論文 / 著書情報
Article / Book Information

題目(和文)	
Title(English)	Improved Robustness for Brain Activity Decoding Based on Information Theoretic Learning
著者(和文)	LIYuanhao
Author(English)	Yuanhao Li
出典(和文)	学位:博士(学術), 学位授与機関:東京工業大学, 報告番号:甲第12603号, 授与年月日:2023年9月22日, 学位の種別:課程博士, 審査員:小池 康晴,吉村 奈津江,金子 寛彦,SLAVAKIS KONSTANTINO,八木 透,小尾 高史
Citation(English)	Degree:Doctor (Academic), Conferring organization: Tokyo Institute of Technology, Report number:甲第12603号, Conferred date:2023/9/22, Degree Type:Course doctor, Examiner:,,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Improved Robustness for Brain Activity Decoding Based on Information Theoretic Learning

LI Yuanhao

Major in Human Centered Science and Biomedical Engineering
Department of Information and Communications Engineering
School of Engineering

This thesis is submitted for the degree of
Doctor of Philosophy

August 2023

東京工業大学
Tokyo Institute of Technology

Doctoral Thesis

Improved Robustness for Brain Activity Decoding Based on Information Theoretic Learning

LI Yuanhao [1,†]

[1] Major in Human Centered Science and Biomedical Engineering
Department of Information and Communications Engineering
School of Engineering
† Supervisors: KOIKE Yasuharu & YOSHIMURA Natsue

Abstract: Brain activity decoding aims to predict the intentions or mental states of the brain by the utilization of brain recording signals. One significant obstacle for brain activity decoding is the brain recording noise that has complicated distribution, which may lead to large performance deterioration for existing brain decoding algorithms. To address this problem, the present thesis aims to propose robust brain activity decoding algorithms under the framework of information theoretic learning to alleviate the negative effects of the adverse brain recording noise. In particular, minimum error entropy criterion and maximum correntropy criterion were utilized to build robust objective functions for brain decoding algorithms. In addition, another significant problem for brain activity decoding, the high-dimensional issue, is also taken into account in this thesis. The proposed algorithms were evaluated systematically with synthetic datasets and real-world brain data. The experimental results demonstrated that information theoretic learning based robust brain decoding algorithms effectively reduce the performance deterioration caused by the noise and realize higher brain decoding accuracy on the real-world noisy brain data.

Keywords: Information theoretic learning; Neural decoding; Robustness; Maximum correntropy criterion; Minimum error entropy criterion; Rényi's quadratic entropy.

Contents

1. Introduction

Humans have been interested in how their brains work and interact with the external world for a long time. In particular, the mysterious phrase "mind reading" is full of fascination for human beings. The brain is the most complex organ in a human body, where the cerebral cortex is the most developed region which dominates all activity processes in the organism and regulates the balance between the organism and the surrounding environment, undertaking the basis for neural activities of higher level [1]. In the past, the investigations regrading the brain have been primarily explored from a medical or anatomical perspective. The first written record of the human brain dates back to Egypt, 4000 years ago. However, the prevailing view then was "heart-centered", that the heart was the source of mental activity, not the brain. It was since the 5th century that the brain was considered as the most important organ. With the lifting of taboos on dissecting human body, the invention of microscope, the creation of comparative anatomy and cranial phrenology, scientists were able to produce a fascinating array of research on the brain. In the 17th century, Thomas Willis rejected the view of "ventricle" and proposed that the higher cognitive functions of the human brain come from the folds of cerebral cortex, instead of the smoother region [2]. In the 1950s and 1960s, Roger Sperry found that the hemispheres in human brains play different cognitive functions, where the left hemisphere can interpret language but not the right hemisphere, by severing the corpus callosum on cats, monkeys, and humans [3]. Investigations on the brain were carried out from the chemical and cytological perspectives as well. For example, the Alzheimer's disease, one frequent neuro-degenerative disease which damages patients' cognition and memory, is commonly thought to be caused by the imbalance in the production and clearance of amyloid-β protein [4]. Studying how the numerous neurons in the brain are connected with each other and the mechanism of synaptic connections can also help to understand the brain [5,6].

With the advent of the information technology era, digital devices such as electroencephalogram (EEG) [7], magnetoencephalography (MEG) [8], electrocorticography (ECoG) [9], as well as functional magnetic resonance imaging (fMRI) [10], were invented with the capability to measure and record the brain activities through different physical quantities, which enabled the digitized description for brain activity. After the physiological activities in the cerebral cortex are converted into numerical digits and stored in the computer, one can utilize mathematical approaches to analyze and process brain activity recording data. In particular, with the investigations of artificial intelligence technology in the recent decades, the machine learning techniques have been more and more widely employed to analyze and process brain activity data [11,12]. Until now, machine learning based cognitive neuroscience has been witnessing the fast growths in the size and complexity of human brain data and the computational methods that allow scientists to study the brain mechanism under more naturalistic conditions [13,14]. The currently popular tools for "data-driven" neuroscience can be classifies into two prevailing forms: encoding and decoding. Encoding models aim to simulate and predict the brain activities under the awareness of external stimulus or spontaneous intention. By comparison, decoding refers to disclosing the received stimulus or spontaneous intention from recorded brain activities of different measurement modalities. A schematic diagram for encoding and decoding is illustrated in Fig. 1.

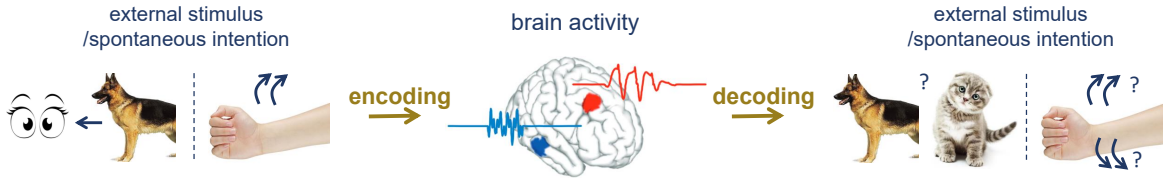

Figure 1. Schematic diagram of brain activity encoding and decoding. Encoding refers to simulating the brain activities with the input of stimulus or spontaneous intention, while decoding aims to expose the received stimulus or intention from the recorded brain activities.

Both encoding and decoding models aim to relate stimulus or mental status with brain signals, while decoding models exhibit an extra potential for the applications which aim to predict the inherent

cognitive activities or utilize neural activity to control external device [13]. More interestingly, brain activity decoding can be considered to be a more scientific name for "mind reading" [15]. Hence, this thesis mainly focuses on the brain activity decoding task. Brain activity decoding frameworks can be classified into two categories according to the property of target variable. The first framework is classification, where the neural activity corresponds to one of a finite set of possible event types, such as to move the left hand or the right hand. Many conventional machine learning algorithms have been employed to structure a classification model to classify the category of neural activities [16,17]. Motor imagery is a canonical paradigm for neural activity classification, where the subject imagines a virtual movement and the classification models is trained to identify the imaginary movement direction or limb [18]. Speech is another popular research topic in brain activity decoding. Discrete speech features have been successfully predicted, including vowels [19], phonemes [20], words [21], and sentences [22]. In addition, classification-based visual decoding can build the relationship between brain activity and the predefined labels for visual stimulus [23–25]. The second framework is reconstruction, which could be also called regression in a machine learning context, where continuous variables are restored from brain recording signals. For example, the parameters of upper limb movements were successfully reconstructed to control a robotic arm [26]. Further, continuous movement trajectories of an upper limb in the three-dimensional space could be directly predicted from brain activities [27,28]. Continuous features for speech reconstruction could be also accurately identified, such as amplitude power and spatiotemporal modulations [29,30], mel-frequency cepstral coefficients [31], and speech envelope [32]. Furthermore, visual reconstruction from brain activity has been receiving a growing attention in the research community [33–38].

The brain-computer interface (BCI) technology is another valuable research topic which is closely related to brain activity decoding. BCI refers to translating the brain signals to proper commands and using external device to help the communication with outside without muscle operation for paralyzed people [39]. BCI technology can be traced back to [40], in which monkeys were trained to modulate their neural activity rates above a threshold to be rewarded. This work demonstrated that the animal could interact with their own neural activities through causal links. This pioneering study and the corresponding technological innovations inspired the first invasive BCI system which was successfully implemented on rats [41]. Fig. 2 illustrates the experiment, where the rats performed a motion task with their neural activities being recorded which were used to replace the movements by controlling the robotic arm. It was found that neural activities were no longer associated with actual movements gradually, which means the rats did not need to move the limbs to generate neural signals. This shows that neurons exhibit powerful plasticity in neural coding and provide theoretical supports for later developments of BCI.

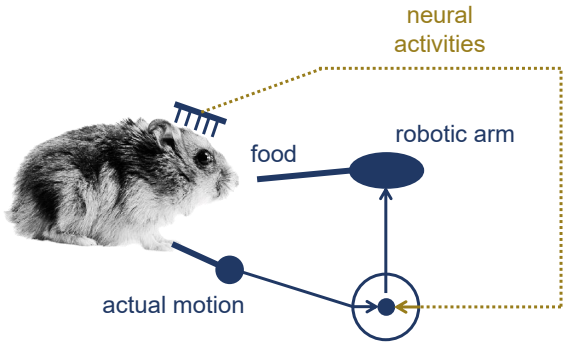

Figure 2. Experiment paradigm of the first invasive BCI system [41]. The rats were found to be able to regulate their neural activities to directly control the robotic arm without actual movements on their limbs.

Over the past twenty years many researchers have evaluated the possibility of realizing auxiliary communication technologies which are not dependent on muscle movements with different sensors

and brain activity types [42–44]. These BCI systems measure brain activities, extract their features, and translate them into target instructions to control external devices. Currently, BCI has been a promising approach to connect human brains and external world directly. A common BCI system is illustrated in Fig. 3, which consists of three central procedures: *(1) Signal Recording*: For different scenarios, choose proper recording modality and design user-friendly experiment, such that the BCI user could focus on performing specific brain activities. *(2) Brain Activity Decoding*: Reading one's intention through brain signals is critical for a BCI system. Machine learning can automatically extract effective features from brain signals, and recognize the pattern of extracted features, e.g. to move the left hand or the right hand. *(3) Control & Feedback*: Based on the user's intention, send appropriate control instructions to the external device to assist the user, and the user receives the feedback and performs the subsequent intentions.

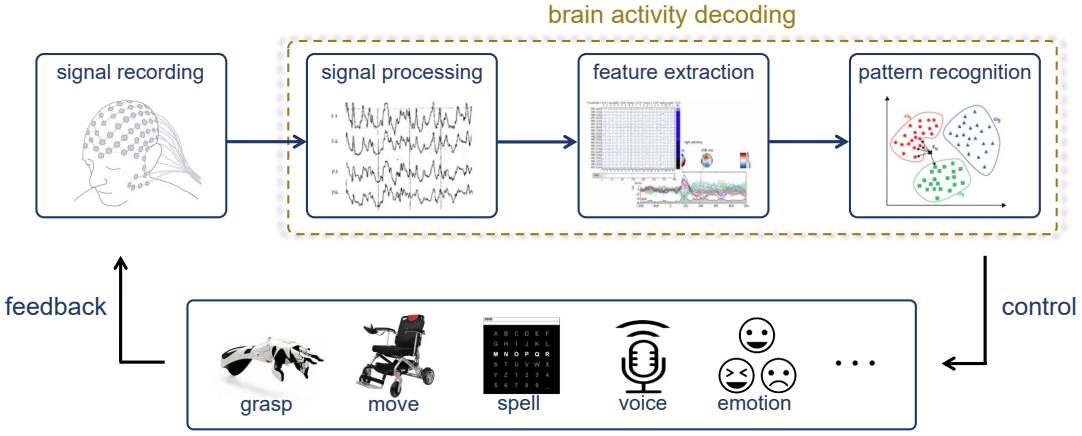

Figure 3. Schematic diagram of the common BCI system. Brain activities are recorded by the proper measurements for specific scenarios. Brain activity decoding translates the brain data to generate an appropriate command for the external device. The user receives an assistance from the external device.

With the development of more advanced brain activity decoding algorithms and experimental paradigms, BCI technology has been revealing increasing prospects under miscellaneous assistance scenarios. The canonical motor imagery paradigm has been widely utilized for controlling a wheelchair [45]. In addition to auxiliary movement, BCI can help re-establish channels of communication with the outside world for the people who lost the ability of speaking. For example, a recent study proposed to decode the imagined handwriting movements from neural activities in the motor cortex and translate them into text in real time, by which the paralyzed subject achieved the typing speed of 90 characters per minute with 94.1% online accuracy [46]. The capability of BCI in repairing or reproducing sensory-motor functions has been also demonstrated by recent scientific and technological advances [47–49].

Despite the developments of brain activity decoding techniques and their promising applications in BCI systems, the performance of brain activity decoding is potentially deteriorated by miscellaneous factors [12,50,51]. One significant obstacle for brain activity decoding is the brain measurement noise accompanied with the recording process of brain activity. For example, EEG signal is prone to electronic and magnetic interference, eye blinks and movements, scalp muscle activities, and so on [52,53]. fMRI signals may be corrupted by head motions, breathing noise, and cardiac noise [54,55]. Although many preprocessing methods have been proposed to denoise brain recordings as far as possible [56,57], one can hardly guarantee the noises to be thoroughly separated from natural brain activities. Furthermore, since the ground truth for clean brain activities can never be accessed with existing recording methods, it will be difficult to objectively evaluate the effectiveness of denoising approaches. The most intuitive influence caused by the brain recording noises is that the brain activity decoding becomes less accurate. This is because the brain recording noises usually exhibit non-Gaussian distributions, which could be intractable for conventional machine learning algorithms [58]. Such performance degradation due to

adverse noises is known as the issue of "robustness" in machine learning, which has been studied for a lone time since 1960s [59], and continually proven to be important for real-world applications [60,61]. There exist several strategies to realize robust machine learning. First, preprocessing approaches could alleviate the adverse effects of noise by removing or reweighting noisy samples [62,63]. Second, meta-learning techniques can be utilized to achieve robust machine learning [64,65]. Moreover, the machine learning model itself could be implemented with a robust formulation by employing a robust objective function in the model learning process [66–69].

This thesis aims to improve the brain activity decoding performance by proposing robust machine learning algorithms to solve the problem of brain measurement noises, from the perspective of objective function. In particular the main motivation of this thesis is *information theoretic learning* (ITL) framework [70], which refers to utilizing information-theory descriptors to structure objective function for machine learning, e.g., error entropy, correntropy, and mutual information, instead of the conventional statistics including the Euclidean distance and variance. ITL has been revealing growing potential for realizing more advanced machine learning methods. For example, mutual information could be employed for feature selection [71] and unsupervised learning in deep generative model [72]. A latest concept in ITL called information bottleneck can be utilized to interpret the learning process of deep learning models [73,74]. On the focus of this thesis, robust machine learning, ITL has realized promising robustness in many machine learning scenarios. In particular, two learning criteria in ITL have been utilized to build robust models, namely the *minimum error entropy* (MEE) criterion and the *maximum correntropy criterion* (MCC), in many machine learning tasks, including classification [75–78], regression [79–84], feature selection [85–89], and so on. Motivated by the promising implementations of MEE and MCC in robust machine learning, this thesis investigated how to use MEE or MCC to improve the robustness of brain activity decoding for superior performance.

Another significant obstacle for brain activity decoding is the high-dimensional problem, where the number of covariates (also called features or explanatory variables) is larger than that of training samples, which would make it difficult for machine learning models to extract the information most relevant to the task. This is because numerous solutions can achieve good results on the training set in a high-dimensional case, whereas few can generalize well to new testing samples. The high-dimensional problem for brain activity decoding mainly arises from the difficulty of collecting a large number of brain activity trials [90], in contrast to the excellent spatial resolution of fMRI [91,92] or high temporal resolution of EEG [93]. To address the high-dimensional problem in brain activity decoding, scientists have implemented two strategies, subspace dimensionality reduction [94–98] and feature selection [99–102]. However, few of existing brain decoding algorithms designed for high-dimensional scenario have taken the noise issue into account. As a result, although these decoding algorithms can solve the high-dimensional problem, their performance may still be affected and limited by the recording noise. To address this issue, the latter part of this thesis further investigated how to embed the robust ITL into high-dimensional brain activity decoding scenarios, so as to solve the "noise" and "high-dimensional" problems simultaneously for better brain decoding performance.

The remainder of this thesis is organized as follows. Section 2 reviews the fundamental machine learning techniques for brain activity decoding, and then gives a brief introduction concerning ITL, in particular about MEE and MCC. In Section 3, an in-depth discussion of MEE for classification task is presented, through which a new learning criterion is proposed for robust classification that is exactly a special case of MEE. In Section 4, the focus is turned to the high-dimensional case, in which the partial least square, a popular approach for subspace dimensionality reduction, is reformulated with MCC for robust implementation. Section 5 investigates how to employ MCC in the sparse Bayesian learning framework for better robustness which can realize effective feature selection in brain decoding. Finally, Section 6 presents a conclusion for this thesis and several discussions for future works corresponding to the previous sections. A wider prospect for brain activity decoding is provided at the end of this thesis.

2. Technical Background

This section briefly reviews the commonly used brain recording approaches and how conventional machine learning can be utilized to predict or reconstruct desired output from the brain activity. Then, the poor robustness of the conventional machine learning techniques is interpreted with illustrative examples. Subsequently, this section gives an introduction about the ITL framework, in particular for MEE and MCC.

2.1. Brain Activity Decoding by Conventional Machine Learning

The commonly used brain recording approaches can be classified into three categories: invasive, non-invasive, and semi-invasive. Invasive modalities, such as the single-unit activities or local field potentials, usually provide better decoding performance, which would suffer pessimistic long-term stability, however, due to the capriciousness in the recorded neuronal-ensemble [103]. By comparison, non-invasive recordings, such as EEG, MEG, and fMRI, can eliminate the need for craniotomy on the brain which will significantly improve the security of experiments. Therefore, non-invasive modalities are the most popular approaches for brain activity recording and are widely exploited to structure BCI systems due to their ease of use [104]. However, non-invasive brain recordings might be limited in their capability and may require considerable training for BCI control [105]. A sophisticated alternative which has better signal quality than non-invasive EEG while exhibits higher long-term stability than invasive modalities, is the semi-invasive ECoG [9], which places electrodes in direct contact with the surface of cerebral cortex without inserting the electrodes into the cortex to avoid surgical damage.

Another way for brain recording taxonomy could be conducted from the morphology of the brain recording signals. The first category is wave-based recording, including EEG, MEG, and ECoG. This kind of brain measurement can be summarized as the recording of changes in the physical quantities at different locations by multiple sensors on the time scale with a high temporal resolution. As a result, the wave-based recording can be expressed by several waves in the same period, as illustrated in Fig. 4 (a). The second class is image-based recording, such as fMRI and functional near-infrared spectroscopy (fNIRS) [106]. Image-based recordings scan the entire brain iteratively, from which each scan yields a high-spatial-resolution image. However, since each scan usually costs considerable time, there exists a significant time lag between each scan, resulting in a low temporal resolution. Image-based recording is illustrated in Fig. 4 (b). Despite the differences in morphology of various brain recording modalities, brain activities or their induced features can be unified in a vector or matrix for the following pattern recognition.

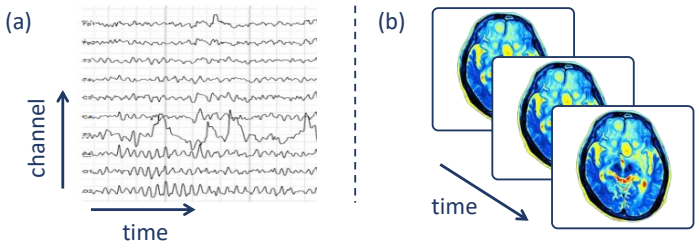

Figure 4. Illustrative examples for (a) wave-based recordings and (b) image-based recordings.

A commonly utilized context for machine learning based brain activity decoding is described in what follows. Consider the recorded brain activities or their induced features, which can be expressed by a D-dimensional vector $\mathbf{x} = (x_1, x_2, \cdots, x_D) \in \mathbb{R}^{1\times D}$, where D denotes the number of covariates or features. $\mathbf{x}$ is usually a continuous variable. The purpose for brain activity decoding is to predict or to reconstruct a target variable t (usually assumed as a scalar) from the brain activity $\mathbf{x}$, which represents the desired intention or the true received stimulus. $\mathbf{x}$ can be regarded to take values in a metric space $\mathcal{X}$ and $\mathcal{T}$ denotes the space for the target variable t. Brain activity decoding can be realized by building a machine learning model $f : \mathcal{X} \to \mathcal{T}$ which can establish the mapping from $\mathbf{x}$ to t. In practice, to train

the machine learning model f, one utilizes a finite dataset $\{(\mathbf{x}_n, t_n)\}_{n=1}^N$ with N observations (samples) which are commonly assumed to be independent and identically distributed (i.i.d). For brain decoding, each pair $(\mathbf{x}_n, t_n)$ including the brain activity $\mathbf{x}_n$ and the corresponding target t_n is usually acquired by an individual trial of cognitive experiments. For a simpler notation, the dataset can be expressed with $\mathbf{X} = (\mathbf{x}_1^T, \mathbf{x}_2^T, \cdots, \mathbf{x}_N^T)^T \in \mathbb{R}^{N\times D}$ and $\mathbf{t} = (t_1, t_2, \cdots, t_N)^T \in \mathbb{R}^{N\times 1}$, where T denotes the transpose and each row of $\mathbf{X}$ represents an individual sample.

Regression means t can be taken from any values in an arbitrary continuous interval. For example, decoding the continuous movement trajectories of a limb from brain activities is a regression task. For regression, the following data generation model is usually used as an assumption

$$t = f^*(\mathbf{x}) + \epsilon \tag{1}$$

where f^* is the desired whereas unknown mapping from $\mathbf{x}$ to t, and ϵ is the measurement noise on this system. For a simplified setting, one could employ the following canonical linear-in-parameter (LIP) model

$$t = \Phi(\mathbf{x})\mathbf{w} + \epsilon \tag{2}$$

where $\Phi(\mathbf{x})$ is a predetermined mapping on $\mathbf{x}$ for feature extraction and $\mathbf{w}$ denotes the model parameter vector which has the same dimension as $\Phi(\mathbf{x})$ for inner product. If the mapping function $\Phi(\cdot)$ is further excluded for simplification, LIP model will degenerate to the linear regression model

$$t = \mathbf{x}\mathbf{w} + \epsilon \tag{3}$$

in which $\mathbf{w} = (w_1, w_2, \cdots, w_D)^T \in \mathbb{R}^{D\times 1}$ is the model parameter which realizes a weighting of each entry of $\mathbf{x}$. The most vintage and widely used method to learn the model parameter $\mathbf{w}$ is to minimize the expectation of the quadratic error

$$\mathbf{w}^* = arg\min_{\mathbf{w}} \mathbb{E}_{p(e)}\left[e^2\right] = arg\min_{\mathbf{w}} \mathbb{E}_{p(t,\mathbf{x})}\left[(t - \mathbf{x}\mathbf{w})^2\right] \tag{4}$$

where $\mathbf{w}^*$ denotes the optimal solution, $\mathbb{E}_{p(\cdot)}[\cdot]$ represents the mathematical expectation with respect to the distribution $p(\cdot)$. The prediction error is defined as the subtraction between the target and the current prediction $e \triangleq t - \mathbf{x}\mathbf{w}$. Eq.(4), which minimizes the variance of ϵ and belongs to the traditional second-order statistics, is called least-square (LS) criterion or mean squared error (MSE) loss function. In practice, the expectation in Eq.(4) is estimated empirically by the finite dataset $\{(\mathbf{x}_n, t_n)\}_{n=1}^N$ with N samples, leading to

$$\mathbf{w}^* = arg\min_{\mathbf{w}} \frac{1}{N}\sum_{n=1}^{N} e_n^2 = arg\min_{\mathbf{w}} \frac{1}{N}\sum_{n=1}^{N} (t_n - \mathbf{x}_n\mathbf{w})^2 \tag{5}$$

which is called empirical risk minimization. By setting the gradient of the MSE loss function to be zero, one can obtain the following closed-form solution

$$\mathbf{w}^* = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{t} \tag{6}$$

This solution is optimal and unbiased if the noise term ϵ exhibits a zero-mean Gaussian distribution.

Classification refers to the condition that t is taken from several discrete values, such as choosing from multiple predetermined options. For example, decoding whether the subject desires to move the left hand or the right hand can be regarded as a classification problem. Binary classification is the most primitive setting where t is taken from either of two options which is commonly denoted by $t \in \{0, 1\}$. There exist two categories for classification models: non-regression-like models and regression-like models [78]. The prediction of non-regression-like models is discrete, and the learning machine can be trained from the perspective of information gain. A representative for the non-regression-like models is the decision tree model [107]. Another category is regression-like classification model, in which the

prediction is probability of continuous variable, including logistic regression [108] and artificial neural networks. The most idealistic solution for classification is supposed to minimize the misclassification rate on the training dataset, whereas the optimization of the corresponding 0-1 loss function is usually intractable. Therefore, many alternatives were proposed by using the convex upper bounds of 0-1 loss, e.g. hinge loss in support vector machine (SVM) and exponential loss in AdaBoost [109,110]. Logistic regression is a widely used probability-based (regression-like) classification model, which uses a linear discriminant function that recognizes two different classes with a weighted summation of each input feature

$$f(\mathbf{x}, \mathbf{w}) = \sum_{d=1}^{D} w_d x_d = \mathbf{x}\mathbf{w} \tag{7}$$

in which $\mathbf{x}$ is the attribute value of the sample and $\mathbf{w}$ is the logistic regression model parameter. A bias term is also usually introduced into the discriminant function, whereas omitted here for the reason of clarity. If the discriminant function $f(\mathbf{x}, \mathbf{w}) < 0$, the corresponding label is predicted as 0. Otherwise, the label is predicted as 1. In $\{0,1\}$-label context, logistic regression computes the probability that $\mathbf{x}$ belongs to class 1 through the *sigmoid* function

$$y \triangleq p(t = 1|\mathbf{x}, \mathbf{w}) = \frac{1}{1 + \exp(-f(\mathbf{x}, \mathbf{w}))} \tag{8}$$

where y denotes the probability for $t = 1$. Logistic regression employs the binomial distribution for the categorical data, where the opposite probability for class 0 is defined by $p(t = 0|\mathbf{x}, \mathbf{w}) = 1 - y$. Given a finite dataset $\{(\mathbf{x}_n, t_n)\}_{n=1}^{N}$ and based on the i.i.d assumption, the likelihood probability can be written

$$\begin{aligned} p(\mathbf{t}|\mathbf{X}, \mathbf{w}) =& p(t_1, t_2, \cdots, t_N|\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N, \mathbf{w}) \\ =& \prod_{n=1}^{N} p(t_n|\mathbf{x}_n, \mathbf{w}) = \prod_{n=1}^{N} y_n^{t_n}(1 - y_n)^{1-t_n} \end{aligned} \tag{9}$$

where $y_n \triangleq p(t_n = 1|\mathbf{x}_n, \mathbf{w})$. The model parameter $\mathbf{w}$ could be learned by maximizing the likelihood function Eq.(9) in a logarithmic form for the maximum likelihood estimation (MLE)

$$\mathbf{w}^* = arg\max_{\mathbf{w}} \log p(\mathbf{t}|\mathbf{X}, \mathbf{w}) = arg\max_{\mathbf{w}} \sum_{n=1}^{N} (t_n \log y_n + (1 - t_n) \log(1 - y_n)) \tag{10}$$

which could be effectively solved by Newton's method since Eq.(10) is a convex optimization problem. After the optimal solution $\mathbf{w}^*$ is acquired, one can predict a new testing sample $\mathbf{x}$ with class 1 provided $f(\mathbf{x}, \mathbf{w}^*) > 0$ (or equally $p(t = 1|\mathbf{x}, \mathbf{w}^*) > 0.5$), or with class 0 otherwise. Eq.(10) can be also interpreted from the perspective of empirical risk minimization that aims to minimize the cross entropy (CE) loss function

$$\mathbf{w}^* = arg\min_{\mathbf{w}} -\frac{1}{N} \sum_{n=1}^{N} (t_n \log y_n + (1 - t_n) \log(1 - y_n)) \tag{11}$$

which is derived from minimizing the Kullback-Leibler divergence between $\{t_n\}_{n=1}^{N}$ and $\{y_n\}_{n=1}^{N}$.

Conventional learning strategies, including the minimization of MSE (for regression) and CE (for classification) loss functions, have been successfully employed in countless real-world applications for pattern recognition. Further, these learning strategies have been also utilized in other data analysis techniques. For example, principal component analysis (PCA), the most famous algorithm in subspace dimensionality reduction, is also formulated by second-order statistics, which exploits the following objective function

$$\mathbf{w}^* = arg\max_{\mathbf{w}} Tr(\mathbf{w}^T \mathbf{X}\mathbf{X}^T \mathbf{w}) = arg\min_{\mathbf{w}} \sum_{n=1}^{N} \left\| \mathbf{x}_n - \mathbf{x}_n \mathbf{w}\mathbf{w}^T \right\|_2^2 \tag{12}$$

where $Tr(\cdot)$ denotes the trace of a matrix, and $\|\cdot\|_2$ is the L_2-norm of a vector. The objective function of PCA could be interpreted to minimize the second order of the reconstruction errors by the projector $\mathbf{w}$. In addition, for the EEG-based motor imagery task, common spatial pattern (CSP) [111–114] algorithm is the most widely used approach for EEG feature extraction, which aims to maximize the separability between the data of different categories with the spatial filter $\mathbf{w}$ by the following objective function

$$\mathbf{w}^* = arg\max_{\mathbf{w}} \frac{\|\mathbf{w}^T\mathbf{X}_1\|_2^2}{\|\mathbf{w}^T\mathbf{X}_0\|_2^2} \tag{13}$$

which maximizes the ratio between the variances of the spatially filtered data of different classes. $\mathbf{X}_1$ and $\mathbf{X}_0$ denote the EEG data matrices for class 1 and class 0, respectively, which are usually centralized and normalized in advance. One can find that, CSP is also based on the second-order statistics, i.e., the variance.

Despite the successful applications of conventional learning strategies, they cannot perform well in all situations. For example, the solution of least-square regression Eq.(6) is optimal only in the case of Gaussian-distributed noise. If the noise ϵ exhibits a non-Gaussian distribution, Eq.(6) will lead to a biased solution. In particular, if the training data is corrupted by outliers, which refer to the samples that are largely deviated from the regular data distribution, the second-order statistics based machine learning model will be deteriorated significantly. This is fundamentally because second-order statistics assigns excessive importance to the large entry for the objective function, while the outlier will exactly correspond to a large prediction error with the desired model parameter [115,116]. As a result, outliers will dominate the learning process and the model cannot extract effective information from regular samples. Fig. 5 illustrates an example about how the CSP algorithm is deteriorated evidently by only one outlier. A synthetic dataset is utilized with two different classes in respective colors. With regular samples, CSP can acquire the spatial filter which maximizes the ratio of filtered variances of different classes, as shown in the solid line. However, when only one outlier is added to the top left corner, the spatial filter is corrupted to be the dashed line, which is obviously deviated from the optimal one.

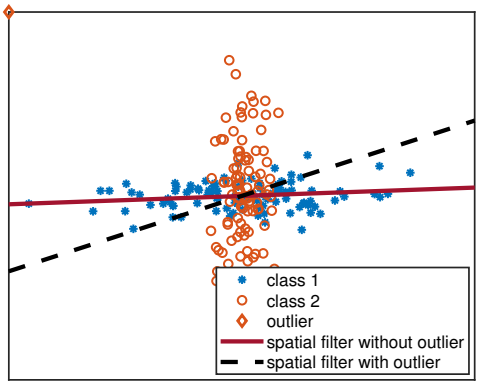

Figure 5. Illustrative example of common spatial pattern and how only one outlier would corrupt the spatial filter significantly.

2.2. *Information Theoretic Learning*

In the middle of the 20th century, information theory was proposed and developed as a pioneering research field for designing communication systems, the core of which locates in the quantification of the abstract "information" through the concept of *entropy* [117]. Claude E. Shannon, the pioneer of information theory, proposed a classical formula to measure the degree of chaos of a random variable which is called Shannon's entropy

$$H_S(p(x)) \triangleq -\int_x p(x)\log p(x)dx \tag{14}$$

where x denotes an arbitrary continuous random variable with its probability density function (PDF) $p(x)$. With the development of information theory, Shannon's entropy H_S has been extended to more generalized forms. One representative is the Rényi's entropy proposed by Alfréd Rényi [118]

$$H_{R,\alpha}(p(x)) \triangleq \frac{1}{1-\alpha} \log \int_x p^{\alpha}(x) dx \tag{15}$$

where $H_{R,\alpha}$ denotes the Rényi's entropy of α-order and α is a free parameter. One can find that Rényi's entropy will degenerate to Shannon's entropy when $\alpha \to 1$.

In the recent two decades, scientists have begun to investigate how to utilize information-theory descriptors to structure the objective function for machine learning model, which is called *Information Theoretic Learning* (ITL) [70]. This thesis mainly focuses on the implementation of ITL in robust machine learning. In what follows, two learning criteria of ITL with exceptional robustness will be introduced, which have a close relation with Rényi's entropy.

2.2.1. Minimum Error Entropy Criterion

A common diagram for supervised machine learning can be expressed in Fig. 6.

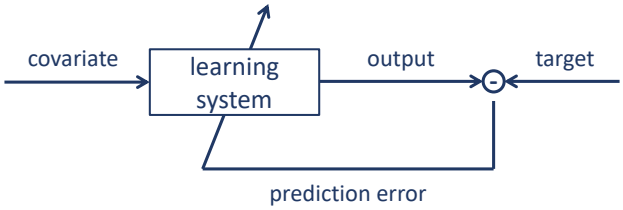

Figure 6. Schematic diagram of supervised machine learning.

One can find that, the prediction error e indicates the difference between the current prediction and the target, which contains the information for such difference. The purpose of supervised machine learning is to restore the target variable as far as possible, which means to preserve the information of the data generating system as much as possible. If the information contained in the prediction error is minimized, it means the learned information in the learning system is maximized for predicting target variable.

As mentioned before, entropy is exactly an adequate concept to measure the information contained in a random variable. Therefore, a natural conception to design a supervised machine learning model is to minimize the entropy of the prediction error, which is called *Minimum Error Entropy* (MEE) criterion. MEE is one fundamental and popular approach in the ITL field, which commonly employs the Rényi's entropy in practice due to its easier implementation than Shannon's entropy. MEE has been utilized to propose state-of-the-art robust algorithms for regression [70,84], feature extraction [89], dimensionality reduction [85,119], subspace clustering [86,120], and so on.

For the mathematical expressions in MEE, the learning criterion for the optimal model parameter $\mathbf{w}^*$ can be denoted by

$$\mathbf{w}^* = arg \min_{\mathbf{w}} H_{R,\alpha}(p(e)) = arg \min_{\mathbf{w}} \frac{1}{1-\alpha} \log \int_e p^{\alpha}(e) de \tag{16}$$

in which $p(e)$ denotes the PDF for the prediction error e. One can further define the *information potential* (IP) as the term in the logarithm

$$I_{\alpha}(p(e)) \triangleq \int_e p^{\alpha}(e)\, de = \mathbb{E}_{p(e)}\left[p^{\alpha-1}(e)\right] \tag{17}$$

For simplicity, the free parameter α is usually set as $\alpha = 2$. Thus, the objective function for MEE will become

$$\mathbf{w}^* = arg\min_{\mathbf{w}} H_{R,2}(p(e)) = arg\min_{\mathbf{w}} -\log \int_e p^2(e)de \tag{18}$$

Since the logarithm function is a monotonically increasing function, Eq.(18) can be also expressed by

$$\mathbf{w}^* = arg\max_{\mathbf{w}} I_2(p(e)) = arg\max_{\mathbf{w}} \mathbb{E}_{p(e)}\left[p(e)\right] \tag{19}$$

To realize the optimization of Eq.(19) with a finite dataset $\{(\mathbf{x}_n, t_n)\}_{n=1}^N$ in practice, one could compute the current predictions $\{y_n\}_{n=1}^N$ by a temporary learning system, and then obtain the prediction errors $\{e_n\}_{n=1}^N$ by $e_n = t_n - y_n$. To acquire an empirical estimation of $p(e)$, denoted by $\hat{p}(e)$, one could utilize the Parzen's estimator [121,122]

$$\hat{p}(e) = \frac{1}{N}\sum_{n=1}^{N} \kappa(e - e_n) \tag{20}$$

where $\kappa(\cdot)$ is a Mercer kernel function which is usually adopted with the Gaussian kernel function

$$\kappa_h(e) = \frac{1}{\sqrt{2\pi h}} \exp(-\frac{e^2}{2h}) \tag{21}$$

with the kernel bandwidth h. Then, the empirical estimation for the second-order IP can be computed

$$\begin{aligned} \hat{I}_2\left(p(e)\right) &= \mathbb{E}_{p(e)}\left[\hat{p}(e)\right] \\ &= \frac{1}{N}\sum_{n=1}^{N} \hat{p}(e_n) \\ &= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} \kappa_h(e_i - e_j) \end{aligned} \tag{22}$$

Thus, the empirical objective function for MEE is written

$$\mathbf{w}^* = arg\max_{\mathbf{w}} \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} \kappa_h(e_i - e_j) \tag{23}$$

One can regard the PDF estimator $\hat{p}(\cdot)$ as an adaptive objective function since it changes with $\{e_n\}_{n=1}^N$, which is different from the conventional ones that are generally invariable. This adaptation will result in extra advantages, as proved theoretically and confirmed numerically [70]. Entropy provides a PDF concentration measure that higher concentration implies lower entropy, which is the initial motivation to use entropic risk functionals. For continuous variable, the local minimum of $H_{R,2}(p(e))$ corresponds to a PDF represented by several Dirac-δ functions, a Dirac-δ comb. When all errors are zero, a single Dirac-δ at the origin for error PDF can be achieved as the ideal situation $p(e) = 0|_{e\neq 0}$. This demands a learning machine to guarantee the convergence of the error PDF towards a single Dirac-δ at the origin.

The robustness of MEE could be briefly explained as follows. In the learning process with regular samples, MEE ensures most of errors are close to zero so as to approach a Dirac-δ function at the origin. If outliers happen, the error PDF will not only hold a main peak at the origin, but also generate small peaks at large errors caused by outliers. This kind of distribution, as mentioned above, is also a local minimum for MEE. It can be also interpreted from Eq.(23). For a large error caused by outlier, its effect on the maximization is weakened since the Gaussian kernel function of Eq.(21) is bounded, which can saturate the summation term $\kappa_h(e_i - e_j)$. Theoretical insights for robustness of MEE are investigated in [70,123,124].

To alleviate the computational bottleneck caused by the double summation in Eq.(23), quantization can be implemented where the error PDF is estimated by a representative codebook with fewer samples,

which is acquired from the original error set $\{e_n\}_{n=1}^N$, so that the inner summation for PDF estimation could be decreased [84]. In this way, an alternative for MEE with quantization, called quantized MEE (QMEE), is expressed

$$\begin{aligned}\mathbf{w}^* &= arg\max_{\mathbf{w}} \frac{1}{N}\sum_{n=1}^{N}\hat{p}(e_n)\\ &\approx arg\max_{\mathbf{w}} \frac{1}{N}\sum_{n=1}^{N}\hat{p}_Q(e_n)\\ &= arg\max_{\mathbf{w}} \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\kappa_h(e_i - Q[e_j])\\ &= arg\max_{\mathbf{w}} \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{M}\varphi_j\kappa_h(e_i - c_j)\end{aligned} \tag{24}$$

where $\hat{p}_Q(e)$ is the estimated error PDF based on some representative samples in the original error set $\{e_n\}_{n=1}^N$. $Q[\cdot]$ denotes a quantization operator that leads to a codebook $C = (c_1, c_2, ..., c_M)$, where $Q[\cdot]$ is a function which maps each error sample e_i to one of the representatives c_j. The parameter $\Phi = (\varphi_1, \varphi_2, ..., \varphi_M)$ denotes the number that how many samples are quantized to the corresponding element in C. Obviously, one can know $\sum_{j=1}^M \varphi_j = N$. Since $\{c_j\}_{j=1}^M$ is a representative description of $\{e_i\}_{i=1}^N$, one usually has $M \ll N$, and thus the complexity to compute the objective function would be decreased from $O(N^2)$ to $O(MN)$. By theoretical analysis and experimental results, QMEE can realize the commensurate performance as the original MEE with proper quantization [84,89]. This is because the elements $\{c_j\}_{j=1}^M$ in codebook are representative enough for the whole errors $\{e_i\}_{i=1}^N$ where each $\{\varphi_j\}_{j=1}^M$ acts as weight, so that QMEE can realize the same effect as the original MEE. To realize proper quantization, an adaptive method was proposed in [89], as summarized in Algorithm 1.

Algorithm 1 Adaptive quantization procedure for QMEE

input:
original error set $\{e_i\}_{i=1}^N$;
quantization threshold ε (usually $\varepsilon = 0.05$ or 0.1);
initialize:
$C_1 = \{e_1\}$, where C_i denotes the codebook at the ith iteration;
output:
quantization result $\{Q[e_i]\}_{i=1}^N$;
Compute the error interval $\psi = \max(e_i) - \min(e_i)$;
for $i = 2, \cdots, N$ **do**
  compute the minimum distance between e_i and C_{i-1}: $dis(e_i, C_{i-1}) = \min_{1\le j\le |C_{i-1}|} |e_i - C_{i-1}(j)|$ where $C_{i-1}(j)$ denotes the j-th element of C_{i-1}, and $|C_{i-1}|$ is the length of C_{i-1};
  if $dis(e_i, C_{i-1}) \le \varepsilon \cdot \psi$ **then**
    keep the codebook unchanged: $C_i = C_{i-1}$ and quantize e_i to the closest code word: $Q[e_i] = C_{i-1}(j^*)$, where $j^* = arg\min |e_i - C_{i-1}(j)|$;
  else
    update the codebook: $C_i = \{C_{i-1}, e_i\}$ and quantize e_i to itself: $Q[e_i] = e_i$;
  end if
end for

2.2.2. Maximum Correntropy Criterion

Another popular learning criterion in ITL with excellent robustness is called *maximum correntropy criterion* (MCC), which is based on a sophisticated measure "correntropy" [79]. Correntropy was first proposed as a generalized correlation function for accurate description of a stochastic process by inner

products of vectors in a kernel feature space [125]. Although the original correntropy was designed for a single stochastic process, this generalized correlation function was further extended to the general case of two arbitrary random variables. For two arbitrary random variables a and b, their correntropy measure is defined by the expectation of the kernel function between them

$$V(a,b) \triangleq \mathbb{E}_{p(a,b)}\left[k(a-b)\right] \tag{25}$$

in which $p(a,b)$ is the joint distribution of a and b. In practice, with their N observations $\{(a_n,b_n)\}_{n=1}^{N}$, the empirical estimation of correntropy is given based on the Gaussian kernel function

$$\hat{V}(a,b) = \frac{1}{N}\sum_{n=1}^{N} k_h(a_n-b_n) \propto \frac{1}{N}\sum_{n=1}^{N} \exp(-\frac{(a_n-b_n)^2}{2h}) \tag{26}$$

An important property of correntropy is that it contains all the even moments of the variable $e \triangleq a-b$

$$V(a,b) = \frac{1}{\sqrt{2\pi h}}\sum_{n=0}^{\infty}\frac{(-1)^n}{2^n n!}\mathbb{E}_{p(a,b)}\left[\frac{(a-b)^{2n}}{h^n}\right] \tag{27}$$

When the kernel bandwidth h is increased, the higher-order moments will decompose faster than the lower-order moments. For an extreme case $h \to \infty$, the second-order moments will dominate Eq.(27) and correntropy will degenerate to MSE between a and b. Hence, correntropy can be regarded as a generalized form for conventional second-order statistics, while all the properties of correntropy are controlled by one single free parameter, kernel bandwidth h. The objective function for MCC could be written

$$\mathbf{w}^* = arg\max_{\mathbf{w}} \frac{1}{N}\sum_{n=1}^{N}\exp(-\frac{e_n^2}{2h}) \tag{28}$$

Using a correntropy-based learning criterion exhibits many benefits. First, the value of correntropy is mainly determined by the Gaussian kernel function k_h along $a=b$, which means it is a local similarity measure and could alleviate the negative effect of large deviations caused by the outliers. Therefore, in a supervised machine learning task, maximizing the correntropy (MCC) between the model prediction and the desired target exhibits exceptional robustness with respect to outliers. From the viewpoint of kernel methods, correntropy induces a nonlinear mapping which transforms data from the original space to an infinite-dimensional reproducing kernel Hilbert space (RKHS). Correntropy is also related closely to the m-estimation that can be regarded as a robust formulation of Welsch m-estimator [126]. Therefore, MCC is essentially a non-parametric m-estimator. In addition, correntropy induces a metric in the sample space which obeys the properties of non-negativity, identity of indiscernible, symmetric, and triangle inequality. Correntropy induced metric (CIM) is defined by

$$CIM(a,b) \triangleq \left[1-V(a,b)\right]^{\frac{1}{2}} \tag{29}$$

which is illustrated with the contours between a random variable and the origin in a two-dimensional space in Fig. 7. One could observe that $CIM(x,0)$ behaves differently in different domains. When x is close to the origin, $CIM(x,0)$ is similar to L_2-norm, which can be found from the Taylor expansion in Eq.(27). Outside of this "Euclidean zone", $CIM(x,0)$ behaves similarly as L_1-norm in a further range, and eventually like L_0-norm which is insensitive to distance. This provides a geometric interpretation for the desired robustness of MCC with respect to the outlier. Correntropy has been further extended with different generalized forms, such as correntropy with a variable center [127], mixture correntropy [128], multi-kernel correntropy [129], and correntropy with a generalized kernel density function [82].

In addition to the excellent robustness of MEE and MCC with respect to the non-Gaussian noise for supervised learning, they have been also utilized for robust data characterization in unsupervised learning or feature extraction. In particular, they have been used to propose robust objective functions

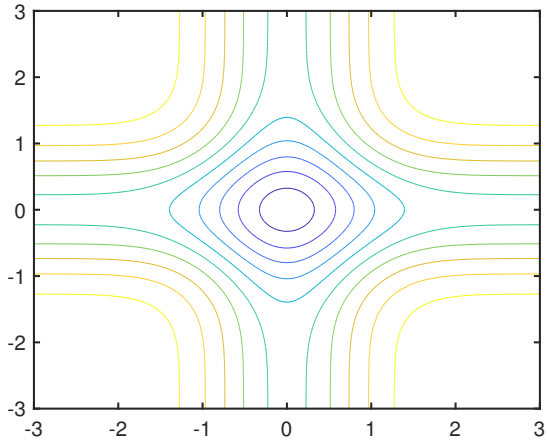

Figure 7. Contours of $CIM(x,0)$ in a two-dimensional space.

for CSP algorithm for EEG feature extraction. In [87], CIM was utilized as a robust substitute for the conventional variance in Eq.(13), leading to CSP-CIM algorithm

$$\mathbf{w}^* = arg\max_{\mathbf{w}} \frac{CIM(\mathbf{w}^T\mathbf{X}_1, 0)}{CIM(\mathbf{w}^T\mathbf{X}_0, 0)} \tag{30}$$

which can effectively deal with the outliers in EEG recordings for feature selection. Compared with correntropy-based learning, MEE not only has good robustness against outliers, but also exhibits good adaptability to other non-Gaussian noises, such as multi-modal distribution noise. Motivated by this, CSP-CIM algorithm was further extended by a QMEE-aware measure of dispersion with the following objective function, named as CSP-QMEE [89]

$$\mathbf{w}^* = arg\max_{\mathbf{w}} \frac{1 - \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{M}\varphi_j\kappa_h(\mathbf{w}^T\mathbf{x}_{1,i} - c_j)}{1 - \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{M}\varphi'_j\kappa_h(\mathbf{w}^T\mathbf{x}_{0,i} - c'_j)} \tag{31}$$

in which φ_j and c_j are obtained by the quantization of $\mathbf{w}^T\mathbf{X}_1$, while φ'_j and c'_j are obtained from the quantization of $\mathbf{w}^T\mathbf{X}_0$. Experimental results demonstrate that CSP-QMEE could realize superior feature extraction for noisy EEG recordings.

3. Restricted Minimum Error Entropy Criterion

Despite the satisfactory robustness of ITL-based CSP algorithms, as introduced before, achieving robustness only in feature extraction is not sufficient. The reason is, even if with an expected spatial filter, to say the features of regular samples are not affected, the deviated features by the contaminated samples still survive in the dataset. These non-informative features (commonly with large amplitudes) are adverse noises to the classification model, which may significantly degenerate the learning process. Therefore, in addition to robust feature extraction, achieving robust classification is also important for robust brain activity decoding, which means the learning process of classification model is less affected by noises than by regular samples [130].

It has been shown that convex loss functions for classification are not robust to outliers [131,132], which mainly arises from the unbounded property of the convex loss functions, which would assign large losses on outliers [132–135]. Consequently, the learning process is mainly determined by outliers, rather than those meaningful samples, and the decision boundaries could be affected severely, leading to significant performance degradation. For example, the poor robustness of logistic regression results from the log likelihood function $\log p(\mathbf{t}|\mathbf{X}, \mathbf{w})$ of Eq.(10), because it assigns excessive weight to large errors [78,136]. Although Eq.(10) or Eq.(11) does not contain the prediction error e_n explicitly, [78] gave

an analytical form of $\log p(\mathbf{t}|\mathbf{X}, \mathbf{w})$ with respect to prediction error. In $\{0, 1\}$-label context, maximizing the log likelihood $\log p(\mathbf{t}|\mathbf{X}, \mathbf{w})$ in Eq.(10) or minimizing the CE loss function in Eq.(11) is equivalent to minimizing the following loss function with respect to prediction error

$$\mathbf{w}^* = arg \min_{\mathbf{w}} \mathbb{E}_{p(e)} \left[-\log(2 - 2te) \right] \tag{32}$$

which is shown in Fig. 8. In particular, one could observe that when the error is close to 1 (the worst case), the loss value will be infinite. As a result, the noise will play a dominant rule with the use of the conventional objective function in logistic regression, and the classification model will have difficulty extracting valid information from noisy data.

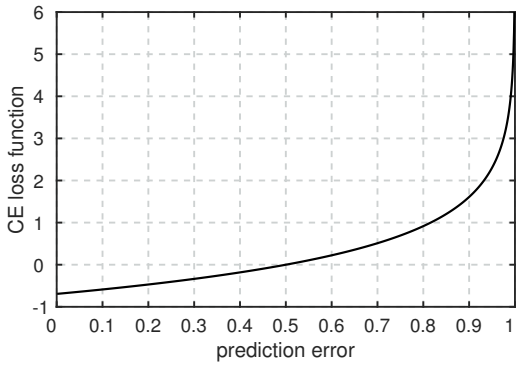

Figure 8. Loss function of cross entropy with respect to prediction error e ($t = 1$).

Different from the noise in regression tasks which means that attribute value diverges from the foreseeable distribution, the noise in classification is more complicated and can be classified into two categories: attribute noise and label noise [131,137]. The attribute noise means a measurement error resulting from noisy sensors, recordings, communications, or data storage, and the label noise means a mistake when labeling samples. As stated in [132], label noise could result from mutual elements as attribute noise, such as communication errors, whereas it mainly arises from expert elements [138]: i) unreliable labeling due to insufficient information, ii) unreliable non-expert for low cost, iii) subjective labeling. Not to mention, class is not always totally distinguishable as *lived* and *died* [139]. The outlier, as a more severe case of noise [140], usually causes serious performance degradation. According to the above taxonomy, it can be stated that attribute outliers are deviate attribute values whereas irrelevant to label information, while label outliers imply that some distinct samples are assigned with wrong labels. Note that mislabeled samples are not necessarily label outliers since they could occur near the boundary region thus being less adverse for classification model [132]. However, for the brain activity decoding task, one could find that the principal contamination in the dataset will happen in the brain recordings that are utilized as the explanatory variables. Therefore, to realize a robust classification for brain activity decoding, one should mainly consider the attribute contamination.

To realize robust classification model, many methods have been proposed to suppress the adverse effects of outliers, such as removing or relabeling training samples in data preprocessing [132,141–143] or re-weighting the samples to reduce the outliers' proportion in the learning process [132,144,145]. In addition, recovery of clean data by robust principal component analysis can realize robust classification as well [146]. Moreover, meta-learning technique can achieve robustness by evaluating the gradient for each data point at the learned parameters [147]. From the perspective of robust objective functions, bounded Savage loss was proposed to construct robust SavageBoost algorithm [134], and [135] further extended this work. In [148], a robust SVM algorithm was developed based on the ramp loss. In [149], the truncated least square loss was proposed for the robust least square SVM. In particular, MCC has also been utilized for robust classification, called C-Loss [75,76]. By contrast, the potential robustness of MEE with respect to outliers in classification has not been thoroughly explored.

This section aims to investigate an implementation of MEE method for robust classification model. Although MEE has been widely used for a robust implementation in other machine learning tasks, as introduced in Section 2.2.1, robust classification based on MEE is a rather vacancy in the literature and has been only discussed in detail in [78]. In this section, an in-depth discussion for MEE in classification is provided, and a new learning criterion for robust classification which is closely related to QMEE is proposed.

3.1. Error Distribution Analysis for Noisy Classification

To facilitate the proposal of a new robust objective function for classification, this part first presents an error distribution analysis in noisy classification tasks. In the context of $\{0,1\}$-label coding scheme and logistic regression, note that the essential prediction is the probability $y \in (0,1)$ of Eq.(8), which is a continuous variable. As a result, one can obtain a continuous error variable $e = t - y \in (-1,1)$ by subtraction. Denoting the class prior probability by $p = p(t=1)$ and $q = 1 - p = p(t=0)$, one can obtain the cumulative distribution function of error $F(e)$

$$\begin{aligned} F(e) &= p(e \leqslant E) \\ &= pp(e \leqslant E|t=1) + qp(e \leqslant E|t=0) \\ &= pp(1-y \leqslant E|t=1) + qp(-y \leqslant E|t=0) \\ &= p(1 - F_{y|t=1}(1-e)) + q(1 - F_{y|t=0}(-e)) \\ &= 1 - pF_{y|t=1}(1-e) - qF_{y|t=0}(-e) \end{aligned} \tag{33}$$

where $F_{y|t=1}$ and $F_{y|t=0}$ denote the class-conditional cumulative distribution functions for class 1 and class 0, respectively. Thus, the error PDF $p(e)$ can be obtained by the differential of $F(e)$

$$p(e) = p \cdot p_{y|t=1}(1-e) + q \cdot p_{y|t=0}(-e) \tag{34}$$

in which $p_{y|t=1}$ and $p_{y|t=0}$ denote the class-conditional distributions of the prediction y. Suppose that the covariates of two classes can be denoted by $\mathbf{x}|_{t=1} \sim p_{\mathbf{x}|t=1}(\mathbf{x})$ and $\mathbf{x}|_{t=0} \sim p_{\mathbf{x}|t=0}(\mathbf{x})$. To acquire $p_{y|t}$ from $p_{\mathbf{x}|t}$, a famous theorem is first given as follows.

Theorem 1: Assume $p_x(x)$ is the PDF of a random variable x, with $\vartheta(x)$ a monotonic and differentiable function. If $p_y(y)$ denotes the PDF of $y = \vartheta(x)$ and $\vartheta'(x) \neq 0, \forall x \in X$, then one has

$$p_y(y) = \begin{cases} \frac{p_x(\vartheta^{-1}(y))}{|\vartheta'(\vartheta^{-1}(y))|} & \inf\vartheta(x) < y < \sup\vartheta(x) \\ 0 & \text{otherwise} \end{cases} \tag{35}$$

in which $x = \vartheta^{-1}(y)$ is the inverse function of $y = \vartheta(x)$. ■

Since the *sigmoid* function of Eq.(8) satisfies the conditions, the following three heuristics with logistic regression can be presented by this theorem. For clarity, supposing class 0 stands for negative and class 1 for positive, one can denote those outliers located in the positive region whereas assigned with negative labels as false negative (*FN*) outliers, and vice versa as false positive (*FP*) outliers.

1: Suppose that $p_{\mathbf{x}|t=1}(\mathbf{x})$ and $p_{\mathbf{x}|t=0}(\mathbf{x})$ are Gaussian distributions, where $p_{\mathbf{x}|t=1}(\mathbf{x}) = \mathcal{N}(\mu_1, \Sigma_1)$ and $p_{\mathbf{x}|t=0}(\mathbf{x}) = \mathcal{N}(\mu_0, \Sigma_0)$, respectively. Given the model parameter $\mathbf{w}$, $\mathbf{xw}$ will be subject to a univariate Gaussian distribution $\mathcal{N}(\mu_t\mathbf{w}, \mathbf{w}^T\Sigma_t\mathbf{w})$. Then one calculates the PDF of $y|_t = \frac{1}{1+\exp(-\mathbf{x}_t\mathbf{w})}$ as

$$p_{y|t}(y) = \frac{1}{y(1-y)} \cdot \frac{\exp(-\frac{(\log(\frac{y}{1-y}) - \mu_t\mathbf{w})^2}{2\mathbf{w}^T\Sigma_t\mathbf{w}})}{\sqrt{2\pi\mathbf{w}^T\Sigma_t\mathbf{w}}} \tag{36}$$

where $y \in (0,1)$. Substituting Eq.(36) into Eq.(34), one obtains

$$p(e) = \frac{p}{e(1-e)} \cdot \frac{\exp(-\frac{(\log(\frac{1-e}{e})-\mu_1\mathbf{w})^2}{2\mathbf{w}^T\Sigma_1\mathbf{w}})}{\sqrt{2\pi\mathbf{w}^T\Sigma_1\mathbf{w}}} - \frac{q}{e(1+e)} \cdot \frac{\exp(-\frac{(\log(\frac{-e}{1+e})-\mu_0\mathbf{w})^2}{2\mathbf{w}^T\Sigma_0\mathbf{w}})}{\sqrt{2\pi\mathbf{w}^T\Sigma_0\mathbf{w}}} \tag{37}$$

The optimal model parameter is supposed to push majority of the prediction $y|_t$ to the corresponding t. Intuitive function curves of $p_{y|t}(y)$ with specific $\mu_t\mathbf{w}$ and $\mathbf{w}^T\Sigma_t\mathbf{w}$ are plotted in Fig. 9 (a) with solid lines. In addition, if the dataset suffers some outliers, one can expect that the predictions of *FP* outliers approach 0 according to the definition, and $p_{y|t=1}(y)$ exhibits a distribution peak at $y=0$ as a result. In the same way, *FN* outliers engender $p_{y|t=0}(y)$ with a distribution peak at $y=0$. The distributions caused by outliers are illustrated in dashed lines. Thereafter, the error PDF $p(e)$ is plotted in Fig. 9 (b) with the same scenario.

2: Suppose that $p_{\mathbf{x}|t=1}(\mathbf{x})$ and $p_{\mathbf{x}|t=0}(\mathbf{x})$ are uniform distributions, in which $p_{\mathbf{x}|t=1}(\mathbf{x}) = \mathcal{U}(a_1,b_1)$ and $p_{\mathbf{x}|t=0}(\mathbf{x}) = \mathcal{U}(a_0,b_0)$, respectively. Comparably, $y|_t$ obeys the following distribution

$$p_{y|t}(y) = \frac{1}{y(1-y)} \cdot \frac{1}{(b_T-a_T)\mathbf{w}} \tag{38}$$

where $y \in (\frac{1}{1+\exp(-a_T\mathbf{w})}, \frac{1}{1+\exp(-b_T\mathbf{w})})$. Subsequently, one calculates the error PDF $p(e)$ as

$$p(e) = \frac{1}{e(1-e)(b_1-a_1)\mathbf{w}} - \frac{1}{e(1+e)(b_0-a_0)\mathbf{w}} \tag{39}$$

Similarly, intuitions of $p_{y|t}(y)$ and $p(e)$ under the assumption of uniform distribution are illustrated in Fig. 9 (c)(d), respectively. The resultant distribution of outliers is plotted in dashed lines as before.

3: Suppose $p_{\mathbf{x}|t=1}(\mathbf{x})$ and $p_{\mathbf{x}|t=0}(\mathbf{x})$ are Gaussian mixture distributions, both of which are composed of two Gaussian distributions, for which one has $p_{\mathbf{x}|t=1}(\mathbf{x}) = k_1^1\mathcal{N}(\mu_1^1,\Sigma_1^1) + k_1^2\mathcal{N}(\mu_1^2,\Sigma_1^2)$ and $p_{\mathbf{x}|t=0}(\mathbf{x}) = k_0^1\mathcal{N}(\mu_0^1,\Sigma_0^1) + k_0^2\mathcal{N}(\mu_0^2,\Sigma_0^2)$. Illustrations of $p_{y|t}(y)$ and $p(e)$ with this assumption are similarly shown in Fig. 9 (e)(f), respectively.

In Fig. 9 (b)(d)(f), one observes that error distribution exhibits three peaks on $\{0,-1,1\}$ remarkably in each scenario. Such consistent occurrence could suggest that the three-peak PDF is probably the optimal error distribution in many circumstances. Previous literature has provided more exhaustive discussions for the error distribution in the presence of outliers. In classification, outliers customarily exhibit the opposite predictions from the corresponding labels, resulting in the worst cases, i.e. $e=\pm1$ [150]. Additionally, [151] presented a literature review about the outlier detection techniques used in logistic regression, which generally identify those with largest distances between predictions and true labels as potential outliers. Moreover, [152] demonstrated that the errors from regular samples usually converge around zero, whereas those from outliers could be more likely to reveal large values.

To formulate the optimal error distribution with the three-peak distribution for noisy classification tasks, one can assume that each peak is close enough to a Dirac-δ function so that the density of the desired error PDF is zero beyond the three peaks. Formally, this three-peak distribution, denoted by $\rho(e)$, can be expressed by

$$\rho(e) = \begin{cases} \zeta_0 & e=0 \\ \zeta_{-1} & e=-1 \\ \zeta_1 & e=1 \\ 0 & \text{otherwise} \end{cases} \tag{40}$$

where ζ_i $(i=0,-1,1)$ denotes the corresponding density for each peak, which is closely related to the proportion of each type of samples. To be specific, ζ_0 is the proportion of regular samples since the corresponding peak results from those samples that are supposed to be classified correctly. Similarly, one can know ζ_1 (or ζ_{-1}) is the proportion of *FP* (or *FN*) outliers. Fig. 10 illustrates a heuristic example

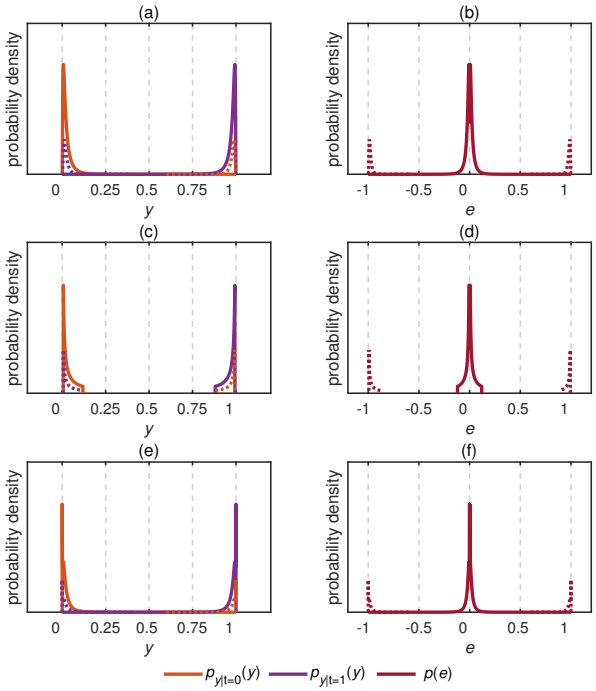

Figure 9. Illustrative examples of $p_{y|t}(y)$ and $p(e)$ (assume $p = q = 0.5$). (a)(b): Gaussian distributions $\mathbf{wx}|_{t=0} \sim \mathcal{N}(-4, 1)$, $\mathbf{wx}|_{t=1} \sim \mathcal{N}(4, 1)$, (c)(d): uniform distributions $\mathbf{wx}|_{t=0} \sim \mathcal{U}(-5, -2)$, $\mathbf{wx}|_{t=1} \sim \mathcal{U}(2, 5)$, and (e)(f): Gaussian-mixture distributions $\mathbf{wx}|_{t=0} \sim 0.3\mathcal{N}(-7, 1) + 0.7\mathcal{N}(-4, 1)$, $\mathbf{wx}|_{t=1} \sim 0.7\mathcal{N}(4, 1) + 0.3\mathcal{N}(7, 1)$.

to support such hypothesis. One could perceive that only the desirable decision boundary realizes a three-peak error distribution with the appropriate ζ values. By comparison, although those wrong solutions also achieve 'three-peak' error distributions, the peaks show considerable deviations from the expected ζ values.

3.2. *Minimum Error Entropy for Classification*

Until now, one can know that the optimal error distribution in a noisy classification task exhibits three-peak distribution, as formulated by $\rho(e)$ of Eq.(40). On the other hand, based on the introduction for MEE in Section 2.2.1, one can also know that MEE is in particular proper for the cases where the error is of multi-modal distribution. Therefore, it will be a natural idea that MEE could realize good robustness in noisy classification tasks. However, compared to noisy regression tasks, one can find that applying MEE (or QMEE) directly to a noisy classification task would encounter additional difficulties. In particular, when the outlier proportion is increased, the performance of MEE-based classification model will degenerate fast and significantly (one will find concrete experimental results for this later in Section 3.4), which has been discussed in detail in [78] that "MEE is harder for classification than for regression". The main difficulties can be explained as follows.

In binary classification, according to Eq.(34) and Eq.(19), the purpose of MEE can be decomposed by

$$\begin{aligned} \min H_{R,2}(p(e)) &\Leftrightarrow \max I_2(p(e)) \\ &\Leftrightarrow \max p^2 I_2\left(p_{y|t=1}(1-e)\right) + q^2 I_2\left(p_{y|t=0}(-e)\right) \end{aligned} \tag{41}$$

in which the class-conditional property causes the difficulty. Recall that, class-conditional distributions, entropies, and information potentials depend on the model parameter $\mathbf{w}$, while this dependency has been omitted for simplicity. Minimizing $H_{R,2}(p(e))$ implies maximizing the sum of $p^2 I_2(p_{y|t=1}(1-e))$

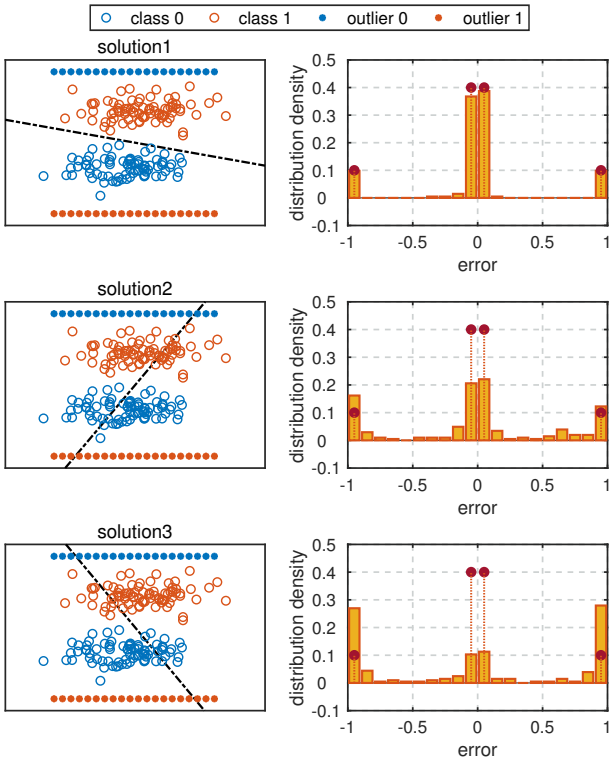

Figure 10. This example consists of 160 regular samples (80 for each class), 20 *FP* outliers, and 20 *FN* outliers, respectively. According to Eq.(40), one can know $\zeta_0 = 0.8$ and $\zeta_{-1} = \zeta_1 = 0.1$, which are marked with red points in the error distribution histograms. Note that ζ_0 is divided equally into two compositions since in practical calculation, errors of regular samples approach zero from both positive and negative directions.

and $q^2 I_2(p_{y|t=0}(-e))$, both of which are dependent on the model parameter $\mathbf{w}$. As a result, it would be hard to say about the minimum of $H_{R,2}(p(e))$ since it depends on p, q, $p_{y|t=1}$, and $p_{y|t=0}$ simultaneously. One has to consider each class-conditional distribution individually and study them together to realize the minimization of $H_{R,2}(p(e))$. By comparison, regression does not suffer from the class-conditional property on $H_{R,2}(p(e))$, which is much easier to deal with.

For a more specific and intuitive scenario in which MEE could fail for classification, MEE based classifiers may predict all samples as the same class with large confidence. For example, suppose that each predicted probability $\{y_n\}_{n=1}^N$ is close to 0. Thus, the errors from 0-class samples would be close to 0, while those from 1-class samples would be close to 1, resulting in the $p(e)$ with two approximate Dirac-δ functions at $\{0, 1\}$, respectively. The basic explanation was already given as before: any Dirac-δ comb achieves local minimum entropy. When this case happens, the classification accuracy could be even the chance level. This instability inspires that, only focusing on minimizing the error entropy, i.e. maximizing the information potential, is not sufficient.

*3.3. **Restricted Minimum Error Entropy Criterion***

To propose a robust learning criterion for noisy classification tasks, the motivation can be obtained from the three-peak optimal error distribution $\rho(e)$. The inspiration is that the optimal parameter $\mathbf{w}^*$ will lead to the error distribution $\rho(e)$. If a classifier is designed to acquire a similar error distribution, it can probably achieve satisfactory robust classification. To implement this conception, one can first get rid of the MEE framework temporarily and focus on driving the error PDF obtained by the learning process towards the optimal three-peak distribution $\rho(e)$ of Eq.(40).

To enforce two distributions as similar as possible, a basic idea is to maximize a similarity measure between their PDFs. There exist many similarity measures for PDF in the literature, for which [153] provides a comprehensive review. Here the elementary *inner product* is used to measure the similarity between two distributions, which is generalized from its use for vectors [154,155]. The inner-product similarity between two arbitrary continuous PDFs $p(x)$ and $g(x)$ is defined by

$$\langle p(x), g(x) \rangle \triangleq \int_x p(x)g(x)dx \tag{42}$$

Now one can maximize this similarity measure between the current error PDF $p(e)$ and the desired distribution $\rho(e)$ by

$$\begin{aligned} &\max \langle p(e), \rho(e) \rangle \\ &\Leftrightarrow \max \int_e p(e)\rho(e)de \\ &\Leftrightarrow \max \zeta_0 p(e=0) + \zeta_{-1} p(e=-1) + \zeta_1 p(e=1) \end{aligned} \tag{43}$$

where the last equality is because $\rho(e)$ always exhibits a zero probability density except for $e = 0, -1$, or 1. In practice, with a finite dataset $\{(\mathbf{x}_n, t_n)\}_{n=1}^N$ which induces the error set $\{e_n\}_{n=1}^N$, maximization of $\langle p(e), \rho(e) \rangle$ is realized with the empirical estimation on $p(e)$ by

$$\begin{aligned} \mathbf{w}^* &= arg\max_{\mathbf{w}} \langle \hat{p}(e), \rho(e) \rangle \\ &= arg\max_{\mathbf{w}} \zeta_0 \hat{p}(e=0) + \zeta_{-1} \hat{p}(e=-1) + \zeta_1 \hat{p}(e=1) \\ &= arg\max_{\mathbf{w}} \begin{pmatrix} \zeta_0 \frac{1}{N} \sum_{n=1}^N \kappa_h (0 - e_n) \\ +\zeta_{-1} \frac{1}{N} \sum_{n=1}^N \kappa_h (-1 - e_n) \\ +\zeta_1 \frac{1}{N} \sum_{n=1}^N \kappa_h (1 - e_n) \end{pmatrix} \\ &= arg\max_{\mathbf{w}} \frac{1}{N} \sum_{n=1}^N \begin{pmatrix} \zeta_0 \kappa_h (e_n) \\ +\zeta_{-1} \kappa_h (e_n + 1) \\ +\zeta_1 \kappa_h (e_n - 1) \end{pmatrix} \\ &= arg\max_{\mathbf{w}} \frac{1}{N^2} \sum_{n=1}^N \begin{pmatrix} N\zeta_0 \kappa_h (e_n) \\ +N\zeta_{-1} \kappa_h (e_n + 1) \\ +N\zeta_1 \kappa_h (e_n - 1) \end{pmatrix} \end{aligned} \tag{44}$$

One might have noticed the comparability between this form and QMEE, because this formula Eq.(44) can be regarded as a special case of QMEE of Eq.(24), if the codebook $C = (0, -1, 1)$, the corresponding quantization number $\Phi = (N\zeta_0, N\zeta_{-1}, N\zeta_1)$, and obviously $M = 3$. Note that the derivation of Eq.(44) has originally nothing to do with the MEE framework because it aims to maximize the inner-product similarity between the current error PDF $p(e)$ and the desired distribution $\rho(e)$.

Returning back to MEE framework, the meaning of Eq.(44) can be interpreted as follows. QMEE aims to concentrate the prediction errors as close as possible to each $\{c_j\}_{j=1}^M$ to achieve a relatively narrow error distribution, where $\{\varphi_j\}_{j=1}^M$ act as weighting parameters. One could expect that if the codebook is assigned with some specific values, QMEE will focus the training errors close to these predetermined positions. By this consideration, QMEE is implemented with a predetermined codebook $C = (0, -1, 1)$, the purpose of which is to restrict errors on these three positions to avoid the undesired double-peak training consequence. QMEE with a restricted codebook, called *restricted MEE* (RMEE), is thus proposed by using the predetermined codebook $C = (0, -1, 1)$

$$\mathbf{w}^* = arg\max_{\mathbf{w}} \frac{1}{N^2} \sum_{n=1}^N \begin{pmatrix} \varphi_0 \kappa_h (e_n) \\ +\varphi_{-1} \kappa_h (e_n + 1) \\ +\varphi_1 \kappa_h (e_n - 1) \end{pmatrix} \tag{45}$$

where $\Phi = (\varphi_0, \varphi_{-1}, \varphi_1) = (N\zeta_0, N\zeta_{-1}, N\zeta_1)$ denotes the corresponding number for each quantization word $C = (0, -1, 1)$. One can find the essential difference between QMEE and the proposed RMEE is, the codebook C of the former is obtained by a data-driven method as in Algorithm 1 that aims to make the elements $\{c_j\}_{j=1}^{M}$ as representative to the entirety as possible, while the latter's is predetermined, which aims to drive the error PDF $p(e)$ towards the desired one $\rho(e)$.

In the following, for the proposed RMEE, the optimization, convergence analysis, and how to determine the hyper-parameters will be discussed.

3.3.1. Optimization

In Eq.(45), the Gaussian kernel function will bring non-convexity in optimization, not to mention the implicit *sigmoid* transformation which is intractable particularly. Therefore, the *half-quadratic* (HQ) technique is utilized to solve this problem, which is often used to solve ITL optimization [76,77,156,157]. To derive the HQ-based optimization, first the following theorem is given.

Theorem 2: Define a convex function $g(v) = -v\log(-v) + v$, in which $v < 0$. Based on the conjugate function theory [158], one has

$$\exp(-\frac{(t-y)^2}{2h}) = \sup_{v<0}\{v\frac{(t-y)^2}{2h} - g(v)\} \tag{46}$$

where the supremum is achieved at $v = -\exp(-\frac{(t-y)^2}{2h}) < 0$.

Proof: By definition, the conjugate function $g^*(u)$ of $g(v) = -v\log(-v) + v$ $(v < 0)$ is written as

$$g^*(u) = \sup_{v<0}\{uv - g(v)\} = \sup_{v<0}\{uv + v\log(-v) - v\} \tag{47}$$

in which v is the optimization variable. One could directly obtain the solution of (47) by making the differential of $(uv + v\log(-v) - v)$ equal to zero since this is a concave function with respect to v. The result is

$$u + \log(-v) = 0 \Rightarrow v = -\exp(-u) < 0 \tag{48}$$

Therefore one can see that

$$g^*(u) = \sup_{v<0}\{uv + v\log(-v) - v\} = \exp(-u) \tag{49}$$

in which the equality is established if and only if $v = -\exp(-u) < 0$. If one replaces u with $\frac{(t-y)^2}{2h}$, then one can obtain

$$\begin{aligned} g^*(\frac{(t-y)^2}{2h}) &= \sup_{v<0}\{\frac{(t-y)^2}{2h}v + v\log(-v) - v\} \\ &= \exp(-\frac{(t-y)^2}{2h}) \end{aligned} \tag{50}$$

where the supremum is achieved at $v = -\exp(-\frac{(t-y)^2}{2h}) < 0$. ■

Thus, the objective function of RMEE Eq.(45) can be rewritten as

$$\begin{aligned}\mathbf{w}^* &= arg\max_{\mathbf{w}} \sum_{n=1}^{N} \begin{pmatrix} \varphi_0 \sup_{u_n<0}\{u_n \frac{e_n^2}{2h} - g(u_n)\} \\ +\varphi_{-1} \sup_{v_n<0}\{v_n \frac{(e_n+1)^2}{2h} - g(v_n)\} \\ +\varphi_1 \sup_{s_n<0}\{s_n \frac{(e_n-1)^2}{2h} - g(s_n)\} \end{pmatrix} \\ &= arg \max_{\mathbf{w},u_n<0,v_n<0,s_n<0} \sum_{n=1}^{N} \begin{pmatrix} \varphi_0(u_n \frac{e_n^2}{2h} - g(u_n)) \\ +\varphi_{-1}(v_n \frac{(e_n+1)^2}{2h} - g(v_n)) \\ +\varphi_1(s_n \frac{(e_n-1)^2}{2h} - g(s_n)) \end{pmatrix} \\ &\triangleq arg \max_{\mathbf{w},u_n<0,v_n<0,s_n<0} J_R(\mathbf{w}, u_n, v_n, s_n)\end{aligned} \tag{51}$$

Now one can optimize $J_R(\mathbf{w}, u_n, v_n, s_n)$ by alternate optimization on $\mathbf{w}$, u_n, v_n, and s_n, respectively. To be specific, in the k-th iteration with the current errors $\{e_n\}_{n=1}^N$, one first optimizes

$$\begin{aligned}(u_n^k, v_n^k, s_n^k) &= arg \max_{u_n<0,v_n<0,s_n<0} \sum_{n=1}^{N} \begin{pmatrix} \varphi_0(u_n \frac{e_n^2}{2h} - g(u_n)) \\ +\varphi_{-1}(v_n \frac{(e_n+1)^2}{2h} - g(v_n)) \\ +\varphi_1(s_n \frac{(e_n-1)^2}{2h} - g(s_n)) \end{pmatrix} \\ &= arg \max_{u_n<0,v_n<0,s_n<0} J_{R1}(u_n, v_n, s_n)\end{aligned} \tag{52}$$

According to **Theorem 2**, the closed-form solution of Eq.(52) is

$$\begin{aligned} u_n^k &= -\exp(-\tfrac{e_n^2}{2h}) < 0 \\ v_n^k &= -\exp(-\tfrac{(e_n+1)^2}{2h}) < 0 \\ s_n^k &= -\exp(-\tfrac{(e_n-1)^2}{2h}) < 0 \\ &(n = 1, 2, ..., N)\end{aligned} \tag{53}$$

Second, with the updated (u_n^k, v_n^k, s_n^k) in the k-th iteration, one obtains $\mathbf{w}^{*,k}$ by solving the following optimization

$$\begin{aligned}\mathbf{w}^{*,k} &= arg\max_{\mathbf{w}} \sum_{n=1}^{N} \begin{pmatrix} \varphi_0(u_n \frac{e_n^2}{2h} - g(u_n)) \\ +\varphi_{-1}(v_n \frac{(e_n+1)^2}{2h} - g(v_n)) \\ +\varphi_1(s_n \frac{(e_n-1)^2}{2h} - g(s_n)) \end{pmatrix} \\ &= arg\max_{\mathbf{w}} \sum_{n=1}^{N} \begin{pmatrix} \varphi_0 u_n (t_n - y_n)^2 \\ +\varphi_{-1} v_n (t_n + 1 - y_n)^2 \\ +\varphi_1 s_n (t_n - 1 - y_n)^2 \end{pmatrix} \\ &= arg\max_{\mathbf{w}} J_{R2}(\mathbf{w})\end{aligned} \tag{54}$$

Note that for different classification models, the forms of y_n are different. For example, in the logistic regression model, y_n is obtained by Eq.(8). Despite the differences in the form of y_n, one could always

optimize $J_{R2}(\mathbf{w})$ of Eq.(54) with the prominent gradient-based method, since the objective function $J_{R2}(\mathbf{w})$ is continuous and differentiable. In the context of logistic regression, the gradient of $J_{R2}(\mathbf{w})$ is

$$\begin{aligned}\frac{\partial J_{R2}(\mathbf{w})}{\partial \mathbf{w}} &= \sum_{n=1}^{N} \begin{pmatrix} \varphi_0 u_n \frac{\partial (t_n - y_n)^2}{\partial \mathbf{w}} \\ +\varphi_{-1} v_n \frac{\partial (t_n + 1 - y_n)^2}{\partial \mathbf{w}} \\ +\varphi_1 s_n \frac{\partial (t_n - 1 - y_n)^2}{\partial \mathbf{w}} \end{pmatrix} \\ &= -2 \sum_{n=1}^{N} \begin{pmatrix} \varphi_0 u_n e_n \\ +\varphi_{-1} v_n (e_n + 1) \\ +\varphi_1 s_n (e_n - 1) \end{pmatrix} \mathbf{x}_n y_n (1 - y_n)\end{aligned} \tag{55}$$

Based on the gradient, one can use the popular momentum-based optimization, such as the popular and efficient *Adam* algorithm [159], to obtain $\mathbf{w}^{*,k}$ in Eq.(54). The HQ-based optimization for RMEE is summarized in Algorithm 2.

Algorithm 2 RMEE for robust classification

1: **input**:
   training samples $\{(\mathbf{x}_n, t_n)\}_{n=1}^N$;
   Gaussian kernel bandwidth h;
   quantization weight $\Phi = (\varphi_0, \varphi_{-1}, \varphi_1)$;
2: **initialize**:
   model parameters $\mathbf{w}$;
3: **output**:
   model parameters $\mathbf{w}$;
4: **repeat**
5: compute the prediction errors $\{e_n\}_{n=1}^N$ at the current model parameter $\mathbf{w}$;
6: update (u_n, v_n, s_n) with Eq.(53);
7: update $\mathbf{w}$ with Eq.(54);
8: **until** the parameter change is small enough or the number of iterations exceeds a predetermined limit

3.3.2. Convergence Analysis

The convergence of HQ-based optimization can be easily proved as follows by

$$\begin{aligned} J_R(\mathbf{w}^{k-1}, u_n^{k-1}, v_n^{k-1}, s_n^{k-1}) &\leqslant J_R(\mathbf{w}^{k-1}, u_n^k, v_n^k, s_n^k) \\ &\leqslant J_R(\mathbf{w}^k, u_n^k, v_n^k, s_n^k) \end{aligned} \tag{56}$$

in which the first inequality is established obviously according to Eq.(52)(53). To establish the second inequality, i.e. $J_R(\mathbf{w}^{k-1}, u_n^k, v_n^k, s_n^k) \leqslant J_R(\mathbf{w}^k, u_n^k, v_n^k, s_n^k)$, the following equivalent inequality is expected with fixing $(u_n, v_n, s_n) = (u_n^k, v_n^k, s_n^k)$ as

$$J_{R2}(\mathbf{w}^{k-1}) \leqslant J_{R2}(\mathbf{w}^k) \tag{57}$$

Hence, to guarantee the convergence of Algorithm 2, one could find it not necessary for $\mathbf{w}$ to achieve a strict maximum in Eq.(54). In contrast, as long as one has $J_{R2}(\mathbf{w}^{k-1}) \leqslant J_{R2}(\mathbf{w}^k)$ at every iteration in training with fixing $(u_n, v_n, s_n) = (u_n^k, v_n^k, s_n^k)$, the inequality Eq.(56) can be established which ensures the convergence of Algorithm 2. Therefore, for the optimization problem in Eq.(54), one only needs to consider whether the new $\mathbf{w}^k$ achieves a larger objective function value than the original $\mathbf{w}^{k-1}$, which brings large convenience in practical implementation.

3.3.3. Hyper-Parameter Determination

The kernel bandwidth h plays a vital role in Parzen-window-based methods. Since there is only one kernel bandwidth, it can be determined by the conservative five-fold cross-validation method. On the other hand, considering the determination of Φ, the optimal values should be the numbers of regular samples, *FN* outliers, and *FP* outliers, corresponding to φ_0, φ_{-1}, and φ_1, respectively. However, this will be intractable unless one has a prior information about the outlier proportion. To determine Φ without any prior information, the following empirical method is utilize to obtain an approximate estimation of outlier proportion. One first uses an initial $\Phi' = (\varphi'_0, \varphi'_{-1}, \varphi'_1) = (N, 0, 0)$, i.e. $\zeta_0 = 1$ and $\zeta_{-1} = \zeta_1 = 0$ in $\rho(e)$, and train the model by Algorithm 2, which means that one expects all samples in the training dataset to achieve minor errors. This will give a resultant model parameter $\mathbf{w}$, from which one can obtain $\{e_n\}_{n=1}^N$ belonging to the continuous interval $(-1, 1)$. Then one estimates the outlier proportion by assuming the correctly predicted samples, whose errors belong to $(-0.5, 0.5)$, are regular samples. On the other hand, the errors belonging to $(-1, -0.5)$ and $(0.5, 1)$ correspond to *FN* and *FP* outliers, respectively. Formally, one has

$$\begin{aligned} \varphi''_0 &= \#\{e_n \in (-0.5, 0.5)\} \\ \varphi''_{-1} &= \#\{e_n \in (-1, -0.5)\} \\ \varphi''_1 &= \#\{e_n \in (0.5, 1)\} \end{aligned} \tag{58}$$

in which $\#\{\cdot\}$ indicates counting the samples that meet the condition. Obviously $\varphi''_0 + \varphi''_{-1} + \varphi''_1 = N$. With the updated $\Phi'' = (\varphi''_0, \varphi''_{-1}, \varphi''_1)$, train the model again by Algorithm 2 and obtain the result of RMEE.

The above procedure is in fact adaptive. When the training dataset does not contain outliers, it can be supposed that almost all sample are classified well, which means φ''_{-1} and φ''_1 will be of small values. Thus in the following training with Φ'', one will still expect almost all examples to achieve zero errors. On the other hand, if there are outliers in training dataset, considerable errors will be outside $(-0.5, 0.5)$. Then, φ''_{-1} and φ''_1 could reflect the outlier proportion to some extent, since higher outlier proportion will generally lead to worse training results, i.e. larger φ''_{-1} and φ''_1. In addition, note that RMEE with the initial weights $\Phi' = (\varphi'_0, \varphi'_{-1}, \varphi'_1) = (N, 0, 0)$ is actually equivalent to the C-Loss. Thus RMEE can be regarded as a more generalized form of C-Loss.

*3.4. **Experiments***

For performance comparison, it is principal to compare RMEE with QMEE to demonstrate the necessity of the proposed restriction. Furthermore, the C-Loss is also involved in comparison, which is a special case of RMEE, when $\Phi = (\varphi_0, \varphi_{-1}, \varphi_1) = (N, 0, 0)$. In addition, the traditional CE loss is involved as well. One might worry that there are too few algorithms for performance comparison. It would be argued that:

1: There are a variety of approaches to realize robust classification, such as removing samples, relabeling samples, weighting samples, etc. What most of these methods have in common is that, the desired robustness is realized in the preprocessing stage before the model learns, rather than in the learning processes. The proposed RMEE is a robust objective function for classification, which means RMEE realizes robustness exactly in the learning process. Therefore, it is not necessary to compare RMEE with those methods that achieve robustness outside the objective function. Even RMEE can be used with these methods together.

2: What should exactly be compared with RMEE are those robust objective functions for classification, among which C-Loss has been proved to be state-of-the-art. Unlike traditional bounded losses, which are usually truncated by hard threshold [133,149], C-Loss is always differentiable, and its kernel size could realize adaptive approximation to various norms in different ranges.

The most canonical performance indicator for classification tasks is the accuracy that is computed by *(TP+TN)/(TP+TN+FP+FN)*. In the following experiments, all the average accuracy are given by 100

Monte-Carlo independent repetitions. Note that as suggested in [131], to evaluate the robustness for different machine learning models, it is preferable to only contaminate training dataset with outliers, and to keep the testing dataset from being contaminated. This policy has been widely recognized and practiced in the robust machine learning literature. Therefore, in what follows, the outlier corruptions are only implemented at the training datasets, while the testing datasets are unchanged.

3.4.1. Synthetic Dataset

First, a linear synthetic dataset is generated to evaluate the performance by a similar method as in [143]. In this synthetic dataset, 1000 i.i.d. training samples and 1000 i.i.d. testing samples are randomly generated with a standard Gaussian distribution $\mathbf{x}_n \sim \mathcal{N}(0, I_D)$, with a random true model parameter $\mathbf{w}^* \sim \mathcal{N}(0, I_D)$, where I_D is the unit matrix of dimension D. The dimension D for this dataset is set as 20. For all the samples, the labels are assigned 1 if $\mathbf{x}_n\mathbf{w}^* \geqslant 0$ by the discriminant function of Eq.(7), or 0 otherwise. The numbers of two classes are supposed to be equal because $\mathbf{w}^*$ always passes through the center of symmetrical Gaussian-distributed samples. In this way, a pure dataset is completed.

According to [131], generally speaking, attribute contamination has no tendency for samples of different classes, because it usually occurs during the measurement process. Therefore, the samples of two classes will sustain attribute contamination with equal probability. To contaminate the attribute values of training samples with outliers, the attribute values of each training sample is corrupted by the following noise

$$\epsilon \sim (1-\theta)\mathcal{N}(\epsilon|0, \Sigma_{small}) + \theta\mathcal{N}(\epsilon|0, \Sigma_{large}) \qquad (0 \leqslant \theta \leqslant 1) \tag{59}$$

in which $\mathcal{N}(0, \Sigma_{small})$ with a small variance is used to generate normal noises, while $\mathcal{N}(0, \Sigma_{large})$ with a large variance can generate outliers. Each noise enforced on the attribute values is sampled from this corruption model, which means every training sample will be contaminated by outliers with the probability θ, or it will be added with normal noise otherwise. For this synthetic dataset, to evaluate the robustness, θ is increased from 0 to 1.0 with a step 0.05, Σ_{small} is set as $0.05I_d$, and Σ_{large} is assessed with $5I_d$, $10I_d$, $20I_d$, $30I_d$, $50I_d$, $100I_d$, $200I_d$, $500I_d$, and $1000I_d$, respectively.

The results of accuracy are plotted in Fig. 11, where one can clearly observe that RMEE achieves the highest accuracy under almost all conditions, which highlights the superiority of MEE for robust classification when restricted by the proposed codebook.

3.4.2. EEG-Based Motor Imagery Dataset

Next, robustness of the proposed RMEE are evaluated on noisy EEG datasets. Considering the potential non-linearity in the features extracted from the EEG signals, in this subsection, the extreme learning machine (ELM) [160] model is utilized for performance evaluation, which is a supervised single-hidden-layer neural network and initializes input weights and hidden layer biases randomly as shown in Fig. 12. The robust variant of ELM based on C-Loss was proposed in [77]. The number of nodes in the hidden layer is set as 50.

Two popular public EEG datasets are employed for performance comparison, including Dataset IIb of BCI Competition IV and Dataset IIIa of BCI Competition III, respectively, both of which include the EEG data of the motor imagery task, which is one of the most popular research topics in the BCI community and has relatively higher requirements for algorithms [161,162]. A common diagram of motor imagery experiment is illustrated in Fig. 13.

Considering robust feature extraction for noisy EEG data, robust CSP approach is effective. Hence, the CSP-L_1 is selected for feature extraction, which was proposed by reformulating the conventional CSP method by the L_1-norm and could realize admirable robustness in feature extraction for noisy multi-channel EEG [163]. The first three spatial filters corresponding to the largest objective function values are used, and vice versa. To say, each trial will be assigned with a 6-dimensional feature.

1) Dataset IIb of BCI Competition IV:

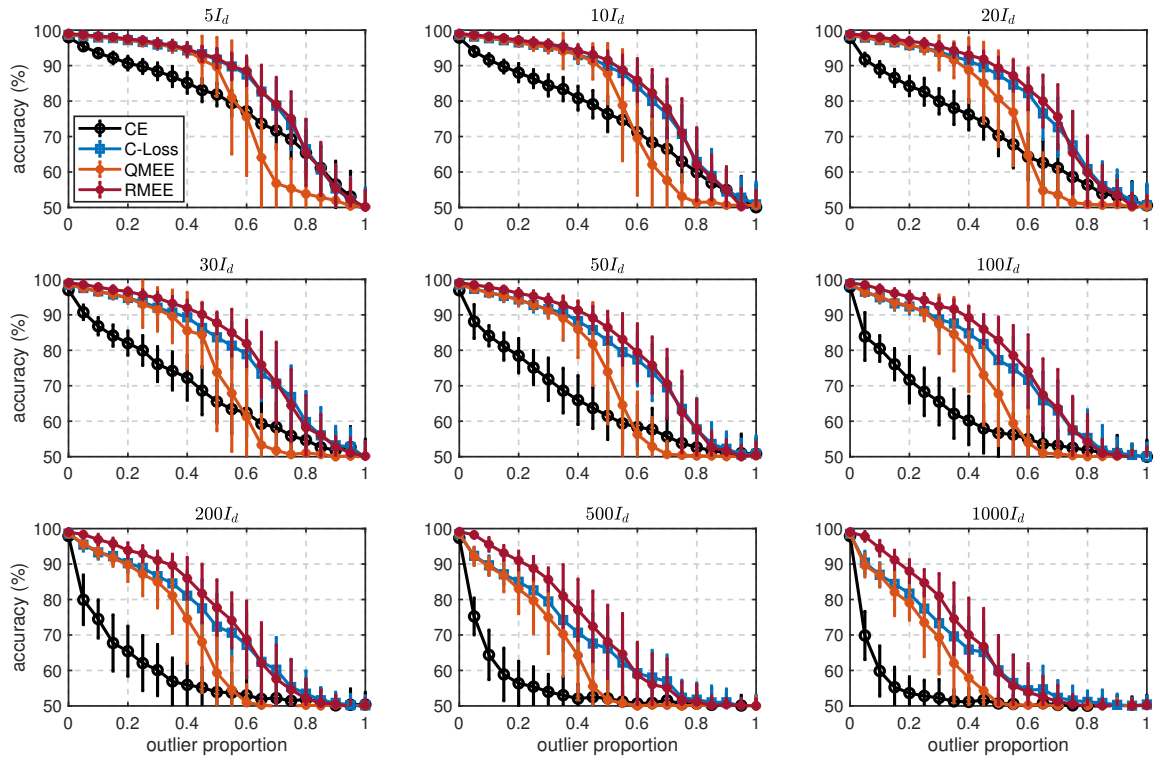

Figure 11. Average classification accuracy of the synthetic dataset contaminated by attribute outliers.

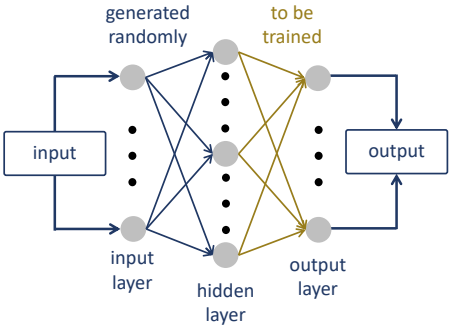

Figure 12. Schematic diagram of the extreme learning machine model.

The Dataset IIb of BCI Competition IV consists of EEG data from nine subjects with three bipolar EEG channels, acquired from a binary motor imagery task, i.e. left hand and right hand. Three bipolar recordings (C3, Cz, and C4) were recorded with sampling frequency of 250 Hz. The EEG signals were bandpass filtered between 0.5 and 100 Hz with a 50 Hz notch filter enabled in recording. In addition, the EEG segments are preprocessed using a 10-order Butterworth filter with cutoff frequencies 8 and 35 Hz. For each subject, five sessions are provided, whereby the first three sessions are for training while the last two are for testing. The numbers of trials of each class for training and testing, respectively, are summarized in Table 1. In this dataset, Subject 1, Subject 2, and Subject 3 are abandoned, because the classification accuracy of these three subjects is even no more than 60% without any contamination, which means the evaluation of robustness on these subjects is relatively meaningless.

For the contamination on EEG data, some training trials are selected randomly, which are replaced with stochastic values of multivariate Gaussian distribution. Since each trial is normalized individually in preprocessing, the covariance of multivariate Gaussian distribution will not make difference. The contaminated trials proportion is increased from 0 to 0.5 with a step 0.05. The results of accuracy are

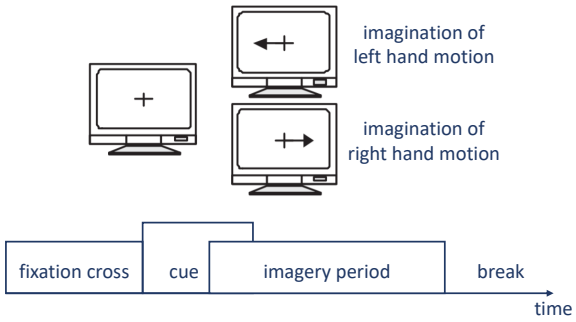

Figure 13. Cue-based experiment paradigm of motor imagery experiments.

Table 1. Numbers of trials in Dataset IIb of BCI Competition IV.

Subject No.	Training Left	Training Right	Testing Left	Testing Right
4	201	198	154	153
5	193	189	134	139
6	142	153	123	128
7	178	178	114	118
8	175	153	119	111
9	157	160	121	124

Table 2. Average classification accuracy of Dataset IIb of BCI Competition IV.

Contaminated Trials Proportion		0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
Subject 4	CE	97.4788	90.8404	85.5993	79.5733	75.1922	69.4235	67.5472	64.8436	61.5049	63.0000	60.4821
	C-Loss	97.2801	96.1889	96.2313	94.4723	90.6352	85.8046	82.7850	**78.3779**	67.5635	**63.4886**	**63.8241**
	QMEE	97.3420	96.9479	96.4853	95.1107	88.4235	77.3746	73.4821	69.4235	64.7850	60.8469	55.1010
	RMEE	**97.6743**	**97.2476**	**96.7394**	**95.7492**	**92.9837**	**87.8893**	**83.4919**	77.4137	**67.7883**	62.4267	61.4821
Contaminated Trials Proportion		0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
Subject 5	CE	**72.3663**	69.0623	65.4982	63.6190	61.9487	59.7766	58.2051	57.8425	56.7473	54.3004	54.8425
	C-Loss	71.2344	71.1941	69.3297	68.6740	68.2125	67.4176	**65.1868**	**65.2637**	**61.5128**	**60.6007**	**57.8901**
	QMEE	71.7179	71.4029	69.7802	64.7912	63.7619	56.6410	56.0073	53.7546	53.2491	51.8828	52.1282
	RMEE	71.7839	**71.5421**	**70.7546**	**70.0476**	**69.2418**	**67.8498**	65.0842	64.8388	60.4835	58.3480	57.5495
Contaminated Trials Proportion		0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
Subject 6	CE	77.7450	74.1594	69.2988	66.5896	62.8964	58.2351	58.5578	57.3665	55.2869	54.4940	53.7211
	C-Loss	78.0757	76.9801	74.9283	73.3785	71.5259	68.1036	65.0956	64.2550	**60.7410**	**59.3904**	**59.0518**
	QMEE	77.0478	77.1594	74.1315	72.4303	71.7371	**68.5179**	**65.7888**	60.4143	55.8964	56.3147	56.1753
	RMEE	**78.5418**	**78.0757**	**75.4343**	**74.4263**	**72.3426**	68.3785	65.3307	**64.8327**	60.2311	58.4861	58.1833
Contaminated Trials Proportion		0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
Subject 7	CE	76.1853	74.6638	74.2629	71.6293	70.5647	71.1466	68.4397	66.9440	66.6810	63.1250	63.0388
	C-Loss	**77.0000**	76.2371	76.0819	75.6250	75.0345	**76.0259**	74.0647	73.8017	**71.3491**	**70.8362**	**66.6509**
	QMEE	75.7845	76.0474	75.8448	73.7845	70.9440	68.2845	64.7500	60.8491	57.6940	54.3879	55.2845
	RMEE	76.7845	**76.4440**	**76.2759**	**76.0086**	**75.3233**	75.2457	**75.0345**	**74.3664**	71.0733	70.2845	65.9353
Contaminated Trials Proportion		0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
Subject 8	CE	91.8087	91.0217	89.9130	85.8957	83.2522	80.1261	76.9304	70.0391	66.4826	63.3739	60.9435
	C-Loss	**92.2217**	91.2435	91.7826	91.5478	90.3826	89.5739	86.3043	82.8565	79.1261	73.0043	70.3435
	QMEE	92.1957	91.4957	90.5435	88.8435	84.4304	78.8913	72.5391	70.0087	66.1478	62.8696	58.3826
	RMEE	92.1261	**92.0826**	**92.0304**	**91.8435**	**91.0826**	**90.4391**	**87.7826**	**84.4870**	**81.4913**	**73.9826**	**71.8391**
Contaminated Trials Proportion		0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
Subject 9	CE	**81.4816**	76.9102	76.1102	75.0898	72.2980	69.4898	65.5347	65.3714	62.8531	61.6286	59.4939
	C-Loss	81.2490	80.1061	79.8286	79.1102	78.6490	77.7959	75.4000	71.8286	65.4000	**63.4245**	**61.4082**
	QMEE	80.9755	80.0082	79.5837	78.8367	76.4367	70.7837	60.8408	56.4857	53.4327	52.3388	52.3633
	RMEE	81.1510	**80.5388**	**80.0449**	**79.7429**	**79.0735**	**78.6857**	**77.0857**	**73.0776**	**66.5469**	62.7837	60.4204

listed in Table 2, where the highest accuracy in each condition is marked in bold. One can obviously find that the proposed RMEE achieves the largest number of the highest accuracy.

2) Dataset IIIa of BCI Competition III:

The Dataset IIIa of BCI Competition III consists of EEG data from 3 subjects with 60 EEG channels, acquired from four different cued motor imagery tasks, i.e. left hand, right hand, both feet, and tongue. Only the trials of left hand and right hand are considered. The EEG signals were sampled in 250 Hz, with filtered between 1 and 50 Hz with Notch filter on. Since all trials are included in one session, the entirety are divided randomly into training and testing parts, where 2/3 trials are for training while

the other 1/3 trials are for testing. In preprocessing, the same conduct is operated as before. For each subject, the numbers of trials of each class for training and testing, respectively, are summarized in Table 3. The contamination of EEG data is similar to the previous EEG dataset, and the contaminated trials proportion is increased from 0 to 0.5 with a step 0.05 as well. The corresponding results are presented in Table 4, where one can observe that the proposed RMEE achieves the highest accuracy in almost all cases for Subject 1 and Subject 3, whereas each criterion is approximate for Subject 2.

Table 3. Numbers of trials in Dataset IIIa of BCI Competition III.

Subject No.	Training Left	Training Right	Testing Left	Testing Right
1	60	60	30	30
2	40	40	20	20
3	40	40	20	20

Table 4. Average classification accuracy of Dataset IIIa of BCI Competition III.

Contaminated Trials Proportion		0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
Subject 1	CE	94.9167	90.3333	88.4167	86.6500	83.4500	80.5167	77.5667	73.5833	69.4833	67.6333	65.9167
	C-Loss	**95.0667**	92.5833	91.3333	90.1667	89.7667	88.0333	87.4333	85.6000	81.8500	77.1833	73.8833
	QMEE	94.6500	92.7500	91.5167	90.3333	89.1833	87.2833	83.9833	76.4333	65.7833	59.5667	58.1333
	RMEE	94.6333	**93.3167**	**91.9833**	**91.0333**	**90.1333**	**89.4000**	**88.7667**	**86.9333**	**84.2000**	**80.1000**	**76.7333**
Contaminated Trials Proportion		0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
Subject 2	CE	**74.525**	68.450	67.625	66.850	63.150	62.200	60.075	60.025	58.600	56.875	56.525
	C-Loss	74.475	70.400	**69.200**	**68.900**	**65.000**	62.900	**63.275**	**60.825**	57.050	56.425	55.100
	QMEE	73.225	**70.975**	67.525	67.100	62.800	59.625	58.150	58.475	57.675	55.050	54.925
	RMEE	74.300	70.900	69.000	68.250	64.850	**63.575**	62.100	60.275	**59.050**	**57.875**	**57.575**
Contaminated Trials Proportion		0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
Subject 3	CE	**84.850**	82.400	80.300	79.650	77.750	76.700	75.975	74.750	73.675	71.100	67.600
	C-Loss	84.725	**84.250**	83.225	82.725	81.775	81.125	79.500	78.975	78.550	76.450	75.125
	QMEE	83.975	83.875	82.925	81.825	79.525	76.200	73.325	70.450	68.200	65.525	62.950
	RMEE	84.325	**84.250**	**83.725**	**83.025**	**82.275**	**81.625**	**80.725**	**79.950**	**79.050**	**76.950**	**75.625**

3.4.3. Machine Learning Benchmark Datasets

In order to further demonstrate the desirable robustness of the proposed RMEE in a wider range of robust classification tasks, besides the toy examples and EEG datasets above, then some popular benchmark datasets are selected from the UCI repository [164] and are contaminated artificially with attribute outliers. The selected datasets are summarized in Table 5. For binary classification, the multi-class datasets are transformed into several 2-class datasets. To make this transformation, a new dataset is built that consists of the samples of one specific class, and the antagonistic label is assigned to the other samples. Thus, a dataset of m classes is converted into m datasets of binary class, which is known as *one vs all*. This helps analyze whether the classifier could extract effective pattern for each class. For the benchmark datasets, 2/3 samples are randomly selected for training, and the other 1/3 samples act as testing samples. Similarly, the ELM model is used under different learning criteria.

Table 5. Benchmark datasets summary.

No.	Dataset	Feature	Class Ratio
1	*Statlog (Australian Credit Approval)*	14	383 : 307
2	*Balance Scale (l. vs all)*	4	337 : 288
3	*Balance Scale (r. vs all)*	4	337 : 288
4	*BUPA Liver Disorders*	6	200 : 145
5	*Connectionist (Sonar, Mines vs. Rocks)*	60	111 : 97
6	*Iris (set. vs all)*	4	100 : 50
7	*Iris (vir. vs all)*	4	100 : 50
8	*Breast Cancer Wisconsin (Original)*	9	458 : 241
9	*Breast Cancer Wisconsin (Diagnostic)*	30	357 : 212
10	*Wholesale Customers*	7	298 : 142

First, each dimension is normalized to zero-mean and unit-variance, so that the diagonal elements of the covariance matrix of the training samples are all 1. Then, the corruption model of Eq.(59) is used

similarly to contaminate the attribute values of training samples. θ is considered with 0, 20%, and 40%, Σ_{small} is set as $0.05I_d$, and Σ_{large} is assessed with $20I_d$, $50I_d$, $100I_d$, $300I_d$, and $1000I_d$, respectively. The average classification accuracy is listed in Table 6. Similarly, the highest accuracy in each condition is marked in bold. One can observe as before that RMEE achieves the highest accuracy in most cases.

Table 6. Average classification accuracy of benchmark datasets contaminated by attribute outliers.

Dataset1	0%	$20I_d$		$50I_d$		$100I_d$		$300I_d$		$1000I_d$	
		20%	40%	20%	40%	20%	40%	20%	40%	20%	40%
CE	85.34	84.47	83.55	84.30	83.29	84.51	81.17	83.26	73.22	79.37	60.62
C-Loss	86.09	85.96	84.57	**85.79**	84.89	**85.97**	84.37	85.02	82.26	84.50	77.21
QMEE	85.19	81.53	77.47	80.38	75.15	80.80	73.20	75.58	70.89	75.37	65.32
RMEE	**86.90**	**86.49**	**85.84**	85.71	**85.07**	85.85	**85.03**	**86.04**	**84.07**	**85.40**	**77.81**

Dataset2	0%	$20I_d$		$50I_d$		$100I_d$		$300I_d$		$1000I_d$	
		20%	40%	20%	40%	20%	40%	20%	40%	20%	40%
CE	**94.69**	91.88	87.10	90.29	83.45	88.95	76.58	81.79	63.40	66.69	55.49
C-Loss	94.21	94.00	92.49	93.64	91.12	93.83	88.85	91.94	82.96	87.31	67.05
QMEE	93.81	90.01	77.54	86.68	74.09	84.04	71.51	80.33	66.70	75.63	60.87
RMEE	94.21	**94.98**	**92.84**	**94.51**	**92.00**	**94.60**	**89.55**	**93.58**	**83.33**	**90.07**	**67.93**

Dataset3	0%	$20I_d$		$50I_d$		$100I_d$		$300I_d$		$1000I_d$	
		20%	40%	20%	40%	20%	40%	20%	40%	20%	40%
CE	94.93	92.02	86.81	90.85	82.44	89.13	75.14	81.13	63.26	65.92	55.15
C-Loss	95.03	93.75	92.36	93.32	91.04	93.40	89.72	92.12	80.73	88.55	65.47
QMEE	94.38	93.00	81.92	90.97	78.25	87.61	76.77	83.10	70.60	80.63	60.83
RMEE	**95.12**	**94.70**	**92.98**	**94.06**	**91.96**	**93.88**	**90.12**	**93.38**	**81.18**	**89.65**	**65.76**

Dataset4	0%	$20I_d$		$50I_d$		$100I_d$		$300I_d$		$1000I_d$	
		20%	40%	20%	40%	20%	40%	20%	40%	20%	40%
CE	72.73	64.27	59.37	62.11	57.96	59.63	57.72	57.54	57.63	57.65	**57.71**
C-Loss	72.50	67.21	59.90	65.03	58.77	61.78	58.03	**59.16**	57.04	**58.12**	57.35
QMEE	68.10	63.71	56.82	61.79	56.83	59.21	53.63	58.77	53.46	54.64	52.53
RMEE	**72.86**	**67.74**	**61.30**	**65.23**	**59.76**	**62.18**	**58.56**	58.88	**57.70**	58.10	57.61

Dataset5	0%	$20I_d$		$50I_d$		$100I_d$		$300I_d$		$1000I_d$	
		20%	40%	20%	40%	20%	40%	20%	40%	20%	40%
CE	77.03	72.84	70.99	72.20	70.97	72.80	69.35	70.41	68.01	70.80	65.33
C-Loss	**77.75**	74.17	**74.62**	74.28	74.20	74.23	73.90	72.61	**72.54**	**72.99**	69.30
QMEE	77.38	70.16	69.17	70.93	68.61	71.84	68.09	70.62	66.70	67.77	65.12
RMEE	77.58	**75.88**	74.32	**75.77**	**74.81**	**75.48**	**74.33**	**73.65**	71.64	72.12	**70.39**

Dataset6	0%	$20I_d$		$50I_d$		$100I_d$		$300I_d$		$1000I_d$	
		20%	40%	20%	40%	20%	40%	20%	40%	20%	40%
CE	**99.78**	**99.62**	**99.44**	**99.78**	97.12	**99.76**	91.86	**97.76**	73.32	83.24	67.96
C-Loss	99.52	99.02	97.06	98.24	**97.78**	97.06	**97.44**	97.70	93.96	**96.64**	78.18
QMEE	99.06	96.72	86.30	95.70	85.34	93.94	84.94	89.32	83.94	89.42	**79.58**
RMEE	99.04	97.64	96.12	97.96	95.70	97.28	96.06	97.62	**95.24**	94.68	79.40

Dataset7	0%	$20I_d$		$50I_d$		$100I_d$		$300I_d$		$1000I_d$	
		20%	40%	20%	40%	20%	40%	20%	40%	20%	40%
CE	**96.24**	92.16	88.86	90.44	83.66	89.78	77.24	83.26	70.08	72.40	67.66
C-Loss	94.70	92.66	90.50	91.12	88.02	91.54	85.28	90.48	80.40	86.14	**73.40**
QMEE	95.00	90.98	84.08	90.66	82.54	88.58	80.64	85.72	76.36	80.58	70.10
RMEE	95.04	**93.18**	**91.48**	**91.96**	**88.28**	**91.94**	**85.70**	**91.40**	**80.54**	**88.68**	71.20

Dataset8	0%	$20I_d$		$50I_d$		$100I_d$		$300I_d$		$1000I_d$	
		20%	40%	20%	40%	20%	40%	20%	40%	20%	40%
CE	**96.31**	**95.98**	**95.12**	95.12	**93.23**	94.73	90.14	93.55	78.84	84.47	65.91
C-Loss	95.50	94.50	94.05	94.83	92.81	94.09	92.26	93.26	91.49	91.67	**83.91**
QMEE	95.54	93.69	92.72	91.46	87.68	88.20	81.39	85.70	76.95	80.99	73.37
RMEE	95.75	95.85	94.77	**95.16**	92.34	**95.33**	**92.45**	**94.74**	**92.11**	**93.25**	83.39

Dataset9	0%	$20I_d$		$50I_d$		$100I_d$		$300I_d$		$1000I_d$	
		20%	40%	20%	40%	20%	40%	20%	40%	20%	40%
CE	96.31	**95.49**	94.38	95.76	93.84	94.57	92.59	93.17	89.35	89.81	78.04
C-Loss	95.89	95.22	93.79	**95.83**	94.63	95.07	94.32	94.27	92.34	93.86	90.38
QMEE	95.56	92.06	90.13	91.94	90.40	88.32	85.92	85.40	75.70	80.13	73.73
RMEE	**96.52**	95.28	**94.73**	95.50	**94.96**	**95.41**	**94.48**	**94.66**	**93.87**	**94.75**	**91.05**

Dataset10	0%	$20I_d$		$50I_d$		$100I_d$		$300I_d$		$1000I_d$	
		20%	40%	20%	40%	20%	40%	20%	40%	20%	40%
CE	**90.68**	86.93	81.14	84.90	76.01	81.65	71.87	75.54	68.67	69.27	67.77
C-Loss	90.21	87.30	82.01	86.06	**82.50**	**85.71**	80.71	**83.59**	72.93	77.14	68.21
QMEE	89.99	78.47	78.27	77.90	73.16	77.88	70.52	77.45	65.42	71.45	61.16
RMEE	89.92	**87.47**	**82.73**	**86.54**	80.07	85.37	**80.90**	83.20	**73.03**	**78.05**	**68.46**

3.5. ***Discussion***

The purpose of this study is to achieve a decent realization of MEE for robust classification. QMEE is regarded as a simplified execution that employs quantization technique to reduce computational complexity. Nevertheless, as demonstrated in experimental results, QMEE fails to achieve expected robustness. Particularly, in considerable situations of Fig. 11, and Table 6 QMEE even acquires worse performance than conventional CE, which indicates that robust classification cannot be realized by directly utilizing QMEE. The proposed RMEE in the present study is a special case of QMEE with the predetermined codebook $C = (0, -1, 1)$, which could optimize the error PDF $p(e)$ towards the optimal distribution $\rho(e)$. Although QMEE exhibits a more generalized form, it would like to be stated it is the larger generalization that deteriorates QMEE in noisy classification task. Proved by the extensive experimental results, RMEE achieves obviously better performance than QMEE in noisy classification which demonstrates superiority of the proposed restriction.

To further emphasize effectiveness and competence of the predetermined codebook $C = (0, -1, 1)$ compared to a data-driven one, the scenario from Section 3.4.1 is employed to investigate the final training error distributions of QMEE and RMEE, respectively, with attribute contamination $200I_d$ and outlier proportion 0.5. The average histograms of the resultant training errors with 100 repetitive trials are illustrated in Fig. 14. One can observe that the average error distribution achieved by QMEE is in a considerably chaotic situation. To ascertain the occurrence, the 100 resultant error distributions are categorized and plotted with the representative averages in the right half of Fig. 14. As one can find, QMEE does occasionally realize the optimal three-peak distribution. To be specific, QMEE achieved the expected consequence in 37 out of 100 repetitions. Nevertheless, in the remaining cases, it realized miscellaneous double-peak error distribution. The essential reason is stated as before: such double-peak error distributions all achieve minimum entropy. Consequently, QMEE cannot guarantee the tendency of the learning results, which engenders inferior robustness in classification. By comparison, the proposed RMEE realized the expected three-peak error distribution with minor fluctuations, which highlights the necessity and validity of the proposed restriction.

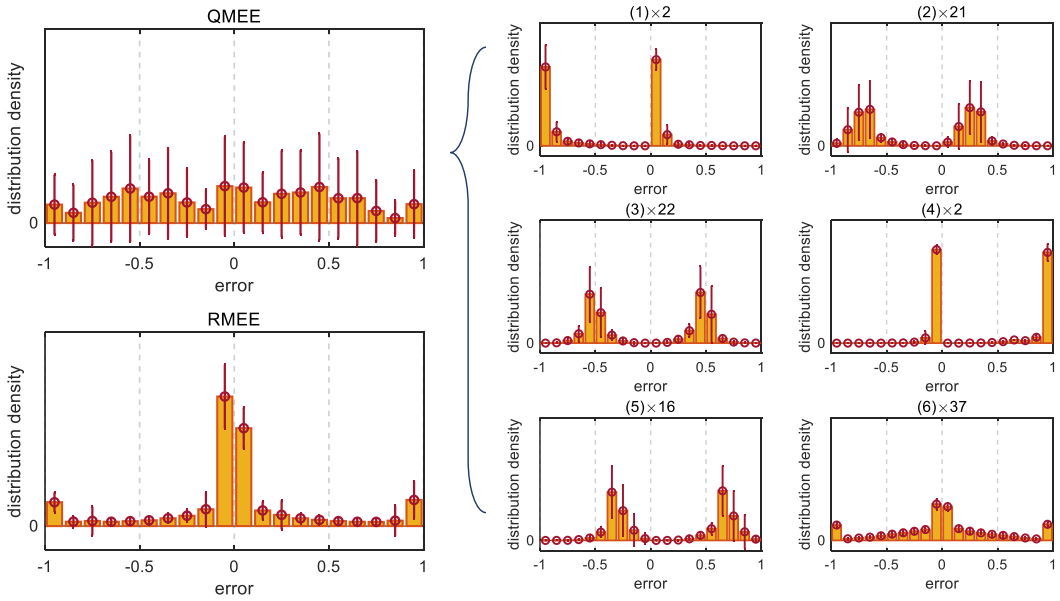

Figure 14. Average distribution histograms of the training errors. Upper left: error distribution achieved by QMEE averaged by 100 trials; Lower left: error distribution achieved by RMEE averaged by 100 trials; Right half: six representative error distributions of QMEE among the 100 trials. The average distributions and the corresponding standard deviations are calculated by manual categorization, and the numbers on each subgraph denote the times that the corresponding situation has occurred.

Considering the determination of Φ for RMEE, as described in Section 3.3.3, RMEE actually uses C-Loss initially to estimate the outlier proportion. To evaluate the exactness, the differences between the estimated outlier proportions and true values are illustrated with three scenarios from Section 3.4.1 with attribute contamination $20I_d$, $50I_d$, and $200I_d$, respectively, in Fig. 15, based on 100 Monte-Carlo repetitions. One observes that the respective curves of estimations and true values exhibit consistency with relative precision.

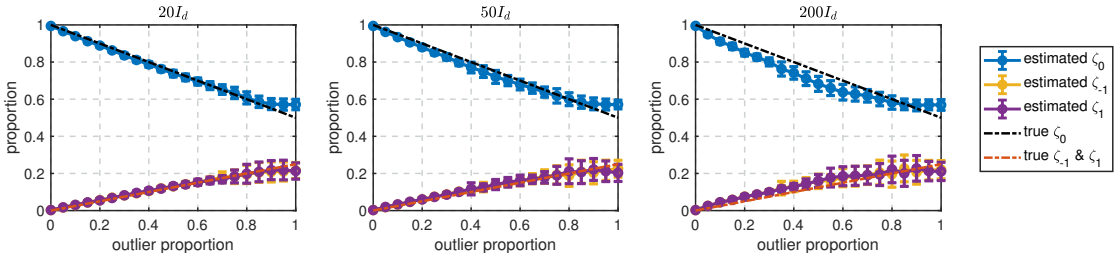

Figure 15. Estimated proportions of respective sample categories and true values. The true values are expectations by calculation according to the contamination method. Note the lateral axis does not denote the actual outlier proportion, because half of the added outliers are innocuous according to the distribution. In the remaining adverse ones, the numbers of *FN* outliers and *FP* outliers are supposed to be equivalent due to the distribution symmetry of the toy dataset and outliers.

4. Partial Maximum Correntropy Regression

From this section, this thesis begins to concentrate on the high-dimensional brain activity decoding tasks, where the purpose is to solve the issues of "robustness" and "high-dimensional" simultaneously by embedding robust ITL methods in existing brain decoding algorithms that can effectively solve the high-dimensional problem. The first strategy to solve the high-dimensional issue is to project the high-dimensional variables to a low-dimensional subspace while retaining as much information as possible in the original space, i.e. the dimensionality reduction technique. The most famous algorithm for this

strategy is the principal component analysis (PCA) algorithm, as introduced in Eq.(12). PCA employs a linear projection on the dataset with maximizing the variance of the data after dimensionality reduction, where a larger variance is considered to contain more information. The objective function of PCA can be also interpreted to minimize the second order of the reconstruction error in the original space. Due to the non-robust property of the conventional variance or second-order statistics, the original PCA algorithm can be significantly deteriorated by non-Gaussian noises or outliers. To address this issue, MCC and MEE have been utilized to reformulate the PCA algorithm for a more robust implementation [85,157,165].

However, PCA algorithm was originally developed as a dimensionality reduction method for the unsupervised learning scenario, which means there is no target to supervise the model parameter. As one can see in Eq.(12), the model parameter for PCA is only learned from the covariate matrix. Despite the successful application of PCA for supervised machine learning, such as the principal component regression (PCR) [166], PCA may be not entirely appropriate for the supervised brain activity decoding task, since the dimensionality reduction in PCA method does not take the target variable into account. As a result, although the latent representations in the low-dimensional space acquired by PCA could retain the maximal information from the original covariate matrix, the most relevant information in the covariate data to the decoding task, i.e. the target variable, may be ignored and lost in the process of dimensionality reduction.

For a better dimensionality reduction technique which can consider the covariate data and target variable simultaneously, the partial least square (PLS) approach was proposed. Although this PLS approach was initially developed for econometrics and chemometrics [167], it has emerged as a popular method for neural imaging and decoding [98,168]. There are two major applications attainable by the PLS approach. First, PLS can be utilized to analyze the correlation relationship between two arbitrary random variables by projecting them to the same low-dimensional subspace and then computing the shared information between them. The second implementation of PLS is for predicting a continuous variable, i.e. regression. PLS-based regression, called PLSR, has been successfully applied to numerous brain activity decoding tasks. In particular, PLSR is a popular method to accomplish the inter-correlated and potentially high-dimensional ECoG decoding tasks to predict continuous variables from ECoG signals as well as its various improved versions in the last decade [27,28,169–175]. For example, [27] successfully predicted the three-dimensional continuous hand trajectories for two monkeys during asynchronous food-reaching tasks from time–frequency features of subdural ECoG signals by PLSR algorithm. They further showed the admirable prediction capability of PLSR in an epidural ECoG study [28]. Recently, different strategies have been investigated to improve the decoding performance of PLSR. For example, multi-way PLSR algorithms have been proposed as a generalization for tensor analysis in the ECoG decoding tasks [28,172,174,176]. Moreover, regularization technique has been used to penalize the objective function with an extra regularization term to achieve desirable prediction [170,173,175].

PLSR solves a regression problem primarily with dimensionality reduction on both explanatory matrix (input) and response matrix (output), in which the dimensionality-reduced samples (commonly called as *latent variables*) for respective sets exhibit maximal correlation, thus structuring association from input variable to output variable. However, the conventional PLSR algorithm and most existing variants are in essence formulated by the least square criterion, which assigns superfluous importance to the deviated noises. On the other hand, although ECoG signal usually exhibits a relatively higher signal-to-noise ratio (SNR) than the non-invasive EEG recording, previous studies have revealed that ECoG is also prone to be contaminated by physiological artifacts with pronounced amplitudes [52,177]. As a result, PLSR may be incompetent for noisy ECoG decoding tasks due to subnormal robustness.

This section aims to investigate a robust implementation for PLSR by reformulating it with ITL method. Since MCC has a simpler formulation than MEE, this section first considers how to use MCC to propose a new robust version for PLSR. Recently, a rudimentary implementation of MCC for PLSR has been investigated in [178], where MCC was employed in the process of dimensionality reduction.

However, the proposed algorithm in [178] is limited in some respects. First, except for the MCC-based dimensionality reduction, it remains acquiring the regression relations under the least square criterion. Second, it only considers the dimensionality reduction for the explanatory matrix. Consequently, one has to calculate the regression coefficients separately for each dimension of the response matrix, which means it could be inadequate for multivariate response prediction.

By comparison, this section aims to realize a more comprehensive implementation of the MCC framework in PLSR. The main contributions of this section are summarized as follows. First, the PLSR algorithm is reformulated thoroughly with the MCC framework, that not only the dimensionality reduction, but also the regression relations between the different variables are established by the MCC framework. Second, both the explanatory matrix (input) and the response matrix (output) are treated with MCC-based dimensionality reduction. As a result, the proposed algorithm is adequate for multivariate response prediction. Finally, for each reconstruction error and prediction error, a Gaussian kernel function with an individual kernel bandwidth is utilized, where each kernel bandwidth could be calculated from the corresponding set of errors directly.

4.1. ***Partial Least Square Regression***

4.1.1. Conventional Partial Least Square Regression

First, a brief introduction for the conventional PLSR algorithm is given in what follows. Because PLSR can predict multivariate target variable, in this section, one can assume that each observation for the target is a L-dimensional vector, denoted by $\mathbf{t} = (t_1, t_2, \cdots, t_L) \in \mathbb{R}^{1\times L}$, and the matrix for all the response variable with N observations can be expressed by $\mathbf{T} = (\mathbf{t}_1^T, \mathbf{t}_2^T, \cdots, \mathbf{t}_N^T)^T \in \mathbb{R}^{N\times L}$, each row of which is an individual L-dimensional target observation. Now, one can consider a dataset associated with the explanatory matrix $\mathbf{X} \in \mathbb{R}^{N\times D}$ and the response matrix $\mathbf{T} \in \mathbb{R}^{N\times L}$ with N observations, in which D and L denote the respective numbers of dimension for explanatory and response, respectively. PLSR is an iterative regression algorithm which executes dimensionality reduction on explanatory and response matrices simultaneously, so that the resultant latent variables in each iteration exhibit maximal covariance. For the first iteration, the original matrices are employed as the current residual matrices, i.e. $\mathbf{X}_1 = \mathbf{X}$ and $\mathbf{T}_1 = \mathbf{T}$. PLSR calculates two projectors $\mathbf{w}_1 \in \mathbb{R}^D$ and $\mathbf{c}_1 \in \mathbb{R}^L$ to acquire the corresponding latent variables, denoted as $\mathbf{r}_1 = \mathbf{X}_1\mathbf{w}_1$ and $\mathbf{u}_1 = \mathbf{T}_1\mathbf{c}_1$, by maximizing their covariance

$$\max_{\|\mathbf{w}_1\|_2=\|\mathbf{c}_1\|_2=1} \mathbf{r}_1^T\mathbf{u}_1 = \mathbf{w}_1^T\mathbf{X}_1^T\mathbf{T}_1\mathbf{c}_1 \tag{60}$$

This equation can be effectively solved by the singular value decomposition (SVD) on $\mathbf{X}_1^T\mathbf{T}_1$. Then, one computes the loading vector $\mathbf{p}_1$ on $\mathbf{X}_1$ by the least square criterion as

$$\min_{\mathbf{p}_1} \|\mathbf{X}_1 - \mathbf{r}_1\mathbf{p}_1^T\|_2^2 \;\Rightarrow\; \mathbf{p}_1 = \mathbf{X}_1^T\mathbf{r}_1/(\mathbf{r}_1^T\mathbf{r}_1) \tag{61}$$

thus organizing the regression relation from $\mathbf{r}_1$ to $\mathbf{X}_1$. PLSR also supposes a linear association from $\mathbf{r}_1$ to $\mathbf{u}_1$ by calculating a regression scalar b_1 by the least square criterion as

$$\min_{b_1} \|\mathbf{u}_1 - \mathbf{r}_1 b_1\|_2^2 \;\Rightarrow\; b_1 = \mathbf{u}_1^T\mathbf{r}_1/(\mathbf{r}_1^T\mathbf{r}_1) \tag{62}$$

The residual matrices for the next iteration are updated by

$$\mathbf{X}_2 = \mathbf{X}_1 - \mathbf{r}_1\mathbf{p}_1^T,\; \mathbf{T}_2 = \mathbf{T}_1 - b_1\mathbf{r}_1\mathbf{c}_1^T \tag{63}$$

Such procedures are repeated by PLSR for the optimal number of factors S, which is usually selected by cross validation. One can then collect the outcomes from each iteration, i.e. $\mathbf{R} = [\mathbf{r}_1, .., \mathbf{r}_S] \in \mathbb{R}^{N\times S}$,

$\mathbf{P} = [\mathbf{p}_1, .., \mathbf{p}_S] \in \mathbb{R}^{D\times S}$, $\mathbf{B} = \text{diag}(b_1, .., b_S) \in \mathbb{R}^{S\times S}$, $\mathbf{C} = [\mathbf{c}_1, .., \mathbf{c}_S] \in \mathbb{R}^{L\times S}$. As a result, one rewrites the decomposition of $\mathbf{X}$ and the predicted response $\hat{\mathbf{T}}$ as

$$\mathbf{X} = \mathbf{R}\mathbf{P}^T, \ \hat{\mathbf{T}} = \mathbf{R}\mathbf{B}\mathbf{C}^T \tag{64}$$

Thus, the prediction from $\mathbf{X}$ to $\hat{\mathbf{T}}$ is structured as

$$\hat{\mathbf{T}} = \mathbf{X}\mathbf{H} \tag{65}$$

in which $\mathbf{H} = \mathbf{P}^{T+}\mathbf{B}\mathbf{C}^T \in \mathbb{R}^{D\times L}$, and $\mathbf{P}^{T+}$ is the pseudo-inverse of $\mathbf{P}^T$.

Maximizing the covariance between the latent variables in Eq.(60) could be rewritten as [179]

$$\min_{\|\mathbf{w}_s\|_2=\|\mathbf{c}_s\|_2=1} \sum_{n=1}^{N} \begin{pmatrix} \|\mathbf{x}_s^n - \mathbf{x}_s^n \mathbf{w}_s \mathbf{w}_s^T\|^2 \\ +\|\mathbf{t}_s^n - \mathbf{t}_s^n \mathbf{c}_s \mathbf{c}_s^T\|^2 \\ +\|\mathbf{x}_s^n \mathbf{w}_s - \mathbf{t}_s^n \mathbf{c}_s\|^2 \end{pmatrix} \tag{66}$$

where the subscript s denotes the s-th decomposition factor, while $\mathbf{x}_s^n$ and $\mathbf{t}_s^n$ denote the n-th samples in the residual matrices $\mathbf{X}_s$ and $\mathbf{T}_s$, respectively. One can observe from Eq.(66) that, PLSR adopts the least square criterion not only to obtain the regression relations in Eq.(61-62), but also for the projectors $\mathbf{w}_s$ and $\mathbf{c}_s$. The connotation of Eq.(66) is interpreted as follows. The first and second terms in summation are the quadratic reconstruction errors for input and output, respectively. The third term denotes the quadratic prediction error for the n-th latent variables. Because each step for PLSR is based on the least square criterion, the prediction from input to output may be deteriorated seriously by noises.

4.1.2. Regularized Partial Least Square Regression

The regularization techniques have been widely utilized to ameliorate the decoding performance of the PLSR algorithm. For example, L_1-regularization on the projectors was employed so as to obtain sparse projectors, conducting the feature selection simultaneously [170]. The authors further extended their study in [173], in which Sobolev-norm and polynomial penalization were introduced into PLSR to strengthen the smoothness of the predicted response. Recently, the state-of-the-art regularized PLSR was proposed by utilizing L_2-regularization to find the regression relation between the latent variables $\mathbf{r}_s$ and $\mathbf{u}_s$, so as to reduce the over-fitting risk of each latent variable on the desired response [175]. In particular, for each decomposition factor, the scalar b_s is acquired with an individual regularization parameter λ_s as

$$\min_{b_s} \|\mathbf{u}_s - \mathbf{r}_s b_s\|_2^2 + \lambda_s b_s^2 \ \Rightarrow \ b_s = \mathbf{u}_s^T \mathbf{r}_s / (\mathbf{r}_s^T \mathbf{r}_s + \lambda_s) \tag{67}$$

Experimental results in [175] showed that the regularization method in Eq.(67) achieved better ECoG decoding performance than regularizing the projectors.

Nevertheless, the regularized PLSR variants are still formulated from the non-robust least square criterion, as a result, remaining prone to suffering the performance deterioration caused by the adverse noises.

4.1.3. Partial Least Square Regression with MCC

Recently, a rudimentary MCC-based PLSR variant has been investigated in [178], named as MCC-PLSR. For a univariate output, according to [178], the dimensionality reduction of Eq.(60) is equal to its quadratic form

$$\max_{\|\mathbf{w}_s\|_2=1} \mathbf{w}_s^T \mathbf{X}_s^T \mathbf{T}_s \mathbf{T}_s^T \mathbf{X}_s \mathbf{w}_s \tag{68}$$

which aims to maximize the quadratic covariance. [178] utilized a similar proposition as in the MCC-based principal component analysis [157], proposing the following objective function

$$\max_{\|\mathbf{w}_s\|_2=1} \sum_{n=1}^{N} k_h(\sqrt{\mathbf{t}_s^{nT}\mathbf{x}_s^n\mathbf{x}_s^{nT}\mathbf{t}_s^n - \mathbf{t}_s^{nT}\mathbf{x}_s^n\mathbf{w}_s\mathbf{w}_s^T\mathbf{x}_s^{nT}\mathbf{t}_s^n}) \tag{69}$$

from which one can calculate a robust projector $\mathbf{w}_s$. Then, one obtains the latent variables by $\mathbf{r}_s = \mathbf{X}_s\mathbf{w}_s$, and acquires other model parameters similarly as in Eq.(61-65).

Despite the robust implementation of the projector $\mathbf{w}_s$ in Eq.(69), the above MCC-PLSR algorithm could be inadequate for the following reasons. First, except for the calculation of $\mathbf{w}_s$, the other model parameters are still acquired under the least square criterion. Second, dimensionality reduction is not considered for the output matrix. As a result, the prediction performance for a multivariate response may be limited.

4.2. *Partial Maximum Correntropy Regression*

In what follows, a comprehensive reformulation of PLSR with the MCC framework is presented. Compared with the existing MCC-PLSR, the proposed method aims to acquire each model parameter by MCC. In addition, the generalization for multivariate response is also taken into account here. The detailed mathematical derivations of the proposed method are given as follows, where the subscript s denoting the s-th decomposition factor is omitted for the purpose of simplicity.

Substituting the least quadratic reconstruction errors and prediction errors in the conventional PLSR of Eq.(66) with the maximum correntropy yields

$$\max_{\|\mathbf{w}\|_2=\|\mathbf{c}\|_2=1} \sum_{n=1}^{N} \begin{pmatrix} k_{h_x}(\mathbf{x}_n - \mathbf{x}_n\mathbf{w}\mathbf{w}^T) \\ +k_{h_t}(\mathbf{t}_n - \mathbf{t}_n\mathbf{c}\mathbf{c}^T) \\ +k_{h_r}(\mathbf{x}_n\mathbf{w} - \mathbf{t}_n\mathbf{c}) \end{pmatrix} \tag{70}$$

where h_x, h_t, and h_r are the Gaussian kernel bandwidths for **X**-reconstruction errors, **T**-reconstruction errors, and the prediction errors, respectively.

Then one transforms the vectors $(\mathbf{x}_n - \mathbf{x}_n\mathbf{w}\mathbf{w}^T)$ and $(\mathbf{t}_n - \mathbf{t}_n\mathbf{c}\mathbf{c}^T)$ into scalars since two projectors $\mathbf{w}$ and $\mathbf{c}$ are unit-length vectors, i.e. $\mathbf{w}^T\mathbf{w} = \mathbf{c}^T\mathbf{c} = 1$,

$$\begin{aligned} \sqrt{\|\mathbf{x}_n - \mathbf{x}_n\mathbf{w}\mathbf{w}^T\|^2} &= \sqrt{\mathbf{x}_n\mathbf{x}_n^T - \mathbf{x}_n\mathbf{w}\mathbf{w}^T\mathbf{x}_n^T} \\ \sqrt{\|\mathbf{t}_n - \mathbf{t}_n\mathbf{c}\mathbf{c}^T\|^2} &= \sqrt{\mathbf{t}_n\mathbf{t}_n^T - \mathbf{t}_n\mathbf{c}\mathbf{c}^T\mathbf{t}_n^T} \end{aligned} \tag{71}$$

Subsequently, one obtains the following optimization problem to acquire the projectors

$$\max_{\|\mathbf{w}\|_2=\|\mathbf{c}\|_2=1} \sum_{n=1}^{N} \begin{pmatrix} k_{h_x}(\sqrt{\mathbf{x}_n\mathbf{x}_n^T - \mathbf{x}_n\mathbf{w}\mathbf{w}^T\mathbf{x}_n^T}) \\ +k_{h_t}(\sqrt{\mathbf{t}_n\mathbf{t}_n^T - \mathbf{t}_n\mathbf{c}\mathbf{c}^T\mathbf{t}_n^T}) \\ +k_{h_r}(\mathbf{x}_n\mathbf{w} - \mathbf{t}_n\mathbf{c}) \end{pmatrix} \tag{72}$$

After obtaining $\mathbf{w}$ and $\mathbf{c}$, one could calculate the latent variables as in the conventional PLSR by $\mathbf{r} = \mathbf{X}\mathbf{w}$ and $\mathbf{u} = \mathbf{T}\mathbf{c}$. Then, to calculate the loading vector $\mathbf{p}$ and the regression coefficient b, one could also adopt the MCC objective function by

$$\max_{\mathbf{p}} \sum_{n=1}^{N} k_{h_p}(\mathbf{x}_n - \mathbf{r}_n\mathbf{p}^T) \tag{73}$$

$$\max_{b} \sum_{n=1}^{N} k_{h_b}(\mathbf{u}_n - \mathbf{r}_n b) \tag{74}$$

in which $\mathbf{r}_n$ and $\mathbf{u}_n$ denote the n-th elements for the latent variables $\mathbf{r}$ and $\mathbf{u}$, respectively. h_p and h_b are the corresponding kernel bandwidths. The residual matrices are then updated similarly as PLSR.

One repeats such procedures for the optimal number of factors and collects the acquired vectors from each iteration to organize the matrices $\mathbf{R}$, $\mathbf{P}$, $\mathbf{B}$, and $\mathbf{C}$, as in the original PLSR. Finally, the predicted response $\hat{\mathbf{T}}$ can be obtained from $\mathbf{X}$ by the regression relationship of Eq.(65). The above-mentioned PLSR which is comprehensively reformulated based on the MCC, is named as *partial maximum correntropy regression* (PMCR). Then the proposed PMCR algorithm is discussed with its optimization, convergence analysis, and determination of hyper-parameters.

4.2.1. Optimization

Three optimization problems in Eq.(72-74) need to be addressed in PMCR. One can first consider Eq.(72) for the calculation of the projectors $\mathbf{w}$ and $\mathbf{c}$. Based on the HQ optimization method as described in Theorem 2, Eq.(72) could be rewritten as

$$\max_{\|\mathbf{w}\|_2=\|\mathbf{c}\|_2=1} \sum_{n=1}^{N} \begin{pmatrix} \sup\{\alpha_n \frac{\mathbf{x}_n\mathbf{x}_n^T-\mathbf{x}_n\mathbf{w}\mathbf{w}^T\mathbf{x}_n^T}{2h_x} - g(\alpha_n)\} \\ +\sup\{\beta_n \frac{\mathbf{t}_n\mathbf{t}_n^T-\mathbf{t}_n\mathbf{c}\mathbf{c}^T\mathbf{t}_n^T}{2h_t} - g(\beta_n)\} \\ +\sup\{\gamma_n \frac{(\mathbf{x}_n\mathbf{w}-\mathbf{t}_n\mathbf{c})^2}{2h_r} - g(\gamma_n)\} \end{pmatrix} \tag{75}$$

where $g(\cdot)$ is the conjugate function as described in Theorem 2. $\{\alpha_n\}_{n=1}^N$, $\{\beta_n\}_{n=1}^N$, and $\{\gamma_n\}_{n=1}^N$ denote three sets of introduced auxiliary variables, respectively. Thus, one can update $(\alpha_n, \beta_n, \gamma_n)$ and $(\mathbf{w}, \mathbf{c})$ alternately to optimize Eq.(72) by

$$\max_{\|\mathbf{w}\|_2=\|\mathbf{c}\|_2=1,\alpha_n,\beta_n,\gamma_n} J \triangleq \sum_{n=1}^{N} \begin{pmatrix} \alpha_n \frac{\mathbf{x}_n\mathbf{x}_n^T-\mathbf{x}_n\mathbf{w}\mathbf{w}^T\mathbf{x}_n^T}{2h_x} - g(\alpha_n) \\ +\beta_n \frac{\mathbf{t}_n\mathbf{t}_n^T-\mathbf{t}_n\mathbf{c}\mathbf{c}^T\mathbf{t}_n^T}{2h_t} - g(\beta_n) \\ +\gamma_n \frac{(\mathbf{x}_n\mathbf{w}-\mathbf{t}_n\mathbf{c})^2}{2h_r} - g(\gamma_n) \end{pmatrix} \tag{76}$$

Since the HQ optimization is an iterative process, one can denote the k-th HQ iteration with the subscript k. First, one can update the auxiliary variables with the current projectors $(\mathbf{w}_k, \mathbf{c}_k)$ by

$$\begin{aligned} \alpha_{n,k+1} &= -\exp(-\frac{\mathbf{x}_n\mathbf{x}_n^T - \mathbf{x}_n\mathbf{w}\mathbf{w}^T\mathbf{x}_n^T}{2h_x}) \\ \beta_{n,k+1} &= -\exp(-\frac{\mathbf{t}_n\mathbf{t}_n^T - \mathbf{t}_n\mathbf{c}\mathbf{c}^T\mathbf{t}_n^T}{2h_t}) \\ \gamma_{n,k+1} &= -\exp(-\frac{(\mathbf{x}_n\mathbf{w} - \mathbf{t}_n\mathbf{c})^2}{2h_r}) \\ (n &= 1, .., N) \end{aligned} \tag{77}$$

Then, to optimize the projectors, one rewrites Eq.(76) by collecting the terms of projectors and omitting the auxiliary variables as

$$\max_{\|\mathbf{w}\|_2=\|\mathbf{c}\|_2=1} J_p \triangleq \sum_{n=1}^{N} \begin{pmatrix} (\frac{\gamma_n}{2h_r} - \frac{\alpha_n}{2h_x})\mathbf{x}_n\mathbf{w}\mathbf{w}^T\mathbf{x}_n^T \\ +(\frac{\gamma_n}{2h_r} - \frac{\beta_n}{2h_t})\mathbf{t}_n\mathbf{c}\mathbf{c}^T\mathbf{t}_n^T \\ -\frac{\gamma_n}{h_r}\mathbf{x}_n\mathbf{w}\mathbf{c}^T\mathbf{t}_n^T \end{pmatrix} \tag{78}$$

which is a quadratic optimization issue constrained by nonlinear condition. To optimize Eq.(78), there exist enormous solutions in the literature, such as the sequential quadratic programming (SQP) which has been widely utilized for nonlinear programming problems [180].

After one obtains the projectors $\mathbf{w}$ and $\mathbf{c}$, the latent variables are computed by $\mathbf{r} = \mathbf{X}\mathbf{w}$ and $\mathbf{u} = \mathbf{T}\mathbf{c}$. Then, Eq.(73-74) can be solved by the following iterative fixed-point optimization method with fast convergence [81]

$$\mathbf{p} = \mathbf{X}^T\Psi_{\mathbf{p}}\mathbf{r}/(\mathbf{r}^T\Psi_{\mathbf{p}}\mathbf{r}) \tag{79}$$

$$b = \mathbf{u}^T\Psi_b\mathbf{r}/(\mathbf{r}^T\Psi_b\mathbf{r}) \tag{80}$$

where $\Psi_{\mathbf{p}}$ and Ψ_b are $N \times N$ diagonal matrices with the diagonal elements $(\Psi_{\mathbf{p}})_{nn} = k_{h_p}(\mathbf{x}_n - \mathbf{r}_n\mathbf{p}^T)$ and $(\Psi_b)_{nn} = k_{h_b}(\mathbf{u}_n - \mathbf{r}_n b)$, respectively. Since $\Psi_{\mathbf{p}}$ and Ψ_b are dependent on the current solutions $\mathbf{p}$ and b, the updates in Eq.(79-80) are fixed-point equations. The comprehensive procedures for PMCR are summarized in Algorithm 3.

Algorithm 3 Partial maximum correntropy regression

input:
  explanatory matrix $\mathbf{X}$ and response matrix $\mathbf{T}$;
  number of decomposition factors S;
initialize:
  $\mathbf{X}_1 = \mathbf{X}$ and $\mathbf{Y}_1 = \mathbf{Y}$;
output:
  prediction model $\hat{\mathbf{T}} = \mathbf{X}\mathbf{H}$;
for $s = 1, 2, .., S$ **do**
  initialize the projectors by the conventional PLSR;
  repeat
    auxiliary-step: update $(\alpha_n, \beta_n, \gamma_n)$ with Eq.(77);
    projector-step: update $(\mathbf{w}_s, \mathbf{c}_s)$ with Eq.(78);
  until the parameter change is small enough
  compute latent variables $\mathbf{r}_s = \mathbf{X}_s\mathbf{w}_s$ and $\mathbf{u}_s = \mathbf{T}_s\mathbf{c}_s$;
  compute $\mathbf{p}_s$ and b_s by the fixed-point method in Eq.(79-80);
  update the residual matrices $\mathbf{X}_{s+1} = \mathbf{X}_s - \mathbf{r}_s\mathbf{p}_s^T$ and $\mathbf{T}_{s+1} = \mathbf{T}_s - b_s\mathbf{r}_s\mathbf{c}_s^T$;
end for
organize the matrices $\mathbf{R} = [\mathbf{r}_1, .., \mathbf{r}_S]$, $\mathbf{P} = [\mathbf{p}_1, .., \mathbf{p}_S]$, $\mathbf{B} = \mathrm{diag}(b_1, .., b_S)$, and $\mathbf{C} = [\mathbf{c}_1, .., \mathbf{c}_S]$;
compute $\mathbf{H} = \mathbf{P}^{T+}\mathbf{B}\mathbf{C}^T$

4.2.2. Convergence Analysis

For the regression relations $\mathbf{p}$ and b, one could find the detailed convergence analysis in [81]. The convergence of the projectors $\mathbf{w}$ and $\mathbf{c}$ is mainly considered here for the optimization issue of Eq.(72). Because correntropy is in nature an m-estimator [79], the local optimums of Eq.(72) will be sufficiently close to the global optimum, which has been proved in a recent theoretical study [181]. Therefore, one only needs to guarantee that Eq.(72) will converge to a local optimum with the HQ optimization.

If one has $J_p(\mathbf{w}_k, \mathbf{c}_k) \leqslant J_p(\mathbf{w}_{k+1}, \mathbf{c}_{k+1})$ when $(\alpha_n, \beta_n, \gamma_n) = (\alpha_{n,k+1}, \beta_{n,k+1}, \gamma_{n,k+1})$, Eq.(72) will converge to a local optimum.

Proof: The convergence is proved as

$$\begin{aligned} &J(\mathbf{w}_k, \mathbf{c}_k, \alpha_{n,k}, \beta_{n,k}, \gamma_{n,k}) \\ \leqslant &J(\mathbf{w}_k, \mathbf{c}_k, \alpha_{n,k+1}, \beta_{n,k+1}, \gamma_{n,k+1}) \\ \leqslant &J(\mathbf{w}_{k+1}, \mathbf{c}_{k+1}, \alpha_{n,k+1}, \beta_{n,k+1}, \gamma_{n,k+1}) \end{aligned} \tag{81}$$

in which the first inequality is guaranteed by the HQ mechanism, and the second inequality arises from the assumption of the present proposition. ■

One could observe that, to guarantee the convergence of Eq.(72), it is unnecessary to attain a strict maximum of Eq.(78) at each projector-step in Algorithm 3. On the contrary, as long as the updated projectors lead to a larger objective function J_p at each projector-step, Eq.(72) will converge to a local

optimum. This brings a great convenience in practice that one only needs a few SQP iterations for the projector-step. One could finish the projector-step once confirming an increase on J_p, thus accelerating the convergence.

4.2.3. Hyper-Parameter Determination

There exist five Gaussian kernel bandwidths h_x, h_t, h_r, h_p, and h_b, respectively, to be determined for the proposed PMCR algorithm. In practice, it will be intractable to determine five hyper-parameters by cross validation. To acquire a proper kernel bandwidth directly, one can employ the *Silverman's rule* [121] which was proposed for density estimation and has been successfully applied for ITL methods [157]. Under the current errors with N observations, the kernel bandwidth is computed by

$$h = 1.06 \times \min\{\sigma_e, \frac{R}{1.34}\} \times (N)^{-1/5} \tag{82}$$

in which σ_e is the standard deviation of the N errors, and R denotes the interquartile range.

4.3. ***Experiments***

For the performance evaluation of the proposed PMCR algorithm, one synthetic dataset and one real ECoG dataset were utilized, respectively. The proposed PMCR algorithm was compared with the conventional PLSR, the state-of-the-art regularized PLSR (RPLSR) [175] described in Eq.(67), and the rudimentary MCC-PLSR [178] described in Section 4.1.3. For evenhanded comparison, each method used an identical number of factors which was selected by the conventional PLSR from five-fold cross-validation. The maximal number of factors was set as 100.

Considering the performance indicators for the evaluation, three typical measures for regression tasks were utilized: i) Pearson's correlation coefficient (r) as defined by

$$r = \frac{Cov(\hat{\mathbf{T}}, \mathbf{T})}{\sqrt{Var(\hat{\mathbf{T}})Var(\mathbf{T})}} \tag{83}$$

where $Cov(\cdot,\cdot)$ and $Var(\cdot)$ denote the covariance and variance, respectively, and ii) root mean squared error (RMSE) which is computed by

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\|\hat{\mathbf{t}}_n - \mathbf{t}_n\|^2} \tag{84}$$

in which $\hat{\mathbf{t}}_n$ and $\mathbf{t}_n$ denote the n-th observations for the prediction $\hat{\mathbf{T}}$ and the target $\mathbf{T}$, respectively, and iii) mean absolute error (MAE) which represents the average L_1-norm distance

$$\text{MAE} = \frac{1}{N}\sum_{n=1}^{N}\|\hat{\mathbf{t}}_n - \mathbf{t}_n\| \tag{85}$$

4.3.1. Synthetic Dataset

1) Dataset Description:

First, an inter-correlated, high-dimensional, and noisy synthetic example was considered, where various PLSR methods were assessed with different levels of contamination. 300 i.i.d latent variables $\mathbf{r} \sim U(0,1)$ were generated randomly for training and testing, respectively, in which U denotes the uniform distribution, and the dimension of $\mathbf{r}$ was set as 20. A hypothesis from the latent variable to the explanatory and response variables was generated to produce the matrices of $\mathbf{X}$ and $\mathbf{T}$. To be specific, the transformation matrices were generated with arbitrary values of the standard normal distribution. The latent variables $\mathbf{r}$ were multiplied with a 20×500 transformation matrix, resulting in a 300×500

explanatory matrix for input. Similarly, a 20×3 transformation matrix was used to acquire a 300×3 response matrix for output. Thus, one predicted the multivariate responses from the 500-dimensional explanatory variables with 300 training samples, and then evaluated the performance on the other 300 testing samples.

Considering the contamination for the synthetic dataset, similarly as before, the explanatory data will be contaminated. To be specific, a certain proportion (from 0 to 1.0 with a step 0.05) of the training samples were randomly selected with equal probability, the inputs of which were then replaced by noises with large amplitude. For the distribution of the noise, a zero-mean Gaussian distribution with large standard deviation was utilized to imitate outliers, for which 30, 100, and 300 were considered, respectively.

2) Results:

Each PLSR algorithm was evaluated with 100 Monte-Carlo repetitive trials, and the regression performance indicators are illustrated in Fig. 16, where the results have been averaged across three dimensions of the output. One could observe from Fig. 16 that, for all the three different noise standard deviations, the proposed PMCR algorithm achieved superior prediction performance compared with the other existing methods consistently for r, RMSE, and MAE, respectively, in particular when the training dataset suffered considerable contamination.

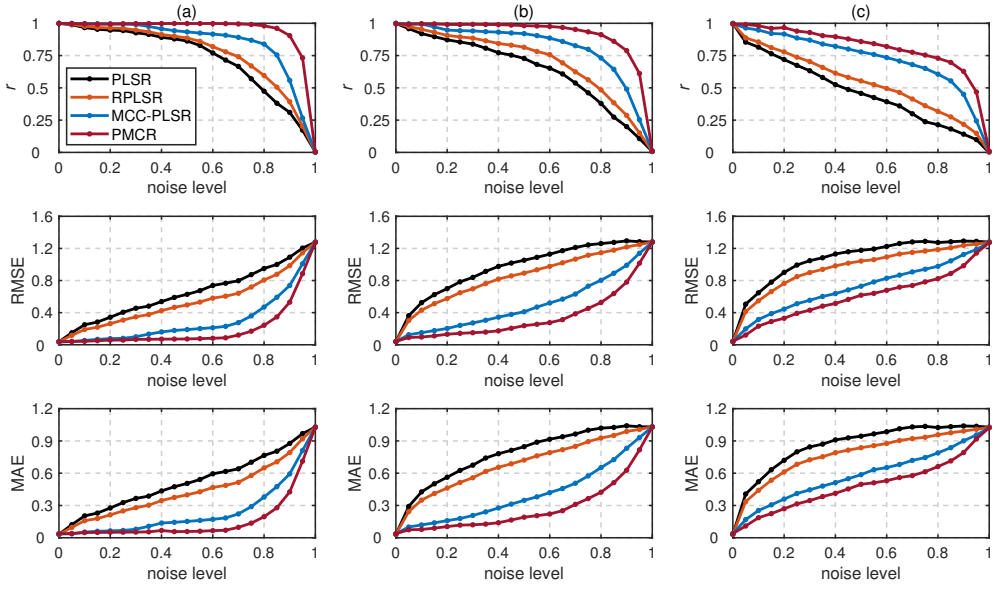

Figure 16. Regression performance of the inter-correlated, high-dimensional, and contaminated synthetic dataset under different noise standard deviations with noise levels from 0 to 1.0. (a): noise standard deviation = 30, (b): noise standard deviation = 100, and (c): noise standard deviation = 300. The performance indicators were acquired from 100 Monte-Carlo repetitive trials and averaged across three dimensions of the output. The proposed PMCR algorithm realized better performance than the existing PLSR algorithms consistently for r, RMSE, and MAE, in particular when the training set was contaminated considerably.

The number of decomposition factors S plays a vital role in PLSR methods, denoting the iteration numbers to decompose the input and output matrices. Since it usually causes a notable effect on the results, additionally, the performance was evaluated with respect to the number of factors for each method. To this end, the above synthetic dataset with the noise standard deviation 100 under three different noise levels 0.2, 0.5, and 0.8 was utilized. The prediction results for each method are presented in Fig. 17 with 100 repetitive trials, with respect to the number of decomposition factors. One observes that, not only the proposed PMCR eventually achieved superior regression performance with the optimal number of factors, but also it realized rather commendable performance with a small number of factors. For example, when the noise level was equal to 0.5, the proposed PMCR achieved its optimal

performance with no more than 20 factors. By comparison, for the other methods, when the number of factors was larger than 20, their performances remained promoting significantly. One can also observe a similar result in other two noise levels. This suggests that, PMCR can abstract substantial information with a rather small number of factors from training samples in a noisy regression task.

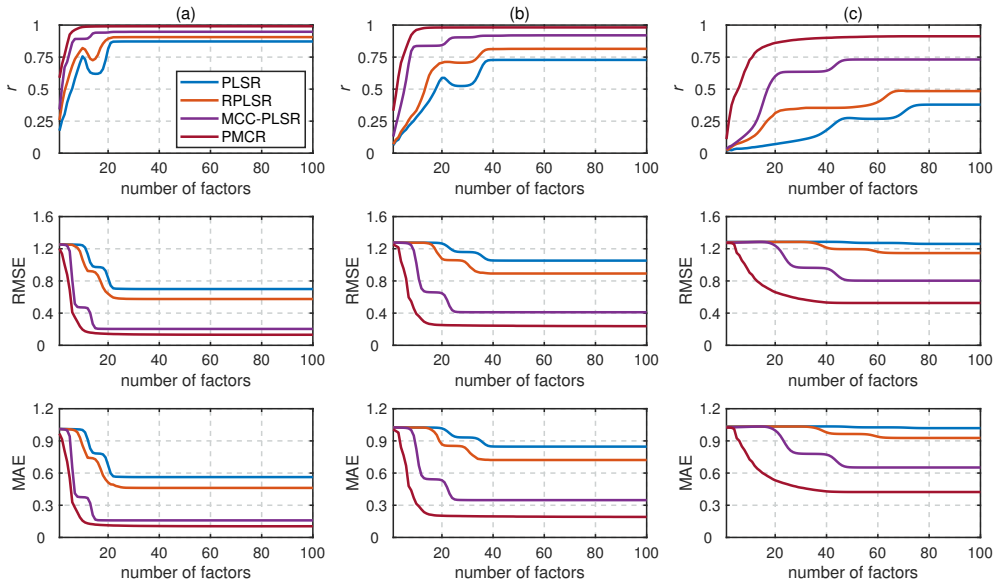

Figure 17. Regression performance of the synthetic dataset with noise standard deviation being 100 under three different noise levels with the number of factors increasing from 1 to 100. (a): noise level = 0.2, (b): noise level = 0.5, and (c): noise level = 0.8. The performance indicators were obtained from 100 repetitive trials and averaged across three dimensions of the output. The proposed PMCR algorithm not only acquired better prediction results than the other algorithms ultimately with the optimal number of factors, but also achieved admirable regression performance with a small number of factors.

4.3.2. ECoG Dataset

To further demonstrate the superior robustness of the PMCR algorithm in real-world brain activity decoding, each PLSR algorithm was evaluated by the publicly available Neurotycho ECoG dataset which was initially proposed in [28].

1) Dataset Description:

Two Japanese macaques, denoted by Monkey B and C, respectively, were commanded to track foods with the right hands, during which the continuous three-dimensional trajectories of right hands with the sampling rate of 120Hz were recorded by an optical motion capture instrument. For both Monkey B and C, ten recording sessions were performed, and each recording session lasted 15 minutes. Two macaques were in advance implanted with customized 64-channel ECoG electrodes on the left hemisphere, which covered the regions from the prefrontal cortex to the parietal cortex. ECoG signals were recorded simultaneously during each session with a sampling rate of 1,000 Hz. In accordance with [28], for each recording session, the data of the first ten minutes was used to train a prediction model, while the data of the remaining five minutes was used to evaluate the prediction performance of the trained model. The schemes of the experiments and ECoG electrodes are shown in Fig. 18 (a) and (b), respectively.

2) Decoding Paradigm:

For ECoG feature extraction, an identical offline decoding paradigm as in [28] was utilized. First, ECoG data was pre-processed with a tenth-order Butterworth bandpass filter with cutoff frequencies from 1 to 400 Hz, and then re-referenced by the common average referencing (CAR) method. Three-

dimensional trajectories of the right wrist were down-sampled to 10 Hz, thus, leading to 9000 samples in one session (10Hz×60sec×15min). The three-dimensional position was predicted from the ECoG signals during the previous one second. To extract the features of ECoG signals, the time–frequency representation was used. To be specific, for the time point t, the ECoG signals at each electrode from t - 1.1 s to t were processed by Morlet wavelet transformation. Ten center frequencies ranging from 10 to 120 Hz with equal spacing on the logarithmic scale were considered for the wavelet transformation, overlaying the frequency bands which are most relevant to motion tasks [28]. Time–frequency feature was then resampled at ten temporal lags with a 0.1 s gap (t - 1 s, t - 0.9 s,..., t - 0.1 s). Thus, the input of each sample exhibited a 6400-dimensional vector (64 channels×10 frequencies×10 temporal lags), and the output was the three-dimensional position of the right hand. As a result, a regression model would be trained with 6000 samples (the first ten minutes) to fit the three-dimensional output from the 6400-dimensional input, and then evaluated by the other 3000 testing samples (the remaining five minutes). The decoding paradigm is illustrated in Fig. 18 (c).

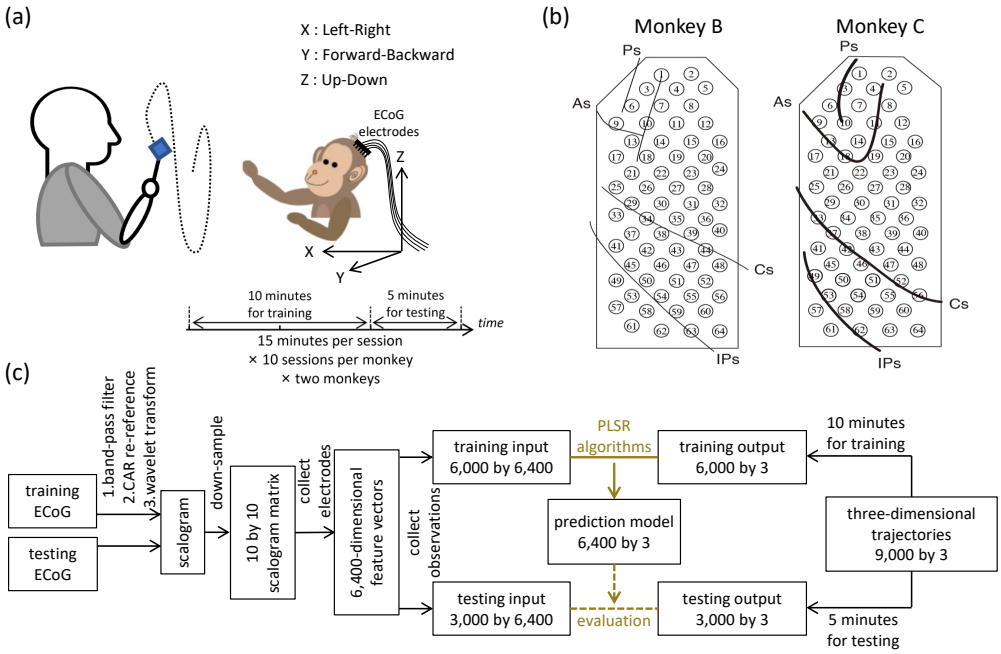

Figure 18. Experimental protocol of the Neurotycho ECoG dataset and decoding paradigm to evaluate the robustness of the different PLSR algorithms. (a): The macaque retrieved foods in a three-dimensional random location, during which the body-centered coordinates of the right wrists and the ECoG signals were recorded simultaneously. (b): Both Monkey B and C were implanted with 64-channel epidural ECoG electrodes on the contralateral (left) hemisphere, overlaying the regions from the prefrontal cortex to the parietal cortex. Ps: principal sulcus, As: arcuate sulcus, Cs: central sulcus, IPs: intraparietal sulcus. (a) and (b) were reproduced from [28], which provides the details of this public dataset. (c): Decoding diagram from ECoG signals to three-dimensional trajectories. The training ECoG signals are contaminated to assess the robustness of different algorithms.

3) Contamination:

To evaluate the robustness of different algorithms in the practical ECoG decoding task, the ECoG signals were artificially contaminated by outlier to simulate the detrimental artifact. To be specific, a certain proportions, 0 (no contamination), 10^{-3}, and 10^{-2}, of the training ECoG samplings were randomly selected and corrupted by outliers which obeyed a zero-mean Gaussian distribution with the variance 50 times that of the signals for the corresponding channel. As stated in [52], blink-related artifacts were remarkably found in ECoG signal that exhibited much larger amplitudes than a normal ECoG recording. Hence, the above method was utilized to generate artificial artifacts to contaminate the ECoG signals.

Note that, for this ECoG dataset, the "Noise Level" signifies the ratio of the contaminated ECoG samplings which is different from the ratio of the deteriorated samples among 6000 training samples. The ratio of the affected training samples can be evidently larger than the indicated noise level, since one contaminated ECoG sampling would deteriorate several time windows in feature extraction. For example, when the noise level is 10^{-3}, the deteriorated proportion of the training samples is (0.6645 ± 0.0089). Furthermore, how the noise influences the time–frequency feature is illustrated in Fig. 19. One could obviously find the heavy-tailed characteristic from the feature noises, which is in particular intractable for the least square criterion.

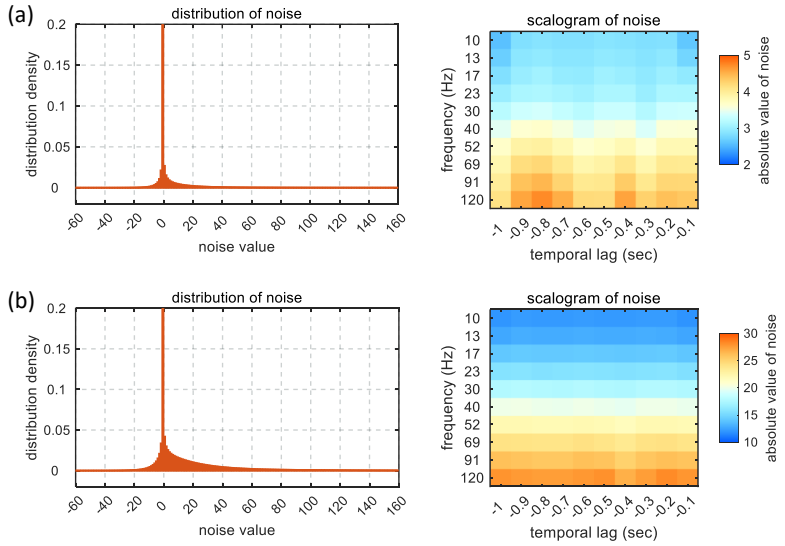

Figure 19. Distributions and scalograms of the time–frequency feature noises resulting from the ECoG sampling contamination. (a): noise level = 10^{-3} (the deteriorated proportion of training set = 0.6645 ± 0.0089), (b): noise level = 10^{-2} (the deteriorated proportion of training set ≈ 1). The time–frequency feature noises were calculated by subtracting the training datasets which were obtained from acoustic and contaminated ECoG signals, respectively. The distributions were averaged by 20 sessions of Monkey B and C, while the scalograms were averaged across all electrodes. The peaks of distributions are truncated to emphasize the heavy-tailed characteristic.

4) Spatio-spectro-temporal Pattern:

Investigating how the spatio-spectro-temporal weights in the regression model contribute to the entirety can help study the neurophysiological pattern. Each element of the trained regression model $\mathbf{H}$ can be denoted by $H_{\text{ch,freq,temp}}$, which corresponds to the ECoG electrode 'ch', the frequency 'freq', and the temporal lag 'temp'. Thus, one can calculate the spatio-spectro-temporal contributions by the ratio between the summation of absolute values of each domain and the summation of absolute values of the entire model

$$W_c(\text{ch}) = \frac{\sum_{\text{freq}} \sum_{\text{temp}} |H_{\text{ch,freq,temp}}|}{\sum_{\text{ch}} \sum_{\text{freq}} \sum_{\text{temp}} |H_{\text{ch,freq,temp}}|} \tag{86}$$

$$W_f(\text{freq}) = \frac{\sum_{\text{ch}} \sum_{\text{temp}} |H_{\text{ch,freq,temp}}|}{\sum_{\text{ch}} \sum_{\text{freq}} \sum_{\text{temp}} |H_{\text{ch,freq,temp}}|} \tag{87}$$

$$W_t(\text{temp}) = \frac{\sum_{\text{ch}} \sum_{\text{freq}} |H_{\text{ch,freq,temp}}|}{\sum_{\text{ch}} \sum_{\text{freq}} \sum_{\text{temp}} |H_{\text{ch,freq,temp}}|} \tag{88}$$

in which $W_c(\text{ch})$, $W_f(\text{freq})$, and $W_t(\text{temp})$ express the contributions of the ECoG electrode 'ch', the frequency 'freq', and the temporal lag 'temp', respectively.

5) Results:

Table 7. Prediction results of each algorithm on the Neurotycho ECoG dataset under three noise levels 0, 10^{-3}, and 10^{-2}, respectively. The results are given in mean±deviation, where the optimal results under each condition are marked in bold. The proposed PMCR realized the optimal results consistently, except for the Y-position under the noise level 0. For each result, (∗) is marked if there exists statistically significant difference between the indicated one and the optimal result in the corresponding condition, according to a paired t-test ($p < 0.05$).

X-position						
algorithm			PLSR	RPLSR	MCC-PLSR	PMCR
noise level	0	r	0.4378±0.0933 (∗)	0.4550±0.0925 (∗)	0.4598±0.0942 (∗)	**0.4679±0.0947**
		RMSE	0.9287±0.0810 (∗)	0.9037±0.0653 (∗)	0.8954±0.0809 (∗)	**0.8835±0.0786**
		MAE	0.7026±0.0640 (∗)	0.6872±0.0530 (∗)	0.6749±0.0628 (∗)	**0.6658±0.0651**
	10^{-3}	r	0.3334±0.1165 (∗)	0.3558±0.1132 (∗)	0.3684±0.1127 (∗)	**0.3873±0.1274**
		RMSE	0.9729±0.0652 (∗)	0.9543±0.0648 (∗)	0.9397±0.0728 (∗)	**0.9276±0.0705**
		MAE	0.7291±0.0756 (∗)	0.7174±0.0689 (∗)	0.7092±0.0786 (∗)	**0.6987±0.0759**
	10^{-2}	r	0.1524±0.1399 (∗)	0.1713±0.1353 (∗)	0.1926±0.1342 (∗)	**0.2238±0.1382**
		RMSE	1.0249±0.1105 (∗)	1.0022±0.1097 (∗)	0.9845±0.1129 (∗)	**0.9681±0.1094**
		MAE	0.7655±0.1428 (∗)	0.7485±0.1383 (∗)	0.7396±0.1392 (∗)	**0.7246±0.1397**
Y-position						
algorithm			PLSR	RPLSR	MCC-PLSR	PMCR
noise level	0	r	0.5426±0.1019 (∗)	**0.5582±0.1026**	0.5547±0.1017	0.5549±0.1022
		RMSE	0.8483±0.0969 (∗)	**0.8198±0.0951**	0.8246±0.0948	0.8233±0.0952
		MAE	0.6487±0.0762 (∗)	**0.6304±0.0796**	0.6362±0.0744	0.6358±0.0759
	10^{-3}	r	0.4114±0.1309 (∗)	0.4284±0.1285 (∗)	0.4425±0.1302 (∗)	**0.4602±0.1296**
		RMSE	0.9188±0.0963 (∗)	0.8962±0.0958 (∗)	0.8795±0.0979 (∗)	**0.8608±0.1002**
		MAE	0.6960±0.1007 (∗)	0.6849±0.1014 (∗)	0.6631±0.0983 (∗)	**0.6539±0.1021**
	10^{-2}	r	0.2084±0.1514 (∗)	0.2206±0.1489 (∗)	0.2593±0.1502 (∗)	**0.2723±0.1537**
		RMSE	0.9781±0.1143 (∗)	0.9542±0.1117 (∗)	0.9306±0.1159	**0.9294±0.1146**
		MAE	0.7354±0.1028 (∗)	0.7173±0.1077 (∗)	0.7086±0.1105	**0.7043±0.1042**
Z-position						
algorithm			PLSR	RPLSR	MCC-PLSR	PMCR
noise level	0	r	0.6320±0.0324 (∗)	0.6395±0.0328 (∗)	0.6482±0.0359	**0.6504±0.0372**
		RMSE	0.7968±0.0281 (∗)	0.7814±0.0293 (∗)	0.7747±0.0296 (∗)	**0.7628±0.0275**
		MAE	0.6181±0.0222 (∗)	0.6102±0.0280 (∗)	0.6055±0.0241	**0.5989±0.0265**
	10^{-3}	r	0.4875±0.0708 (∗)	0.4935±0.0701 (∗)	0.5158±0.0857 (∗)	**0.5259±0.0814**
		RMSE	0.9272±0.0712 (∗)	0.9129±0.0682 (∗)	0.8958±0.0742 (∗)	**0.8834±0.0738**
		MAE	0.6932±0.0800 (∗)	0.6894±0.0814 (∗)	0.6804±0.0852 (∗)	**0.6645±0.0782**
	10^{-2}	r	0.2399±0.1185 (∗)	0.2456±0.1173 (∗)	0.2615±0.1148 (∗)	**0.2803±0.1186**
		RMSE	1.0168±0.0804 (∗)	0.9917±0.0785 (∗)	0.9605±0.0842 (∗)	**0.9485±0.0809**
		MAE	0.7532±0.0883 (∗)	0.7429±0.0892 (∗)	0.7208±0.0893	**0.7146±0.0887**

First, each algorithms was evaluated with the uncontaminated ECoG signals. Accordingly, when the noise level was zero, the average performance indicators were obtained by 20 acoustic sessions (Monkey B and C). Then each session was contaminated with 5 repetitive trials. Thus, for each noise level, each algorithm was evaluated for 100 times (20 sessions × 5 repetitive trials). Table 7 presents the performance indicators for each algorithm with the noise levels 0, 10^{-3}, and 10^{-2}, respectively. In each row of a specific condition, the optimal result is marked in bold. Moreover, the other results are marked with (∗) if there exists a statistically significant difference between the current result and the optimal result under each condition. One could observe in Table 7 that, the proposed PMCR realized the optimal prediction results consistently, except the Y-axis under noise level 0. On most conditions, PMCR outperformed the other methods with statistically significant difference. One can observe that, when the noise level was 0, PMCR achieved better results than the other algorithms for X-axis and Z-axis. One major reason is, in the acoustic sessions, the motion-related artifacts have been evidently found in the ECoG signals [28], which further demonstrates the necessity of utilizing PMCR in real-world ECoG decoding tasks.

In addition, how the neurophysiological patterns for different algorithms were influenced by the sampling noises is studied. The differences between the spatial, the spectral, and the temporal weights which were respectively acquired from the acoustic and the contaminated sessions are shown in Fig.

20 under the noise level 10^{-3}. The regression model concerning Monkey B's Z-position was used here. The influence is also quantified by computing the summation of the absolute values of the difference between the patterns that were attained from the acoustic and the contaminated sessions, respectively. To be specific, $\sum|W_c(\text{ch}) - W'_c(\text{ch})|$, $\sum|W_f(\text{freq}) - W'_f(\text{freq})|$, and $\sum|W_t(\text{temp}) - W'_t(\text{temp})|$ express the pattern deterioration for the spatial, the spectral, and the temporal patterns, respectively, where $W_c(\text{ch})$, $W_f(\text{freq})$, and $W_t(\text{temp})$ were obtained by the acoustic sessions, and $W'_c(\text{ch})$, $W'_f(\text{freq})$, and $W'_t(\text{temp})$ were obtained by the contaminated sessions. One observes from Fig. 20 that, the proposed PMCR algorithm realized the minimal pattern deterioration for each domain. This further demonstrates the robustness of PMCR in noisy ECoG decoding tasks.

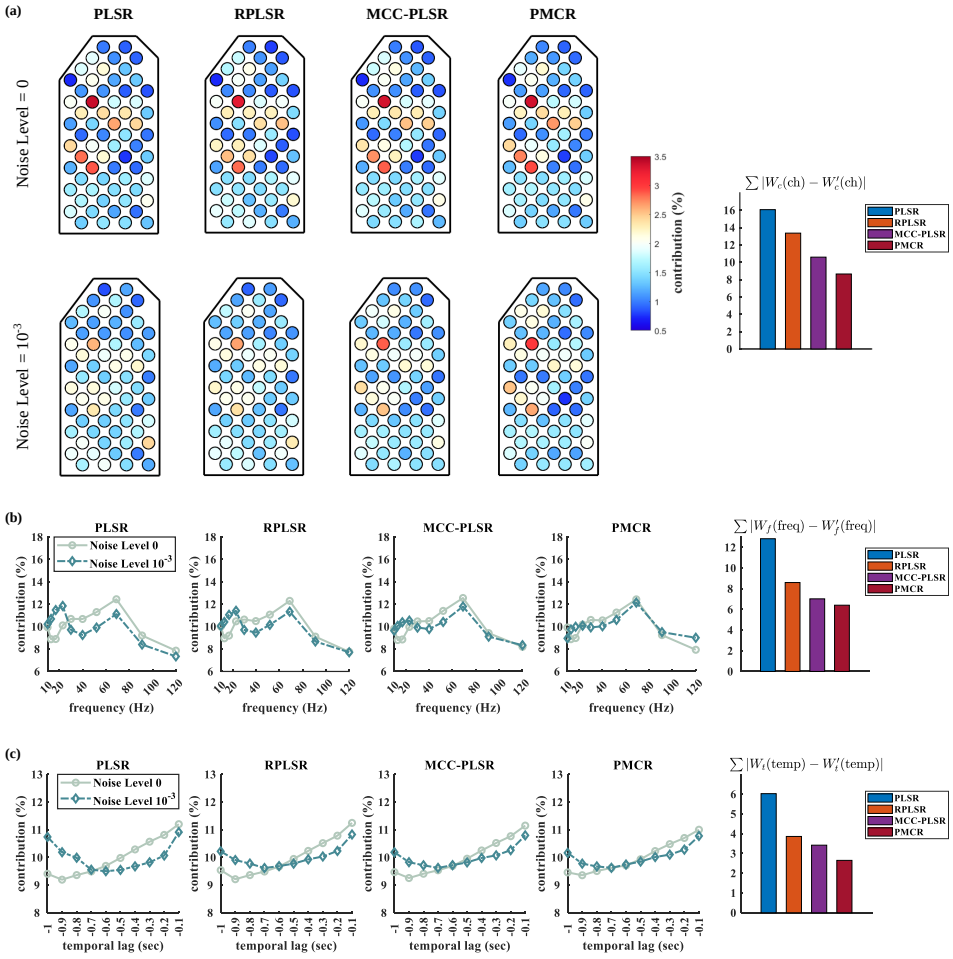

Figure 20. Spatio-spectro-temporal contributions of the prediction model for Monkey B's Z-position under noise levels 0 and 10^{-3}. (a): spatial patterns, (b): spectral patterns, and (c): temporal patterns. For each domain, the quantitative deterioration is calculated by the absolute value summation of the difference between the original and the deteriorated patterns. The original patterns $W_c(\text{ch})$, $W_f(\text{freq})$, and $W_t(\text{temp})$ were averaged across the 10 acoustic sessions of Monkey B, while the deteriorated patterns $W'_c(\text{ch})$, $W'_f(\text{freq})$, and $W'_t(\text{temp})$ were averaged across 50 trials (10 sessions of Monkey B $\times$ 5 repetitive trials). The proposed PMCR achieved the minimal deterioration for each domain.

4.4. Discussion

This section aims to propose a new robust version for PLSR using the MCC framework, which is named as PMCR. Similarly as the existing PLSR methods, the proposed PMCR decomposes the input matrix and the output matrix iteratively for S decomposition factors. The crucial differences of PMCR are stated in what follows. First, the objective function regarding the projectors $\mathbf{w}_s$ and $\mathbf{c}_s$ in Eq.(72) can

be considered as a generalized form of the conventional PLSR in Eq.(66), and is also closely related to the calculation in MCC-PLSR of Eq.(69) under specific conditions. As proved in [79], maximizing the correntropy between two variables, if the kernel bandwidth tends to infinity, is equal to minimizing their quadratic Euclidean distance. Hence, if one assumes $h_x, h_t, h_r \to \infty$, the projector calculation of PMCR will degenerate to the conventional PLSR. Then, the differences between MCC-PLSR and the proposed PMCR are discussed as follows. For a univariate response, the projector $\mathbf{c}$ for dimensionality reduction regarding the response could be ignored. Thus, one can rewrite the dimensionality reduction in PMCR of Eq.(72) as

$$\max_{\|\mathbf{w}\|_2=1} \sum_{n=1}^{N} \left(k_{h_x}(\sqrt{\mathbf{x}_n\mathbf{x}_n^T - \mathbf{x}_n\mathbf{w}\mathbf{w}^T\mathbf{x}_n^T}) + k_{h_r}(\mathbf{x}_n\mathbf{w} - \mathbf{t}_n) \right) \tag{89}$$

which could be regarded as a generalized form for the quadratic error minimization as

$$\min_{\|\mathbf{w}\|_2=1} \sum_{n=1}^{N} \left(\|\mathbf{x}_n - \mathbf{x}_n\mathbf{w}\mathbf{w}^T\|^2 + \|\mathbf{x}_n\mathbf{w} - \mathbf{t}_n\|^2 \right) \Leftrightarrow \max_{\|\mathbf{w}\|_2=1} \quad \mathbf{w}^T\mathbf{X}^T\mathbf{T} \tag{90}$$

which is essentially equal to the conventional PLSR for univariate output. By comparison, MCC-PLSR adopts the MCC framework for the quadratic covariance of Eq.(68), which can be written as

$$\min_{\|\mathbf{w}\|_2=1} \sum_{n=1}^{N} \|\mathbf{t}_n^T\mathbf{x}_n - \mathbf{t}_n^T\mathbf{x}_n\mathbf{w}\mathbf{w}^T\|^2 \Leftrightarrow \max_{\|\mathbf{w}\|_2=1} \mathbf{w}^T\mathbf{X}^T\mathbf{T}\mathbf{T}^T\mathbf{X}\mathbf{w} \tag{91}$$

which is the special case of MCC-PLSR when the kernel bandwidth in Eq.(69) tends to infinity. Thus, the connection between PMCR of Eq.(72) and MCC-PLSR of Eq.(69) could be interpreted as in what follows. One can observe that, the starting points of PMCR and MCC-PLSR are different. The proposed PMCR begins from the original covariance maximization, while MCC-PLSR was proposed from the quadratic covariance. Therefore, it would be argued that the proposed PMCR is a more rational robust version for PLSR. Moreover, note that the above discussion is given under the premise of a univariate output, which is only a special case of degradation for PMCR. One the other hand, considering the calculations of the loading vector $\mathbf{p}_s$ and the regression coefficient b_s, the proposed PMCR utilizes the MCC in Eq.(73-74), while the conventional PLSR and MCC-PLSR use the least square criterion. As mentioned above, Eq.(73-74) can be also regarded as generalized forms of square error minimization. In summary, the proposed PMCR is more generalized than the conventional PLSR and MCC-PLSR.

In addition, advantages and disadvantages of the proposed PMCR are also discussed as follows. The essential benefit of utilizing the PMCR algorithm in a noisy ECoG decoding task is the conspicuous robustness with respect to the noises, as was demonstrated with extensive experiments in Section 4.3. Further, mathematically the proposed PMCR is more generalized than the conventional PLSR and MCC-PLSR. As was mentioned above, the conventional PLSR and MCC-PLSR could be regarded as special cases of the proposed PMCR under specific conditions. In particular, compared to MCC-PLSR, the proposed PMCR takes into account the dimensionality reduction for the response matrix. Hence, PMCR could realize better prediction performance for multivariate response. However, PMCR might suffer the performance degradation resulting from inadequate kernel bandwidths that are calculated by the *Silverman's rule* of Eq.(82). Although the experimental results demonstrated empirically that, the proposed PMCR could perform efficiently with the kernel bandwidths acquired by Eq.(82), it may be difficult to guarantee that the *Silverman's rule* can always provide adequate bandwidths. Therefore, a better way to determine the kernel bandwidth is supposed to be investigated in the future works. On the other hand, the proposed PMCR is effective to deal with outliers, while it may be inadequate for multi-modal-distributed noise because MCC utilizes only one kernel function for each reconstruction error. To address this issue, it will be promising to use MEE to reformulate the PLSR algorithm.

5. Correntropy-Based Automatic Relevance Determination

In addition to the dimensionality reduction techniques, the sparse learning is another effective approach to address the high-dimensional problem by employing a reduced set of covariates for the prediction task. For regression task, conventional LS criterion based regression Eq.(4-6) is only effective for well-posed cases. In the high-dimensional situation, i.e. $D > N$, the LS solution Eq.(6) will be ill-posed which results in poor generalization performance. Similarly, in a high-dimensional classification task, directly utilizing binomial MLE of Eq.(10) or CE loss minimum of Eq.(11) will also lead to poor generalization to new testing samples. A useful solution is to select a subset of covariates and prune those less important features, called sparse learning. In the final model parameter, many components will be zero so that the corresponding dimensions are pruned. The sparse model has been increasingly attractive for brain decoding because the selected features could indicate the spatio-temporal patterns relevant to the specific cognitive tasks [99,101,102,182].

The idealized sparse model is realized by minimizing the L_0-regularized cost function

$$\mathbf{w}^* = arg\min_{\mathbf{w}} \mathbb{E}_{p(e)}\left[\mathcal{L}_e\right] + \lambda\|\mathbf{w}\|_0 \tag{92}$$

in which $\mathcal{L}_e$ is an arbitrary loss function with respect to the prediction error e. λ is a hyper-parameter tuning the regularization strength, and $\|\mathbf{w}\|_0$ is the L_0-norm of $\mathbf{w}$ denoting the number of non-zero components in $\mathbf{w}$. Compared to empirical risk minimization (ERM) which only contains loss function, Eq.(92) is called structural risk minimization (SRM) which further considers the balance between the model's complexity against its success at fitting the training data. Because solving Eq.(92) is NP-hard, L_0-norm is usually replaced with its tightest *convex* relaxation L_1-norm [183] which leads to the famous LASSO algorithm [184,185]

$$\mathbf{w}^* = arg\min_{\mathbf{w}} \mathbb{E}_{p(e)}\left[\mathcal{L}_e\right] + \lambda\|\mathbf{w}\|_1 \tag{93}$$

which has been well studied for sparse learning [186–189]. However, the hyper-parameter λ is usually a nuisance which would require manual tuning or time-consuming cross-validation.

An alternative approach to realize a sparse machine learning model is to employ a sparse prior distribution on the model parameters and update the model parameter from the Bayesian perspective. In essence, L_1-regularization is equivalent to utilizing a Laplacian prior distribution [184]. In addition, the EP-GIG prior has been also investigated for sparse regression in a Bayesian framework [190]. This section mainly concentrates on the automatic relevance determination (ARD) technique [191] which is a hierarchical sparse prior and has proved to be more adequate than the Laplacian prior (equally L_1-regularization) for feature selection [183]. A notable advantage of the ARD technique is that one does not need to adjust the regularization parameter manually [186], i.e., "adaptive sparseness". ARD-based sparse models have been widely utilized for brain activity decoding, including EEG decoding [192–195], fMRI decoding [33,99,196–198], and current source density analysis [199–201]. Nevertheless, existing sparse Bayesian learning models are formulated from conventional likelihood functions, such as the non-robust Gaussian and binomial likelihoods, which will result in poor robustness with respect to non-Gaussian noises or outliers.

On the other hand, as introduced before, MCC is highly efficient for noisy data analysis [79,80,82,83], which has been also used for robust sparse learning integrating with L_1-regularization [202–204] or other regularization terms [205,206]. However, as mentioned above, these regularization terms need a careful tuning on regularization hyper-parameters. This section aims to investigate how to introduce MCC-based robust learning into the ARD-based sparse Bayesian learning framework, such that MCC can be implemented with "adaptive sparseness".

5.1. *Automatic Relevance Determination for Sparse Learning*

The automatic relevance determination (ARD) technique was originally proposed in [207], which has been receiving growing attention with the proposal of relevance vector machine (RVM) [208–210], a

Bayesian treating of support vector machine (SVM). In what follows, the ARD technique was reviewed briefly with its implementation in sparse Bayesian regression and classification.

First, consider the regression scenario here. Based on the assumption of a linear regression model of Eq.(3), the classical approach is to suppose a zero-mean Gaussian distribution for the noise term ϵ

$$\epsilon \sim \mathcal{N}(\epsilon|0, \sigma^2) \tag{94}$$

in which the noise variance is denoted by σ^2. Based on the data generation assumption of Eq.(3), one can obtain the distribution for the target variable t by

$$p(t|\mathbf{x}, \mathbf{w}, \sigma^2) = \mathcal{N}(t|\mathbf{x}\mathbf{w}, \sigma^2) \tag{95}$$

which is a Gaussian distribution over t with mean $\mathbf{x}\mathbf{w}$ and variance σ^2. By a finite dataset $\{(\mathbf{x}_n, t_n)\}_{n=1}^N$ and the i.i.d assumption, one can write the probability for the whole dataset, i.e. the likelihood function, by

$$p(\mathbf{t}|\mathbf{w}, \sigma^2) = \prod_{n=1}^{N} p(t_n|\mathbf{x}_n, \mathbf{w}, \sigma^2) = (2\pi\sigma^2)^{-N/2} \exp\{-\frac{1}{2\sigma^2}\|\mathbf{t} - \mathbf{X}\mathbf{w}\|^2\} \tag{96}$$

in which the dependence upon the covariate matrix $\mathbf{X}$ is omitted for simplicity. One could obviously observe that the MLE solution of Eq.(96) is equivalent to the least square criterion of Eq.(5).

ARD assigns a prior distribution for the model parameter $\mathbf{w}$ with a hierarchical form. Specifically, ARD utilizes a zero-mean and anisotropic Gaussian distribution for each entry of the model parameter $\mathbf{w}$ by

$$p(\mathbf{w}|\mathbf{a}) = \prod_{d=1}^{D} p(w_d|a_d) = \prod_{d=1}^{D} \mathcal{N}(w_d|0, a_d^{-1}) \tag{97}$$

in which each w_d is assumed with a Gaussian distribution of zero mean and variance a_d^{-1}. The hyper-parameter $\mathbf{a} = (a_1, a_2, \cdots, a_D)$ which denotes the inverse variances for $\mathbf{w}$ is called relevance parameter, controlling the possible range for corresponding w_d. Each relevance parameter a_d is then assumed by the non-informative Jeffreys hyper-prior (which is actually an *improper* prior since its integral is infinite and thus it is not normalizable) [211]

$$p(\mathbf{a}) = \prod_{d=1}^{D} p(a_d) = \prod_{d=1}^{D} a_d^{-1} \tag{98}$$

The prior distribution for noise variance σ^2 is usually assumed to be non-informative as well

$$p(\sigma^2) = (\sigma^2)^{-1} \tag{99}$$

By defining the likelihood function in Eq.(96) and the prior distributions in Eq.(97-99), one can write analytically the posterior distribution over $\mathbf{w}$ by

$$\begin{aligned} p(\mathbf{w}|\mathbf{t}, \mathbf{a}, \sigma^2) &= \frac{p(\mathbf{t}|\mathbf{w}, \sigma^2)p(\mathbf{w}|\mathbf{a})}{p(\mathbf{t}|\mathbf{a}, \sigma^2)} \\ &= \frac{p(\mathbf{t}|\mathbf{w}, \sigma^2)p(\mathbf{w}|\mathbf{a})}{\int p(\mathbf{t}|\mathbf{w}, \sigma^2)p(\mathbf{w}|\mathbf{a})d\mathbf{w}} \\ &= (2\pi)^{-D/2}|\Sigma|^{-1/2} \exp\{-\frac{1}{2}(\mathbf{w} - \boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{w} - \boldsymbol{\mu})\} \end{aligned} \tag{100}$$

in which the covariance and mean for $\mathbf{w}$ are computed by

$$\begin{aligned} \Sigma &= (\sigma^{-2}\mathbf{X}^T\mathbf{X} + \mathbf{A})^{-1} \\ \boldsymbol{\mu} &= \sigma^{-2}\Sigma\mathbf{X}^T\mathbf{t} \end{aligned} \tag{101}$$

with $\mathbf{A} = diag(a_1, a_2, \cdots, a_D)$. To obtain the whole posterior distribution

$$p(\mathbf{w}, \mathbf{a}, \sigma^2|\mathbf{t}) = p(\mathbf{w}|\mathbf{t}, \mathbf{a}, \sigma^2)p(\mathbf{a}, \sigma^2|\mathbf{t}) \tag{102}$$

one could observe that the hyper-parameter posterior distribution could be denoted by $p(\mathbf{a}, \sigma^2|\mathbf{t}) \propto p(\mathbf{t}|\mathbf{a}, \sigma^2)p(\mathbf{a})p(\sigma^2)$. Utilizing the non-informative hyper-priors, one only need to optimize $\mathbf{a}$ and σ^2 so that the *marginal likelihood* $p(\mathbf{t}|\mathbf{a}, \sigma^2)$ is maximized

$$\begin{aligned} p(\mathbf{t}|\mathbf{a}, \sigma^2) &= \int p(\mathbf{t}|\mathbf{w}, \sigma^2)p(\mathbf{w}|\mathbf{a})d\mathbf{w} \\ &=(2\pi)^{-D/2}|\sigma^2\mathbf{I} + \mathbf{X}\mathbf{A}^{-1}\mathbf{X}^T|^{-1/2} \exp\{-\frac{1}{2}\mathbf{t}^T(\sigma^2\mathbf{I} + \mathbf{X}\mathbf{A}^{-1}\mathbf{X}^T)^{-1}\mathbf{t}\} \end{aligned} \tag{103}$$

To maximize Eq.(103), setting the differentiation to zero yields the following update

$$a_d = \frac{\gamma_d}{\mu_d^2} \tag{104}$$

in which μ_d is the d-th component of $\boldsymbol{\mu}$ and γ_d is defined by $\gamma_d \triangleq 1 - a_d\Sigma_{dd}$ with Σ_{dd} the d-th diagonal element of Σ. σ^2 is updated by

$$\sigma^2 = \frac{\|\mathbf{t} - \mathbf{X}\boldsymbol{\mu}\|^2}{N - \sum_{d=1}^{D}\gamma_d} \tag{105}$$

Updating (101)(104)(105) alternately, one can obtain the *maximum a posteriori* (MAP) estimations for all the unknown variables. In particular, during the inference, those a_d which correspond to irrelevant features will diverge to arbitrarily large numbers, so that the probability density of the corresponding w_d focuses at the origin, thus pruning the irrelevant features and realizing sparse regression.

The above-described optimization involves maximization of *marginal likelihood* $p(\mathbf{t}|\mathbf{a}, \sigma^2)$ Eq.(103), which is known as the *type-II maximum likelihood* [211]. Furthermore, the model could be optimized in other ways. For example, Expectation-Maximum (EM) could be employed by regarding the relevance parameter $\mathbf{a}$ as the hidden variables [186]. One could also use the variational Bayesian (VB) method with surrogate function to approximate the posterior distribution for each random variable [210]. Due to the inadequate assumption of Gaussian-distributed noise in Eq.(94), this conventional ARD-based sparse regression may suffer significant performance degeneration in a realistic non-Gaussian scenario, in particular in the presence of outliers [79,83,89,212].

On the other hand, to realize sparse classification, ARD has also been successfully implemented. In particular, ARD was introduced into the logistic regression model in [99], named as sparse logistic regression (SLR) algorithm. The notable difference for ARD-based regression and classification is that, the likelihood function will be different. For example, the likelihood function for logistic regression has been given previously in Eq.(9), and also rewritten here

$$p(\mathbf{t}|\mathbf{w}) = \prod_{n=1}^{N} p(t_n|\mathbf{w}) = \prod_{n=1}^{N} y_n^{t_n}(1 - y_n)^{1-t_n} \tag{106}$$

where the dependence upon the covariate $\mathbf{X}$ is also omitted. However, one can find that it would be intractable to derive an analytical posterior distribution as in regression

$$p(\mathbf{w}|\mathbf{t}, \mathbf{a}) = \frac{\int p(\mathbf{t}|\mathbf{w})p(\mathbf{w}|\mathbf{a})p(\mathbf{a})d\mathbf{a}}{\int\int p(\mathbf{t}|\mathbf{w})p(\mathbf{w}|\mathbf{a})p(\mathbf{a})d\mathbf{a}d\mathbf{w}} \tag{107}$$

which is because the likelihood function of Eq.(106) is not the conjugate function with the ARD priors as defined in Eq.(97-99). To realize the MAP estimation for ARD-based logistic regression, [99] used

the variational inference [213] approach to approximate the true posterior distribution. To be specific, to infer the posterior distribution for $\mathbf{w}$, variational inference defines the following free energy function

$$F(q(\mathbf{w},\mathbf{a})) \triangleq -\mathbb{E}_{q(\mathbf{w},\mathbf{a})}\left[\log \frac{p(\mathbf{t},\mathbf{w},\mathbf{a})}{q(\mathbf{w},\mathbf{a})}\right] = -\int q(\mathbf{w},\mathbf{a}) \log \frac{p(\mathbf{t},\mathbf{w},\mathbf{a})}{q(\mathbf{w},\mathbf{a})} d\mathbf{a} d\mathbf{w} \tag{108}$$

in which $q(\mathbf{w},\mathbf{a})$ is an approximation for the true joint posterior distribution $p(\mathbf{w},\mathbf{a}|\mathbf{t})$. When the free energy $F(q(\mathbf{w},\mathbf{a}))$ is minimized, the Kullback-Leibler divergence between $q(\mathbf{w},\mathbf{a})$ and $p(\mathbf{w},\mathbf{a}|\mathbf{t})$ will be also minimized, which means a maximal similarity between them, so that the approximation to $p(\mathbf{w},\mathbf{a}|\mathbf{t})$ can be realized. To accomplish the free energy minimization, one could further assume the conditional independence between $\mathbf{w}$ and $\mathbf{a}$ by $q(\mathbf{w},\mathbf{a}) = q_{\mathbf{w}}(\mathbf{w})q_{\mathbf{a}}(\mathbf{a})$. Thus, the free energy becomes

$$F(q_{\mathbf{w}}(\mathbf{w})q_{\mathbf{a}}(\mathbf{a})) = -\int q_{\mathbf{w}}(\mathbf{w})q_{\mathbf{a}}(\mathbf{a}) \log \frac{p(\mathbf{t},\mathbf{w},\mathbf{a})}{q_{\mathbf{w}}(\mathbf{w})q_{\mathbf{a}}(\mathbf{a})} d\mathbf{a} d\mathbf{w} \tag{109}$$

By doing so, one can minimize the free energy alternately with respect to $q_{\mathbf{w}}(\mathbf{w})$ and $q_{\mathbf{a}}(\mathbf{a})$ by

$$\begin{aligned} \text{w-step:} \quad & \log q_{\mathbf{w}}(\mathbf{w}) = \mathbb{E}_{q_{\mathbf{a}}(\mathbf{a})}\left[\log p(\mathbf{t},\mathbf{w},\mathbf{a})\right] + const \\ \text{a-step:} \quad & \log q_{\mathbf{a}}(\mathbf{a}) = \mathbb{E}_{q_{\mathbf{w}}(\mathbf{w})}\left[\log p(\mathbf{t},\mathbf{w},\mathbf{a})\right] + const \end{aligned} \tag{110}$$

Despite the exceptional capability for feature selection, SLR may suffer a significant performance degradation resulting from noises in practice. The MAP estimation of Eq.(107) can be rewritten by integration as

$$p(\mathbf{w}|\mathbf{t}) = \frac{p(\mathbf{t}|\mathbf{w})p(\mathbf{w})}{p(\mathbf{t})} \propto p(\mathbf{t}|\mathbf{w})p(\mathbf{w}) \tag{111}$$

in which $p(\mathbf{t})$ is a constant that is usually called evidence, and $p(\mathbf{w}) = \int p(\mathbf{w}|\mathbf{a})p(\mathbf{a})d\mathbf{a}$ indicates the prior distribution for $\mathbf{w}$ by integrating out $\mathbf{a}$. Since the logarithm is a monotonically increasing function, MAP is equal to

$$\begin{aligned} \max\ p(\mathbf{w}|\mathbf{t}) &\Leftrightarrow \max\ \log p(\mathbf{w}|\mathbf{t}) \\ &\Leftrightarrow \max\ \log p(\mathbf{t}|\mathbf{w}) + \log p(\mathbf{w}) \end{aligned} \tag{112}$$

from which one can find the log likelihood function term $\log p(\mathbf{t}|\mathbf{w})$ which is actually equivalent to the non-robust CE loss as shown in Eq.(10-11). As demonstrated in Section 3.4, this binomial assumption or equally the CE loss function exhibits poor robustness in noisy classification tasks.

5.2. *Correntropy-Based Sparse Logistic Regression*

To ameliorate the inadequate robustness of ARD-based sparse Bayesian learning, an investigation of how to employ robust MCC method in a Bayesian learning framework is explored. First, this section focuses on the SLR algorithm and discusses if it can be possible to replace the non-robust likelihood function with the MCC objective function. Since C-loss can outperform CE loss significantly in a noisy classification task, this proposal is supposed to be effective intuitively by using MCC instead of the log likelihood $\log p(\mathbf{t}|\mathbf{w})$.

To be specific, because the likelihood measures the probability that the prediction is equal to the target output, comparably, one could employ the robust correntropy instead to measure the similarity between prediction and desired output, motivated by the splendid robustness of MCC. Therefore, the correntropy similarity between the prediction and target $V(t,y)$ is utilized, in which y is the predicted probability as computed by Eq.(8), to substitute the non-robust log likelihood $\log p(\mathbf{t}|\mathbf{w})$. The superior robustness of maximizing the correntropy $V(t,y)$, or using C-loss for classification, has been verified on ordinary logistic regression model and radial basis function network with extensive experimental results in [75,136], while also is demonstrated in Section 3.4.

To coordinate the correntropy term $V(t,y)$ to the Bayesian derivation of SLR, one could rewrite the log joint distribution $\log p(\mathbf{t},\mathbf{w},\mathbf{a})$ by decomposition

$$\log p(\mathbf{t},\mathbf{w},\mathbf{a}) = \log p(\mathbf{t}|\mathbf{w}) + \log p(\mathbf{w}|\mathbf{a}) + \log p(\mathbf{a}) \tag{113}$$

from which one can also find the non-robust $\log p(\mathbf{t}|\mathbf{w})$. Here, a correntropy-based pseudo log joint distribution $\log p_c(\mathbf{t},\mathbf{w},\mathbf{a})$ is proposed by

$$\log p_c(\mathbf{t},\mathbf{w},\mathbf{a}) \triangleq V(\mathbf{t},\mathbf{y}) + \log p(\mathbf{w}|\mathbf{a}) + \log p(\mathbf{a}) \tag{114}$$

where $p_c(\mathbf{t},\mathbf{w},\mathbf{a})$ is called pseudo joint distribution since one finds its integration over all values cannot be normalized to be one. $\mathbf{y}$ is the collection of the predicted probability $\mathbf{y} = (y_1, y_2, \cdots, y_N)^T \in \mathbb{R}^{N\times 1}$.

By defining the novel pseudo joint distribution $p_c(\mathbf{t},\mathbf{w},\mathbf{a})$, the free energy minimization of Eq.(109) is reformulated by

$$\min -\int q_{\mathbf{w}}(\mathbf{w}) q_{\mathbf{a}}(\mathbf{a}) \log \frac{p_c(\mathbf{t},\mathbf{w},\mathbf{a})}{q_{\mathbf{w}}(\mathbf{w}) q_{\mathbf{a}}(\mathbf{a})} d\mathbf{a} d\mathbf{w} \tag{115}$$

Similarly, one can acquire the following alternate optimization

$$\begin{aligned} \text{w-step:} \quad & \log q_{\mathbf{w}}(\mathbf{w}) = \mathbb{E}_{q_{\mathbf{a}}(\mathbf{a})}\left[\log p_c(\mathbf{t},\mathbf{w},\mathbf{a})\right] + const \\ \text{a-step:} \quad & \log q_{\mathbf{a}}(\mathbf{a}) = \mathbb{E}_{q_{\mathbf{w}}(\mathbf{w})}\left[\log p_c(\mathbf{t},\mathbf{w},\mathbf{a})\right] + const \end{aligned} \tag{116}$$

Computing the expectation and omitting the constant in w-step, one obtains

$$\log q_{\mathbf{w}}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N} \exp(-\frac{(t_n - y_n)^2}{2h}) - \frac{1}{2}\mathbf{w}^T\mathbf{A}\mathbf{w} \tag{117}$$

where $\mathbf{A} = diag(a_1, a_2, \cdots, a_D)$. However, one can find that $q_{\mathbf{w}}(\mathbf{w})$ cannot be expressed by any forms of an arbitrary distribution, and thus the distribution for $\mathbf{w}$ remains unclear. To address this issue, w-step is further approximated by the Laplacian approximation method by

$$\log q_{\mathbf{w}}(\mathbf{w}) \approx \log q_{\mathbf{w}}(\mathbf{w}^*) - \frac{1}{2}(\mathbf{w} - \mathbf{w}^*)^T H(\mathbf{w}^*)(\mathbf{w} - \mathbf{w}^*) \tag{118}$$

in which $\mathbf{w}^*$ denotes the maximum point of $\log q_{\mathbf{w}}(\mathbf{w})$, and $H(\mathbf{w}^*)$ is the negative Hessian matrix of $\log q_{\mathbf{w}}(\mathbf{w})$ at $\mathbf{w}^*$. Thus, $q_{\mathbf{w}}(\mathbf{w})$ is approximated with a Gaussian distribution by

$$q_{\mathbf{w}}(\mathbf{w}) \approx \mathcal{N}(\mathbf{w}|\mathbf{w}^*, S(\mathbf{w}^*)) \tag{119}$$

where $S(\mathbf{w}^*) \triangleq H(\mathbf{w}^*)^{-1}$. The gradient of $\log q_{\mathbf{w}}(\mathbf{w})$ with respect to model parameter $\mathbf{w}$ is given by

$$\frac{\partial \log q_{\mathbf{w}}(\mathbf{w})}{\partial \mathbf{w}} = \frac{1}{N\sigma^2}\sum_{n=1}^{N} \exp(-\frac{e_n^2}{2h}) e_n y_n (1 - y_n)\mathbf{x}_n - \mathbf{A}\mathbf{w} \tag{120}$$

The Hessian matrix of $\log q_{\mathbf{w}}(\mathbf{w})$ is given by

$$\frac{\partial^2 \log q_{\mathbf{w}}(\mathbf{w})}{\partial \mathbf{w} \partial \mathbf{w}^t} = \frac{1}{Nh}\sum_{n=1}^{N} \mathbf{x}_n^T \left\{ \exp(-\frac{e_n^2}{2h}) \left[(\frac{e_n^2}{h} - 1) y_n^2 (1 - y_n)^2 + e_n y_n (1 - y_n)(1 - 2y_n) \right] \right\} \mathbf{x}_n - \mathbf{A} \tag{121}$$

For a-step, given $q_{\mathbf{w}}(\mathbf{w}) \approx \mathcal{N}(\mathbf{w}|\mathbf{w}^*, S(\mathbf{w}^*))$, one can obtain

$$\log q_{\mathbf{a}}(\mathbf{a}) = -\frac{1}{2}\sum_{d=1}^{D} (a_d(w_d^{*2} + s_d^2) + \log a_d) \tag{122}$$

where $\mathbf{w}^*$ is the posterior mean acquired from the w-step, and s_d^2 is the d-th diagonal element in $S(\mathbf{w}^*)$. $q_{\mathbf{a}}(\mathbf{a})$ could be regarded to obey the following Gamma distribution

$$q_{\mathbf{a}}(\mathbf{a}) = \prod_{d=1}^{D} q_{a_d}(a_d) = \prod_{d=1}^{D} \Gamma(a_d^*, \frac{1}{2}) \tag{123}$$

in which $\Gamma(a_d^*, \frac{1}{2})$ is the Gamma distribution with the degree of freedom being $\frac{1}{2}$ and the expectation being a_d^* that is

$$a_d^* = \frac{1}{w_d^{*2} + s_d^2} \tag{124}$$

The reformulated ARD-based sparse logistic regression with the correntropy learning framework, proposed as above, is named as CSLR. Then, the optimization for w-step and a-step will be discussed. In w-step, $\log q_{\mathbf{w}}(\mathbf{w})$ of Eq.(117) is in essence equal to an L_2-regularized MCC-based logistic regression. Although this is a non-convex problem because of the integration of sigmoid function and Gaussian kernel function, it is acceptable to obtain a local optimum for w-step, because it has been proved that any local optimums of regularized m-estimation are sufficiently close to the global optimum [214] and correntropy is exactly a robust formulation of the Welsch m-estimator [79]. To acquire a local optimum for w-step, one can similarly utilize the HQ technique which has been discussed in detail in Section 3.3.1. Considering the update for $\mathbf{a}$, one can use the following rule to accelerate the convergence

$$a_d^* = \frac{1 - a_d^* s_d^2}{w_d^{*2}} \tag{125}$$

which is motivated by the effective number of parameters [191].

CSLR executes the w-step and a-step alternately, updating the model parameters and relevance parameters. During the model training, the relevance parameters of the irrelevant features can diverge to infinity, that the probability density of corresponding model parameters is distributed at zero, thus pruning irrelevant features and obtaining a sparse classifier. In practice, one could set an upper limit, such as 10^8. If a_d exceeds the upper limit, the corresponding features will be pruned in the subsequent model training. The proposed CSLR for robust sparse classification is summarized in Algorithm 4.

Algorithm 4 CSLR for robust sparse classification

input:
  training samples $\{(\mathbf{x}_n, t_n)\}_{n=1}^N$;
  Gaussian kernel bandwidth h;
  threshold for relevance parameter $a_{\max}$;
initialize:
  model parameters w_d $(d = 1, \cdots, D)$;
  relevance parameters a_d $(d = 1, \cdots, D)$;
output:
  model parameters w_d $(d = 1, \cdots, D)$
repeat
  w-step: update w_d according to HQ technique;
  a-step: update a_d according to Eq.(125);
  if $a_d \geqslant a_{\max}$ **then**
    adjust the corresponding model parameters to zero and prune the corresponding features from the samples in the following iterations
  end if
until the parameter change is small enough or the number of iterations exceeds a predetermined limit

5.3. ***Experiments***

For performance evaluation, the proposed CSLR algorithm was evaluated with a synthetic dataset, an EEG-based motor imagery dataset, and an fMRI-based visual reconstruction dataset, respectively, and was compared to the baseline, the original SLR algorithm. 10^8 was used as the threshold concerning the relevance parameter $a_{\max}$ for both CSLR and SLR. The attributes were normalized such that each dimension was of mean 0 and variance 1 before utilizing the classification algorithms. The maximum iteration number for free energy maximization was set as 300 for both CSLR and SLR.

The kernel bandwidth h is an important hyper-parameter for the proposed CSLR. For the synthetic dataset, five-fold cross-validation method was utilized to select a proper kernel size for each condition from the following twenty values: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.2, 1.4, 1.6, 1.8, 2, 4, 7, 10, 30, and 100, which exhibited the highest average classification accuracy on the validation set. Then, each value was evaluated separately with the synthetic example by which the best value was decided and used for the subsequent real data analysis.

5.3.1. Synthetic Dataset

1) Dataset Description:

First, a noisy and high-dimensional synthetic dataset was considered with which CSLR and SLR algorithms were evaluated in respect of classification accuracy and feature selection. 300 i.i.d training samples and 300 i.i.d. testing samples are randomly generated with 500-dimensional multivariate standard normal distribution. The true solution was a 500-dimensional vector, where only the first five components were relevant to the label while the other 495 components were equal to zero

$$\mathbf{w}^* = [\overbrace{w_1^*, w_2^*, \cdots, w_5^*}^{\text{500-dimensional}}, \underbrace{0, 0, 0, 0, \cdots, 0}_{\text{495 components}}]^T \tag{126}$$

where the non-zero components were separately subject to the univariate standard normal distribution. For each sample, the label was assigned 1 if the product between the corresponding attribute and $\mathbf{w}^*$ was larger than 0, otherwise assigned 0. Thus, one is supposed to train the classifiers with 300 training samples by 500 features, and evaluate them on the other 300 testing samples.

Considering the contamination for this synthetic dataset, two corruption models were utilized according to [215], as shown in Fig. 21. The sample contamination indicates that undivided samples are corrupted while the arbitrary contamination means any arbitrary elements in the attribute matrix may be corrupted. To contaminate the data, a certain proportion of samples or elements are randomly selected and their attributes are replaced with the zero-mean Gaussian distributed noises with the following different standard deviations: 0.1, 0.3, 0.7, 1.0, 2.0, and 3.0. Similarly as before, only the 300 training samples were corrupted in this synthetic dataset. For both sample and arbitrary contamination, the proportion of the corrupted samples/elements was increased from 0 to 1.0 with a step 0.05.

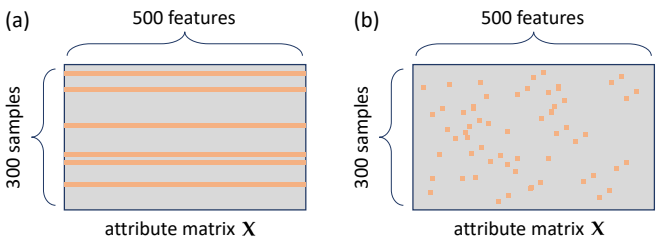

Figure 21. Two corruption models were utilized for the synthetic dataset: (a) sample contamination (b) arbitrary contamination. The attribute matrix $\mathbf{X}$ is the collection of all $\mathbf{x}_n$, each row of which represents an individual sample.

2) Results:

CSLR and SLR were evaluated with 100 Monte-Carlo repetitions on this synthetic dataset. First, the average classification accuracy for sample contamination and arbitrary contamination is illustrated in Fig. 22 (a) and (b), respectively. As one could observe in Fig. 22, for both sample contamination and arbitrary contamination, the proposed CSLR outperformed the original SLR algorithm significantly when the training data suffered corruption under each noise standard deviation.

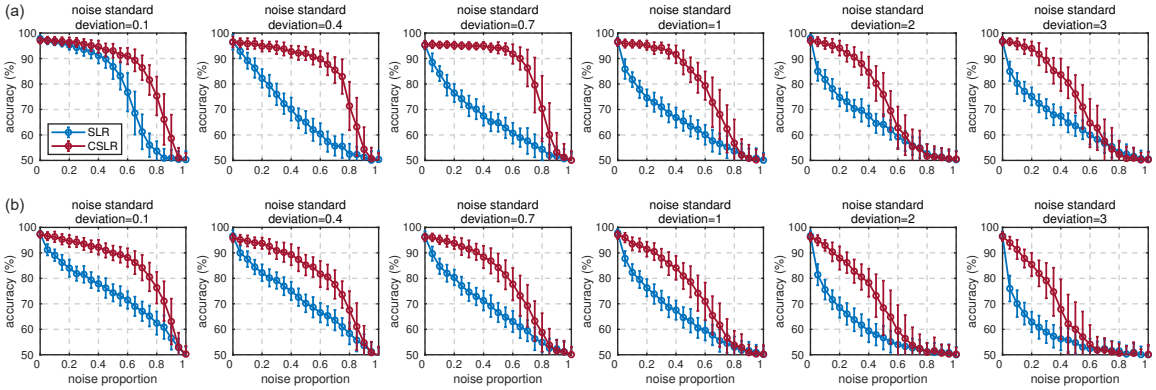

Figure 22. Classification accuracy on the noisy and high-dimensional synthetic example under two different contamination models: (a) sample contamination (b) arbitrary contamination. The results are averaged across 100 Monte-Carlo repetitions, where the error bar indicates the corresponding standard deviation.

In addition, the capability of each algorithm for feature selection was also evaluated. The feature selection can be regarded as an unbalanced binary classification, in which there were 5 relevant features and 495 irrelevant features. The sparse classifiers would select features in the model training, which would be regarded relevant to the classification task, while the other pruned features were considered to be irrelevant. Thus, one could evaluate the feature selection results with the true relevant/irrelevant labels. For this unbalanced binary classification, a comprehensive performance indicator, F1-score, was utilized which is the harmonic mean of *Precision* and *Recall*. The expressions of *Precision*, *Recall*, and F1-score are given by

$$\begin{aligned} Precision &= \frac{TP}{TP+FP} \\ Recall &= \frac{TP}{TP+FN} \\ F_1 &= 2 \times \frac{Precision \times Recall}{Precision + Recall} \end{aligned} \tag{127}$$

where the confusion matrix for the feature selection is shown in Fig. 23 for the calculations of *TP*, *FP*, and *FN*.

		predicted condition	
		relevant feature	irrelevant feature
truth	relevant feature	true positive (TP)	false negative (FN)
	irrelevant feature	false positive (FP)	true negative (TN)

Figure 23. Confusion matrix for feature selection, which exhibits an unbalanced binary classification (5 relevant features vs 495 irrelevant features). 'Relevant' is supposed as 'positive', while 'Irrelevant' is supposed as 'negative'.

Similarly, the identical synthetic dataset was used to evaluate the feature selection results with 100 Monte-Carlo repetitions. The number of selected features and F1-score for each algorithm are shown in Fig. 24 with sample and arbitrary corruption, respectively, where the noise standard deviation was set as 1.0. As one can see in Fig. 24, CSLR always selected fewer features than SLR with both sample and arbitrary contamination. More importantly, CSLR achieved a higher F1-score for the feature selection than SLR under most noise proportions.

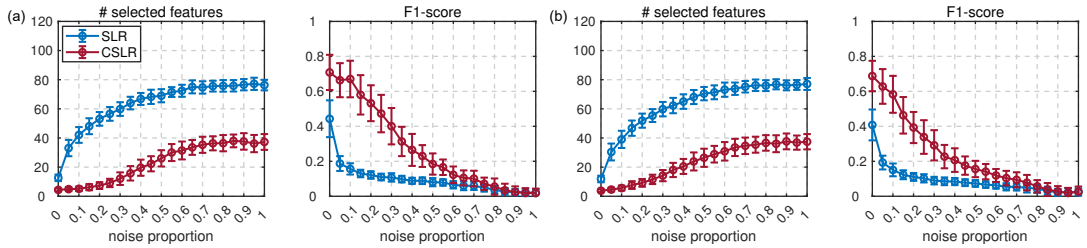

Figure 24. Number of selected features and F1-score for feature selection: (a) sample contamination (b) arbitrary contamination. The error bar denotes the standard deviation. The average results and the corresponding standard deviations are obtained from 100 Monte-Carlo repetitions. The noise standard deviation was set as 1.0.

To study the effects of the kernel bandwidth h, evaluated each candidate value was evaluated with sample and arbitrary contamination, respectively, where the noise standard deviation was set as 1.0. The average classification accuracy and feature selection F1-score of each kernel bandwidth are illustrated in Fig. 25, obtained from 100 Monte-Carlo repetitions. One could find an apparent effect of kernel bandwidth on the proposed CSLR. In particular, one could observe that the kernel bandwidth 0.4 achieved the highest classification accuracy with sample corruption, and the highest F1-score for feature selection in both sample and arbitrary contamination. However, in arbitrary contamination, when the kernel bandwidth is larger than 0.5, the classification accuracy achieved the maximum consistently. Therefore, a rather conservative kernel bandwidth 0.5 will be used for the subsequent real data analysis.

5.3.2. EEG-Based Motor Imagery Dataset

1) Dataset Description:

Ten healthy subjects were involved in this experiment. Their brain activities during the experiment were recorded by a 64-channel EEG recording system at 2048 Hz. A customized GVS instrument was used in parallel to induce the sensory feedback. In the experiment, the subjects were required to keep their eyes closed and to imagine the motions (forward/backward), with random voice-based cues. After 3-second cue period, the GVS instrument started to stimulate the subjects with four directions for 0.5 second: forward/backward/left/right. Then, the subjects rested for 3 seconds with a beep cue. A schematic diagram of one trial is illustrated in Fig. 26. Each subject participated in 6 sessions, where each session consisted of 60 trials. Thus, the data of each subject contained 360 trials. For each subject, the directions of the GVS-induced sensory feedback were identical with the imagined motion directions for 180 trials, while were inconsistent for the other 180 trials. One is supposed to predict whether the direction of motor imagery is consistent with the GVS-induced sensory feedback: match or mismatch. One can find a comprehensive description of this dataset in [195].

2) Decoding Paradigm:

To achieve the application for real-time BCI system, the EEG data was used for decoding analysis with a rather raw state for this experiment [194,195], where the absolute magnitudes of EEG recording were employed as the classification features. In this EEG decoding task, an identical decoding paradigm as in [194,195] was employed as follows.

1. First, the EEG recordings were down-sampled to 512 Hz.

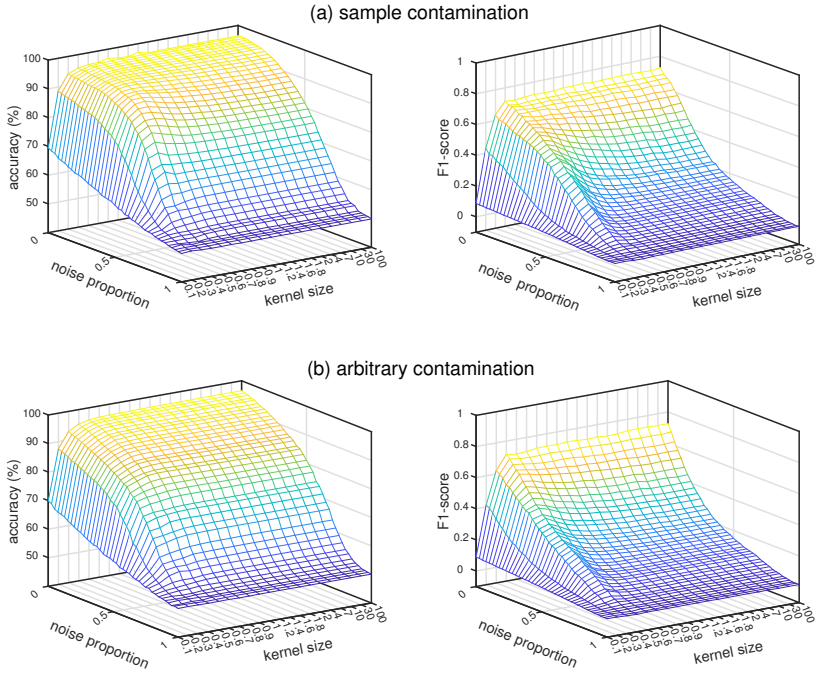

Figure 25. Classification accuracy and F1-score for feature selection of each kernel bandwidth with respect to the noise proportion: (a) sample contamination (b) arbitrary contamination. The results were averaged across 100 Monte-Carlo repetitions. The noise standard deviation was set as 1.0.

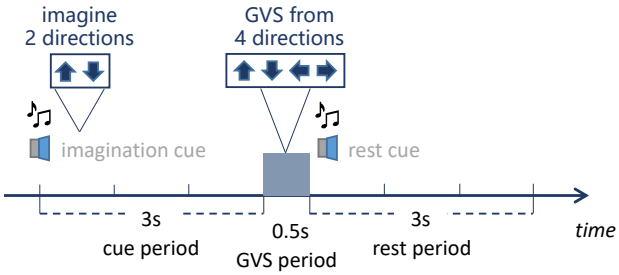

Figure 26. Schematic diagram of EEG-based motor imagery experiment with GVS-induced sensory feedback.

2. For each subject, the 360 trials were reordered randomly with their labels within respective class(match/mismatch).
3. The reordered dataset was separated, the first 80% defined as training, while the remaining 20% as testing. Thus, for either class, the first 144 trails were employed for training while the remaining 36 trails were utilized as testing data. The training and testing trials for each class were further combined to form the eventual dataset.
4. The absolute magnitude of the EEG recording during the GVS period was used as feature for each 100 ms duration (0–0.1 s, 0.1–0.2 s, 0.2-0.3 s, 0.3–0.4 s, 0.4–0.5 s).
5. As a result, there were 288 training samples (80%) and 72 testing samples (20%) with 3,264 features (64 channels $\times$ 51 samplings during 100 ms). The classifiers were trained on the training data and were evaluated on the testing set. Both training and testing data exhibited a balanced class distribution.
6. Step 2–Step 5 were implemented by 20 repetitions to summarize the results.

3) Results:

The average classification accuracy in each 100 ms decoding window is shown in Fig. 27 for each subject from 20 repetitions. The whiskers represent the corresponding standard deviations. It was also

examined whether there existed statistically significant difference between the accuracy obtained by CSLR and SLR, respectively, according to a paired t-test with $p < 0.01$. Among the 50 conditions in total (10 subjects$\times$5 decoding windows), the proposed CSLR achieved statistically higher classification accuracy in 44 conditions. Additionally, the average classification accuracy across a total of 10 subjects in each decoding window and across all decoding windows is shown in Table 8. The higher accuracy under each decoding window is marked in bold. One can observe that the proposed CSLR achieved higher average accuracy than SLR under each decoding window and across all the decoding windows as well with statistically significant difference.

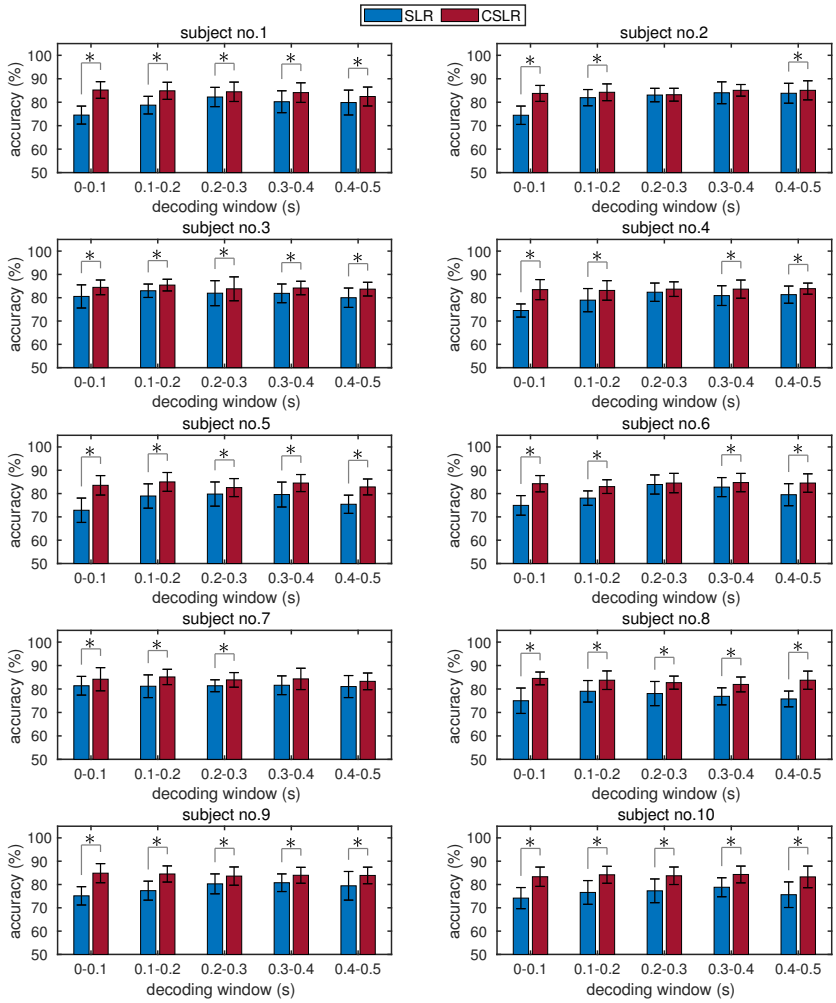

Figure 27. Classification accuracy on the EEG-based motor imagery dataset with GVS-induced sensory feedback. The results are averaged across 20 repetitions, where the whiskers denote the standard deviations. '$*$' indicates statistically significant difference according to a paired t-test ($p < 0.01$).

Furthermore, the the spatial patterns for the classification models acquired by CSLR and SLR were studied, respectively, by calculating how much each channel contributed to the whole classification model. The element of the trained model parameter can be denoted by $w_{\mathrm{ch,temp}}$ which corresponds to the EEG channel "ch" and the sampling time "temp". The spatial contribution for the channel "ch", denoted by $W(\mathrm{ch})$, is calculated by the ratio between the summation of the absolute values of model parameters, which correspond to the current channel, and the whole classification model

$$W(\mathrm{ch}) = \frac{\sum_{\mathrm{temp}} \|w_{\mathrm{ch,temp}}\|}{\sum_{\mathrm{ch}} \sum_{\mathrm{temp}} \|w_{\mathrm{ch,temp}}\|} \tag{128}$$

Table 8. Average classification accuracy across ten subjects in each decoding window and across all decoding windows with different classification algorithms. '$*$' indicates statistically significant difference according to a paired t-test ($p < 0.01$).

Classifier	EEG Channels	Decoding Window (s)					
		0-0.1 ($*$)	0.1-0.2 ($*$)	0.2-0.3 ($*$)	0.3-0.4 ($*$)	0.4-0.5 ($*$)	Average ($*$)
SLR	All 64 Channels	75.75±5.12	79.38±4.70	81.03±4.82	80.74±4.70	79.18±5.35	79.22±5.28
CSLR	All 64 Channels	**84.15±3.90**	**84.32±3.65**	**83.62±3.79**	**84.08±3.72**	**83.66±3.77**	**83.97±3.77**

The spatial contribution for each EEG channel is illustrated in Fig. 28 (a) for SLR and CSLR, respectively, averaged across a total of ten subjects and the five decoding windows. Furthermore, the top 5 EEG channels with the maximal spatial contributions are shown in black circles while the other top 16 EEG channels are presented in gray circles in Fig. 28 (b) and (c) for SLR and CSLR, respectively. To further demonstrate the superior capability of feature selection for CSLR, the EEG data from the respective top 5 channels was used under the same decoding paradigm with the *generic logistic regression* algorithm without any sparse priors. Average accuracy across all subjects is listed in Table 9. One can observe that even by the elementary MLE-based logistic regression without any sparse priors, the top 5 channels selected by CSLR showed significantly higher accuracy in three decoding windows and also showed significant difference for the average across all the decoding windows.

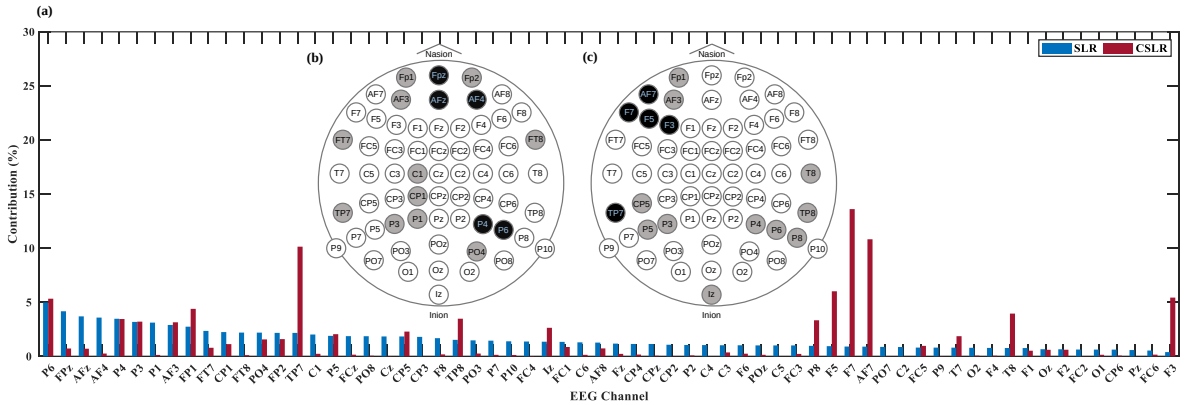

Figure 28. Spatial patterns obtained by SLR and CSLR in EEG-based motor imagery dataset with GVS-induced sensory feedback. The contribution of each EEG channel $W(\mathrm{ch})$ is shown in (a), sorted in descending order according to the spatial contribution by SLR. The top 5 EEG channels with the maximal spatial contributions are plotted in black circles, while the other channels with the top 16 spatial contributions are plotted in gray circles for SLR in (b) and CSLR in (c), respectively.

Table 9. Average classification accuracy across ten subjects in each decoding window and across all decoding windows with the *generic logistic regression* algorithm and the top 5 EEG channels selected by different sparse classifiers. '$*$' is statistically significant difference according to a paired t-test ($p < 0.01$).

Classifier	EEG Channels	Decoding Window (s)					
		0-0.1 ($*$)	0.1-0.2 ($*$)	0.2-0.3	0.3-0.4	0.4-0.5 ($*$)	Average ($*$)
generic logistic regression	top 5 channels selected by SLR	79.86±4.94	84.04±3.72	83.78±3.90	83.64±3.75	83.37±4.15	82.94±4.37
generic logistic regression	top 5 channels selected by CSLR	**81.10±4.07**	**84.24±3.82**	**83.84±3.89**	83.64±3.64	**83.69±3.91**	**83.30±4.04**

5.3.3. fMRI-Based Visual Reconstruction Dataset

1) Dataset Description:

The subject was watching visual images consisting of 10×10 square patches. Every patch was either a homogeneous gray area or flickering at 6Hz. The brain activity of the subject was recorded

simultaneously by fMRI signals. This dataset consists of 2 sessions: one random image session and one figure image session. In the random image session, the shown images were formed in stochastic patterns. 440 different random images in total were presented to the subject. Each stimulus block lasted 6s, followed with a 6s rest period. In the figure image session, there were 3 types of figure images: geometric, alphabet letter layout 1, and alphabet letter layout 2. Each type had 40 blocks totally. Each stimulus block lasted 12s, followed by 12s rest. For geometric images, five shapes were presented 8 times. For alphabet letter layout 1, five letters were presented 8 times. For alphabet letter layout 2, ten letters were presented 4 times. The preprocessing for fMRI data was identical to the original study. In the following analysis, V1 and V2 regions were utilized to reconstruct the images, in which 1,698 voxels in total were involved. More details of the experiment can be found in [33] and this dataset is publicly available http://brainliner.jp/data/brainliner/Visual_Image_Reconstruction.

2) Decoding Paradigm:

The random image session was used to train reconstruction models with different classifiers while the figure image session was utilized to evaluate the reconstruction performance. Similarly as in [33], a linear combination of local image decoders was used to reconstruct the 10×10 images by a local image basis of size 1×1. Accordingly, for a 10×10 image, 100 individual binary classifiers will be trained for each pixel, which predicted if the pixel was flickering or a gray area. Then the predicted contrast values of each pixel were combined as the final reconstruction with the combination coefficients, which were obtained by 10-fold cross-validation in the random image session. A total of 440 random images were separated equally into nine training groups and one validation group. The local decoders with 100 classifiers were trained on the nine training groups. Then, one could calculate the non-negative combination coefficients by minimizing the sum of the square error between the true and the predicted validation group. The eventual combination coefficients were averaged by the cross validations. Then, one can retrain the local decoders with all 440 random images and integrated them by the combination coefficients to reconstruct the visual stimulus for the figure image session.

3) Results:

CSLR and SLR were evaluated separately with the above-mentioned decoding paradigm. The reconstructed images of figure image session are illustrated in Fig. 29 (a) by each type, in comparison to the original visual stimulus. Further, the spatial correlation and mean squared error (mse) between the original and the reconstructed visual stimulus were quantified for each figure image category in Fig. 29 (b). One can observe that, the proposed CSLR achieved higher spatial correlation while lower mse than SLR with statistically significant difference according to a paired t- test with $p < 0.01$.

In addition, the feature selection was considered in this fMRI dataset as well. For the ultimate 100 local decoders which were trained by 440 random images to reconstruct the figure image, the number of selected features for each decoder (classifier) was counted. Further, for the selected features (voxels), the percentage of V1 voxels for each decoder was defined by computing the ratio between the number of the selected V1 voxels and that of all selected voxels. The results are shown by boxplot in Fig. 30. One observes that CSLR selected fewer features for fMRI decoding than SLR while it was more likely to select the voxels from the V1 region which has the largest contributions to the visual reconstruction task according to [33].

5.4. Discussion

The proposed CSLR was demonstrated by the experimental results to achieve higher classification accuracy in noisy and high-dimensional decoding tasks. In the synthetic dataset, CSLR realized nearly identical classification accuracy as SLR when there was no contamination in the data. After artificial contamination was added in the attribute matrix, especially when the noise standard deviation is larger than 0.4 for sample contamination and all the noise standard deviations for arbitrary contamination, one sees in Fig. 22 that significant performance degradation happened to SLR even though the noise proportion is equal to 0.05. In contrast, the proposed CSLR realized much less performance degradation than SLR when the dataset was corrupted. Overall, in this noisy high-dimensional toy dataset, except

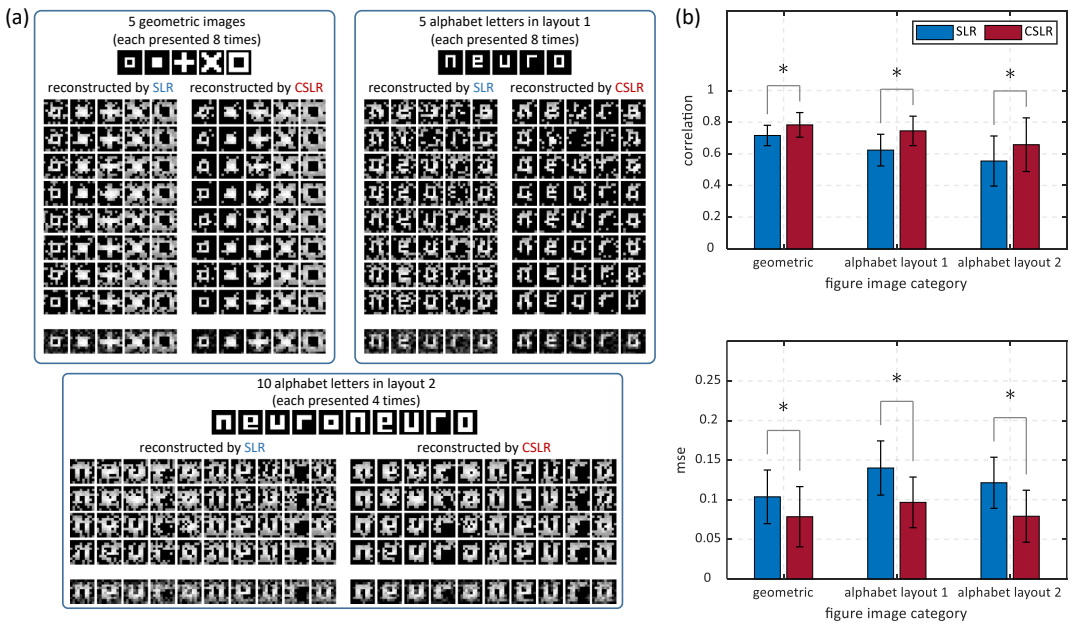

Figure 29. Diagrammatic results on the fMRI-based visual stimulus reconstruction dataset: (a) The comparison between the original and the reconstructed visual stimulus by CSLR and SLR, respectively, for three different categories in the figure image session. For each category, a total of 40 images were presented to the subject and then reconstructed by the fMRI signals. The bottom rows illustrate the average reconstructed visual images for each kind of the presented figure images. (b) The spatial correlation (upper) and mean squared error (bottom) between the original and the reconstructed visual stimulus for each category, averaged across the corresponding 40 stimulus blocks. The error bars indicate the standard deviations. '$*$' means statistically significant difference according to a paired t-test with $p < 0.01$.

for when the noise proportion is equal to zero or close to 1.0, the proposed CSLR almost always showed better results than SLR. Next, for the EEG data, CSLR achieved statistically higher accuracy than SLR for 44 conditions among a total of 50 scenarios (10 subjects $\times$ 5 decoding windows), while also realized higher average accuracy in the remaining 6 conditions though without significant difference (Fig. 27). In summary, the average classification accuracy for all subjects and decoding windows was improved by 4.75% (Table 8). Finally, one can observe from the fMRI-based visual reconstruction results in Fig. 29 that, the reconstructed images by CSLR are usually more legible and closer to the original stimulus. Quantitatively, CSLR achieved higher spatial correlation while lower mse than SLR with statistically significant difference.

On the other hand, CSLR can select a more informative set of features. In Fig. 24, the number of selected features by SLR is 12.66$\pm$2.25 without contamination, while is 4.40$\pm$0.96 for CSLR, which is closer to the true number of the five relevant features. Meanwhile, CSLR achieved considerably higher F1-score in feature selection (0.708$\pm$0.101 vs 0.443$\pm$0.105) than SLR. When the data was contaminated, the number of selected features and F1-score were obviously affected for SLR, while CSLR effectively suppressed the negative effects of corruption by comparison. For the EEG dataset, one could see the spatial patterns in Fig. 28. Both SLR and CSLR assigned large weights for mainly three Brodmann areas: BA9 (dorsolateral prefrontal cortex, e.g. AF3, AF4, AFz, F3, F5), BA10 (anterior prefrontal cortex, e.g. Fp1, Fp2, Fpz, AF7) and BA39 (angular gyrus, e.g. P3, P4, P5, P6). These regions are all potentially related to the GVS-based prediction error decoding in which BA9 and BA10 are involved in cognitive processes, while BA9 and BA39 are responsible for spatial imagery [216,217]. It would be difficult to determine which spatial pattern is more precise because the GVS-based prediction error decoding is a rather new paradigm and the physiological mechanism has yet been fully explored. To investigate

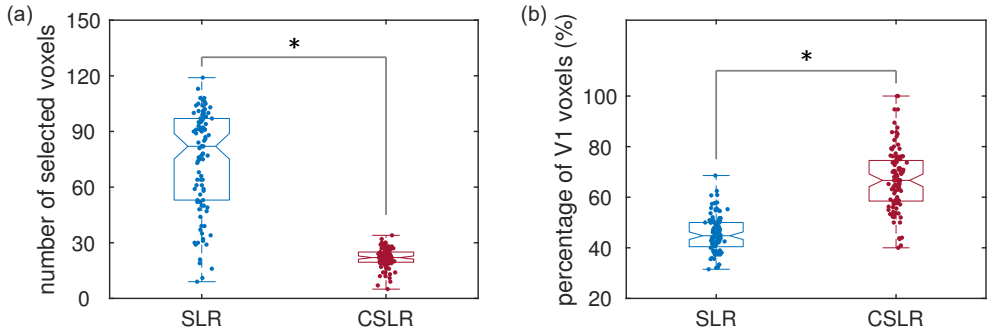

Figure 30. Feature selection for the eventual 100 local decoders: (a) number of selected voxels (b) percentage of V1 voxels. '$*$' means statistically significant difference according to a paired t-test with $p < 0.01$.

which EEG spatial pattern is more meaningful, the top 5 EEG channels by SLR and CSLR were assessed by a *generic logistic regression* model, respectively. Even though by a totally identical classification algorithm, the top 5 EEG channels from CSLR achieved significantly higher accuracy in three decoding windows and for the average across all decoding windows (Table 9), which suggests that the top 5 channels of CSLR contained more valid information for the decoding task. For the fMRI dataset, the voxels from V1 and V2 regions were utilized for visual decoding. As reported in [33], only using the V1 voxels revealed the best reconstruction, indicating that V1 region contains the most dependable information for visual reconstruction. Hence, for a good feature selection, V1 voxels should be the majority in the selected voxels. Compared to SLR, the proposed CSLR selected a significantly higher percentage of V1 voxels (Fig. 30), which means CSLR was more likely to select the more informative V1 voxels than SLR.

5.5. *Rethinking the Data Assumption under MCC*

Although MCC has successfully realized a significant improvement on robustness for ARD-based SLR algorithm for sparse classification, it would be insightful to investigate why MCC can be utilized as a robust substitute for the conventional non-robust likelihood functions. To make such an investigation, here, consider the MCC objective function for linear regression model with a finite dataset $\{(\mathbf{x}_n, t_n)\}_{n=1}^N$

$$\begin{aligned} \mathbf{w} &= arg\max_{\mathbf{w}} \frac{1}{N}\sum_{n=1}^{N} \exp(-\frac{(t_n - \mathbf{x}_n\mathbf{w})^2}{2h}) \\ &= arg\max_{\mathbf{w}} \frac{1}{N}\sum_{n=1}^{N} \exp(-\frac{e_n^2}{2h}) \end{aligned} \tag{129}$$

Since the denominator N will be fixed by a known dataset, it can be omitted. By doing so, one can find that MCC will be equivalent to a multiplication form through an exponential function

$$\begin{aligned} \mathbf{w} &= arg\max_{\mathbf{w}} \sum_{n=1}^{N} \exp(-\frac{e_n^2}{2h}) \\ &= arg\max_{\mathbf{w}} \prod_{n=1}^{N} \exp\{\exp(-\frac{e_n^2}{2h})\} \\ &= arg\max_{\mathbf{w}} \prod_{n=1}^{N} \exp\{\exp(-\frac{(t_n - \mathbf{x}_n\mathbf{w})^2}{2h})\} \end{aligned} \tag{130}$$

which can be extraordinarily regarded as a likelihood function maximum if one assumes independence for each t_n and defines the following PDF for the noise distribution

$$\mathcal{C}(e|0,h) \triangleq \exp\{\exp(-\frac{e^2}{2h})\} \tag{131}$$

in which $\mathcal{C}(e|0,h)$ is defined as a correntropy-aware PDF over e with the zero mean and the shape parameter h. Utilizing such an assumption on the noise distribution, one can obtain the PDF of t by $p(t|\mathbf{x}) = \mathcal{C}(t|\mathbf{x}\mathbf{w},h)$. Hence, assuming the independence for t_n, one can find the MLE based on the defined PDF $\mathcal{C}$ will be equivalent to the original MCC by Eq.(130).

It is important to discuss the property of the defined error assumption $\mathcal{C}(e|0,h)$. Unsurprisingly, it is not a "well-defined" PDF since one sees that its integral is infinite, thus, being an *improper* distribution [211]. Even more, when e is far from the origin, the probability density defined by $\mathcal{C}(e|0,h)$ is close to 1, rather than a normal case 0, which seems to be a *deviant* PDF. Nevertheless, it is empirically verified that such a *deviant* MCC-aware noise distribution can largely improve the robust property for an ARD-based sparse regression model. Some examples for $\mathcal{C}(e|0,h)$ are shown in Fig. 31 with different h values. A further discussion for this *deviant* noise assumption is given in Section 5.5.3.

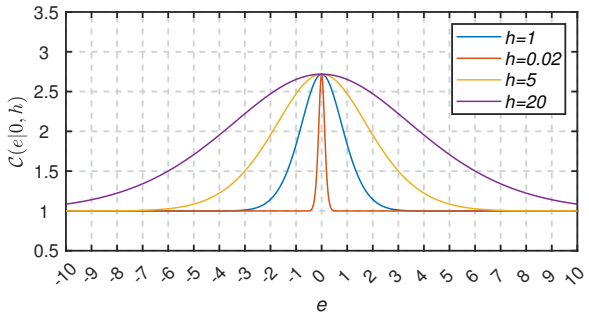

Figure 31. MCC-aware noise distribution $\mathcal{C}(e|0,h)$ with different h values.

5.5.1. MCC-ARD for Robust Sparse Regression

Based on the MCC-aware noise assumption $\mathcal{C}(e|0,h)$, one can derive the MCC-based regression with the ARD technique under a Bayesian inference framework for high-dimensional case. In detail, based on Eq.(131), the likelihood function can be written by

$$\begin{aligned} p(\mathbf{t}|\mathbf{w},h) &= \prod_{n=1}^{N} \mathcal{C}(t_n|\mathbf{x}_n\mathbf{w},h) \\ &= \prod_{n=1}^{N} \exp\{\exp(-\frac{(t_n-\mathbf{x}_n\mathbf{w})^2}{2h})\} \end{aligned} \tag{132}$$

Similarly, one can find that the utilization of the MCC-aware likelihood function in Eq.(132) would obstruct the analytical derivation for the posterior distribution $p(\mathbf{w}|\mathbf{t},\mathbf{a},h)$, since MCC-aware likelihood function is not conjugate with the Gaussian priors $p(\mathbf{w}|\mathbf{a})$ of Eq.(97). Therefore, to realize the MAP estimation, one can adopt the variational inference approach as in Section 5.2.

In detail, to realize a similar variational inference as in Eq.(115), one can first write the following log joint distribution by

$$\begin{aligned} &\log p(\mathbf{w},\mathbf{a},\mathbf{t},h) = \log p(\mathbf{t}|\mathbf{w},h) + \log p(\mathbf{w}|\mathbf{a}) + \log p(\mathbf{a}) \\ &= \sum_{n=1}^{N} \exp(-\frac{(t_n-\mathbf{x}_n\mathbf{w})^2}{2h}) - \frac{1}{2}\mathbf{w}^T\mathbf{A}\mathbf{w} - \frac{1}{2}\log|\mathbf{A}| + const \end{aligned} \tag{133}$$

Gathering the relevant terms with respect to $\mathbf{w}$ and $\mathbf{a}$, one then obtains

$$\begin{aligned} \log q_{\mathbf{w}}(\mathbf{w}) &= \sum_{n=1}^{N} \exp(-\frac{(t_n - \mathbf{x}_n\mathbf{w})^2}{2h}) - \frac{1}{2}\mathbf{w}^T \mathbb{E}_{q_{\mathbf{a}}(\mathbf{a})}[\mathbf{A}]\,\mathbf{w} \\ \log q_{\mathbf{a}}(\mathbf{a}) &= -\frac{1}{2}\sum_{d=1}^{D} a_d \mathbb{E}_{q_{\mathbf{w}}(\mathbf{w})}\left[w_d^2\right] - \frac{1}{2}\sum_{d=1}^{D} \log a_d \end{aligned} \tag{134}$$

Similarly, the Laplacian approximation is necessary for the following approximation on $\log q_{\mathbf{w}}(\mathbf{w})$

$$\log q_{\mathbf{w}}(\mathbf{w}) \approx \log q_{\mathbf{w}}(\mathbf{w}^*) - \frac{1}{2}(\mathbf{w} - \mathbf{w}^*)^T H(\mathbf{w}^*)(\mathbf{w} - \mathbf{w}^*) \tag{135}$$

in which the negative Hessian matrix of $\log q_{\mathbf{w}}(\mathbf{w})$, denoted by $H(\mathbf{w})$, is different from the logistic regression model, which is expressed by

$$H(\mathbf{w}) = -\frac{\partial^2 \log q_{\mathbf{w}}(\mathbf{w})}{\partial \mathbf{w} \partial \mathbf{w}^T} = -\frac{1}{h}\sum_{n=1}^{N} \mathbf{x}_n^T \left\{ \exp(-\frac{e_n^2}{2h})(\frac{e_n^2}{h} - 1) \right\} \mathbf{x}_n + \mathbb{E}_{q_{\mathbf{a}}(\mathbf{a})}[\mathbf{A}] \tag{136}$$

One can observe that w-step in MCC-ARD regression is actually equal to L_2-regularized MCC-based regression with the current $\mathbf{a}$ values of Eq.(134), which can be effectively optimized by the fixed-point update with fast convergence [81]

$$\mathbf{w} = (\mathbf{X}^T \Psi \mathbf{X} + \mathbf{A})^{-1} \mathbf{X}^T \Psi \mathbf{t} \tag{137}$$

in which Ψ is a $N \times N$ diagonal matrix with the diagonal element $\Psi_{nn} = \exp(-e_n^2/2h)$. After obtaining the maximum point $\mathbf{w}^*$ for $\log q_{\mathbf{w}}(\mathbf{w})$, the relevance parameters $\mathbf{a}$ could be optimized identically as in Section 5.2. Thus, MCC-ARD for robust sparse regression is summarized in Algorithm 5.

Algorithm 5 MCC-ARD for robust sparse regression

input:
  training samples $\{(\mathbf{x}_n, t_n)\}_{n=1}^N$;
  Gaussian kernel bandwidth h;
  threshold for relevance parameter $a_{\max}$;
initialize:
  model parameter w_d $(d = 1, \cdots, D)$;
  relevance parameter a_d $(d = 1, \cdots, D)$;
output:
  model parameter w_d $(d = 1, \cdots, D)$
repeat
  w-step: update **w** according to Eq.(137);
  a-step: update **a** according to Eq.(125);
  if $a_d \geqslant a_{\max}$ **then**
    set the corresponding w_d to zero and prune this dimension in the following updates
  end if
until the number of iterations is larger than an upper limit or the parameter change is small enough

5.5.2. Simulations

The proposed MCC-ARD algorithm for robust sparse regression was evaluated by a noisy and high-dimensional synthetic dataset, and compared with the conventional ARD-based sparse regression as introduced in Section 5.1 (denoted by LS-ARD). It was also compared with the L_1-regularized MCC [202–204] (MCC-L_1) optimized with an EM method [186,189]. The kernel bandwidth h for both MCC-

ARD and MCC-L_1 are selected by cross validation, while the latter utilizes another cross validation for regularization parameter λ. The pruning threshold $a_{\max}$ is set as 10^6 for both LS-ARD and MCC-ARD.

A noisy and high-dimensional synthetic dataset was generated with the following method. 300 i.i.d training samples and 300 i.i.d testing samples were generated randomly by the 1000-dimensional standard normal distribution. The target variable was obtained by the linear regression assumption and a sparse true solution $\mathbf{w}^*$ as

$$\mathbf{w}^* = [\overbrace{w_1^*, w_2^*, \cdots, w_{30}^*, \underbrace{0, 0, 0, 0, \cdots, 0}_{970 \text{ components}}}^{1000\text{-dimensional}}]^T \tag{138}$$

which is a 1000-dimensional vector where only the first 30 dimensions are non-zero components while the other 970 components are zero. The non-zero elements were randomly generated from the standard normal distribution. To assess the robustness of each algorithm, the following noise distribution is used on the model output

$$\epsilon \sim (1-\theta)\mathcal{N}(\epsilon|0, 0.05) + \theta\mathcal{L}(\epsilon|0, \tau) \qquad (0 \leqslant \theta \leqslant 1) \tag{139}$$

in which $\mathcal{L}(\epsilon|0, \tau)$ denotes the Laplace distribution over ϵ with zero mean and the scale parameter τ to imitate outliers, and θ means the proportion of outliers among the additive noise. The following values were considered for the outlier scale parameter τ: 2, 5, and 10, indicating increasing strengths for the outliers. The outlier proportion θ is increased from 0 to 1.0 with a step 0.05. The regression performance is evaluated by two classical regression performance indicators, correlation coefficient (r) and root mean squared error (RMSE), as defined in Eq.(83-84). The regression performance of each algorithm with 100 Monte-Carlo repetitions is presented in Fig. 32. One could observe that, the proposed MCC-ARD outperforms the conventional LS-ARD largely by significantly higher r and lower RMSE, when the high-dimensional data is contaminated by the non-Gaussian noises under each scale parameter τ. One further perceives that the proposed MCC-ARD achieves higher r than the existing MCC-L_1 under each scale parameter τ, and lower RMSE for $\tau = 2$ and 5. MCC-ARD and MCC-L_1 realize similar RMSE when $\tau = 10$. When τ becomes larger than 10, the conclusion of performance comparison is analogous to the case when τ is equal to 10. Note that, the proposed MCC-ARD method only has one hyperparameter h to be tuned, while MCC-L_1 needs to tune two vital hyper-parameters, namely, the kernel size h and the regularization parameter λ.

On the other hand, the feature selection for this synthetic dataset is also considered in the presence of outliers. Similarly as in Section 5.3.1, the feature selection problem can be regarded as an unbalanced classification task with the relevant/irrelevant label for each feature. The F1-score as defined in Eq.(127) was used to evaluate the quality for feature selection. Fig. 33 illustrates the number of selected features and F1-score of feature selection for each algorithm.

One observes that when the data is contaminated by the outliers, the number of selected features by MCC-ARD is closer to the ground truth of 30 relevant features, compared with the conventional LS-ARD and existing MCC-L_1. Notably MCC-ARD reveals significantly higher F1-score in the feature selection than other two algorithms in the presence of outliers, showing exceptional feature selection capability in a noisy and high-dimensional scenario. Even more, MCC-ARD also gives higher F1-score without outlier contamination (proportion=0).

5.5.3. Discussion: MCC-Aware Noise Assumption

It is indispensable to discuss if the MCC-aware noise assumption $\mathcal{C}(e|0, h)$ of Eq.(131) is adequate to be utilized in a robust regression model from a Bayesian perspective. Conventionally, an *improper* distribution, referring to a non-normalizable PDF, can be only permitted for a prior distribution (and the resultant posterior distribution) in a traditional Bayesian regime [211]. The likelihood function (equally the noise distribution) has for the first time been supposed with such a *deviant* distribution

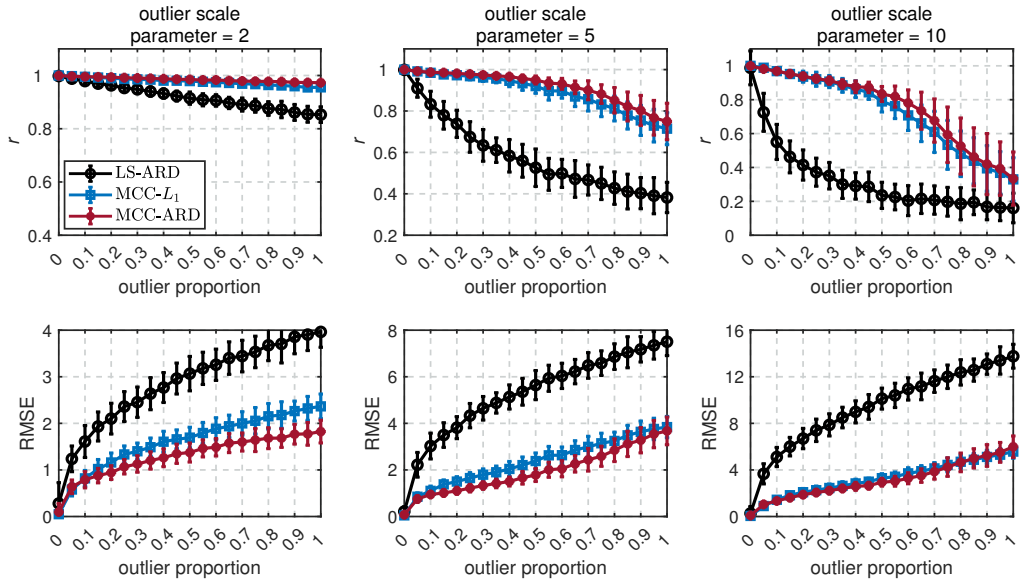

Figure 32. Correlation coefficient (r) and root mean squared error (RMSE) with the noisy and high-dimensional dataset under different outlier proportions and scale parameters.

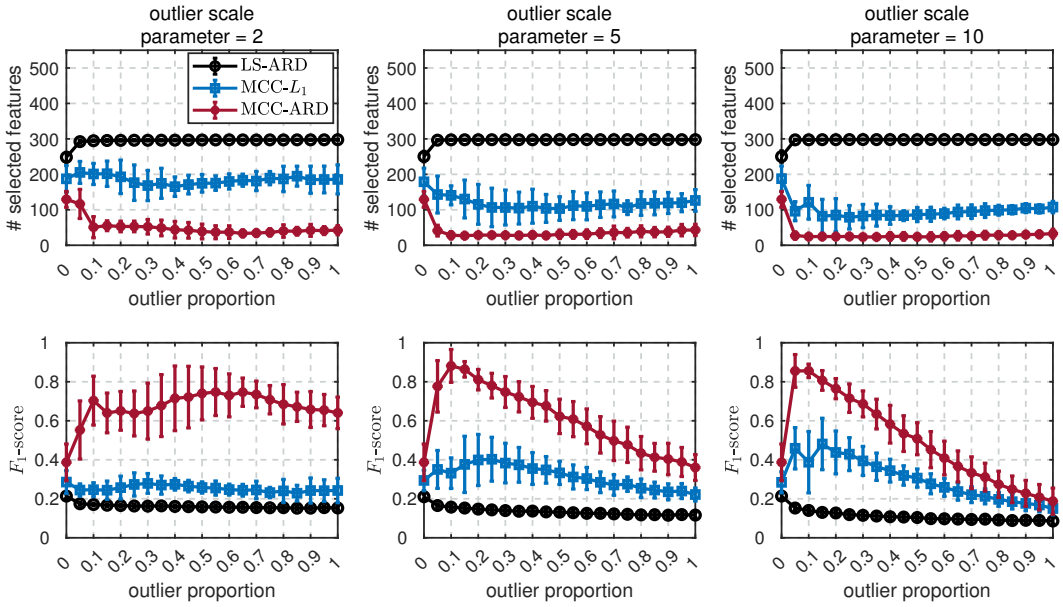

Figure 33. Number of selected features and F1-score of each regression algorithm.

$\mathcal{C}(e|0,h)$, which does not even converge to 0 far from the origin. To verify the validity of such a *deviant* noise assumption, one can define the following noise distribution

$$\mathcal{C}'(e|0,h) \triangleq \exp\{\exp(-\frac{e^2}{2h})\} - 1 \tag{140}$$

which is a simple translation of $\mathcal{C}(e|0,h)$ towards the horizontal axis, and can be proved a normalizable PDF by elementary derivation, as illustrated in Fig. 34. By this *proper* noise distribution, one can make a similar derivation as in Section 5.5.1, and compare the experimental results utilizing the identical synthetic dataset from Section 5.5.2. As is illustrated in Fig. 35, for each outlier scale parameter, the *deviant* MCC-ARD outperforms evidently the *proper* one. In particular, when the outlier scale parameter is 10, the *proper* MCC-ARD even achieves similar results with the conventional LS-ARD, showing poor robustness compared with the *deviant* one. Therefore, the validity of the MCC-aware *deviant* noise

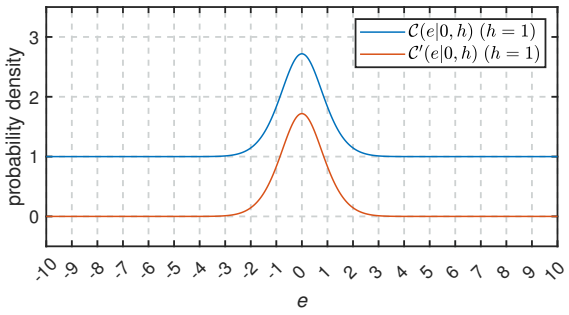

Figure 34. Comparison between *deviant* $\mathcal{C}(e|0,h)$ and *proper* $\mathcal{C}'(e|0,h)$.

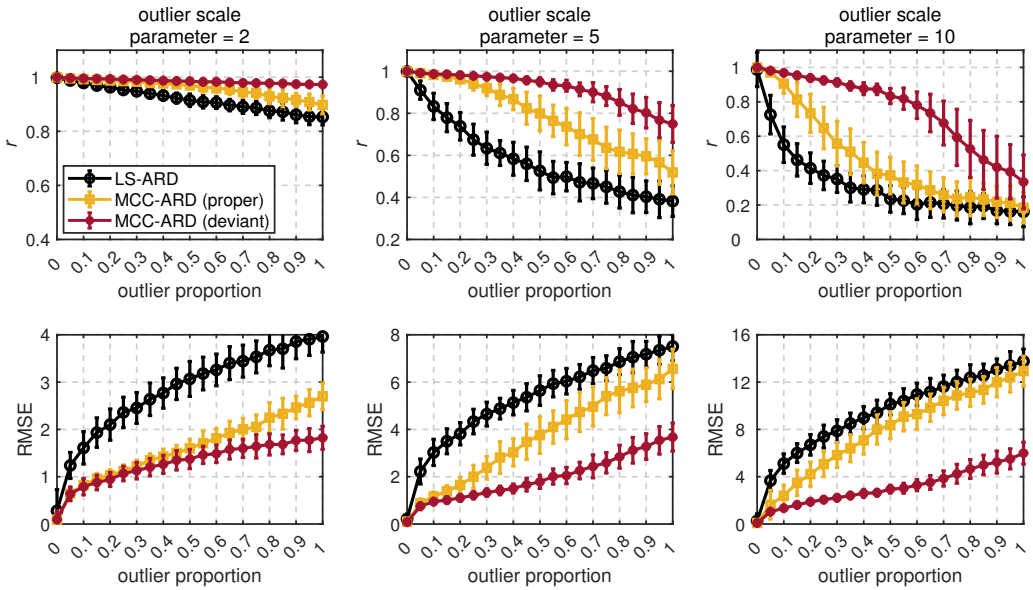

Figure 35. Correlation coefficient (r) and root mean squared error (RMSE) for the MCC-ARD regression algorithms which are derived by the *proper* $\mathcal{C}'(e|0,h)$ and the *deviant* $\mathcal{C}(e|0,h)$, respectively.

distribution $\mathcal{C}(e|0,h)$ of Eq.(131) is empirically proved. The robustness of $\mathcal{C}(e|0,h)$ can be interpreted heuristically as follows.

The prominent characteristic of the *deviant* $\mathcal{C}(e|0,h)$ is that, its probability density acquires the maximum at the origin while it converges to 1 when $e \to \infty$. In a usual noise assumption, e.g. Gaussian distribution, the probability density converges to 0 when e is arbitrarily large, which seems to be a reasonable hypothesis. However, if a dataset is in particular prone to adverse outliers, this hypothesis would be unreliable, because some errors with large values do happen, indicating non-zero probability density even though far from the origin. By comparison, the *deviant* $\mathcal{C}(e|0,h)$ exactly assumes non-zero density for the arbitrarily large error. Thus, it can be argued that the MCC-aware $\mathcal{C}(e|0,h)$ is a more rational noise assumption when the dataset is prone to outliers, as demonstrated by the simulation results.

6. Conclusion & Future Works

6.1. Conclusion

This thesis aims to realize a better brain activity decoding performance by addressing the problem where the conventional learning criteria for machine learning may be significantly deteriorated by the non-Gaussian noises or outliers inherent in the brain recordings from existing measurement techniques. The main motivation of this thesis is the ITL framework, which adopts information-theory descriptors

to formulate the objective functions for machine learning models. In particular, two popular learning criteria, namely MEE and MCC, were utilized in this thesis to propose robust brain decoding algorithms to improve the decoding performance for real-world noisy brain recordings. First, this thesis considers the noisy classification task. It was found that the optimal error distribution for a noisy classification scenario exhibits a three-peak distribution, for which the original MEE (or QMEE) is supposed to reveal satisfactory robustness, whereas they showed unexpected instability. By investigating the reason, this thesis proposed a new learning criterion for robust classification, which is a special case of QMEE with a restricted codebook, thus named by RMEE. For the proposed RMEE, the discussions for optimization and convergence analysis are given. In performance evaluation, first, RMEE based logistic regression showed better robustness in the synthetic dataset. For noisy EEG datasets, RMEE based ELM achieved the highest accuracy in most cases. In addition, RMEE based ELM realized promising performance in other benchmark datasets as well. Then, this thesis also takes another issue for brain activity decoding, the high-dimensional problem, into account, by studying how to embed the robust ITL approach into the existing algorithms for the high-dimensional brain decoding task. First, this thesis investigated a robust implementation for the dimensionality reduction based decoding algorithm, for which a novel robust variant for PLSR algorithm was proposed by reformulating the non-robust least square criterion by the sophisticated MCC framework. The proposed PMCR algorithm implements the decomposition for input and output simultaneously, and acquires each model parameter with MCC. The experimental results with the synthetic dataset and Neurotycho ECoG dataset demonstrate that, the proposed PMCR could outperform the existing PLSR algorithms, revealing promising robustness for high-dimensional and noisy ECoG decoding. Subsequently, this thesis discussed the integration of MCC with the feature selection strategy to realize robust and sparse brain decoding. To be specific, MCC was integrated with the sparse Bayesian learning approach and ARD method for adaptive sparseness. The proposed CSLR algorithm was evaluated on different noisy and high-dimensional classification scenarios, including a toy example, the EEG decoding task, and the fMRI-based visual decoding task. Experimental results demonstrated that CSLR can realize better classification accuracy and feature selection for brain activity decoding tasks. Furthermore, this thesis exposed the inherent noise assumption under the MCC-based regression and derived an explicit MCC-aware noise assumption $\mathcal{C}(e|0,h)$. By integrating this MCC-aware noise assumption and the ARD method, MCC-based robust regression can be also implemented with the "*adaptive sparseness*". The proposed MCC-ARD algorithm for robust sparse regression realized superior regression performance and feature selection in a noisy and high-dimensional scenario. The corresponding works presented in this thesis were published in [136,218–220]. The proposed methods are summarized in Fig. 36 with their applicable conditions.

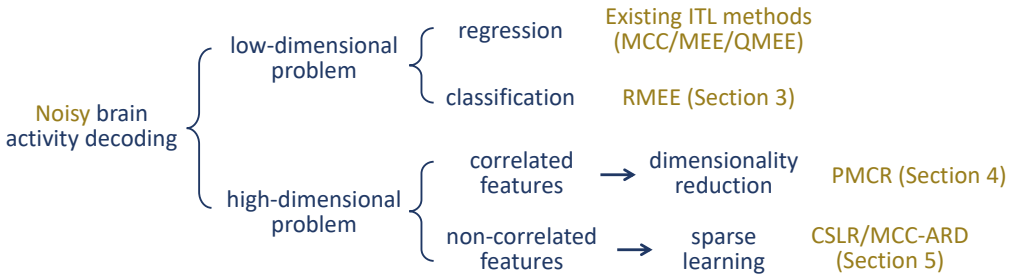

Figure 36. Summary of the proposed methods with their applicable conditions.

6.2. *Future Works*

Further improvements for the proposed algorithms in this thesis will be also discussed as in what follows.

For the proposed RMEE learning criterion for robust classification, the first interesting future work is how to estimate a more accurate estimation of the real outlier proportion without any prior information. Although the empirical method utilized in Section 3.3.3 realized a rather accurate result,

it needs to be analyzed with more theoretical guarantees. As categorized in [78], classifiers are divided into regression-like and non-regression-like ones where prediction error is of continuous and discrete value, respectively. For the regression-like classifiers, such as a wide variety of neural network models for classification, it is argued that the proposed RMEE could be a promising alternative for those tasks prone to severe noises, since its effectiveness has been verified on the ELM model. On the other hand, the implementation of RMEE for non-regression-like classifiers needs further exploration. For example, in the decision trees and the $\{0,1\}$-label context, the prediction is discrete 0 or 1, and hence one obtains discrete error $e \in \{0,-1,1\}$, but not $e \in (-1,1)$ that belongs to a continuous interval as in this thesis. Whether the proposed RMEE can achieve satisfactory performance for non-regression-like classifiers requires further studies.

For the proposed PMCR algorithm, it exhibits the supplementary potential for further performance improvements with regularization techniques, as well as in the existing regularized PLSR algorithms. For example, L_1-regularization could be utilized in Eq.(72) to encourage sparse and robust projectors. In addition, if one requires better smoothness on the predicted output, polynomial or Sobolev-norm penalization could be utilized in PMCR. Moreover, L_2-regularization could be utilized for Eq.(74) to decrease the over-fitting risk considering the regression scalar b_s. In addition, the multi-way PLSR is an important generalization for this algorithm, which establishes the regression relationship between tensor variables with dimensionality reduction by tensor factorization technique. In the literature, the multi-way PLSR was usually reported to achieve superior decoding capability than the generic PLSR algorithm in the brain decoding task, where the spatio-spectro-temporal feature is organized with the tensor form. Essentially, the multi-way PLSR decomposes the input and output under the least square criterion by minimizing the Frobenius-norm [221]. Therefore, the multi-way PLSR is also prone to the performance deterioration caused by noises. Extending the PMCR algorithm to multi-way application can probably improve the prediction performance further. Promisingly, MCC has been demonstrated effective for tensor variable analysis in a recent study [222].

Another fundamental problem regarding the performance improvement for noisy brain activity decoding is the brain recording noise. Although the clear definition and statistical properties for the brain recording noise could help to develop more advanced denoising or robust decoding algorithms, it would be difficult to give a conclusion regarding the properties of the brain recording noise, because different kinds of artifacts may happen at the same time, thus being fused with each other. In addition, some experiments might be more prone to artifacts since eye movements or muscle movements are involved in some tasks. A common way to evaluate the effectiveness of robust decoding algorithms for noisy brain recordings is to artificially contaminate the brain data with outliers, because the real-world artifacts usually exhibit larger amplitudes than normal samplings [87,89,163]. Similar to the relevant literature, Section 3 and Section 4 utilized artificial outliers to contaminate the brain recordings for the performance comparison across different algorithms. However, this contamination might be inaccurate to simulate real-world brain recording noises. Therefore, in Section 5, this thesis employed the original brain recordings directly, without any artificial contamination. This further emphasizes the necessity and effectiveness of robust decoding for real-world noisy brain data analysis. In future works, direct evaluation on original brain recordings should be considered first for performance comparison across different algorithms.

In addition, two other important discussions are presented as follows for the future works.

6.2.1. Multi-Class Classification

This thesis proposed two robust classification methods. The first method is the RMEE learning criterion which can be used in various classification models, while the second one is the CSLR algorithm which is effective for robust and sparse classification in the high-dimensional scenario. However, these two methods were only considered for the binary classification in this thesis which may seriously limit their applications for real-world scenarios. Therefore, a discussion for their generalization to the multi-class case is given as follows.

Consider the multi-class cases in which there exist C different potential labels to be classified. One is supposed to utilize individual models parameters for each class. To be specific, the class c will have an individual linear discriminant function

$$f_c(\mathbf{x}, \mathbf{w}^c) = \sum_{d=1}^{D} w_d^c x_d = \mathbf{x}\mathbf{w}^c \quad (c = 1, \cdots, C) \tag{141}$$

where $\mathbf{w}^c$ denotes the model parameter for c-th class. Thus, one can calculate the probability that the n-th sample belongs to c-th class through the *softmax* function by

$$y_n^c \triangleq p(t_n = c) = \frac{\exp(f_c(\mathbf{x}, \mathbf{w}^c))}{\sum_{k=1}^{C} \exp(f_k(\mathbf{x}, \mathbf{w}^k))} \quad (c = 1, \cdots, C) \tag{142}$$

In multi-class case, one commonly utilizes the *one-hot* coding for label expression, e.g. $t_n = (1, 0, \cdots, 0)$ if n-th sample belongs to the first class. Similarly, one can obtain the prediction by $y_n = (y_n^1, \cdots, y_n^C)$. Thus, the prediction error becomes a C-dimensional vector by subtraction $e_n = t_n - y_n \in \mathbb{R}^C$. Extended from binary case, one can imagine that in multi-class cases errors are distributed on a high-dimensional cube ranging between $(-1, 1)$. In this way, the implementations of RMEE and CSLR for a multi-class scenario can refer to those studies that applied MEE or MCC to multi-dimensional errors. Multi-class classifiers can further improve the brain activity decoding performance. For example, multiscale local image decoders were proved to show better visual reconstruction results in [33] than only utilizing the decoder of 1×1 size, by combining 1×1, 1×2, 2×1, and 2×2 decoders. Since the decoders for the other scales require multi-class classification, in the future works, multi-class CSLR algorithm will be proposed and implemented with the multiscale decoders.

6.2.2. Determination for Kernel Bandwidth

Another important topic for MEE and MCC is the kernel bandwidth determination. In this thesis, the kernel bandwidth h is selected by the cross validation method or computed by the *Silverman's rule*. Cross validation may be time-consuming if the dataset exhibits a large size, while *Silverman's rule* may lead to a less proper kernel bandwidth as mentioned in [202,203]. To explore a better way to determine the kernel bandwidth, the Bayesian learning framework may be a good motivation.

To be specific, it will be interesting to investigate how to treat the kernel bandwidth as a random variable so that one can integrate h with the Bayesian perspective and optimize it automatically in the process of Bayesian inference. For example, here the robust and sparse regression based on MCC-ARD can be rethought by treating the kernel bandwidth h as a random variable. Suppose that h is assigned with the non-informative hyper-prior, then the log joint distribution is written by

$$\begin{aligned} &\log p(\mathbf{w}, \mathbf{a}, \mathbf{t}, h) \\ =& \log p(\mathbf{t}|\mathbf{w}, h) + \log p(\mathbf{w}|\mathbf{a}) + \log p(\mathbf{a}) + \log p(h) \\ =& \sum_{n=1}^{N} \exp(-\frac{(t_n - \mathbf{x}_n\mathbf{w})^2}{2h}) - \frac{1}{2}\mathbf{w}^T\mathbf{A}\mathbf{w} - \frac{1}{2}\log|\mathbf{A}| - \log h + const \end{aligned} \tag{143}$$

Accordingly, the variational inference becomes

$$\log q_{\mathbf{w}}(\mathbf{w}) = \mathbb{E}_{q_{\mathbf{a}}(\mathbf{a})q_h(h)}\left[\log p(\mathbf{w}, \mathbf{a}, \mathbf{t}, h)\right] = \sum_{n=1}^{N} \mathbb{E}_{q_h(h)}\left[\exp(-\frac{e_n^2}{2h})\right] - \frac{1}{2}\mathbf{w}^T\mathbb{E}_{q_{\mathbf{a}}(\mathbf{a})}\left[\mathbf{A}\right]\mathbf{w} \tag{144}$$

$$\log q_{\mathbf{a}}(\mathbf{a}) = \mathbb{E}_{q_{\mathbf{w}}(\mathbf{w})q_h(h)}\left[\log p(\mathbf{w}, \mathbf{a}, \mathbf{t}, h)\right] = -\frac{1}{2}\sum_{d=1}^{D} a_d \mathbb{E}_{q_{\mathbf{w}}(\mathbf{w})}\left[w_d^2\right] - \frac{1}{2}\sum_{d=1}^{D}\log a_d \tag{145}$$

$$\log q_h(h) = \mathbb{E}_{q_{\mathbf{w}}(\mathbf{w})q_{\mathbf{a}}(\mathbf{a})}\left[\log p(\mathbf{w}, \mathbf{a}, \mathbf{t}, h)\right] = \sum_{n=1}^{N} \mathbb{E}_{q_{\mathbf{w}}(\mathbf{w})}\left[\exp(-\frac{(t_n - \mathbf{x}_n\mathbf{w})^2}{2h})\right] - \log h \tag{146}$$

where, however, one can find that the expectations with respect to the correntropy term in $\log q_{\mathbf{w}}(\mathbf{w})$ and $\log q_h(h)$ is pretty hard to compute analytically. Thus, some other approximations are essential to treat the kernel bandwidth h as a random variable. In the future works, this will be investigated more in depth, thus realizing "*adaptive robustness*".

*6.3. **A Wider Prospect for Brain Activity Decoding***

Finally, a wider prospect for brain activity decoding is discussed in what follows. One could find that almost all the machine learning models in this thesis are limited to the linear formulation which may be insufficient to realize good enough brain decoding performance. However, how to design non-linear while informative features has been a difficult question for a long time, i.e. feature engineering. A good news is, the development of deep learning models can provide a powerful tool for non-linear feature extraction and pattern recognition. A conventional brain decoding framework usually predicts the linear feature of target from a linear representation of brain recordings. By comparison, one could employ a non-linear representation for either input or output. For example, instead of reconstructing the original visual stimulus, [198] utilized a convolutional neural network (CNN) model for feature extraction on the visual stimulus, and then predicted the CNN-based visual features from fMRI signals, thus realizing a much more complicated visual decoding task than [33]. Moreover, the deep generative models have been recently used for visual reconstruction that have been trained previously on a large number of naturalistic images [36–38]. It will be of large potential to investigate how to propose more advanced brain decoding framework with a motivation from the recent developments of the computer vision or natural language processing, since vision and language are two significant cognitive functions for human brains. The recently developed large language model, such as GPT, will be big inspiration for future studies about human brain.

Acknowledgments

First of all, I would like to express my sincere gratitude to my main supervisor, Prof. Yasuharu Koike, who has provided me with powerful research support and warm solicitude during my five-year master and doctoral programs in TokyoTech. I also desire to express my gratitude to my sub supervisor, Prof. Natsue Yoshimura, who has provided me considerable guidance for my past research. Moreover, I desire to show my consistent gratitude to my previous supervisor during the undergraduate in XJTU, Prof. Badong Chen, who provided enlightening suggestions and constructive comments in my studies. In addition, I want to appreciate Dr. Okito Yamashita who served as my supervisor during my research internship at ATR and provided me with significant guidance for my research.

Next, I would like to appreciate my parents, who have given me crucial spiritual and financial assistance in the past five years. In particular, during the atrocious outbreaks of COVID-19, my parents gave me huge encouragement so that I managed to get through those difficult time. I desire to thank these two dearest people for everything I can remember and imagine.

In addition, I desire to say a huge thanks to my dear friends who gave me spiritual companionship when I was having a hard time. I also want to thank the lab members in TokyoTech for their help.

Finally, I also desire to thank the economic support from "Cross the border! TokyoTech Pioneering Doctoral Research Project" with the living expenses which enabled me to concentrate on my research and the research funds for purchasing necessary research instruments, conducting the valuable research internship, and attending the international academic conference.

This is the end of the previous story, while is also the beginning of a new life. Even though there were many difficulties and sufferings in the past, they are all part of the present "completion". Looking back at the past, my skiff has traveled a thousand miles. Best wishes to all the people who have helped me, and also to myself.

References

1. Shipp, S. Structure and function of the cerebral cortex. *Current Biology* **2007**, *17*, 443–449.
2. Arráez-Aybar, L.A.; Navia-Álvarez, P.; Fuentes-Redondo, T.; Bueno-López, J.L. Thomas Willis, a pioneer in translational research in anatomy (on the 350th anniversary of Cerebri anatome). *Journal of anatomy* **2015**, *226*, 289–300.
3. Sperry, R.W. Hemisphere deconnection and unity in conscious awareness. *American psychologist* **1968**, *23*, 723.
4. Selkoe, D.J. Translating cell biology into therapeutic advances in Alzheimer's disease. *Nature* **1999**, *399*, A23–A31.
5. Maynard, S.A.; Ranft, J.; Triller, A. Quantifying postsynaptic receptor dynamics: insights into synaptic function. *Nature Reviews Neuroscience* **2023**, *24*, 4–22.
6. Thiebaut de Schotten, M.; Forkel, S.J. The emergent properties of the connected brain. *Science* **2022**, *378*, 505–510.
7. Teplan, M.; others. Fundamentals of EEG measurement. *Measurement science review* **2002**, *2*, 1–11.
8. da Silva, F.L. EEG and MEG: relevance to neuroscience. *Neuron* **2013**, *80*, 1112–1128.
9. Buzsáki, G.; Anastassiou, C.A.; Koch, C. The origin of extracellular fields and currents—EEG, ECoG, LFP and spikes. *Nature reviews neuroscience* **2012**, *13*, 407–420.
10. Heeger, D.J.; Ress, D. What does fMRI tell us about neuronal activity? *Nature reviews neuroscience* **2002**, *3*, 142–151.
11. Glaser, J.I.; Benjamin, A.S.; Chowdhury, R.H.; Perich, M.G.; Miller, L.E.; Kording, K.P. Machine learning for neural decoding. *Eneuro* **2020**, *7*.
12. Saeidi, M.; Karwowski, W.; Farahani, F.V.; Fiok, K.; Taiar, R.; Hancock, P.; Al-Juaid, A. Neural decoding of EEG signals with machine learning: a systematic review. *Brain Sciences* **2021**, *11*, 1525.
13. Holdgraf, C.R.; Rieger, J.W.; Micheli, C.; Martin, S.; Knight, R.T.; Theunissen, F.E. Encoding and decoding models in cognitive electrophysiology. *Frontiers in systems neuroscience* **2017**, *11*, 61.
14. Xu, L.; Xu, M.; Jung, T.P.; Ming, D. Review of brain encoding and decoding mechanisms for EEG-based brain–computer interface. *Cognitive Neurodynamics* **2021**, *15*, 569–584.
15. Norman, K.A.; Polyn, S.M.; Detre, G.J.; Haxby, J.V. Beyond mind-reading: multi-voxel pattern analysis of fMRI data. *Trends in cognitive sciences* **2006**, *10*, 424–430.
16. Lotte, F.; Congedo, M.; Lécuyer, A.; Lamarche, F.; Arnaldi, B. A review of classification algorithms for EEG-based brain–computer interfaces. *Journal of neural engineering* **2007**, *4*, R1.
17. Lotte, F.; Bougrain, L.; Cichocki, A.; Clerc, M.; Congedo, M.; Rakotomamonjy, A.; Yger, F. A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update. *Journal of neural engineering* **2018**, *15*, 031005.
18. McAvinue, L.P.; Robertson, I.H. Measuring motor imagery ability: a review. *European journal of cognitive psychology* **2008**, *20*, 232–251.
19. Pei, X.; Barbour, D.L.; Leuthardt, E.C.; Schalk, G. Decoding vowels and consonants in spoken and imagined words using electrocorticographic signals in humans. *Journal of neural engineering* **2011**, *8*, 046028.
20. Brumberg, J.S.; Wright, E.J.; Andreasen, D.S.; Guenther, F.H.; Kennedy, P.R. Classification of intended phoneme production from chronic intracortical microelectrode recordings in speech motor cortex. *Frontiers in neuroscience* **2011**, p. 65.
21. Martin, S.; Brunner, P.; Iturrate, I.; Millán, J.d.R.; Schalk, G.; Knight, R.T.; Pasley, B.N. Word pair classification during imagined speech using direct brain recordings. *Scientific reports* **2016**, *6*, 25803.
22. Zhang, D.; Gong, E.; Wu, W.; Lin, J.; Zhou, W.; Hong, B. Spoken sentences decoding based on intracranial high gamma response using dynamic time warping. 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE, 2012, pp. 3292–3295.
23. Kamitani, Y.; Tong, F. Decoding the visual and subjective contents of the human brain. *Nature neuroscience* **2005**, *8*, 679–685.
24. Van Gerven, M.A.; Cseke, B.; De Lange, F.P.; Heskes, T. Efficient Bayesian multivariate fMRI analysis using a sparsifying spatio-temporal prior. *NeuroImage* **2010**, *50*, 150–161.
25. Damarla, S.R.; Just, M.A. Decoding the representation of numerical values from brain activation patterns. *Human brain mapping* **2013**, *34*, 2624–2634.

26. Hochberg, L.R.; Bacher, D.; Jarosiewicz, B.; Masse, N.Y.; Simeral, J.D.; Vogel, J.; Haddadin, S.; Liu, J.; Cash, S.S.; Van Der Smagt, P.; others. Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. *Nature* **2012**, *485*, 372–375.
27. Chao, Z.C.; Nagasaka, Y.; Fujii, N. Long-term asynchronous decoding of arm motion using electrocorticographic signals in monkey. *Frontiers in neuroengineering* **2010**, *3*, 3.
28. Shimoda, K.; Nagasaka, Y.; Chao, Z.C.; Fujii, N. Decoding continuous three-dimensional hand trajectories from epidural electrocorticographic signals in Japanese macaques. *Journal of neural engineering* **2012**, *9*, 036015.
29. Pasley, B.N.; David, S.V.; Mesgarani, N.; Flinker, A.; Shamma, S.A.; Crone, N.E.; Knight, R.T.; Chang, E.F. Reconstructing speech from human auditory cortex. *PLoS biology* **2012**, *10*, e1001251.
30. Martin, S.; Brunner, P.; Holdgraf, C.; Heinze, H.J.; Crone, N.E.; Rieger, J.; Schalk, G.; Knight, R.T.; Pasley, B.N. Decoding spectrotemporal features of overt and covert speech from the human cortex. *Frontiers in neuroengineering* **2014**, *7*, 14.
31. Chakrabarti, S.; Krusienski, D.J.; Schalk, G.; Brumberg, J.S. Predicting mel-frequency cepstral coefficients from electrocorticographic signals during continuous speech production. Abstract presented at Proceedings of the Sixth International IEEE/EMBS Neural Engineering Conference, San Diego, CA, 2013.
32. Kubanek, J.; Brunner, P.; Gunduz, A.; Poeppel, D.; Schalk, G. The tracking of speech envelope in the human cortex. *PloS one* **2013**, *8*, e53398.
33. Miyawaki, Y.; Uchida, H.; Yamashita, O.; Sato, M.a.; Morito, Y.; Tanabe, H.C.; Sadato, N.; Kamitani, Y. Visual image reconstruction from human brain activity using a combination of multiscale local image decoders. *Neuron* **2008**, *60*, 915–929.
34. Naselaris, T.; Prenger, R.J.; Kay, K.N.; Oliver, M.; Gallant, J.L. Bayesian reconstruction of natural images from human brain activity. *Neuron* **2009**, *63*, 902–915.
35. Nishimoto, S.; Vu, A.T.; Naselaris, T.; Benjamini, Y.; Yu, B.; Gallant, J.L. Reconstructing visual experiences from brain activity evoked by natural movies. *Current biology* **2011**, *21*, 1641–1646.
36. Beliy, R.; Gaziv, G.; Hoogi, A.; Strappini, F.; Golan, T.; Irani, M. From voxels to pixels and back: Self-supervision in natural-image reconstruction from fMRI. *Advances in Neural Information Processing Systems* **2019**, *32*.
37. Fang, T.; Qi, Y.; Pan, G. Reconstructing perceptive images from brain activity by shape-semantic gan. *Advances in Neural Information Processing Systems* **2020**, *33*, 13038–13048.
38. Gaziv, G.; Beliy, R.; Granot, N.; Hoogi, A.; Strappini, F.; Golan, T.; Irani, M. Self-supervised natural image reconstruction and large-scale semantic classification from brain activity. *NeuroImage* **2022**, *254*, 119121.
39. Wolpaw, J.R.; Birbaumer, N.; McFarland, D.J.; Pfurtscheller, G.; Vaughan, T.M. Brain–computer interfaces for communication and control. *Clinical neurophysiology* **2002**, *113*, 767–791.
40. Fetz, E.E. Operant conditioning of cortical unit activity. *Science* **1969**, *163*, 955–958.
41. Chapin, J.K.; Moxon, K.A.; Markowitz, R.S.; Nicolelis, M.A. Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex. *Nature neuroscience* **1999**, *2*, 664–670.
42. Birbaumer, N.; Ghanayim, N.; Hinterberger, T.; Iversen, I.; Kotchoubey, B.; Kübler, A.; Perelmouter, J.; Taub, E.; Flor, H. A spelling device for the paralysed. *Nature* **1999**, *398*, 297–298.
43. Donoghue, J.P.; Nurmikko, A.; Black, M.; Hochberg, L.R. Assistive technology and robotic control using motor cortex ensemble-based neural interface systems in humans with tetraplegia. *The Journal of physiology* **2007**, *579*, 603–611.
44. McFarland, D.J.; Neat, G.W.; Read, R.F.; Wolpaw, J.R. An EEG-based method for graded cursor control. *Psychobiology* **1993**, *21*, 77–81.
45. Palumbo, A.; Gramigna, V.; Calabrese, B.; Ielpo, N. Motor-imagery EEG-based BCIs in wheelchair movement and control: A systematic literature review. *Sensors* **2021**, *21*, 6285.
46. Willett, F.R.; Avansino, D.T.; Hochberg, L.R.; Henderson, J.M.; Shenoy, K.V. High-performance brain-to-text communication via handwriting. *Nature* **2021**, *593*, 249–254.
47. Donoghue, J.P. Connecting cortex to machines: recent advances in brain interfaces. *Nature neuroscience* **2002**, *5*, 1085–1088.
48. Mussa-Ivaldi, F.A.; Miller, L.E. Brain–machine interfaces: computational demands and clinical needs meet basic neuroscience. *TRENDS in Neurosciences* **2003**, *26*, 329–334.

49. Lebedev, M.A.; Nicolelis, M.A. Brain–machine interfaces: past, present and future. *TRENDS in Neurosciences* **2006**, *29*, 536–546.
50. Brandman, D.M.; Cash, S.S.; Hochberg, L.R. human intracortical recording and neural decoding for brain–computer interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **2017**, *25*, 1687–1696.
51. Rybář, M.; Daly, I. Neural decoding of semantic concepts: A systematic literature review. *Journal of Neural Engineering* **2022**.
52. Ball, T.; Kern, M.; Mutschler, I.; Aertsen, A.; Schulze-Bonhage, A. Signal quality of simultaneously recorded invasive and non-invasive EEG. *Neuroimage* **2009**, *46*, 708–716.
53. Sweeney, K.T.; Ward, T.E.; McLoone, S.F. Artifact removal in physiological signals—Practices and possibilities. *IEEE transactions on information technology in biomedicine* **2012**, *16*, 488–500.
54. Lund, T.E.; Madsen, K.H.; Sidaros, K.; Luo, W.L.; Nichols, T.E. Non-white noise in fMRI: does modelling have an impact? *Neuroimage* **2006**, *29*, 54–66.
55. Liu, T.T. Noise contributions to the fMRI signal: An overview. *NeuroImage* **2016**, *143*, 141–151.
56. Jung, T.P.; Makeig, S.; Humphries, C.; Lee, T.W.; Mckeown, M.J.; Iragui, V.; Sejnowski, T.J. Removing electroencephalographic artifacts by blind source separation. *Psychophysiology* **2000**, *37*, 163–178.
57. Urigüen, J.A.; Garcia-Zapirain, B. EEG artifact removal—state-of-the-art and guidelines. *Journal of neural engineering* **2015**, *12*, 031001.
58. Constable, C. Parameter estimation in non-Gaussian noise. *Geophysical Journal International* **1988**, *94*, 131–142.
59. Huber, P.J. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics* **1964**, pp. 73–101.
60. Qayyum, A.; Qadir, J.; Bilal, M.; Al-Fuqaha, A. Secure and robust machine learning for healthcare: A survey. *IEEE Reviews in Biomedical Engineering* **2020**, *14*, 156–180.
61. Shafique, M.; Naseer, M.; Theocharides, T.; Kyrkou, C.; Mutlu, O.; Orosa, L.; Choi, J. Robust machine learning systems: Challenges, current trends, perspectives, and the road ahead. *IEEE Design & Test* **2020**, *37*, 30–57.
62. Krauledat, M.; Dornhege, G.; Blankertz, B.; Müller, K.R.; others. Robustifying EEG data analysis by removing outliers. *Chaos and Complexity Letters* **2007**, *2*, 259–274.
63. Ren, M.; Zeng, W.; Yang, B.; Urtasun, R. Learning to reweight examples for robust deep learning. International conference on machine learning. PMLR, 2018, pp. 4334–4343.
64. Diakonikolas, I.; Kamath, G.; Kane, D.; Li, J.; Steinhardt, J.; Stewart, A. Sever: A robust meta-algorithm for stochastic optimization. International Conference on Machine Learning. PMLR, 2019, pp. 1596–1606.
65. Wang, Z.; Hu, G.; Hu, Q. Training noise-robust deep neural networks via meta-learning. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 4524–4533.
66. Black, M.J.; Rangarajan, A. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International journal of computer vision* **1996**, *19*, 57–91.
67. Wang, D.; Romagnoli, J. A framework for robust data reconciliation based on a generalized objective function. *Industrial & engineering chemistry research* **2003**, *42*, 3075–3084.
68. Barron, J.T. A general and adaptive robust loss function. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4331–4339.
69. De Menezes, D.; Prata, D.M.; Secchi, A.R.; Pinto, J.C. A review on robust M-estimators for regression analysis. *Computers & Chemical Engineering* **2021**, *147*, 107254.
70. Principe, J.C. *Information theoretic learning: Renyi's entropy and kernel perspectives*; Springer Science & Business Media, 2010.
71. Vergara, J.R.; Estévez, P.A. A review of feature selection methods based on mutual information. *Neural computing and applications* **2014**, *24*, 175–186.
72. Hjelm, R.D.; Fedorov, A.; Lavoie-Marchildon, S.; Grewal, K.; Bachman, P.; Trischler, A.; Bengio, Y. Learning deep representations by mutual information estimation and maximization. International Conference on Learning Representations, 2019.
73. Tishby, N.; Zaslavsky, N. Deep learning and the information bottleneck principle. 2015 ieee information theory workshop (itw). IEEE, 2015, pp. 1–5.

74. Saxe, A.M.; Bansal, Y.; Dapello, J.; Advani, M.; Kolchinsky, A.; Tracey, B.D.; Cox, D.D. On the information bottleneck theory of deep learning. *Journal of Statistical Mechanics: Theory and Experiment* **2019**, *2019*, 124020.
75. Singh, A.; Pokharel, R.; Principe, J. The C-loss function for pattern classification. *Pattern Recognition* **2014**, *47*, 441–453.
76. Xu, G.; Hu, B.G.; Principe, J.C. Robust C-loss kernel classifiers. *IEEE transactions on neural networks and learning systems* **2016**, *29*, 510–522.
77. Ren, Z.; Yang, L. Correntropy-based robust extreme learning machine for classification. *Neurocomputing* **2018**, *313*, 74–84.
78. de Sá, J.P.M.; Silva, L.M.; Santos, J.M.; Alexandre, L.A. *Minimum error entropy classification*; Springer, 2013.
79. Liu, W.; Pokharel, P.P.; Principe, J.C. Correntropy: Properties and applications in non-Gaussian signal processing. *IEEE Transactions on signal processing* **2007**, *55*, 5286–5298.
80. Feng, Y.; Huang, X.; Shi, L.; Yang, Y.; Suykens, J.A.; others. Learning with the maximum correntropy criterion induced losses for regression. *Journal of Machine Learning Research* **2015**, *16*, 993–1034.
81. Chen, B.; Wang, J.; Zhao, H.; Zheng, N.; Principe, J.C. Convergence of a fixed-point algorithm under maximum correntropy criterion. *IEEE Signal Processing Letters* **2015**, *22*, 1723–1727.
82. Chen, B.; Xing, L.; Zhao, H.; Zheng, N.; Prı, J.C.; others. Generalized correntropy for robust adaptive filtering. *IEEE Transactions on Signal Processing* **2016**, *64*, 3376–3387.
83. Ma, W.; Zheng, D.; Li, Y.; Zhang, Z.; Chen, B. Bias-compensated normalized maximum correntropy criterion algorithm for system identification with noisy input. *Signal Processing* **2018**, *152*, 160–164.
84. Chen, B.; Xing, L.; Nanning, Z.; Príncipe, J.C. Quantized Minimum Error Entropy Criterion. *IEEE Transactions on Neural Networks and Learning Systems* **2019**, *30*, 1370–1380.
85. Guo, Z.; Yue, H.; Wang, H. A modified PCA based on the minimum error entropy. Proceedings of the 2004 American Control Conference. IEEE, 2004, Vol. 4, pp. 3800–3801.
86. Wang, Y.; Tang, Y.Y.; Li, L. Minimum error entropy based sparse representation for robust subspace clustering. *IEEE Transactions on Signal Processing* **2015**, *63*, 4010–4021.
87. Dong, J.; Chen, B.; Lu, N.; Wang, H.; Zheng, N. Correntropy induced metric based common spatial patterns. 2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP). IEEE, 2017, pp. 1–6.
88. Zhou, N.; Xu, Y.; Cheng, H.; Yuan, Z.; Chen, B. Maximum correntropy criterion-based sparse subspace learning for unsupervised feature selection. *IEEE Transactions on Circuits and Systems for Video Technology* **2017**, *29*, 404–417.
89. Chen, B.; Li, Y.; Dong, J.; Lu, N.; Qin, J. Common spatial patterns based on the quantized minimum error entropy criterion. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **2018**, *50*, 4557–4568.
90. Nicolas-Alonso, L.F.; Gomez-Gil, J. Brain computer interfaces, a review. *sensors* **2012**, *12*, 1211–1279.
91. Davatzikos, C.; Resnick, S.M.; Wu, X.; Parmpi, P.; Clark, C.M. Individual patient diagnosis of AD and FTD via high-dimensional pattern classification of MRI. *Neuroimage* **2008**, *41*, 1220–1227.
92. Casanova, R.; Whitlow, C.T.; Wagner, B.; Williamson, J.; Shumaker, S.A.; Maldjian, J.A.; Espeland, M.A. High dimensional classification of structural MRI Alzheimer's disease data based on large scale regularization. *Frontiers in neuroinformatics* **2011**, *5*, 22.
93. Hemmelmann, C.; Horn, M.; Reiterer, S.; Schack, B.; Süsse, T.; Weiss, S. Multivariate tests for the evaluation of high-dimensional EEG data. *Journal of Neuroscience Methods* **2004**, *139*, 111–120.
94. Yu, X.; Chum, P.; Sim, K.B. Analysis the effect of PCA for feature reduction in non-stationary EEG based motor imagery of BCI system. *Optik* **2014**, *125*, 1498–1502.
95. Zhong, Y.; Wang, H.; Lu, G.; Zhang, Z.; Jiao, Q.; Liu, Y. Detecting functional connectivity in fMRI using PCA and regression analysis. *Brain topography* **2009**, *22*, 134–144.
96. Chen, C.; Cao, X.; Tian, L. Partial least squares regression performs well in MRI-based individualized estimations. *Frontiers in neuroscience* **2019**, *13*, 1282.
97. McIntosh, A.R.; Lobaugh, N.J. Partial least squares analysis of neuroimaging data: applications and advances. *Neuroimage* **2004**, *23*, S250–S263.
98. Krishnan, A.; Williams, L.J.; McIntosh, A.R.; Abdi, H. Partial Least Squares (PLS) methods for neuroimaging: a tutorial and review. *Neuroimage* **2011**, *56*, 455–475.
99. Yamashita, O.; Sato, M.a.; Yoshioka, T.; Tong, F.; Kamitani, Y. Sparse estimation automatically selects voxels relevant for the decoding of fMRI activity patterns. *NeuroImage* **2008**, *42*, 1414–1429.

100. van Gerven, M.; Hesse, C.; Jensen, O.; Heskes, T. Interpreting single trial data using groupwise regularisation. *NeuroImage* **2009**, *46*, 665–676.
101. Ryali, S.; Supekar, K.; Abrams, D.A.; Menon, V. Sparse logistic regression for whole-brain classification of fMRI data. *NeuroImage* **2010**, *51*, 752–764.
102. Satake, E.; Majima, K.; Aoki, S.C.; Kamitani, Y. Sparse ordinal logistic regression and its application to brain decoding. *Frontiers in neuroinformatics* **2018**, *12*, 51.
103. Chestek, C.A.; Batista, A.P.; Santhanam, G.; Byron, M.Y.; Afshar, A.; Cunningham, J.P.; Gilja, V.; Ryu, S.I.; Churchland, M.M.; Shenoy, K.V. Single-neuron stability during repeated reaching in macaque premotor cortex. *Journal of Neuroscience* **2007**, *27*, 10742–10750.
104. Zhuang, M.; Wu, Q.; Wan, F.; Hu, Y. State-of-the-art non-invasive brain–computer interface for neural rehabilitation: A review. *Journal of Neurorestoratology* **2020**, *8*, 12–25.
105. Amiri, S.; Fazel-Rezai, R.; Asadpour, V. A review of hybrid brain-computer interface systems. *Advances in Human-Computer Interaction* **2013**, *2013*.
106. Tak, S.; Ye, J.C. Statistical analysis of fNIRS data: a comprehensive review. *Neuroimage* **2014**, *85*, 72–91.
107. Myles, A.J.; Feudale, R.N.; Liu, Y.; Woody, N.A.; Brown, S.D. An introduction to decision tree modeling. *Journal of Chemometrics: A Journal of the Chemometrics Society* **2004**, *18*, 275–285.
108. Maalouf, M. Logistic regression in data analysis: an overview. *International Journal of Data Analysis Techniques and Strategies* **2011**, *3*, 281–299.
109. Zhang, T. Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics* **2004**, pp. 56–85.
110. Bartlett, P.L.; Jordan, M.I.; McAuliffe, J.D. Convexity, classification, and risk bounds. *Journal of the American Statistical Association* **2006**, *101*, 138–156.
111. Koles, Z.J.; Lazar, M.S.; Zhou, S.Z. Spatial patterns underlying population differences in the background EEG. *Brain topography* **1990**, *2*, 275–284.
112. Ang, K.K.; Chin, Z.Y.; Zhang, H.; Guan, C. Filter bank common spatial pattern (FBCSP) in brain-computer interface. 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence). IEEE, 2008, pp. 2390–2397.
113. Lu, H.; Eng, H.L.; Guan, C.; Plataniotis, K.N.; Venetsanopoulos, A.N. Regularized common spatial pattern with aggregation for EEG classification in small-sample setting. *IEEE transactions on Biomedical Engineering* **2010**, *57*, 2936–2946.
114. Ang, K.K.; Chin, Z.Y.; Wang, C.; Guan, C.; Zhang, H. Filter bank common spatial pattern algorithm on BCI competition IV datasets 2a and 2b. *Frontiers in neuroscience* **2012**, *6*, 39.
115. Aggarwal, C.C.; Aggarwal, C.C. *An introduction to outlier analysis*; Springer, 2017.
116. Wang, H.; Bah, M.J.; Hammad, M. Progress in outlier detection techniques: A survey. *Ieee Access* **2019**, *7*, 107964–108000.
117. Shannon, C.E. A mathematical theory of communication. *The Bell system technical journal* **1948**, *27*, 379–423.
118. Rényi, A. On measures of entropy and information. Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics. University of California Press, 1961, Vol. 4, pp. 547–562.
119. He, R.; Hu, B.; Yuan, X.; Zheng, W.S. Principal component analysis based on non-parametric maximum entropy. *Neurocomputing* **2010**, *73*, 1840–1852.
120. Li, Y.; Zhou, J.; Zheng, X.; Tian, J.; Tang, Y.Y. Robust Subspace Clustering with Independent and Piecewise Identically Distributed Noise Modeling. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 8720–8729.
121. Silverman, B.W. Density estimation for statistics and data analysis. *Technometrics* **1986**, *29*, 495–495.
122. Parzen, E. On estimation of a probability density function and mode. *The annals of mathematical statistics* **1962**, *33*, 1065–1076.
123. Hu, T.; Fan, J.; Wu, Q.; Zhou, D.X. Learning theory approach to minimum error entropy criterion. *Journal of Machine Learning Research* **2013**, *14*, 377–397.
124. Chen, B.; Xing, L.; Xu, B.; Zhao, H.; Principe, J.C. Insights into the robustness of minimum error entropy estimation. *IEEE transactions on neural networks and learning systems* **2016**, *29*, 731–737.
125. Santamaría, I.; Pokharel, P.P.; Principe, J.C. Generalized correlation function: definition, properties, and application to blind equalization. *IEEE Transactions on Signal Processing* **2006**, *54*, 2187–2197.

126. Huber, P.J. *Robust statistics*; Vol. 523, John Wiley & Sons, 2004.
127. Chen, B.; Wang, X.; Li, Y.; Principe, J.C. Maximum correntropy criterion with variable center. *IEEE Signal Processing Letters* **2019**, *26*, 1212–1216.
128. Chen, B.; Wang, X.; Lu, N.; Wang, S.; Cao, J.; Qin, J. Mixture correntropy for robust learning. *Pattern Recognition* **2018**, *79*, 318–327.
129. Chen, B.; Xie, Y.; Wang, X.; Yuan, Z.; Ren, P.; Qin, J. Multikernel correntropy for robust learning. *IEEE Transactions on Cybernetics* **2021**, *52*, 13500–13511.
130. Hampel, F.R.; Ronchetti, E.M.; Rousseeuw, P.J.; Stahel, W.A. *Robust statistics: the approach based on influence functions*; John Wiley & Sons, 1986.
131. Zhu, X.; Wu, X. Class noise vs. attribute noise: A quantitative study. *Artificial intelligence review* **2004**, *22*, 177–210.
132. Frénay, B.; Verleysen, M. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems* **2013**, *25*, 845–869.
133. Wu, Y.; Liu, Y. Robust truncated hinge loss support vector machines. *Journal of the American Statistical Association* **2007**, *102*, 974–983.
134. Masnadi-Shirazi, H.; Vasconcelos, N. On the design of loss functions for classification: theory, robustness to outliers, and savageboost. Advances in neural information processing systems, 2009, pp. 1049–1056.
135. Miao, Q.; Cao, Y.; Xia, G.; Gong, M.; Liu, J.; Song, J. Rboost: label noise-robust boosting algorithm based on a nonconvex loss function and the numerically stable base learners. *IEEE transactions on neural networks and learning systems* **2015**, *27*, 2216–2228.
136. Li, Y.; Chen, B.; Yoshimura, N.; Koike, Y. Restricted minimum error entropy criterion for robust classification. *IEEE Transactions on Neural Networks and Learning Systems* **2022**, *33*, 6599–6612.
137. Quinlan, J.R. Induction of decision trees. *Machine learning* **1986**, *1*, 81–106.
138. Hickey, R.J. Noise modelling and evaluating learning from examples. *Artificial Intelligence* **1996**, *82*, 157–179.
139. Bross, I. Misclassification in 2 x 2 tables. *Biometrics* **1954**, *10*, 478–486.
140. Collett, D.; Lewis, T. The subjective nature of outlier rejection procedures. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **1976**, *25*, 228–237.
141. Hodge, V.; Austin, J. A survey of outlier detection methodologies. *Artificial intelligence review* **2004**, *22*, 85–126.
142. Chandola, V.; Banerjee, A.; Kumar, V. Anomaly detection: A survey. *ACM computing surveys (CSUR)* **2009**, *41*, 1–58.
143. Feng, J.; Xu, H.; Mannor, S.; Yan, S. Robust logistic regression and classification. Advances in neural information processing systems, 2014, pp. 253–261.
144. Suykens, J.A.; De Brabanter, J.; Lukas, L.; Vandewalle, J. Weighted least squares support vector machines: robustness and sparse approximation. *Neurocomputing* **2002**, *48*, 85–105.
145. Byrnes, P.G.; DiazDelaO, F.A. Kernel Logistic Regression: A Robust Weighting for Imbalanced Classes with Noisy Labels. 2018 International Conference on Machine Learning and Data Engineering (iCMLDE). IEEE, 2018, pp. 30–34.
146. Yin, M.; Zeng, D.; Gao, J.; Wu, Z.; Xie, S. Robust multinomial logistic regression based on rpca. *IEEE Journal of Selected Topics in Signal Processing* **2018**, *12*, 1144–1154.
147. Diakonikolas, I.; Kamath, G.; Kane, D.; Li, J.; Steinhardt, J.; Stewart, A. Sever: A robust meta-algorithm for stochastic optimization. International Conference on Machine Learning, 2019, pp. 1596–1606.
148. Ertekin, S.; Bottou, L.; Giles, C.L. Nonconvex online support vector machines. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2010**, *33*, 368–381.
149. Yang, X.; Tan, L.; He, L. A robust least squares support vector machine for regression and classification with noise. *Neurocomputing* **2014**, *140*, 41–52.
150. Collett, D. *Modelling binary data*; CRC press, 2002.
151. Jennings, D.E. Outliers and residual distributions in logistic regression. *Journal of the American Statistical Association* **1986**, *81*, 987–990.
152. Landwehr, J.M.; Pregibon, D.; Shoemaker, A.C. Graphical methods for assessing logistic regression models. *Journal of the American Statistical Association* **1984**, *79*, 61–71.
153. Cha, S.H. Comprehensive survey on distance/similarity measures between probability density functions. *City* **2007**, *1*, 1.

154. Deza, M.M.; Deza, E. *Dictionary of distances*; Elsevier, 2006.
155. Duda, R.O.; Hart, P.E.; Stork, D.G. *Pattern classification*; John Wiley & Sons, 2012.
156. Yuan, X.T.; Hu, B.G. Robust feature extraction via information theoretic learning. Proceedings of the 26th annual international conference on machine learning, 2009, pp. 1193–1200.
157. He, R.; Hu, B.G.; Zheng, W.S.; Kong, X.W. Robust principal component analysis based on maximum correntropy criterion. *IEEE Transactions on Image Processing* **2011**, *20*, 1485–1494.
158. Boyd, S.; Boyd, S.P.; Vandenberghe, L. *Convex optimization*; Cambridge university press, 2004.
159. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* **2014**.
160. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: theory and applications. *Neurocomputing* **2006**, *70*, 489–501.
161. Bashashati, A.; Fatourechi, M.; Ward, R.K.; Birch, G.E. A survey of signal processing algorithms in brain–computer interfaces based on electrical brain signals. *Journal of Neural engineering* **2007**, *4*, R32.
162. He, L.; Hu, D.; Wan, M.; Wen, Y.; Von Deneen, K.M.; Zhou, M. Common Bayesian network for classification of EEG-based multiclass motor imagery BCI. *IEEE Transactions on Systems, man, and cybernetics: systems* **2015**, *46*, 843–854.
163. Wang, H.; Tang, Q.; Zheng, W. L1-norm-based common spatial patterns. *IEEE Transactions on Biomedical Engineering* **2011**, *59*, 653–662.
164. Asuncion, A.; Newman, D. UCI machine learning repository, 2007.
165. Chereau, J.P.; Scalzo, B.; Mandic, D.P. Robust PCA Through Maximum Correntropy Power Iterations. ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021, pp. 4985–4989.
166. Massy, W.F. Principal components regression in exploratory statistical research. *Journal of the American Statistical Association* **1965**, *60*, 234–256.
167. Wold, H. Estimation of principal components and related models by iterative least squares. *Multivariate analysis* **1966**, pp. 391–420.
168. Zhao, Q.; Zhang, L.; Cichocki, A. Multilinear and nonlinear generalizations of partial least squares: an overview of recent advances. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **2014**, *4*, 104–115.
169. Eliseyev, A.; Moro, C.; Costecalde, T.; Torres, N.; Gharbi, S.; Mestais, C.; Benabid, A.L.; Aksenova, T. Iterative N-way partial least squares for a binary self-paced brain–computer interface in freely moving animals. *Journal of neural engineering* **2011**, *8*, 046012.
170. Eliseyev, A.; Moro, C.; Faber, J.; Wyss, A.; Torres, N.; Mestais, C.; Benabid, A.L.; Aksenova, T. L1-penalized N-way PLS for subset of electrodes selection in BCI experiments. *Journal of neural engineering* **2012**, *9*, 045010.
171. Zhao, Q.; Caiafa, C.F.; Mandic, D.P.; Chao, Z.C.; Nagasaka, Y.; Fujii, N.; Zhang, L.; Cichocki, A. Higher order partial least squares (HOPLS): A generalized multilinear regression method. *IEEE transactions on pattern analysis and machine intelligence* **2012**, *35*, 1660–1673.
172. Zhao, Q.; Zhou, G.; Adalı, T.; Zhang, L.; Cichocki, A. Kernel-based tensor partial least squares for reconstruction of limb movements. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2013, pp. 3577–3581.
173. Eliseyev, A.; Aksenova, T. Penalized multi-way partial least squares for smooth trajectory decoding from electrocorticographic (ECoG) recording. *PloS one* **2016**, *11*, e0154878.
174. Eliseyev, A.; Auboiroux, V.; Costecalde, T.; Langar, L.; Charvet, G.; Mestais, C.; Aksenova, T.; Benabid, A.L. Recursive exponentially weighted n-way partial least squares regression with recursive-validation of hyper-parameters in brain-computer interface applications. *Scientific reports* **2017**, *7*, 1–15.
175. Foodeh, R.; Ebadollahi, S.; Daliri, M.R. Regularized partial least square regression for continuous decoding in brain-computer interfaces. *Neuroinformatics* **2020**, *18*, 465–477.
176. Bro, R. Multiway calibration. multilinear pls. *Journal of chemometrics* **1996**, *10*, 47–61.
177. Otsubo, H.; Ochi, A.; Imai, K.; Akiyama, T.; Fujimoto, A.; Go, C.; Dirks, P.; Donner, E.J. High-frequency oscillations of ictal muscle activity and epileptogenic discharges on intracranial EEG in a temporal lobe epilepsy patient. *Clinical Neurophysiology* **2008**, *119*, 862–868.
178. Mou, Y.; Zhou, L.; Chen, W.; Fan, J.; Zhao, X. Maximum correntropy criterion partial least squares. *Optik* **2018**, *165*, 137–147.

179. Barker, M.; Rayens, W. Partial least squares for discrimination. *Journal of Chemometrics: A Journal of the Chemometrics Society* **2003**, *17*, 166–173.
180. Fletcher, R. *Practical methods of optimization*; John Wiley & Sons, 2013.
181. Loh, P.L.; Wainwright, M.J. Regularized M-estimators with nonconvexity: Statistical and algorithmic theory for local optima. *The Journal of Machine Learning Research* **2015**, *16*, 559–616.
182. Ganesh, G.; Burdet, E.; Haruno, M.; Kawato, M. Sparse linear regression for reconstructing muscle activity from human cortical fMRI. *Neuroimage* **2008**, *42*, 1463–1472.
183. Wipf, D.; Nagarajan, S. A new view of automatic relevance determination. *Advances in neural information processing systems* **2007**, *20*.
184. Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)* **1996**, *58*, 267–288.
185. Ng, A.Y. Feature selection, L 1 vs. L 2 regularization, and rotational invariance. Proceedings of the twenty-first international conference on Machine learning, 2004, p. 78.
186. Figueiredo, M.A. Adaptive sparseness for supervised learning. *IEEE transactions on pattern analysis and machine intelligence* **2003**, *25*, 1150–1159.
187. Krishnapuram, B.; Harternink, A.; Carin, L.; Figueiredo, M.A. A Bayesian approach to joint feature selection and classifier design. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2004**, *26*, 1105–1111.
188. Krishnapuram, B.; Carin, L.; Figueiredo, M.A.; Hartemink, A.J. Sparse multinomial logistic regression: Fast algorithms and generalization bounds. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2005**, *27*, 957–968.
189. Schmidt, M.; Fung, G.; Rosales, R. Fast optimization methods for l1 regularization: A comparative study and two new approaches. European Conference on Machine Learning. Springer, 2007, pp. 286–297.
190. Zhang, Z.; Wang, S.; Liu, D.; Jordan, M.I.; Lawrence, N. EP-GIG Priors and Applications in Bayesian Sparse Learning. *Journal of Machine Learning Research* **2012**, *13*.
191. MacKay, D.J. Bayesian interpolation. *Neural computation* **1992**, *4*, 415–447.
192. Lisi, G.; Noda, T.; Morimoto, J. Decoding the ERD/ERS: influence of afferent input induced by a leg assistive robot. *Frontiers in systems neuroscience* **2014**, *8*, 85.
193. Lisi, G.; Morimoto, J. EEG single-trial detection of gait speed changes during treadmill walk. *PloS one* **2015**, *10*, e0125479.
194. Ganesh, G.; Nakamura, K.; Saetia, S.; Tobar, A.M.; Yoshida, E.; Ando, H.; Yoshimura, N.; Koike, Y. Utilizing sensory prediction errors for movement intention decoding: a new methodology. *Science Advances* **2018**, *4*, eaaq0183.
195. Shi, Y.; Ganesh, G.; Ando, H.; Koike, Y.; Yoshida, E.; Yoshimura, N. Galvanic Vestibular Stimulation-Based Prediction Error Decoding and Channel Optimization. *International Journal of Neural Systems* **2021**, *31*, 2150034.
196. Shibata, K.; Watanabe, T.; Sasaki, Y.; Kawato, M. Perceptual learning incepted by decoded fMRI neurofeedback without stimulus presentation. *science* **2011**, *334*, 1413–1415.
197. Yahata, N.; Morimoto, J.; Hashimoto, R.; Lisi, G.; Shibata, K.; Kawakubo, Y.; Kuwabara, H.; Kuroda, M.; Yamada, T.; Megumi, F.; others. A small number of abnormal brain connections predicts adult autism spectrum disorder. *Nature communications* **2016**, *7*, 1–12.
198. Horikawa, T.; Kamitani, Y. Generic decoding of seen and imagined objects using hierarchical visual features. *Nature communications* **2017**, *8*, 1–15.
199. Morioka, H.; Kanemura, A.; Morimoto, S.; Yoshioka, T.; Oba, S.; Kawanabe, M.; Ishii, S. Decoding spatial attention by using cortical currents estimated from electroencephalography with near-infrared spectroscopy prior information. *Neuroimage* **2014**, *90*, 128–139.
200. Yoshimura, N.; Nishimoto, A.; Belkacem, A.N.; Shin, D.; Kambara, H.; Hanakawa, T.; Koike, Y. Decoding of covert vowel articulation using electroencephalography cortical currents. *Frontiers in neuroscience* **2016**, *10*, 175.
201. Mejia Tobar, A.; Hyoudou, R.; Kita, K.; Nakamura, T.; Kambara, H.; Ogata, Y.; Hanakawa, T.; Koike, Y.; Yoshimura, N. Decoding of ankle flexion and extension from cortical current sources estimated from non-invasive brain activity recording methods. *Frontiers in neuroscience* **2018**, *11*, 733.
202. He, R.; Zheng, W.S.; Hu, B.G. Maximum correntropy criterion for robust face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2010**, *33*, 1561–1576.

203. He, R.; Zheng, W.S.; Hu, B.G.; Kong, X.W. A regularized correntropy framework for robust pattern recognition. *Neural computation* **2011**, *23*, 2074–2100.
204. He, R.; Zheng, W.S.; Tan, T.; Sun, Z. Half-quadratic-based iterative minimization for robust sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2013**, *36*, 261–275.
205. Ma, W.; Qu, H.; Gui, G.; Xu, L.; Zhao, J.; Chen, B. Maximum correntropy criterion based sparse adaptive filtering algorithms for robust channel estimation under non-Gaussian environments. *Journal of the Franklin Institute* **2015**, *352*, 2708–2727.
206. Lu, M.; Xing, L.; Zheng, N.; Chen, B. Robust sparse channel estimation based on maximum mixture correntropy criterion. 2020 International Joint Conference on Neural Networks (IJCNN). IEEE, 2020, pp. 1–6.
207. MacKay, D.J. A practical Bayesian framework for backpropagation networks. *Neural computation* **1992**, *4*, 448–472.
208. Tipping, M. The relevance vector machine. *Advances in neural information processing systems* **1999**, *12*.
209. Tipping, M.E. Sparse Bayesian learning and the relevance vector machine. *Journal of machine learning research* **2001**, *1*, 211–244.
210. Bishop, C.M.; Tipping, M.E. Variational Relevance Vector Machines. Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann, 2000, pp. 46–53.
211. Gelman, A.; Carlin, J.B.; Stern, H.S.; Rubin, D.B. *Bayesian data analysis*; Chapman and Hall/CRC, 1995.
212. Xing, L.; Mi, Y.; Li, Y.; Chen, B. Robust locality preserving projection based on kernel risk-sensitive loss. 2018 International Joint Conference on Neural Networks (IJCNN). IEEE, 2018, pp. 1–7.
213. Blei, D.M.; Kucukelbir, A.; McAuliffe, J.D. Variational inference: A review for statisticians. *Journal of the American statistical Association* **2017**, *112*, 859–877.
214. Loh, P.L.; Wainwright, M.J. Regularized M-estimators with nonconvexity: Statistical and algorithmic theory for local optima. *Advances in Neural Information Processing Systems* **2013**, *26*.
215. Chen, Y.; Caramanis, C.; Mannor, S. Robust sparse regression under adversarial corruption. International Conference on Machine Learning. PMLR, 2013, pp. 774–782.
216. Knauff, M.; Mulack, T.; Kassubek, J.; Salih, H.R.; Greenlee, M.W. Spatial imagery in deductive reasoning: a functional MRI study. *Cognitive Brain Research* **2002**, *13*, 203–212.
217. Strotzer, M. One century of brain mapping using Brodmann areas. *Clinical Neuroradiology* **2009**, *19*, 179–186.
218. Li, Y.; Chen, B.; Wang, G.; Yoshimura, N.; Koike, Y. Partial maximum correntropy regression for robust electrocorticography decoding. *Frontiers in Neuroscience* **2023**, *17*, 1213035.
219. Li, Y.; Chen, B.; Shi, Y.; Yoshimura, N.; Koike, Y. Correntropy-based logistic regression with automatic relevance determination for robust sparse brain activity decoding. *IEEE Transactions on Biomedical Engineering* **2023**.
220. Li, Y.; Chen, B.; Yamashita, O.; Yoshimura, N.; Koike, Y. Adaptive sparseness for correntropy-based robust regression via automatic relevance determination. 2023 International Joint Conference on Neural Networks (IJCNN). IEEE, 2023, pp. 1–8.
221. Kolda, T.G.; Bader, B.W. Tensor decompositions and applications. *SIAM review* **2009**, *51*, 455–500.
222. Zhang, M.; Gao, Y.; Sun, C.; La Salle, J.; Liang, J. Robust tensor factorization using maximum correntropy criterion. 2016 23rd International Conference on Pattern Recognition (ICPR). IEEE, 2016, pp. 4184–4189.