustering latency, by reducing the number of read accesses from the clustering memory, this design achieves a maximum latency of 50 μs from the spike detection time for a load of 320 spike/s/ch, which is the lowest latency among all compared works.

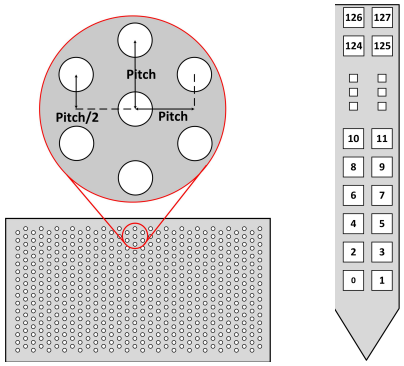
V. DISCUSSIONS

A. Feature Space Trade-Offs

This work targets high-density MEAs that enable single-cell resolution recordings. As shown in Section II-B, the spatial footprint of a spike across neighboring electrodes often carries more discriminative information than its temporal waveform. While combining both spatial and temporal features is ideal, our results show that spatial information alone is sufficient to distinguish between clusters in roughly 90% of cases (see Fig. 10), making it a power- and area-efficient choice for hardware-limited systems.

However, the effectiveness of spatial features depends on electrode density and array geometry. Arrays with large inter-electrode spacing, such as the Utah array, or non-square configurations with many border electrodes, may not provide adequate spatial resolution. In such cases, incorporating temporal features becomes essential to preserve sorting accuracy.

Additionally, in tissues where neurons are distributed in three dimensions, 2-D MEAs may miss key spatial information from deeper neurons. Here, temporal features like spike amplitude, which tend to decrease with distance, can help distinguish units. When spatial clustering merges units with



(a) (b)

Fig. 21. Geometry configuration of recording arrays used in the datasets. (a) ExVivo and MEArec. (b) NP.

TABLE II
DATASET DETAILS

Dataset	Config.	Pitch	#Cells	SNR (μ, σ)	Spike Rate (sp/s/cell)
ExVivo-1	16×32	60 μm	381	(15.3, 7.7)	14
ExVivo-2	16×32	60 μm	146	(8.9, 2.5)	11.69
MEArec-1	32×32	36 μm	500	(11.2, 3.6)	20
MEArec-2	32×32	36 μm	250	(11.0, 3.7)	20
NP	64×2	20 μm	8	(38.9, 33.1)	2.26

overlapping footprints but different amplitudes, a hierarchical clustering strategy that refines results using temporal features can recover accuracy with minimal added complexity.

B. Scalability

Increasing the number of recording channels affects three main components of the system: the wired-OR compression, the pre-processing (spike detection), and the post-processing stages. As discussed in [51], a higher number of channels increases the probability of collisions in the wired-OR structure. However, this can be mitigated by either increasing the number of output wires (i.e., recording channels per group) or increasing the number of quantization bits. Importantly, these adjustments help maintain a constant compression rate while reducing the probability of collisions, thereby supporting scalability without compromising data integrity.

In the pre-processing phase, each channel is equipped with its own spike detector and a dedicated memory block with four locations to store wired-OR outputs. As the number of channels increases, so do the required hardware resources and the processing workload. Increasing the number of collision-free samples per sampling interval results in a higher number of clock cycles required, necessitating an increase in clock frequency to maintain real-time operation. Consequently, the silicon area and power consumption in this stage scale almost linearly with the number of channels, assuming a small routing overhead.

In the post-processing stage, increasing the number of recording channels results in a higher number of detected spikes from the pre-processing stage, which increases the overall processing workload. Consequently, the number of clock cycles required grows, and the clock frequency of this stage must be increased to sustain real-time performance. The rate adapter FIFO must be expanded to accommodate the increased number of detected spikes, while the clustering memory size grows proportionally with the number of recording channels. However, due to the structured memory design, the number of memory accesses per spike remains constant, allowing the post-processing stage to scale linearly with the number of recording channels in terms of memory requirements and computational load.

In both the pre-processing and post-processing phases, if increasing the clock frequency is not feasible due to technological limitations, parallelization can be employed. In the pre-processing phase, the array can be divided into sub-groups, with each sub-group processed in parallel. Since each channel already has its own spike pre-detection unit and input memory, parallelization merely involves duplicating spike detection units, which has a negligible impact on power and area. In the post-processing phase, although duplicating the feature extraction blocks is a straightforward option, parallelizing the SOM clustering is more challenging. This is mainly due to the channels at the boundaries of each sub-group. This can be addressed by slightly overlapping the clustering memories, which introduces only a negligible overhead.

VI. CONCLUSION

This work presents a 1024-channel ultra-low-power online spike sorting chip, fabricated in a 40-nm CMOS process, with a core area of 0.3 mm^2 and power consumption of 76 μW. The chip integrates spike detection, spatial feature extraction, and SOM clustering, validated with real and synthetic datasets containing hundreds of cells. The design outperforms other online spike sorters in accuracy and offers competitive performance with software-based methods, improving power efficiency by 23$\times$ and area efficiency by 3$\times$. Additionally, it achieves the lowest clustering latency of 50 μs for a load of 320 spike/s/ch. These results demonstrate the chip's potential for large-scale neural recordings, with future work focused on further adaptability for real-world neuroscience applications.

APPENDIX

A. Dataset Information

To evaluate the performance of the proposed spike sorting method, the processing pipeline is validated using five distinct datasets, including both ex vivo and artificial recordings (see Table II). Two ex vivo datasets, ExVivo-1 and ExVivo-2, are obtained from MEA recordings from the macaque retina and bandpass filtered between 300 and 5000 Hz [62]. These datasets include recordings from 381 and 146 cells, respectively (spike sorting results obtained using Kilosort [56]). The electrodes are spaced 60 μm apart, have a diameter of 7.5 μm, and have a 16×32 electrode configuration shown in Fig. 21(a).

While ex vivo datasets provide valuable real-world insights, the accuracy of spike times and clusters depends on the specific spike sorting algorithm, and ground truth results are not available. Furthermore, it is not possible to arbitrarily vary certain characteristics of the dataset, such as spike SNR

(defined as single-ended spike amplitude divided by noise standard deviation) and firing rate, to study their effect on the spike sorting accuracy. To address these limitations, two artificial datasets are created with varying spike SNR and firing rates using the Python platforms MEArec and Spike Interface [57], [58]. These artificial datasets feature a 36-μm pitch size between electrodes and a 32×32 electrode configuration, as shown in Fig. 21(a). MEArec-1 includes 500 cells, while MEArec-2 includes 250 cells (available online [55]). A third-order Butterworth bandpass filter with cutoff frequencies of 300–6000 Hz was applied to both datasets.

Finally, an NP dataset [45] previously used by [39] and [40] is selected for benchmarking spike sorting performance. This dataset, referred to as NP, comprises eight cells and employs a 64×2 electrode configuration with a 20-μm electrode pitch, as shown in Fig. 21(b).

REFERENCES

[1] M. Shaeri, U. Shin, A. Yadav, R. Caramellino, G. Rainer, and M. Shoaran, "A 2.46-mm^2 miniaturized brain–machine interface (MiBMI) enabling 31-class brain-to-text decoding," *IEEE J. Solid-State Circuits*, vol. 59, no. 11, pp. 3566–3579, Nov. 2024.

[2] F. R. Willett et al., "A high-performance speech neuroprosthesis," *Nature*, vol. 620, no. 7976, pp. 1031–1036, Aug. 2023.

[3] S. L. Metzger et al., "A high-performance neuroprosthesis for speech decoding and avatar control," *Nature*, vol. 620, no. 7976, pp. 1037–1046, Aug. 2023.

[4] F. R. Willett, D. T. Avansino, L. R. Hochberg, J. M. Henderson, and K. V. Shenoy, "High-performance brain-to-text communication via handwriting," *Nature*, vol. 593, no. 7858, pp. 249–254, May 2021.

[5] L. R. Hochberg et al., "Reach and grasp by people with tetraplegia using a neurally controlled robotic arm," *Nature*, vol. 485, no. 7398, pp. 372–375, May 2012.

[6] M. Jang et al., "A 1024-channel 268 nw/pixel 36×36 μm^2/channel data-compressive neural recording IC for high-bandwidth brain–computer interfaces," *IEEE J. Solid-State Circuits*, vol. 59, no. 4, pp. 1123–1136, Apr. 2024.

[7] X. Yang et al., "An AC-coupled 1st-order Δ-$\Delta\Sigma$ readout IC for area-efficient neural signal acquisition," *IEEE J. Solid-State Circuits*, vol. 58, no. 4, pp. 949–960, Apr. 2023.

[8] Z. Zhao et al., "Ultraflexible electrode arrays for months-long high-density electrophysiological mapping of thousands of neurons in rodents," *Nature Biomed. Eng.*, vol. 7, no. 4, pp. 520–532, Oct. 2022, doi: 10.1038/s41551-022-00941-y.

[9] D.-Y. Yoon, S. Pinto, S. Chung, P. Merolla, T.-W. Koh, and D. Seo, "A 1024-channel simultaneous recording neural SoC with stimulation and real-time spike detection," in *Proc. Symp. VLSI Circuits*, Jun. 2021, pp. 1–2.

[10] N. A. Steinmetz et al., "Neuropixels 2.0: A miniaturized high-density probe for stable, long-term brain recordings," *Science*, vol. 372, no. 6539, Apr. 2021, Art. no. eabf4588, doi: 10.1126/science.abf4588.

[11] S. Wang et al., "A compact quad-shank CMOS neural probe with 5,120 addressable recording sites and 384 fully differential parallel channels," *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 6, pp. 1625–1634, Dec. 2019.

[12] N. Even-Chen et al., "Power-saving design opportunities for wireless intracortical brain–computer interfaces," *Nature Biomed. Eng.*, vol. 4, no. 10, pp. 984–996, Aug. 2020, doi: 10.1038/s41551-020-0595-9.

[13] D. Valencia, P. P. Mercier, and A. Alimohammad, "Efficient in vivo neural signal compression using an autoencoder-based neural network," *IEEE Trans. Biomed. Circuits Syst.*, vol. 18, no. 3, pp. 691–701, Jun. 2024.

[14] D. G. Muratore, P. Tandon, M. Wootters, E. J. Chichilnisky, S. Mitra, and B. Murmann, "A data-compressive wired-OR readout for massively parallel neural recording," *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 6, pp. 1128–1140, Dec. 2019.

[15] C. Aprile et al., "Adaptive learning-based compressive sampling for low-power wireless implants," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 11, pp. 3929–3941, Nov. 2018.

[16] T. Wu, W. Zhao, E. Keefer, and Z. Yang, "Deep compressive autoencoder for action potential compression in large-scale neural recording," *J. Neural Eng.*, vol. 15, no. 6, Dec. 2018, Art. no. 066019.

[17] X. Liu et al., "A fully integrated wireless compressed sensing neural signal acquisition system for chronic recording and brain machine interface," *IEEE Trans. Biomed. Circuits Syst.*, vol. 10, no. 4, pp. 874–883, Aug. 2016.

[18] M. Shoaran, M. H. Kamal, C. Pollo, P. Vandergheynst, and A. Schmid, "Compact low-power cortical recording architecture for compressive multichannel data acquisition," *IEEE Trans. Biomed. Circuits Syst.*, vol. 8, no. 6, pp. 857–870, Dec. 2014.

[19] M. Zaidi et al., "Inferring light responses of primate retinal ganglion cells using intrinsic electrical signatures," *J. Neural Eng.*, vol. 20, no. 4, Aug. 2023, Art. no. 045001.

[20] S. Mukhopadhyay and G. C. Ray, "A new interpretation of nonlinear energy operator and its efficacy in spike detection," *IEEE Trans. Biomed. Eng.*, vol. 45, no. 2, pp. 180–187, Feb. 1998.

[21] K. H. Kim and S. J. Kim, "A wavelet-based method for action potential detection from extracellular neural signal recording with low signal-to-noise ratio," *IEEE Trans. Biomed. Eng.*, vol. 50, no. 8, pp. 999–1011, Aug. 2003, doi: 10.1109/TBME.2003.814523.

[22] A. T. Do, S. M. A. Zeinolabedin, D. Jeon, D. Sylvester, and T. T. Kim, "An area-efficient 128-channel spike sorting processor for real-time neural recording with 0.175 μW/channel in 65-nm CMOS," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 1, pp. 126–137, Jan. 2019.

[23] G. Hilgen et al., "Unsupervised spike sorting for large-scale, high-density multielectrode arrays," *Cell Rep.*, vol. 18, no. 10, pp. 2521–2532, Mar. 2017, doi: 10.1016/j.celrep.2017.02.038.

[24] Y. Yang and A. J. Mason, "Frequency band separability feature extraction method with weighted Haar wavelet implementation for implantable spike sorting," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 6, pp. 530–538, Jun. 2017.

[25] S. E. Paraskevopoulou, D. Y. Barsakcioglu, M. R. Saberi, A. Eftekhar, and T. G. Constandinou, "Feature extraction using first and second derivative extrema (FSDE) for real-time and hardware-efficient spike sorting," *J. Neurosci. Methods*, vol. 215, no. 1, pp. 29–37, Apr. 2013, doi: 10.1016/j.jneumeth.2013.01.012.

[26] A. M. Kamboh and A. J. Mason, "Computationally efficient neural feature extraction for spike sorting in implantable high-density recording systems," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 21, no. 1, pp. 1–9, Jan. 2013.

[27] M. Shaeri and A. M. Sodagar, "A framework for on-implant spike sorting based on salient feature selection," *Nature Commun.*, vol. 11, no. 1, p. 3278, Jun. 2020, doi: 10.1038/s41467-020-17031-9.

[28] R. J. Vogelstein, K. Murari, P. H. Thakur, C. Diehl, S. Chakrabartty, and G. Cauwenberghs, "Spike sorting with support vector machines," in *Proc. 26th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, vol. 3, Apr. 2004, pp. 546–549.

[29] D. Valencia and A. Alimohammad, "An efficient hardware architecture for template matching-based spike sorting," *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 3, pp. 481–492, Jun. 2019.

[30] J. Wouters, F. Kloosterman, and A. Bertrand, "Towards online spike sorting for high-density neural probes using discriminative template matching with suppression of interfering spikes," *J. Neural Eng.*, vol. 15, no. 5, Jul. 2018, Art. no. 056005, doi: 10.1088/1741-2552/aace8a.

[31] U. Rutishauser, E. M. Schuman, and A. N. Mamelak, "Online detection and sorting of extracellularly recorded action potentials in human medial temporal lobe recordings, in vivo," *J. Neurosci. Methods*, vol. 154, nos. 1–2, pp. 204–224, Jun. 2006, doi: 10.1016/j.jneumeth.2005.12.033.

[32] Y. Yang and A. J. Mason, "On-chip spike clustering & classification using self organizing map for neural recording implants," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Nov. 2011, pp. 145–148.

[33] S. E. Paraskevopoulou, D. Wu, A. Eftekhar, and T. G. Constandinou, "Hierarchical adaptive means (HAM) clustering for hardware-efficient, unsupervised and real-time spike sorting," *J. Neurosci. Methods*, vol. 235, pp. 145–156, Sep. 2014, doi: 10.1016/j.jneumeth.2014.07.004.

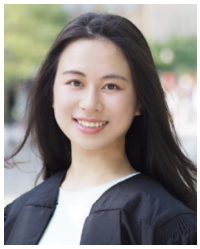
[34] C. Seong, W. Lee, and D. Jeon, "A multi-channel spike sorting processor with accurate clustering algorithm using convolutional autoencoder," *IEEE Trans. Biomed. Circuits Syst.*, vol. 15, no. 6, pp. 1441–1453, Dec. 2021.

[35] D. Valencia and A. Alimohammad, "Neural spike sorting using binarized neural networks," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 29, pp. 206–214, 2021.

[36] E. Choi et al., "A Δ-based spike sorting SoC with end-to-end implementation of event-driven binary autoencoder neural network in analog CIM achieving 94.54% accuracy and 3.11μW/ch," in *Proc. IEEE Symp. VLSI Technol. Circuits (VLSI Technology Circuits)*, Jun. 2024, pp. 1–2.
[37] P. D. Wolf, "Thermal considerations for the design of an implanted cortical brain–machine interface (BMI)," in *Indwelling Neural Implants: Strategies for Contending With the in Vivo Environment*. Boca Raton, FL, USA: CRC Press, 2008.
[38] J. B. Troy and B. B. Lee, "Steady discharges of macaque retinal ganglion cells," *Vis. Neurosci.*, vol. 11, no. 1, pp. 111–118, Jan. 1994.
[39] Y. Chen et al., "A 384-channel online-spike-sorting IC using unsupervised geo-OSort clustering and achieving 0.0013 mm^2/Ch and 1.78μW/ch," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2023, pp. 486–488.
[40] Y. Chen et al., "An online-spike-sorting IC using unsupervised geometry-aware OSort clustering for efficient embedded neural-signal processing," *IEEE J. Solid-State Circuits*, vol. 58, no. 11, pp. 2990–3002, Nov. 2023.
[41] H. Hao, J. Chen, A. G. Richardson, J. Van der Spiegel, and F. Aflatouni, "A 10.8 μW neural signal recorder and processor with unsupervised analog classifier for spike sorting," *IEEE Trans. Biomed. Circuits Syst.*, vol. 15, no. 2, pp. 351–364, Apr. 2021.
[42] F. Kalantari, H. Hosseini-Nejad, and A. M. Sodagar, "Hardware-efficient, on-the-fly, on-implant spike sorter dedicated to brain-implantable microsystems," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 30, no. 8, pp. 1098–1106, Aug. 2022.
[43] S. M. A. Zeinolabedin et al., "A 16-channel fully configurable neural SoC with 1.52 μW/ch signal acquisition, 2.79 μW/ch real-time spike classifier, and 1.79 TOPS/W deep neural network accelerator in 22 nm FDSOI," *IEEE Trans. Biomed. Circuits Syst.*, vol. 16, no. 1, pp. 94–107, Feb. 2022.
[44] R. Q. Quiroga, Z. Nadasdy, and Y. Ben-Shaul, "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering," *Neural Comput.*, vol. 16, no. 8, pp. 1661–1687, Aug. 2004, doi: 10.1162/089976604774201631.
[45] (2016). *Neuropixels Datasets, 'Sorting Comparison Results'*. Accessed: May 10, 2023. [Online]. Available: http://phy.cortexlab.net/data/sortingComparison/
[46] L. Schäffer, Z. Nagy, Z. Kincses, R. Fiáth, and I. Ulbert, "Spatial information based OSort for real-time spike sorting using FPGA," *IEEE Trans. Biomed. Eng.*, vol. 68, no. 1, pp. 99–108, Jan. 2021.
[47] A. Akhoundi, Y. Landbrug, P. Yan, E. J. Chichilnisky, B. Murmann, and D. G. Muratore, "15.2 A 1024-channel 0.00029 mm^2/ch 74nW/ch online spatial spike-sorting chip with event-driven spike detection and self-organizing map clustering," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 68, Feb. 2025, pp. 268–270.
[48] M. Nawrot, "Precisely timed signal transmission in neocortical networks with reliable intermediate-range projections," *Frontiers Neural Circuits*, vol. 3, p. 371, Feb. 2009.
[49] G. T. Neske, S. L. Patrick, and B. W. Connors, "Contributions of diverse excitatory and inhibitory neurons to recurrent network activity in cerebral cortex," *J. Neurosci.*, vol. 35, no. 3, pp. 1089–1105, Jan. 2015.
[50] F. S. Chance, L. F. Abbott, and A. D. Reyes, "Gain modulation from background synaptic input," *Neuron*, vol. 35, no. 4, pp. 773–782, Aug. 2002.
[51] P. Yan et al., "Data compression versus signal fidelity tradeoff in wired-OR analog-to-digital compressive arrays for neural recording," *IEEE Trans. Biomed. Circuits Syst.*, vol. 17, no. 4, pp. 754–767, Aug. 2023.
[52] Z. Zhang and T. G. Constandinou, "Firing-rate-modulated spike detection and neural decoding co-design," *J. Neural Eng.*, vol. 20, no. 3, May 2023, Art. no. 036003, doi: 10.1088/1741-2552/accece.
[53] M. Jang et al., "A 1024-channel 268 nW/pixel 36×36 μm^2/ch data-compressive neural recording IC for high-bandwidth brain–computer interfaces," in *Proc. IEEE Symp. VLSI Technol. Circuits (VLSI Technol. Circuits)*, Jun. 2023, pp. 1–2.
[54] H. Yassin, A. Akhoundi, E.-S. Hasaneen, and D. G. Muratore, "A power-efficient oscillatory synchronization feature extractor for closed-loop neuromodulation," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 71, no. 6, pp. 3176–3180, Jun. 2024.
[55] A. Akhoundi, P. Yan, Y. Landbrug, B. Murmann, E. Chichilnisky, and D. Muratore. (2025). *Fixed-Point MATLAB Model for Spatial Spike Sorting*. Accessed: Jul. 11, 2025. [Online]. Available: https://gitlab.tudelft.nl/sbi_tudelft-public-code/spatial_spike_sorting.git
[56] M. Pachitariu, S. Sridhar, J. Pennington, and C. Stringer, "Spike sorting with Kilosort4," *Nature Methods*, vol. 21, no. 5, pp. 914–921, May 2024.
[57] A. P. Buccino and G. T. Einevoll, "MEArec: A fast and customizable testbench simulator for ground-truth extracellular spiking activity," *Neuroinformatics*, vol. 19, no. 1, pp. 185–204, Jan. 2021.
[58] A. P. Buccino et al., "SpikeInterface, a unified framework for spike sorting," *eLife*, vol. 9, p. 61834, Nov. 2020.
[59] J. Magland et al., "SpikeForest, reproducible Web-facing ground-truth validation of automated neural spike sorters," *eLife*, vol. 9, May 2020, Art. no. e55167.
[60] E. G. Wu et al., "Decomposition of retinal ganglion cell electrical images for cell type and functional inference," *J. Neural Eng.*, vol. 22, no. 4, Aug. 2025, Art. no. 046007.
[61] A. M. Litke et al., "What does the eye tell the brain: Development of a system for the large-scale recording of retinal output activity," *IEEE Trans. Nucl. Sci.*, vol. 51, no. 4, pp. 1434–1440, Aug. 2004.
[62] W. Dabrowski et al., "Development of integrated circuits for readout of microelectrode arrays to image neuronal activity in live retinal tissue," in *Proc. IEEE Nucl. Sci. Symp. Conf. Rec.*, Feb. 2003, pp. 956–960.

Arash Akhoundi (Graduate Student Member, IEEE) received the B.Sc. degree in electrical engineering from the University of Tehran, Tehran, Iran, in 2015, and the M.Sc. degree in electrical engineering from the Sharif University of Technology, Tehran, in 2018. He is currently pursuing the Ph.D. degree with the Bioelectronics Section, Department of Microelectronics, Delft University of Technology, Delft, The Netherlands.

From 2018 to 2022, he worked as a Digital Signal Processing and FPGA Engineer, where he gained experience in signal processing implementation and digital IC design. His research focuses on massively parallel neural interfaces, including on-chip data compression, low-power spike sorting, and real-time neural signal processing.



Pumiao Yan (Member, IEEE) received the B.Sc. degree in electrical and computer engineering from Cornell University, Ithaca, NY, USA, in 2018, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, USA, in 2020 and 2025, respectively.

She was a Seth A. Ritch Bio-X Graduate Student Fellow during her Ph.D. studies. Her research focuses on algorithm-hardware co-design and signal processing for analog-to-digital compression hardware architectures for neural interfaces.

Yawende Landbrug (Student Member, IEEE) received the dual M.Sc. degree in biomedical engineering from TU Delft, Delft, The Netherlands, and in neuroscience from Erasmus University Rotterdam, Rotterdam, The Netherlands.

He is currently a Biomedical Engineer and a Neuroscientist with expertise in neurotechnology and neural data analysis. He has contributed to research projects at Erasmus MC and collaborative projects between TU Delft and Stanford University, Stanford, CA, USA. His work includes the development and validation of a hardware-efficient, online, on-chip spike detection algorithm that outperforms state-of-the-art techniques, as well as investigating altered brain connectivity in ASD models using functional ultrasound imaging and electrophysiology. Professionally, he has experience with neuroimaging systems, including functional ultrasound, EEG, and fNIRS. His research bridges biomedical engineering and neuroscience, with the goal of advancing scalable, real-time neural recording systems for brain–computer interfaces.

Madeline Hays is currently a Dual Bioengineering-Electrical Engineering Graduate Researcher at Stanford Artificial Retina Project. Her work focuses on validating the high-density, data-compressive neural interface for restoring vision. She engineered a custom experimental platform enabling long-duration stimulation and recording of retinal ganglion cells in ex vivo primate and rodent retina. She also co-develops the software framework for real-time neural data acquisition, display, and response. Analyses from these recordings investigate the tradeoff of data compression on spike fidelity and cell-type identification. These findings inform rapid calibration of whole cell populations at single-cell resolution, enabling precise targeting in future in vivo epiretinal implants. Her work is made possible by close collaboration with the talented, interdisciplinary team spanning neuroscience, engineering, and clinical research across multiple institutions.

Boris Murmann (Fellow, IEEE) received the Dipl.-Ing. (FH) degree in communications engineering from Fachhochschule Dieburg, Dieburg, Germany, in 1994, the M.S. degree in electrical engineering from Santa Clara University, Santa Clara, CA, USA, in 1999, and the Ph.D. degree in electrical engineering from the University of California at Berkeley, Berkeley, CA, USA, in 2003.

From 1994 to 1997, he was with Neutron Mikroelektronik GmbH, Hanau, Germany, where he was involved in the development of low-power and smart-power application-specific integrated circuits (ASICs) in automotive CMOS technology. From 2004 to 2023, he was with the Department of Electrical Engineering, Stanford University, Stanford, CA, USA, where he was an Assistant Professor, an Associate Professor, and a Full Professor. He is currently with the Department of Electrical and Computer Engineering, University of Hawai'i at Mānoa, Honolulu, HI, USA. His research interests are in the area of mixed-signal integrated circuit design, including sensor interfaces, A/D and D/A conversion, high-speed communication links, embedded machine learning (tinyML), electronic design automation, as well as open-source chip design.

Dr. Murmann was a co-recipient of the Best Student Paper Award from the Very Large-Scale Integration Circuits Symposium in 2008 and 2021, and the Best Invited Paper Award from the IEEE Custom Integrated Circuits Conference (CICC) in 2008. He received the Agilent Early Career Professor Award in 2009, the Friedrich Wilhelm Bessel Research Award in 2012, the SIA-SRC University Researcher Award for lifetime research contributions to U.S. semiconductor industry in 2021, and the SRC Aristotle Award for contributions to teaching and mentorship in 2024. He was the 2017 Program Chair of the IEEE International Solid-State Circuits Conference (ISSCC) and the 2023 General Co-Chair of the IEEE International Symposium on Circuits and Systems (ISCAS). He serves as the Editor-in-Chief for IEEE JOURNAL OF SOLID-STATE CIRCUITS.

E. J. Chichilnisky received the B.A. degree in mathematics from Princeton University, Princeton, NJ, USA, and the M.S. degree in mathematics and the Ph.D. degree in neuroscience from Stanford University, Stanford, CA, USA.

He is currently the John R. Adler Professor of neurosurgery and a Professor of ophthalmology with Stanford University, where he has been working since 2013. Previously, he worked at the Salk Institute for Biological Studies for 15 years. His research has focused on understanding the spatiotemporal patterns of electrical activity in the retina that convey visual information to the brain, and their origins in retinal circuitry, using large-scale multi-electrode recordings. His ongoing work now focuses on using basic science knowledge along with electrical stimulation to develop a novel high-fidelity artificial retina for treating people with low vision.

Dr. Chichilnisky was a recipient of an Alfred P. Sloan Research Fellowship, a McKnight Scholar Award, a McKnight Technological Innovation in Neuroscience Award, and a Research to Prevent Blindness Stein Innovation Award.

Dante G. Muratore (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from the Politecnico of Torino, Turin, Italy, in 2012 and 2013, respectively, and the Ph.D. degree in microelectronics from the Integrated Microsystems Laboratory, University of Pavia, Pavia, Italy, in 2017.

From 2015 to 2016, he was a Visiting Scholar with Microsystems Technology Laboratories, Massachusetts Institute of Technology, Cambridge, MA, USA. From 2016 to 2020, he was a Post-Doctoral Fellow with Stanford University, Stanford, CA, USA. Since 2020, he has been an Assistant Professor with the Department of Microelectronics, Delft University of Technology, Delft, The Netherlands, leading the Smart Brain Interfaces Group. His group investigates hardware and system solutions for high-bandwidth brain–machine interfaces that can interact with the nervous system at natural resolution. They contribute solutions for massively parallel bidirectional interfaces, on-chip neural signal processing, and wireless power and data transfer.

Dr. Muratore was a recipient of the Wu Tsai Neurosciences Institute Interdisciplinary Scholar Award.