Feature-Selection-Based Transfer Learning for Intracortical Brain–Machine Interface Decoding

Peng Zhang, *Member, IEEE*, Wei Li, *Member, IEEE*, Xuan Ma, *Member, IEEE*, Jiping He, *Senior Member, IEEE*, Jian Huang, *Senior Member, IEEE*, and Qiang Li, *Member, IEEE*

***Abstract*— The time spent in collecting current samples for decoder calibration and the computational burden brought by high-dimensional neural recordings remain two challenging problems in intracortical brain-machine interfaces (iBMIs). Decoder calibration optimization approaches have been proposed, and neuron selection methods have been used to reduce computational burden. However, few methods can solve both problems simultaneously. In this article, we present a symmetrical-uncertainty-based transfer learning (SUTL) method that combines transfer learning with feature selection. The proposed method uses symmetrical uncertainty to quantitatively measure three indices for feature selection: stationarity, importance and redundancy of the feature. By selecting the stationary features, the disparities between the historical data and current data can be diminished, and the historical data can be effectively used for decoder calibration, thereby reducing the demand for current data. After selecting the important and non-redundant features, only the channels corresponding to them need to work; thus, the computational burden is reduced. The proposed method was tested on neural data recorded from two rhesus macaques to decode the reaching position or grasping gesture. The results showed that the SUTL method diminished the disparities between the historical data and current data, while achieving superior decoding performance with the needs of only ten current samples each category, less than 10% the number of features and 30% the number of neural recording channels. Additionally, unlike most studies on iBMIs, feature selection was implemented instead of neuron selection, and the average decoding accuracy achieved by the former was 6.6% higher.**

Manuscript received March 12, 2020; revised July 8, 2020; accepted October 22, 2020. Date of publication October 27, 2020; date of current version February 25, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant U1913207, Grant 61233015, Grant 61473131, and Grant 62006087; in part by the National Program on Key Basic Research Project of China (973 Program) under Grant 2013CB329506; in part by the Postdoctoral Science Foundation of China under Grant 2019M652652; and in part by the Neural Interfaces and Rehabilitation Technology Research Center, Huazhong University of Science and Technology (HUST). *(Corresponding authors: Qiang Li; Jian Huang.)*

Peng Zhang and Qiang Li are with the Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: liqiang8@hust.edu.cn).

Wei Li and Jian Huang are with the Key Laboratory of Ministry of Education for Image Processing and Intelligent Control, School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: huang_jan@mail.hust.edu.cn).

Xuan Ma is with the Department of Physiology, Feinberg School of Medicine, Northwestern University, Chicago, IL 60611 USA.

Jiping He is with the Advanced Innovation Center for Intelligent Robots and Systems, Beijing Institute of Technology, Beijing 100081, China.

Digital Object Identifier 10.1109/TNSRE.2020.3034234

***Index Terms*— Intracortical brain-machine interface, decoder calibration, transfer learning, feature selection, symmetrical uncertainty.**

I. Introduction

INTRACORTICAL brain-machine interfaces aim to help paralysed patients and amputees regain motor functions, by translating neural activities directly into motor commands to control assistive devices [1]–[4]. Neural activities are recorded by implanting neural electrodes into various brain regions, such as the primary motor cortex (M1) [1], the premotor cortex (PM) [5] and the posterior parietal cortex (PPC) [6]. As technology has advanced, iBMIs have achieved great progress [7], [8], and have been deployed in clinical research [9], [10]. However, there are still many problems and challenges, especially in clinical research [11], [12].

With the development of micro-electronics technology, electrodes with more channels have been developed, and more neurons from different brain regions can now be recorded synchronously [13], [14]. More channels guarantee more information, which, however, corresponds to greater computational burden. Redundant information in the neural data might affect the generalization ability and decoding accuracy of the decoder [15], [16]. Large computational burden places high demands on the hardware of the equipment, and results in high power consumption, thereby limiting the clinical application of the iBMIs. Therefore, our first question is, how can the computational burden be reduced while ensuring decoding performance?

Furthermore, a decoder always needs to be recalibrated prior to usage, and recalibration requires large current sample set [17]–[19]. Because the relationship between the recorded neural activities and the specific movement intentions might vary over time [19]–[21], the historical data cannot be used directly for decoder calibration and it would take a long time to collect the new data before using the iBMIs properly [22], [23]. Hence, our second question is, how can the calibration time be reduced while maintaining the decoder's performance?

A major solution to the problem presented by the first question is to ascertain the importance of each feature, and eliminate redundant ones before proceeding with decoding; this is known as feature selection in iBMIs [24], [25]. After feature selection, less features are used to build the decoder to achieve faster training speed. Only the channels corresponding

to these selected features are required to record neural signals; computational burden can be reduced with fewer channels working. For the second question, one approach is to design a calibration strategy that can achieve good performance by maximising the historical data to reduce the demand for current data, thereby significantly reducing the time devoted to collecting new training data [18]. In this study, we aim to find a method that can solve the two problems simultaneously.

Feature selection can be summarized as selecting relevance features and removing redundant features for a classification task [26]. Based on the dependency on the classifier in evaluation criteria, feature selection can be roughly divided into three categories, filter methods, wrapper methods and embedded methods [27]. The filter methods use feature ranking techniques to select the important features and filter out the less relevant variables as a pre-processing step, which is independent of the chosen classifier [28]. The filter methods select feature subsets by exploiting the essential characteristics of the training data, such as Distance measures, Information or uncertainty measures, Dependency measures and so on [28]. Relief [29], Fisher score [30] and Minimum-Redundancy-Maximum-Relevance (mRmR) [31] are among the most representative algorithms of the filter methods.

Different from the filter methods, the wrapper methods require a predetermined classifier to select the features with maximum predictive accuracy [28]. In additional to the classifier, how to search the subset of features from all possible feature subsets is also important in the wrapper methods and can be divided into two categories: Sequential and Metaheurisitc algorithms. In general, wrapper methods can achieve better predictive accuracy than the filter methods, but they require much higher computational cost due to the repeated classifier training.

The embedded methods combine the characteristics of both the filter methods and the wrapper methods and incorporate the feature selection as part of the classifier training. There are mainly three types of embedded methods, the pruning methods, the model methods with a build-in mechanism and the regularization model methods. Feature selection has been widely used in machine learning research field, such as text categorization, image processing, gene selection and so on [28]. Many brain-machine interface researches based on electroencephalogram (EEG) also integrate feature selection into their decoding algorithms [32]. However, in the iBMIs researches, instead of feature selection, neuron selection is often used, in which many features in a neuron are regarded as a whole to be evaluated and each feature is not evaluated separately.

Recently, several studies have provided tools for finding important neurons in iBMIs. Sanchez *et al.* [15] proposed three methods for quantitatively rating the importance of neurons: single neuron correlation analysis, sensitivity analysis using a vector linear model, and a model-independent cellular directional tuning analysis. Wahnoun *et al.* [33] used the individual removal error to measure each neurons contribution to the overall control. Kai *et al.* [16] proposed a local-learning-based method to rank neurons by maximizing the distance between the local neuronal patterns while deploying L1 norm regularization. More related studies are discussed in the review [34]. Many neuron selection methods have been proposed; however, these studies mainly focus on neuron selection, and ignore feature selection; they also disregard the problems in decoder calibration.

An effective way to improve the efficiency of decoder calibration is to use large historical sample set and a small current sample set. This is a typical transfer learning (TL) task [35]. TL aims to extract knowledge from the source domain (e.g. historical data) to the target domain (e.g. current data) to diminish the disparities between the two domains and does not require both domains to have the same sample distribution [36]. TL has been widely used in EEG-based brain-machine interface to transfer knowledge from session to session or subject to subject [37]–[40], however, it is rarely applied in iBMIs. In our previous study, we introduced TL into iBMIs decoder calibration and proposed a principal component analysis-based domain adaptation method which projected the historical dataset and the current dataset to a similar new feature space. Thus, the disparity among different datasets was diminished and better decoding performance was achieved [18]. Farshchian *et al.* proposed three TL methods based on the canonical correlation analysis, Kullback-Leibler divergence, and adversarial domain adaptation network; consequently, they achieved stable decoding performance even when there were very long intervals between the training data and testing data [41]. These studies effectively exploited the accumulated historical data towards achieving reliable iBMIs while reducing decoder calibration time. However, the problem of computational burden was not considered in these studies and all the channels still had to record neural signals.

In this study, we attempted to combine TL with feature selection to solve the problem of decoder calibration and reduce the computational burden of iBMIs. The SUTL method was proposed and tested on neural data recorded from two rhesus macaques to decode the reaching positions or grasping gestures. The SUTL method used symmetrical uncertainty (SU) to measure the stationarity of features between the historical data and current data. A higher SU indicated greater stationarity, and the disparity between the historical data and current data could be diminished by using only the stationary features. In addition to the stationarity of the features, the SUTL method also considered the importance and redundancy of each feature in finding the most appropriate features. For a better evaluation of the performance of the SUTL method, different calibration schemes were proposed and compared in a simulated online situation.

The main contributions of this study are as follows: 1) A combination of TL and feature selection was introduced to simultaneously solve the problem of decoder calibration and high computational burden in iBMIs. 2) Feature selection was chosen over neuron selection in the SUTL method, as a result of which better performance was achieved. 3) The SUTL method achieved better decoding performance than other calibration schemes while requiring fewer current samples, fewer features, and fewer channels to record signals. Thus, the time required for collecting new training data in decoder

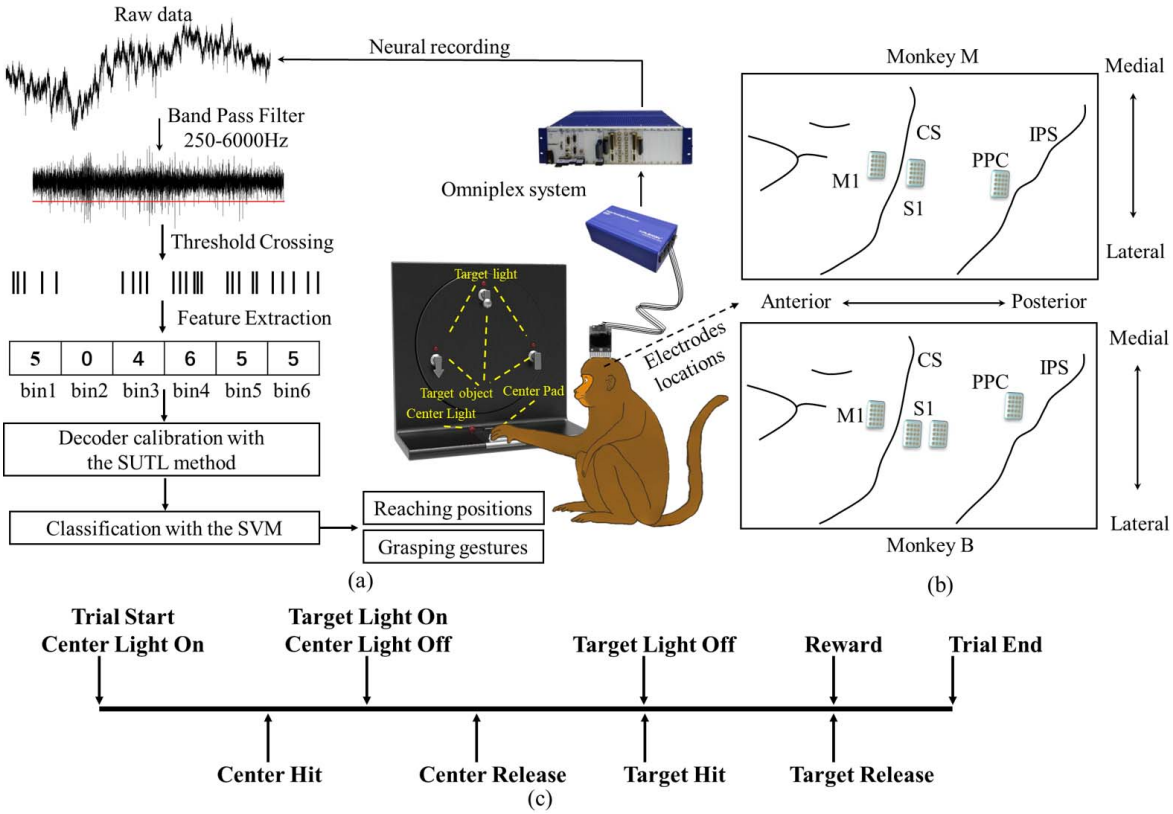

Fig. 1. The experiment procedure, electrodes implantation locations and the sequences of the experimental paradigm. (a) The experiment procedure. The monkey was trained to perform spatial reaching and grasping task while neural signals were recorded from the cortex with the Omniplex system. The raw data were bandpass filtered between 250 Hz and 6 kHz and then were applied with threshold crossing method to get spike counts. For each bin in a decoding time window, spike counts were extracted as the features. After feature extraction, data could be used to calibrate the SVM decoder with the SUTL method and then decode the neural activities into different reaching positions or grasping gestures. (b)Electrodes implantation locations. (c) The sequence of the experimental paradigm.

calibration could be significantly reduced, and computational burden could be decreased.

The rest of this article is organized as follows. Section II describes the procedure of the animal behavioural task, neural data acquisition, and feature extraction, followed by the detailed description of the SUTL method and other decoder calibration schemes. Section III presents the results of comparing the decoding performance of feature selection and neuron selection, and the effectiveness of the SUTL method and other methods; it also presents the quantitative assessment of the reduction of decoder calibration time and computational burden, and the assessment of the disparities between the historical data and current data after the SUTL method was implemented. The properties of the features selected by the SUTL were also explored. Section IV and Section V present the discussion and a brief conclusion.

II. Material and Methods

All the experiments and surgical procedures in this study were approved by the Institutional Animal Care and Use Committee at Wuhan University Center for Animal Experiment.

A. Experimental Set-up and Data Collection

The experimental procedure, electrode implant locations, and data processing procedure are shown in Fig. 1. Two adult male rhesus macaques (Monkeys M and B) were trained to perform spatial reaching and grasping tasks with their right hands; their left arms were restrained. The behavioural task was guided by an experimental apparatus, as shown in Fig. 1(a). The experimental apparatus mainly consisted of a center pad at the bottom and three target objects on the turntable of the front panel. Beside the center pad and each target object, there was a light-emitting diode. For Monkey M, the experimental apparatus contained three target objects with the same shape, and the monkey was guided to grasp the same shaped target object in different positions. For Monkey B, the experimental apparatus contained three target objects of different shapes (cube, triangle, and sphere). Monkey B was guided to reach out in the same position to grasp the target objects of different shapes; the target objects were transferred to the same position using the turntable, which was connected to a motor [42], [43].

The sequence of the behavioural task is shown in Fig. 1(c). Each trial began with the center light on, and the monkey was guided to put its hand on the center pad. After a holding time of approximately 500 ms, the center light went out, and one arbitrary target light came on, cueing the monkey to reach out and grasp the target object. The events that the monkey released the hand from the center pad and that the hand touched the target object were called “Center Release”

and "Target Hit", respectively. After a target holding time, the target light went out, and the monkey received a few drops of water as reward when released the hand. A new trial commenced after a time interval of approximately 2s.

After the monkeys had become familiar with this behavioural task, electrodes were implanted into their cortices to record neural signals. We surgically implanted two 32-channel Utah arrays and one 16-channel floating microelectrode array (FMA) (Microprobe Inc.) into the arm/hand area of the M1, somatosensory cortex (S1), and PPC of Monkey M. Four 32-channel FMA arrays were implanted in Monkey B, into the M1, S1 and PPC (two electrodes were implanted into the S1, marked as S1_a and S1_b respectively). The implantation sites were identified based on the brain landmarks, and further confirmed through intracortical microstimulation. The electrode implant locations are described in Fig. 1(b), and more details about the surgical procedure can be found in [18].

Multichannel neural signals and behavioural data were recorded using a 128-channel Omniplex system (Plexon, Inc.). The sampling rate was 40 kHz, and the neural signals were bandpass filtered between 250 Hz and 6 kHz. Spike counts were detected using the threshold crossing method, and the threshold was set to -4.5 times the root mean square value of the spike band in each channel [44], [45]. The threshold crossing method is a standard practice in iBMIs. All the spikes recorded in one channel were regarded as emanating from the same neuron. All 80 channels of the three arrays implanted in Monkey M could record neural signals, whereas only 66 channels from the arrays implanted into the S1 and the PPC of Monkey B functioned properly. Therefore, in the following analysis, there are 80 neurons for Monkey M, and 66 neurons for Monkey B.

B. Feature Extraction and Data Preparation

In this study, the neural recordings obtained from Monkey M were used to decode reaching positions (three-category classification). The time period of 200 ms before and 100 ms after the event, Center Release, was chosen to conduct the following analysis. The neural recordings from Monkey B were used to decode grasping gestures (three-category classification), and the time period of 100 ms before and 200 ms after the event, Target Hit, was chosen for analysis.

For both monkeys, the decoding time period was 300 ms and was binned by a 50-ms window. As described in Fig. 2, the spike counts in each bin were used as the feature, and each neuron contained six features. The feature vectors of all the neurons were combined, and employed as the neural activity vector that was subsequently used for decoding. For feature selection, each feature was evaluated separately; for neuron selection, on the other hand, the six features were evaluated as a whole. This was the difference between the feature selection and the neuron selection. There were 80 neurons and 66 neurons in Monkey M and Monkey B, respectively; therefore, each dataset contained 480 features from Monkey M, and 396 features from Monkey B. The experimental data used in this study was obtained from three data sessions for Monkey M, and one data session for Monkey B. Each of

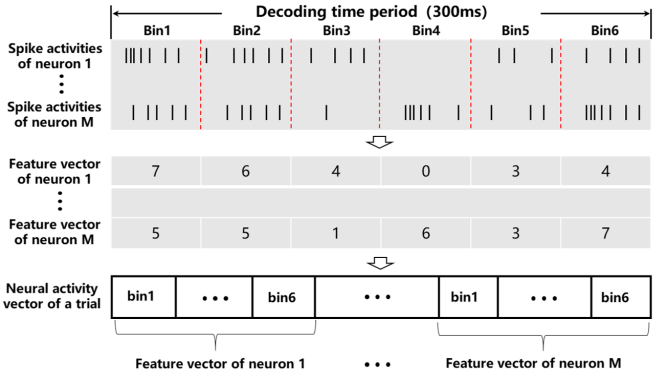

Fig. 2. The procedure of the feature extraction. The decoding time period was 300 ms and was binned into six windows. These black vertical lines represent the spike activities. The spike counts in each bin were used as the feature and the feature vector of a neuron could be obtained. For each trial, the feature vectors of all neurons were combined to form the neural activity vector.

Monkey M's sessions contained four datasets corresponding to four consecutive days; the data session of Monkey B contained ten datasets corresponding to ten consecutive days. In each session, the same number of trials were conducted for each category in each dataset, adding up to a total of more than 300 trials in each dataset for both monkeys.

C. Symmetrical-Uncertainty-Based TL (SUTL)

The SUTL method selected features by measuring correlation. The correlation of two variables could be effectively and quantitatively measured based on the SU, which was introduced by Hall [46], based on the information-theoretical concept of entropy [47]–[49]. The values of the SU between two variables lie between 0 and 1, and a higher SU value indicates higher correlation between the two variables. The value of the SU between two variables, X and Y, is defined below:

$$SU(X,Y) = \frac{2 \cdot I(X \mid Y)}{H(X) + H(Y)} \tag{1}$$

where $I(X \mid Y)$ is the mutual information between X and Y; $H(X)$ and $H(Y)$ are the entropy of the variables, X and Y, respectively. The entropy is a measure of the uncertainty of a random variable; the entropy of X is defined below:

$$H(x) = -\sum_{i} P(x_i) log_2(P(x_i)) \tag{2}$$

The mutual information $I(X \mid Y)$ is a measure of the exchanged information between these two variables and can also measure the correlation. However, the mutual information is biased in favor of features with more values; the SU overcomes this drawback by normalizing its values within the range [0, 1]. The $I(X \mid Y)$ is defined as:

$$I(X \mid Y) = H(X) - H(X \mid Y) \tag{3}$$

where $H(X \mid Y)$ is the conditional entropy of X, given Y. The conditional entropy quantifies the amount of information required to describe the outcome of a random variable, X,

given that the value of another random variable, Y, is known. The $H(X \mid Y)$ is defined as:

$$H(X \mid Y) = -\sum_{j} P(x_i) \sum_{i} P(x_i \mid y_j) log_2(P(x_i \mid y_j)) \quad (4)$$

where $P(x_i)$ is the prior probability for all the values of X, and $P(x_i \mid y_i)$ is the posterior probability of X, given the values of Y.

In the SUTL method, given the current data $A = \{p_n\}_{n=1}^{K}$, $p_n \in \mathbb{R}^{M\times 1}$, p is the feature of the current data. K is the number of the features, and M is the number of trials in the current data. The label of the current data is Y, $Y \in \mathbb{R}^{M\times 1}$. Given the historical data from one day before as $B = \{q_n\}_{n=1}^{K}$, $q_n \in \mathbb{R}^{N\times 1}$, N is the number of trials in the historical data and $N \gg M$. The label of the historical data is T, $T \in \mathbb{R}^{N\times 1}$. The SUTL method was implemented by considering the stationarity, importance and redundancy of the features. First, the stationarity of a feature was evaluated using the SU between the same feature in the historical data and current data. The stationarity evaluation worked under the premise that the data obtained on the same day were subject to the same distribution. The stationarity evaluation was marked as S, which was defined below:

$$S(n) = SU(p_n, q_n) \quad (5)$$

where n is the feature number, $n = 1 : K$. Because the sample size of the historical data was much larger than that of the current data, the historical data had to be divided into many parts; each part was composed of the same sample size as the current data. Then, for each feature, each of these parts was used to evaluate the stationarity with the current data, and their mean value was taken as the final stationarity value. The stationarity value could be ranked in descending order, and the top-J values that exceeded the threshold would be preselected as the subset of the features that were stationary.

In feature selection, the filter methods use the ranking techniques to order the features and choose a suitable threshold to select the appropriate number of top ranked features. The selection of the threshold is important, however, there is no theoretical basis to determine the threshold [50], [51]. For most studies, the threshold is chosen empirically based on a fixed number or percentage of the ranked features (such as the top 50% of the ranked features) [52], which is fast and easy to use in practical application. Additionally, threshold can be selected automatically by measuring data complexity [53], [54] or by using statistical support [51]. These researches have achieved good performance, but there is still a small problem. In these methods, the threshold is selected by a coefficient which is further determined by experience. There may not be essential difference between this method and the fixed percentage threshold method. Threshold can also be selected automatically by maximizing classification accuracy [55]. In this method, different thresholds are picked and tested to select the threshold with the best classification results. However, this method has to implement repeated classifier training which requires more time and higher computational cost. Therefore, this method might not be suitable for online application.

In the SU measurement studies, it is widely accepted by researchers that the top-J values were chosen based on a user-defined threshold [48] or by a user-defined proportion of the features [49]. Therefore, in this study, we empirically selected the top half features as the stationary features. The impact of this proportion on the decoding performance was presented in the results section and was further discussed in the discussion section.

Subsequently, the importance of the features was evaluated based on the SU between the feature and the label of the data, and was marked as I, which was defined below:

$$I(n) = SU(q_n, T) \quad (6)$$

where n is the feature number, $n = 1 : K$. Then, the features were ranked in descending order of the value of I. As with the stationarity, the top half ones were preselected as the subset of the features that were important and most relevant to the target variable. The intersection of the features between the stationary feature subset and important feature subset was selected as the new features.

For this new feature subset, the redundancy was further considered. To measure how much a candidate feature was redundant in this feature subset, we calculated the SU between this feature and any other feature in this subset. The redundancy value (R) of the feature m was defined below:

$$R(m) = \frac{\sum_k SU(F_m, F_k)}{SU(F_m, T)} \quad (7)$$

where F is the feature subset; and m and k were the feature numbers. k = 1 : size of the preselected subset, and $k \neq m$. The threshold of the redundancy value was defined below:

$$TD = \frac{mean(\sum_k SU(F_n, F_k))}{mean(SU(F_n, T))} \quad (8)$$

where F is the feature subset; and n and k are the feature numbers. $n = 1$: size of the preselected subset; $k = 1$:size of the preselected subset, and $k \neq n$. The R exceeding the threshold was considered redundant and removed from the final subset. Then, the historical data (containing many trials) and current data (containing only few trials) with selected features were combined to train the decoder; future current data could be tested by also using only these selected features. The SUTL method is applicable to both the feature selection and neuron selection, with the only difference being that the feature will be replaced by the feature subsets in the neuron selection. The pseudocode of this algorithm is shown in Algorithm 1.

D. Quantitative Evaluation of the Sample Distribution Before and After Applying the SUTL

We selected stationary and appropriate features using the proposed SUTL method. Therefore, the data following the application of the SUTL method should yield better clustering results. To quantitatively evaluate this clustering effect, the Davies-Bouldin index (DBI) of the full-dimensional data before and after applying the SUTL were measured and compared. The DBI was an internal evaluation scheme based

Algorithm 1 The SUTL.

Input: small current sample set with N features, $\{F_n\}_{n=1}^{N}$, $F_n \in \mathbb{R}^{M\times 1}$; large historical sample set with N features, $\{HF_n\}_{n=1}^{N}$, $HF_n \in \mathbb{R}^{(I*M)\times 1}$; class label of current sample set, T, $T \in \mathbb{R}^{M\times 1}$; class label of historical sample set, HT, $HT \in \mathbb{R}^{(I*M)\times 1}$;

Output: new sample set with K selected features, $\{Y_k\}_{k=1}^{K}$, $K < N$, $Y_k \in \mathbb{R}^{(I*M+M)\times 1}$;

```
//select the stationary features
Randomly divide the {HF_n}_{n=1}^N into I parts {HFP_n}_{n=1}^N, HFP_n ∈ R^{M×1}
for i = 1 to N do
    for j = 1 to I do
        SU_st(j)=SU(HFP_j, F_i);
    end for
    Compute the mean value of SU_st(j) as mean_SU_st(i);
end for
mean_SU_st(i) values were ranked in descending order;
Select the features corresponding to the top half of the mean_SU_st(i) values to be {ST_n}_{n=1}^{N/2};
//Select the important features
for i = 1 to N do
    Compute the SU between feature and class variable, SU_nT(i) = SU(F_i, T);
end for
SU_nT(i) values were ranked in descending order;
Select the features corresponding to the top half of the SU_nT values to be {SI_n}_{n=1}^{N/2};
Select the feature intersection between {ST_n}_{n=1}^{N/2} and {SI_n}_{n=1}^{N/2} as {STI_n}_{n=1}^{D}
//Eliminate the redundant features
Compute the threshold TD in Eq.(8)
for i = 1 to D do
    Compute the R in Eq.(7),
    if R < TD then
        This feature was selected to {Y_k}.
    end if
end for
```

on the ratio of the within-cluster and between-cluster distances, which was defined as:

$$DBI = \frac{1}{k}\sum_{i=1}^{k} max_{j\neq i}\left\{D_{i,j}\right\} \tag{9}$$

where $D_{i,j}$ was the ratio of the within-cluster and between-cluster distance for the i_{th} and j_{th} clusters, and was defined as:

$$D_{i,j} = \frac{\bar{d}_i + \bar{d}_j}{d_{i,j}} \tag{10}$$

$\bar{d}_i$ was the average distance between each point in the i_{th} cluster and the centroid of the i_{th} cluster. $\bar{d}_j$ was similar to $\bar{d}_i$. $d_{i,j}$ was the Euclidean distance between the centroids of the i_{th} and j_{th} clusters. For the DBI of two clusters, the lower score denotes more distinct clusters.

E. Decoder Calibration and Data Testing Schemes

In this study, we adopted the support vector machine (SVM) to decode the reaching position or grasping gesture from the neural activities. The SVM was used with the Radial Basis Function (RBF) kernel and was implemented using the LIBSVM [56]. Moreover, in the decoder calibration for each dataset, the grid-search method introduced by the LIBSVM was used to obtain the best penalty parameter, C, and the kernel parameter, γ, which were the two main parameters for the RBF kernel; they varied for each dataset testing.

To comprehensively evaluate the performance of the SUTL method, five other calibration and testing schemes were presented, and compared in a simulated online test. In the simulated online test, only the historical data and small current dataset (ten trials for each category) were used for decoder calibration; the testing data were invisible. To describe these six schemes more clearly, the diagram of the composition of the training data and testing data is shown in Fig. 3. These four schemes are as follows:

Short current session calibration (SCC): This scheme was widely used in the online test [2], [3], [5]. A few samples from the current data (ten trials of each category in this study) were used to calibrate the decoder, and the testing data from the current day were classified for testing.

Complete historical session calibration (CPC): This scheme used the entire historical data from one day before as the training data for calibrating the decoder and classified the testing data from the current day [22], [57].

Mixed session calibration (MC): This scheme directly combined the whole historical data from one day before with a few current data to calibrate the decoder, and the testing data for the current day were classified [18].

SUTL: As described in Section C, this scheme was similar to the MC scheme, which used both the historical data from one day before and a few of the current data as the training data; however, these data were not mixed directly.

Recursive Feature Elimination for Support Vector Machines(SVM-RFE): This was a widely used embedding feature selection algorithm [58]. It trained an SVM classifier iteratively and the least important features would be removed in each iteration by using weight as the ranking criterion. As with the filter method, ranked features could be obtained after applying the SVM-RFE. The training and testing data used in this algorithm were the same as in the SUTL.

Fisher score(Fscore): This was a widely used filter feature selection algorithm and was a simple but effective technique to measure the discrimination of two sets [30]. The training and testing data used in this algorithm were the same as in the SUTL. The Fscore and the SVM-RFE were presented to be compared with the SUTL in feature selection.

To establish a better basis for comparing these decoder calibration and testing schemes, the same testing data were used in all of them. For the small portions of the current data used in some schemes, the data for these 30 trials were randomly chosen from the remaining current data and were repeated 100 times to achieve the mean decoding accuracy.

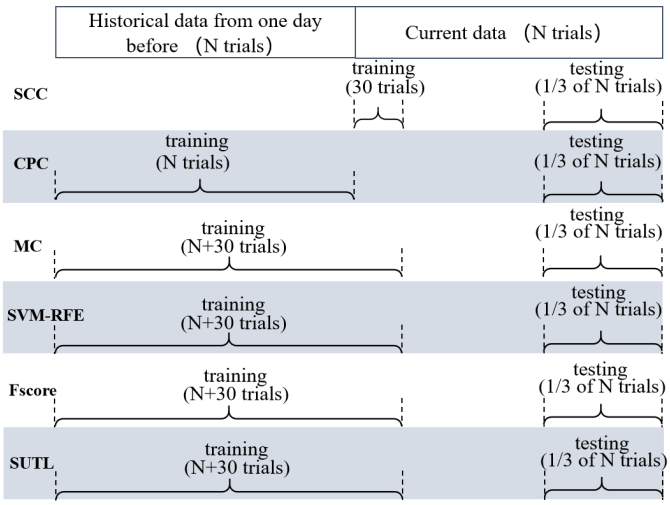

Fig. 3. The diagram of the composition of training data and testing data for the SCC, CPC, MC, SVM-RFE, Fscore and SUTL schemes. The historical data from one day before contained 630 and 360 trials in monkeys M and B, respectively. The same testing data were used in all these schemes.

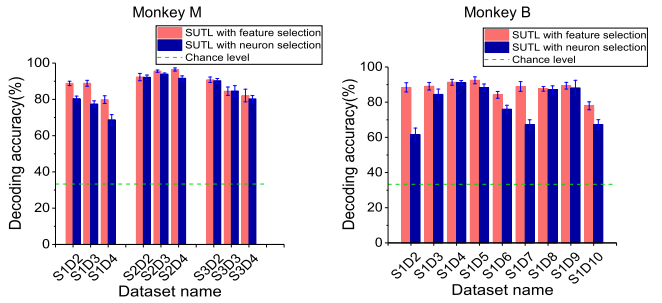

Fig. 4. Comparison of the performance of the SUTL method by feature selection and neuron selection. The pink and blue bar represent the mean classification accuracy of the decoders calibrated using SUTL method with feature selection and with neuron selection respectively. Error bars represent standard deviations of the 100 repeated decoding accuracies. Datasets were labeled by name of the session and day, for example, S1D1 meant the dataset was from the first day of session 1.

III. Results

A. Comparison Between Feature Selection and Neuron Selection in SUTL

Both the feature selection and neuron selection in the SUTL method were expected to exhibit good and robust decoding performance. To verify whether feature selection or neuron selection was more effective, they were implemented separately in the SUTL method, and the corresponding classification results for all the data sessions are shown in Fig. 4. For some datasets, the neuron selection achieved similar decoding performance as the feature selection. However, for other datasets, the neuron selection was significantly less effective, compared to the feature selection. The decoding performance of the neuron selection was not robust. In general, the feature selection achieved better and more robust performance than the neuron selection for both Monkeys (Monkey B: mean decoding accuracy was 88.8% versus 84.4%, with a standard deviation of 5.8 versus 8.4; Monkey H: mean decoding accuracy was 87.7% versus 79.0%, with a standard deviation of 4.3 versus 11.2). The following analyses were based on the feature selection.

B. Comparison of Decoding Performance Between SUTL and Other Schemes

To evaluate our SUTL method, we implemented three common calibration schemes (SCC, CPC and MC) and two feature selection algorithms (SVM-RFE and Fscore) for comparison. The decoding accuracies and standard deviations of these methods for each dataset are illustrated in Fig. 5. The decoding accuracies and standard deviations of these methods across all data sessions are shown in Table I. The SUTL method significantly outperformed the other five methods for both monkeys. For the SCC scheme, the available training data were too few to guarantee the robustness of the calibrated decoder. For the CPC and MC schemes, the historical data were directly used for the decoder calibration, and the disparities between the historical data and current data were not diminished. Therefore, they exhibited good decoding performance only when the disparities were small. The SVM-RFE and the Fscore did not consider the stationarity of the feature between the historical data and current data, which might affect the decoding performance. By comparison, the SUTL method effectively diminished the disparities between the historical data and current data by selecting the stationary features, thus, it could achieve good and robust decoding performance.

In the SUTL method, the proportion of the selected features in the process of considering the stationarity and the importance of the feature would affect the final number of selected features and the decoding performance. We subsequently explored the impact of this proportion on the performance of the SUTL method. The proportions of the selected features in considering the stationarity and the importance were gradually increased from 1/8 to all, with a step of 1/8 for Monkey M; and were increased from 1/6 to all, with a step of 1/6 for Monkey B. The corresponding classification accuracies for these different proportions are shown in Fig. 6.

In general, the results indicated that the decoding accuracy first increased with the increase in the proportion; subsequently, however, it later decreased with the increase in the proportion for both the stationarity and importance. In considering only the stationarity, the best mean decoding accuracy was achieved when the proportion for the stationarity was set to 1/2 of all the features for both monkeys, which can be seen from the mean accuracy of the column in Fig. 6. This may be because a lower proportion would result in the loss of some important features while a higher proportion might result in the inclusion of more non-stationary features. For considering only the importance, the best mean decoding accuracy could be achieved when the proportion for the importance was set to 1/2 of all the features for both monkeys, which can be seen from the mean accuracy of the row in Fig. 6. The decoding performance for the lower proportion would be more easily affected by the important but non-stationary features, and a higher proportion might result in the inclusion of more unimportant features. To obtain a compromise between the

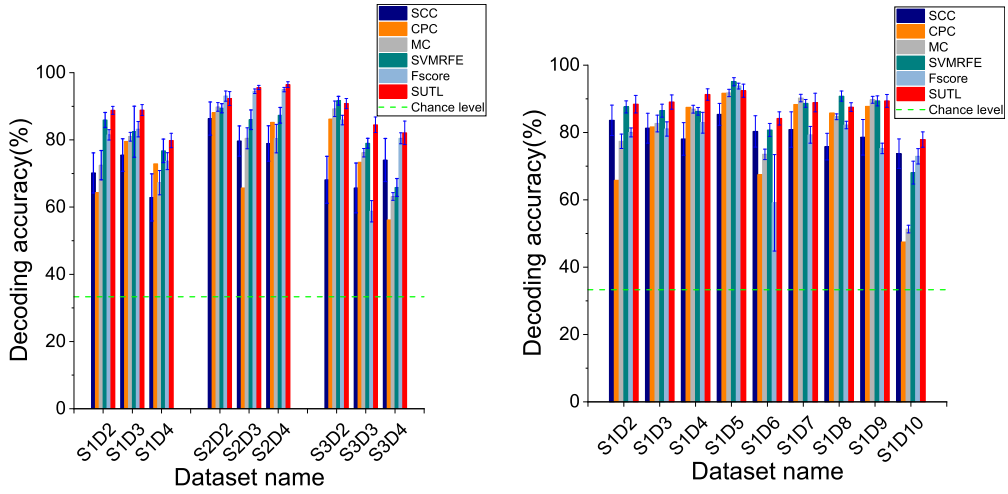

Fig. 5. Comparison of the performance of six decoder calibration schemes across all sessions in two monkeys. The decoding accuracy of each dataset was the mean value of 100 repeated computations, in which the current small sample set used for training were randomly selected from the current data. The dark blue, orange, grey, green, light blue and red bars represent the mean classification accuracy of the decoders calibrated using SCC, CPC, MC, SVM-RFE, Fsore and SUTL scheme, respectively. Error bars represent standard deviations of the 100 repeated decoding accuracies. The results of chance level were presented for reference. Datasets were labeled by name of the session and day, for example, S1D2 meant the testing dataset was from the second day of session 1.

TABLE I
THE MEAN DECODING ACCURACIES AND STANDARD DEVIATIONS FOR ALL DATA SESSIONS OF FOUR METHODS IN THE SIMULATED ONLINE TEST

Monkey M						
METHOD	SCC	CPC	MC	SVM-RFE	Fscore	SUTL
ACCURACY (%)	73.5±7.50	74.6±11.1	77.8±9.1	82.7±7.90	82.9±11.50	89.4±5.3
Monkey B						
METHOD	SCC	CPC	MC	SVM-RFE	Fscore	SUTL
ACCURACY (%)	79.7±3.6	78.2±15.2	81.2±12.9	85.9±7.7	78.5±9.3	87.7±4.3

stationarity and the importance, the proportion of 1/2 of all the features for both the stationarity and the importance would be a good choice. Therefore, in considering the stationarity and the importance, the top half of the features were selected.

C. Quantitative Assessment of Decoder Calibration Time Reduction in SUTL

To further evaluate the efficiency of the decoder calibration in the proposed SUTL method, the SCC scheme, which is the most commonly used method in decoder calibration for iBMIs, was used for comparison. The decoder calibration time mainly contained two parts, the time spent in collecting new current samples and the running time of the algorithm. For the time spent in collecting new current samples, it could be measured by the number of new current samples that were used in decoder calibration. Here, we used increased number of experimental trials from the current dataset to calibrate the decoder in the SCC scheme and identified how many current samples were required by the SCC scheme to achieve the same classification accuracy as the SUTL method employing only ten current samples of each category. As shown in Table II, the SCC methods required many more current samples to achieve the same decoding accuracy as the SUTL method in all the data sessions for both monkeys. Especially in some cases (11/18), the SCC method could not achieve the same decoding accuracy as the SUTL method, even with up to 140 current samples per category for both monkeys. In all the cases, the SCC scheme had to use more than 93 current samples per category to achieve the same decoding accuracy as the SUTL method, which only used ten current samples per category. Therefore, the SUTL method required approximately 89.2% (83/93) less current samples, compared with the SCC method.

The time complexity and running time were also compared between the SUTL and the SCC schemes. In the SUTL, the time complexity analysis mainly contained two parts, the time complexity of the feature selection algorithm and that of the SVM. For the feature selection algorithm, the time complexity was $O(n^2)$; for the SVM, the time complexity was $O(n^3)$, where n was the number of training samples [59]. Therefore, the time complexity of the SUTL was $O(n^3)$. There was no feature selection algorithm in the SCC, so the time complexity of the SCC method also was $O(n^3)$.

The running time of the SUTL and the SCC was compared under the condition that they achieved the same decoding accuracy(running on the same computer). The results of the running time are summarized in Fig. 7. The SUTL took less time than

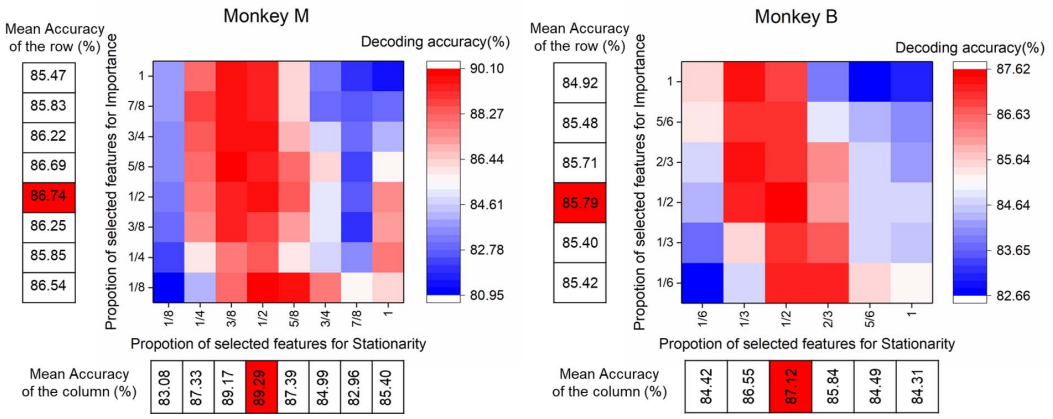

Fig. 6. The mean classification accuracy of the decoder calibrated using the SUTL method when the proportions of the selected features for stationarity and importance were increased gradually. The horizontal coordinate is the proportion of the selected features for stationarity, and the ordinate is the proportion of the selected features for importance. Each cell is the mean decoding accuracy across all data sessions for the corresponding selected proportion. Moreover, the mean decoding accuracy of the row and the column of the heat map are presented to find the optimal proportion for considering the importance and the stationarity of the feature separately. The optimal proportion was 1/2 for both the importance and the stationarity of the feature and was marked in red colour in the tale of the mean accuracy of the row or the column.

TABLE II
THE NUMBER OF CURRENT SAMPLES FOR SCC

Monkey M									
DATA NAME	S1D2	S1D3	S1D4	S2D2	S2D3	S2D4	S3D2	S3D3	S3D4
Number of trials	No	No	No	No	No	No	No	No	73
Monkey B									
DATA NAME	S1D2	S1D3	S1D4	S1D5	S1D6	S1D7	S1D8	S1D9	S1D10
Number of trials	24	57	No	No	23	33	No	77	27

D, day; S, session; No, could not. The number of current samples required for decoder calibration in the SCC method to achieve the same classification accuracy as the SUTL method (using only ten current trials for decoder calibration) across all sessions in two monkeys. "No" indicates that the classification accuracy obtained using the SUTL method could not be achieved by using the SCC method even with up to 140 current samples per category for monkey M and monkey B.

the SCC and the feature selection time in SUTL was only about 1s. The SCC took more time because it needed more samples and more features to train the SVM. The datasets in Monkey B took less time because they contained less historical samples than the datasets in Monkey M. Generally, collection of new current samples would always take more time (minutes to tens of minutes) than the running time of the algorithm. We have effectively reduced the demand for the number of current new samples in the SUTL and the total running time of less than 30s was acceptable.

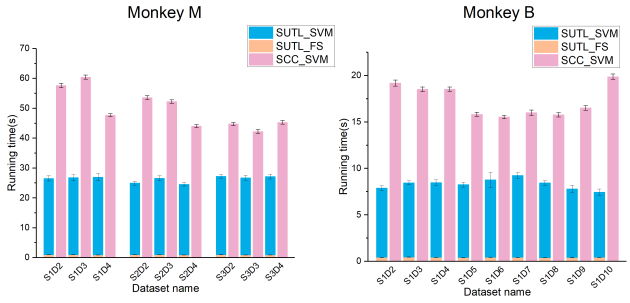

Fig. 7. The running time of the SUTL method across all sessions in two monkeys. The blue and orange bars represent the mean running time for training the SVM classifier and selecting features respectively in the SUTL method. The pink bar represent the mean running time for training the SVM classifier in the SCC. Error bars represent standard deviations of the 100 repeated running time.

D. Quantitative Assessment of Computational Burden Reduction in SUTL

The computational burden was mainly evaluated based on the number of features and corresponding number of channels recording neural signals during the decoding. In the SUTL method, the features were finally selected after three processing steps: stationarity selection, importance selection, and redundancy selection. In each step, some features were eliminated and only a small fraction of the full original features were selected. The finally selected features and corresponding channels in the SUTL method are shown in Table III. The mean value of the final number of selected features for all the sessions are 52 (Monkey M) and 42 (Monkey B), out of a total number of 480 and 396 original features in the data collected from the two monkeys. To include these selected features, the mean value of the final number of selected channels are 26 (Monkey M) and 19 (Monkey B), out of a total number

of 80 and 66 original working channels. For considering the number of selected features, the computational burden was respectively reduced by 89.2% and 89.4% for Monkey M and Monkey B. Fewer features can accelerate the decoder training speed. In considering the number of channels that need to record neural signals during the decoding, the computational burden was respectively reduced by 67.5% and 71.2% for Monkey M and Monkey B.

E. Assessment of Disparity Between Historical Data and Current Data After Applying SUTL

The SUTL method could achieve better decoding performance than the other five schemes; we attributed this to the fact that it could effectively diminish the disparities between the historical data and current data. To explore whether these disparities could be effectively diminished by the SUTL method, the disparities before and after applying the SUTL method were visualized and quantitatively measured. The current data and historical data from one day before were combined to construct datasets and were reduced to two dimensions using the principal component analysis for visualization. Taking Dataset 1 of Session 1 for Monkey M as an example, the results of dimensionality reduction are shown in Fig. 8a. Based on the results, the distribution of the historical data is clearly separated from the current data for the full ensemble data; however, the historical data and current data were well clustered after the SUTL method was applied. For quantitatively measuring the disparities, the DBI values of the datasets before and after applying the SUTL were measured and compared. The DBI value can quantitatively evaluate the clustering result of the data, which can indicate the sample distribution of the datasets (a lower DBI score denotes more distinct clusters) [60]. As shown in Fig. 8b, the data processed using the SUTL method achieved significantly lower DBI values than those that were processed without applying it for both monkeys. These results indicated that the disparities between historical data and current data have been effectively diminished, which is the key that the SUTL method could be effective. Moreover, the data from the same day were clustered together and this indicated that the above assumption was tenable, and that the data obtained on the same day were subject to the same distribution.

F. The Generalization Ability of the Selected Features in SUTL

The generalization ability of the selected features is important in the feature selection. However, it is hard to evaluate, especially in our SUTL algorithm. Because of the nonstationarity of the neural recordings, there are disparities between different datasets. When one wants to decode a new dataset, the SUTL has to be implemented again to select new features for constructing new classifier and the selected features for different dataset may be different. We tried to evaluate the generalization ability by two methods.

The first method evaluated the generalization ability by comparing the decoding accuracy of training data with that of testing data after finishing the feature selection. If the testing

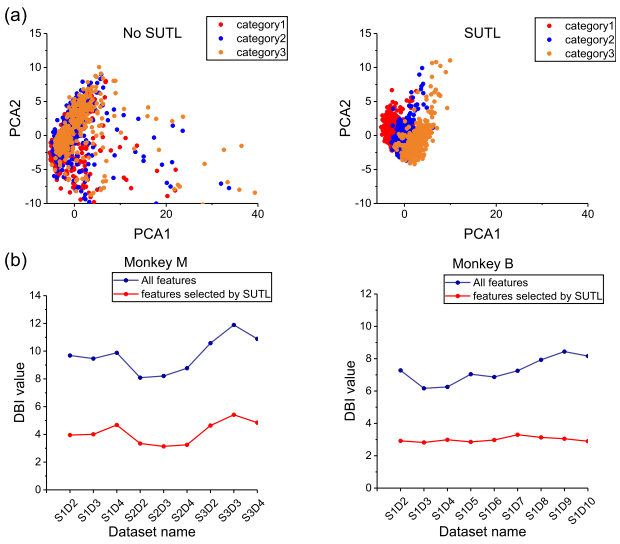

Fig. 8. The visualization and quantitative measurement of sample distribution before and after applying the SUTL method. (a) The visualization of the sample distribution for the Dataset 1 in Session 1 of monkey M before and after applying the SUTL method. (b) Each current data and the historical data from one day before were combined to construct datasets. The DBI values of each dataset with the full features (blue line) and the selected features by the SUTL method (red line) were presented. A lower DBI value indicated better cluster separation.

data could achieve similar decoding accuracy as the training data, the generalization ability could be considered good. The decoding accuracy of the training data was obtained by 5-fold cross validation. The results of the comparison are shown in Fig. 9a. There were about 2.9% and 5.1% difference in mean decoding accuracy between the two methods for Monkey M and B respectively. We thought that the difference in the decoding accuracy was small and the generalization ability was good.

The second method used the Dice-Sorensens index (DSI) to measure the overlap of the selected features between two different datasets, which could evaluate the generalization ability or the stability of the selected features [61]. The results of the DSI for both monkeys are shown in Fig. 9b. It can be seen that the DSI is approximately 0.6 between datasets from two consecutive days, which indicated that about 60 percent of the selected features were the same for the two datasets. Considering the disparities between two datasets, we thought that the generalization ability of the selected features was good.

G. Properties of Features Selected by SUTL

The above analyses have proven that the SUTL method could be effective. Analysing the properties of the features selected by the SUTL may provide some insights for future endeavours to improve decoding performance. We counted the number of selected features belonging to each brain region and to each bin of the decoding time period. In this study, the neuron signals were effectively recorded from the M1 (192 features), S1 (192 features), and PPC (96 features) of Monkey M and the S1_a (186 features), S1_b (42 features), and PPC (168 features) of Monkey B. For each of these

TABLE III
THE FINAL NUMBER OF FEATURES AND CORRESPONDING CHANNELS

Monkey M									
DATA NAME	S1D2	S1D3	S1D4	S2D2	S2D3	S2D4	S3D2	S3D3	S3D4
Number of features	52	52	56	48	52	48	54	50	54
Number of channels	26	25	29	24	25	23	27	29	30
Monkey B									
DATA NAME	S1D2	S1D3	S1D4	S1D5	S1D6	S1D7	S1D8	S1D9	S1D10
Number of features	38	44	44	42	45	50	44	38	36
Number of channels	19	17	20	20	22	20	20	19	16

The final number of features and corresponding channels selected by the SUTL method across all sessions in two monkeys.

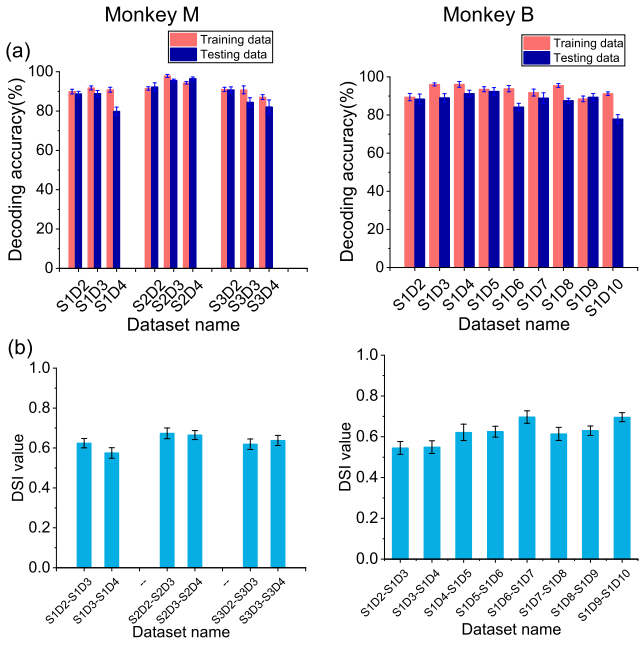

Fig. 9. The evaluation of the generalization ability of the selected features in SUTL. (a) The comparison of decoding accuracy between the training and testing data. The pink and blue bar represent the mean classification accuracy in the training and testing data, respectively. Error bars represent standard deviations of the 100 repeated decoding accuracies. (b) The Dice-Sorensens index(DSI) of selected features between datasets from two consecutive days in two monkeys.

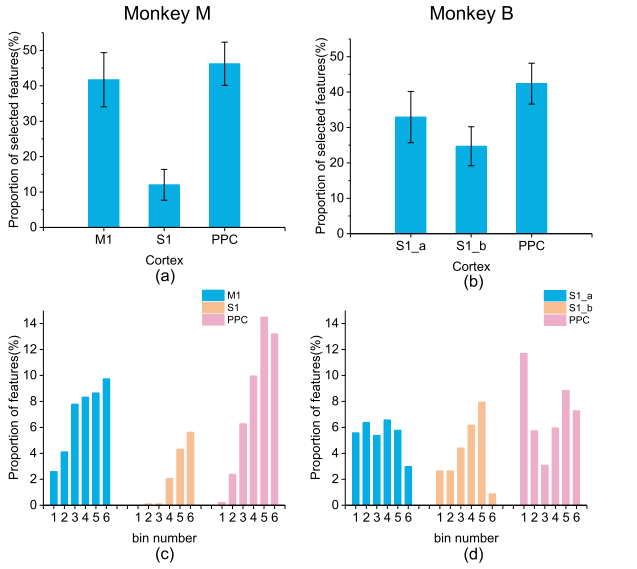

Fig. 10. The distribution of the selected features in three brain regions for (a) monkey M and (b) monkey B, and the distribution of the selected features in different bins of the decoding time period for (c) monkey M and (d) monkey B.

brain regions, the proportion of the final selected features was computed as (the number of final selected features) / (the number of all features), which was then normalized among these brain regions. A similar approach was used to count the number of selected features belonging to each bin of the decoding time period. The results are shown in Fig. 10.

For Monkey M, the proportion of the final selected features was the smallest in the S1, whereas those in the M1 and the PPC were comparable. For Monkey B, the proportions of the final selected features from the three brain regions were similar. This indicated that the M1 and PPC might play more important role than the S1 during the reaching movement, whereas both the S1 and PPC played important roles during the grasping gesture decoding. Fig. 10(c) and (d) show the distribution of the final selected features in each bin of the decoding time period. In Monkey M, the proportion of the final selected features increased over time in all the three brain regions. This indicated that more effective features were activated after the reaching movement was executed, and that these brain regions might be related more to the reaching movement execution than they were to the planning. In Monkey B, the proportion of the final selected features first increased, and then decreased in the S1. On the contrary, the proportion of the final selected features decreased first, and then increased in the PPC.

IV. DISCUSSION

This study incorporated TL into the iBMIs to diminish the disparities between the historical data and current data. Subsequently, the historical data could be effectively used for decoder calibration, and the demand for new current training data could be reduced while better decoding performance could be achieved. Moreover, by selecting the features that are most relevant to classification, and removing redundant

features, the number of features for decoding and the number of channels that had to work were reduced, as a result of which the computational burden and energy consumption reduced.

The nonstationarity of the neural recordings is common in iBMIs and leads to daily retraining of the decoder [19], [20]. TL is particularly suitable for solving this problem, because the essence of the nonstationarity in iBMIs is to make the sample distribution of source domain and target domain distinct from each other. TL can eliminate this difference in sample distribution. Moreover, besides the disparity, there are also many similarities between the source domain and the target domain. Currently, electrode arrays (such as the Utah array and the FMA array) are widely used as cortex implants in iBMIs. These arrays are floated on the cortex and can record neural signals relatively stably [14]. Although there will be some nonstationarities for various reasons [19], the data from two adjacent days are similar, and the properties of many features between these two days are stationary. The SUTL works by diminishing the disparities and preserving similarities.

In most of the previous studies on iBMIs [15], [16], neuron selection was implemented to reduce computational burden while maintaining the decoding performance. In neuron selection, each neuron contained several features and these features were selected as a whole [24], [25]. Advances in iBMIs have made it possible to implant more electrodes into different brain regions [62], such as the M1, S1 and PPC, as was done in this study. Neural signals from different brain regions were recorded synchronously, and the same time window was used for all neurons to extract features for the decoding. However, the neurons from different brain regions might have different functions [63]; hence, using the same time window in the feature extraction might not be optimal for all the neurons. For neuron selection, one neuron could be selected, however, not all the features in this neuron were suitable for decoding,which could be verified by the results in Fig. 10, and the unrelated features could not be eliminated. Feature selection could avoid such problems. In this study, both neuron selection and feature selection have been tested in the SUTL method. The results indicated that feature selection exhibited better decoding performance than neuron selection in the SUTL method. Hence, feature selection might be a more suitable choice than neuron selection for TL in iBMIs.

In the stationarity and importance evaluation, a threshold was needed to select the top J ranked features. The threshold was empirically chosen as the top half ranked features in this study and the reason for choosing this threshold was described below. The electrode array used in this study could relatively stably record neural signals. Even though the disparities between datasets were unavoidable, there were more similarities than the disparities, which could be verified by the results in Fig.9b. Therefore, selecting the top half ranked features in the stationarity evaluation was reasonable. In the iBMIs, dozens or even hundreds of neurons were recorded to decode only three or more types of movements. Therefore, it is common that more than 50% of the neurons were redundant, which could also be seen from other related publications [15], [16], [33]. Therefore, selecting the top half ranked features in the importance evaluation also was reasonable. Moreover, in addition to selecting the fixed threshold to 50% in stationarity and importance evaluation, the redundancy of these selected features was further considered to ensure that the redundant features could be effectively removed in the SUTL. Finally, in the online application of the iBMIs, real-time performance is important. Using the fixed percentage threshold is very efficient and can be implemented in real time.

By taking advantage of the historical data, the efficiency of the decoder calibration has been improved by the SUTL method. The reduction in the computational burden is mainly reflected in the reduction of the feature number and the number of channels that are required to work. In the online decoding, the reduction of the feature number will not have much impact on the overall computational burden; its main purpose is to accelerate the training of the decoder in the decoder calibration phase. The computational burden is mainly affected by the number of channels that need to record neural signals. For example, without the SUTL method, 80 channels are required to record signals; however, only 20 channels are required after applying the SUTL method. Then, in terms of the hardware, we can temporarily turn off the data collection function of these 60 channels, which will greatly reduce the overall computational burden and energy consumption, which is of great significance for the future wearable iBMIs.

At present, electrode arrays are mainly implemented in the M1 cortex for many motor iBMIs [1]. This can achieve good decoding performance, and will reduce the difficulty and risk associated with implantation into multiple brain regions. In this study, electrode arrays were implanted into multiple cortices. Through the analysis of the selected features, it was found that both the M1 and the PPC cortices were important in decoding the reaching position, which is also consistent with the conclusions of other related studies [6], [64]. Electrode arrays implanted into multiple cortices, rather than one cortex, might achieve better and more robust performance. The electrode array was also implanted into the S1 cortex in this study, and we found that the S1 played a much smaller role than the M1 and PPC during the reaching position decoding. This might be related to the decoding time period we chose and the physiological properties of the S1 cortex. However, in motor iBMIs, S1 is not a preferred cortex. Because the iBMIs are mainly used by the paralyzed patients, who control the prosthesis instead of their real arms, the implication of which is that S1 cortex cannot be directly activated in paralyzed patients.

The SUTL method evaluates the stationarity of the features by measuring the correlation between the same feature in the source domain and target domain. There is a shortage in this method. Normally, the samples from the target domain used to measure the stationarity are relatively few (such as ten trials for each category in this study), as a result of which the stationarity measurement might be significantly affected by these small samples. In other words, if there is a big difference between these small samples and the subsequent new data, it is likely that the decoding performance will be reduced. Therefore, the premise of our study is that data obtained on the same day are subject to the same distribution, and the

small number of current samples used for calibration needs to be able to represent the distribution of the subsequent new samples. Because of the stability of the electrode array in neural recording, this assumption is still feasible and our decoding results are also consistent with this assumption.

V. Conclusion

In this study, we proposed a novel TL method, SUTL, to improve the decoder calibration and decoding performance of iBMIs. The method could effectively use the historical data and only required a small current sample set to achieve good decoding accuracy within a shorter decoder calibration time and with less computational burden. Moreover, we demonstrated that the feature selection can achieve better performance than the neuron selection in our SUTL method. With the reduction in the decoder calibration time and computational burden while maintaining the decoder performance, our proposed method has the potential to promote the development of iBMIs in clinical research.

References

[1] M. J. Vansteensel *et al.*, "Fully implanted brain–computer interface in a locked-in patient with ALS," *New England J. Med.*, vol. 375, no. 21, pp. 2060–2066, Nov. 2016.

[2] L. R. Hochberg *et al.*, "Neuronal ensemble control of prosthetic devices by a human with tetraplegia," *Nature*, vol. 442, no. 7099, pp. 164–171, Jul. 2006.

[3] J. L. Collinger *et al.*, "High-performance neuroprosthetic control by an individual with tetraplegia," *Lancet*, vol. 381, no. 9866, pp. 557–564, Feb. 2013.

[4] D. M. Brandman, S. S. Cash, and L. R. Hochberg, "Review: Human intracortical recording and neural decoding for brain–computer interfaces," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 10, pp. 1687–1696, Oct. 2017.

[5] B. R. Townsend, E. Subasi, and H. Scherberger, "Grasp movement decoding from premotor and parietal cortex," *J. Neurosci.*, vol. 31, no. 40, pp. 14386–14398, Oct. 2011.

[6] T. Aflalo *et al.*, "Decoding motor imagery from the posterior parietal cortex of a tetraplegic human," *Science*, vol. 348, no. 6237, pp. 906–910, May 2015.

[7] M. Velliste, S. Perel, M. C. Spalding, A. S. Whitford, and A. B. Schwartz, "Cortical control of a prosthetic arm for self-feeding," *Nature*, vol. 453, no. 7198, pp. 1098–1101, Jun. 2008.

[8] M. Clerc, "Brain computer interfaces, principles and practise," *Biomed. Eng. Online*, vol. 12, no. 22, pp. 2012–2424, 2013.

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

[10] V. Gilja *et al.*, "Clinical translation of a high-performance neural prosthesis," *Nature Med.*, vol. 21, no. 10, pp. 1142–1145, Oct. 2015.

[11] V. Gilja, C. A. Chestek, I. Diester, J. M. Henderson, K. Deisseroth, and K. V. Shenoy, "Challenges and opportunities for next-generation intracortically based neural prostheses," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 7, pp. 1891–1899, Jul. 2011.

[12] S. N. Abdulkader, A. Atia, and M.-S. M. Mostafa, "Brain computer interfacing: Applications and challenges," *Egyptian Inform. J.*, vol. 16, pp. 213–230, Jul. 2015.

[13] S. Guan *et al.*, "Elastocapillary self-assembled neurotassels for stable neural activity recordings," *Sci. Adv.*, vol. 5, no. 3, Mar. 2019, Art. no. eaav2842.

[14] M. L. Homer, A. V. Nurmikko, J. P. Donoghue, and L. R. Hochberg, "Sensors and decoding for intracortical brain computer interfaces," *Annu. Rev. Biomed. Eng.*, vol. 15, no. 1, pp. 383–405, Jul. 2013.

[15] J. C. Sanchez, J. M. Carmena, M. A. Lebedev, M. A. L. Nicolelis, J. G. Harris, and J. C. Principe, "Ascertaining the importance of neurons to develop better brain-machine interfaces," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 943–953, Jun. 2004.

[16] K. Xu *et al.*, "Local-learning-based neuron selection for grasping gesture prediction in motor brain machine interfaces," *J. Neural Eng.*, vol. 10, no. 2, Apr. 2013, Art. no. 026008.

[17] G. Santhanam, S. I. Ryu, B. M. Yu, A. Afshar, and K. V. Shenoy, "A high-performance brain–computer interface," *Nature*, vol. 442, no. 7099, pp. 195–198, 2006.

[18] P. Zhang *et al.*, "Decoder calibration with ultra small current sample set for intracortical brain–machine interface," *J. Neural Eng.*, vol. 15, no. 2, Apr. 2018, Art. no. 026019.

[19] N. Paul, J. C. Kao, J. M. Fan, S. D. Stavisky, S. I. Ryu, and K. V. Shenoy, "Performance sustaining intracortical neural prostheses," *J. Neural Eng.*, vol. 11, no. 6, p. 66003, 2014.

[20] C. A. Chestek *et al.*, "Long-term stability of neural prosthetic control signals from silicon cortical arrays in rhesus macaque motor cortex," *J. Neural Eng.*, vol. 8, no. 4, p. 45005, 2011.

[21] F. R. Willett *et al.*, "Principled BCI decoder design and parameter selection using a feedback control model," *Sci. Rep.*, vol. 9, no. 1, Dec. 2019, Art. no. 8881.

[22] J. A. Perge *et al.*, "Intra-day signal instabilities affect decoding performance in an intracortical neural interface system," *J. Neural Eng.*, vol. 10, no. 3, p. 36004, 2013.

[23] G. Santhanam *et al.*, "HermesB: A continuous neural recording system for freely behaving primates," *IEEE Trans. Biomed. Eng.*, vol. 54, no. 11, pp. 2037–2050, Nov. 2007.

[24] G. Singhal, V. Aggarwal, S. Acharya, J. Aguayo, J. He, and N. Thakor, "Ensemble fractional sensitivity: A quantitative approach to neuron selection for decoding motor tasks," *Comput. Intell. Neurosci.*, vol. 2010, no. 5, p. 6, 2014.

[25] H. J. Hwang, J. M. Hahne, and K. R. Müller, "Channel selection for simultaneous and proportional myoelectric prosthesis control of multiple degrees-of-freedom," *J. Neural Eng.*, vol. 11, no. 5, p. 56008, 2014.

[26] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, nos. 7–8, pp. 1157–1182, Jan. 2003.

[27] V. Kumar, "Feature selection: A literature review," *Smart Comput. Rev.*, vol. 4, no. 3, pp. 211–229, Jun. 2014.

[28] J. Tang, S. Alelyani, and H. Liu, "Feature selection for classification: A review," in *Data Classification: Algorithms and Applications*, C. C. Aggarwal, Ed. Boca Raton, FL, USA: CRC Press, 2013.

[29] M. Robnik-Sikonja and I. Kononenko, "Theoretical and empirical analysis of reliefF and RReliefF," *Mach. Learn.*, vol. 53, pp. 23–69, Oct. 2003.

[30] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd ed. New York, NY, USA: Wiley, 2001, pp. 37–64.

[31] 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.

[32] I. Koprinska, "Feature selection for brain–computer interfaces," in *Proc. Int. Workshop New Frontiers Appl. Data Mining (PAKDD)*, in Lecture Notes in Computer Science, vol. 5669, 2009, pp. 106–117.

[33] R. Wahnoun, J. He, and S. I. H. Tillery, "Selection and parameterization of cortical neurons for neuroprosthetic control," *J. Neural Eng.*, vol. 3, no. 2, p. 162, 2006.

[34] Z. Li, "Decoding methods for neural prostheses: Where have we reached?" *Frontiers Syst. Neurosci.*, vol. 8, no. 8, p. 129, Jul. 2014.

[35] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[36] P. Wang, J. Lu, B. Zhang, and Z. Tang, "A review on transfer learning for brain–computer interface classification," in *Proc. 5th Int. Conf. Inf. Sci. Technol. (ICIST)*, Apr. 2015, pp. 315–322.

[37] D. Wu, "Online and offline domain adaptation for reducing BCI calibration effort," *IEEE Trans. Human-Machine Syst.*, vol. 47, no. 4, pp. 550–563, Aug. 2017.

[38] H. Cho, M. Ahn, K. Kim, and S. C. Jun, "Increasing session-to-session transfer in a brain computer interface with on-site background noise acquisition," *J. Neural Eng.*, vol. 12, no. 6, p. 66009, 2015.

[39] V. Jayaram, M. Alamgir, Y. Altun, B. Scholkopf, and M. Grosse-Wentrup, "Transfer learning in brain–computer interfaces," *IEEE Comput. Intell. Mag.*, vol. 11, no. 1, pp. 20–31, Feb. 2016.

[40] R. Zink, B. Hunyadi, S. V. Huffel, and M. D. Vos, "Tensor-based classification of an auditory mobile BCI without a subject-specific calibration phase," *J. Neural Eng.*, vol. 13, no. 2, p. 26005, 2016.

[41] A. Farshchian, J. A. Gallego, J. P. Cohen, Y. Bengio, and S. A. Solla, "Adversarial domain adaptation for stable brain–machine interfaces," in *Proc. 7th Int. Conf. Learn. Represent.*, 2019, pp. 1–14.

[42] P. Zhang, X. Ma, H. Huang, and J. He, "Predicting hand orientation in reach-to-grasp tasks using neural activities from primary motor cortex," in *Proc. 36th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2014, pp. 1306–1309.

[43] J. He, X. Ma, and J. He, "A neurobehavioral device to study the neural mechanism in reach to grasp task," in *Proc. IEEE Int. Conf. Mechatronics Automat.*, Aug. 2012, pp. 2146–2151.

[44] B. P. Christie *et al.*, "Comparison of spike sorting and thresholding of voltage waveforms for intracortical brain–machine interface performance," *J. Neural Eng.*, vol. 12, no. 1, p. 16009, 2014.

[45] J. Dai *et al.*, "Reliability of motor and sensory neural decoding by threshold crossings for intracortical brain–machine interface," *J. Neural Eng.*, vol. 16, no. 3, p. 36011, 2019.

[46] M. A. Hall, "Correlation-based feature selection for discrete and numeric class machine learning," in *Proc. 17th Int. Conf. Mach. Learn.*, 2000, pp. 359–366.

[47] J. Li, W. Chen, T. Zhao, S. Zhang, and X. Zheng, "Symmetrical uncertainty based neural selection in motor cortex brain–machine interfaces," in *Proc. World Congr. Med. Phys. Biomed. Eng.*, vol. 39, 2012, pp. 1545–1548.

[48] B. Singh, N. Kushwaha, and O. P. Vyas, "A feature subset selection technique for high dimensional data using symmetric uncertainty," *J. Data Anal. Inf. Process.*, vol. 2, no. 4, pp. 95–105, 2014.

[49] H. M. Al-Angari, G. Kanitz, S. Tarantino, and C. Cipriani, "Distance and mutual information methods for EMG feature and channel subset selection for classification of hand movements," *Biomed. Signal Process. Control*, vol. 27, pp. 24–31, May 2016.

[50] J. Dai, Z. He, and F. Hu, "A high performance algorithm for text feature automatic selection," in *Proc. Int. Symp. Inf. Process. (ISIP)*, 2009, pp. 1–6.

[51] P. Pramokchon and P. Piamsa-Nga, "Effective threshold estimation for filter-based feature selection," in *Proc. Int. Comput. Sci. Eng. Conf. (ICSEC)*, Dec. 2016, pp. 1–6.

[52] V. Bolón-Canedo, N. Sánchez-Maroo, and A. Alonso-Betanzos, "A review of feature selection methods on synthetic data," *Knowl. Inf. Syst.*, vol. 34, no. 3, pp. 906–910, 2013.

[53] B. Seijo-Pardo, V. Bolón-Canedo, and A. Alonso-Betanzos, "Using data complexity measures for thresholding in feature selection rankers," in *Proc. Conf. Spanish Assoc. Artif. Intell.*, 2016, pp. 121–131.

[54] B. Seijo-Pardo, V. Bolón-Canedo, and A. Alonso-Betanzos, "On developing an automatic threshold applied to feature selection ensembles," *Inf. Fusion*, vol. 45, pp. 227–245, Jan. 2019.

[55] Y. W. Chen and C. J. Lin, "Combining SVMs with various feature selection strategies," in *Feature Extraction* (Studies in Fuzziness and Soft Computing), vol. 207. 2008, pp. 315–324.

[56] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, Apr. 2011.

[57] W. Bishop *et al.*, "Self-recalibrating classifiers for intracortical brain–computer interfaces," *J. Neural Eng.*, vol. 11, no. 2, p. 26001, 2014.

[58] K. Yan and D. Zhang, "Feature selection and analysis on correlated gas sensor data with recursive feature elimination," *Sens. Actuators B, Chem.*, vol. 212, pp. 353–363, Jun. 2015.

[59] A. Abdiansah and R. Wardoyo, "Time complexity analysis of support vector machines (SVM) in LibSVM," *Int. J. Comput. Appl.*, vol. 128, no. 3, pp. 28–34, Oct. 2015.

[60] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979.

[61] U. M. Khaire and R. Dhanalakshmi, "Stability of feature selection algorithm: A review," *J. King Saud Univ.-Comput. Inf. Sci.*, Jun. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1319157819304379

[62] J. J. Jun *et al.*, "Fully integrated silicon probes for high-density recording of neural activity," *Nature*, vol. 551, no. 7679, pp. 232–236, Nov. 2017.

[63] S. B. Eickhoff *et al.*, "Assignment of functional activations to probabilistic cytoarchitectonic areas revisited," *NeuroImage*, vol. 36, no. 3, pp. 511–521, Jul. 2007.

[64] H. Cui, "Forward prediction in the posterior parietal cortex and dynamic brain–machine interface," *Frontiers Integrative Neurosci.*, vol. 10, p. 35, Oct. 2016.