Spike Sorting Algorithms and Their Efficient Hardware Implementation: A Comprehensive Survey

Tim Zhang, Mostafa Rahimi Azghadi*, Corey Lammie, Amirali Amirsoleimani, Roman Genov

***Abstract*—*Objective*: Spike sorting is a set of techniques used to analyze extracellular neural recordings, attributing individual spikes to individual neurons. This field has gained significant interest in neuroscience due to advances in implantable microelectrode arrays, capable of recording thousands of neurons simultaneously. High-density electrodes, combined with efficient and accurate spike sorting systems, are essential for various applications, including Brain Machine Interfaces (BMI), experimental neural prosthetics, real-time neurological disorder monitoring, and neuroscience research. However, given the resource constraints of modern applications, relying solely on algorithmic innovation is not enough. Instead, a co-optimization approach that combines hardware and spike sorting algorithms must be taken to develop neural recording systems suitable for resource-constrained environments, such as wearable devices and BMIs. This co-design requires careful consideration when selecting appropriate spike-sorting algorithms that match specific hardware and use cases. *Approach*: We investigated the recent literature on spike sorting, both in terms of hardware advancements and algorithms innovations. Moreover, we dedicated special attention to identifying suitable algorithm-hardware combinations, and their respective real-world applicabilities. *Main Results*: In this review, we first examined the current progress in algorithms, and described the recent departure from the conventional "3-step" algorithms in favor of more advanced template matching or machine-learning-based techniques. Next, we explored innovative hardware options, including Application-Specific Integrated Circuits (ASICs), Field-Programmable Gate Arrays (FPGAs), and In-Memory Computing Devices (IMCs). Additionally, the challenges and future opportunities for spike sorting are discussed. *Significance*: This comprehensive review systematically summarizes the latest spike sorting techniques and demonstrates how they enable researchers to overcome traditional obstacles and unlock novel applications. Our goal is for this work to serve as a roadmap for future researchers seeking to identify the most appropriate spike sorting implementations for various experimental settings. By doing so, we aim to facilitate the advancement of this exciting field and promote the development of innovative solutions that drive progress in neural engineering research.**

***Index Terms*—Spike Sorting, Hardware, Machine Learning, Neuromorphic Engineering.**

I. Introduction

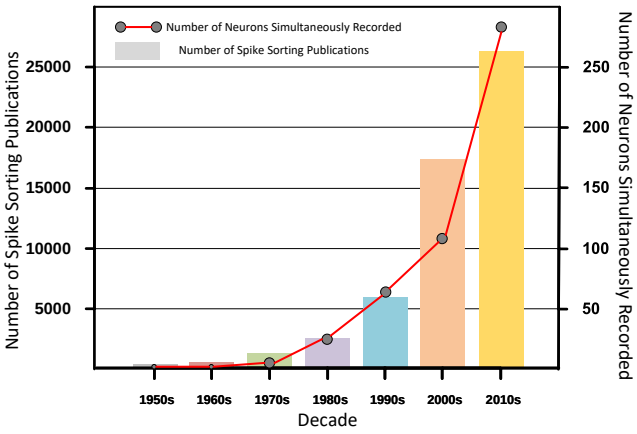

Fig. 1: The number of simultaneously recorded neurons (line) and the number of spike-sorting-related publications (bar), reported over the past seven decades. The trend line shows an exponential increase in spike sorting publications accompanying the improvement in recording technologies capable of recording an increasing number of neurons simultaneously.

ELECTROPHYSIOLOGY, the study of ions flow and electrical properties of biological cells and tissues, has long been a field of interest to neuroscientists. Earlier groundbreaking efforts focused on recording single cell activities, with innovations such as the patch clamp technique pioneered by Hodgkin and Huxley [1], enabled scientists to develop an understanding of how neurons functioned on a single-cell level. In the past few decades, however, neuroscientists have become more interested in studying networks of neurons and their interactions, which give rise to complex higher-order functions such as movement, perception, and memory. Hence, extracellular recording of neurons has become the most popular technique, as it has the ability to capture the activity of multiple nearby neurons and is relatively easy to implement [2]. Extracellular recordings involve the placement of electrodes in between neurons, that measure the electrical potential changes in the extracellular medium [3, 4]. Such recordings where spikes from more than one neuron are detected are also sometimes termed "multispike train" [5].

Spike-sorting refers to a method that detects individual spikes (action potentials) from extracellular neural recordings and classifies them according to their shapes, which attributes

*Corresponding Author: M.R. Azghadi, mostafa.rahimiazghadi@jcu.edu.au
T. Zhang is with the Department of Bioengineering, McGill University, Montreal H3A 0E9, Canada
M. Rahimi Azghadi and C. Lammie are with College of Science and Engineering, James Cook University, QLD 4811, Australia
A. Amirsoleimani is with the Department of Electrical Engineering and Computer Science, York University, Toronto ON M3J 1P3, Canada
R. Genov is with the Department of Electrical and Computer Engineering, University of Toronto, Toronto M5S, Canada

detected spikes to the originating neurons. Spike-sorting algorithms operate on the principle that different neurons tend to produce spikes of different shapes, due to their varying proximity to a recording electrode, as well as their varying morphology of dendritic trees [3]. Preceding the popularization of spike sorting, the pioneering of the field focused on shaped-based multispike train analysis [5].

Earlier implementations of extracellular recordings were performed with a single electrode, and could only detect and sort 3-5 neurons [5, 6]. Newer Microelectrode Arrays (MEAs) have a much higher electrode count and their density, enabling parallel recording of thousands of neurons [7, 8]. Novel electrode setups even allow for a single cell's signal to be picked up by multiple electrodes [3]. Continually improving electrode array implementations enables an increasing number of neurons to be recorded. When paired with spike sorting algorithms, scientists are able to study the behavior of massive sets of individual neurons simultaneously, greatly facilitating the development of novel therapeutic applications and their ability to perform research in the neuroscience field. This trend is reflected in the increase in the number of spike sorting publications correlating with an increase in the number of neurons simultaneously recorded, as shown in Fig. 1.

High-density MEAs have been actively used to perform research involving the restoration of motor functions for tetraplegic and paralysis patients [9, 10], and to mitigate symptoms of neurodegenerative diseases [11]. Furthermore, neuroprostheses based on spike sorting have been developed for animals such as monkeys, which is able to decode motions from their neural recordings [12]. Spike sorting is also used extensively for neuroscience research and to observe neuron population response to stimuli. The stimuli can either be in the form of sensory input such as visual stimuli [13, 14], or artificially introduced neuronal perturbations such as via optogenetic manipulations [15, 16]. For a comprehensive review of opto-electrophysiological techniques and the novel opportunities and challenges associated with this emerging technique, see [17]. Similarly, spike sorting is also used to decode the functions of various groups of neurons [18–20].

A general processing pipeline of current spike sorting systems is depicted in Fig. 2. As this figure shows, first, the raw signals are collected using an MEA with several channels [8]. The processors run spike sorting algorithms that are, conventionally, comprised of three steps [4]: i) Spike detection through pre-processing, ii) Feature extraction, and iii) Classification. The last step classifies each spike to its originating neuron. However, some spike sorting algorithms are not comprised of the three conventional steps. Examples of these include template matching algorithms [21–25], which detect spikes and perform classification at the same time, or Artificial Neural Network (ANN) based models [26, 27], that do not require explicit feature engineering.

An important distinction has been made between online and offline spike sorting systems, with the former referring to algorithms that can sort spikes in real-time, and the latter referring to ones that cannot. The online applicability is governed by several factors, including the mathematical nature of the algorithm, its computational complexity, and the processing latency [29]. Both online and offline processing may be done remotely, or on-site in an implanted processing module [4].

The majority of publications concerning spike sorting revolve around new techniques, or a combination of existing techniques, to improve one or more of the above-mentioned steps shown in Fig. 2. Algorithms can be further classified as supervised [30] or unsupervised [31], with the former referring to algorithms that require training with ground truths, while the latter does not. There are also hybrid approaches that use a mixture of online and offline steps, or a mixture of supervised and unsupervised steps [32].

In addition, spike sorting systems usually suffer from numerous constraints, including the maximum heat dissipation, physical size, and the energy consumption of implantable computing devices [33]. These limitations are especially acute for online real-time spike sorting applications, such as neuroprosthesis, where spike classification tasks need to be performed on-implant, at a speed faster than the rate of incoming spikes. Hence, numerous research efforts [34–36] have been dedicated to reducing computational power consumption and increasing throughput, either by developing novel algorithms minimizing mathematical complexity, or by developing more efficient application-specific hardware, or a combination of the two to achieve better software-hardware co-optimization.

To facilitate the co-design between spike sorting algorithms and to select their suitable hardware given a specific use case such as Brain Machine Interface (BMI), neural prosthetics, or real-time brain monitoring, in this paper, a comprehensive synthesis of the spike sorting systems literature is performed. Comparisons are made among various algorithms as well as among different hardware implementations, giving future researchers an overarching overview of the spike sorting algorithm-hardware options for various applications and settings. We also survey available packages and resources for software and hardware implementations of spike sorting algorithms. Furthermore, we provide our analysis of the field and identify and discuss key challenges hindering future applicability of the spike sorting systems in the algorithm-, hardware-, and applications-level. The structure of our paper is shown in Fig. 3.

II. Motivations and Related Work

As the field of neuroimplants and neuroengineering continue to advance and improve, spike sorting is rapidly evolving from theoretical concepts to hardware systems capable of real-world applications. Along with this development, the research focus that used to be mainly around pure algorithm developments has evolved and broadened to hardware-algorithm co-optimized designs that are more practical and more reconcilable with the physical constraints associated with neural engineering applications. Therefore, this survey aims to provide a comprehensive overview and analysis of currently available spike sorting algorithms, as well as hardware deployment technologies, which are often neglected by previous review works. In addition, properties of various hardware and algorithm combinations will be identified and discussed.

Several comprehensive reviews have been published for spike sorting. Among these, [4, 37, 38] have attracted the most

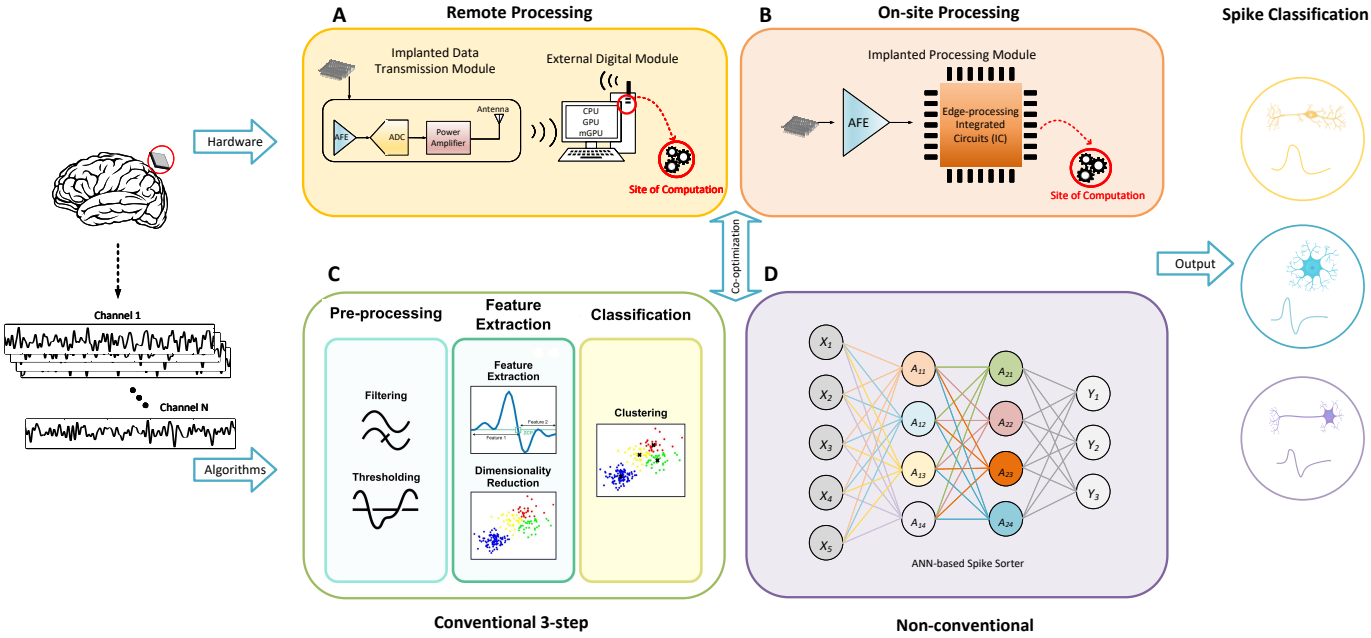

Fig. 2: A generalized spike sorting pipeline is demonstrated, from extracellular recordings to isolated spike trains. After collection of multi-channel recordings [28], suitable processing hardware (A, B) and sorting algorithm(s) must be identified (C, D). Hardware implementations can perform remote processing, as shown in (A), or on-site processing, as shown in (B). Remote processing is typically performed using implantable data transmission modules, and an external digital module (A), while on-site processing is usually performed using an analogue-front-end, which feeds the signal into a highly-efficient processing module. This is commonly realized in hardware using novel ASIC or IMC technologies. Sorting algorithms can generally be categorized as either conventional 3-step, as shown in (C), or non-conventional, as shown in (D). Conventional 3-step algorithms consist of preprocessing, feature extraction, and classification stages [4], while non-conventional ANN-based algorithmsc[26, 27] or template matching algorithms [22–24] do not have clearly defined stages.

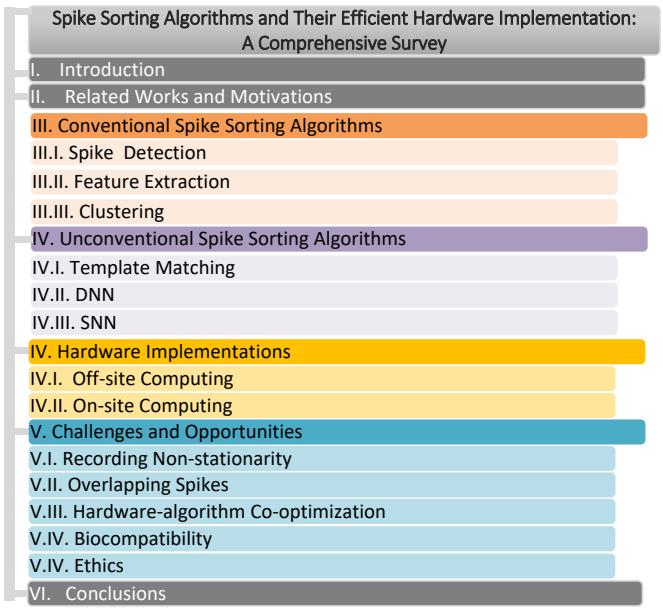

Fig. 3: Paper outline.

citations and attention. These studies mostly cover the algorithm aspect only and mainly only cover the 3-step conventional algorithms in detail. However, with the rapid emergence of unconventional algorithms that remove the clear distinction between detection, feature extraction, and classification, these reviews are no longer fully representative of the current spike sorting systems literature. Hence, in addition to our unique hardware-conscious approach, our paper complements the current literature and expands upon the previous works by including more novel algorithms.

Table I provides a comparison between our work and other prominent spike sorting reviews, demonstrating the areas that have been expanded upon and the impact of our work. As the Table demonstrates, none of the previous spike sorting review papers has covered hardware technologies. Similarly, none of these works have investigated the use of more advanced processing algorithms such as Convolutional Neural Networks (CNNs) and Spiking Neural Networks (SNNs). Furthermore, the co-optimization of the hardware and software has not been discussed in previous studies. These identified gaps motivated us to comprehensively review the literature to provide a new treatise to help neural engineers and scientists better perceive the field from both the hardware and software perspectives.

III. Conventional Spike-Sorting Algorithms

As mentioned in Section I, spike sorting algorithms can be typically categorized into ”conventional”, which follows the ”spike detection, feature extraction, clustering” procedure, or ”non-conventional” algorithms where the steps are not

TABLE I: Comparison with previous spike sorting review publications

Year	Reference	Spike Detection	Feature Extraction	Clustering	Other Algorithms	Hardware
1999	[37]	Amplitude threshold	PCA	KNN,K-means,Baysian	Template Matching, ICA.	NA
2008	[38]	Amplitude threshold, NEO, SWTP	PCA,DWT,DD,IT	NA	NA	NA
2011	[39]	Amplitude threshold, NEO, SWTP	PCA, DWT, DD, it	K-means, valley-seeking, SPC	Osort, Noise estimation, Gaussian, Baysian statistical models.	NA
2015	[4]	Amplitude threshold, NEO, fuzzy theory	PCA, DWT, combination features	K-means, Gaussian mixture, SPC.	Template matching	NA
2016	[3]	Pre-processed multi-channel detection	NA	NA	Template matching, Multi-channel template matching	NA
2022	This	Amplitude threshold, NEO, SWTP	PCA, DWT, geometric, salient, combination	K-means, GMM, Mean shift, SPC, Fuzzy C-means	Template matching, CNN, SNN	CPU, GPU, FPGA, ASIC, IMC

distinctly separable. This section is dedicated to providing the theoretical and mathematical background for the conventional algorithms covered in this review.

A. Spike Detection

Spike detection refers to the step of a spike sorting algorithm that isolates individual spikes from continuous neural recording. Bandpass filtering is often first used to eliminate high-frequency artefacts and low-frequency noise [4]. For online applications, a causal filter is required as the user does not have access to future time samples, while non-causal filters are preferred for offline analysis for their greater versatility. Typically, a spiking event is defined to be 1ms-3ms. Given a common recording sampling frequency of 30,000 Hz, an isolated spike is 30-90 samples in duration [4]. Currently, 3 automatic detection techniques are widely accepted, all of which are covered in this survey. Each technique offers varying advantages and disadvantages, as well as requiring a different level of a priori assumptions, which are discussed in their respective subsections.

1) Amplitude Thresholding: The simplest automatic algorithm is applying an amplitude threshold to the filtered signal. This operates on the principle that depolarization and repolarization stages of a nearby neuron firing will cause sharp increases and decreases in voltage measured from the extracellular medium, which can be used to detect spiking events. A threshold value too stringent will lead to missed spikes for lower amplitude spikes, while a threshold value too lenient leads to false positives. Due to varying experimental setups and signal-to-noise ratios (SNR), the threshold must vary across different experiments and should be set automatically relative to the estimate of signal noise. The most commonly accepted estimate was proposed in [40], which assumes the noise component of the signal is normally distributed and the probability of the spike component is small compared to the noise component. The proposed threshold value and the estimated noise standard deviation are shown in Eq.(1), where $x[n]$ evaluates sampled (discretized) data points of the signal.

$$\begin{aligned} Thr &= 4\sigma_N, \\ \sigma_N &= \text{median}\left\{\frac{|\text{x[n]}|}{0.6745}\right\} \end{aligned} \tag{1}$$

2) Nonlinear Energy Operator: The Nonlinear energy operator (NEO), sometimes referred to as the Teager energy operator (TEO), is a more powerful detection method especially under low SNR [41] as it utilizes both the frequency and amplitude information. The mathematical definition of continuous time NEO ψ is shown in Eq. (2) for sampled signal $x(n)$. The discrete-time version is shown in Eq. (3). Additionally, some studies recommend convolving the NEO time series with a smoothing window to eliminate spurious peaks, as NEO only considers 3 points and is prone to noise [30, 42, 43].

$$\psi(x(t)) = \left(\frac{dx(t)}{dt}\right)^2 - x(t)\left(\frac{d^2x(t)}{dt^2}\right) \tag{2}$$

$$\psi[x(n)] = x^2(n) - x(n+1)\cdot x(n-1) \tag{3}$$

It has been shown that there is an instantaneous increase in both signal amplitude and frequency when spikes are fired [41], which is reflected by an increase in NEO. This method improves upon the simple amplitude threshold method as the latter only considers increases in power during spiking events but not frequency. Unlike the amplitude threshold, the NEO detection threshold is less commonly agreed upon; several publications opted for manual tuning based on the specific data collected from the experimental setup and did not explicitly provide an equation [30, 43, 44]. However, some publications such as [41] provided an automatic threshold equation shown in Eq. (4), which can be used as a starting point before tuning.

$$Thr = C\frac{1}{N}\sum_{n=1}^{N}\psi[x(n)], \tag{4}$$

3) Wavelet Transform Product: Wavelet transform has also been proposed for spike detection [45–47], due to its ability to incorporate information from both the time domain and the frequency domain. Both the use of discrete wavelet transform (DWT) and continuous wavelet transform (CWT) [48] techniques have been proposed, but the former is more commonly cited due to its wider applicability and simplicity. Unlike previous techniques, this technique does not assume that extracellular noise is white noise, which has been shown to be a naive assumption [48]. The conceptual basis for this technique relies on the fact that mother wavelet functions are finite-length "spiky" waveforms. When used for sliding wavelet decomposition, it can be considered as a "template matching" process that assesses the similarity between the wavelet and the segment it encounters. Hence, the wavelet algorithm will output high values when the signal segment displays a high resemblance to the wavelet function. There are various choices for the mother wavelet function as shown in Fig. 4 (A), each with various shape which exhibits a difference in time and frequency domain behaviors. It is reasonable to deduce that a wavelet resembling the neuronal spikes should be chosen such as the Haar wavelet, Daubechies wavelet, and Biorthogonal wavelet. The mathematical definition of wavelet transform is shown in Eq. (5), where ψ is the wavelet function, τ is time translation, and α is a scaling factor. This definition closely resembles that of the correlation equation between the function of interest $x(t)$ and a translated/scaled version wavelet function. DWT refers to algorithms where the wavelet function can only take on a finite number of transformations, and α can only take values 2^j where $j = (1, 2, ...5)$. The wavelet coefficients W(α, τ) for each value of j is calculated for each time point and summed across all time points. The value of J that yields the greatest sum of the coefficient is termed $j^{(\max)}$. Next, $P(n)$, the point-wise product of wavelet coefficients over 3 consecutive scales up to $j^{(\max)}$ is calculated, for each time sample as shown in Eq. (6). Similar to the previous technique, a smoothing window can be convolved with the output $P(n)$ time series to mitigate spurious peaks due to cross terms and background noise.

$$W(\alpha, \tau) = \int_{-\infty}^{\infty} x(t) \frac{1}{\alpha^{1/2}} \psi \left(\frac{t - \tau}{\alpha} \right) dt \tag{5}$$

$$P(n) = \prod_{j=j_{\max}(-)2}^{j_{\max}} \left| W(2^j, n) \right| \tag{6}$$

From the three aforementioned spike detection techniques, the first two are more commonly used for online implementation, due to their lower computational resource requirement for hardware implementation. However, they are prone to reporting false-positives and false-negatives with evolving noise levels [49]. Consequently, in environments where sufficient computational resources are available, and high-accuracy is critical, the third technique may be preferable. We note that many algorithms have been proposed to utilize the distributed nature of spike amplitudes, widths, and frequencies, to detect lower amplitude spikes in high-noise environments [50].

B. Feature Extraction

The feature extraction step isolates representative features that best separate different spike classes [5]. Some feature extraction algorithms are coupled with additional dimensionality reduction steps to output 2 to 3 final features that are used as inputs to the clustering algorithms. This process aims to maximize separation between classes and addresses the "curse of dimensionality" [51], which ensures that clustering algorithms such as K-means can be effectively implemented in the next step.

1) Principal Component Analysis (PCA): PCA is originally proposed as a dimensionality reduction method that aims to faithfully represent the original data in lower dimensions, by projecting the original higher dimensional data points to lower dimensional principal components that capture the maximum variance within the data [52]. In the context of spike sorting, PCA has been the most common feature extraction method due to its simplicity and wide applicability. All time samples of the detected spikes are used as inputs $x(n)$, eigenvalue decomposition is performed on the covariance matrix, and the eigenvectors represent the directions onto which the original data is projected. The principal component coefficients are calculated as shown in Eq. (7)

$$c_i = \sum_{n=1}^{N} PC_i(n) \cdot x(n). \tag{7}$$

Principal components can also be obtained by alternative methods such as Singular Value Decomposition (SVD) and Hebbian Learning. Despite its popularity, PCA suffers from two main limitations. Firstly, PCA only extracts principal components that capture the maximum variance within the dataset, however, that does not guarantee maximum separation between spike groups. Secondly, PCA's effectiveness is dependent on spike alignment [53].

2) Wavelet Transform: Wavelet transform is a time-frequency representation of a signal, commonly used for signal analysis. Wavelet transform is similar in concept to Fourier Transforms, except that Fourier transform maps the time-domain function purely into frequency domain causing loss of specific time-localization features [54], while wavelet transform retains both frequency and time domain information. Wavelet transform uses finite mother wavelet functions as basis functions and expresses the original function as a linear combination of wavelets. The wavelet functions $\psi_{j,k}$ form the orthogonal basis space. The wavelet transform is defined as the convolution between the original signal $x(t)$ and the wavelet function $\psi_{j,k}$. In this paper, we mainly focus on discrete wavelet transform due to its simplicity and popularity. By convolving the original signal with wavelet functions of different parameters derived from the mother wavelet, details of a signal at different resolutions can be quantified. This algorithm is named the multiresolution decomposition [54]. Various mother wavelets can be used, amongst which the Haar wavelet and the Daubechies wavelets are the most popular for analyzing neurophysiological recordings due to their compact support and orthogonality, which allows for discriminative features of the spikes to be identified by a few wavelet coefficients without

a priori assumptions on the spike shapes [40, 47, 55]. Common wavelet families are shown in Fig. 4 (A).

Multiresolution decomposition requires the input vector to be of size 2^p and yields an output wavelet feature vector of the same size as the input vector. The output vector is obtained by convolving the input vector with a cascade of wavelet functions and downsampling as shown in Fig. 4 (B). Each step of the cascade involves a "high pass" wavelet function and a "low pass" wavelet function then downsampling to half of the original vector length. The vector that results from the operation with the high pass filter is temporarily stored, and the vector that results from the operation with the low pass filter is passed to the next step of the cascade. By the end of the cascade, the output vector, the combination of the outputs from all the high pass filters, is the same size as the input vector.

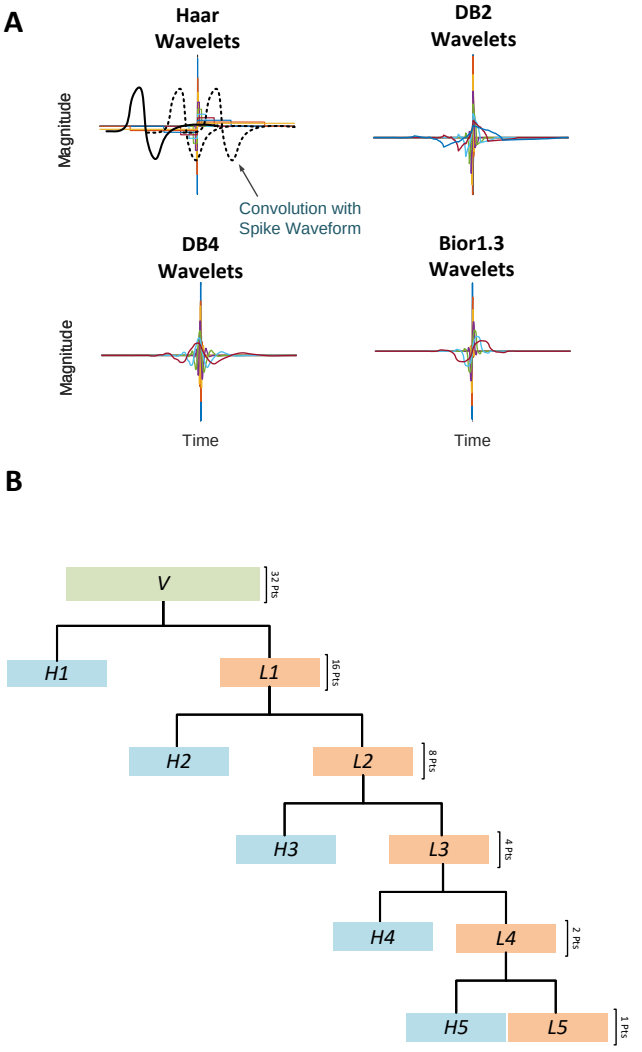

Fig. 4: A) Common wavelet families. B) Processing pipeline for multiresolution wavelet decomposition with cascading filter banks, where Hs represent high pass filter bank matrices and Ls represent low pass filter bank matrices.

3) Geometric Features: Geometric features refer to a general class of features based on the shapes of the detected spikes. These features were popular in the earlier days of spike sorting development due to their lower computation complexity. However, these algorithms are still relevant today, especially for online real-time applications that require on-implant computations where limited computing resources pose significant hurdles. More details regarding hardware opportunities for such algorithms will be explored in Section V. Here, we cover several prominent geometric features. The most visually distinctive geometric feature include the relative amplitudes of spike lobes, the peak to trough amplitude difference, as well as the position of the peak and trough of the spike. However, these amplitude-based features are shown to have quickly degrading quality with the increase of noise [35].

Hence, improved geometric features involving the area of various sections of the spike were proposed. The methods utilizing the areas under the positive and negative sections of the spike is sometimes referred to as the "integral Transform" (IT) [56], which is shown in Eq. (8), where the boundaries of the "positive" lobe, Na, and the boundary of the "negative" lobes, Nb are parameters that need to be trained.

$$I_A = \frac{1}{N_A}\sum_{n=n_A}^{n_A+N_A} x(n), \quad I_B = \frac{1}{N_B}\sum_{n=n_B}^{n_B+N_B} x(n). \tag{8}$$

Zero-crossing features (ZCF) shown in Eq. (9) are another geometric feature type. Unlike IT which requires parameter training, ZCF's lobe boundaries are adaptive, since the time point of the zero crossing event differs with each spike [57][35]. The two algorithm's similarities and differences are visualized in Fig. 5.

$$ZC1 = \sum_{n=0}^{Z-1} x(n), \quad ZC2 = \sum_{n=Z}^{k-1} x(n) \tag{9}$$

4) Derivative based features: Similar to geometric features, derivative-based features are also loosely based on the shape of the spike in the time domain. Here, we cover the most popular form of derivative-based feature extraction method, i.e. the First and Second Derivative Extrema (FSDE) proposed in [58]. This family of algorithm is one of the first implemented for multi-channel on-chip spike sorting applications, due to their superior efficiency-accuracy balance [31] . In this approach, the first and second derivative of every sample point of the detected spike is calculated with Eq. (10), where time series FD represents the first derivative, SD represents the second derivative, and S is the detected spike. The positive peak of FD, the negative peak of SD, and the positive peak of SD are the 3 features extracted.

$$\begin{aligned} \mathrm{FD}(n) &= s(n) - s(n-1) \\ \mathrm{SD}(n) &= \mathrm{FD}(n) - \mathrm{FD}(n-1) \end{aligned} \tag{10}$$

5) Salient Features: The salient features technique was first proposed by Shaeri and Sodagar in [32]. Salient features are defined to be a set of features that maximizes discrimination between spike classes. A subset of 2 features from a K-features feature space is extracted for each class to maximize the discrimination between the class and all other classes. The feature space is defined to be time point samples of

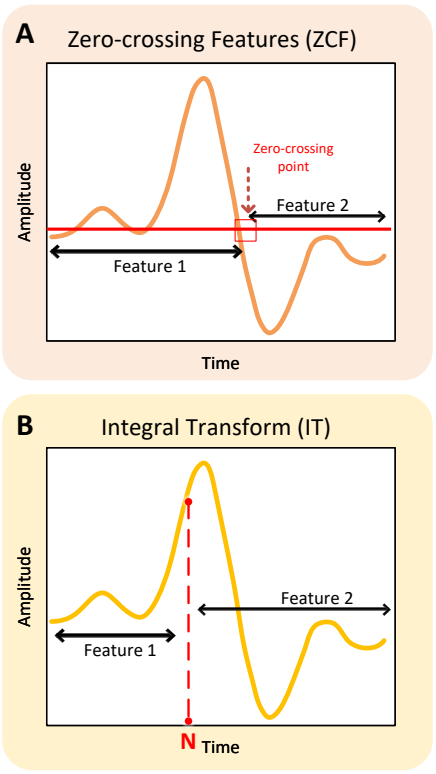

Fig. 5: Illustration comparing and contrasting the two most widely-used geometric features in spike sorting.

extracted spikes, and a subset of 2 salient features i and j are identified within the feature space. A discrimination index is defined in Eq. (11), as a measure of normalized distance between spike classes, where (μ_i,μ_j), (σ_i, σ_j) and (P_i,P_j) are the means, standard deviation, and probability of occurrences, respectively.

$$d_{ij} = \mathrm{e}^{\frac{|\mu_i - \mu_j|}{\sqrt{P_i \sigma_i^2 + P_j \sigma_j^2}}} \tag{11}$$

The salience of a feature is defined as the product of both its ability to discriminate the class of interest and its ability to keep the homogeneity of the distribution of all other classes with respect to the class of interest. The former is measured by the geometric mean of associated distances, and the latter is measured by the ratio of the geometric mean to the arithmetic mean. The overall product is shown in Eq. (12).

$$\varsigma_i[k] = \frac{\left(\prod_{j=1(j\neq i)}^{N_c} \left(d_{ij}[k]\right)^{P_j}\right)^2}{\sum_{j=1(j\neq i)}^{N_c} P_j \times d_{ij}[k]} \tag{12}$$

The first salient feature is the feature with the highest ς_i value, while the second salient feature is defined as the most uncorrelated feature to the first one that also best isolates the class of interest, as shown in Eq. 13, where ρ is the correlation between the k'th feature and the h'th salient feature.

$$\begin{aligned} k_i^1 &= \underset{\kappa\in\{1,2,..K\}}{\arg\max} \{\varsigma_i[\kappa]\} \\ k_i^2 &= \underset{\kappa\in\{1,2,..K\}}{\arg\max} \{\varsigma_i[\kappa] \times (1 - \rho_i(\kappa, 1))\} \end{aligned} \tag{13}$$

6) Linear Discriminant Analysis (LDA): Similar to PCA, LDA is also a linear projection technique initially proposed by Fisher [59]. LDA functions on the principle of explicitly minimizing intra-class variance shown in Eq.14 and maximize inter-class variance shown in Eq.15, where x_i is the i th data point in the k th cluster C_k, μ_k represents the mean value of data points in k th cluster, μ represents the mean of all data points, n is the total number of data points.

$$S_w = \sum_{k=1}^{K} \sum_{x_i \in C_k}^{n_k} (x_i - \mu_k)(x_i - \mu_k)^T \tag{14}$$

$$S_b = \frac{\sum_{k=1}^{K} n_k (\mu_k - \mu)(\mu_k - \mu)^T}{n} \tag{15}$$

The cost function as in Eq.16 is optimized by updating matrix W and then class label i iteratively. LDA has recently emerged in the spike sorting field [60, 61] due to its ability to ability to provide better class separability especially in high-noise recordings.

$$J = \frac{\mathrm{tr}\left(W^T S_b W\right)}{\mathrm{tr}\left(W^T S_w W\right)} \tag{16}$$

7) Combination Features: Some newly proposed algorithms use a combination of multiple aforementioned features, using Fuzzy logic or adaptive probabilistic weights to extract the final features that provide the best separation [44, 62, 63]. Such solutions can mitigate the shortcomings of individual feature extraction algorithms, thereby, they are very effective for increasing accuracy typically at the expense of higher computing resource requirements.

C. Clustering

The final step of the conventional spike sorting pipeline is clustering the extracted spikes that have been projected onto the feature space. The technique chosen should be efficient, accurate, and require minimal user intervention. It is often assumed that the variations within clusters are caused by noise superimposed onto the true spike waveforms [64]. Hence, the majority of clustering algorithms assume Gaussian clusters such as K-means and Gaussian Mixture Model. Conversely, some methods that do not make such assumption, such as Superparamagnetic Clustering. Generally clustering algorithms with Gaussian cluster assumptions perform worse since most neural recordings do not have stationary shape with uncorrelated noise [37].

1) K-means: K-means clustering is one of the most commonly used techniques of unsupervised learning due to its simplicity. This algorithm involves setting K random initial centroids, which are centers for each respective cluster. In each iteration, each data point is first assigned to a cluster based on the shortest Euclidean distance, then the cluster mean is calculated and used as the new centroid. This process is repeated until the centroid location stabilizes [65]. A challenge with using K-means is that it requires a predefined number of clusters K, which can be manually determined from visual inspection. The final results are hard classification results, meaning that definitive cluster memberships are assigned to each data point.

2) Expectation Maximization (EM) Gaussian Mixture Model (GMM): Expectation Maximization (EM) clustering algorithms are similar in nature to K-means as they also involve the assignment step followed by the update step. However, EM provides soft classification results, which assigns the probability of belonging to a certain cluster to each data point [66]. Typically, multivariate Gaussian probability functions are fitted to the dataset. This is advantageous in many cases as multivariate Gaussian can be adapted to more cluster shapes compared to K-means and it enables more refined tuning opportunities such as determining outliers. The probability of belonging to cluster k for a datapoint is termed the "responsibility", calculated by Eq. (17), given the mean of each Gaussian cluster (μ), the variance of each Gaussian cluster (Σ) and the relative probability of each cluster (π). After assigning "responsibilities" to each datapoint, the 3 parameters are updated according to Eq. (18). This model has been widely adopted for spike clustering [64] due to experimental results showing its ability to reasonable model the noise.

$$p\left(z=k \mid x^{(n)}\right)=\frac{\pi_k \mathcal{N}\left(x^{(n)} ; \mu_k, \Sigma_k\right)}{\sum_c \pi_c \mathcal{N}\left(x^{nn} ; \mu_c, \Sigma_c\right)} \tag{17}$$

$$\begin{aligned}
\mu_k &\leftarrow \frac{1}{\sum_n r_{n,k}} \sum_n r_{n,k} x^{(n)} \\
\Sigma_k &\leftarrow \frac{1}{\sum_{n'} r_{n',k}} \sum_n r_{n,k}\left(x^{(n)}-\mu_k\right)\left(x^{(n)}-\mu_k\right)^{\top} \\
\pi_k &\leftarrow \frac{\sum_n r_{n,k}}{N}
\end{aligned} \tag{18}$$

3) Mean shift: Mean shift is a centroid-based clustering algorithm; it improves upon K-means by eliminating the need for a predefined number of clusters K or the need to assume the cluster distributions [67]. The algorithm first estimates a kernel density function (KDF) shown in Eq. (19), where k represents the symmetrical kernel function, often Gaussian.

$$f(x)=\sum_i K\left(x-x_i\right)=\sum_i k\left(\frac{\left\|x-x_i\right\|^2}{h^2}\right) \tag{19}$$

Once a kernel value has been assigned to each datapoint, the algorithm iteratively "shifts" the datapoints "up" the KDF map according to the gradient function in Eq. (20), where $g(x)$ is the first derivative of $k(x)$ [67].

$$\nabla f(x)=\frac{2 c_{k,d}}{n h^{d+2}}\left[\sum_{i=1}^n g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)\right]\left[\frac{\sum_{i=1}^n x_i g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)}-x\right] \tag{20}$$

Higher values of h lead to "steeper" KDFs which results in more clusters. Convergence is reached when the gradient approaches 0. The challenge with mean shift compared to other algorithms is its increased computational complexity of $O(n^2)$.

4) Superparamagenetic Clustering: Superparamagnetic Clustering (SPC) is a non-parametric clustering technique pioneered by [68], and has been popularized for spike sorting applications by Rodrigo *et al.* [40]. SPC does not involve a well-defined centroid, nor a cluster distribution function. This algorithm is inspired by the interactions between nearby superparamagnetic particles under various temperatures. Interactions are first simulated mathematically between different points X_i and X_j by Eq. (21), where a is the average distance between nearest neighbors.

$$J_{ij}=\begin{cases}\frac{1}{K} \exp\left(-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{2a^2}\right) \\ 0.\end{cases} \tag{21}$$

Next, a "Potts spin" state variable s from 1 to q is assigned to each point x_i, where q is normally chosen as 20 as in [40, 68], followed by N Monte Carlo iterations at different temperatures using the Wolf Algorithm. The state value is updated based on the probability function in Eq. (22), where T is the temperature.

$$p_{ij}=1-\exp\left(-\frac{J_{ij}}{T}\delta_{s_i,s_j}\right), \delta_{s_i,s_j}=\begin{cases}1 & \text{if } s_i=s_j \\ 0 & \text{otherwise}\end{cases} \tag{22}$$

Evidently, closest neighbors with larger J_{ij} will have a higher probability of updating s together. Neighbors update their s values iteratively until they no longer change. This process is repeated with respect to other points. A number v is generated from a uniform distribution [0,1], if $v < pij$, a "bond" is established. A cluster is defined as all points connected by "bonds". A variable $c^{m_{ij}}$ is defined as 1 if x_i and x_j are in the same cluster, or 0 otherwise. C_{ij} is defined as the mean of all $c^{m_{ij}}$ for all 1 to M different temperature iterations. Then, the spin-spin correlation function is calculated with Eq. (23). A threshold θ is set as a parameter, if $G_{ij} > \theta$ then x_i and x_j belongs to the same cluster.

$$G_{ij}=\frac{(q-1)c_{ij}+1}{q} \tag{23}$$

5) Fuzzy C-means: Fuzzy C-means (FCM) is another commonly used soft classifier first applied to spike sorting in [69], meaning that the results are in the form of probabilities. The number of clusters k is usually predetermined, but it can also be adaptive by applying a variation of the FCM algorithm [70]. The iterative update process is similar to previous clustering techniques. The algorithm first randomly initializes cluster membership values μ_{ij} for each point X_i with respect to cluster center C_j. The cluster centers are then calculated according to Eq. (24), where D is the number of data points, N is the number of clusters, and m is the fuzziness parameter.

$$c_j=\frac{\sum_{i=1}^D \mu_{ij}^m x_i}{\sum_{i=1}^D \mu_{ij}^m.} \tag{24}$$

The clusters memberships are then updated according to Eq. (25).

$$\mu_{ij}=\frac{1}{\sum_{k=1}^N\left(\frac{\|x_i-c_j\|}{\|x_i-c_k\|}\right)^{\frac{2}{m-1}}}. \tag{25}$$

Experimentally, FCM typically performs better than K-means as it provides soft cluster memberships, however, it tends to require more computational power. Furthermore, it is more suitable to be applied in online clustering applications due to its ability to account for non-stationarity during the defuzzification process [71].

A visual comparison between the 4 most commonly used feature extraction algorithms discussed above is shown in Fig. 6, using criteria proposed in [29]. This figure demonstrates that there generally exists a trade-off between the different criteria, each new development in the algorithm tends to focus on optimizing one or multiple aspects while striking a balance with the others. A more well-rounded algorithm is desirable for real-world application, and the selection of which algorithm to use should be based on the importance the researcher assigns to each criterion given the application settings.

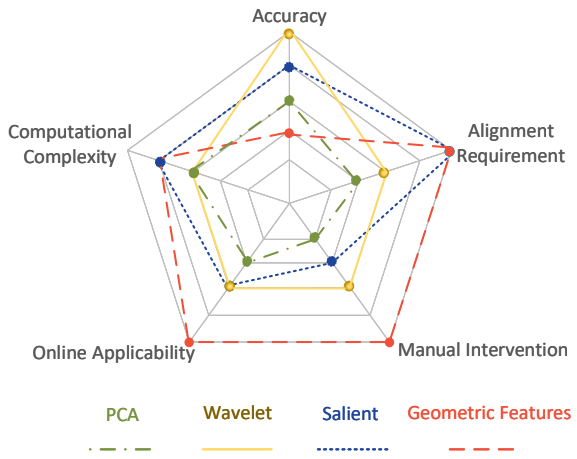

Fig. 6: Spider diagram comparing the 4 most commonly used feature extraction methods for spike sorting.

IV. Unconventional Spike-sorting Techniques

Some novel algorithms have been proposed that do not follow the conventional "spike detection, feature extraction, and clustering" pipeline. Instead, these algorithms are able to perform two or more of these steps simultaneously, in an effort to increase efficiency and accuracy. In this survey paper, we refer to such algorithms as "unconventional" and discuss their three main categories below.

A. Template matching

Similar to wavelet transform product detection and wavelet transform feature extraction algorithms, template matching, as shown in the left panel of Fig. 7, detects spikes by using a sliding template to perform linear discriminant analysis and detect segments of neural recording that provide a high resemblance to the template spike. Different approaches to determining the measure for resemblance have been proposed. In this review, we focus on one of the most recent and streamlined implementations proposed in [72], i.e. the normalized-template-matching (NTM) algorithm. As with other template matching techniques, an initial template selection is required to extract mean waveforms for each class. This requires the use of another spike sorting algorithm such as thresholding with PCA as well as manual adjustments. Subsequently, the template waveform vector is sled along the neural recording and the cross-correlation between the signal segment of interest and the template is calculated with dot product as shown in Eq. (26), which can be rewritten as in Eq. (27), where μ is the template waveform vector and V(t) is the signal segment. The term Cosine(θ) measures the degree of correlation, where a value close to 1 represent strong similarity, while a value close to 0 represents low similarity. If Eqs. (26) and (27) are rearranged, the correlation index $x(t)$ is derived in Eq. (28). A threshold for the correlation index $x(t)$ is then calculated on a per-experiment basis that maximizes detection accuracy.

$$C_i(t) = V(t) \cdot \mu_i^T \tag{26}$$

$$V(t) \cdot \mu_i^T = \|V(t)\| \left\|\mu_i^T\right\| \cos(\theta) \tag{27}$$

$$x_i(t) = \frac{C(t)}{\|V(t)\| \left\|\mu_i^T\right\|} = \cos(\theta) \tag{28}$$

This technique requires a priori knowledge of the waveforms, which leads to requiring offline training or manual intervention. Despite being similar in nature, wavelet transform techniques use multiple dilated or shrunken variations of the mother wavelet and employ Kolmogorov–Smirnov (KS) tests to determine the coefficients that correspond to the wavelet variation that provides the best separation. This requires fewer prior assumptions compared to template matching techniques, which have fixed templates for each predetermined class of spikes. This may cause the performance of template matching algorithms to suffer in real-world applications, as neural recordings are considered non-stationary, hence the templates may need to be re-calibrated frequently.

There exists more advanced template matching algorithms such as the ones introduced in [22–25], which more effectively counter confounds including the issue of temporally and spatially overlapping spikes and waveforms shift due to electrode movement relative to the tissue. These algorithms function on the basis of decomposing the extracellular recording as a sum of templates as in Eq. (29), where $\vec{x}(t)$ is the recorded raw waveform over multiple electrodes, $\vec{w}_j$ is the template, t_i is the putative spike time, a_{ij} is the amplitude factor for spike time t_i for cluster j, and $\vec{e}(t)$ is the background noise component. The templates are generated either prior to the matching process, similar to the method mentioned in the previous paragraph, or they may be generated iteratively, like a k-means process as in [22].

$$\vec{x}(t) = \sum_{ij} a_{ij}\vec{w}_j\left(t - t_i\right) + \vec{e}(t). \tag{29}$$

During the classification process, most algorithms involve a greedy approach that finds the template that matches the raw data given a certain acceptance criterion. If accepted, the template is subtracted from the raw data, then the process is repeated as shown in Fig. 7. This iterative procedure can also be incorporated after a 3-step conventional pipeline has been used to extract the templates, then the signals are reconstructed as a linear sum of the templates, which can identify overlapping spikes and detect previously undetected low SNR spikes [25, 73].

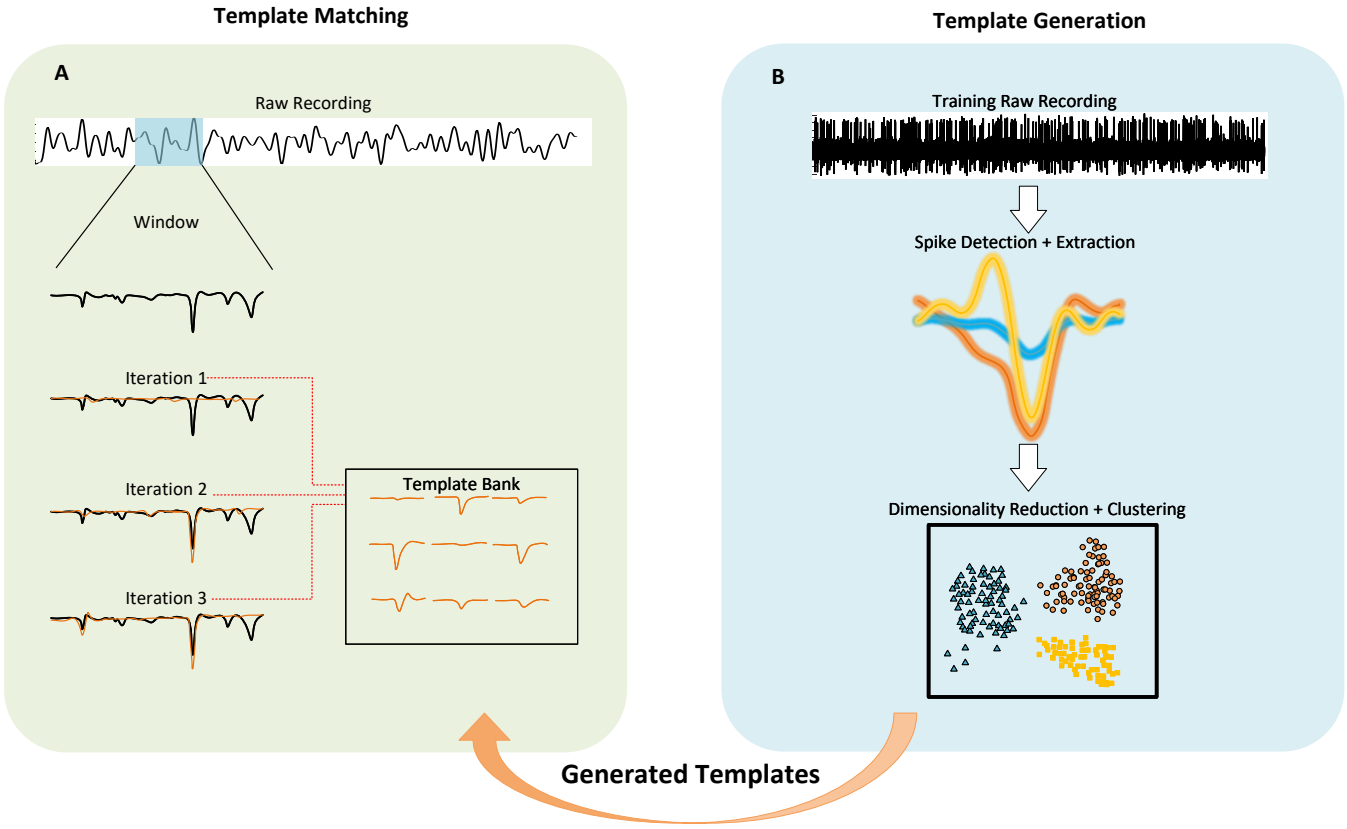

Fig. 7: (A) shows the template generation process, where conventional algorithms are typically first deployed to isolate the spike trains and generate the templates. (B) demonstrates how the generated templates make up the template bank, which are then used to represent the signal as a combination of spike waveform templates by iteratively matching the signal with templates.

B. Deep Neural Networks

Recent astonishing developments in both deep learning algorithms and hardware have led to an increasing number of deep learning methods being applied to spike sorting applications [26, 27, 74–76], including the usage of both convolutional neural network (CNN) and Recurrent Neural Network (RNN). In one of the first attempts to use neural networks for spike sorting, [30] proposed a simple approach, where each time point sample from the extracted spike was inputted as a feature into a 3-layer artificial neural network with 10-12 hidden nodes, and the training was performed with adaptive learning rate backpropagation.

Recent instances of deep ANN implementations are no longer limited to classifying single spikes. Instead, they leverage the size of DNN to take input data from all recording channels. This is especially important for high-density MEA or tetrode recordings, where closely spaced electrodes provide recordings that are not independent of one another. Although some CNN implementations tackle the entire spike sorting pipeline such as in [26, 76, 77], others aim to address more specific steps such as in [26, 27, 78].

In [76], the authors utilize both a CNN and an Recurrent Neural Network (RNN) to perform spike sorting in an end-to-end manner. CNN is responsible for spike detection. The network consists of 2 convolutional layers, followed by 1 max-pooling layer. The first convolution layer receives 128 features corresponding to a single timestep sample from the 128-channel MEA. The output of the CNN is fed as features into a Long Short Term Memory (LSTM) RNNs, which is trained using truncated backpropagation through time (TBPTT) with 20-time steps. The LSTM is able to process temporal information as it receives the stream of outputs from the CNN.

Alternatively, [27] focuses on using DNNs in only one step of spike sorting, which is selecting channels that generate spikes instead of artefacts. In this method, first, a traditional method such as amplitude threshold is used for spike detection. Then, successive detected spikes from a single channel are concatenated into a batch with a typical size of 20. The concatenated batch spike matrix is used as the input into a CNN (4 convolutional layers and 3 pooling layers), which determines if the channel yields spikes or artefacts.

Another demonstrative work that utilizes CNN to tackle a specific spike sorting obstacle is [78]. It resolves the issue about overlapping spikes that traditionally cannot be separated in the feature space. The CNN used is composed of 2 hidden layers operating on a custom cost function, that generates new feature vectors that behave as a linear superposition in the feature space for overlapping spikes. On a related note, [26]

TABLE II: Comparison of ANN Spike Sorting Techniques

Year	Reference	ANN Architecture	Targeted Pipeline
2000	[30]	3-layer MLP	Performs classification for extracted spikes.
2020	[77]	CNN with 4 convolutional layers, 2 pooling layers, 1 FC layer.	Classification of extracted spikes.
2020	[76]	CNN + LSTM RNN	Spike detection with CNN and classification with RNN.
2019	[27]	CNN with 4 convolutional layers and 3 pooling layers.	Automatically detect channels yielding useful spikes instead of artefacts.
2020	[78]	CNN with 2 convolutional layers.	Resolve overlapping spikes into decomposed feature vectors.
2015	[80]	2-layer STDP SNN	Clustering for extracted spikes.
2016	[81]	2-layer STDP SNN with filter signal as input features	End-to-end from detection to clustering.
2019	[82]	2-layer STDP SNN with attention, adaptive thresholding and delayed synapses.	End-to-end with the ability to work with tetrode data.
2018	[83]	Supervised SNN with rate encoding.	Clustering of detected spikes.

is a demonstrative work that deploys modular ANNs during multiple stages of the spike sorting pipeline, which is able to resolve overlapping spikes and denoise the recording with sparse deconvolutions.

While ANNs show immense potential for the new generation of spike sorting systems in terms of improved accuracy and overlapping spike decomposition [75], there are several hurdles that remain to be tackled. Firstly, supervised ANN methods usually require lots of labeled training data, which is not readily available in neural recordings. Second, ANN approaches are often more computation-demanding, posing challenges when deploying in resource-constrained settings such as real-time spike sorters and neural decoders. Additionally, ANNs are usually considered as black-box algorithms that often lack explainability. Lastly, ANNs are prone to overfitting, which is a term in machine learning referring to the model that fits too closely to the training data and lacks generalizability. Overfitting occurs when a model is overly complex, or if the training dataset is small and noisy [79]. This issue can hinder spike sorting as different neural recording datasets can exhibit highly variable properties and noise levels.

C. Spiking Neural Network

SNNs is a type of novel neuromorphic neural network which recently gained attention in the scientific community due to its potential energy efficiency and noise tolerance. It functions more similarly to a biological neural network where the information is passed between neurons in spikes. It has garnered interest, especially in the neuroengineering field due to its potential to directly interface with biological neurons as the neuromorphic computing system draws significant parallels between itself and the biological counterpart [84], both on an algorithm and hardware level as demonstrated in Fig. 8. The parallelism can lead to an increase in efficiency in spike sorting systems not just due to the innately efficient nature of neuromorphic hardware, but also due to the fact that most neuromorphic systems pass information around in time-series data, which allow for the time-series neural recordings to be processed with minimal data conversion.

One of the most widely-used learning algorithms in SNNs is the unsupervised training scheme named Spike Timing Dependent Plasticity (STDP), which is based on the observed mechanism of biological neurons that "neurons fire together wire together" [85]. It is a variation of Hebbian learning [86] which can be broken down into 2 mechanisms: the long-term potential (LTP) and long-term depression (LTD) [87]. LTP increases the synaptic weights when the pre-synaptic neuron is activated before the post-synaptic neuron, whereas LTD decreases the synaptic weight with probability when the post-synaptic neuron is activated before the pre-synaptic neuron. The mathematical depiction is shown in Eq. 30, where Δt is the timing difference between post-synaptic and pre-synaptic neurons, Δw is the synaptic weight change, A^- and A^+ are the synaptic amplitudes, τ_- and τ_+ are the depression and potentiation time constants, respectively.

$$\Delta w = \begin{cases} \Delta w^+ = A^+ e^{\left(\frac{-\Delta t}{\tau_+}\right)} & \text{if } \Delta t > 0 \\ \Delta w^- = -A^- e^{\left(\frac{\Delta t}{\tau_-}\right)} & \text{if } \Delta t \leq 0 \end{cases} \tag{30}$$

The technique proposed in [80] is an earlier iteration of the STDP-based spike sorting system with a winner-take-all (WTA) mechanism, using each time sample of the detected spike has an input feature, leveraging an encoder to achieve an optimal balance between accuracy, stability, and hardware overhead.

A later more comprehensive implementation of spike sorting using a 2-layer SNN is proposed in [81]. In this work, waveforms no longer need to be explicitly detected first. The first layer contains 32 input neurons, each receiving the neural recording as a bandpassed signal filtered with 32 different frequencies. The first layer is then fully connected to 5 output neurons, each corresponding to a class of spikes. Neurons in both layers follow the Leaky Integrate Fire (LIF) model. The filtered analog signals inputted at each neuron transfer through 32x5 synapses that change their weights based on STDP. LIF parameters require manual tuning to suit each possible waveforms so that higher energy in a certain frequency band leads to more input spikes generated.

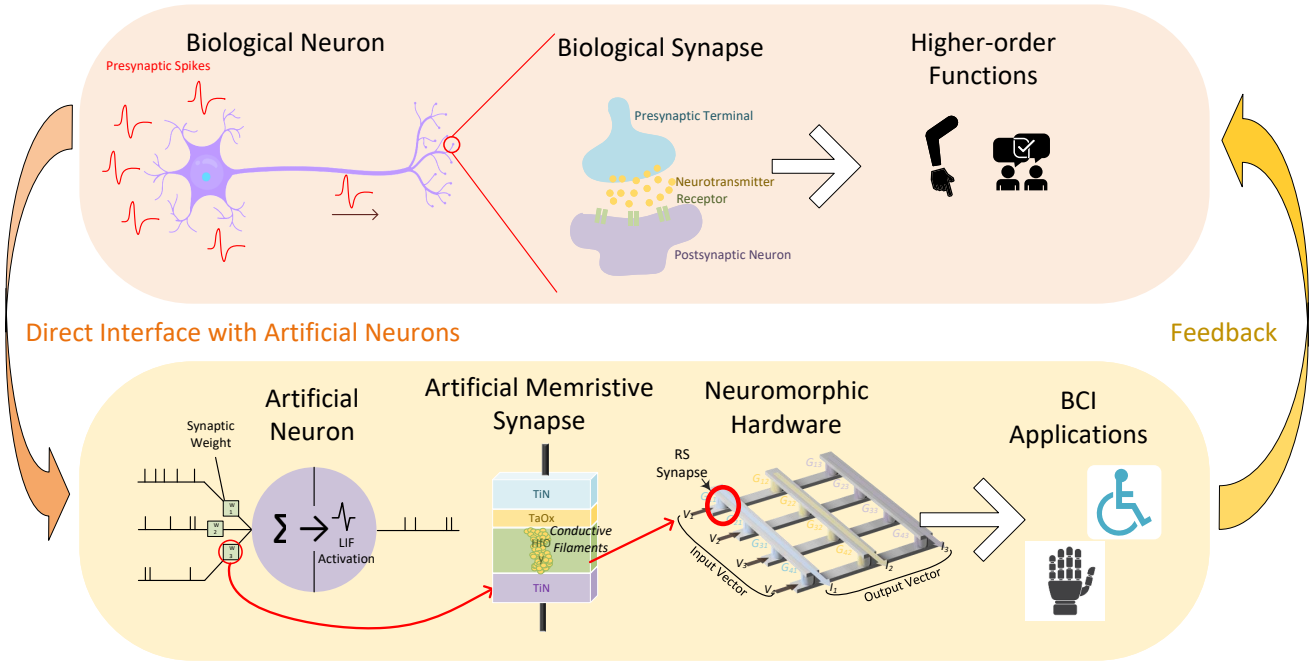

Fig. 8: The pipeline of neural signal processing utilizing neuromorphic hardware. Due to the functional similarities between the artificial neuromorphic hardware and the biological neurons and synapses, neuromorphic artificial neural networks can directly interface with biological neurons and more efficiently decode neural signals for a variety of applications. The interface even allows for feedback to be sent back to the biological neurons.

[82] improves upon the previous STDP architecture by introducing a multi-layered attention mechanism while incorporating delayed synapses and threshold adaptation to counter time-varying patterns. Additionally, this work adapts the network for tetrode recordings, demonstrating the efficacy and efficiency of STDPs in high-density electrode settings.

There are also supervised variants of SNNs proposed for spike sorting [83, 88]. In these works, the detected spike is first rate-encoded, meaning that the analog waveform is converted to a binary spike train whose frequency is proportional to the magnitude of the analog signal. Generally, such algorithms first involve a K-means clustering stage to generate training data, which is subsequently used to train the SNN with a modified loss function.

V. Hardware Implementations

This section will overview the state-of-the-art hardware implementations of the algorithms from the previous section. As shown in Fig. 2, spike sorting hardware is typically designed with one of the two remote or on-site processing methods in mind. We refer to on-site processing as systems that perform spike sorting in real-time at the site of spike collection, without storing or transmitting raw spike waveforms. Such systems focus more on speed, size, and power consumption. On the other hand, we refer to external processing as systems that store or transmit spikes external to the site of collection, either in a real-time or non-real-time fashion. Such systems are typically composed of an on-site spike transmission/compression module and an external processing module, where versatility and precision are emphasized over other constraints.

A. External Processing Systems

1) Conventional CPU/GPU/mGPU: Theoretically, all algorithms from Section III can be implemented on conventional CPU or GPU with varying degrees of optimization. Therefore, in this subsection, we mainly cover the pre-built packages developed for conventional CPU/GPU to process spikes, since efforts have already gone into optimizing for appropriate hardware usage, leveraging the full capabilities of such hardware. A table summarizing key features of different packages is shown in Table III.

a) KlustaKwik: KlustaKwik [89–91] is one of the earlier packages for a streamlined unsupervised offline software-based spike sorting, capable of processing single electrode or tetrode recordings. It is built upon the conventional 3-step spike sorting process. Spike detection is realized with root mean squared power threshold, while features are extracted with PCA, and clustering can be executed manually in the GUI, or automatically using multiple iterations of fitting a Gaussian Mixture Model in order to determine the optimal number of clusters. Updates have been made to the package over the years to offer improvements [89]. Major updates include its new ability to work with higher electrode counts to overcome the problem of dimensionality and the need for human intervention. The detection is updated to use SpikeDetekt which is able to detect temporally overlapping spikes resulting from higher electrode counts with a double-threshold flood fill

TABLE III: Comparison For Existing Pre-built Pipelines

Package	Online Applicability	Multi-channel Applicability	Programming Language	Training Requirements	Conventional 3-step
Osort	✓	✓	Matlab	Training is not required as the clusters are updated iteratively	NA
KlastaKwik	✗	✓	C++ and Python	New iterations of the algorithm mostly eliminates the need for any manual intervention and training	Detection: Power threshold. FE: PCA. Clustering: GMM
WaveClus	✗	✓	Matlab package with comprehensive GUI	No training required, but detection threshold can be manually adjusted	Detection: Amplitude threshold FE: DWT Clustering: SPC
MountainSort	✗	✓	C++ and Python	Requires no training and minimal user intervention	NA
KiloSort	✓	✓	Matlab package, optimized for CPU and GPU acceleration.	Minimal human curation is required, the templates are generated automatically and merged automatically	NA
SpyKING CIRCUS	✗	✓	Python	SpyKING circus reduces the burden of manual curation associated with high electrode-count data. However, spike detection threshold is manually tuned	NA

algorithm. The clustering step is also improved to use a novel "masked EM algorithm", where Gaussian mixtures are fitted, but with each feature vector replaced by a virtual ensemble, in which features with masks near 0 are replaced by noise distribution. This package was initially developed on Matlab for CPU, but recent updates from [89] use a combination of Python and C++, expanding to the use of GPU for the acceleration of larger datasets as well as data visualization.

b) WaveClus: WaveClus has been the benchmark for a pre-built unsupervised spike sorting algorithm, created by Quiroga *et al.* based on their publication in [40], which has been updated in [92]. It is the most widely cited and commonly compared to newer publications. The package is built with MatLab graphical user interface (GUI) and reads directly from time series neural recordings where the user can specify the sampling frequency. The package performs filtering, spike detection, feature extraction, and classification.

The GUI allows users to select the bandpass filter frequencies. Detection is performed with the aforementioned amplitude thresholding technique, where the user has the freedom to manually set the threshold level for fine-tuning the balance between type 1 and type 2 detection errors. Feature extraction is performed with wavelet transform, and finally clustering utilizes the superparamagnetic clustering algorithm.

c) Osort: Another widely used pre-built package that precedes the WaveClus is Osort [93]. Unlike WaveClus, Osort can be implemented in real-time for online applications in addition to its capacity for offline processing. Osort builds the clusters and updates them iteratively over the length of the recording. This MatLab GUI has the benefit of being able to integrate directly with the Neuralynx recording platform or Blackrock microelectrode arrays. Osort also has the advantage over many other online techniques due to its lack of training phase and does not require pre-defined clusters.

Similar to the conventional 3-step algorithms mentioned above, Osort has a discrete spike detection step. Osort employs an energy threshold. The extracted spike is then upsampled using FFT. The number of neurons present as well as the assignment of each spike to a neuron is based on the distance metric defined in

$$d_{\mathrm{S}}\left(\vec{S}_i, \vec{S}_j\right) = \sum_{k=1}^{N} \left(S_i(k) - S_j(k)\right)^2 . \tag{31}$$

As each spike is detected, the distance between the spike and its mean waveforms is calculated. If the distance exceeds a certain threshold, the spike is established as a new class, otherwise, it is assigned to the class with the lowest distance, and the mean waveform for that class is updated.

d) MountainSort: MountainSort [94] is one of the more recently developed packages offering fully automated functionalities requiring minimal pre-defined parameters and human intervention while delivering improved computational speed with modern desktop processors. It is primarily built with C++ and python, taking advantage of multi-threaded processors. MountainSort marks a radical departure from the conventional 3-step algorithms, which often require human intervention to some degree at one or more of their steps [94]. Additionally, this algorithm functions without assumptions for cluster distributions. Mountainsort only relies on the single assumption that the clusters are unimodal in lower dimensional space.

This algorithm prefers using tetrode or high-density microelectrode array recordings as input instead of focusing only on one channel at a time. Mountainsort functions on the principle of performing the spike sorting pipeline separately on neighborhoods first, where a neighborhood is defined as feature space from the central electrode and surrounding electrodes within a pre-defined radius. Each electrode can be defined

as the central electrode, hence the number of neighborhoods equals the number of available electrodes. In the case of tetrodes, each neighborhood comprises of 4 electrodes. Note that higher density microelectrode array recordings can also be used where each neighborhood consists of 6-7 electrodes. After spike sorting has been performed on each neighborhood, the clusters are consolidated to reduce redundancies. To elaborate, for each neighborhood, Mountainsort first performs bandpass filtering with FFT while suppressing high voltage artefacts. Then, a spatially-whitening filter is applied which removes correlations between channels not caused by signals of interest. A spike is detected when an amplitude threshold is exceeded on multiple channels within a short time frame. Feature extraction is performed using PCA to reduce the multi-electrode spike value to a 10-dimensional vector. Clustering is performed using the novel ISO-SPLIT algorithm, a density-based non-parametric clustering algorithm that functions on the principles of unimodal statistical tests [95].

e) KiloSort: KiloSort [22] is a relatively recently developed package that addresses several obstacles in high-density electrode array online applications. It is built upon principles of template matching. Firstly, spatial masking is applied to reduce the dimensionality of spikes and minimize effects from the low SNR channels, utilizing singular value decomposition (SVD) of spatiotemporal waveforms. A generative model is used to obtain the templates iteratively online. The manual curation needs are further reduced with KiloSort by employing a post-hoc template merging. It is built for Matlab on CPUs but has also been optimized for accelerating highly parallelized GPU operations.

f) SpyKING CIRCUS: SpyKING CIRCUS [25] is a recently developed comprehensive toolbox for high electrode count offline sorting. Predominantly, the algorithm contains a clustering step followed by template matching. Spikes are first detected in channels with an amplitude threshold. Detected spikes are initially grouped according to the electrode in order to apply to mask, which assumes a single neuron can only influence the electrodes in its close vicinity, hence only signals from channels close to the peaking channel are kept. Subsequently, PCA projects the spikes onto 5 principal components, followed by a novel density-based clustering algorithm proposed in [96].

In order to overcome the overlapping spikes issue, SpyKING CIRCUS goes further to implement an additional template matching step. The template is first created from each cluster, composed of the average waveform and the direction of the largest variance orthogonal to the average waveform. It is assumed that every variation of a waveform is a linear combination of these two components. A greedy iterative approach is taken next for classification. Given the raw data, a template whose first component had the most similarity to the raw spike is selected, and its amplitude is matched to the signal. If the amplitude falls between the determined thresholds, the two components of the template are matched with and subtracted from the raw signal. This process is iterated until all spikes are classified. The package is available for Python, utilizing CPUs.

2) Data Transmission and Compression Hardware: Processing data external to the site of neural recording requires the collected data to be transmitted wirelessly. Since most of such modules are implanted, they are especially resource-constrained. Therefore, researchers must strike a balance between latency, data resolution, and power consumption during the design process.

A comprehensive system overview for neural data telemetry has been presented in [97]. A neural signal telemetry platform typically consists of an analog preamplifier, ADCs, a digital processor, and a wireless transmitter. The analog preamplifier provides a voltage gain to the collected extracellular potentials and applies a bandpass filter to avoid aliasing during digitization. The amplifier may introduce thermal and flicker noises. ADCs are ubiquitous in spike sorting systems, ranging from 8-bits to 12-bits. Efforts have been made to improve the energy efficiency and area of this component while maintaining sufficient sampling rate and bit rate [97].

While some developments focus on improving the efficiency of data telemetry modules, some other publications such as [98] focus on developing an online interpolation and alignment module to cope with lower bit rate transmitted data, ultimately reducing power consumption of the overall system.

B. On-site Processing

1) Field Programmable Gate Array (FPGA) and Application-specific Integrated Circuits (ASIC): Field Programmable Gate Arrays (FPGAs) belongs to a class of integrated circuits that consists of interconnected configurable logic blocks, which can be programmed after manufacturing for rapid prototyping. A finalized FPGA design can be mass manufactured as Application-specific-integrated-circuit (ASIC) with improved computational efficiency. Due to their similar properties and functionalities, this section will include both FPGA and ASIC spike sorting hardware.

FPGAs are ubiquitous in online spike sorting applications. They enable hardware accelerations for particular workloads. Moreover, they can be designed with specific latency and power constraints in mind, which are crucial for on-site spike sorters. Spike sorting techniques based on FPGAs can be broadly categorized into several classes, the most prevalent amongst which are the template matching algorithms, due to their simplicity and compatibility with FPGAs. The most streamlined approach to template matching is Osort covered in the previous section, which was originally developed on conventional hardware for online spike sorting. More recently, modified FPGA implementations have been proposed as in [36, 99–102] with reduced memory access and numerical operations. The earliest FPGA Osort implementation is proposed in [36] with minimal algorithm modifications compared to its conventional CPU-based counterpart, hence its maximum latency of 11.1ms does not guarantee online functionality. [101] improves upon [36] by optimizing for larger data rate from multi-channel operation while reducing memory requirements. This can realize the capability for sorting 128-channels in real time.

[99] uses a modified memory configuration scheme, in which the cluster merging and averaging are performed con-

currently to spike shape comparisons, as opposed to sequential cluster assignment, averaging, and merging from traditional Osort [93]. Additionally, a cluster is only averaged when a new spike is assigned to it, and only the newly updated cluster is compared with others for potential merging. One of the newest iterations of FPGA-based Osort proposed takes into account the spatial correlation between channels, which utilizes a spatial window, operating on cluster memory [102] for improved utilization of multi-channel information. There also exists other variations of FPGA-based template matching that do not fall under the Osort generalization, such as techniques proposed in [34, 103], that diverge from Osort in terms of processing pipeline or template similarity measures.

Various other approaches have been implemented for spike sorting on FPGAs. These include Hebbian PCA where the eigenvectors can be trained iteratively and then used for prediction similar to machine learning algorithms [104–106]. Meanwhile, geometric features are also commonly implemented on FPGA and ASIC including amplitude extremes and discrete derivatives [51, 107, 108].

Recently, researchers are also starting to shift their attention to optimizing clustering techniques on novel hardware, as most current clustering techniques require temporarily storing large amounts of data in memory, which is computationally expensive and difficult to implement for online systems. One notable example is the Enhanced Growing Neural Gas (EGNG) algorithm proposed in [109], which employs a small number of EGNG nodes and edges to learn the neural spike distributions, minimizing memory use. Additionally, it does not assume Gaussian clusters.

2) In-memory Computing: In highly data-centric applications, traditional digital systems suffer from the Von-Neumann bottleneck which refers to the high computing time and energy cost associated with the frequent transfer of digital data between memory and processing units. Hence, in-memory computing hardware has been proposed where computations are performed in the memory itself by exploiting physical attributes of specific memory devices [110–112]. This improved efficiency reveals its potential to be used in neuro-implants, such as neurological disorder detection [113], as well as spike sorting. A demonstrative pipeline of how a spike sorter neural network can be implemented on a memristor-based IMC platform versus the same network using a digital processor is shown on the right-hand side of Fig. 9, where the difference in VMM calculations is accentuated. Here, it is shown how VMM can be implemented more efficiently by summing the current proportional to the synaptic weight, using the basic Ohm's and Kirchhoff's laws.

For IMC systems, distinctions can be made between the 2 types of utilized memory devices, i.e. conventional charged-based and emerging resistance-based. Charged-based memory devices such as DRAM, flash, or SRAM store information as the presence or absence of charge and can be found in common consumer electronics. Resistance-based memory (memristive) such as resistive random-access memory (RRAM) and phase-change memory (PCM), on the other hand, store information as resistance values of blocks arranged in 2D arrays.

In this review, we focus on IMCs performed in memristive crossbar configurations, due to their efficiency and suitable attributes. Memristors are circuit elements initially proposed by Chua in [114], which are typically manufactured in metal-oxide-metal configurations [111, 112], and the resistance value is determined by atomic arrangements caused by oxygen vacancies in the middle layer. In a low resistance state, the conductive filaments within the memristor have high oxygen vacancies. By applying appropriate voltage pulses above a certain voltage threshold, the vacancies are migrated back toward the top electrode raising the resistance. The resistance value can be changed by applying voltage pulses through SET and RESET cycles.

Some works utilize memristive devices to implement variations of the STDP algorithms mentioned in Section IV-C. STDP can be natively implemented on a memristor-crossbar device, where each memristor acts as a synapse whose conductance represents the synaptic weights [115, 116]. Memristors can mimic a synapse as its conductance can be varied via voltage waveforms that encode input pre- and post-synaptic spikes. CMOS circuits convert the time difference between pre and post-synaptic neurons into voltage pulses that can be applied to the memristor modulating its conductance [112, 116]. It has been shown that the parameters including switching probabilities and ΔT can be trained initially using a genetic algorithm and be implemented on any neural recording without re-calibration. The supervised SNN variants for spike sorting mentioned previously have also been implemented efficiently on the memristor crossbar architecture as demonstrated in [83], where the spike train input is converted to current and SNN's weights are mapped as memristors' conductance across the crossbar. Ohm's law governs the analog computation and the result is read out as voltages.

Another application of memristors in spike sorting is to implement a class of algorithms generally referred to as signal processing techniques that measure the resistance changing behavior of memristors, as proposed in [117–120]. Essentially, the continuous time analog neural recording is directly passed into a memristor as voltages. Memristor's inherent voltage threshold acts as a spike detector, similar to the "amplitude threshold" detection technique. If the threshold is surpassed, the voltage from the detected spike will cause a change in conductance value for the memristor. According to [117], the resistance change curve is affected by both the voltage amplitude as well as the amplitude variation within the spike. Consequently, the change in resistance encodes 2 features regarding a spike, which can be used to perform classification. Different bin lengths or window lengths can be set up to divide a spike into segments to input into different memristors and multiple resistance change values can be measured to use as multiple features. Fig. 10 illustrates a general configuration for such systems, where the columns highlighted in red are the ones turned on during the specific time bin, and then the resistance change is measured.

An overview comparison between recent spike sorting hardware is shown in Fig. 11 in terms of power consumption and area, grouped according to the technology class. It is apparent that there is a decreasing trend in both power consumption and area over the years, and it is evident that processors

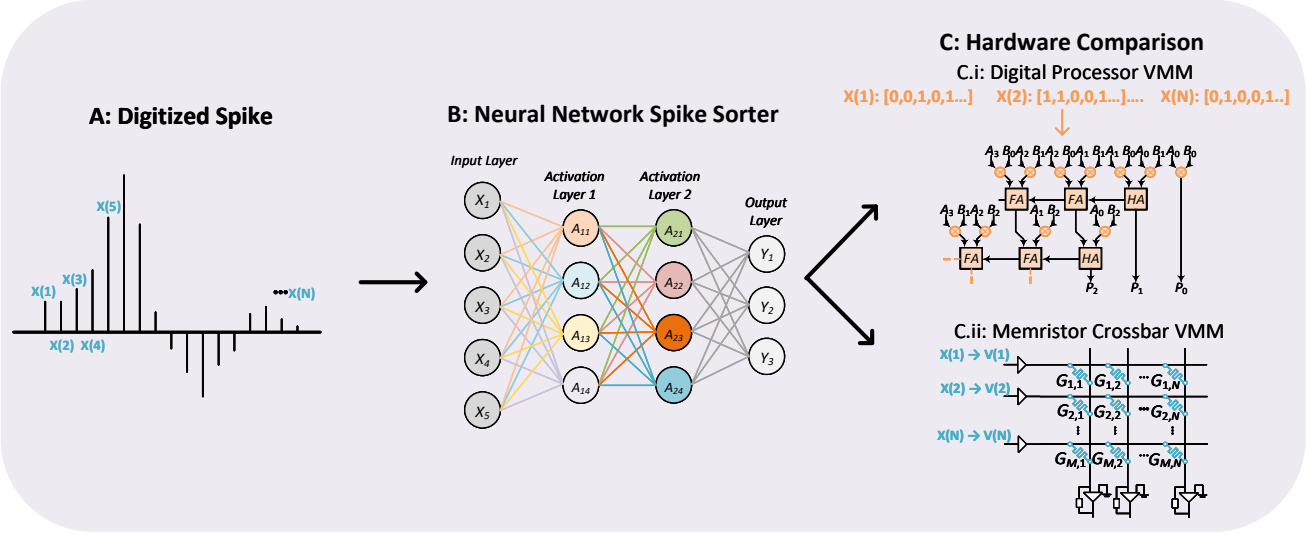

Fig. 9: A typical implementation of ANN-based spike sorting systems, where features of the signal are extracted. Typically, the amplitude of the discretized signal is indicated atx points in time (A). Then, these are passed into a neural work outputting the class of the spike (B). Lastly, (C) illustrates the two types of hardware suitable for the ANN in (B). (C.i) shows digital devices performing vector-matrix-multiplication (VMM) operations digitally with logic gates, while (C.ii) shows in-memory-computing devices accelerating ANNs by performing the same VMM operations in a mixed-signal IMC fashion.

TABLE IV: Comparison of Several Hardware Spike Sorting Systems

Year	Reference	Hardware	Algorithm	Power/Ch (uW)	Latency/Ch (ms)	Area/Ch (mm2)
2009	[121]	FPGA	PCA	256	41.8	28.32
2009	[122]	ASIC	Geometric Features	14.6	NA	1.36
2010	[100]	FPGA	Osort	4.68	NA	2.45
2013	[106]	ASIC	Hebbian PCA	440.3	NA	2.45
2015	[104]	ASIC	Hebbian PCA	85.8	NA	0.083
2015	[80]	ASIC	SNN	9.3	NA	0.25
2017	[34]	FPGA	Template Matching	0.064	0.00055	0.3
2017	[123]	IMC	Template Matching	3.15	NA	0.0005
2020	[124]	ASIC	BNN	2.02	11	0.33

belonging to the same class reside in close proximity on the graph displaying similar hardware characteristics. Furthermore, a general hardware comparison is shown in Fig. 12 to demonstrate the strengths and weaknesses of each option.

VI. Challenges and Future Outlook

A. Recording non-stationarity

Generally, extracellular recordings are considered nonstationary as neural signals are affected by multiple factors that evolve with time [125, 126]. Spike shapes can change in the short term due to bursting which changes the neuron membrane conductance [126] or they can change in the long term due to electrode drifts, which refers to the change of electrodes' positions with respect to the electrodes as a result of pressures from surrounding tissues. Such drifts introduce distortions that pose challenges to spike sorting systems, especially the ones that require a degree of manual intervention.

Multiple approaches have been explored to counter this challenge. Firstly, some studies [35, 44, 127] have investigated developing unsupervised adaptive algorithms with parameters that evolve with time, adapting to the time-varying spike waveforms. Strategies for achieving this adaptability can be summarized as performing feature engineering that makes minimal a priori assumptions regarding spike shapes. Conversely, researchers have looked at improving existing non-adaptive algorithms by introducing efficient automatic retrain-

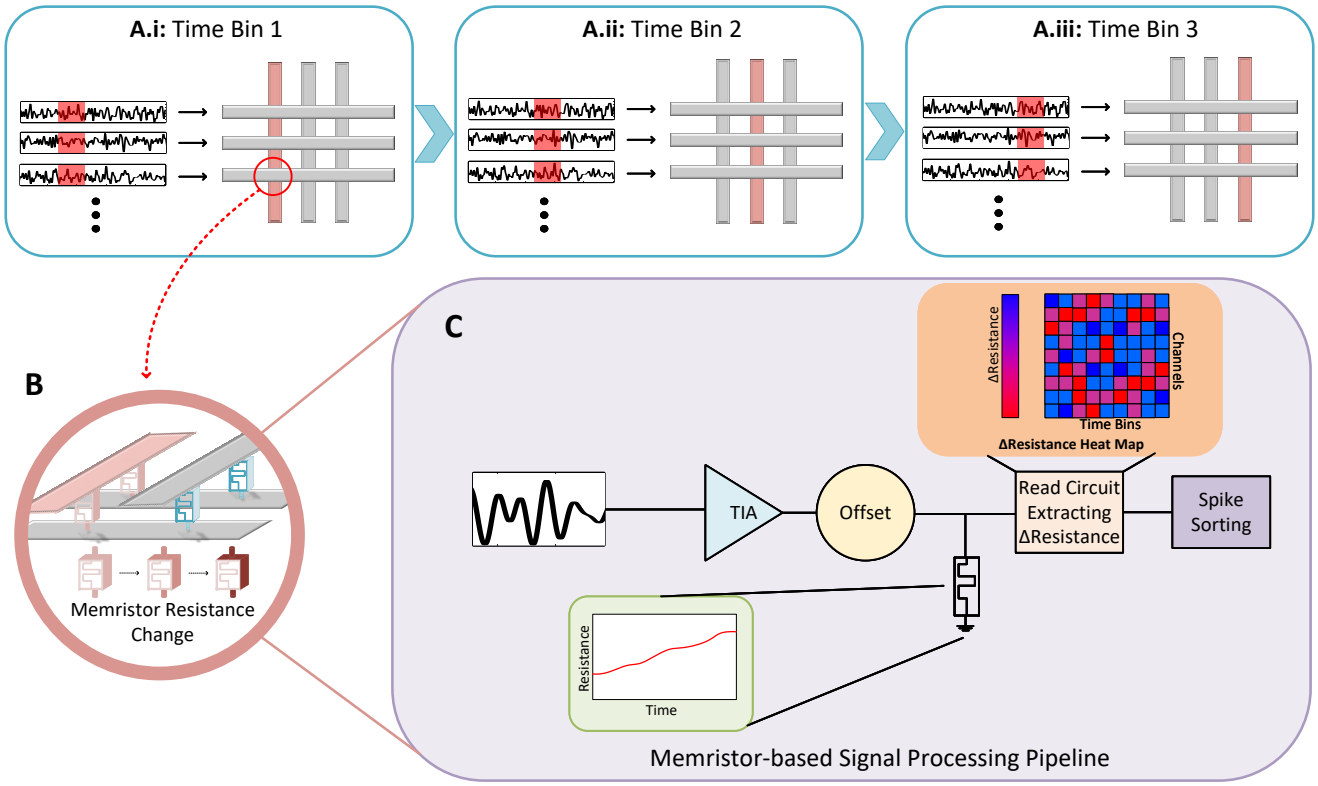

Fig. 10: A demonstrative implementation of a memristor-based spike sorting platform, where (A.i-A.iii) illustrate three time bins, where each column is activated sequentially. The construction of such a memristor-based processor is visualised in (B). Each row corresponds to each recording, and a memristor device is sandwiched in-between each row and column, with changing resistance driven by the electrical waveforms passing through it. An envisioned hardware pipeline is shown in (C), where the signal is passed through a transimpedance amplifier (TIA) and offset. The processed recording is then passed through the memristor crossbar, and the resistance change is read-out using a read circuit.

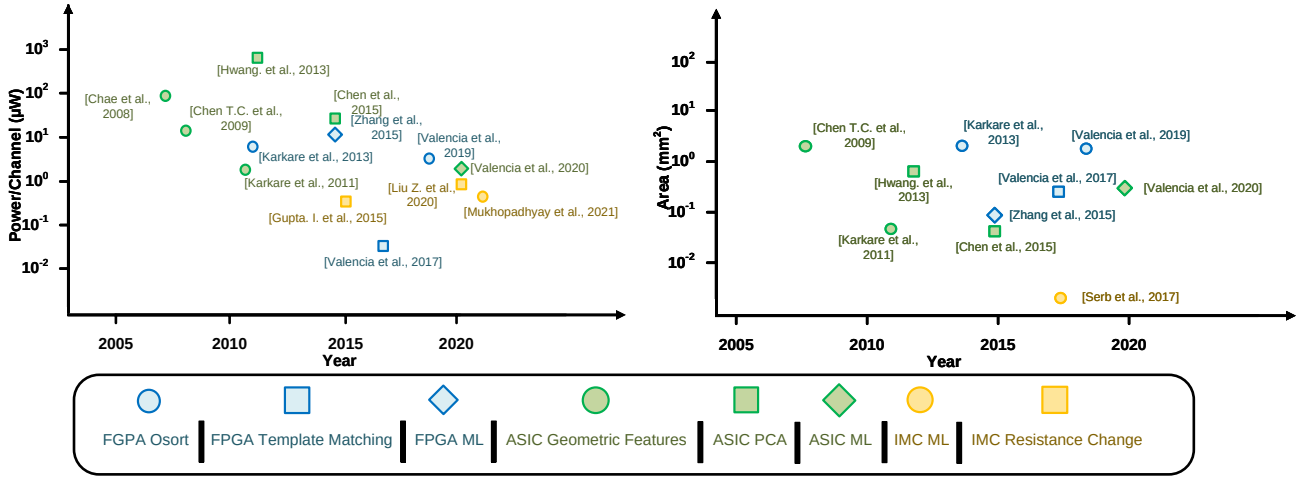

Fig. 11: Power and area comparisons between hardware-based spike sorting publications from the past 2 decades, grouped according to the hardware technology used.

ing schemes periodically [38, 83]. Typically this retraining involves evaluating a custom cost function that measures performance degradation with the progression of time, if the degradation surpasses a certain threshold, then retraining is triggered.

Another approach that has been commonly investigated is to model the changes of spike waveforms [127, 128] or to model the shift in electrode position with respect to time [28, 129] using statistical methods. The former typically involves generative models that capture the change in waveforms after the

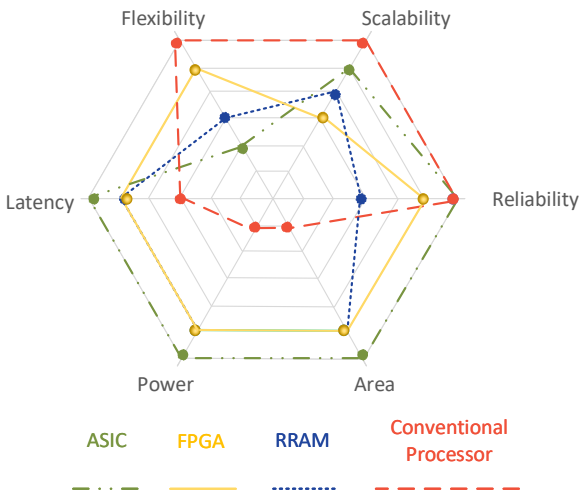

Fig. 12: Spider diagram comparing the properties of the four main spike sorting hardware options.

extracellular signal has been recorded [128]. The latter refers to a more involved process that first constructs a raster plot illustrating the changes of electrode position over time, which can then be used to apply motion correction to the recording to counter effects of drift right at the signal collection stage [28].

Despite these advances, there is still a lack of a standardized technique for overcoming the issue of non-stationarity. The first approach with adaptive features often has limited ability to counter the effects of prolonged electrode drift, especially during the clustering stage. The approach to modeling the evolution of spike waveforms over time, on the other hand, is relatively computationally intensive, which renders it unsuitable for online uses. The problem of nonstationarity is arguably a more pressing issue for online applications since there can be fewer opportunities for manual intervention and retraining. To fully understand the evolution of neural recordings, more research should be conducted to examine the interaction between brain tissues and implanted electrodes in order to understand the core factors behind signal shifts. More details regarding biocompatibility is discussed in Section VI-H.

B. Overlapping spikes

Overlapping spikes has been a major obstacle for spike sorting ever since the dawn of the field [130, 131], yet there still exists no consensus on the best way of resolving it. This phenomenon refers to temporally overlapping spikes, detected on the same electrode. The superposition waveform deviates from other single-unit waveforms and confounds the sorting pipeline. The performance degradation is especially acute for conventional clustering approaches [90]. The most commonly accepted technique for resolving overlapping spikes is template matching as mentioned in Section IV-A, where a greedy approach is taken to iteratively subtract spike templates from the detected waveforms in order to break down any potential overlaps [21, 22, 24, 25]. Alternatively, Baysian statistical techniques have also been employed to resolve overlapping spikes [132, 133], which consider the degree of certainty of the classification results. More recent algorithmic developments using Artificial Neural Network (ANN) are also specifically designed to counter this issue. Examples are [75, 78], which are detailed in Section IV-B. However, all of the algorithmic solutions currently capable of resolving overlaps are relatively computationally complex as they typically involve an iterative process instead of a simple pipeline, hence their online applicability lacks thorough investigation from the processing time and energy consumption perspective. Intuitively, such an iterative process should benefit from specialized hardware with a high degree of parallelization and reduced memory access. Hence, more future research should be carried out on efficient implementations of spike overlapping detection and resolution, making them suitable for real-time on-site processing.

In addition to algorithmic solutions, extracellular signal collection hardware has also made great strides towards Microelectrode Array (MEA) with higher density electrodes, which provides previously unachievable spatial resolution for the recording, which reveals opportunities to address the overlapping spikes issue which will be covered in details in the next section.

C. High-density Electrode Arrays

The recent advancements in material engineering fuel the emergence of increasingly higher density Microelectrode Arrays (MEAs) [28, 134]. A more comprehensive review of Microelectrode Array (MEA) technology can be found in [20]. The increase in spatial resolution of the acquired signals is driving the adoption of algorithms that can fully take advantage of larger data sizes. In this paper, we will have a refined focus on the new challenges as well as opportunities associated with improved MEAs.

One of the challenges that arise is the massive increase in data rate generated from these high density probes. Neuropixel probes can generate 1GB/min for 382 channels at 30kHz [55], which requires algorithms that are resource efficient and able to process input from all channels simultaneously. As seen in the previous sections, some spike sorting pipelines are designed from the ground up to maximize the use of parallel computing, such as in [22, 89, 135, 136].

The spatial resolution improvement brought by the higher density MEAs unlocks a vast number of opportunities if properly utilized. For instance, a Neuropixel probe can simultaneously sample more areas of the brain, including different layers of superior colliculus and periaqueductal gray in mouse brains [8]. This ability to potentially monitor both the input and output simultaneously allows for neuroscience research of information processing in the brain like never before. Furthermore, given sufficient spatial resolution, certain electrodes will only detect spikes from isolated single neurons, stemming the overlapping spikes issue at its core [37, 135, 137].

Lastly, efforts are also being continually dedicated towards making neural recordings more standardized and accessible.

One example of this is the recently introduced NWM file type and its associated tools [61]. Such developments are essential in ensuring efficient data sharing integration between labs and equipment, and reproducibility.

D. Noise Source

While covering the different spike sorting techniques, we briefly mentioned the impact of noise in applications involving extracellular recordings or recorded spike trains. In this subsection, we provide a more detailed discussion of the sources and characteristics of this noise. Since the noise is aggregated from multiple sources forming non-white noise, accurately modeling it remains a challenge. A more comprehensive understanding of this aspect is crucial for spike sorting as it is often assumed that noise overlapping with true spike waveforms results in the variability in spike shapes [64, 90, 125].

The primary type of noise present in neural recording systems stems from electrical circuits, which are an integral component of spike sorting systems. Typically, electronic noise consists of thermal noise and flicker noise, both of which can be accurately estimated via circuit design tools. Since the voltage spikes are low in amplitude and duration, neural acquisition amplifiers are required, which have been shown to be the most significant component that introduces electronic noise. Significant efforts have been dedicated to designing low-noise neural recording platforms, as detailed in [138–140].

Another significant noise rooted in implantable hardware is the electrode-electrolyte interface noise [141]. This is a type of noise that arises from the electrochemical transport phenomenon of charges at the electrode-electrolyte interface. Over time, the electrode becomes encapsulated by glial tissues, a problem that will be covered in the VI-H section. This leads to increased capacitance on the electrodes and higher thermal noise, which eliminates signal resolution at higher frequencies (300-5000Hz) [142]. The specific mechanisms of this noise process are dependent on multiple factors including the size and material of the electrodes, which are beyond the scope of this survey; a comprehensive analysis has been detailed in [143]. Recent progress in high impedance CMOS based MEAs with improved density and spatial resolution places emphasis on this type of noise, which needs to be accounted for to enable accurate simulations of the cell-electrode interaction. A mathematical model for such interfacing noise in implantable MEAs for neural recording applications has been performed in [141, 144].

In the case of neural recordings, it has been shown that the most dominant source of noise is the aggregated spiking activities captured from distant background neurons. These multi-unit activities have amplitudes inversely proportional to their distances from the electrode, rendering them impossible to isolate. This type of noise is less frequently investigated and modeled. Generally, an amplitude threshold is used to distinguish between nearby spikes and background multi-unit noise [40]. Moreover, improved spatial resolution achieved by high-density MEAs also alleviates this issue.

E. Training Datasets and Evaluation Schemes

Unlike the computer vision field where standardized labeled datasets exist such as MNIST, CIFAR-10, and ImageNet, the spike sorting field lacks standardized extracellular recordings with ground truth spike labels and spiking times. The first implication of this challenge is that there lacks a standardized benchmark for accuracy comparison across multiple spike sorting systems [29]. Due to the highly variable spike shapes, noise structures, and recording settings, different datasets may lead to drastically different accuracy when using the same algorithm. The second implication of this issue is that it hinders the usability of supervised algorithms such as template matching as they depend on high-quality labeled data.

Currently, most spike sorting systems are evaluated with synthetic datasets with ground truth labels. Such datasets are typically created from real spike shapes, either obtained from manual classification or the patch clamp technique recording from a single cell [145]. Earlier synthetic datasets take a simpler approach where Gaussian noise is added to the spike waveforms [133], while later approaches aim to more realistically model the noise structure [145]. Similarly, some works resort to ”hybrid ground truth recordings”, which are based on real neural recordings but with manually added de-noised ground truth waveforms [22]. However, realistic multi-channel recordings are more challenging for synthetic techniques to construct, due to the complex interactions between spiking neurons and multiple surrounding electrodes. Hence, some works use real in-vitro or in-vivo recordings with manually curated labels [25].

As different biological recordings can differ in signal properties, researchers may be interested in examining the viability of a certain spike sorting pipeline on real biological recordings such as the notable public collection maintained at Collaborative Research in Computational Neuroscience (CRCNS)[146]. Alternative accuracy evaluation schemes have also been proposed to circumvent the need for labels. Statistical methods have been proposed such as in [64] that explicitly takes into account the noise that leads to the variance of clusters, leading to the conclusion that a more uniform variance of spike waveforms indicates a higher quality of classification. Similarly, intracluster variance (ICV) has been proposed to measure the compactness of each cluster [71], serving as a viable alternative to conventional accuracy measures [29]. Rand index [147] and Jaccard index [148] are also popular metrics frequently employed in the unsupervised machine learning field, which make them suitable for quantitatively evaluating spike sorting algorithms without labeled data [149]. A more comprehensive proposal has been presented in [150], which introduces a suite of metrics that measure the statistical ”confidence” of an algorithm’s output based on the concept of bootstrapping. Lastly, efforts are also being continually dedicated to making neural recordings more standardized and accessible, such as introducing the NWM file type and its associated tools [151]. Such development is essential in ensuring efficient data sharing and integration between labs.

Further efforts can be dedicated to streamlining the performance comparison process. Currently, several software pack-

ages and frameworks are being developed [71, 152, 153] to allow fast performance comparison across the same datasets in a controlled environment. These toolboxes are a valid starting point for evaluating future spike sorters, and they could continue to improve by incorporating more datasets and testing criteria in order to better approximate their multi-faceted real-world performances.

Despite the recent explosion in data recording rates driving the adoption of fully-autonomous sorting algorithms, future researchers should not solely rely on quantitative measures and neglect the quality of the spike trains [154]. The most common procedure is to manually visualize the spike waveforms from each cluster, as unsupervised techniques often create clusters that can be mostly noise instead of spike waveforms. Inspection of waveforms can also reveal a temporal shift in spike waveforms that indicates electrode drift or electrode encapsulation. Moreover, isolated spike trains should show reasonable inter-spike intervals that correspond to the neuron's biological refractory period; frequent violation of this principle may suggest that spikes from multiple neurons are incorrectly attributed to a single neuron.

F. Hardware-algorithm Co-optimization

As the focus of this paper, we believe that in order to bring spike sorting closer to a streamlined technique with more real-world implications, a concerted effort must be made by researchers in both the algorithm and hardware frontier. Throughout this survey, the reader is made aware that the performance of a spike sorting system is heavily dependent on the algorithm-hardware combination, instead of being dictated by either one of the two aspects.

More specifically, it is likely that future research attention would be placed more on on-site computing rather than cloud computing or remote computing for several reasons. Firstly, the improvements in chip design and shrinking transistor sizes allow implantable on-site processors to be more effective than ever, similar focus shift to edge-computing is already apparent in other biomedical fields as detailed in [155]. Secondly, it omits the need for heavy data transmission, which leads to an increased hardware overhead and introduces additional latency, hindering real-time applicability. Lastly, the recording subject may prefer the neural recordings to be processed without transmitting externally due to the extremely private nature of such collected data.

In recent years there has already been a surge in hardware-based spike sorting publications, but most of them revolve around modifying existing simpler algorithms and designing the hardware for it. The field calls for the development of more novel algorithms with the consideration of matching hardware to achieve maximum efficiency, similar to the approach taken in the RRAM-based SNN spike sorting systems covered in Section IV-C, which are designed from the ground up with co-optimization in mind.

A co-optimization strategy roadmap is presented in Fig.13, where each column represents an aspect of the optimization process. The leftmost and rightmost column represents the paradigms in algorithm and hardware developments, respectively. The connections between columns represent the viable combinations between the compatible options as detailed in this review work, allowing future researchers to identify a suitable spike sorting system for the intended applications.

G. The Importance of Spike Sorting

As spike sorting technology advances at a rapid pace, some researchers are starting to rethink its role in many applications, and investigating whether there are alternative techniques that accomplish similar tasks at a reduced computational cost. We therefore dedicate this subsection to surveying the evolving role of spike sorting in different applications, as well as novel alternatives that may be of interest to readers.

Performing spike sorting in the motor cortex has been a gold standard approach for developing motor Brain Machine Interface (BMI). However, recent works have proposed systems using local field potentials (LFPs) [142], threshold crossing events [156] and Kalman filters [157–159]. While each of these alternatives have been proven suitable under certain conditions, it is commonly agreed that spike sorting still offers the highest degree of freedom and accuracy of decoding user intentions in a wide range of settings [26, 160]. However, whether the improved accuracy is worthy of the increased hardware complexity depends on the specific experimental setting, and should be thoroughly considered by potential researchers.

While some applications are able to primarily decode information from multi-neuron activities and can circumvent the need for spike sorting [161], the majority of neuroscience research that investigates the neuronal information encoding [162–164], single neuron responses [7] and neuropathophysiology [165, 166] still heavily rely on the identification of single unit spikes.

Spike sorting has high clinical significance for neurological disorders such as Parkinson's Disease. It has been shown that sorted spike trains collected from Subthalamic nucleus with deep brain stimulation (DBS) devices reveal important information regarding the oscillatory, bursting and synchronization of neurons in patients [166–168]. Such information could aid with the better positioning of DBS implantation and integration [168].

Furthermore, it has been shown that different spike sorting algorithms with varying degrees of accuracy can lead to different interpretations of the recorded signal [7, 63, 166]. Hence, there still exists a demand for more accurate and efficient spike-sorting algorithms to spearhead neuronal research or be used in conjunction with other techniques such as calcium imaging.

H. Hardware Biocompatibility

Biocompatibility is a major challenge for spike sorting systems, as they require invasive implants. The first point of consideration for biocompatibility is the power budget. The maximum amount of power that can be supplied to an implant, for example for a Brain Machine Interface (BMI) application, is mainly limited by the maximum thermal dissipation that is safe for brain tissues. This is below 0.5°C [33, 169].

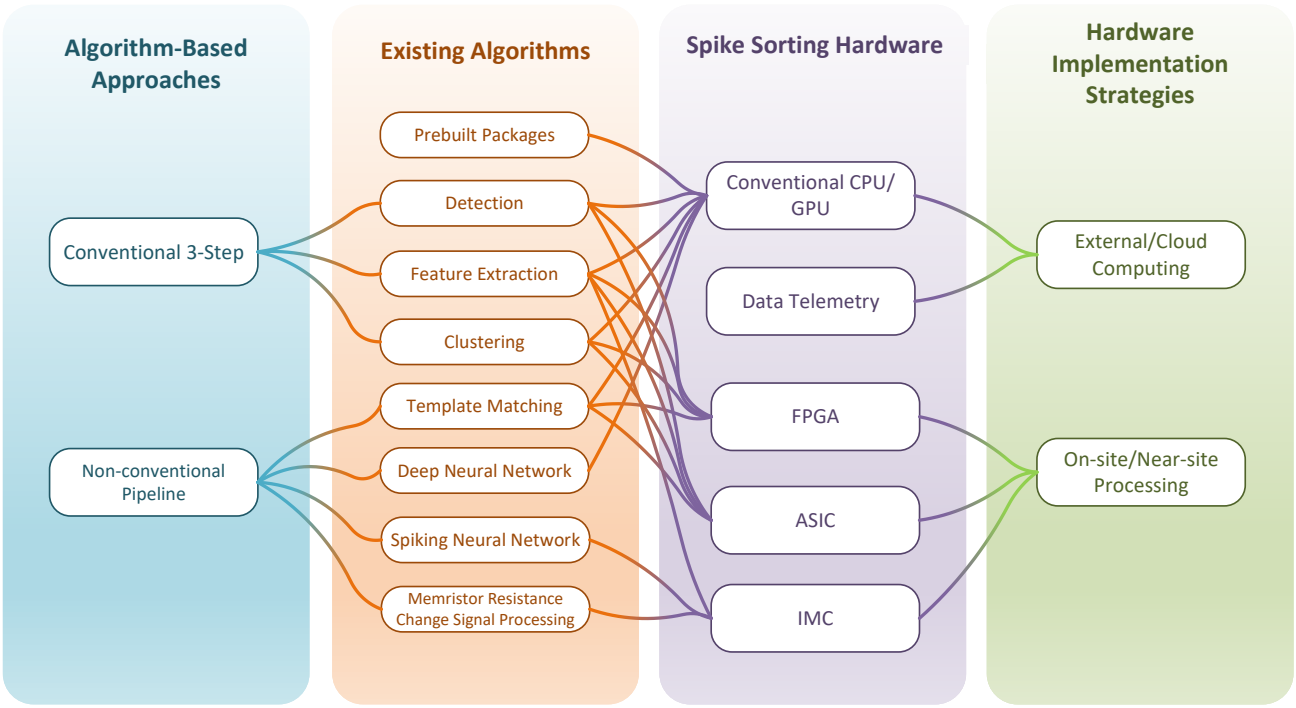

Fig. 13: Roadmap of algorithm-hardware co-optimization strategies. Each column represents an optimizable design option, each line connecting between elements within columns represents a suitable combination.

Wireless power transmission module, data transmission module, processing module, and Microelectrode Array (MEA) are all components of heat generation, and improving their efficiencies is a primary area for improvement. Furthermore, the heat dissipation dynamics of spike sorting systems should be investigated and modeled, enabling future researchers to estimate the thermal behavior of proposed implantable systems. Currently, such studies exist but remain scarce [170, 171].

Another aspect of examining biocompatibility is tissue damage caused by the foreign object and the implantation process. There exists a substantial mechanical property mismatch between the neural tissue (Young's modulus in the range of kPa) and electrodes (Young's modulus in the range of GPa)[103, 172], which causes blood-brain barrier leakage, neuronal degeneration, and glial scarring [103, 173]. On the other hand, such incompatibility also causes material and structural degradation for the electrodes, which hinders recording fidelity [174]. Current technologies render long-term Brain Machine Interface (BMI) implants impractical. Future developments for MEAs technology should not solely be focused on increasing electrode density, but also work towards improving the mechanical quality for reduced tissue damage and electrode endurance [175].

I. Ethics

Neural recording systems, such as spike sorting systems intended for applications such as Brain Machine Interface (BMI), neurodegenerative disease monitoring, or brain science research, can potentially allow for the reading of human brains. Such advancements in technology will inevitably be ethically provocative to society. Spike sorting can also inspire downstream applications that can provide feedback mechanisms, such that it can "write" to the brain. Deep brain stimulation systems (DBS), a type of Brain Machine Interface (BMI), have been shown to elicit impulse-control issues [176, 177], which raises the question of autonomy for those using brain modulating systems.

Moreover, modern spike sorting systems and Brain Machine Interface (BMI) systems often involve the use of Artificial Intelligence to interpret brain signals, which is a "black box" prediction system. This sparks additional ethical concerns as the user is not "directly" in control of such systems, such as prostheses. This problem is exacerbated if such BMI components become an essential part of life. Additionally, AI systems learn from massive datasets, all recorded from the brain, which raises privacy concerns, especially considering the "thoughts" in one's brain are the most private aspect of life.

It is without a doubt that spike sorting systems and a myriad of BMI applications can unlock countless benefits to society, however, it is a brave new field and regulators did not yet have time to explore all the potential implications of such technology. Hence, it is important for the scientific community to explore and research ethics equally as fervently as pushing the technical boundaries forward.

VII. Conclusion

Spike sorting is a fundamental signal-processing technique in the field of neuroscience and neural engineering. Despite its emergence decades ago, we show that it is still extremely relevant today. Fueled by the advancements in electrode arrays, algorithms, and computation hardware, spike sorting is being used to unlock an ever-growing variety of applications, from BCIs, to neuroprosthetics, to neuron populations research. Moreover, more opportunities await as spike sorting is being used in conjunction with novel neuroscience techniques such as optogenetic manipulation of neurons.

However, there is no clear consensus on the best implementations for spike sorting, nor widespread adaptations in commercial applications, unlike other fundamental signal processing techniques such as physiological system identification. This could be a result of the combination of algorithm and hardware challenges. This study bridges the gap that exists in previous works by covering both novel algorithms as well as novel hardware, with an emphasis on algorithm-hardware co-optimizations for real-world applicability. We hope to assist future researchers in selecting the appropriate algorithm and hardware for the specific applications while bringing forward potential future research directions to address the remaining shortcomings of the various techniques and implementations.

References

[1] A. L. Hodgkin and A. F. Huxley, "Currents carried by sodium and potassium ions through the membrane of the giant axon of loligo," *The Journal of physiology*, vol. 116, no. 4, pp. 449–472, 1952.

[2] J. C. Chang, G. J. Brewer, and B. C. Wheeler, "Microelectrode array recordings of patterned hippocampal neurons for four weeks," *Biomedical Microdevices*, vol. 2, no. 4, pp. 245–253, 2000.

[3] B. Lefebvre, P. Yger, and O. Marre, "Recent progress in multi-electrode spike sorting methods," *Journal of Physiology-Paris*, vol. 110, no. 4, pp. 327–335, 2016.

[4] H. G. Rey, C. Pedreira, and R. Q. Quiroga, "Past, present and future of spike sorting techniques," *Brain research bulletin*, vol. 119, pp. 106–117, 2015.

[5] M. Abeles and M. H. Goldstein, "Multispike train analysis," *Proceedings of the IEEE*, vol. 65, no. 5, pp. 762–773, 1977.

[6] G. L. Gerstein and W. A. Clark, "Simultaneous studies of firing patterns in several neurons," *Science*, vol. 143, no. 3612, pp. 1325–1327, 1964. [Online]. Available: https://science.sciencemag.org/content/143/3612/1325

[7] P. N. Steinmetz, "Estimates of distributed coding of visual objects by single neurons in the human brain depend on which spike sorting technique is used," *Journal of Neural Engineering*, vol. 17, no. 2, p. 026030, 2020.

[8] N. A. Steinmetz, C. Koch, K. D. Harris, and M. Carandini, "Challenges and opportunities for large-scale electrophysiology with neuropixels probes," *Current opinion in neurobiology*, vol. 50, pp. 92–100, 2018.

[9] L. R. Hochberg, M. D. Serruya, G. M. Friehs, J. A. Mukand, M. Saleh, A. H. Caplan, A. Branner, D. Chen, R. D. Penn, and J. P. Donoghue, "Neuronal ensemble control of prosthetic devices by a human with tetraplegia," *Nature*, vol. 442, no. 7099, pp. 164–171, 2006.

[10] M. A. Nicolelis, "Actions from thoughts," *Nature*, vol. 409, no. 6818, pp. 403–407, 2001.

[11] T. W. Berger, A. Ahuja, S. H. Courellis, S. A. Deadwyler, G. Erinjippurath, G. A. Gerhardt, G. Gholmieh, J. J. Granacki, R. Hampson, M. C. Hsaio *et al.*, "Restoring lost cognitive function," *IEEE Engineering in Medicine and Biology Magazine*, vol. 24, no. 5, pp. 30–44, 2005.

[12] K. Hu, M. Jamali, Z. B. Moses, C. A. Ortega, G. N. Friedman, W. Xu, and Z. M. Williams, "Decoding unconstrained arm movements in primates using high-density electrocorticography signals for brain-machine interface use," *Scientific reports*, vol. 8, no. 1, pp. 1–13, 2018.

[13] R. Q. Quiroga, L. Reddy, C. Koch, and I. Fried, "Decoding visual inputs from multiple neurons in the human temporal lobe," *Journal of neurophysiology*, vol. 98, no. 4, pp. 1997–2007, 2007.

[14] H. G. Rey, M. J. Ison, C. Pedreira, A. Valentin, G. Alarcon, R. Selway, M. P. Richardson, and R. Quian Quiroga, "Single-cell recordings in the human medial temporal lobe," *Journal of anatomy*, vol. 227, no. 4, pp. 394–408, 2015.

[15] A. V. Kravitz, B. S. Freeze, P. R. Parker, K. Kay, M. T. Thwin, K. Deisseroth, and A. C. Kreitzer, "Regulation of parkinsonian motor behaviours by optogenetic control of basal ganglia circuitry," *Nature*, vol. 466, no. 7306, pp. 622–626, 2010.

[16] A. V. Kravitz, S. F. Owen, and A. C. Kreitzer, "Optogenetic identification of striatal projection neuron subtypes during in vivo recordings," *Brain research*, vol. 1511, pp. 21–32, 2013.

[17] Y. Zhou, E. Liu, H. Muller, and B. Cui, "Optical electrophysiology: Toward the goal of label-free voltage imaging," *Journal of the American Chemical Society*, vol. 143, no. 28, pp. 10 482–10 499, 2021.

[18] A. S. Tolias, A. S. Ecker, A. G. Siapas, A. Hoenselaar, G. A. Keliris, and N. K. Logothetis, "Recording chronically from the same neurons in awake, behaving primates," *Journal of neurophysiology*, vol. 98, no. 6, pp. 3780–3790, 2007.

[19] G. Buzsáki, "Large-scale recording of neuronal ensembles," *Nature neuroscience*, vol. 7, no. 5, pp. 446–451, 2004.

[20] M. E. J. Obien, K. Deligkaris, T. Bullmann, D. J. Bakkum, and U. Frey, "Revealing neuronal function through microelectrode array recordings," *Frontiers in neuroscience*, vol. 8, p. 423, 2015.

[21] Y. Mokri, R. F. Salazar, B. Goodell, J. Baker, C. M. Gray, and S.-C. Yen, "Sorting overlapping spike waveforms from electrode and tetrode recordings," *Frontiers in neuroinformatics*, vol. 11, p. 53, 2017.

[22] M. Pachitariu, N. A. Steinmetz, S. N. Kadir, M. Carandini, and K. D. Harris, "Fast and accurate spike sorting of high-channel count probes with kilosort," *Advances*

in neural information processing systems, vol. 29, 2016.

[23] P. Yger, G. L. Spampinato, E. Esposito, B. Lefebvre, S. Deny, C. Gardella, M. Stimberg, F. Jetter, G. Zeck, S. Picaud *et al.*, "Fast and accurate spike sorting in vitro and in vivo for up to thousands of electrodes," *BioRxiv*, p. 067843, 2016.

[24] J. W. Pillow, J. Shlens, E. Chichilnisky, and E. P. Simoncelli, "A model-based spike sorting algorithm for removing correlation artifacts in multi-neuron recordings," *PloS one*, vol. 8, no. 5, p. e62123, 2013.

[25] P. Yger, G. L. Spampinato, E. Esposito, B. Lefebvre, S. Deny, C. Gardella, M. Stimberg, F. Jetter, G. Zeck, S. Picaud *et al.*, "A spike sorting toolbox for up to thousands of electrodes validated with ground truth recordings in vitro and in vivo," *Elife*, vol. 7, p. e34518, 2018.

[26] J. Lee, C. Mitelut, H. Shokri, I. Kinsella, N. Dethe, S. Wu, K. Li, E. B. Reyes, D. Turcu, E. Batty *et al.*, "Yass: Yet another spike sorter applied to large-scale multi-electrode array recordings in primate retina," *BioRxiv*, pp. 2020–03, 2020.

[27] M. Saif-ur Rehman, R. Lienkämper, Y. Parpaley, J. Wellmer, C. Liu, B. Lee, S. Kellis, R. Andersen, I. Iossifidis, T. Glasmachers *et al.*, "Spikedeeptector: A deep-learning based method for detection of neural spiking activity," *Journal of neural engineering*, vol. 16, no. 5, p. 056003, 2019.

[28] N. A. Steinmetz, C. Aydin, A. Lebedeva, M. Okun, M. Pachitariu, M. Bauza, M. Beau, J. Bhagat, C. Böhm, M. Broux *et al.*, "Neuropixels 2.0: A miniaturized high-density probe for stable, long-term brain recordings," *Science*, vol. 372, no. 6539, p. eabf4588, 2021.

[29] T. Zhang, C. Lammie, M. R. Azghadi, A. Amirsoleimani, M. Ahmadi, and R. Genov, "Toward a formalized approach for spike sorting algorithms and hardware evaluation," in *2022 IEEE 65th International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE, 2022, pp. 1–4.

[30] K. H. Kim and S. J. Kim, "Neural spike sorting under nearly 0-db signal-to-noise ratio using nonlinear energy operator and artificial neural-network classifier," *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 10, pp. 1406–1411, 2000.

[31] M. Zamani and A. Demosthenous, "Feature extraction using extrema sampling of discrete derivatives for spike sorting in implantable upper-limb neural prostheses," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 4, pp. 716–726, 2014.

[32] M. Shaeri and A. M. Sodagar, "A framework for on-implant spike sorting based on salient feature selection," *Nature communications*, vol. 11, no. 1, pp. 1–9, 2020.

[33] A. B. Rapeaux and T. G. Constandinou, "Implantable brain machine interfaces: first-in-human studies, technology challenges and trends," *Current opinion in biotechnology*, vol. 72, pp. 102–111, 2021.

[34] D. Valencia and A. Alimohammad, "An efficient hardware architecture for template matching-based spike sorting," *IEEE transactions on biomedical circuits and systems*, vol. 13, no. 3, pp. 481–492, 2019.

[35] A. M. Kamboh and A. J. Mason, "Computationally efficient neural feature extraction for spike sorting in implantable high-density recording systems," *IEEE transactions on neural systems and rehabilitation engineering*, vol. 21, no. 1, pp. 1–9, 2012.

[36] S. Gibson, J. W. Judy, and D. Marković, "An fpga-based platform for accelerated offline spike sorting," *Journal of neuroscience methods*, vol. 215, no. 1, pp. 1–11, 2013.

[37] M. S. Lewicki, "A review of methods for spike sorting: the detection and classification of neural action potentials," *Network: Computation in Neural Systems*, vol. 9, no. 4, p. R53, 1998.

[38] S. Gibson, J. W. Judy, and D. Markovic, "Comparison of spike-sorting algorithms for future hardware implementation," in *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2008, pp. 5015–5020.

[39] S. Gibson, J. W. Judy, and D. Marković, "Spike sorting: The first step in decoding the brain: The first step in decoding the brain," *IEEE Signal processing magazine*, vol. 29, no. 1, pp. 124–143, 2011.

[40] R. Q. Quiroga, Z. Nadasdy, and Y. Ben-Shaul, "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering," *Neural computation*, vol. 16, no. 8, pp. 1661–1687, 2004.

[41] S. Mukhopadhyay and G. Ray, "A new interpretation of nonlinear energy operator and its efficacy in spike detection," *IEEE Transactions on Biomedical Engineering*, vol. 45, no. 2, pp. 180–187, 1998.

[42] D. Jones and T. Parks, "A resolution comparison of several time-frequency representations," *IEEE Transactions on Signal Processing*, vol. 40, no. 2, pp. 413–420, 1992.

[43] J. H. Choi, H. K. Jung, and T. Kim, "A new action potential detector using the mteo and its effects on spike sorting systems at low signal-to-noise ratios," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 4, pp. 738–746, 2006.

[44] R. Bestel, A. W. Daus, and C. Thielemann, "A novel automated spike sorting algorithm with adaptable feature extraction," *Journal of Neuroscience Methods*, vol. 211, no. 1, pp. 168–178, 2012.

[45] 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 Transactions on Biomedical Engineering*, vol. 50, no. 8, pp. 999–1011, 2003.

[46] Z. Nenadic and J. Burdick, "Spike detection using the continuous wavelet transform," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 1, pp. 74–87, 2005.

[47] E. Hulata, R. Segev, and E. Ben-Jacob, "A method for spike sorting and detection based on wavelet packets and shannon's mutual information," *Journal of Neuroscience Methods*, vol. 117, no. 1, pp. 1–12, 2002.

[48] Z. Nenadic and J. W. Burdick, "Spike detection using the continuous wavelet transform," *IEEE transactions*

on Biomedical Engineering, vol. 52, no. 1, pp. 74–87, 2004.

[49] K. J. Paralikar, C. R. Rao, and R. S. Clement, "New approaches to eliminating common-noise artifacts in recordings from intracortical microelectrode arrays: Inter-electrode correlation and virtual referencing," *Journal of neuroscience methods*, vol. 181, no. 1, pp. 27–35, 2009.

[50] T. Takekawa, K. Ota, M. Murayama, and T. Fukai, "Spike detection from noisy neural data in linear-probe recordings," *European Journal of Neuroscience*, vol. 39, no. 11, pp. 1943–1950, 2014.

[51] L. Chen, *Curse of Dimensionality*. Boston, MA: Springer US, 2009, pp. 545–546.

[52] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[53] A. Zviagintsev, Y. Perelman, and R. Ginosar, "Algorithms and architectures for low power spike detection and alignment," *Journal of Neural Engineering*, vol. 3, no. 1, p. 35, 2006.

[54] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," in *Fundamental Papers in Wavelet Theory*. Princeton University Press, 2009, pp. 494–513.

[55] J. C. Letelier and P. P. Weber, "Spike sorting based on discrete wavelet transform coefficients," *Journal of neuroscience methods*, vol. 101, no. 2, pp. 93–106, 2000.

[56] A. Zviagintsev, Y. Perelman, and R. Ginosar, "Low-power architectures for spike sorting," in *Conference Proceedings. 2nd International IEEE EMBS Conference on Neural Engineering, 2005.* IEEE, 2005, pp. 162–165.

[57] C. R. Caro-Martín, J. M. Delgado-García, A. Gruart, and R. Sánchez-Campusano, "Spike sorting based on shape, phase, and distribution features, and k-tops clustering with validity and error indices," *Scientific reports*, vol. 8, no. 1, pp. 1–28, 2018.

[58] T. I. Aksenova, O. K. Chibirova, O. A. Dryga, I. V. Tetko, A.-L. Benabid, and A. E. Villa, "An unsupervised automatic method for sorting neuronal spike waveforms in awake and freely moving animals," *Methods*, vol. 30, no. 2, pp. 178–187, 2003.

[59] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[60] Y. Zhang, J. Han, T. Liu, Z. Yang, W. Chen, and S. Zhang, "A robust spike sorting method based on the joint optimization of linear discrimination analysis and density peaks," *Scientific Reports*, vol. 12, no. 1, p. 15504, 2022.

[61] M. R. Keshtkaran and Z. Yang, "Noise-robust unsupervised spike sorting based on discriminative subspace learning with outlier handling," *Journal of neural engineering*, vol. 14, no. 3, p. 036003, 2017.

[62] K. Balasubramanian and I. Obeid, "Fuzzy logic-based spike sorting system," *Journal of Neuroscience Methods*, vol. 198, no. 1, pp. 125–134, 2011.

[63] S. Knieling, K. S. Sridharan, P. Belardinelli, G. Naros, D. Weiss, F. Mormann, and A. Gharabaghi, "An unsupervised online spike-sorting framework," *International journal of neural systems*, vol. 26, no. 05, p. 1550042, 2016.

[64] C. Pouzat, O. Mazor, and G. Laurent, "Using noise signature to optimize spike-sorting and to assess neuronal classification quality," *Journal of neuroscience methods*, vol. 122, no. 1, pp. 43–57, 2002.

[65] B. Clarke, E. Fokoue, and H. H. Zhang, *Principles and theory for data mining and machine learning*. Springer Science & Business Media, 2009.

[66] D. A. Reynolds, "Gaussian mixture models." *Encyclopedia of biometrics*, vol. 741, pp. 659–663, 2009.

[67] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[68] M. Blatt, S. Wiseman, and E. Domany, "Superparamagnetic clustering of data," *Physical review letters*, vol. 76, no. 18, p. 3251, 1996.

[69] G. Zouridakis and D. C. Tam, "Identification of reliable spike templates in multi-unit extracellular recordings using fuzzy clustering," *Computer methods and programs in biomedicine*, vol. 61, no. 2, pp. 91–98, 2000.

[70] M.-S. Yang and Y. Nataliani, "Robust-learning fuzzy c-means clustering algorithm with unknown number of clusters," *Pattern Recognition*, vol. 71, pp. 45–59, 2017.

[71] G. Regalia, S. Coelli, E. Biffi, G. Ferrigno, and A. Pedrocchi, "A framework for the comparative assessment of neuronal spike sorting algorithms towards more accurate off-line and on-line microelectrode arrays data analysis," *Computational intelligence and neuroscience*, vol. 2016, 2016.

[72] K. J. Laboy-Juárez, S. Ahn, and D. E. Feldman, "A normalized template matching method for improving spike detection in extracellular voltage recordings," *Scientific reports*, vol. 9, no. 1, pp. 1–12, 2019.

[73] A. P. Buccino, S. Garcia, and P. Yger, "Spike sorting: new trends and challenges of the era of high-density probes," *Progress in Biomedical Engineering*, 2022.

[74] S. Yamada, H. Kage, M. Nakashima, S. Shiono, and M. Maeda, "Data processing for multi-channel optical recording: action potential detection by neural network," *Journal of Neuroscience Methods*, vol. 43, no. 1, pp. 23–33, 1992.

[75] R. Chandra and L. Optican, "Detection, classification, and superposition resolution of action potentials in multiunit single-channel recordings by an on-line real-time neural network," *IEEE Transactions on Biomedical Engineering*, vol. 44, no. 5, pp. 403–412, 1997.

[76] M. Rácz, C. Liber, E. Németh, R. Fiáth, J. Rokai, I. Harmati, I. Ulbert, and G. Márton, "Spike detection and sorting with deep learning," *Journal of neural engineering*, vol. 17, no. 1, p. 016038, 2020.

[77] Z. Li, Y. Wang, N. Zhang, and X. Li, "An accurate and robust method for spike sorting based on convolutional

neural networks," *Brain Sciences*, vol. 10, no. 11, p. 835, 2020.

[78] J. Wouters, F. Kloosterman, and A. Bertrand, "A neural network-based spike sorting feature map that resolves spike overlap in the feature space," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 1175–1179.

[79] X. Ying, "An overview of overfitting and its solutions," in *Journal of physics: Conference series*, vol. 1168. IOP Publishing, 2019, p. 022022.

[80] B. Zhang, Z. Jiang, Q. Wang, J.-S. Seo, and M. Seok, "A neuromorphic neural spike clustering processor for deep-brain sensing and stimulation systems," in *2015 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*. IEEE, 2015, pp. 91–97.

[81] T. Werner, E. Vianello, O. Bichler, D. Garbin, D. Cattaert, B. Yvert, B. De Salvo, and L. Perniola, "Spiking neural networks based on oxram synapses for real-time unsupervised spike sorting," *Frontiers in neuroscience*, vol. 10, p. 474, 2016.

[82] M. Bernert and B. Yvert, "An attention-based spiking neural network for unsupervised spike-sorting," *International journal of neural systems*, vol. 29, no. 08, p. 1850059, 2019.

[83] A. K. Mukhopadhyay, I. Chakrabarti, A. Basu, and M. Sharad, "Power efficient spiking neural network classifier based on memristive crossbar network for spike sorting application," *arXiv preprint arXiv:1802.09047*, 2018.

[84] F. Boi, T. Moraitis, V. De Feo, F. Diotalevi, C. Bartolozzi, G. Indiveri, and A. Vato, "A bidirectional brain-machine interface featuring a neuromorphic hardware decoder," *Frontiers in neuroscience*, vol. 10, p. 563, 2016.

[85] H. Markram, W. Gerstner, and P. J. Sjöström, "Spike-timing-dependent plasticity: a comprehensive overview," *Frontiers in synaptic neuroscience*, vol. 4, p. 2, 2012.

[86] D. O. Hebb, *The organization of behavior: A neuropsychological theory*. Psychology Press, 2005.

[87] M. R. Azghadi, N. Iannella, S. F. Al-Sarawi, G. Indiveri, and D. Abbott, "Spike-based synaptic plasticity in silicon: design, implementation, application, and challenges," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 717–737, 2014.

[88] R. Pathak, S. Dash, A. K. Mukhopadhyay, A. Basu, and M. Sharad, "Low power implantable spike sorting scheme based on neuromorphic classifier with supervised training engine," in *2017 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE, 2017, pp. 266–271.

[89] C. Rossant, S. N. Kadir, D. F. Goodman, J. Schulman, M. L. Hunter, A. B. Saleem, A. Grosmark, M. Belluscio, G. H. Denfield, A. S. Ecker *et al.*, "Spike sorting for large, dense electrode arrays," *Nature neuroscience*, vol. 19, no. 4, pp. 634–641, 2016.

[90] K. D. Harris, D. A. Henze, J. Csicsvari, H. Hirase, and G. Buzsaki, "Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements," *Journal of neurophysiology*, vol. 84, no. 1, pp. 401–414, 2000.

[91] S. N. Kadir, D. F. Goodman, and K. D. Harris, "High-dimensional cluster analysis with the masked em algorithm," *Neural computation*, vol. 26, no. 11, pp. 2379–2394, 2014.

[92] F. J. Chaure, H. G. Rey, and R. Quian Quiroga, "A novel and fully automatic spike-sorting implementation with variable number of features," *Journal of neurophysiology*, vol. 120, no. 4, pp. 1859–1871, 2018.

[93] 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," *Journal of neuroscience methods*, vol. 154, no. 1-2, pp. 204–224, 2006.

[94] J. E. Chung, J. F. Magland, A. H. Barnett, V. M. Tolosa, A. C. Tooker, K. Y. Lee, K. G. Shah, S. H. Felix, L. M. Frank, and L. F. Greengard, "A fully automated approach to spike sorting," *Neuron*, vol. 95, no. 6, pp. 1381–1394, 2017.

[95] J. F. Magland and A. H. Barnett, "Unimodal clustering using isotonic regression: Iso-split," *arXiv preprint arXiv:1508.04841*, 2015.

[96] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[97] R. J. Chandler, S. Gibson, V. Karkare, S. Farshchi, D. Markovic, and J. W. Judy, "A system-level view of optimizing high-channel-count wireless biosignal telemetry," in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2009, pp. 5525–5530.

[98] T.-C. Chen, T.-C. Ma, Y.-Y. Chen, and L.-G. Chen, "Low power and high accuracy spike sorting microprocessor with on-line interpolation and re-alignment in 90nm cmos process," in *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2012, pp. 4485–4488.

[99] D. Valencia and A. Alimohammad, "A real-time spike sorting system using parallel osort clustering," *IEEE transactions on biomedical circuits and systems*, vol. 13, no. 6, pp. 1700–1713, 2019.

[100] V. Karkare, S. Gibson, and D. Marković, "A 75-μw, 16-channel neural spike-sorting processor with unsupervised clustering," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 9, pp. 2230–2238, 2013.

[101] L. Schäffer, Z. Nagy, Z. Kineses, and R. Fiáth, "Fpga-based neural probe positioning to improve spike sorting with osort algorithm," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2017, pp. 1–4.

[102] L. Schäffer, Z. Nagy, Z. Kincses, R. Fiáth, and I. Ulbert, "Spatial information based osort for real-time spike sorting using fpga," *IEEE Transactions on Biomedical Engineering*, vol. 68, no. 1, pp. 99–108, 2020.

[103] S. Luan, I. Williams, M. Maslik, Y. Liu, F. D. Carvalho, A. Jackson, R. Q. Quiroga, and T. G. Constandinou, "Compact standalone platform for neural recording with real-time spike sorting and data logging," *Journal of Neural Engineering*, vol. 15, no. 4, p. 046014, may 2018. [Online]. Available: https://dx.doi.org/10.1088/1741-2552/aabc23

[104] Y.-L. Chen, W.-J. Hwang, and C.-E. Ke, "An efficient vlsi architecture for multi-channel spike sorting using a generalized hebbian algorithm," *Sensors*, vol. 15, no. 8, pp. 19 830–19 851, 2015.

[105] B. Yu, T. Mak, X. Li, F. Xia, A. Yakovlev, Y. Sun, and C.-S. Poon, "Real-time fpga-based multichannel spike sorting using hebbian eigenfilters," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 1, no. 4, pp. 502–515, 2011.

[106] W.-J. Hwang, W.-H. Lee, S.-J. Lin, and S.-Y. Lai, "Efficient architecture for spike sorting in reconfigurable hardware," *Sensors*, vol. 13, no. 11, pp. 14 860–14 887, 2013.

[107] V. Karkare, S. Gibson, and D. Marković, "A 130-μw, 64-channel neural spike-sorting dsp chip," *IEEE journal of solid-state circuits*, vol. 46, no. 5, pp. 1214–1222, 2011.

[108] M. Chae, W. Liu, Z. Yang, T. Chen, J. Kim, M. Sivaprakasam, and M. Yuce, "A 128-channel 6mw wireless neural recording ic with on-the-fly spike sorting and uwb tansmitter," in *2008 IEEE International Solid-State Circuits Conference-Digest of Technical Papers*. IEEE, 2008, pp. 146–603.

[109] Z. Mohammadi, J. M. Kincaid, S. H. Pun, A. Klug, C. Liu, and T. C. Lei, "Computationally inexpensive enhanced growing neural gas algorithm for real-time adaptive neural spike clustering," *Journal of Neural Engineering*, vol. 16, no. 5, p. 056007, 2019.

[110] A. Amirsoleimani, F. Alibart, V. Yon, J. Xu, M. R. Pazhouhandeh, S. Ecoffey, Y. Beilliard, R. Genov, and D. Drouin, "In-memory vector-matrix multiplication in monolithic complementary metal–oxide–semiconductor-memristor integrated circuits: Design choices, challenges, and perspectives," *Advanced Intelligent Systems*, vol. 2, no. 11, p. 2000115, 2020.

[111] A. Sebastian, M. Le Gallo, R. Khaddam-Aljameh, and E. Eleftheriou, "Memory devices and applications for in-memory computing," *Nature nanotechnology*, vol. 15, no. 7, pp. 529–544, 2020.

[112] M. Rahimi Azghadi, Y.-C. Chen, J. K. Eshraghian, J. Chen, C.-Y. Lin, A. Amirsoleimani, A. Mehonic, A. J. Kenyon, B. Fowler, J. C. Lee *et al.*, "Complementary metal-oxide semiconductor and memristive hardware for neuromorphic computing," *Advanced Intelligent Systems*, vol. 2, no. 5, p. 1900189, 2020.

[113] C. Li, C. Lammie, X. Dong, A. Amirsoleimani, M. R. Azghadi, and R. Genov, "Seizure detection and prediction by parallel memristive convolutional neural networks," *IEEE Transactions on Biomedical Circuits and Systems*, 2022.

[114] L. Chua, "Memristor-the missing circuit element," *IEEE Transactions on circuit theory*, vol. 18, no. 5, pp. 507–519, 1971.

[115] S. Boyn, J. Grollier, G. Lecerf, B. Xu, N. Locatelli, S. Fusil, S. Girod, C. Carrétéro, K. Garcia, S. Xavier *et al.*, "Learning through ferroelectric domain dynamics in solid-state synapses," *Nature communications*, vol. 8, no. 1, pp. 1–7, 2017.

[116] M. R. Azghadi, B. Linares-Barranco, D. Abbott, and P. H. Leong, "A hybrid cmos-memristor neuromorphic synapse," *IEEE transactions on biomedical circuits and systems*, vol. 11, no. 2, pp. 434–445, 2016.

[117] Z. Liu, J. Tang, B. Gao, X. Li, P. Yao, Y. Lin, D. Liu, B. Hong, H. Qian, and H. Wu, "Multichannel parallel processing of neural signals in memristor arrays," *Science advances*, vol. 6, no. 41, p. eabc4797, 2020.

[118] I. Gupta, A. Serb, A. Khiat, M. Trapatseli, and T. Prodromakis, "Spike sorting using non-volatile metal-oxide memristors," *Faraday discussions*, vol. 213, pp. 511–520, 2019.

[119] I. Gupta, A. Serb, A. Khiat, and T. Prodromakis, "Towards a memristor-based spike-sorting platform," in *2016 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. IEEE, 2016, pp. 408–411.

[120] I. Gupta, A. Serb, A. Khiat, R. Zeitler, S. Vassanelli, and T. Prodromakis, "Real-time encoding and compression of neuronal spikes by metal-oxide memristors," *Nature communications*, vol. 7, no. 1, pp. 1–9, 2016.

[121] T.-C. Chen, K. Chen, Z. Yang, K. Cockerham, and W. Liu, "A biomedical multiprocessor soc for closed-loop neuroprosthetic applications," in *2009 IEEE International Solid-State Circuits Conference-Digest of Technical Papers*. IEEE, 2009, pp. 434–435.

[122] T.-C. Chen, W. Liu, and L.-G. Chen, "128-channel spike sorting processor with a parallel-folding structure in 90nm process," in *2009 IEEE International Symposium on Circuits and Systems*. IEEE, 2009, pp. 1253–1256.

[123] A. Serb, C. Papavassiliou, and T. Prodromakis, "A memristor-cmos hybrid architecture concept for online template matching," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2017, pp. 1–4.

[124] D. Valencia and A. Alimohammad, "Neural spike sorting using binarized neural networks," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 29, pp. 206–214, 2020.

[125] M. S. Fee, P. P. Mitra, and D. Kleinfeld, "Variability of extracellular spike waveforms of cortical neurons," *Journal of neurophysiology*, vol. 76, no. 6, pp. 3823–3833, 1996.

[126] R. Snider and A. Bonds, "Classification of non-stationary neural signals," *Journal of neuroscience methods*, vol. 84, no. 1-2, pp. 155–166, 1998.

[127] R. Toosi, M. A. Akhaee, and M.-R. A. Dehaqani, "An automatic spike sorting algorithm based on adaptive spike detection and a mixture of skew-t distributions," *Scientific Reports*, vol. 11, no. 1, pp. 1–18, 2021.

[128] K. Q. Shan, E. V. Lubenov, and A. G. Siapas, "Model-based spike sorting with a mixture of drifting t-

distributions," *Journal of neuroscience methods*, vol. 288, pp. 82–98, 2017.

[129] J. Boussard, E. Varol, H. D. Lee, N. Dethe, and L. Paninski, "Three-dimensional spike localization and improved motion correction for neuropixels recordings," *Advances in Neural Information Processing Systems*, vol. 34, pp. 22 095–22 105, 2021.

[130] V. Prochazka, B. Conrad, and F. Sindermann, "A neuroelectric signal recognition system," *Electroencephalography and clinical neurophysiology*, vol. 32, no. 1, pp. 95–97, 1972.

[131] S. Garcia, A. P. Buccino, and P. Yger, "How do spike collisions affect spike sorting performance?" *bioRxiv*, 2021.

[132] J. S. Prentice, J. Homann, K. D. Simmons, G. Tkačik, V. Balasubramanian, and P. C. Nelson, "Fast, scalable, bayesian spike identification for multi-electrode arrays," *PloS one*, vol. 6, no. 7, p. e19884, 2011.

[133] M. Lewicki, "Bayesian modeling and classification of neural signals," *Advances in Neural Information Processing Systems*, vol. 6, 1993.

[134] J. J. Jun, N. A. Steinmetz, J. H. Siegle, D. J. Denman, M. Bauza, B. Barbarits, A. K. Lee, C. A. Anastassiou, A. Andrei, Ç. Aydın *et al.*, "Fully integrated silicon probes for high-density recording of neural activity," *Nature*, vol. 551, no. 7679, pp. 232–236, 2017.

[135] G. Hilgen, M. Sorbaro, S. Pirmoradian, J.-O. Muthmann, I. E. Kepiro, S. Ullo, C. J. Ramirez, A. P. Encinas, A. Maccione, L. Berdondini *et al.*, "Unsupervised spike sorting for large-scale, high-density multielectrode arrays," *Cell reports*, vol. 18, no. 10, pp. 2521–2532, 2017.

[136] J. J. Jun, C. Mitelut, C. Lai, S. L. Gratiy, C. A. Anastassiou, and T. D. Harris, "Real-time spike sorting platform for high-density extracellular probes with ground-truth validation and drift correction," *BioRxiv*, p. 101030, 2017.

[137] D. Carlson and L. Carin, "Continuing progress of spike sorting in the era of big data," *Current opinion in neurobiology*, vol. 55, pp. 90–96, 2019.

[138] D. Fan, D. Rich, T. Holtzman, P. Ruther, J. W. Dalley, A. Lopez, M. A. Rossi, J. W. Barter, D. Salas-Meza, S. Herwik *et al.*, "A wireless multi-channel recording system for freely behaving mice and rats," *PloS one*, vol. 6, no. 7, p. e22033, 2011.

[139] R. R. Harrison and C. Charles, "A low-power low-noise cmos amplifier for neural recording applications," *IEEE Journal of solid-state circuits*, vol. 38, no. 6, pp. 958–965, 2003.

[140] R. R. Harrison, P. T. Watkins, R. J. Kier, R. O. Lovejoy, D. J. Black, B. Greger, and F. Solzbacher, "A low-power integrated circuit for a wireless 100-electrode neural recording system," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 1, pp. 123–133, 2006.

[141] N. Joye, A. Schmid, and Y. Leblebici, "A cell-electrode interface noise model for high-density microelectrode arrays," in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2009, pp. 3247–3250.

[142] D. A. Heldman and D. W. Moran, "Local field potentials for bci control," *Handbook of clinical neurology*, vol. 168, pp. 279–288, 2020.

[143] A. Hassibi, R. Navid, R. W. Dutton, and T. H. Lee, "Comprehensive study of noise processes in electrode electrolyte interfaces," *Journal of applied physics*, vol. 96, no. 2, pp. 1074–1082, 2004.

[144] C. M. López, M. Welkenhuysen, S. Musa, W. Eberle, C. Bartic, R. Puers, and G. Gielen, "Towards a noise prediction model for in vivo neural recording," in *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2012, pp. 759–762.

[145] J. Martinez, C. Pedreira, M. J. Ison, and R. Q. Quiroga, "Realistic simulation of extracellular recordings," *Journal of neuroscience methods*, vol. 184, no. 2, pp. 285–293, 2009.

[146] J. L. Teeters and F. T. Sommer, "Crcns. org: a repository of high-quality data sets and tools for computational neuroscience," *BMC Neuroscience*, vol. 10, no. 1, pp. 1–1, 2009.

[147] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical association*, vol. 66, no. 336, pp. 846–850, 1971.

[148] R. K. Brouwer, "Extending the rand, adjusted rand and jaccard indices to fuzzy partitions," *Journal of Intelligent Information Systems*, vol. 32, pp. 213–235, 2009.

[149] R. Veerabhadrappa, M. Ul Hassan, J. Zhang, and A. Bhatti, "Compatibility evaluation of clustering algorithms for contemporary extracellular neural spike sorting," *Frontiers in systems neuroscience*, vol. 14, p. 34, 2020.

[150] A. H. Barnett, J. F. Magland, and L. F. Greengard, "Validation of neural spike sorting algorithms without ground-truth information," *Journal of neuroscience methods*, vol. 264, pp. 65–77, 2016.

[151] O. Rübel, A. Tritt, R. Ly, B. K. Dichter, S. Ghosh, L. Niu, P. Baker, I. Soltesz, L. Ng, K. Svoboda *et al.*, "The neurodata without borders ecosystem for neurophysiological data science," *Elife*, vol. 11, p. e78362, 2022.

[152] A. P. Buccino, C. L. Hurwitz, S. Garcia, J. Magland, J. H. Siegle, R. Hurwitz, and M. H. Hennig, "Spikeinterface, a unified framework for spike sorting," *Elife*, vol. 9, p. e61834, 2020.

[153] J. Magland, J. J. Jun, E. Lovero, A. J. Morley, C. L. Hurwitz, A. P. Buccino, S. Garcia, and A. H. Barnett, "Spikeforest, reproducible web-facing ground-truth validation of automated neural spike sorters," *Elife*, vol. 9, p. e55167, 2020.

[154] D. N. Hill, S. B. Mehta, and D. Kleinfeld, "Quality metrics to accompany spike sorting of extracellular signals," *Journal of Neuroscience*, vol. 31, no. 24, pp. 8699–8705, 2011.

[155] M. R. Azghadi, C. Lammie, J. K. Eshraghian, M. Payvand, E. Donati, B. Linares-Barranco, and G. Indiveri,

"Hardware implementation of deep network accelerators towards healthcare and biomedical applications," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 14, no. 6, pp. 1138–1159, 2020.

[156] B. P. Christie, D. M. Tat, Z. T. Irwin, V. Gilja, P. Nuyujukian, J. D. Foster, S. I. Ryu, K. V. Shenoy, D. E. Thompson, and C. A. Chestek, "Comparison of spike sorting and thresholding of voltage waveforms for intracortical brain–machine interface performance," *Journal of neural engineering*, vol. 12, no. 1, p. 016009, 2014.

[157] N. Even-Chen, D. G. Muratore, S. D. Stavisky, L. R. Hochberg, J. M. Henderson, B. Murmann, and K. V. Shenoy, "Power-saving design opportunities for wireless intracortical brain–computer interfaces," *Nature Biomedical Engineering*, vol. 4, no. 10, pp. 984–996, 2020.

[158] W. Wu, M. Black, Y. Gao, M. Serruya, A. Shaikhouni, J. Donoghue, and E. Bienenstock, "Neural decoding of cursor motion using a kalman filter," *Advances in neural information processing systems*, vol. 15, 2002.

[159] W. Wu, Y. Gao, E. Bienenstock, J. P. Donoghue, and M. J. Black, "Bayesian population decoding of motor cortical activity using a kalman filter," *Neural computation*, vol. 18, no. 1, pp. 80–118, 2006.

[160] S. Todorova, P. Sadtler, A. Batista, S. Chase, and V. Ventura, "To sort or not to sort: the impact of spike-sorting on neural decoding performance," *Journal of neural engineering*, vol. 11, no. 5, p. 056005, 2014.

[161] R. B. Bod, J. Rokai, D. Meszéna, R. Fiáth, I. Ulbert, and G. Márton, "From end to end: Gaining, sorting, and employing high-density neural single unit recordings," *Frontiers in Neuroinformatics*, vol. 16, 2022.

[162] J. B. Isbister, V. Reyes-Puerta, J.-J. Sun, I. Horenko, and H. J. Luhmann, "Clustering and control for adaptation uncovers time-warped spike time patterns in cortical networks in vivo," *Scientific Reports*, vol. 11, no. 1, pp. 1–20, 2021.

[163] J. Ladenbauer, S. McKenzie, D. F. English, O. Hagens, and S. Ostojic, "Inferring and validating mechanistic models of neural microcircuits based on spike-train data," *Nature communications*, vol. 10, no. 1, p. 4933, 2019.

[164] T. P. Reber, M. Bausch, S. Mackay, J. Boström, C. E. Elger, and F. Mormann, "Representation of abstract semantic knowledge in populations of human single neurons in the medial temporal lobe," *PLoS Biology*, vol. 17, no. 6, p. e3000290, 2019.

[165] H. Kaku, M. Ozturk, A. Viswanathan, J. Jimenez-Shahed, S. Sheth, and N. F. Ince, "Grouping neuronal spiking patterns in the subthalamic nucleus of parkinsonian patients," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2019, pp. 4221–4224.

[166] J. Sukiban, N. Voges, T. A. Dembek, R. Pauli, V. Visser-Vandewalle, M. Denker, I. Weber, L. Timmermann, and S. Grün, "Evaluation of spike sorting algorithms: application to human subthalamic nucleus recordings and simulations," *Neuroscience*, vol. 414, pp. 168–185, 2019.

[167] W. Hutchison, R. Allan, H. Opitz, R. Levy, J. Dostrovsky, A. Lang, and A. Lozano, "Neurophysiological identification of the subthalamic nucleus in surgery for parkinson's disease," *Annals of Neurology: Official Journal of the American Neurological Association and the Child Neurology Society*, vol. 44, no. 4, pp. 622–628, 1998.

[168] O. K. Chibirova, T. I. Aksenova, A.-L. Benabid, S. Chabardes, S. Larouche, J. Rouat, and A. E. Villa, "Unsupervised spike sorting of extracellular electrophysiological recording in subthalamic nucleus of parkinsonian patients," *Biosystems*, vol. 79, no. 1-3, pp. 159–171, 2005.

[169] A. Nurmikko, "Challenges for large-scale cortical interfaces," *Neuron*, vol. 108, no. 2, pp. 259–269, 2020.

[170] S. Kim, P. Tathireddy, R. A. Normann, and F. Solzbacher, "In vitro and in vivo study of temperature increases in the brain due to a neural implant," in *2007 3rd international IEEE/EMBS conference on neural engineering*. IEEE, 2007, pp. 163–166.

[171] L. Luan, J. T. Robinson, B. Aazhang, T. Chi, K. Yang, X. Li, H. Rathore, A. Singer, S. Yellapantula, Y. Fan *et al.*, "Recent advances in electrical neural interface engineering: minimal invasiveness, longevity, and scalability," *Neuron*, vol. 108, no. 2, pp. 302–321, 2020.

[172] M. Ferguson, D. Sharma, D. Ross, and F. Zhao, "A critical review of microelectrode arrays and strategies for improving neural interfaces," *Advanced Healthcare Materials*, vol. 8, no. 19, p. 1900558, 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/adhm.201900558

[173] T. D. Kozai, A. S. Jaquins-Gerstl, A. L. Vazquez, A. C. Michael, and X. T. Cui, "Brain tissue responses to neural implants impact signal sensitivity and intervention strategies," *ACS chemical neuroscience*, vol. 6, no. 1, pp. 48–67, 2015.

[174] P. J. Gilgunn, X. C. Ong, S. N. Flesher, A. B. Schwartz, and R. A. Gaunt, "Structural analysis of explanted microelectrode arrays," in *2013 6th International IEEE/EMBS Conference on Neural Engineering (NER)*. IEEE, 2013, pp. 719–722.

[175] J. P. Seymour and D. R. Kipke, "Neural probe design for reduced tissue encapsulation in cns," *Biomaterials*, vol. 28, no. 25, pp. 3594–3607, 2007.

[176] L. Drew, "The ethics of brain-computer interfaces," *Nature*, vol. 571, no. 7766, pp. S19–S19, 2019.

[177] F. Gilbert, M. Cook, T. O'Brien, and J. Illes, "Embodiment and estrangement: results from a first-in-human "intelligent bci" trial," *Science and engineering ethics*, vol. 25, no. 1, pp. 83–96, 2019.