WMsorting: Wavelet Packets' Decomposition and Mutual Information-Based Spike Sorting Method

Libo Huang, *Student Member, IEEE*, Bingo Wing-Kuen Ling, *Senior Member, IEEE*, Ruichu Cai, *Member, IEEE*, Yan Zeng, Jiong He, and Yao Chen

***Abstract*— In recent years, the signal processing opportunities with the multi-channel recording and the high precision detection provided by the development of new extracellular multi-electrodes are increasing. Hence, designing new spike sorting algorithms are both attractive and challenging. These algorithms are used to distinguish the individual neurons' activity from the dense and simultaneously recorded neural action potentials with high accuracy. However, since the overlapping phenomenon often inevitably arises in the recorded data, they are not accurate enough in practical situations, especially when the noise level is high. In this paper, a spike feature extraction method based on the wavelet packets' decomposition and the mutual information is proposed. This is a highly accurate semi-supervised solution with a short training phase for performing the automation of the spike sorting framework. Furthermore, the evaluations are performed on different public datasets. The raw data are not only suffered from multiple noises (from 5% level to 20% level) but also includes various degrees of overlapping spikes at different times. The clustering results demonstrate the effectiveness of our proposed algorithm. In addition, it achieves a good anti-noise performance with ensuring a high clustering accuracy (up to 99.76%) compared with the state-of-the-art methods.**

***Index Terms*— Feature extraction, mutual information, spike sorting, wavelet packets decomposition**

Manuscript received March 31, 2019; accepted March 31, 2019. Date of publication April 4, 2019; date of current version June 28, 2019. This work was supported in part by the National Nature Science Foundation of China under Grant U1701266, Grant 61372173, Grant 61671163, and Grant 61876043 and in part by the Natural Science Foundation of Guangdong under Grant 2014A030306004 and Grant 2014A030308008. *(Corresponding authors: Bingo Wing-Kuen Ling; Ruichu Cai.)*

L. Huang and B. W.-K. Ling are with the Guangzhou Higher Education Mega Center, Faculty of Information Engineering, Guangdong University of Technology, Guangzhou 510006, China (e-mail: www.huanglibo@gmail.com; yongquanling@gdut.edu.cn).

R. Cai and Y. Zeng are with the Guangzhou Higher Education Mega Center, Faculty of Computer Science, Guangdong University of Technology, Guangzhou 510006, China (e-mail: cairuichu@gmail.com; yanazeng013@gmail.com).

J. He was with Advanced Digital Sciences Center, Singapore 138602. He is now with the DAMO Academy, Alibaba Group, Singapore 068811 (e-mail: hejiongsg@gmail.com)

Y. Chen is with the Guangzhou Higher Education Mega Center, Faculty of Computer Science, Guangdong University of Technology, Guangzhou 510006, China, and also with the Advanced Digital Sciences Center, Singapore 138602 (e-mail: yao.chen@adsc-create.edu.sg).

Digital Object Identifier 10.1109/TNB.2019.2909010

I. Introduction

STUDYING neural coding mechanism and synergistic behavior between neuronal ensembles based on the action potential activities is one of the most basic mechanisms in interdisciplinary fields such as neuroscience, cognitive neuropsychology and human-computer interface study. Recently, researchers have also demonstrated this study's significance in medicine for treating diseases like delirium, epilepsy, memory loss, etc. [1]. Current action potentials of neurons are mainly acquired by planting Multi-Electrode Arrays (MEA) in the corresponding tissue (e.g., cerebral parietal). However, due to the limitation of acquisition technology, as well as the small size and the large quantity of neurons, signals recorded are often overlapped and suffered from high noise. As an instance, nerve cells surrounding the MEA are first indiscriminately detected. Then the superimposed signals of the action potential, the local field potential and the noise potential are collected. They are relatively issued by a plurality of single neurons, nerve clusters or others around the probe. Fortunately, among the three kind of potentials, the action potentials generated by the nearest but different neurons are fairly varied. Thus it is possible to perform the detection of active potential on the acquired signals (i.e., spike detection), and conduct the extraction and classification of each spike to the corresponding neuron. This process is called spike sorting [2]–[4].

Two major problems to be urgently solved in the spike sorting are listed as below. 1) Since the signal acquisition technology is limited, the collected spike signals will inevitably contain noise interference which is needed to be eliminated. 2) Due to the distribution characteristics of neurons, the collected signals are irregularly superimposed by multiple spike potentials that are needed to be recognized. In particular, the shapes of the overlapping spikes will appear with various and complicated changes in which the superposition time and the unit waveform are always different. This even further aggravates the difficulties of the sorting task.

To address these issues, we developed a spike sorting method based on Wavelet Packets Decomposition (WPD) and Mutual Information (MI), named WMsorting, in this paper. More specifically, we employ WPD to represent the time-frequency characteristics of the signals. Besides,

Mutual Information along with Conditional Mutual Information (MI/CMI) of the extracted time-frequency data is utilized to eliminate most of the features which are redundant or irrelevant for the clustering. It is worth noting that the calculation of the MI/CMI between variables is inseparable from the estimation of entropy. Hence, the k-Nearest Neighbor estimation algorithm, one of the widely accepted methods in estimating the entropy, is applied. Each spike then could be substituted by the most representative features, i.e., the selected WPD coefficients, according to the estimated MI/CMI. A Fuzzy C-Means clustering algorithm [5] is used to further classify the spikes based on the information from the above processes. Experiments on 20 public datasets verify that this strategy not only has an improved classification precision when compared with state-of-the-art solutions on the data with high noise level, but also provides good extraction performance on the overlapping spikes. The overall result finally confirms our proposed method as an innovative solution to meet the challenges of spike sorting.

A shorter version of this work was presented in [6] and we make this version available in order to demonstrate more concrete results, method and discussions than the shorter one. The main contributions of this paper are as follows:

- An effective method to select the most discriminative and dependable WPD coefficients with MI/CMI for the spike signals is developed.
- A redundancy elimination solution with CMI is designed and implemented to overcome the problem with only using MI for selecting the featured WPD coefficients. Moreover, the MI/CMI of WPD coefficients is well estimated.
- Highly accurate sorting results are obtained with robustness in the presence of different testing datasets, as well as spikes with diverse noise levels and overlapping degrees.
- Effectiveness of our proposed method is demonstrated by comparing our solution with baseline solutions and state of art solutions.

The rest of this paper is organized as follows. Section II introduces the related work about spike sorting. Fundamental theory about wavelet packets decomposition and mutual information are provided in the Section III. In Section IV, the detailed procedure and algorithmic designs of our solution are proposed. The experimental evaluations on the popular public datasets are presented in Section V. Finally, Section VI concludes this paper.

II. Related Work

The spontaneous action potential is known as the dominating medium to issue and transmit messages between neurons. Studying the action potential can be of great benefits to identify the mechanism of signal transmission in brains. In other words, classifying these potentials with high accuracy and reliability will stimulate the researchers to establish the perfect and physiological theory basis so as to promote the scientific advancement of human brain. Furthermore, with the development of micro-fabrication technology, the large-scale electrode arrays are deployed to simultaneously record the signals of closely-spaced neurons in multiple channels [2], [7]. This renders our classification work more arduous. Therefore, it is crucial and imperious to develop accurate and reliable spike sorting algorithms to meet the urgent needs in the biological neural research community.

Conventional spike sorting frameworks are decomposed into four major steps, as shown in Fig.1. Specifically, those are raw data processing, spike detection, feature extraction and the clustering of the features as well as the spikes.

1) Raw data processing. Filtering the raw data with a band-pass filter so as to maintain the signals in specific frequency and reduce the noise level.

2) Spike detection. Detect the spike signals ($\sim$2 to 3ms long) with certain detection method. It is typically performed with an amplitude threshold on the filtered data. Additionally, there are some other alternative sub-steps that can be used to improve the detection of the spikes, such as spike interpolation to smooth the data, spike alignment to align all spikes to the peak, etc.

3) Feature extraction. With the detected spikes, the extraction of the most discriminative features from the spike shapes is the most critical process that could provide the information for further clustering of them.

4) Clustering of the spikes. With the guidance of the obtained feature sets, clustering the original spikes into groups is needed. These groups are named the neurons corresponding to the specific spikes.

As shown in the left side of Fig.1, the neurons (red, blue, green) near the electrode tip implanted in the cerebral parietal transmit relatively distinct action potentials, and those can be separated by the spike classification technique. However, for neurons (black) far from the electrodes, although they can be detected, their potentials are weak and not easy to be separated and classified. Thereby they contribute to a composite signal, namely the multi-unit activity. Signals from farther neurons (gray) or other devices are associated with the background noise. All those signals mentioned above compose of the raw data collected from the signal acquisition probe, the sorting procedure then is needed [3], [8]. It is worth noting that we chiefly concentrate on the design and optimization of those computer-assisted procedures since they are the core steps of spike sorting.

The performance of the effectiveness of spike sorting primarily depends on the quality of discriminative features extracted from the action potentials of neurons [9]. The sorting methods based on the spike waveform characteristics can be dated back to 1965, when Simon et al. stressed the importance of the process of extracting feature in spike sorting projects [10]. Glaser et al. firstly integrated pattern recognition methods for the extraction of the features of spikes [11]. Among the numerical methods in the literature, Principal Component Analysis (PCA) based method is one of the most classical and successful ones. It is still widely adopted in numerical spike sorting applications today [7], [12], [13]. However, when using single-electrode multi-channel acquisition techniques, the signal superposition phenomenon is inevitably prevailing [13]. Due to PCA's ineffectiveness on discovering the features of single spike amid

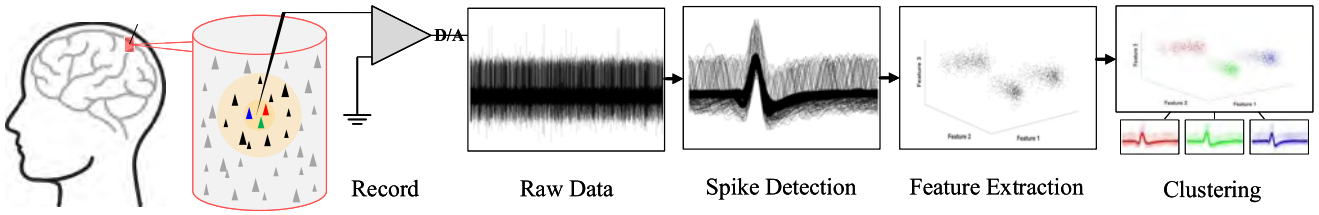


Fig. 1. Flow chart of conventional spike sorting framework.

the overlapped spikes, it is challenging for it to distinguish the superposition spikes with similar morphology. To relieve such overlapping problem of spike sorting, an emblematic approach arises which is called template matching [14]. However, as a primary step, forming the template of various spike morphologies strictly requires the prior knowledge of the signals, which limits its application. Another kind of emblematic approaches are based on current popular neural network algorithms [2]. It aims at training common decision boundaries to resolve the overlapping spikes. The serious drawbacks of these approaches, however, are that the decision boundaries seriously depend on the initial labeling and a long training stage is also required. Similar with the methods based on neural networks, the classification performance of the noise assist strategies also depends on the level of labeling, when it is introduced into the spike sorting [15]. Meanwhile, this strategy's performance would be heavily influenced by the amount of training data. Additionally, Independent Component Analysis is introduced to separate the overlapping spikes due to its brilliant capabilities in blind source separation. But it often confronts the problem that the required number of overlapping spikes should be greater than that of spike components [16].

Furthermore, wavelet and wavelet packets are broadly applied [17]–[19] to extract features in neuron research community because they can well express the differences between various spike signals. Since wavelet packets are conducted in both low- and high-frequency domain while wavelet only in low-frequency domain, the information obtained through wavelet packets is more full-scale than that through wavelet.

Hence, inspired from existing works [3], [7], [8], [20], [21] as well as the excellent performance of MI/CMI in feature engineering [22], [23], a novel spike sorting solution based on WPD and MI/CMI, WMsoting, is presented. WPD is applied mainly to decompose every spike into wavelet parent function space so that the decomposed spikes can obtain considerable feature with time-frequency information. MI and CMI are regarded as the feature selection criteria because they can not only well represent the dependency between two or more random variables, but also possess the robustness for classifiers. In other words, they are the best potential solution to select the most representative locations of the WPD coefficients, whose details can be referred in our WMsorting mechanism.

III. Preliminaries

In this section we review the process of Wavelet Packets Decomposition, Mutual Information and Conditional Mutual Information to demonstrate their capacities in feature extraction and selection theoretically.

A. Wavelet Packets Decomposition

Wavelet Packets Decomposition is a direct expansion of the conventional Wavelet Transform (WT), which can well possess location features in both time and frequency domains.

Specifically, WT with respect to one continuous signal $f(t)$ can be defined as Eq.1.

$$L_\Psi f(a,t) = \frac{1}{\sqrt{(|a|)}} \int_{-\infty}^{\infty} \Psi(\frac{u-t}{a}) f(u) \mathrm{d}u \tag{1}$$

where $\Psi_{a,t}(u)$ is a function that represents the resulting wavelet based on a specific mother wavelet parameterized by two independent variables u and a. u is an additional time variable of describing the temporal shift of the wavelet against the actual signal and is acting on the time domain. a determines the actual frequency and is the modified variant to scale the mother wavelet, generating different daughter wavelets to highlight the specific frequency domains of the decomposed signal.

As for the discrete signals, WT can be treated as an iterative decomposition on this time signal at a low-pass scale, thereby the multi-resolution analysis in the low frequency space arises. Hence, due to the frequency-time uncertainty principle, as each splitting process is performed, the obtained frequency scale resolution increases while the time resolution decreases accordingly. However, multi-resolution analysis in the standard discrete WT can only concentrate on the approximation space instead of the detail space, which will ignore the crucial information of high-pass components.

On the contrary, WPD continues to decompose both the high- and low-pass components at each step of the analysis process. It is scilicet that WPD can decompose both the approximation space and detail space to get new lower resolution approximation spaces plus detail spaces. The definition of WPD is given in the following. Let $\phi(t)$ and $\psi(t)$ denote the scaling function and the corresponding wavelet mother function, respectively. Define $\Psi^0(t) = \phi(t)$ and $\Psi^1(t) = \psi(t)$, then we can construct the wavelet basis function based on the well-known two-scale equations, shown as below.

$$\Psi_{j,k}^{2i}(t) = \frac{1}{\sqrt{2}} \Psi^{2i}\left(\frac{2^j k - t}{2^j}\right) \tag{2}$$

$$\Psi_{j,k}^{2i+1}(t) = \frac{1}{\sqrt{2}} \Psi^{2i+1}\left(\frac{2^j k - t}{2^j}\right) \tag{3}$$

where i is the index of the node and j indicates the level of decomposition. The corresponding discrete WT coefficients

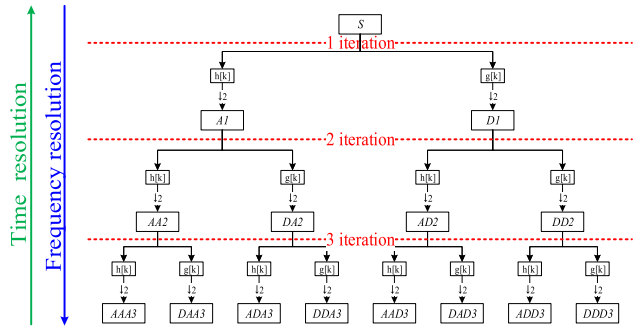


Fig. 2. A three level wavelet packets decomposition tree.

of the above signal at j^{th} level and k^{th} point then can be computed via the following Eq.4 and Eq.5:

$$l_j^{2i}(k) = \int f(t)\Psi_{j,k}^{2i}(t)\mathrm{d}t = \sum_n h(n)d_{j-1}^{i}(2k-n) \quad (4)$$

$$l_j^{2i+1}(k) = \int f(t)\Psi_{j,k}^{2i+1}(t)\mathrm{d}t = \sum_n g(n)d_{j-1}^{i}(2k-n) \quad (5)$$

where $h(n)$ and $g(n) = (-1)^{1-n}h(n-1)$ are a pair of quadrature mirror filters [17].

The resulting WPD can be viewed as a complete wavelet packets tree. An example with 3-level WPD is illustrated in Fig.2, where symbols "S", "A" and "D" represent the original signal, the approximation coefficient and the detail coefficient, respectively. $\downarrow 2$ indicates a sub-sampling process. As shown in Fig.2, both the approximations and details at a certain level are decomposed at the next level in the wavelet packets analysis, which means the WPD can provide a more precise frequency resolution than the WT. Further, nodes in this WPD tree correspond to different sets of WPD coefficients, which can be generated by different wavelet filters. Since each coefficient describes the magnitude of a certain frequency range within a specific time interval, there always exists a considerable amount of redundant information with every additional iteration [24]. Consequently, derivation of the wavelet scale coefficients is one of the most crucial steps in spike sorting.

B. Mutual Information and Conditional Mutual Information

Mutual Information can measure the shared information between two random variables, while the Conditional Mutual Information is that of the expected shared information between two variables given a third one.

The MI of two continuous variables X and Y is defined as Eq.6,

$$I(X;Y) = \iint p(x,y)log\frac{p(x,y)}{p(x)p(y)}\mathrm{d}x\mathrm{d}y \quad (6)$$

where $p(x)$ and $p(y)$ are the distributions of variables X and Y, respectively, and $p(x,y)$ is their joint distribution. From its definition, it can be viewed that MI is the similarity of the joint distribution $p(x,y)$ to the product of factored marginal distribution $p(x)p(y)$. Specifically, when the MI between X and Y is zero, it means they are independent to each other, and as the MI value increases, it presents the information they shared is increasing.

The CMI of X and Y given a third random variable Z is defined as Eq.7,

$$I(X;Y|Z) = \iiint p(x,y,z)\log\frac{p(z)p(x,y,z)}{p(x,z)p(y,z)}\mathrm{d}x\mathrm{d}y\mathrm{d}z \quad (7)$$

where $p(x,y,z)$ is the joint distributions of variables X, Y and Z.

Based on the characteristic of MI, it could be presented by the entropy and joint entropy, which are used for measuring the uncertainty of variables [22], [25]. The entropy of a random continuous variable and the joint entropy of two variables are presented in Eq.8 and in Eq.9, respectively. It is worth noting that we have limited the joint entropy to a two-variable case for the sake of simplicity, but there is no such limitation in practice.

$$H(X) = -\int p(x)\log p(x)\mathrm{d}x \quad (8)$$

$$H(X,Y) = -\iint p(x,y)\log p(x,y)\mathrm{d}x\mathrm{d}y \quad (9)$$

Correspondingly, for discrete random variables, the entropy and joint entropy can be further defined as below.

$$H(X) = -\sum_{x\in X} p(x)\log p(x) \quad (10)$$

$$H(X,Y) = -\sum_{x\in X}\sum_{y\in Y} p(x,y)\log p(x,y) \quad (11)$$

Combining Eq.6, Eq.8 and Eq.9, the MI of two variables can be further formulated with the entropy and the joint entropy as shown in Eq.12. In the same way, with Eq.7, Eq.8 and Eq.9, the CMI can be obtained shown in Eq.13.

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \quad (12)$$

$$I(X;Y|Z) = H(X,Z) + H(Y,Z) - H(X,Y,Z) - H(Z) \quad (13)$$

IV. WMsorting Framework

Due to the limitations of signal acquisition technology, there must inevitably exist noise interference needed to be eliminated in the raw spike signals. Furthermore, because of the distribution characteristics of the neurons, those signals are irregularly superimposed by a plurality of single action potentials. Such potentials also need to be identified. What's worse is that the superimposed signals can be credibly complicated because 1)the superposition time is disparate; 2) the duration of overlapping is various; 3) the superposed spike units are different. In this section, to address the overlapping and noise disturbance problems, we propose a solution to sort spikes accurately, called WMsorting. The specific characteristics are shown as below.

WMsorting follows a similar but different four major steps, compared with the conventional framework presented in Section II. Details are demonstrated in Fig.3: In step 1, the pre-process for filtering and detecting the raw data is needed. In step 2, WPD is utilized to tackle the pre-processed

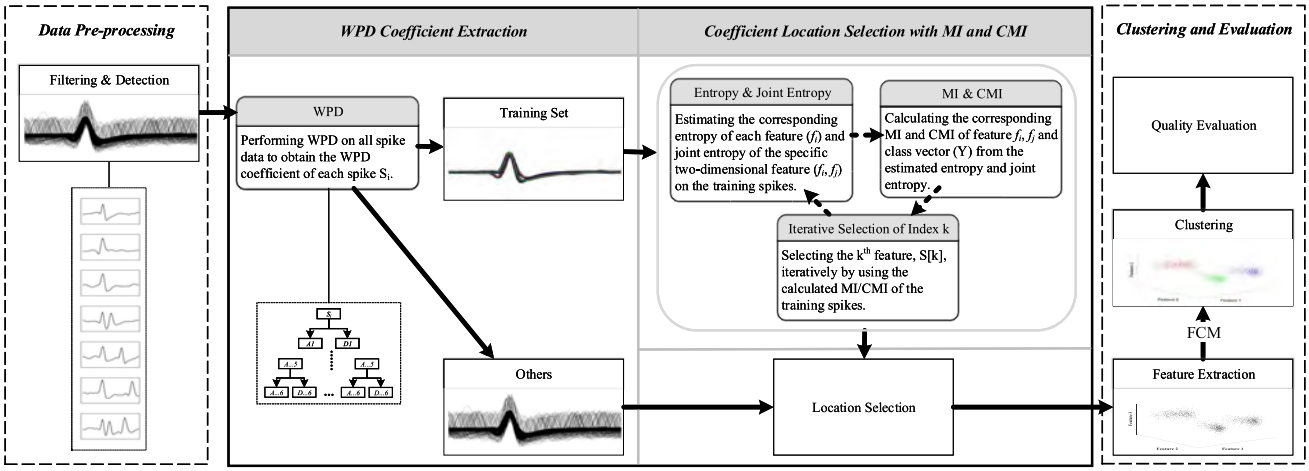


Fig. 3. Major steps of the WMsorting framework.

data so that feature coefficients of each spike can be obtained. In step 3, only a small set of the spikes are randomly chosen as the training dataset before estimating the MI/CMI between the label vectors and the feature coefficients of spikes. Then in accordance with these MI/CMI values, we iteratively select the dominated WPD coefficient locations as feature indexes. Finally in step 4, with the feature indexes extracted in step 3, we cluster the remaining spikes through performing a clustering algorithm, such as Fuzzy C-means clustering algorithm [5].

A. Data Pre-Processing

Data pre-processing here includes filtering the raw data and detecting spikes. Both of them are indispensable. This is because filtering enables the contributions of spikes and local field potentials to be separated while detecting enables the process of preventing spikes from background noise.

Firstly, the collected raw signals are filtered by a band-pass filter, a four pole butterworth filter with 300-6000Hz. It means all low and high frequency activities can be kept, respectively. After that, the spikes on top of the background noisy activity can be visualized and detected, using automatic threshold detection. Hence, all the spike signals are obtained. The process above is the same as [8].

B. WPD Coefficient Extraction

In this paper we use the coefficient data decomposed from WPD directly, other than using statistical information of wavelet coefficients, sub-band energies or other transformation modes of coefficients, for feature extraction in our proposed WMsorting method for two reasons [19]:

- Compared with the methods that use statistical information of wavelet coefficients or sub-band energies, coefficients from WPD can describe the original spikes more accurately without losing any key information, while maintaining their decomposition characteristics with high resolving power.
- Compared with the methods that use transformation modes of coefficients, WPD coefficient extraction method simplifies the data processing drastically, whose coefficients can represent the time-frequency information more specifically as well.

The performance of spike signal analysis with WPD is significantly impacted by two primary parameters, the wavelet basis function and the decomposition level. Among the various wavelet basis functions, we choose *Daubechies* wavelet family for their orthogonality properties and efficient implementation capability [26]. Specifically, the *Daubechies*2 (*db*2) is chosen. On the other hand, the WPD decomposition level is determined by the sampling frequency of the input signals. This is because the higher the level of decomposition, the higher the frequency resolution of the relative signal. Since the sampling frequency in our targeting dataset is 24KHz, the numerical values for each spike data is 64. We therefore set the decomposition level to be 6 in WMsorting, which is the maximum. That is, 384 coefficients for each input spike, indexed from 0 to 383 based on a WPD tree, are produced after employing WPD.

C. Coefficient Location Selection With MI and CMI

After WPD coefficients are extracted from every spike, randomly selecting a small amount of spikes as a training dataset is needed. Concretely, we select 60 spikes for each single category, referring to the work in [27]. The following procedure in this step aims to work on the selected training spikes and find out the most representative featured coefficients, selecting their corresponding indexes as output. In this section, we will first introduce the location selection algorithm based on MI/CMI then the method of calculating MI/CMI on the training spike dataset is given.

1) Coefficients Location Selection Algorithm: Selecting the locations of the coefficients means to select the indexes of the most representative WPD coefficients. There is no doubt that using all the WPD coefficients as features to clustering will confront the curse of dimensionality [20]. To avoid it, it is advisable to reduce the dimension for the raw featured spikes by conducting a location selection algorithm [22].

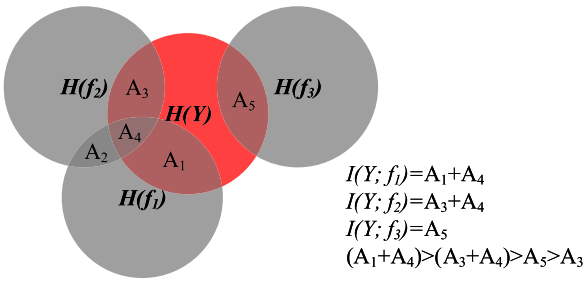


Fig. 4. An example of redundancy problem.

A detailed definition of Coefficients Location Selection is: Given a set of coefficient data $\{f_{mn}|m = 1, \ldots, M; n = 1, \ldots, N\}$ and a label vector $\{Y_n|n = 1, \ldots, N\}$, where M represents the total number of coefficients for a single spike and N stands for the sample size, the goal is to construct a dataset S, which contains $K(\ll M)$ most representative coefficient labels. (For convenience, we use f_m and Y to stand for $f_{m\cdot}$ and $Y_\cdot$, respectively, where $\cdot$ means the slice index from 1 to N.)

The coefficients location selection algorithm is an algorithm of iteratively selecting the most distinguished features [28]. When selecting the first feature, the MI between each coefficient and the label vector is calculated and the feature with the largest MI value is then achieved. In the subsequent procedure to select the next feature, the redundancy problem will occur [22], [23]. So, conditional mutual information is engaged to tackle it.

An example is illustrated in the Fig.4 to represent the redundancy problem, where $f_i (i = 1, 2, 3)$ and Y represent the coefficient feature i and the label vector, respectively, and their entropy is represented by H, characterized by a grey or red circle. $I(Y; f_1)$ denotes the mutual information between Y and f_1, which can be characterized by the overlapped area of two or more circles. That is, $I(Y; f_1) = A_1 + A_4$, $I(Y; f_2) = A_3 + A_4$ and $I(Y; f_3) = A_5$. As shown in Fig.4, we see that $(A_1 + A_4) > (A_3 + A_4) > A_5 > A_3$ holds by comparing their area sizes. Suppose the number of features needed to be selected is fixed, i.e., $K = 2$, and the first feature f_1 has been selected. Now the second one needs to be determined. Considering that A_4 is redundant since it exists in f_1 as well as f_2, selecting f_2 as the second feature will lead to redundancy problem, because the actual information gain of selecting f_2 is far less than that of selecting f_3 due to $A_3 < A_5$. If eventually redundancy problem occurs between selected features, it will undermine the accuracy of the whole method.

To ease this problem, we employ the condition mutual information. Taking Fig.4 for instance as well, when the first coefficient feature f_1 is selected, the CMI between other features f_2, f_3 and Y given f_1 ought to be calculated, other than the MI. Namely, $I(Y; f_2|f_1) = A_3$, $I(Y; f_3|f_1) = A_5$ are calculated and compared, hence the next coefficient feature f_3 is correctly determined for $I(Y; f_2|f_1) < I(Y; f_3|f_1)$. Since MI causes the redundancy problem, we combine MI with CMI for coefficient feature selection. Detailed descriptions are as follows.

Firstly, by calculating the mutual information $I(Y; f_m)$ between each feature $\{f_m|m = 1, \ldots, M\}$ and the category label vector Y, the first feature with the maximum MI is obtained. Accordingly, the corresponding index is classified into the optimal feature set S using Eq.14.

$$S[1] = \arg\max_m I(Y; f_m), \quad m = 1, \ldots, M \tag{14}$$

Secondly, update the remaining feature set $\bar{S}$. Then the CMI between f_k and Y with the given set S is approximated by the minimum CMI between f_k and Y given one of the features in S, f_i, as shown in Eq.15,

$$I(Y; f_k|S) \approx \min_{f_i \in S} I(Y; f_k|f_i), \quad f_k \in \bar{S} \tag{15}$$

where f_k is one of the predetermined features in the set $\bar{S}$. Eq.15 is broadly accepted by taking advantage of the characteristics of CMI, i.e., 1) directly approximating the $I(Y; f_k|S)$ can be inaccurate and unreliable; 2) the so-called minimum CMI is the nearest and the most correct way to estimate $I(Y; f_k|S)$ since there lies larger differences for other single features in the conditional set S.

Combining the discussion above as well as Eq.14 and Eq.15, the procedure with respect to the feature selection in S can be simply defined as Eq.16,

$$\begin{aligned} S[k] &= \arg\max_{f_k \in \bar{S}} I(Y; f_k|S) \\ &= \begin{cases} \arg\max_{f_k \in \bar{S}} I(Y; f_k), & |S| = 0 \\ \arg\max_{f_k \in \bar{S}} \min_{f_i \in S} I(Y; f_k|f_i), & |S| > 0. \end{cases} \end{aligned} \tag{16}$$

If S is empty, the resulting label of coefficient is chosen with the maximum MI value. And if not, the minimum CMI given a feature f_i in S is determined first before f_k is chosen as the next feature with the maximum CMI. It's worth mentioning that the volume of S is pre-fixed as an input parameter.

Finally, the whole selection process is iteratively performed. Note that Eq.15 has monotonic property obviously [22]. That is, if the condition $S_u \subset S_v$ is satisfied for any two selected feature subsets S_u and S_v, $\min_{f_i \in S_u} I(Y; f_k|f_i) \leq \min_{f_i \in S_v} I(Y; f_k|f_i)$ will hold. The pseudo code of this selection procedure is shown step-by-step in Alg.1.

2) MI/CMI Estimation Through Entropy and Joint Entropy: The calculation of MI/CMI in the previous selection process in WMsorting is based on entropy and joint entropy of WPD coefficients. It is worth mentioning that each spike is considered to be a continuous numerical signal. It is the same as those extracted WPD coefficients, while the associated cluster label is discrete.

In order to get the MI/CMI between the coefficients and cluster label vector, we take advantage of the discrete case as described in Section III-B, and then derive the formulation for the discrete-continuous mixtures based on training spike datasets. The entropy of coefficients X or labels vector Z is defined as follows, respectively.

$$H(X) = -\int \mu(x) \log \mu(x) \mathrm{d}x \tag{17}$$

$$H(Z) = -\sum_{z \in Z} p(z) \log p(z) \tag{18}$$

Algorithm 1 Location Selection of Coefficient

Input: The coefficient dataset f_{mn}; The label vector Y_n;
Output: The optimal feature set location S;

$S_0 \leftarrow \emptyset$
for $i = 1 \rightarrow M$ **do**
  $MI(Y; f_i|S_0) \leftarrow I(Y; f_i)$
  $flag_i \leftarrow 0$
end for
for $k = 1 \rightarrow K$ **do**
  $f_k \leftarrow \arg\max_{f_i \in \bar{S}_{k-1}} MI(Y; f_i|S_{k-1})$
  $S_k \leftarrow S_{k-1} \cup \{f_k\}$
  $\bar{S}_k \leftarrow \bar{S}_{k-1} - \{f_k\}$
  $MI(Y; f_k|S_k) \leftarrow 0$
  Update the vector $\bar{S}_k$, based on the value MI, sorting $\bar{S}_k$ in descending order
  $\max MI \leftarrow 0$
  for $f_i \in \bar{S}_k$ **do**
    if $MI(Y; f_i|S_{flag_i}) > \max MI$ **then**
      $MI(Y; f_i|S_k) \leftarrow$
      $\min\{MI(Y; f_i|S_{k-1}), MI(Y; f_i|S_k - S_{flag_i})\}$
      $flag_i \leftarrow k$
      if $MI(Y; f_i|S_k) > \max MI$ **then**
        $\max MI \leftarrow MI(Y; f_i|S_k)$
      end if
    end if
  end for
end for
return S_k;

where $\mu(x)$ is the coefficient density function of X and $p(z)$ is the labels' density function, which are easy to calculate by the ratio of each cluster's sample size to the overall volume.

Similarly, the joint entropy between spike coefficients and cluster labels can be easily derived,

$$H(X,Z) = -\iint p(x,z)\log p(x,z)\mathrm{d}x\mathrm{d}z = -\sum_{z\in Z}\int p(z)\mu(x|z)\log p(z)\mu(x|z)\mathrm{d}x \quad (19)$$

where $\mu(x|z)$ is the conditional density function of X when the specific label Z is known. When adding another coefficient Y, the feature's dimension then increases to 2. So the Eq.19 is transformed accordingly.

$$H(X,Y,Z) = -\sum_{z\in Z}\iint p(z)\mu(x,y|z)\log p(z)\mu(x,y|z)\mathrm{d}x\mathrm{d}y \quad (20)$$

where $\mu(x,y|z)$ is the conditional joint probability density function of X and Y given Z.

By far, the selection procedure can be performed as soon as the related density functions in Eq.17, Eq.19 and Eq.20 are estimated. In this paper, the k-Nearest Neighbor-based estimation algorithm is adopted, whose detailed descriptions can be referred to in the Appendix A. Further, the density estimation here considers continuous random variables of both one dimension and two dimensions.

D. Clustering the Selected Coefficients

In the last step, the set of WPD coefficients is collected from the chosen locations, and the rest of clustering process only takes the coefficient data from the location in S into consideration. The clustering algorithm can be replaced by any other alternatives on the condition that the extracted features are representative enough. Fuzzy C-Means (FCM) [5] clustering algorithm is applied, which outputs the clustered spikes with labels for further analysis.

V. Experiments and Results Analysis

To verify the effectiveness of our proposed solution, e.g., the clustering accuracy and robustness against noise, we first exhibit all the evaluation settings, including a benchmark dataset in V-A.1, the evaluation methods in V-A.2 and the evaluation criteria in V-A.3. Particularly, besides the intuitively employed baseline methods, PCA-based method as well as correlation coefficient based method, we adopt two state of art methods as our comparison targets, which are described in V-A.2.

With all these settings, we evaluate the performance of our proposed methods. It mainly consists of 1) analysis of sorting accuracy; 2) analysis of Micro and Macro average score; 3) analysis of clustering of overlapped spikes; 4) analysis of resistance to noise; and 5) analysis of stability of selected WPD coefficients. Detailed explanation of the evaluation settings and the corresponding analysis on the collected results are clearly stated in this section.

A. Evaluation Settings

1) Benchmark Datasets: To verify the effectiveness of WMsorting, we use one proverbial database from "Wave_clus" [8] as our benchmark dataset. This database is composed of four big datasets, named C_Easy1, C_Easy2, C_Difficult1, and C_Difficult2. And each dataset is constructed from 594 different average spike shapes, which are edited from recordings in the neocortex and basal ganglia of humans. Detailed descriptions of the dataset are in [29], which all the datasets used in our experiments are also online available.

The following Table I demonstrates detailed information for each dataset, in which "easy" and "difficult" are two variants of them accordingly. They are set to distinguish the overlapping degrees between spikes. Note that the noise levels are ranging from 0.05 to 0.4.

2) Evaluation Methods: We compare the results of our framework with four different methods, which are **PCA-based** and the correlation-based (**CORR**) feature extraction both with FCM as clustering methods and other two public methods, **Fusion+SVM** and **FSDE + K-means**.

The baseline approaches adopt 1) PCA to directly extract K features from the original spikes, and 2) correlation between each WPD coefficient and label vector as Eq.21 to pick the optimal related ones as features.

$$Corr_k = \frac{f_k \cdot Y}{|f_k| \cdot |Y|} \quad (21)$$

TABLE I
BENCHMARK DATASETS

DS	NL	SN(O)	Num/Clusters (Overlapped)
C_Easy1	005	3514(785)	1165(250) 1157(275) 1192(260)
	010	3522(769)	1151(248) 1134(264) 1237(257)
	015	3477(784)	1132(242) 1188(272) 1157(270)
	020	3474(796)	1198(279) 1128(248) 1148(269)
	025	3298(712)	1094(237) 1089(229) 1115(246)
	030	3475(846)	1162(294) 1164(285) 1149(267)
	035	3534(832)	1208(285) 1137(269) 1189(278)
	040	3386(741)	1079(238) 1158(261) 1149(242)
C_Easy2	005	3410(791)	1130(274) 1113(257) 1167(260)
	010	3520(826)	1160(269) 1146(280) 1214(277)
	015	3411(763)	1181(265) 1098(237) 1132(261)
	020	3526(811)	1186(262) 1188(278) 1152(271)
C_Difficult1	005	3383(767)	1115(244) 1113(256) 1155(267)
	010	3448(810)	1164(260) 1155(269) 1129(281)
	015	3472(812)	1159(275) 1172(260) 1141(277)
	020	3414(790)	1136(267) 1099(257) 1179(266)
C_Difficult2	005	3364(829)	1120(271) 1109(274) 1135(284)
	010	3462(720)	1187(230) 1136(238) 1139(252)
	015	3440(809)	1142(284) 1113(262) 1185(263)
	020	3493(777)	1151(260) 1195(277) 1147(240)

DS stands for each dataset, NL stands for noise levels, SN(O) stands for the number of spikes and (overlapped spikes) and Num/Clusters stands for the number of spikes in each cluster.

where f_k and Y indicate one feature and the label vector of a dataset, respectively and $|*|$ stands for the modulus operation. Then these two approaches use the identical clustering method (i.e., FCM in the experiments) in the sorting procedure. We take the PCA-based method as one of our baseline methods to demonstrate the validity of applying WPD and MI/CMI in WMsorting. And the CORR so as to confirm the efficacy of WMsoting during the period of selecting the WPD coefficients.

The other two comparison methods, 1) Fusion feature strategy along with Support Vector Machine (Fusion + SVM), and 2) First and Second Derivative Extrema of spike data combining with k-means clustering (FSDE + K-means), are briefly introduced as follows. The spike features of Fusion + SVM are composed by fusing 32 wavelet coefficients with 32 principal components extracted from the raw spike data through the Locality Preserving Projection algorithm [27]. With the Support Vector Machine methodology finally, the sorting procedure is finished. FSDE + K-means method is based on using both the first and the second derivatives to characterize three morphologies of the spike waveform, i.e., slopes, curvature, and amplitude. For clustering, it employs a well known semi-supervised algorithm, k-means, to finally cluster the spikes based on the information from the previous procedure [27], [30].

3) Evaluation Criteria: Two primary criteria are used to evaluate the experiment performance, 1) accuracy measured by the error rate and the misclassification number, and 2) F-measure evaluate the macroscopic along with microscopic classification effects. The followings are describing in detail about these criteria.

The error rate is one of the most well known indicators, which is calculated by the percentage of the falsely classified samples over all data. In other words, when accuracy is on behalf of the truly classification ratio, the error rate is equal to the accuracy subtracted from 1. That is, "$Error\_rate + Accuracy = 1$". And the number of misclassification means the concretely predicted quantity of those off-target samples in the clustering process.

F-measure, one commonly used as our criteria to evaluate the classification quality method, is also employed. And it is associated with the recall ($\rho_i = \frac{TP_i}{TP_i+FN_i}$) and precision ($\pi_i = \frac{TP_i}{TP_i+FP_i}$) which are directly calculated by True Positives (TP), False Positives (FP) and False Negatives (FN) within each category i. The overall F-measure score of the entire classification can be computed by two different types of average values, i.e., $micro-average$ and $macro-average$ [31]. On the one hand, micro-averaged F-measure, MicF, is computed globally over all category decisions as Eq.22,.

$$MicF = \frac{2\pi\rho}{\pi+\rho} \tag{22}$$

where ρ and π are obtained by summing all individual decisions as $\rho = \frac{\sum_{i=1}^{M} TP_i}{\sum_{i=1}^{M}(TP_i+FN_i)}$ and $\pi = \frac{\sum_{i=1}^{M} TP_i}{\sum_{i=1}^{M}(TP_i+FP_i)}$, M being the number of categories. On the other hand, the macro-averaging F-measure is defined as:

$$MacF = \frac{\sum_{i=1}^{M} F_i}{M} \tag{23}$$

where $F_i = \frac{2\pi_i\rho_i}{\pi_i+\rho_i}$ and M means the same as above.

The F-measure values fall between (0,1), and the larger F-measure values, the higher classification quality. Specifically, MicF assigns equal weight to each spike. Thus it can be regarded as an average over all spike pairs and the method's classification performance will account for the main part. As for MacF, the average of overall categories F_i is derived after they are individually computed locally. Since equal weight is assigned to each class in spite of the frequency, it is sensitive to the classification quality compared with MicF. MacF becomes more sensitive when the categories are rare. Therefore, we choose the measurements scores of MicF and MacF in that both of them are informative.

B. Results Analysis

The detailed experimental results and analysis are presented in this section.

1) Analysis of Sorting Accuracy: We firstly compare the classification accuracy of our proposed solution with other four approaches, including two baselines and two comparisons as we have described in Sec. V-A.2. The fixed feature numbers are decided on each comparison approach in [27] and [30]. Note that the number of features is always artificially determined and also beyond the range of this article. The final results are demonstrated in Table II.

Generally, our proposed approach, WMsorting, shows remarkably higher classification accuracy compared with the other four solutions, PCA-based, CORR, FSDE + SVM and Fusion + K-means. In other words, we easily derive that MI/CMI achieves the selection procedure on the WPD coefficients better than CORR. Although four comparison methods show better accuracy for several datasets with lower

TABLE II
OVERALL ACCURACY COMPARISON AGAINST OTHER FOUR SOLUTIONS

DS	NL	PCA+FCM		FSDE+K-means	CORR+FCM		Fusion+SVM	WMsorting	
		FN=3	FN=10	FN=3	FN=3	FN=10	FN=10	FN=3	FN=10
C_Easy1	005	99.37%	99.35%	94.62%	97.50%	97.38%	98.66%	**99.60%**	99.52%
	010	**99.72%**	**99.72%**	95.54%	94.04%	96.45%	98.98%	99.66%	99.57%
	015	99.25%	99.28%	94.45%	90.54%	94.94%	98.22%	**99.71%**	99.63%
	020	99.40%	99.40%	95.08%	88.77%	92.43%	97.35%	**99.57%**	99.48%
	025	99.24%	99.24%		84.41%	86.84%	95.45%	**99.45%**	99.42%
	030	98.73%	98.59%		81.50%	80.83%	88.66%	**99.57%**	99.48%
	035	97.76%	95.16%		77.02%	73.80%	83.22%	**99.43%**	99.38%
	040	96.49%	68.54%		75.58%	64.62%	78.12%	**99.76%**	99.70%
C_Easy2	005	98.48%	98.68%	94.81%	93.20%	96.04%	92.23%	**99.50%**	99.41%
	010	97.16%	98.24%	94.83%	86.02%	82.19%	92.93%	**99.52%**	99.40%
	015	92.52%	94.49%	94.96%	83.05%	82.82%	89.80%	**99.44%**	99.38%
	020	85.20%	88.60%	92.71%	79.81%	78.22%	86.24%	99.29%	**99.35%**
C_Difficult1	005	95.86%	72.54%	94.50%	83.48%	86.08%	**97.58%**	94.47%	95.00%
	010	89.56%	66.11%	94.78%	65.69%	71.55%	**94.81%**	92.89%	93.76%
	015	76.41%	61.33%	**93.81%**	57.49%	58.84%	87.85%	90.18%	91.18%
	020	63.03%	54.05%	**90.60%**	53.72%	53.81%	78.59%	85.38%	86.45%
C_Difficult2	005	98.69%	98.81%	94.38%	91.50%	94.50%	87.40%	99.23%	**99.44%**
	010	98.64%	98.76%	94.48%	90.96%	96.33%	88.07%	98.93%	**99.51%**
	015	94.39%	97.33%	87.18%	88.17%	96.02%	74.65%	98.05%	**99.51%**
	020	84.63%	83.37%	81.71%	84.77%	95.48%	67.25%	95.99%	**99.66%**

DS, NL and FN stand for the same meanings as the notations in Table I.

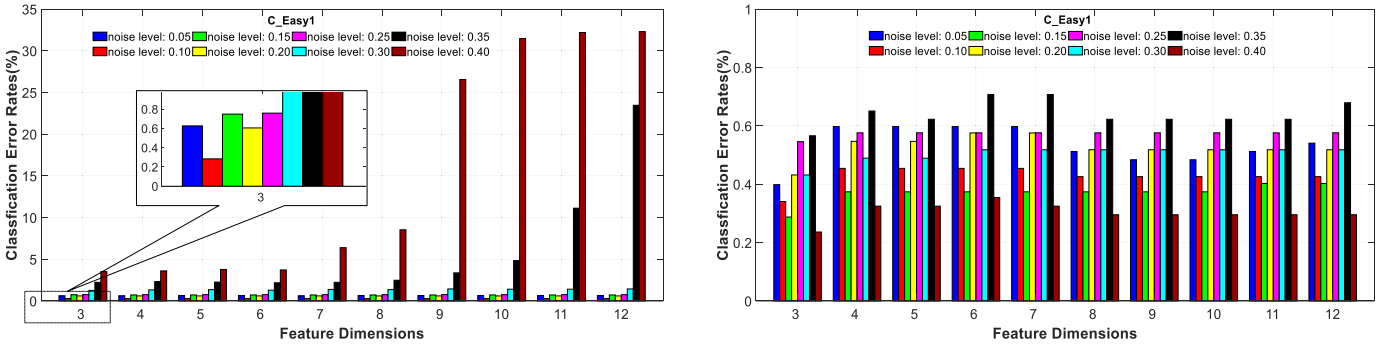


Fig. 5. Clustering error rates of PCA-based solution and WMsorting. (a) PCA-based results on C_Easy1. (b) WMsorting results on C_Easy1.

noise level, they become worse incredibly for higher noise level. Further, WMsorting is robust on the classification accuracy and still has similar accuracy on the C_Easy1 and C_Easy2 datasets as the others.

The dimension of features is an important parameter in sorting evaluation. Dataset Easy_1 is analyzed with higher noise level resolution than other three datasets. The classification error rates of Easy_1 therefore are recorded under different feature dimensions and noise levels in Fig.5. It is worth noting that the error rate is extremely unstable and can be neglected when the feature dimension is less than three.

Overall, our proposed method has lower clustering error rates at all different noise levels and all feature numbers than PCA-based results. Although the PCA-based method is stable to the feature number with lower noise (0.05 - 0.3), it becomes extremely worse after the noise level is over 0.35, whose error rate even reaches 32%. On the other hand, the results we received stay under 0.8% for all noise levels and for all the feature numbers on the same dataset, showing more stable performance. The above contrasting results verify the noise resistance and robustness property of our proposed method.

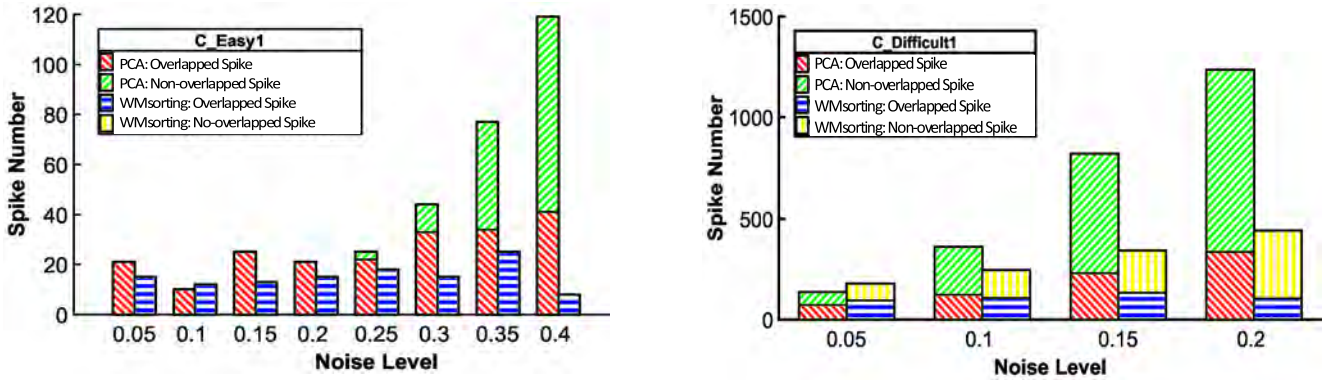
2) Analysis of Micro and Macro Average Score: In order to examine the local and global classification quality of our method, F-Measure has been tested in two baseline approaches, PCA and CORR-based methods, as well as our WMsorting. In addition, the number of wrongly clustered spikes, especially the overlapped spike, is also included.

As shown in Table III, all 20 experimental data are employed and the selected feature dimension of each data is determined with reference to the best MicF score of PCA-based method. On the one hand, we conclude that all the three methods can achieve a well locally performance focusing on the corresponding comparable MicF and MacF scores of each data. On the other hand, WMsorting finds general features even though the features number is determined by the best result from PCA-based method. Specifically, for the high noise level datasets C_Difficult1_noise020, WMsorting achieves more 22.35% MicF score improvement than that

TABLE III
F-MEASURE AND MISCLASSIFICATION NUMBERS ON ALL DATASETS

DS	NL	SN(O)	FN	PCA			CORR			WMsorting		
				MN(O)	MicF	MacF	MN(O)	MicF	MacF	MN(O)	MicF	MacF
C_Easy1	005	3514(785)	3	21(21)	99.37%	99.38%	88(88)	97.50%	97.48%	15(15)	**99.60%**	99.60%
	010	3522(769)	3	10(10)	**99.72%**	99.72%	210(79)	94.04%	94.01%	12(12)	99.66%	99.66%
	015	3477(784)	4	25(25)	99.28%	99.28%	325(116)	90.65%	90.70%	13(13)	**99.63%**	99.63%
	020	3474(796)	3	21(21)	99.40%	99.40%	390(122)	88.77%	88.58%	15(15)	**99.57%**	99.56%
	025	3298(712)	3	25(22)	99.24%	99.24%	514(127)	84.41%	84.39%	18(18)	**99.45%**	99.46%
	030	3475(846)	3	44(33)	98.73%	98.73%	43(190)	81.50%	81.50%	15(15)	**99.57%**	99.57%
	035	3534(832)	6	77(34)	97.82%	97.85%	1024(251)	71.02%	70.85%	25(25)	**99.29%**	99.29%
	040	3386(741)	3	119(41)	96.49%	96.46%	827(208)	75.58%	76.03%	8(8)	**99.76%**	99.76%
C_Easy2	005	3410(791)	12	44(44)	98.71%	98.71%	116(110)	96.60%	96.56%	18(18)	**99.47%**	99.47%
	010	3520(826)	11	61(51)	98.27%	98.24%	658(203)	81.33%	79.61%	18(18)	**99.49%**	99.49%
	015	3411(763)	17	178(80)	94.78%	94.78%	584(161)	82.88%	82.90%	25(25)	**99.27%**	99.27%
	020	3526(811)	6	396(131)	88.77%	88.85%	782(205)	77.82%	78.01%	29(29)	**99.18%**	99.18%
C_Difficult1	005	3383(767)	7	135(73)	**96.01%**	96.02%	470(182)	86.11%	86.16%	178(96)	94.74%	94.75%
	010	3448(810)	3	360(123)	89.56%	89.52%	1183(312)	65.69%	66.29%	245(107)	**92.89%**	92.89%
	015	3472(812)	3	819(231)	76.41%	76.27%	1476(374)	57.49%	58.21%	342(133)	**90.18%**	90.21%
	020	3414(790)	3	1263(335)	63.03%	62.80%	1580(375)	53.72%	54.06%	499(165)	**85.38%**	85.56%
C_Difficult2	005	3364(829)	12	39(39)	98.84%	98.84%	165(89)	95.10%	95.12%	12(12)	**99.64%**	99.64%
	010	3462(720)	9	41(40)	98.82%	98.82%	132(67)	96.19%	96.16%	15(15)	**99.57%**	99.56%
	015	3440(809)	12	91(75)	97.35%	97.37%	128(78)	96.28%	96.27%	11(11)	**99.68%**	99.68%
	020	3493(777)	6	344(108)	90.15%	89.98%	220(101)	93.70%	93.79%	54(27)	**98.45%**	99.46%

DS, SN(O) and FN stand for the same meanings as the notations in Table I; MN(O) means the number of misclassification spikes (and overlapped spikes).



(a) (b)

Fig. 6. The clustering results of overlapped and non-overlapped spikes. (a) C_Easy1 with PCA and WMsorting. (b) C_Difficult1 with PCA and WMsorting.

of PCA-based method, which equals 63.03%. And in almost all datasets except C_Difficult1, the off-target clustering only happens on the overlapped spikes, which will be further analyzed in subsection.

3) Analysis of Clustering of Overlapped Spikes: Correctly classifying the overlapping spikes is one of the most challenging tasks in spike sorting. Fig.6 demonstrates the number of misclassification spikes over C_Easy1 and C_Difficult1 datasets using WMsorting and PCA-based method. Since the results on other two datasets show similar trends and the F-measure score of the CORR based method is much worse than other methods as displayed in Table III, there is no need to present them.

As seen in Fig.6, the WMsorting has a more satisfactory performance in both non-overlapped and overlapped spikes compared with PCA-based strategy. Transforming the spike data into wavelet basis function space, WPD procedure ensures as little effects from overlapping in the original raw data as possible. Along with the assistance from MI/CMI, WPD coefficient selected from the corresponding location could represent the original spike in a near lose-less way. Therefore, the results explicitly demonstrate WMsorting without any overlapped spike outperforms PCA-based method in dataset C_Easy1 and generally achieves fewer off-target number than PCA-based in dataset C_Difficult1.

4) Analysis of Resistance to Noise: Fig.7 compares the noise impact on WMsorting and the PCA-based solution with the same parameters and processes other than the feature selection procedure. In general, the PCA-based method indicates worse resistance to noise impact than our solution, in which the noise has a nearly negligible interference over three datasets except C_Difficult1. On the C_Difficult1 dataset, the classification

TABLE IV
THE SELECTED FEATURE INDEXES AND TIMES WITH DIFFERENT NOISE LEVELS

FN \ NL	C_Easy1(FeaIdx|SelTim)								MCF
	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	
1	44|10	42|10	42|10	42|10	42|10	42|10	44|10	44|10	42
2	163|10	44|10	44|10	44|10	118|10	44|10	118|10	118|10	44
3	226|10	118|10	118|10	118|10	243|10	118|10	187|10	187|10	250
4	227|10	163|10	187|10	187|10	250|10	187|10	250|10	243|10	290
5	228|10	187|10	250|10	250|10	314|10	250|10	306|10	250|10	291
6	244|10	226|10	290|10	253|10	315|10	290|10	314|10	390|10	314
7	250|10	250|10	314|10	314|10	380|10	306|10	315|10	306|10	380
8	290|10	290|10	315|10	315|10	44|9	314|10	380|10	314|10	187
9	291|10	291|10	380|9	380|10	290|9	315|10	243|9	315|10	226
10	314|10	314|10	253|9	290|8	187|8	380|10	290|9	380|10	227
11	316|10	315|10	291|9	291|8	254|8	243|8	42|8	42|8	315
12	379|10	379|10	379|9	227|7	291|8	226|7	226|8	226|8	379
13	380|10	380|10	164|7	254|7	306|8	291|7	227|6	228|6	118
14	42|6	253|9	226|7	379|7	164|6	119|6	244|6	244|6	253
15	187|5	227|8	227|6	382|7	379|6	164|5	295|6	291|6	243
16	377|5	316|8	316|6	226|6	253|5	227|5	291|5	227|5	306

FN and NL stand for the same meaning as the notations in Table I; FeaIdx|SelTim indicates the index of specific feature, and the corresponding selected times; MCF means the most common feature index appeared in the whole table.

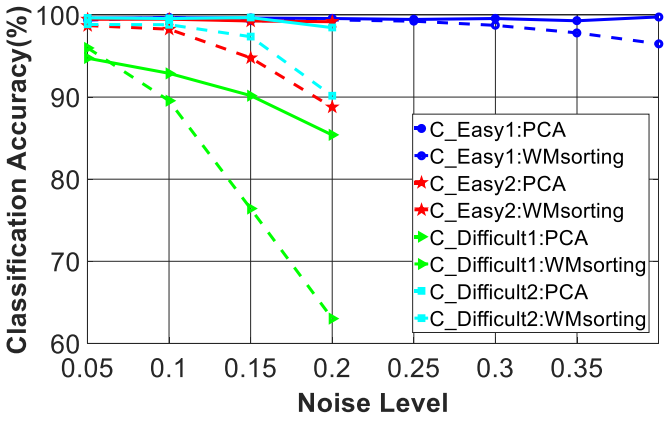


Fig. 7. Classification accuracy with different noise levels.

accuracy using PCA-based procedure rapidly decreases while ours maintains a higher accuracy until the noise level is up to 0.2. Overall, the performance of WMsorting in anti-noise is still acceptable.

5) Analysis of Stability of Selected WPD coefficients: After the preferable experimental results have been confirmed, the stability of WMsorting is also needed to be verified. That is, since the uncertainty is originated from the randomness of selecting the training sets as shown in Sec. IV, we conduct experiments on different training sets to verify its stability.

Inspired by the existing work, strategy B.632+ [22] is employed to evaluate the capability of feature selection: firstly fix the feature number as 16, iteratively run the feature selection process for 10 times on each randomly selected training datasets, and then record 16 feature indexes with the most selections over all 384 WPD coefficients. The indexes of selected features and the corresponding appearance times are presented in Table IV. Note that the features are the locations of coefficients, and the present order in the selection procedure does not affect the results. As is shown, there are only small changes on the selected features at different noise levels, which further confirm the stability and robustness of our proposed feature selection procedure.

VI. CONCLUSION

In this paper, a high accuracy spike sorting method based on Wavelet Packets Decomposition and Mutual Information, WMsorting, is proposed. It achieves a good performance in that WPD can well characterize the local time-frequency features and MI/CMI can thoroughly indicate the relevance between two or more random variables to ease the redundancy problem. Even under the situations of heavy overlapping spikes and high level noise, it still works better, compared with the state of the art methods. Definitely, compared with the CORR-based method, the selection strategy based on MI/CMI has been confirmed, selecting more representative and stable WPD coefficients. The experimental results also verify that WMsorting outperforms PCA-based methods in both clustering overlapped spikes and noise resistance, achieving up to 22.35% improvements in micro-average F-measure score.

For the sake of clarity and brevity, we have mainly focused on improving the accuracy and robustness of spike sorting, not paying much attention on considering the computational requirements and analyzing the algorithm complexity. Hence, in the future work, it is imperiously essential to explore both of them, deriving an algorithm with the trade-off between accuracy and computational complexity. Besides, integrating this high-accuracy and low-complexity algorithms into real-time systems is also attractive and challenging since real-time spike processing did matter in neuroscience and psychology research.

APPENDIX A
K-NEAREST NEIGHBOR-BASED ESTIMATION ALGORITHM

The K-Nearest Neighbor-based estimation algorithm is efficient and adaptive. It has minimal bias [32]. Given a

continuous random variable X with a proper density function $\mu(x)$, the unbiased estimator of Shannon entropy can be written as Eq.24 according to III-B.

$$H(X) = -\int \mu(x)\log\mu(x)\mathrm{d}x \approx -N^{-1}\sum_{i=1}^{N}\log\hat{\mu}(x_i) \quad (24)$$

where X has N observations $(x_1, \ldots, x_N)$ and $\hat{\mu}(x_i)$ denotes the unbiased estimators of $\mu(x)$. Denote $P_k(\varepsilon)$ by the probability distribution for the distance between x_i and its k^{th} nearest neighbor, and denote p_i by the mass of the ε ball, with $p_i(\varepsilon) = \int_{\|\xi - x_i\| < \varepsilon/2}\mu(\xi)\mathrm{d}\xi$, centered at x_i.

Using the trinomial formulation, we can get

$$P_k(\varepsilon) = k\binom{N-1}{k}\frac{\mathrm{d}p_i(\varepsilon)}{\mathrm{d}\varepsilon}p_i^{k-1}(1-p_i)^{N-k-1} \quad (25)$$

and the expectation value of $\log p_i$ is

$$\mathrm{E}(\log p_i) = \int_0^\infty P_k(\varepsilon)\log p_i(\varepsilon)\mathrm{d}\varepsilon = \psi(k) - \psi(N) \quad (26)$$

in which $\psi(x)$ is the digamma function. Assuming that $\mu(x)$ is constant in the entire ε ball, i.e., $p_i(\varepsilon) \approx c_d\varepsilon^d\mu(x_i)$, the unbiased estimators can be estimated,

$$\log\hat{\mu}(x_i) = \psi(k) - \psi(N) - d\mathrm{E}(\log\varepsilon) - \log c_d \quad (27)$$

where d stands for the dimension of x_i and c_d is the volume of d-dimensional unit ball, such as c_d is 1 for the maximum norm in this paper. And then the entropy is

$$\widehat{H}(X) = -\psi(k) + \psi(M) + \log c_d + \frac{d}{M}\sum_{i=1}^{M}\log\varepsilon(i) \quad (28)$$

where $\varepsilon(i)$ indicates the twice times distance from the sample x_i to its k^{th} neighbor.

Similarly, the unbiased estimators of the joint density and joint entropy can be written as follows,

$$\begin{aligned}\log\hat{\mu}(x_i, y_i) &= \psi(k) - \psi(N) - (d_X + d_Y)\mathrm{E}(\log\varepsilon) \\ &\quad - \log(c_{d_X}c_{d_Y}) \quad (29)\\ \widehat{H}(X,Y) &= -\psi(k) + \psi(M) + \log(c_{d_x}c_{d_y}) \\ &\quad + \frac{d_x + d_y}{M}\sum_{i=1}^{M}\log\varepsilon(i) \quad (30)\end{aligned}$$

where d is replaced by $d_X + d_Y$, c_d by $c_{d_X}c_{d_Y}$, and x_i by (x_i, y_i) corresponding to the Eq.27 and Eq.28. And the $\varepsilon(i)$ can be easily obtained when the sample dimension extends under the maximum norm. An example is illustrated in the Appendix B. For more details about this estimation method, please refer to the existing work [25], [32].

Appendix B
An Example of Maximum Norm

As illustrated in Fig.8, randomly selecting a point a among M samples, the distance from a to its k^{th} nearest neighbor point is denoted by $\varepsilon(a)/2$. Moreover, let us denote by $\varepsilon_x(a)/2$ and $\varepsilon_y(a)/2$ the distance between the same points after the projection on the subspace X and Y, respectively. Hence, $\varepsilon(a) = \max\{\varepsilon_x(a), \varepsilon_y(a)\}$ holds according to the maximum norm. Let $n_x(a)$ be the number of sample points whose distances from the point a are strictly less than $\varepsilon_x(a)/2$, and define $n_y(a)$ in the same way. If the sample point a and $k = 1$ are fixed as shown in Fig.8, we conclude that $n_x(a) = 3$ and $n_y(a) = 4$. Otherwise, if the maximum norm is not adopted, $n_y(a)$ will be less than 4 and furthermore, it is difficult for k-NN estimation algorithm to be expended into high dimensional space.

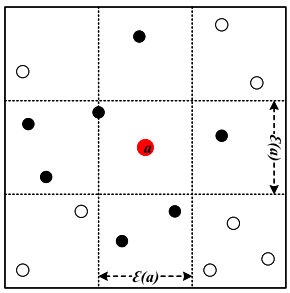


Fig. 8. An example of maximum norm.

References

[1] 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 Process. Mag.*, vol. 29, no. 1, pp. 124–143, Jan. 2012.
[2] M. S. Lewicki, "A review of methods for spike sorting: The detection and classification of neural action potentials," *Netw., Comput. Neural Syst.*, vol. 9, no. 4, pp. R53–R78, Jul. 1998.
[3] H. G. Rey, C. Pedreira, and R. Q. Quiroga, "Past, present and future of spike sorting techniques," *Brain Res. Bulletin*, vol. 119, pp. 106–117, Oct. 2015.
[4] R. Q. Quiroga, "Spike sorting," *Current Biol.*, vol. 22, no. 2, pp. R45–R46, Jan. 2012.
[5] D. A. Adamos, N. A. Laskaris, E. K. Kosmidis, and G. Theophilidis, "Nass: An empirical approach to spike sorting with overlap resolution based on a hybrid noise-assisted methodology," *J. Neurosci. Methods*, vol. 190, no. 1, pp. 129–142, Jun. 2010.
[6] Y. Chen, L. Huang, J. He, K. Zhao, R. Cai, and Z. Hao, "HASS: High accuracy spike sorting with wavelet package decomposition and mutual information," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Dec. 2018, pp. 831–838.
[7] C. Rossant *et al.*, "Spike sorting for large, dense electrode arrays," *Nature Neurosci.*, vol. 19, no. 4, pp. 634–641, Apr. 2016.
[8] R. Quian, Z. Nadasdy, and Y. Ben-Shaul, "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering," *Neural Comput.*, vol. 16, no. 8, pp. 1661–1687, Aug. 2004.
[9] A. Pavlov, V. A. Makarov, I. Makarova, and F. Panetsos, "Sorting of neural spikes: When wavelet based methods outperform principal component analysis," *Natural Comput.*, vol. 6, no. 3, pp. 269–281, Sep. 2007.
[10] W. Simon, "The real-time sorting of neuro-electric action potentials in multiple unit studies," *Electroencephalogr. Clin. Neurophysiol.*, vol. 18, no. 2, pp. 192–195, Feb. 1965.
[11] E. M. Glaser and W. B. Marks, "Separation of neuronal activity by waveform analysis," *Adv. Biomed. Eng.*, vol. 1, pp. 77–136, Jan. 1971.
[12] T. Takekawa, Y. Isomura, and T. Fukai, "Accurate spike sorting for multi-unit recordings," *Eur. J. Neurosci.*, vol. 31, no. 2, pp. 263–272, Jan. 2010.
[13] 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," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 4448–4456.
[14] P.-M. Zhang, J.-Y. Wu, Y. Zhou, P.-J. Liang, and J.-Q. Yuan, "Spike sorting based on automatic template reconstruction with a partial solution to the overlapping problem," *J. Neurosci. Methods*, vol. 135, nos. 1–2, pp. 55–65, May 2004.

[15] I. Obeid and P. D. Wolf, "Evaluation of spike-detection algorithms fora brain-machine interface application," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 905–911, Jun. 2004.
[16] G. D. Brown, S. Yamada, and T. J. Sejnowski, "Independent component analysis at the neural cocktail party," *Trends Neurosci.*, vol. 24, no. 1, pp. 54–63, Jan. 2001.
[17] J.-Z. Xue, H. Zhang, C.-X. Zheng, and X.-G. Yan, "Wavelet packet transform for feature extraction of EEG during mental tasks," in *Proc. Int. Conf. Mach. Learn. Cybern.*, vol. 1, Nov. 2003, pp. 360–363.
[18] W. Ting, Y. Guo-zheng, Y. Bang-hua, and S. Hong, "EEG feature extraction based on wavelet packet decomposition for brain computer interface," *Measurement*, vol. 41, no. 6, pp. 618–625, Jul. 2008.
[19] Y. Zhang, B. Liu, X. Ji, and D. Huang, "Classification of EEG signals based on autoregressive model and wavelet packet decomposition," *Neural Process. Lett.*, vol. 45, no. 2, pp. 365–378, Apr. 2017. doi: 10.1007/s11063-016-9530-1.
[20] R. Cai, Z. Zhang, and Z. Hao, "Bassum: A Bayesian semi-supervised method for classification feature selection," *Pattern Recognit.*, vol. 44, no. 4, pp. 811–820, Apr. 2011.
[21] R. Cai, Z. Zhang, A. K. H. Tung, C. Dai, and Z. Hao, "A general framework of hierarchical clustering and its applications," *Inf. Sci.*, vol. 272, pp. 29–48, Jul. 2014.
[22] R. Cai, Z. Hao, X. Yang, and W. Wen, "An efficient gene selection algorithm based on mutual information," *Neurocomputing*, vol. 72, nos. 4–6, pp. 991–999, Jan. 2009.
[23] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.
[24] R. Bestel, A. W. Daus, and C. Thielemann, "A novel automated spike sorting algorithm with adaptable feature extraction," *J. Neurosci. Methods*, vol. 211, no. 1, pp. 168–178, Oct. 2012.
[25] W. Gao, S. Kannan, S. Oh, and P. Viswanath, "Estimating mutual information for discrete-continuous mixtures," in *Advances in Neural Information Processing Systems*, I. Guyon *et al.*, Eds. Red Hook, NY, USA: Curran Associates, 2017, pp. 5986–5997.
[26] H. Adeli, Z. Zhou, and N. Dadmehr, "Analysis of EEG records in an epileptic patient using wavelet transform," *J. Neurosci. Methods*, vol. 123, no. 1, pp. 69–87, Feb. 2003.
[27] H. G. Li, R. Q. Song, and J. W. Liu, "Low-dimensional feature fusion strategy for overlapping neuron spike sorting," *Neurocomputing*, vol. 281, pp. 152–159, Mar. 2018.
[28] D. Koller and M. Sahami, "Toward optimal feature selection," Stanford InfoLab, Stanford, CA, USA, Tech. Rep. 1996-77, Feb. 1996. [Online]. Available: http://ilpubs.stanford.edu:8090/208/
[29] R. Q. Quiroga. *Wave_Clus: Unsupervised Spike Detection and Sorting*. Accessed: Jul. 29, 2017. [Online]. Available: https://vis.caltech.edu/~rodri/Wave_clus/Wave_clus_home.htm
[30] S. E. Paraskevopoulou, D. Y. Barsakcioglu, M. R. Saberi, A. Eftekhar, and T. G. Constandinou, "Feature extraction using first and second derivative extrema (fsde) for real-time and hardware-efficient spike sorting," *J. Neurosci. Methods*, vol. 215, no. 1, pp. 29–37, Apr. 2013.
[31] A. Özgür, L. Özgür, and T. Güüngör, "Text categorization with class-based and corpus-based keyword selection," in *Proc. Int. Symp. Comput. Inf. Sci.*, Oct. 2005, pp. 606–615.
[32] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, Jun. 2004, Art. no. 066138.