# Estimating the Memory Order of Electrocorticography Recordings

Yonathan Murin, *Member, IEEE*, Andrea Goldsmith, *Fellow, IEEE*, and Behnaam Aazhang, *Fellow, IEEE*

***Abstract*—*Objective:*** **This paper presents a data-driven method for estimating the memory order (the average length of the statistical dependence of a given sample on previous samples) of a recorded electrocorticography (ECoG) sequence. *Methods:* The proposed inference method is based on the relationship between the loss in predicting the next sample in a time-series and the dependence of this sample on the previous samples. Specifically, the memory order is estimated to be the number of past samples that minimize the least squares error (LSE) in predicting the next sample. To deal with the lack of an analytical model for ECoG recordings, the proposed method combines a collection of different predictors, thereby achieving LSE at least as low as the LSE achieved by each of the different predictors. *Results:* ECoG recordings from six patients with epilepsy were analyzed, and the empirical cumulative density functions (ECDFs) of the memory orders estimated from these recordings were generated, for rest as well as pre-ictal time intervals. For pre-ictal time intervals, the electrodes corresponding to the seizure-onset-zone were separately analyzed. The estimated ECDFs were different between patients and between different types of blocks. For all the analyzed patients, the estimated memory orders were on the order of tens of milliseconds (up to 100 ms). *Significance:* The proposed method facilitates the estimation of the *causal* associations between ECoG recordings, as these associations strongly depend on the recordings' memory. An improved estimation of causal associations can improve the performance of algorithms that use ECoG recordings to localize the epileptogenic zone. Such algorithms can aid doctors in their pre-surgical planning for the surgery of patients with epilepsy.**

***Index Terms*—Electrocorticography recordings, Markov order, non-parametric estimation, prediction.**

## I. Introduction

A COMMON approach in quantitative analysis of complex brain networks is to use graph theory, where the brain is modeled as a network, and the connectivity in the network is given by the estimated statistical dependency between the observed signals [1]. Several studies applied this approach using symmetrical measures of statistical dependence that do not involve the statistical dependence of a given sample on previous samples. In this work we estimate this dependence via a parameter we call the *memory order*, defined as the average length of previous samples with which a given sample has nonzero statistical dependence. Examples of such measures include correlation, coherence, and mutual information (see [1]–[6] and references therein). Another line of works considered asymmetric measures and models that are based on the memory of the observed signals. Examples of such models are dynamic causal models [7], directional dynamical models [8], and auto-regressive (AR) models [9]. In contrast to these approaches, where it is assumed that the observed data *follows a parametric model*, in some cases one is interested in quantifying the *causal* association between the observed signals (time-series) *in the absence* of a known model [10]–[12]. Since the causal association strongly depends on the signal's memory (and in general the order of this memory is not known), the order of the memory must be *estimated from the data* before estimating the desired statistical measure.

In this work we study the estimation of the memory orders of invasive EEG recordings, also known as electrocorticography (ECoG). In ECoG grids or strips of electrodes are placed on the cerebral cortex measuring its electrical activity. The common applications of ECoG are in identification of the seizure onset zone (SOZ) during pre-surgical planning in epileptic patients with focal epilepsy [3], [10], [11], and brain-computer-interfaces [13]–[15].

To the best of our knowledge *there is no established statistical model for ECoG recordings*. Thus, one method to estimate the underlying memory order of observed ECoG time-series is to use a *data-driven* (i.e., non-parametric) method. This is the approach we take in this paper. Specifically, in our data-driven approach to this problem we assume that the observed ECoG signal was generated by a random Markov process of order $M$ which we would like to estimate. We refer to $M$ as the memory order of the ECoG signal. Assuming that the observed ECoG signal, or more generally a time-series, follows a Markov model is a common assumption for systems that are known to have *finite memory*. For example, Markov models were used to model the memory of neural recordings in [16], [17], of financial models in [18] and [19], and in analyzing social network dynamics in [20], [21].

Manuscript received June 3, 2018; revised October 13, 2018; accepted January 16, 2019. Date of publication January 30, 2019; date of current version September 18, 2019. The work of Y. Murin and A. Goldsmith was supported in part by the National Science Foundation, Center for Science of Information (CSoI), under Grant NSF-CCF-0939370. The work of B. Aazhang was supported in part by the National Science Foundation under Grant NSF-1406447. *(Corresponding author: Yonathan Murin.)*

Y. Murin is with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA (e-mail: moriny@stanford.edu).

A. Goldsmith is with the Department of Electrical Engineering, Stanford University.

B. Aazhang is with the Department of Electrical and Computer Engineering, Rice University.

Digital Object Identifier 10.1109/TBME.2019.2896076

Previous studies that used the memory order of ECoG signals, see [10], [11], assumed this memory to be on the order of tens of milliseconds, yet, no extensive study of the memory order parameter was carried out to justify this assumption. In the current work we show that the memory order can be seen as a random variable with a distribution that may vary between recording electrodes, between different patients, and between different states of the brain (rest or the first seconds of an evolving epileptic seizure). Our results show that for all studied patients, and for both analyzed states, the distribution of the estimated memory order is supported in the range of 0 to 100 milliseconds, namely, most of the estimated memory orders were *significantly smaller than 100 milliseconds*. The proposed method can also be used to estimate the *joint* memory order of two time-series, thus facilitating a data-driven estimation of information theoretic functionals, such as directed information [22] and transfer entropy [23], that quantify the causal influence between two random processes [24].

The proposed data-driven approach uses techniques that originate from the machine learning literature: the Markov order is chosen to minimize the $\ell_2$ loss, also known as least squares error (LSE), in *predicting the next sample from the last $M$ samples*. This is motivated by observing that if the considered sequence is Markovian with order $M$, then prediction based on $m < M$ samples can (on average) be improved by increasing $m$. On the other hand, since the number of samples is finite, prediction (or learning) based on memory of $m > M$ samples might be less accurate since there are less samples per learned parameter. From an information theoretic perspective, the optimal predictor can extract the required information from the $m > M$ memory samples by ignoring the unnecessary information. However, in practice, since the optimal predictor is not known, it is possible that using $m > M$ samples will add undesired noise, leading to less accurate prediction. This degradation is illustrated by several examples in the simulation study presented in Section III. Furthermore, when the Markov order is used in the estimation of statistical functionals such as directed information (or transfer entropy), using $m > M$ can significantly decrease the estimation accuracy due to the fact that the considered state-space is larger than needed,[1] see [27, Sec. 5] for more discussion regarding this phenomenon.

Estimating the memory order via prediction implicitly assumes that the prediction method can efficiently predict the next sample from previous samples. However, as we do not have an underlying model for the observed ECoG signals, we do not know which prediction method is optimal. To (partially) account for this challenge we propose to use *a group of different predictors*, and select the one that obtains the smallest *empirical* loss for the *specific* sequence. This is motivated by the idea of boosting where several weak classifiers (or predictors) are combined into one stronger classifier (or predictor), resulting in a more powerful classification (prediction) method [28, Ch. 10]. To avoid overfitting we use the standard technique of splitting the observed sequences into training and test data; the prediction model is learned over the training data and the empirical loss is calculated over the test data [28].

As the class of possible predictors is of infinite cardinality, one cannot exhaustively search for the optimal one. Thus, we consider a finite-cardinality set of predictors, and we must hence decide *which predictors should be included in this set.* Our results were obtained by considering several common predictors: simple linear regression, linear regression with interacting terms, regression using pure quadratic terms, linear support vector regression (LSVR), support vector regression with Gaussian kernel (GSVR), and prediction based on $k$-nearest-neighbors ($k$-NN). The first five predictors apply a global prediction, i.e., they use the data to fit a model predicting the next sample. On the other hand, the $k$-NN is local in the sense that it predicts the next sample based on the local environment of the past samples, without fitting a joint model. As we show in the simulation study, when the signal changes fast and the memory order is small, $k$-NN can be very accurate.

We use different versions of linear regression to capture different (not necessarily linear) interactions between past samples, and the support vector machines are chosen to deal with general non-linear dynamics (as shown in the sequel, the LSVR is less useful). Note that while the GSVR may seem the predictor that covers the most general class of models, it depends on several hyper-parameters that must be carefully set. Thus, when the number of samples is finite and one is constrained by computational complexity considerations, it is not guaranteed that the GSVR will have the best performance. We emphasize that by considering prediction based on multiple predictors, one can reduce the empirical error (over the test data). Yet, this comes at the cost of increasing computational complexity, as discussed in more detail in Section V-C. Moreover, since the underlying model is not known, it is not clear if the optimal predictor (in the sense of achieving the average lowest possible $\ell_2$ loss) was considered in our group. While the set of predictors considered in this work reflects the current state-of-the-art in prediction, this method can be expanded to include additional predictors that currently exist or that arise in future research.

Analyzing the ECoG recordings we observed that the simple linear regression and the GSVR are most frequently selected as the predictor that achieves the lowest $\ell_2$ loss (note that neither one of these methods is universally better than the other in our analysis). The efficacy of the simple linear predictor may be explained by observing that the sampled signal changes significantly slower than the sampling rate. Thus, fitting a simple linear model can be done efficiently. Moreover, the simple linear predictor has the lowest number of parameters and therefore the highest number of samples per fitted parameter. Regarding the GSVR, as stated above, it represents the largest class of models compared to other used predictors, thus it is expected to have relatively good prediction performance (with the drawback of having several hyperparameters that need to be optimized).

We analyzed ECoG recordings taken from six epileptic patients, listed in the iEEG portal for epilepsy research [29]. These recordings were taken as part of the presurgical procedure for localizing the patients epileptogenic zones. We analyzed two types of time intervals: 10 seconds intervals when the patient is

[1]The number of samples required for accurate estimation grows exponentially with the state-space size, see [25] and [26].

resting (awake), and 10 seconds at the beginning of an epileptic seizure (the time intervals at the beginning of a seizure are commonly used in localizing the SOZ). When analyzing the time intervals at the beginning of a seizure we separately analyzed the recordings taken from the region identified as the SOZ, and recordings taken from a region that is not identified to be part of the SOZ. Treating the memory order as a RV, for each patient, and for each state of the brain (rest and pre-ictal), we generated an empirical cumulative density function (ECDF) of the estimated memory order. Our results indicate that in the majority of cases the memory order is *significantly smaller* than 100 milliseconds, namely, the distribution is supported on the interval 0 to 100 milliseconds.

The rest of this paper is organized as follows: The problem formulation, background on existing estimation techniques, and the proposed estimation method are presented in Section II. A simulation study of memory order estimation from non-linear noisy data (which supports the validity of the proposed method) is presented in Section III. The analysis of ECoG signals appears in Section IV, and the results are discussed in Section V. Concluding remarks are provided in Section VI.

## II. Problem Formulation and Methods

### A. Problem Formulation

We first introduce our notation. We denote random variables (RVs) by upper case letters, $X$, and their realizations with the corresponding lower case letters. We use the short-hand notation $X_1^N$ to denote the sequence $\{X_1, X_2, \ldots, X_N\}$. We denote random processes using boldface letters, e.g., $\mathbf{X}$, and sets using calligraphic letters, e.g., $\mathcal{A}$, where $\mathcal{R}$ denotes the set of real numbers and $\mathcal{R}_+$ denotes the set of positive real numbers.

Let $\mathbf{X}$ be a discrete-time continuous-amplitude random process, and let $X_1^N \in \mathcal{R}^N$ be an $N$-length sample path of $\mathbf{X}$. We assume that $\mathbf{X}$ is stationary, ergodic, and Markovian of order $M$ in the observed sequence. Stationarity implies that the statistics of the considered random process are constant throughout the observed sequence; ergodicity ensures that the observed sequences truly represent the underlying process; Markovity, as stated above, is a common assumption in modeling real systems with *finite* memory. The assumption of a Markov process with order $M$ is formulated as:

$$f_X(x_i|X_1^{i-1}) = f_X(x_i|X_{i-M}^{i-1}), \quad \forall i > M. \tag{1}$$

Note that from a practical perspective one can require the density of $X$ to obey $f_X(x_i|X_1^{i-1}) \approx f_X(x_i|X_{i-M}^{i-1}), i > M$. While it is a reasonable to assume that the underlying process is a Markov process (or at least approximately Markov), its order $M$ is in general not known. Our objective in this work is to *design a non-parametric approach for inferring $M$ from the observed sequence*.

### B. Background on Existing Techniques

The problem of estimating the Markov order of an observed time-series is also referred to as the model order selection problem (in this context it is implicitly assumed that the data follows a known model, for instance, the vector AR model). In such a case standard techniques like the Akaike or Bayesian information criteria are commonly used to adjust the maximum-likelihood (ML) estimation of the model order [30], thus, choosing a model order that avoids overfitting (yet, fits the data well). While this approach works well when the model for the data is known, it cannot be directly applied in the non-parametric setting.[2]

The work [31] also formulated the problem of inferring the memory order as a prediction problem, and suggested to infer it using Takens delay embedding theorem [32] while optimizing the embedding delay according to the Ragwitz criterion [33]. In this approach one predicts the sample $X_{n+1}$ based on the $k$-NN of the tuple $(X_{n-m+1}, \ldots X_n)$.[3] Here $k$ is a design parameter and $m$ belongs to the range of possible memory orders. The selected memory order is the one that minimizes the average prediction loss. While [33] proposed to use prediction via simple averaging of the $k$-NN responses, [34] and [35] proposed to replace the simple averaging with a linear regression. Prediction based on the $k$-NN is relatively computationally efficient and works well for small values of $M$ (as we show in Section III), however, when $M$ is large prediction based on the $k$-NN principle suffers from the *curse of dimensionality* (see the discussion in [28, Sections 2.5 and 6.3]).[4] The term *curse of dimensionality*, suggested in [36], refers to the phenomena where an algorithm that works well for low-dimensional input becomes intractable for high-dimensional input. Particularly, correct generalization becomes exponentially more difficult when the dimensionality of the input grows since the *fixed-size* training data covers a decreasing fraction of the input space.

A different approach for inferring the Markov order is by minimizing the delayed mutual information [37], namely, finding the value $m$ for which the (estimated) mutual information between $X_{n+1}$ and $(X_n, \ldots X_{n-m+1})$ first reaches its maximum. This idea was extended in [38] to using sequential forward selection of past samples based on conditional mutual information and in [39] to selecting an arbitrary subset of past samples (up to a given maximal delay). To estimate the mutual information these works use the estimator of [40] that builds upon the $k$-NN principle. Thus, similarly to the $k$-NN prediction methods, when $M$ is large, the methods based on estimating mutual information also suffer from the curse of dimensionality. Another drawback of these methods is their very high computational complexity (see the discussion in [39]).

### C. Description of the Proposed Estimation Method

From (1) it follows that optimal prediction of $X_i$ based on the past samples $X_1^{i-1}$ can be obtained by using only the last $M$ samples $X_{i-M}^{i-1}$. Moreover, following ideas similar to those

[2]Note that the ML estimation requires knowing the underlying probability density function of the data, and complete non-parametric estimation of a continuous density is considered to be a highly complicated task.

[3]The $k$-NN are found among the tuples $(X_{j-m+1}, \ldots X_j), j \notin \{n - 2m \cdots n + 2m\}$.

[4]All the discussed prediction approaches implicitly assume that the data is stationary and relatively smooth. In [2] it is argued that ECoG recordings are stationary only for few seconds and therefore the amount of data available for prediction is limited.

applied in [31], [33], [39], the value of $M$ can be estimated as the number of past samples that leads to the best prediction of future samples (on average). To formally quantify this argument, let $\mathcal{M}$ be the set of possible memory orders (this set is assumed to be known; for the case of ECoG signals we verify this assumption via censoring, see Section IV), and for a given $m \in \mathcal{M}$ let $Z_{i,m} \triangleq X_{i-m}^{i-1} \in \mathcal{R}^m, i > m$. Moreover, let $\ell : \mathcal{R} \times \mathcal{R} \to \mathcal{R}_+$ be a loss function, and $\varphi_m : \mathcal{R}^m \to \mathcal{R}$ be a predictor of $X_i$ from $Z_{i,m}$.[5] In this work we use the common $\ell_2$ loss, also referred to as LSE. Using the sequence of predictors $\varphi_m$, we define:[6]

$$\varepsilon_m = \frac{1}{N-m-1} \sum_{i=m+1}^{N} \ell(\varphi_m(Z_{i,m}), X_i), \tag{2}$$

and the memory order can be estimated via:

$$\hat{M} = \underset{m \in \mathcal{M}}{\operatorname{argmin}} \, \varepsilon_m . \tag{3}$$

Thus, the memory order that minimizes the *average empirical loss* is selected as the estimate. Minimizing the average empirical loss is motivated by treating $X_i$ and $Z_{i,m}$ as instances of the RVs $X$ and $Z_m$, respectively, and then interpreting $\varepsilon_m$ as an estimator of $\mathbb{E}\{\ell(\varphi_m(Z_m), X)\}$, where here the expectation averages over everything that is random. Note that the RVs $X$ and $Z_m$ are dependent. Unfortunately, as discussed in [28, Ch. 7], $\varepsilon_m$ is not a good estimate for $\mathbb{E}\{\ell(\varphi_m(Z_m), X)\}$ since it constantly decreases with $m$ due to overfitting to the specific observed sequences.

To address the overfitting problem we split the sequence $X_1^N$ into three parts: training, validation, and test. The training and validation parts are used to choose the best predictor, *for a given* $m$, between a set of possible predictors, while the test is used to estimate $\mathbb{E}\{\ell(\varphi_m(Z_m), X)\}$ (also known as the generalization error) and compare between the different elements of $\mathcal{M}$. Specifically, let $\mathcal{T}$ denote the training part, $\mathcal{V}$ denote the validation part, and $\mathcal{T}^*$ denote the test part. Further define $\varphi_{m|\mathcal{T},\mathcal{V}}(Z_{i,m})$ to be the prediction function learned from the training and validation parts. Then, the memory order can be estimated as:

$$M_0 = \underset{m \in \mathcal{M}}{\operatorname{argmin}} \frac{1}{|\mathcal{T}^*|} \sum_{i \in \mathcal{T}^*} \ell(\varphi_{m|\mathcal{T},\mathcal{V}}(Z_{i,m}), X_i). \tag{4}$$

In practice, as (4) applies an empirical averaging to approximate the expectation $\mathbb{E}\{\ell(\varphi_m(Z_m), X)\}$, we are interested in the *minimal* memory order that leads to a low loss in predicting the next sample. This follows since it is possible that, due to noisy perturbations of the data, the minimum in (4) is due to the specific data and might be larger than the effective memory order. This is undesirable since, as stated earlier, when the Markov order is used in the estimation of statistical functionals that incorporate memory, having $m > M$ can significantly decrease the estimation accuracy. Therefore, we define

$$L_0 \triangleq \frac{1}{|\mathcal{T}^*|} \sum_{i \in \mathcal{T}^*} \ell(\varphi_{M_0|\mathcal{T},\mathcal{V}}(Z_{i,M_0}), X_i)$$

to be the loss achieved by the memory order $M_0$, see (4). We further let $\delta > 0$ be a small number, and define the set $\mathcal{M}_0(\delta)$ as:

$$\mathcal{M}_0(\delta) = \left\{ m : \frac{1}{|\mathcal{T}^*|} \sum_{i \in \mathcal{T}^*} \ell(\varphi_{m|\mathcal{T},\mathcal{V}}(Z_{i,m}), X_i) \le L_0 \cdot (1+\delta) \right\}.$$

The memory order is now estimated via:

$$\hat{M} = \underset{m \in \mathcal{M}_0(\delta)}{\operatorname{argmin}} \frac{1}{|\mathcal{T}^*|} \sum_{i \in \mathcal{T}^*} \ell(\varphi_{m|\mathcal{T},\mathcal{V}}(Z_{i,m}), X_i). \tag{5}$$

Note the fundamental difference between (3) and (5): in (3) the predictor is trained over the *whole* observed data and the obtained average loss is used for selecting the model order. On the other hand, in (5) the predictor is trained on the training and validation parts, and the memory order is selected based on the loss calculated from the test part, thus better capturing the generalization error. The interpretation of (5) is that we choose the minimal memory order that achieves loss larger by at most a factor of $(1+\delta)$ compared to the minimal loss achieved over the whole set $\mathcal{M}$. Next, we elaborate on the prediction method $\varphi_m$.

### D. Prediction Method

As discussed in the introduction, instead of using a single prediction method (see, for example, [33]–[35]) we propose to use the training part of the data to train several predictors and estimate their resulting average loss using the *validation part of the data*. This is motivated by the fact that, to the best of our knowledge, there is no accepted statistical model for ECoG recordings. Since prediction methods are in general based on statistical models, the best approach to prediction in the absence of a statistical model is unclear. As we discuss in Remark 1 below, by leveraging the strength of different predictors we improve the total prediction power.

Denoting the $j^{\text{th}}$ predictor, trained using $\mathcal{T}$, by $\varphi_{m|\mathcal{T}}^{(j)}$, its loss is estimated *over the validation data* as:

$$\lambda_{m,j} = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \ell(\varphi_{m|\mathcal{T}}^{(j)}(Z_{i,m}), X_i). \tag{6}$$

Then, as $\varphi_{m|\mathcal{T},\mathcal{V}}$ we choose the estimator that results in the smallest loss *over the validation data*:

$$\varphi_{m|\mathcal{T},\mathcal{V}} = \varphi_{m|\mathcal{T}}^{(j_m^*)}, \quad j_m^* = \underset{j}{\operatorname{argmin}} \, \lambda_{m,j}. \tag{7}$$

*Remark 1 (Predictor selection vs. order estimation):* Note that the predictors are compared based on their loss calculated over the *validation* data. On the other hand, the memory order is estimated based on the loss values calculated over the *test* data. Thus, combining the different predictors can be viewed as considering a single compound predictor with an improved prediction power (compared to each individual predictor).

[5] Using the approach common in learning theory, $\varphi_m(\,)$ *is not a fixed predictor, but a predictor trained on the observed data.*

[6] Here $\varphi_m$ is trained over *all the observed data.*

In the current work we used $\delta = 0.01$, and considered the following six distinct predictors ($j \in \{1, 2, \ldots, 6\}$):

1) **Simple linear:** A linear regression model that includes an intercept and linear terms of $X_{i-m}^{i-1}$.
2) **Linear interactions:** A linear regression that includes an intercept, linear terms, and *all* products of pairs of distinct elements of $X_{i-m}^{i-1}$ (no squared terms).
3) **Pure quadratic:** A linear regression that includes an intercept, linear terms, and squared terms of $X_{i-m}^{i-1}$ (no interaction terms).
4) **LSVR:** A linear support vector regression, see [41, Ch. 2].
5) **GSVR:** A support vector regression with a Gaussian kernel, see [41, Ch. 2].
6) **$k$-NN:** The $k$-NN predictor proposed in [33] where Euclidean distance is used, and $k = 25$.

A few remarks are in order regarding these predictors:

*Remark 2 (Global prediction vs. local prediction):* The $k$-NN predictor is fundamentally different than the other predictors in the sense that, for this predictor, each sample is *separately* predicted based on its $k$ closest neighbors (in the $Z_{i,m} = X_{i-m}^{i-1}$ space). On the other hand, in methods (1)–(5) we fit a predictive model based on *all* training samples and this model is then used for prediction.

*Remark 3 (Hyperparameters):* The SVR methods have a set of hyperparameters that can significantly effect the achieved loss, namely, one *must* optimize over the set of hyperparameters as the achieved loss can significantly differ from one configuration to another. Specifically, for LSVR one should set the $\epsilon$-sensitivity and box constraint values, while for GSVR one should in addition set the scale of the Gaussian kernel (see [41, Ch. 2] for a detailed discussion regarding the importance of these parameters). Unfortunately, using fixed values for these parameters leads to poor results in some of the ECoG recordings. Thus, these hyperparameters should be optimized for each data-set. In this work we optimized the hyperparameters using a grid search:[7] We first used a coarse logarithmic grid and trained the relevant SVR predictor using the specific hyperparameters configuration. We evaluated the loss for each configuration over the validation part of the data. We then found the configuration resulting in the minimal loss and repeated the above process with a linear grid around this configuration. The configuration that yielded the minimum loss was then selected and its average loss was calculated over the test part of the data.

*Remark 4 (The value of $k$):* The value of $k$ in the $k$-NN predictor can also be viewed as a hyperparameter. Testing a range of values of $k$ we noticed that the results do not differ much. Therefore, we fixed $k = 25$ which seemed to yield a good bias-variance tradeoff.

*Remark 5 (Data splitting):* The results in the following sections were obtained using 70% of the data for training, 15% for validation, and 15% for test.

We summarize the proposed estimation method in Algorithm 1.

[7]We also tried a random search, yet, this resulted in higher loss values.

**Algorithm 1:** Data-Driven Estimation of the Memory Order.

```
Normalize the observed sequence to have zero mean and unit variance (zscore)
for m ∈ M do
    for j = 1,2,...,6 do
        for every hyperparameter configuration do
            Train a model using T
            Test the trained model using V
            Select the configuration with the minimal loss
        end for
    end for
    Select φ_{m|T,V}^{(j_m^*)} via (6)–(7)
end for
Estimate the memory order via (5)
```

## III. Simulation Results

Next, we test the proposed estimation method in several simulated test cases.

### A. Noisy Sum of Sines

The first example is motivated by a scenario discussed in [42, Eq. (8)]. We first generated 10000 samples using the mapping (the first samples were set to zero):

$$y_{n+d} = \sum_{j=0}^{3} \sin((j+1) \cdot y_{n+j} + j + 1), \quad d \geq 4. \tag{8}$$

Then, we normalized the empirical variance of the sequence $y_1^N$ to unity (omitting initialization effects). Finally we generated the sequence $X_1^N$ via

$$X_n = y_n + \alpha Z_n, \tag{9}$$

where the samples $Z_n \sim \mathcal{N}(0, 1)$ are independently and identically distributed (i.i.d.) and $\alpha \geq 0$ sets the signal-to-noise ratio (SNR). The memory order was estimated from the last 1000 samples of the sequence $X_1^N$. For $\alpha = 0$, it is clear that the memory order of this time-series is $d$ since $x_{n+d}$ is a deterministic function of $x_{n+3}, \ldots x_n$. For $\alpha > 0$ this deterministic relationship no longer holds, yet the memory order is still $d$ since $Z_n$ is independent of all other variables. Fig. 1 depicts the average $\ell_2$ loss (over 20 independent trials) achieved by the proposed algorithm for $\mathcal{M} = \{1, 2, \ldots, 8\}$, $d = \{4, 5\}$ and several values of $\alpha$. Note that $\alpha = 0.1, 0.199, 0.316$ correspond to SNR $= 10$dB, 7dB, 5dB, respectively. It can be observed that for all simulated configurations the correct memory order is estimated. Somewhat surprisingly, even though it is local and not global, the most efficient predictor for all the examined configurations was the $k$-NN predictor ($k$-NN was selected via (6)–(7)). While, the *Simple linear* and *LSVR* constantly resulted in a wrong estimation of the memory order (the average $\ell_2$ loss was almost flat as a function of $m$), the other predictors were able to correctly estimate the model order in some of the configurations. After the $k$-NN predictor, the most accurate one was the *Pure quadratic*, yet, its

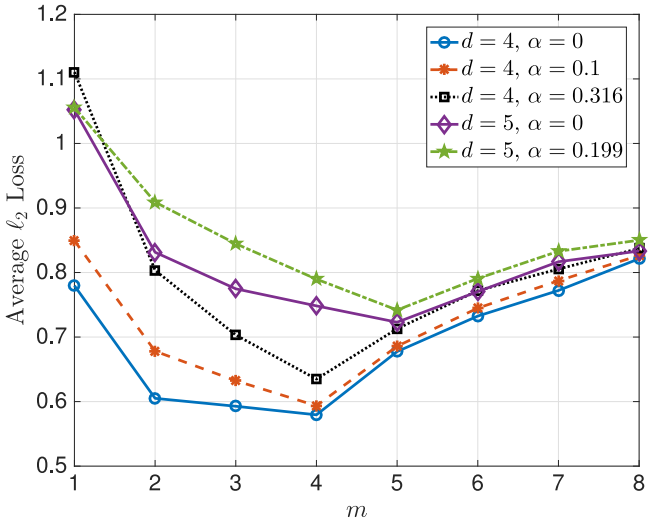


Fig. 1. Average $\ell_2$ loss versus tested memory order for the sum of sines model (8).

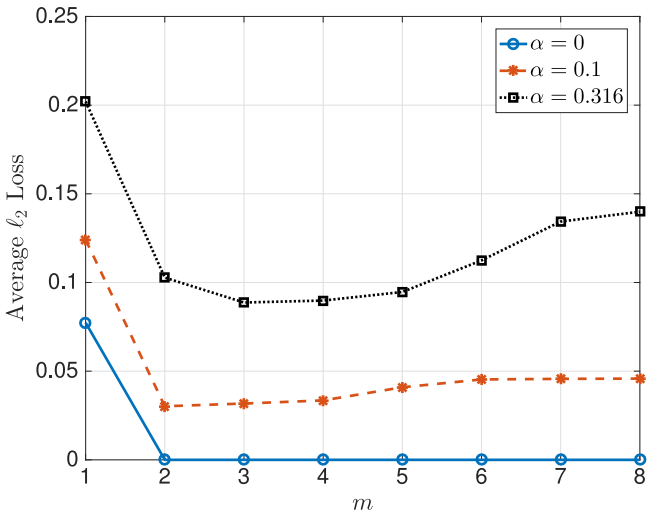


Fig. 2. Average $\ell_2$ loss versus tested memory order for the Hénon map (10).

average $\ell_2$ loss was significantly higher then the one achieved by $k$-NN.

### B. Hénon Map

We next consider a time series generated based on the Hénon map, a discrete-time dynamical system that exhibits chaotic behavior and is parametrized by $(a, b) \in \mathcal{R}^2$, see [43]. We generated 10000 samples as (initial values set to zeros)

$$y_{n+2} = 1 - a y_{n+1}^2 + b y_n . \tag{10}$$

We then normalized the empirical variance of the sequence $y_1^N$ to unity (omitting initialization effects), and generated the sequence $X_1^N$ via (9) with $\alpha \geq 0$ which sets the SNR. The memory order was estimated from the last 1000 samples of the sequence $X_1^N$. The memory order of this map is 2. Fig. 2 presents

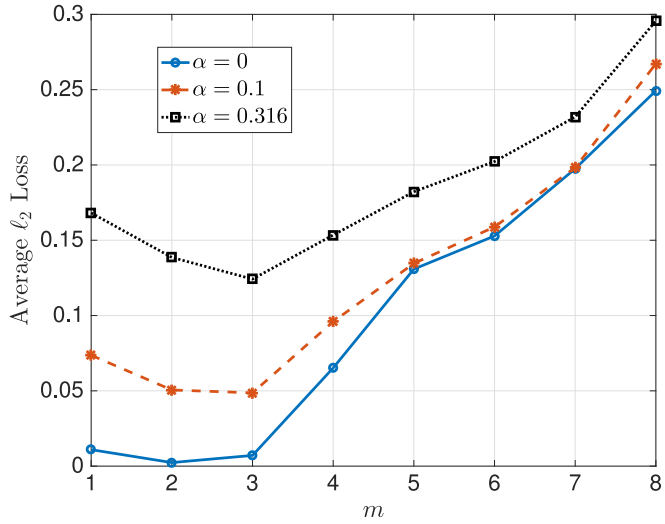


Fig. 3. Average $\ell_2$ loss versus tested memory order for the Duffing map model (11).

the average $\ell_2$ loss (over 20 independent trials) achieved by the proposed algorithm for $(a, b) = (1.4, 0.3)$; this pair results in a stable map. Here, again, we set $\mathcal{M} = \{1, 2, \ldots, 8\}$, and tested three values of $\alpha$. It can be observed that for the case of $\alpha = 0$ the average $\ell_2$ loss is practically zero for any $m \geq 2$. This is achieved by the *Pure quadratic* predictor that assumes exactly the same structure as implemented in the map (10). In this case the $k$-NN performs only slightly worse. The picture changes when one considers noisy observations ($\alpha > 0$). In this case the $k$-NN achieves a loss lower than the *Pure quadratic* predictor, yet the gap is not large. It can be observed that for $\alpha = 0.1$ the algorithm infers $\hat{M} = 2$, while in the more noisy case of $\alpha = 0.316$ the algorithm infers $\hat{M} = 3$.

### C. Duffing Map

We next consider the Duffing chaotic map which involves a power (of previous elements) higher than 2. For this setting we first generated 10000 samples as (initial values set to (0.15, 0.15))

$$y_{n+2} = -b y_n - a y_{n+1} - y_{n+1}^3 . \tag{11}$$

Then, as before, we normalized the empirical variance of the sequence $y_1^N$ to unity (omitting initialization effects), and generated the sequence $X_1^N$ via (9) with $\alpha \geq 0$ which sets the SNR. The parameters of the map were set to $(a, b) = (2.75, 0.15)$, yielding a stable map, while $\mathcal{M} = \{1, 2, \ldots, 8\}$. Fig. 3 depicts the average $\ell_s$ loss (over 20 independent trials) achieved by the proposed algorithm. As before, we tested three values of $\alpha$, corresponding to the noiseless case, SNR of 10dB, and SNR of 5dB. In contrast to the Hénon map, here none of the predictors match with the structure of the underlying time-series. Even for the noiseless case, the $k$-NN achieved loss significantly lower than the other predictors, and the other predictors did not yield correct estimations. It can be observed that for the noiseless case the algorithm estimates $\hat{M} = 2$, while for SNRs of 10dB and 5dB, $\hat{M} = 3$ is estimated.

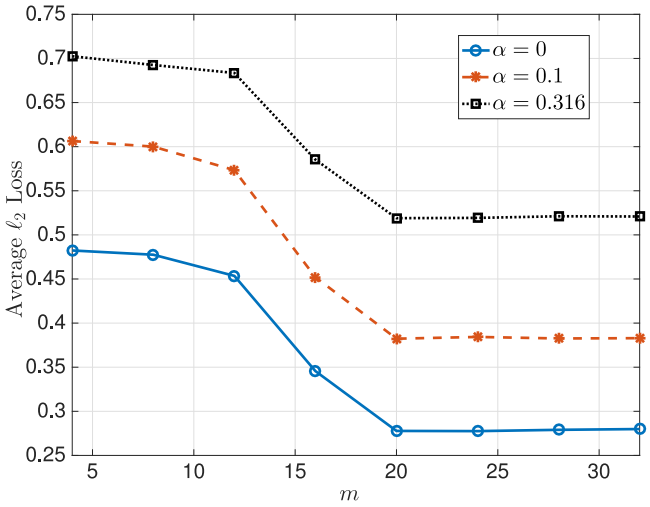


Fig. 4. Average $\ell_2$ loss versus tested memory order for the noisy auto-regressive process (12).

### D. Auto-Regressive Process With Medium Memory

In all previous scenarios the memory order was relatively short (up to five samples). On the other hand, in ECoG recordings we expect to observe significantly longer delays (tens of samples). To test this aspect we first generated a noisy auto-regressive process as follows:

$$Y_{n+19} = -\frac{1}{4}\sum_{j=0}^{3} Y_{n+j} + W_{n+19}, \tag{12}$$

where $W_n \sim \mathcal{N}(0,1)$ are i.i.d. Then we normalized the empirical variance of the sequence $Y_1^N$ to unity (omitting initialization effects), and created the sequence $X_1^N$ via (9). As before, $\alpha$ sets the SNR. The memory order in the noisy sequence $X_1^N$ is 19. We generated 10000 samples (initial values were uniformly distributed in [0,1]) and estimated the memory order from the last 4000 samples. This was repeated for three values of $\alpha$ corresponding to the noiseless case, and SNRs of 10db and 5dB. For the memory order range we use $\mathcal{M} = \{4, 8, \ldots, 32\}$. Fig. 4 depicts the average $\ell_2$ loss achieved by the proposed algorithm for the three values of $\alpha$. In contrast to the previous sections, here $k$-NN achieves the *highest* average loss among all predictors. We believe that this is due to two reasons. First, the structure of the generated data is simpler, thus enabling the other predictors to take advantage of their global nature. Second, as discussed in the introduction, when the memory order is high, $k$-NN is predicted to perform poorly due to the curse of dimensionality. We also observed that other than the $k$-NN predictor, all the predictors achieved very similar losses. Fig. 4 indicates that for all values of $\alpha$ the proposed algorithm estimated the memory order to be 20, which is the closet value in the set $\mathcal{M}$ to the actual memory order.

## IV. Memory Order of ECoG Recordings

We next study the memory order of ECoG recordings. Specifically, we apply our proposed algorithm, described in Section II, to estimate the memory orders of ECoG recordings listed in the International Epilepsy Electrophysiology (iEEG) portal [29].

### A. Description of the Analyzed Data

We analyze ECoG recordings taken from six epileptic patients (data-sets), all listed in the iEEG portal, see Table I for specific information per patient. The data-sets, in addition to the raw ECoG recordings, contain annotations information that indicates the state of the patient in a given time interval (resting, a seizure have started - pre-ictal, etc.). The labeling of these time intervals was done by expert epilepsy neurologists based on video recordings of the patients and the recorded signals.

In this work we analyze three types of recordings:

1) Recordings taken from electrodes that *are not identified to belong to the SOZ* over a time interval of 10 seconds at the *beginning* of the seizure. This is a 10 seconds time interval starting at the first sample indicated in the annotations to correspond to a pre-ictal state (note that in all considered data-sets the ictal state was longer than 10 seconds). We refer to these recordings as **pre-ictal blocks**.
2) Recordings taken from electrodes that *are identified to belong to the SOZ* over a time interval of 10 seconds at the *beginning* of the seizure. We refer to these recordings as **SOZ blocks**. Note that, since in focal seizures the abnormal neural activity originates from the SOZ, this area is commonly identified based on recordings taken at the beginning of seizures [10], [11], [44].
3) **Rest blocks** are *randomly sampled* from intervals that exclude seizures, artifacts, and blocks just before and after seizures. In this time, the patient is resting (awake). These blocks should indicate on the normal (non-seizure) activity of the brain.

The length of the analyzed blocks was chosen to be 10 seconds. This length provides a good tradeoff between having enough samples for analysis (see the sampling rates in Table I) and having the analyzed signals *approximately* stationary. Note that according to [2] ECoG signals are approximately stationary over a few seconds, while in [11] and [45] this duration was taken to be 10 seconds. Moreover, the works [10] and [44] assumed an approximate stationarity over longer time intervals of several tens of seconds. Fig. 5 provides an example for the raw recordings, taken from data-set `Study_020`.

### B. Pre-Processing

Before estimating the memory order using the method discussed in Section II, the data is pre-processed:

1) First, the common reference is removed from all the recorded signals as discussed in [4] (the work [4] used the same database).
2) Then, each recording is filtered using a 60 Hz notch filter to remove the line-noise captured in the recording process [4], [46]. While filtering the line-noise slightly increases the memory order, the filters' impulse response decays very fast, thus, its impact on the estimated

TABLE I

PATIENT INFORMATION. DATA-SETS `Study_020` – `Study_037` WERE RECORDED AT MAYO CLINIC, ROCHESTER, MN; DATA-SET `HUP70_phaseII` WAS RECORDED AT THE HOSPITAL OF THE UNIVERSITY OF PENNSYLVANIA, PHILADELPHIA, PA. RF - RIGHT FRONTAL, LF - LEFT FRONTAL, LP - LEFT PERIROLANDIC, CP - COMPLEX-PARTIAL, CPG - COMPLEX PARTIAL WITH SECONDARY GENERALIZATION, GA - GENERALIZED ATONIC, SP - SIMPLE PARTIAL, NR - NO RESECTION

| Patient (iEEG Portal) | Sex | Seizure Onset | Seizure Type | #Seizures | SOZ Size | Grid Size | Grid Name | Sampling Frequency | Outcome Class |
|---|---|---|---|---|---|---|---|---|---|
| `Study_020` | M | RF | CPG | 8 | 9 | $4 \times 6$ | RAG | 500 Hz | IV |
| `Study_021` | M | RF | CPG | 13 | 11 | $6 \times 8$ | RFG | 500 Hz | I |
| `Study_022` | F | Unknown | CPG | 7 | 11 | $4 \times 6$ | TIG | 500 Hz | V |
| `Study_033` | M | LF | GA | 17 | 6 | $8 \times 8$ | LG | 500 Hz | V |
| `Study_037` | F | Unknown | CP | 8 | 9 | $8 \times 8$ | RPG | 500 Hz | NR |
| `HUP70_phaseII` | M | LP | SP | 8 | 8 | $8 \times 8$ | RG | 512 Hz | NR |

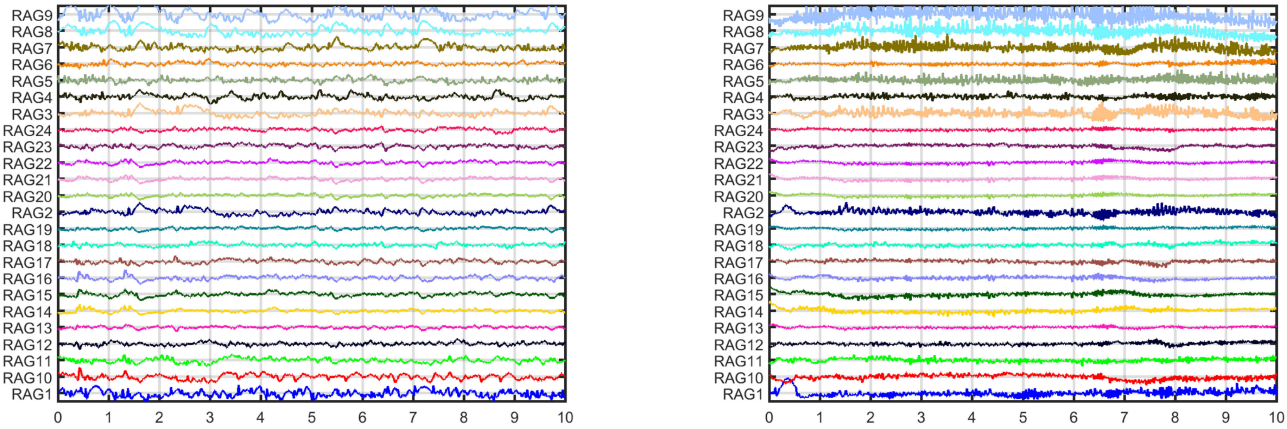


Fig. 5. Raw recordings taken from data-set `Study_020`. On the left - an example of a rest block. On the right - an example of a pre-ictal block.

memory order is minor (we use the filter provided in https://www.mathworks.com/help/dsp/ref/iirnotch.html).

### C. Estimation Results

Let the number of the electrodes in the recording grid be denoted by $A_{\text{Grid}}$, for instance, for `Study_020` $A_{\text{Grid}} = 24$, while for `Study_033` $A_{\text{Grid}} = 64$, see Table I. Thus, *for each block* (time interval of 10 seconds), we observe $A_{\text{Grid}}$ recorded ECoG *time-series*. Further, let $A_{\text{rest}}$ and $A_{\text{pi}}$ denote the *number of analyzed rest and pre-ictal blocks*, respectively. For the rest blocks (for all data-sets), $A_{\text{rest}}$ was chosen to have about 600 ECoG time-series in total. For instance, for `Study_020`, which contains 24 time-series in a block (24 electrodes), we analyzed $A_{\text{rest}} = 25$ blocks, while for `Study_033` we analyzed $A_{\text{rest}} = 10$ blocks. Since most of the data-sets include only a limited number of recorded seizures, we analyzed all the seizures.

Figs. 6–11 depict the *empirical CDFs* of the estimated memory orders, in milliseconds, for each of the data-sets. When generating these ECDFs we used $\mathcal{M} = 5, 8, \ldots, 50$. To capture the variations in the estimated memory orders (for a given data-set and state of the brain), Figs. 6–11 depict the *empirical CDFs* of the estimated memory orders, in milliseconds. When generating these ECDFs we used $\mathcal{M} = 5, 8, \ldots, 50$. We chose to present the ECDF and not estimate the density, since estimating the density is very sensitive to the estimation method and its parameters (e.g., scale of the kernel), while obtaining the ECDF is significantly more robust. The ECDF also provides valuable information about the different quantiles of the data. Since retrieving information about the shape of the density from the ECDF is challenging, we also provide the mean, standard deviation (Std), and median of the estimated memory orders in Table II. Finally, Table II also presents a measure of the average goodness of fit per data-set and block type. As a measure

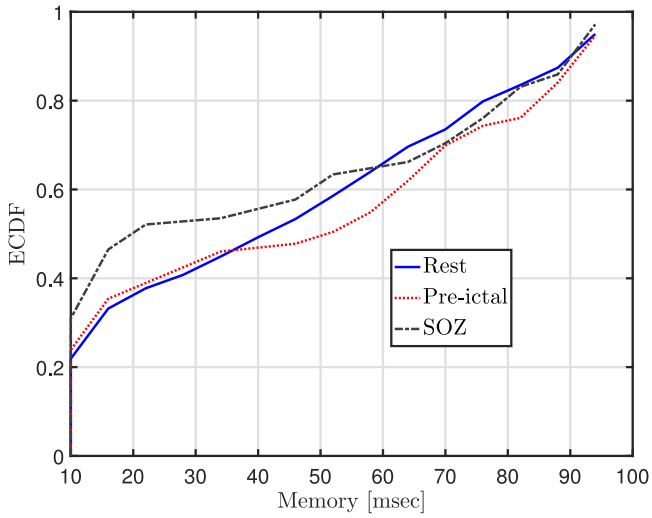


Fig. 6. ECDF of the estimated memory orders, in milliseconds, for data-set `Study_020`.

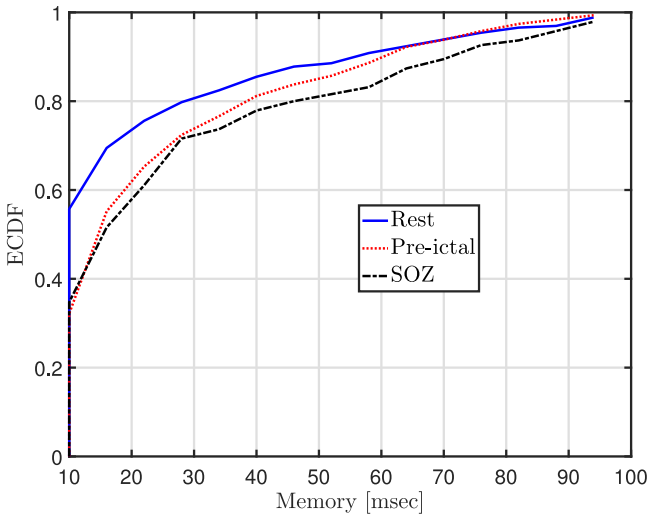


Fig. 7. ECDF of the estimated memory orders, in milliseconds, for data-set `Study_021`.

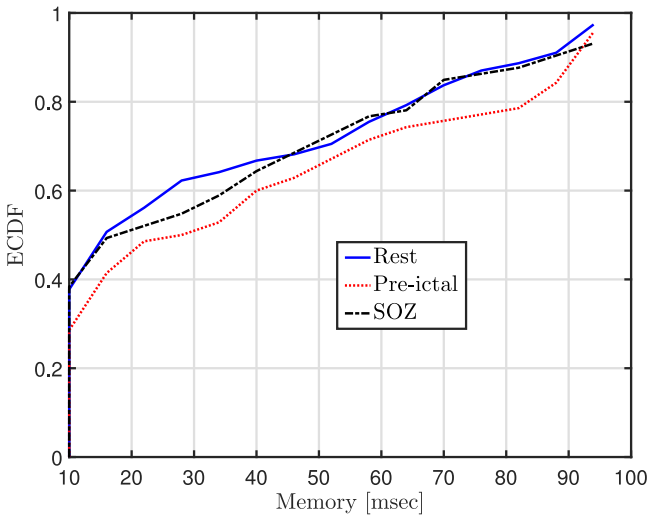


Fig. 8. ECDF of the estimated memory orders, in milliseconds, for data-set `Study_022`.

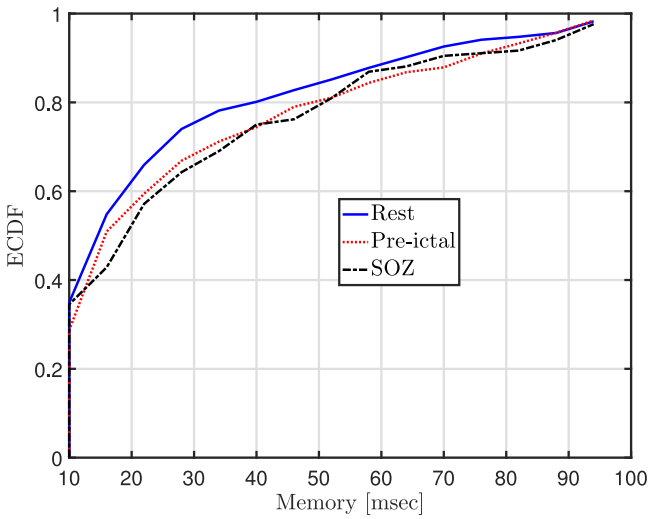


Fig. 9. ECDF of the estimated memory orders, in milliseconds, for data-set `Study_033`.

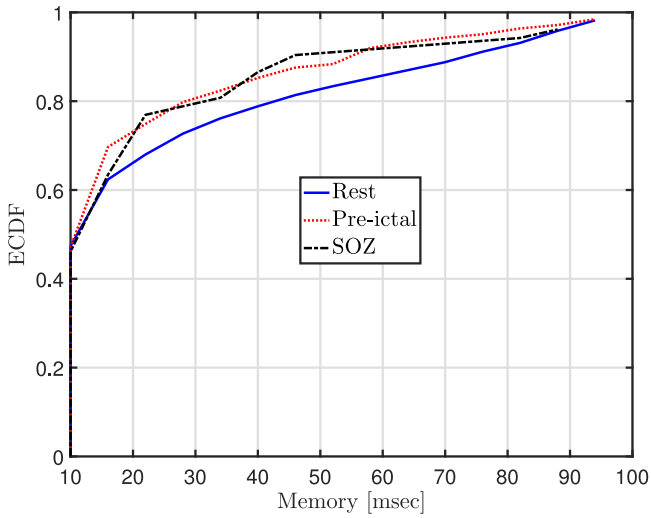


Fig. 10. ECDF of the estimated memory orders, in milliseconds, for data-set `Study_037`.

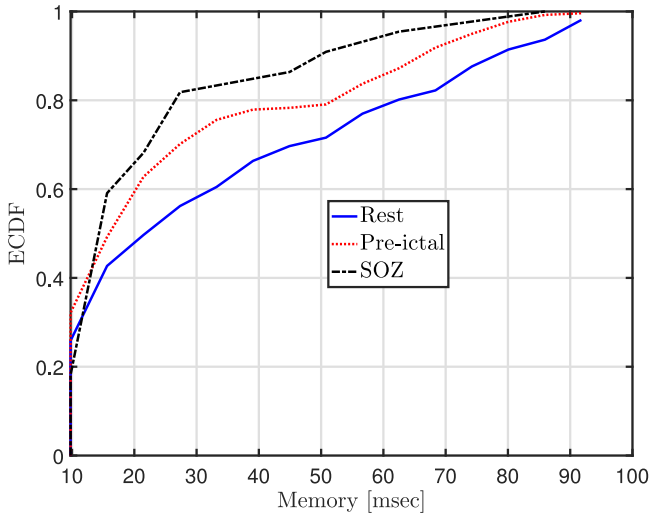


Fig. 11. ECDF of the estimated memory orders, in milliseconds, for data-set `HUP70_phaseII`.

of goodness of fit we use the root-mean-square-error (RMSE), evaluated on the test data. The values specified in Table II are the RMSE averaged over all time-series corresponding to the same data-set and block type.

It can be observed that the ECDFs are fundamentally different from one data-set (patient) to another and between different types of blocks. This can be due to a fundamental difference between the brain of the different patients, or due to the brain region where the grid was located (see Table I for the area in the brain where the grids are located), or due to the proximity of the grid to the SOZ. Possible other causes for a difference in the estimated memory orders can be the age of the patient or even its dominant hemisphere. This difference can also be observed in Table II where the mean, Std, and median vary between data-sets. A common phenomena in all the data-sets is a significant density mass at the low tested values. At the same time we note that in some of the data-sets this mass is larger and high estimated memory orders are less frequent, see, for instance,

TABLE II
STATISTICAL PARAMETERS OF THE ESTIMATED MEMORY ORDERS. EMPIRICAL MEAN, STANDARD DEVIATION, MEDIAN (IN MILLISECONDS) OF THE ESTIMATED MEMORY ORDERS, AS WELL AS THE AVERAGE RMSE FOR THE DIFFERENT TYPES OF DATA (REST, PRE-ICTAL, AND SOZ)

| Data-set | Data Type | Mean | Std | Median | Average RMSE |
|---|---|---|---|---|---|
| Study_020 | Rest | 45.675 | 31.381 | 40 | 0.033 |
| | pre-ictal | 49.13 | 33.609 | 52 | 0.025 |
| | SOZ | 42.78 | 33.878 | 22 | 0.023 |
| Study_021 | Rest | 22.375 | 21.827 | 10 | 0.046 |
| | pre-ictal | 27.41 | 22.43 | 16 | 0.039 |
| | SOZ | 29.44 | 25.82 | 16 | 0.038 |
| Study_022 | Rest | 35.991 | 30.449 | 22 | 0.048 |
| | pre-ictal | 41.885 | 32.904 | 31 | 0.081 |
| | SOZ | 36.630 | 30.756 | 22 | 0.092 |
| Study_033 | Rest | 26.437 | 23.159 | 16 | 0.045 |
| | pre-ictal | 29.231 | 25.664 | 16 | 0.034 |
| | SOZ | 30.112 | 25.995 | 22 | 0.037 |
| Study_037 | Rest | 27.213 | 25.758 | 16 | 0.055 |
| | pre-ictal | 22.613 | 21.179 | 16 | 0.063 |
| | SOZ | 23.104 | 22.401 | 16 | 0.058 |
| HUP70_phaseII | Rest | 35.767 | 27.537 | 21.484 | 0.039 |
| | pre-ictal | 30.457 | 23.064 | 21.484 | 0.066 |
| | SOZ | 24.946 | 19.446 | 15.625 | 0.067 |

Study_020 vs. Study_021 in Figs. 6–7. We did not observe a consistent relationship between the memory orders estimated from electrodes corresponding to the SOZ and memory orders estimated from electrodes in pre-ictal blocks that are not identified as part of the SOZ. We also did not observe a consistent relationship between pre-ictal and rest blocks. For instance, in some of the data-sets the memory order corresponding to the SOZ electrodes are smaller (e.g., Study_20), while in others the opposite holds (e.g., Study_33).

Since the tested range of memory orders is finite (the largest value is about 100 milliseconds), when $m = 50$ was estimated we can only conclude that the memory order is *at least* about 100 milliseconds. Therefore, when generating the ECDFs we used right censoring; this is reflected in Figs. 6–11 by the fact that the ECDFs do not reach the value 1. Closely observing Figs. 6–11 it can be noticed that in all data-sets, less than 6% of the values were censored. In fact, in most of the cases less than 4% of the values were censored. Thus, *with high probability the memory order is smaller than 100 milliseconds.* This upper bound on the memory order implies that most ECoG recordings contain information in frequencies larger than 10 Hz (recall that large memory order implies slowly decaying dependence, and this translates into narrower spectral content centered around DC). Figs. 6–11 also show that there is a non-negligible probability mass corresponding to the low memory orders. Recalling that short time dependence corresponds to information in relatively high frequencies, this may supprot using high frequency oscillations (oscillations in frequencies larger than 60Hz, which corresponds to about 16 miliseconds) as good bio-markers in epilepsy [47]–[49]. At the same time we note that, as stated in [47], to record high frequency oscillations, the ECoG recordings better be sampled at a minimum sampling frequency of 2 KHz.

In addition to the estimation results reported in Figs. 6–11, it is important to verify the statistical significance of the estimated values. Since the null-distribution of the estimated values is not known, we tested the statistical significance of the *achieved prediction loss* using the non-parametric stationary bootstrap procedure developed in [50]. Specifically, our results build upon the Markovity assumption (1), namely, the last $M$ samples can help in predicting the next sample. By shuffling the data as discussed in [50], this structure can be destroyed. Then, by applying the prediction method discussed above we can create an empirical (bootstrapped) null distribution for the estimate loss, for every $m \in \mathcal{M}$. For all the results reported in Figs. 6–11, the obtained loss is significantly lower than the loss resulting from bootstrapping.

Finally, we note that one does not have a ground truth when inferring the memory order of ECoG recordings. Moreover, since to the best of our knowledge there is no accepted statistical model for these signals, deriving performance guarantees is a highly complicated challenge. Based on our experience, simulation study, and by comparing different prediction methods, we believe that the ECDFs reported in Figs. 6–11 are reliable and accurately represent the true memory orders.

## V. DISCUSSION

In this section we discuss several aspects of the proposed estimation method with respect to the results presented in the previous section.

### A. Relation to Localization of the SOZ via Estimation of Statistical Functionals With Memory

In recent years the use of measures of causal influence (in particular directed information [22] and transfer entropy [23]) for model free analysis of causal interactions in neuroscience has seen a dramatic rise of interest, see, for example, the works [51]–[57]. In particular, the works [10], [11], [44] used the directed information functional[8] (note that under mild assumptions directed information is equivalent to transfer entropy) as a fundamental building block in novel algorithms for localizing the SOZ in patients with focal epilepsy. Motivated by the observation that focal epileptic seizures *start at the SOZ and then spread*, these works estimated the pair-wise directed information (causal influence) between different recorded signals, and then applied graph processing to estimate the location of the SOZ.

The efficacy of such algorithms strongly depends on accurate estimation of the causal influences, which, in turn, requires accurate knowledge of the memory order of the considered processes (see the numerical examples in [24] and [27, Sec. 5]). As in practice the memory order of an observed ECoG recording is not known, one must estimate it from the data prior to estimating the causal influence. The framework proposed in the current paper provides a tool for non-parametric estimation of the memory

[8]We refer the reader to [27] for the definition of directed information between discrete-time continuous-amplitude processes, as well as for a discussion on how this functional can be estimated.

order. This can facilitate the estimation of directed information in such algorithms, thus leading to an improved localization of the SOZ.

When estimating statistical functionals with memory (such as directed information) from observed sequences, two contradicting constraints must be taken into account. On the one hand, the observed sequences should be (approximately) stationary. This usually limits the number of samples available for estimation. On the other hand, as stated in [25, Thm. 1], the number of samples required for accurate estimation grows *exponentially* with the assumed memory order (this follows as increasing the memory order increases the state space [28]). Thus, when the assumed memory order is large the number of samples required for accurate estimation is enormous. In view of the first constraint, *if the memory order is too large, one cannot hope to accurately estimate the causal influence (directed information) from the observed sequences*.

The algorithm proposed in the current work can help dealing with the above challenge. Based on accurate estimation of the model order (or even based on knowledge of the range of the possible model orders), one can down-sample the observed sequences, thus, still accounting for most of the memory in the observed signals. This approach was taken in [11], where the down-sampling factor was heuristically chosen. A non-parametric method for estimating the memory order can improve this process, and the resulting estimations. In a similar manner, the algorithm proposed in the current work can also enhance the approach taken in [10], where it was proposed to decimate the memory in the estimation procedure by a heuristically chosen fixed factor. The technique of [10] can also be optimized using an accurate estimate of the memory orders of the raw signals that are obtained through our algorithm.

In conclusion, the method proposed in the current work can enhance the estimation of causal associations between ECoG recordings proposed in prior works, as these prior estimation methods strongly depend on the memory order of the underlying processes. Moreover, as the estimation of causal associations lies at the heart of several algorithms for localizing the SOZ, the performance of such algorithms can also be improved using the proposed method to estimate the memory order.

### B. Joint Memory Order and Feature Selection

Our approach can be extended to estimating the *joint* memory order involving two sequences $X_1^N, Y_1^N \in \mathcal{R}^N$. Formally, (1) can be extended as:

$$f_X(x_i|X_1^{i-1}, Y_1^{i-1}) = f_X\left(x_i|X_{i-M_X^{(X)}}^{i-1}, Y_{i-M_Y^{(X)}}^{i-1}\right), \quad (13)$$

for $i > \max\{M_X^{(X)}, M_Y^{(X)}\}$, where $M_X^{(X)}$ and $M_Y^{(X)}$ are the corresponding memory orders reflecting the dependence of $X_i$ on previous samples of $X$ and $Y$.[9] Now, instead of predicting $X_i$ based on $X_{i-m_X}^{i-1}$, one predicts based on $\{X_{i-m_X}^{i-1}, Y_{i-m_Y}^{i-1}\}$ for $m_X \in \mathcal{M}_X^{(X)}$ and $m_Y \in \mathcal{M}_Y^{(X)}$. It should be noted, however, that in such cases, for a fixed number of samples, due to the larger state space, one might need to use less complicated prediction models.

The above approach can be also used for feature selection (in the context of prediction). Assume that one is interested in predicting $X_i$ based on its past and asks which of the sequences $Y_1^{i-1}$ or $V_1^{i-1}$ can result in better prediction performance. Further assuming that the dependence on previous samples is finite (Markovity), we can evaluate the average test loss for predicting the next sample of $X$ for a range of joint memory orders $(M_X^{(X)}, M_Y^{(X)})$ and $(M_X^{(X)}, M_V^{(X)})$, and choose the configuration with the smallest empirical loss. Thus, we simultaneously estimate the joint memory order and infer which sequence better improves the prediction.

### C. Computational Load

One of the main challenges in obtaining the results reported in Figs. 6–11 is the required computational effort. While the computational complexities of the simple linear regression, linear interactions, pure quadratic, and $k$-NN predictors are moderate, training the support vector regression predictors requires a significant computational effort and therefore induces a significant processing time. One of the key reasons for this high computational complexity is the selection of optimal hyperparameters, which requires executing the training tens or even hundreds of times for each block and each memory order.[10]. If one is interested in inferring the memory order of a single time-series, prediction for each of the candidate memory order values can be executed in parallel (16 parallel processors), thus resulting in a reasonable computation time. On the other hand, generating Figs. 6–11 cannot be quickly executed in parallel due to the extremely high number of predictions (generating Figs. 6–11 requires about 120,000 predictions, not including hyper-parameter optimization that adds a factor of few hundreds).

### D. Selected Predictors

Recall (7) where, for a given $m$, we select the predictor that minimizes the average loss. Then, (5) selects the memory order as the one that minimizes the loss among all candidate memory orders. A natural question is: *what is the frequency of being selected for each of the considered predictors?*

To answer this question, Table III provides the portion of the data where each predictor was selected, namely, where each predictor obtained the minimum loss among all predictors and all tested memory orders. While Table III presents the results for data-sets `Study_033`, `Study_037` and `HUP70_phaseII`, similar trends were observed in all analyzed data-sets. Table III indicates that the frequently selected predictors, for rest blocks, are the *simple linear predictor, the GSVR, and the pure quadratic predictor*, in this order. On the other hand, for pre-ictal and SOZ blocks the order is *GSVR, the simple linear predictor, and the pure quadratic predictor*.

[9]Note that $M_X^{(Y)}$ and $M_Y^{(Y)}$ can be significantly different from $M_X^{(X)}$ and $M_Y^{(X)}$.

[10]For the GSVR we have 3 hyperparameters. Having both the coarse and fine grids with 5 values per hyper-parameter (this is a relatively moderate value), leads to about 250 executions

TABLE III
PREDICTOR USAGE. RELATIVE FREQUENCY OF EFFECTIVELY USING EACH OF THE PREDICTORS FOR DIFFERENT BLOCK TYPES

| Data-set | Block Type | Simple Linear | Linear Interactions | Pure Quadratic | LSVR | GSVR | $k$-NN |
|---|---|---|---|---|---|---|---|
| Study_033 | Rest | 0.368 | 0.125 | 0.189 | 0.021 | 0.297 | 0 |
| | pre-ictal | 0.349 | 0.112 | 0.180 | 0.006 | 0.353 | 0 |
| | SOZ | 0.226 | 0.142 | 0.131 | 0 | 0.501 | 0 |
| Study_037 | Rest | 0.255 | 0.166 | 0.222 | 0.025 | 0.333 | 0 |
| | pre-ictal | 0.231 | 0.142 | 0.153 | 0.015 | 0.459 | 0 |
| | SOZ | 0.269 | 0.197 | 0.139 | 0 | 0.395 | 0 |
| HUP70_phaseII | Rest | 0.533 | 0.075 | 0.195 | 0 | 0.197 | 0 |
| | pre-ictal | 0.298 | 0.228 | 0.166 | 0 | 0.308 | 0 |
| | SOZ | 0.237 | 0.261 | 0.181 | 0 | 0.321 | 0 |

The fact that the simple linear predictor is ranked first for rest blocks can be explained by noting that the *sampled signal changes much slower than the sampling rate*, thus, *fitting a simple linear model can be done accurately*. Moreover, compared to the other considered linear predictors, the simple linear predictor has the lowest number of parameters and therefore the highest number of samples per fitted parameter.

The fact that the GSVR is ranked first for pre-ictal and SOZ blocks can be explained by the fact that *GSVR represents the largest class of models* compared to other considered predictors. While this may hint that GSVR should achieve the lowest loss also for rest blocks, we recall that the performance of the GSVR depends on the selected hyperparameters. We conjecture that since the selection procedure is constrained by computational considerations, in some cases the selected hyperparameters are sub-optimal, leading to an increase in the achieved loss.

Finally, it can be observed that while it is seldom the case that the linear interactions predictor is selected, the *LSVR and the $k$-NN are almost never selected*. Regarding the LSVR, it suffers from the same challenge as GSVR in terms of selecting the hyperparameters. However, the class of models it represents is similar to the simple linear predictor. Thus, the performance of the LSVR is inferior to the performance of the simple linear predictor.

Regarding the $k$-NN, note that the results reported in Table III are different from those discussed in Sections III-A–III-C, where the $k$-NN was selected most of the time. We conjecture that this poor performance of the $k$-NN reflected in Table III is partially due to the fact that $k$-NN applies a local prediction method (predicts based only on the local neighborhood), hence it is not a global predictor as the other candidates are.[11] In addition, the $K$-NN duffers from the curse of dimensionality (for large values of $m$, significantly more samples are required for the $k$-NN to work well) as stated in the introduction.

## E. Comparing the Estimations of Different Predictors

The results detailed in Table III raise the following question: Do the considered predictors estimate a significantly different model order? To answer this question we note that, as indicated above, the estimations of the $k$-NN predictor are fundamentally different than the estimations of the other predictors as in all cases $k$-NN selects $m = 5$ and the $\ell_2$ loss increases with $m$. This is clearly different than the results reported in Figs. 6–11. Fig. 12 depicts the estimated ECDFs for data-set Study_033, for rest and pre-ictal blocks, and when *only the simple linear* predictor is used or *only the GSVR* predictor is used. It can be observed, as expected, that the GSVR results in smaller estimated values compared to the simple linear predictor. As a reference Fig. 12 also includes the ECDFs using the proposed algorithm.

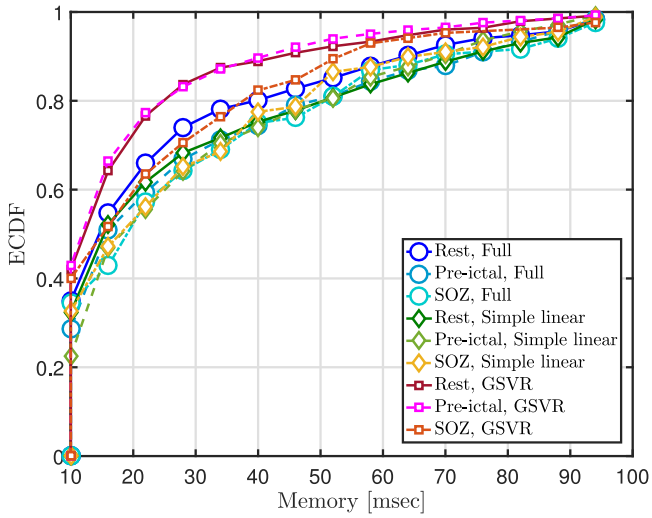


Fig. 12. ECDF of the estimated memory orders of different predictors, in milliseconds, for data-set Study_033. "Full" corresponds to the proposed algorithm.

## F. Extending the Predictors Ensemble

The framework proposed in the current paper can be extended to include more predictors as part of the prediction process, at the cost of increasing computational complexity. For instance, one can consider using a support vector regression with a polynomial kernel [41], a weighted $k$-NN, or a (recurrent) neural network. While neural networks were shown to be efficient in time-series prediction, see for example [58]–[60], in some cases they require a higher number of samples compared to the other methods, and their computational complexity is significantly

[11] When the signal of interest changes fast as in Sections III-A–III-C it is more complicated to fit a global model and local prediction may perform better.

higher. Therefore, in this paper we did not apply prediction using neural networks, and we leave this extension to future research.

## VI. Conclusion

We have developed a novel non-parametric algorithm for estimating the memory order of ECoG recordings. As the true memory order of ECoG recordings is not known, we first validated the proposed algorithm over noisy and non-linear simulated scenarios (time-series). These results indicated that the memory orders based on our prediction method matched the memory orders used to generate the sequences. The algorithm was then used to analyze ECoG recordings taken from epileptic patients listed in the iEEG portal for epilepsy research. The memory order of both rest and pre-ictal blocks for these patients was estimated, where for pre-ictal blocks the recordings taken from the SOZ were separately analyzed. Our analysis showed that the estimated ECDFs can significantly differ between patients (data-sets), between recording electrodes, and between different types of blocks. The results further showed that for all analyzed patients, the estimated memory orders are on the order of *tens of milliseconds* (the strict majority of the estimated values are smaller than 100 milliseconds).

The proposed algorithm is significant as it develops a foundation for more accurate estimation of model-free causal interactions between time-series, particularly for ECoG data where, to the best of our knowledge, there are no commonly accepted statistical models for these recordings. As part of future work it is desirable to improve the computational efficiency of the considered predictors. This is important as the number of analyzed blocks per patient is high, which leads to high computational complexity and commensurate delays in the signal analysis. Better implementation of the considered predictors will facilitate analyzing a larger number of patients over a shorter time period. Another research direction is to extend the class of predictors in order to find the optimal one, for example, by including neural networks for time-series prediction, and examining the impact on the resulting ECDFs.

## Acknowledgment

The authors would like to thank the reviewers for their comments, which greatly improved the results of the manuscript.

## References

[1] E. Bullmore and O. Sporns, "Complex brain networks: graph theoretical analysis of structural and functional systems," *Nature Rev. Neurosci.*, vol. 10, pp. 1–13, 2009.

[2] M. A. Kramer *et al.*, "Coalescence and fragmentation of cortical networks during focal seizures," *J. Neurosci.*, vol. 30, no. 30, pp. 10076–10085, 2010.

[3] P. van Mierlo *et al.*, "Functional brain connectivity from EEG in epilepsy: Seizure prediction and epileptogenic focus localization," *Pro. Neurobiol.*, vol. 121, pp. 19–35, 2014.

[4] A. N. Khambhati *et al.*, "Dynamic network drivers of seizure generation, propagation and termination in human neocortical epilepsy," *PLOS Comput. Biol.*, vol. 11, no. 12, pp. 1–19, 2015.

[5] S. P. Burns *et al.*, "Network dynamics of the brain and influence of the epileptic seizure onset zone," *Proc. Nat. Acad. Sci.*, vol. 111, no. 49, pp. E5321–E5330, 2014.

[6] J. Hlinkaa *et al.*, "Functional connectivity in resting-state FMRI: Is linear correlation sufficient?" *Neuroimage*, vol. 54, no. 3, pp. 2218–2225, 2011.

[7] K. J. Friston *et al.*, "Dynamic causal modelling," *Neuroimage*, vol. 19, pp. 1273–1302, 2003.

[8] T. Zhang *et al.*, "A dynamic directional model for effective brain connectivity using electrocorticographic (ECoG) time series," *J. Amer. Statist. Assoc.*, vol. 110, no. 509, pp. 93–106, 2015.

[9] J. J. Wright *et al.*, "Autoregression models of eeg," *Biol. Cybern.*, vol. 62, no. 3, pp. 201–210, 1990.

[10] R. Malladi *et al.*, "Identifying seizure onset zone from the causal connectivity inferred using directed information," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 7, pp. 1267–1283, Oct. 2016.

[11] Y. Murin *et al.*, "Sozrank: A new approach for localizing the epileptic seizure onset zone," *PLOS Comput. Biol.*, vol. 14, no. 1, pp. 1–26, 2018.

[12] Z. Cai *et al.*, "Inferring neuronal network functional connectivity with directed information," *J. Neurophysiol.*, vol. 118, pp. 1055–1069, 2017.

[13] P. Shenoy *et al.*, "Generalized features for electrocorticographic BCIs," *IEEE Trans. Biomed. Eng.*, vol. 55, no. 1, pp. 273–280, Jan. 2008.

[14] M. M. Shanechi *et al.*, "Robust brain-machine interface design using optimal feedback control modeling and adaptive point process filtering," *PLOS Comput. Biol.*, vol. 12, no. 4, pp. 1–29, 2016.

[15] Y. Yang and M. M. Shanechi, "An adaptive and generalizable closed-loop system for control of medically induced coma and other states of anesthesia," *J. Neural Eng.*, vol. 13, no. 6, Nov. 2016, Art. no. 066019.

[16] D. Lederman and J. Tabrikian, "Classification of multichannel eeg patterns using parallel hidden markov models," *Med. Biol. Eng. Comput.*, vol. 50, no. 4, pp. 319–328, Apr. 2012.

[17] T. Wissel *et al.*, "Hidden Markov model and support vector machine based decoding of finger movements using electrocorticography," *J. Neural Eng.*, vol. 10, no. 5, Oct. 2013, Art. no. 056020.

[18] C. Erlwein, "Applications of hidden Markov models in financial modelling," Ph.D. dissertation, Dept. Math. Sci., School Inf. Syst. Comput. Math., Brunel Univ., London, U.K., 2008.

[19] J. G. Dias *et al.*, "Mixture hidden markov models in finance research," in *Proc. Adv. Data Anal., Data Handling Bus. Intell.*, Jul. 2009, pp. 451–459.

[20] T. A. B. Snijders, "The statistical evaluation of social network dynamics," *Sociol. Methodol.*, vol. 31, no. 1, pp. 361–395, 2001.

[21] C. Heaukulani and Z. Ghahramani, "Dynamic probabilistic models for latent feature propagation in social networks," in *Proc. 30th Int. Conf. Mach. Learn.*, Atlanta, GA, USA, 2013, pp. 275–283.

[22] J. Jiao *et al.*, "Universal estimation of directed information," *IEEE Trans. Inf. Theory*, vol. 59, no. 10, pp. 6220–6242, Oct. 2013.

[23] R. Vicente *et al.*, "Transfer entropy—a model-free measure of connectivity for the neurosciences," *J. Comput. Neurosci.*, vol. 30, no. 1, pp. 45–67, 2011.

[24] Y. Murin and A. Goldsmith, "Using Markov properties of ECoG signals to infer neuron connectivity," in *Proc. Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, USA, Nov. 2018.

[25] Z. Goldfeld *et al.*, "Estimating differential entropy under Gaussian convolutions," 2018, arXiv:1810.11589.

[26] J. Jiao *et al.*, "Minimax estimation of functionals of discrete distributions," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2835–2885, May 2015.

[27] Y. Murin, "$k$-nn estimation of directed information," Dept. Elect. Eng., Stanford Univ., Stanford, CA, USA, Tech. Rep., 2017. [Online]. Available: https://arxiv.org/abs/1711.08516

[28] T. Hastie *et al.*, *The Elements of Statistical Learning*, 2nd ed. New York, NY, USA: Springer-Verlag, 2009.

[29] J. Wagenaar *et al.*, "A multimodal platform for cloud-based collaborative research," in *Proc. IEEE/EMBS Conf. Neural Eng.*, San Diego, CA, USA, Nov. 2013, pp. 1386–1389.

[30] A. D. R. McQuarrie and C. L. Tsai, *Regression and Time Series Model Selection*, 1st ed. Singapore: World Scientific, 1998.

[31] M. Wibral *et al.*, "Measuring information-transfer delays," *PLOS One*, vol. 8, no. 2, pp. 1–19, 2013.

[32] F. Takens, "Detecting strange attractors in turbulence," *Lecture Notes Math.*, vol. 898, pp. 366–381, 1981.

[33] M. Ragwtiz and H. Kantz, "Markov models from data by simple nonlinear time series predictors in delay embedding spaces," *Phys. Rev. E*, vol. 65, no. 5, pp. 1–12, 2002.

[34] Q. Meng and Y. Peng, "A new local linear prediction model for chaotic time series," *Phys. Lett. A*, vol. 370, pp. 465–470, 2007.

[35] P. Zhao *et al.*, "Chaotic time series prediction: From one to another," *Phys. Lett. A*, vol. 373, pp. 2174–2177, 2009.

[36] R. E. Bellman, *Adaptive Control Processes*. Princeton, NJ, USA: Princeton Univ. Press, 1961.

[37] J. Martinerie *et al.*, "Mutual information, strange attractors, and the optimal estimation of dimension," *Phys. Rev. A*, vol. 45, no. 10, pp. 7058–7064, 1992.
[38] D. Kugiumtzis, "Direct-coupling information measure from nonuniform embedding," *Phys. Rev. E*, vol. 87, 2013, Art. no. 062918.
[39] J. Runge *et al.*, "Optimal model-free prediction from multivariate time series," *Phys. Rev. E*, vol. 91, 2015, Art. no. 052909.
[40] A. Kraskov *et al.*, "Estimating mutual information," *Phys. Rev. E*, vol. 69, no. 6, 2004, Art. no. 066138.
[41] T. M. Huang *et al.*, *Kernel Based Algorithms for Mining Huge Data Sets*, 2nd ed. New York, NY, USA: Springer-Verlag, 2006.
[42] L. Cao, "Practical method for determining the minimum embedding dimension of a scalar time series," *Physica D*, vol. 110, pp. 43–50, 1997.
[43] M. Hénon, "A two-dimensional mapping with a strange attractor," *Commun. Math. Phys.*, vol. 50, no. 1, pp. 69–77, 1976.
[44] S. Sabesan *et al.*, "Information flow and application to epileptogenic focus localization from intracranial EEG," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 17, no. 3, pp. 244–253, Jun. 2009.
[45] Y. Yang *et al.*, "Dynamic tracking of non-stationarity in human ECoG activity," in *Proc. 39th Int. Conf. IEEE Eng. Med. Biol. Soc.*, Seogwipo, South Korea, 2017, pp. 1660–1663.
[46] N. Soltani, "*Inferring signaling structures in the brain via directed information*," Ph.D. dissertation, Dept. Elect. Eng., Stanford Univ., Stanford, CA, USA, 2015.
[47] M. Zijlmans *et al.*, "High-frequency oscillations as a new biomarker in epilepsy," *Ann. Neurol.*, vol. 71, no. 2, pp. 169–178, 2012.
[48] W. J. Zweiphenning *et al.*, "High frequency oscillations and high frequency functional network characteristics in the intraoperative electrocorticogram in epilepsy," *Neuroimage, Clin.*, vol. 12, pp. 928–939, 2016.
[49] S. V. Gliske *et al.*, "Effect of sampling rate and filter settings on high frequency oscillation detections," *Clin. Neurophysiol.*, vol. 127, pp. 3042–3050, 2016.
[50] C. Diks and J. DeGoede, *"A general nonparametric bootstrap test for Granger causality,"* in *Global Analysis of Dynamical Systems* Inst. of Phys., 2001.
[51] V. A. Vakorin *et al.*, "Exploring transient transfer entropy based on a group-wise ICA decomposition of eeg data," *Neuroimage*, vol. 49, no. 2, pp. 1593–1600, 2010.
[52] A. Bühlmann and G. Deco, "Optimal information transfer in the cortex through synchronization," *PLOS Comput. Biol.*, vol. 6, no. 9, 2010, Art. no. e1000934.
[53] J. T. Lizier *et al.*, "Multivariate informationtheoretic measures reveal directed information structure and task relevant changes in FMRI connectivity," *J. Comput. Neurosci.*, vol. 30, no. 1, pp. 85–107, 2011.
[54] F. De Vico Fallani *et al.*, "Graph analysis of functional brain networks: Practical issues in translational neuroscience," *Philosoph. Trans. Royal Soc. London B, Biol. Sci.*, vol. 369, no. 1653, 2014, Art. no. 20130521.
[55] D. Chicharro and S. Panzeri, "Algorithms of causal inference for the analysis of effective connectivity among brain regions," *Frontiers Neuroinform.*, vol. 8, pp. 1–17, 2014.
[56] A. Thul *et al.*, "Eeg entropy measures indicate decrease of cortical information processing in disorders of consciousness," *Clin. Neurophysiol.*, vol. 127, no. 2, pp. 1419–1427, 2016.
[57] S. Dimitriadis *et al.*, "Revealing cross-frequency causal interactions during a mental arithmetic task through symbolic transfer entropy: A novel vector-quantization approach," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 24, no. 10, pp. 1017–1028, Oct. 2016.
[58] Y. Ouyang and H. Yin, "Time series prediction with a non-causal neural network," in *Proc. IEEE Conf. Comput. Intell. Financial. Eng. Econ.*, London, U.K., 2014, pp. 25–31.
[59] D. Reid *et al.*, "Financial time series prediction using spiking neural networks," *PLOS One*, vol. 9, no. 8, pp. 1–13, 2014.
[60] T. Guo *et al.*, "Robust online time series prediction with recurrent neural networks," in *Proc. IEEE Int. Conf. Data Sci. Adv. Analytics*, Montreal, QC, Canada, 2016, pp. 816–825.