MACHINE LEARNING IN NATURAL COMPLEX SYSTEMS

EDITED BY: Andre Gruning, Thomas Hartung, Axel Hutt, Raina Robeva and
Alex Hansen

PUBLISHED IN: Frontiers in Applied Mathematics and Statistics,
Frontiers in Medicine, Frontiers in Artificial Intelligence,
Frontiers in Physics, Frontiers in Earth Science and
Frontiers in Systems Neuroscience

frontiers Research Topics

ISSN 1664-8714
ISBN 978-2-88976-369-6
DOI 10.3389/978-2-88976-369-6

About Frontiers

Frontiers is more than just an open-access publisher of scholarly articles: it is a pioneering approach to the world of academia, radically improving the way scholarly research is managed. The grand vision of Frontiers is a world where all people have an equal opportunity to seek, share and generate knowledge. Frontiers provides immediate and permanent online open access to all its publications, but this alone is not enough to realize our grand goals.

Frontiers Journal Series

The Frontiers Journal Series is a multi-tier and interdisciplinary set of open-access, online journals, promising a paradigm shift from the current review, selection and dissemination processes in academic publishing. All Frontiers journals are driven by researchers for researchers; therefore, they constitute a service to the scholarly community. At the same time, the Frontiers Journal Series operates on a revolutionary invention, the tiered publishing system, initially addressing specific communities of scholars, and gradually climbing up to broader public understanding, thus serving the interests of the lay society, too.

Dedication to Quality

Each Frontiers article is a landmark of the highest quality, thanks to genuinely collaborative interactions between authors and review editors, who include some of the world's best academicians. Research must be certified by peers before entering a stream of knowledge that may eventually reach the public - and shape society; therefore, Frontiers only applies the most rigorous and unbiased reviews.
Frontiers revolutionizes research publishing by freely delivering the most outstanding research, evaluated with no bias from both the academic and social point of view. By applying the most advanced information technologies, Frontiers is catapulting scholarly publishing into a new generation.

What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the Frontiers Journals Series: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area! Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers Editorial Office: frontiersin.org/about/contact

MACHINE LEARNING IN NATURAL COMPLEX SYSTEMS

Topic Editors:
Andre Gruning, University of Applied Sciences Stralsund, Germany
Thomas Hartung, Johns Hopkins University, United States
Axel Hutt, Inria Nancy - Grand-Est research centre, France
Raina Robeva, Randolph–Macon College, United States
Alex Hansen, Norwegian University of Science and Technology, Norway

Citation: Gruning, A., Hartung, T., Hutt, A., Robeva, R., Hansen, A., eds. (2023). Machine Learning in Natural Complex Systems. Lausanne: Frontiers Media SA. doi: 10.3389/978-2-88976-369-6

Table of Contents

frontiers in Applied Mathematics and Statistics

EDITORIAL
published: 24 March 2022
doi: 10.3389/fams.2022.869999

Editorial: Machine Learning in Natural Complex Systems

Axel Hutt [1]*, André Grüning [2], Alex Hansen [3], Thomas Hartung [4] and Raina Robeva [5]

[1] Université de Strasbourg, CNRS, INRIA, ICube, MLMS, MIMESIS, Strasbourg, France, [2] Mathematics and Computational, Intelligence Faculty of Electrical Engineering and Computer Science, Stralsund University of Applied Sciences, Stralsund, Germany, [3] Department of Physics, Faculty of Natural Sciences, Norwegian University of Science and Technology, Trondheim, Norway, [4] Bloomberg School of Public Health, Johns Hopkins University, Baltimore, MD, United States, [5] Department of Mathematics, Randolph-Macon College, Ashland, VA, United States

Keywords: model improvement, learning rules, deep learning, data assimilation, big data

Editorial on the Research Topic

Machine Learning in Natural Complex Systems

For many decades, scientists strive to develop intelligent machines to reproduce or improve human intelligent actions. Machine Learning (ML), a research branch of artificial intelligence, aims to learn and simulate specific intelligent human actions and has been applied to a large variety of artificial and natural systems. At first glance, it is tempting to automize solving complex problems by machines. Specifically, in ML this is done by learning features of a data training data set, i.e., learning hidden relations in the data. Since the learned patterns and the subsequent prediction is based on the training set, the success of ML heavily depends on the training data. This renders the application of ML to natural systems rather challenging, since natural systems' dynamics are complex, and the training data set has to reflect the diverse dynamics. Typically, this renders the data set huge in size and thus such methods apply well to so-called *Big Data*. Examples for applications are visual pattern recognition, language processing and signal processing. Moreover, ML techniques compete with model approaches, which have built-in relations between system elements and which do not depend on specific training sets. In certain research fields, such as meteorology, system models and corresponding techniques are already so powerful that a benefit of ML is still unclear. However, recent techniques combine ML and models to achieve the best insight into the dynamics of systems. The present Research Topic collects work from a large variety of research fields on natural complex systems and provides a good up-to-date overview of the field of ML.

An important percentage of ML applications focuses on classification of datasets. In medical science, a combination of different data types has proven to improve patient classification and prediction. In Zhu et al., the authors have shown that a deep learning classification of both patient and radiograph variables allow to predict osteonecrosis and thus may prevent its aggravation at an early stage. Similarly, Xue et al. shows that the combination of diverse patient variables, such as level of cholesterol and patient age, permits to predict pulmonary complications after gastrointestinal surgery by ML techniques. The latter studies demonstrate that ML techniques permit predictions in patients with a very high success rate. In line with this approach, Liu G. et al. reviews ML techniques for epilepsy detection in electroencephalographic signals (EEG). Such latter techniques involve the additional complexity of a temporal sequence in the data compared to the temporally static data in the studies mentioned earlier. The authors of Liu J. et al. also consider EEG signals and demonstrate how to classify emotions by ML techniques. A different application is considered in Peng et al. considering observed time series in a meteorological model. Here, the authors successfully find relationships between the climate variables using ML on the basis of precipitation observations.

OPEN ACCESS

Edited and reviewed by:
Ulrich Parlitz,
Max Planck Society, Germany

***Correspondence:**
Axel Hutt
axel.hutt@inria.fr

Specialty section:
This article was submitted to Dynamical Systems, a section of the journal Frontiers in Applied Mathematics and Statistics

Received: 05 February 2022
Accepted: 21 February 2022
Published: 24 March 2022

Citation:
Hutt A, Grüning A, Hansen A, Hartung T and Robeva R (2022) Editorial: Machine Learning in Natural Complex Systems. Front. Appl. Math. Stat. 8:869999. doi: 10.3389/fams.2022.869999

Although classification tasks play an important role in current ML research, existing ML techniques may also serve as tools to achieve a different research goal. For instance, Wang et al. shows how ML techniques may assist to learn time-dependent neural activity in order to understand better the relation of neural activity to behavior. ML may also help to understand complex systems on the basis of incomplete observations. Herzog et al. demonstrate how to complete sparsely sampled mechanical cardiac dynamics to predict electric heart activity by employing a dynamical model. The combination of ML techniques and mathematical models, as in this latter work, is especially powerful and promises to boost theoretical models of complex systems. This is shown nicely as well in Kigure, where the author demonstrates a numerical speed-up in a computational fluid dynamics model by employing ML.

In recent years, experimental observation techniques have been improved and are providing more and more data with increased temporal and spatial resolution. This progress demands advanced analysis techniques. For instance, today it is possible to observe neural activity in brain tissue by electrode grids with a high spatial resolution. Since the corresponding observed time series are correlated by virtue of the increased resolution, Geddes et al. propose a new data analysis technique for multivariate signals to extract novel underlying dynamic features. However, an improved temporal resolution of observations also requests improved analysis techniques that take into account multiple time scales. To this end, Manneschi et al. propose an extended reservoir computing technique to better learn computationally observed time series. Moreover, new observational data may also exhibit sparse dynamics that represents an important challenge to existing ML techniques. Nascimento et al. investigates how various sparse dynamics models can be combined with a graphical approach on the basis of EEG data to retrieve valuable insights into the brain's network connectivity. These latter data evolve on a certain grid in space and/or time and may be used to identify hidden patterns of functional and effective connectivity of brain networks. However, this is more difficult in other data, such as the temporally-changing positions of single objects in an environment. The authors Fromreide and Hansen address such data and present a method to predict the motion of moving ships. They demonstrate how to introduce movement patterns dependent on the land/sea environment.

In data analysis, a first step identifies data properties which are then examined in subsequent steps. In order to extract knowledge from a dataset and learn more about the system under study, it may be beneficial to look for structural patterns in the data. For instance, certain symmetries in datasets may immediately indicate pieces of redundant information and permit a simplifying dimensionality reduction. Capobianco addresses this issue and argues that data symmetries may enhance ML performance. Moreover, it may be promising to define novel similarity measures in life science data which are adapted to the respective data type (e.g., genetic sequences, chromosome data or chemical structure formulas). In Münch et al., the authors present strategies and concepts how to employ data-driven similarity measures in ML.

In conclusion, the Research Topic *Machine Learning in Natural Complex Systems* showcases the powerful impact of ML on the study of complex systems and the enormous potential this approach holds for gaining further knowledge about the complexities of their dynamics. We trust that the readers will enjoy these articles as much as we did and hope that the collection will help generate further discussion and inspire further discoveries.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers
in Applied Mathematics and Statistics

PERSPECTIVE
published: 09 July 2020
doi: 10.3389/fams.2020.00023

Inference From Complex Networks: Role of Symmetry and Applicability to Images

*Enrico Capobianco**

Institute of Data Science and Computing, University of Miami, Coral Gables, FL, United States

Symmetry is a mathematical concept only partially explored in networks, especially at the applicative level. One reason is a certain lack of interpretable inference obtained from networks. While the network systemic associations (links) between entities (nodes) emerge from the underlying dependence structure, this latter is only partially explicit via the established direct interactors and remains to a certain extent latent (distant node predicted paths). Verifiability of significant hubs, connectors, paths, and modules allows to build a knowledge base useful to infer latencies and/or validate complex associations. When symmetry is searched in images, reflection, translation and rotation are applicable transformations in *n*-dimensional Euclidean space that computational algorithms target. There is symmetry when original and transformed images cannot be distinguished. Once collected together, such transformations form an automorphism group, indicating a stable and robust global characteristic. It is common to step from images to quantifiable features for conducting inference. Deep learning is typically used to classify whole images reconstructed from the myriads of features in which these images are decomposed. However, with images considered at multiple scales and locations, symmetries are valuable for describing local characteristics. Casting local features into a network framework enables their associations to be explored by similarity or dissimilarity criteria. This is quite intriguing because network configurations may display topological features and connectivity patterns associated with synchronization and symmetry that reduce the redundancy of features to more compact functional descriptions. Then, identifying anomalies from unusual events, behaviors, patterns would spot network vulnerabilities and signs of symmetry breaking.

Keywords: networks, entropy, symmetry, controllability, synchronization, deep learning

OPEN ACCESS

Edited by:
Raina Robeva,
Randolph–Macon College,
United States

Reviewed by:
Axel Hutt,
Inria Nancy—Grand-Est Research Centre, France
Miguel Pineda,
University College London,
United Kingdom

***Correspondence:**
Enrico Capobianco
ecapobianco@med.miami.edu

Specialty section:
This article was submitted to Dynamical Systems, a section of the journal Frontiers in Applied Mathematics and Statistics

Received: *18 April 2020*
Accepted: *25 May 2020*
Published: *09 July 2020*

Citation:
Capobianco E (2020) Inference From Complex Networks: Role of Symmetry and Applicability to Images. Front. Appl. Math. Stat. 6:23. doi: 10.3389/fams.2020.00023

INTRODUCTORY REMARKS AND METHODOLOGICAL BACKGROUND

In physical systems, symmetry commonly means invariance. Therefore, this is like to say that a system looks the same from different observation angles. Equivalently, due to the presence of intrinsic regularities, a system's characteristic may be an observable output in correspondence with multiple interrelated inputs. As symmetries influence the system's functionality and models allow the interpretation of the function dynamics, a general concern is model misspecification. This implies that an inference model m applied to sample data $x\epsilon X$ may fail due to the wrong choice of model family $M(x,\theta)$ characterized by parameters $\theta\epsilon\Theta$ and likelihood function $L(\theta|x)$. For example, the identifiability of parameters may be violated. Among other possible reasons, parameter non-identifiability may occur due to the presence of symmetry [1].

Formally, symmetry indicates a type of invariance such that a mathematical object remains unchanged under a set of transformations. For instance, a symmetry of a function is a transformation of the function that leaves its graph unchanged. More in general, a symmetry maps an object onto itself, thus preserving its structure. In mathematical terms, a symmetry $s\colon \theta \rightarrow \theta$ is a measurable function that makes θ in a model no longer identifiable from the given data x. The natural way to obviate to this effect of symmetry is to re-parameterize the model, i.e., $\theta \rightarrow \psi$, with $\psi\epsilon\Psi$. Alternatively, one can constrain the parameters, i.e., $\theta \rightarrow \theta^c$. In such cases, the recourse to "symmetry breaking" solutions can be needed to augment the performance of inferential algorithms. Symmetry breaking implies that small system's fluctuations vary beyond established thresholds and that the system may cross a critical point due to the variation governing its state dynamics. Overall, this determines the system's fate because once such transition occurs, it may orient the system toward a different state compared to the initial one.

There are several types of symmetries, the most common being permutations. Some symmetries are local and grouping them leads to transformations that change the system in different ways at different places in space and time. Instead, a symmetry is global if it acts similarly at every point. The fact that for a global symmetry the corresponding parameter is constant explains why the transformation is applied uniformly across the entire system, unlike with a local symmetry in which its parameter is a function of position, and the transformation is applied differently at different points in space and time.

In the system of interest here, i.e., a network N, symmetry depends on the preferential attachment growth rule inducing a bias toward multiple short branches that tend to repeat themselves in a tree more than longer ones, and the definition of symmetry is inherent to nodes and their permutations [2]. Given the N adjacency matrix A, in which $A_{ij} = 1$ (nodes i,j are adjacent) and 0 (otherwise), a symmetry is present when a permutation P is applied to A, leaving it unchanged, i.e., $PA = A$. This so-called automorphism (*Aut*) means that nodes are topologically equivalent if their permutation does not affect the network structure. This offers a rationale for considering network redundancy and its reduction into network quotients or skeletons [3–5]. Here, structural network properties, including heterogeneity and complexity, remain while repetitions are excluded.

Symmetries may also be combined, but of interest are especially the minimal ones calling all participating nodes to form a symmetric structure, or an orbit O (in which a set of nodes is mapped onto each other). Formally, given $I(N)$ as a set of nodes in N forming a group $Aut(N)$, the orbit of the node $i\epsilon I(N)$ is a set $O(i) = \{\pi \bullet i\epsilon I(N)\colon \pi\epsilon Aut(N)\}$ (see [4]). Thus, nodes belonging to an orbit are intrinsically related by simple permutations in $Aut(N)$. A symmetric network partitioning of its nodes into orbits is thus establishing disjoint equivalence classes for each node, which forms an automorphism partition (*Aut-P*) [6]. Of interest is also its entropy, which is a measure of the network structural heterogeneity defined as:

$$E^{Aut-P} = -\sum\nolimits_k p_k logp_k \tag{1}$$

with $k = 1,K$, for $K = dim(Aut\text{-}P)$, and with $\mathrm{p_k}$ the associated probabilities (computed as node ratios between *Aut-P* and network). The normalized entropy is then defined as $E_N^{Aut-P} = E^{Aut-P}/log\,K$.

It is important to consider a system in steady state (equilibrium) vs. possible departure from it. Such dynamics are regulated by the spectral characteristics. In particular, the eigenvalues of a symmetric structure are decomposable into two types: redundant and non-redundant. Redundancy is in correspondence with the eigenvectors localized on the symmetric structure. The non-redundant eigenvectors refer to eigenvalues with the same values relatively to the nodes of a given orbit. This is relevant for the purpose of reducing the network to the consideration of orbits instead of nodes, which eliminates the redundancy. Conversely, this is a structural property of symmetric networks.

Networks are also a paradigm of complexity. If we insert a probability measure P^N in a network N, which defines a probability distribution $Pr(N)$ on its nodes, each having a probability pr_i, we can also define a Network Entropy as:

$$E^N = min\sum\nolimits_i pr_i logpr_i \tag{2}$$

This entropy measures the randomness degree in the network and implies the importance of stochasticity, with a role played by symmetries too. Namely, measures may vary depending on fuzzy symmetries [7] (based on fuzzy measure theory) and stochastic symmetries [8] (based on network ensembles). Furthermore, network communities (defined as cohesively connected sets of functionally similar nodes) exert strong inference impacts, although there are cases in which functional node similarity refers to symmetry rather than community effects. A typical example of such ambiguity is provided by studies of brain areas.

The role of symmetries is especially relevant with reference to network synchronization (see [9]), and computational methods to break such symmetries (isolated desynchronization) are discussed in [10]. The construction of functional networks depends on the relationships between their coupling components, which makes synchronization motifs central features. Functional networks are in general heterogeneous or non-symmetrical structures. This reflects the fact that disruption of the couplings generates symmetry-breaking in the network, and also loosens the inherent distribution of synchronization motifs. However, if the couplings are able to sustain synchronization, then symmetry will be characterizing the functional network [11].

Finally, controllability refers to the ability of a dynamical system to step between states, say from an initial to a final state, in finite time [12, 13]. This possibility can be verified quite straightforwardly in linear time-invariant systems by the so-called Kalman's rank condition. In particular, given a canonical system $dz = Az + Bv$, with state vector *z,* and *A*, and *B* as state and control parameter matrices, respectively, the requirement is: *rank[B,AB,…..,* A^{P-1} *B]* $= P$. Naturally enough, a complication is when the system's parameters are unknown. Furthermore, a common strategy to establish network controllability turns to the identification of a minimal set of driver nodes. This implies that controlling this set allows to exert control over the entire system [14].

With reference to symmetry, what reported in [15] is relevant for the focus on network controllability but considering the impact of individual rather than global dynamics, thus departing from network topology. In real-world networks, assumptions like the independence of parameters must be relaxed to face the presence of dependence and even interdependence structure. Thus, a global symmetry aspect is a characteristic to consider in assessing both single and combined networks, and establishing the highest possible controllability, i.e., the lowest number of critical nodes, becomes a primary goal in applications. Intuitively, specific hub or module dynamics may remain locally informative but become less relevant for controllability scopes compared with the densities at which they present at network scale, which reflects their possible influence. **Figure 1** reconciles the main concepts in this section.

Nevertheless, real networks complexity may be reduced by considering node-specific hidden variables that once transformed may reveal latent symmetries. Such transformations may extend to statistical ensembles designed through families of stochastic networks [16]. For the networks partition $N = [N_1, N_2, \ldots, N_Q]$ and the associated probabilities assigned to each of them *Pr(N)* $= [Pr(N_1), Pr(N_2), \ldots Pr(N_Q)]$, it holds that $\sum_N Pr(N) = \sum_q Pr(N_q) = 1$ for $q = 1,Q$. These networks will be stochastically symmetric under a transformation if each member network has the same properties under the transformation. An associated entropy optimization problem is the one searching for the *Pr(N)* that maximizes the Shannon-Gibbs entropy:

$$E^{SG} = -\sum\nolimits_N Pr(N) log Pr(N) \tag{3}$$

This problem usually involves a topologically constrained network, particularly when the ensemble network functions as a null model (more details on probability functions and entropies can be found in [8]).

COMPUTATIONAL ASPECTS OF NETWORK SYMMETRY

Symmetry exerts several types if influences (see **Figure 2**). A first main question generally addressed is how to quantify the redundancy that is due to the presence of symmetry in a network. One way is through the compression ratio [17], a measure that compares the full network N to its quotient Q (counting one node per orbit) computed in two possible ways as either:

$$C^1 r = (n_N / n_Q)^2 \tag{4}$$

where the ratio is between the number of nodes n_N and the number of orbits n_Q in the full network, or

$$C^2 r = l_N / l_Q \tag{5}$$

this time with edges l_N and l_Q (representing average connectivity) used in a sparse network.

It is between these two measures that the redundancy of an arbitrary network can fluctuate. A direct compression of symmetry is also possible, through the quotient matrix *Q(A),* obtained from the adjacency matrix *A* and the characteristic matrix *S* (i.e., referred to the network partition associated to the quotient):

$$Q(A) = S^T AS \tag{6}$$

Importantly, symmetry leaves also a spectral signature through the presence of peaks in the spectral density, to which redundant eigenvectors are associated, i.e.,

$$\rho(\lambda) = 1/N \sum\nolimits_{i=1,N} \delta(\lambda - \lambda_i) \tag{7}$$

or a sum of Dirac delta functions with $\lambda_i$as the largest eigenvalue [4]. When the spectrum of a network adjacency matrix is considered, symmetries differentiate according to whether a symmetric structure is present, for then the eigenvalues can be decomposed into two classes: a) with redundant eigenvalues, in correspondence to eigenvectors localized on the symmetric structure and b) with non-redundant eigenvalues, i.e., whose eigenvectors have identical values in all elements that correspond to nodes in the same orbit. As an alternative, similar results were obtained for the Jacobian matrix in a food web model [18].

Looking beyond partitions from nodes to orbits, one can consider equitable partitions in which two node clusters have a special relationship, one in which each node of one cluster has exactly the same number of neighbor nodes in the other cluster. Interestingly, this complementary partition has relevance for synchronous patterns, as in both types of partitions nodes in the same cluster can synchronize and in different clusters cannot [19].

It is consolidated nowadays as a practice the fact to extensively use *Deep Learning* (DL) in image applications. It is rarer to find symmetry at the core of studies, despite the relevance [20]. An interesting study [21] of generative adversarial networks (GAD) has exploited horizontal symmetry (usually not considered in such form) by adopting two methods, one checking what symmetry loss allows generated images and flipped versions to be classified the same way, and another that inverts the flipped images aiming at reconstructing with minimal distortion. Another convolutional neural

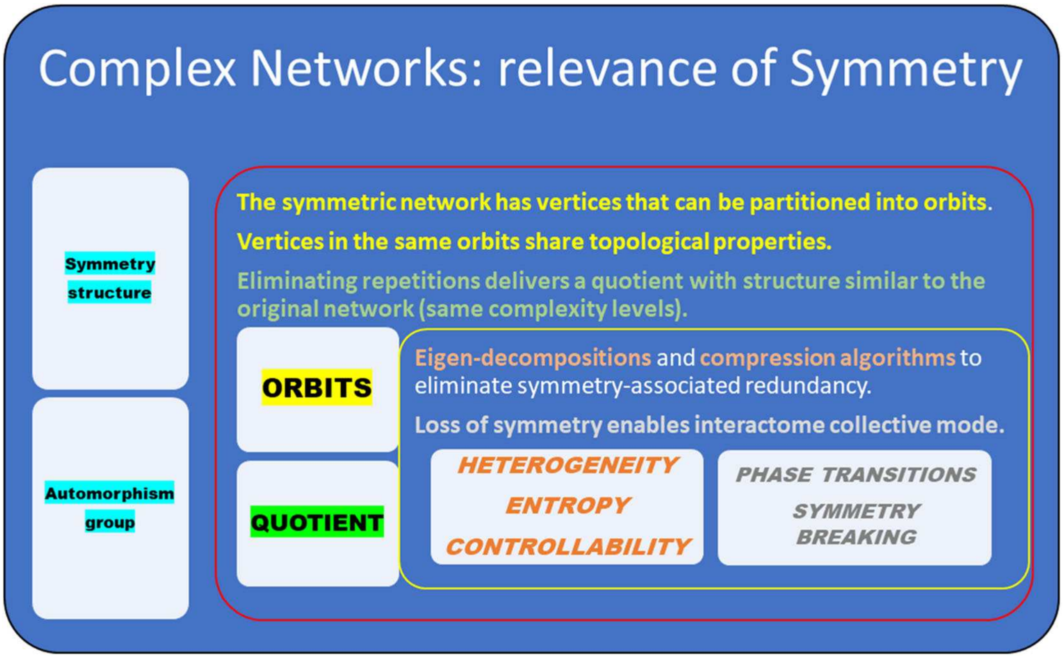

Complex Networks: relevance of Symmetry
Symmetry structure
Automorphism group
The symmetric network has vertices that can be partitioned into orbits.
Vertices in the same orbits share topological properties.
Eliminating repetitions delivers a quotient with structure similar to the original network (same complexity levels).
ORBITS
QUOTIENT
Eigen-decompositions and compression algorithms to eliminate symmetry-associated redundancy.
Loss of symmetry enables interactome collective mode.
HETEROGENEITY
ENTROPY
CONTROLLABILITY
PHASE TRANSITIONS
SYMMETRY BREAKING

FIGURE 1 | Hierarchy of symmetry relationships in networks.

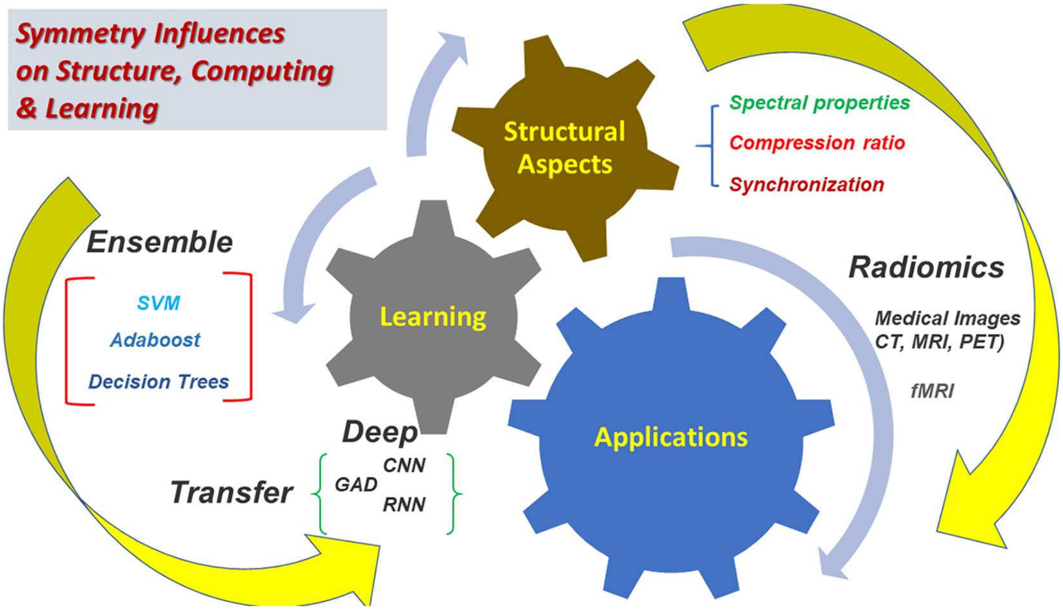


FIGURE 2 | Symmetry outreach.

neural networks (CNN) study [22] inspired by primary visual cortex processes has investigated the impact of symmetry constraints in convolutional layers for image classification. As a result, similar performance was found in a setting with reduced number of parameters due to replacement of random weights by symmetry constraints during backpropagation.

Interestingly, the imposition of symmetry constraints to reduce the number of parameters was assessed in another study [23] in which accuracy loss was absolutely limited even in overparameterized settings (both CNN and recurrent neural networks or RNN) from CIFAR, ImageNet etc. As a general remark, data paucity and non-standard statistical settings (non-independently identically distributed, unsupervised etc.)

have suggested to embed learning architectures with invariance under symmetry transformations. In a study [24] on probabilistic symmetry, a link with functional symmetry was established such that the structure of networks could be completely characterized by invariance usable to construct both stochastic and deterministic solutions.

Different applications have been presented by two further studies. One study [25] was centered on a spreading model in a setting of multilayer networks that considers non-equilibrium phase transition when either susceptible-infected-susceptible or susceptible-infected-recovered spreading dynamics being integrated. Using a biased diffusion process among different layers, the discontinuous transition goes together with a spontaneous symmetry breaking in occupation probabilities of individuals in each layer. Another interesting study [26] was proposed on the symmetry of weighted brain networks to decipher the roles of individual brain areas and the redundancy of connectivity. The structural symmetry of every pair of nodes in the network was quantified by the isomorphism of the residual graphs. Then, fMRI was performed on subjects with a condition, i.e., inattentive type of Attention Deficit Hyperactivity Disorder, showing the emergence of higher level of network symmetry compared to the development group. As a methodological note, symmetry levels threshold-sensitivity was observed.

IMAGE NETWORKS AND ROLE OF SYMMETRY

Radiomics [27–29] promotes the role of images as mineable data [30]. These are features providing the medical imaging community with a wealth of quantitative information usable for diagnosis, therapy evaluation, patient risk profiling and stratification. The literature of radiomics is dense of applications centered on pipelines that leverage machine learning (ML) implementation of feature selection and image fusion methods. Especially the DL community found in radiomics a precious source of data and a testing ground for ML and statistical methods. In turn, the advent of this discipline has stimulated increasingly integrative research on cancer biology, genomics and clinics with unprecedented scale and resolution depth [31]. Thus, it has become possible to investigate the complex hierarchical organization of medical images.

The key factor is designing DL and ML solutions targeted to a synthesis of various types of measurements and predictions, either assembled together in the attempt to concentrate information or distributed across network layers in order to reconcile all the differently channeled information into scores. These two types of approach, concentric one and distributed the other, may present quite different symmetry scenarios, despite a differentiation occurs according to the achievable precision, thus depending on imaging technologies and combination of modalities together with data gaps.

From a modeling standpoint, and by keeping symmetry in mind, *transfer learning* (TL) [32–36] and ensemble modeling (EM) seems especially relevant. The TL's paradigm is that features may be learned in a certain application domain and then applied to different domains. The role of symmetry could be an important criterion to judge generalizability first and feasibility after, for then leaving to the ability of TL leveraging other similarity aspects (related or not to symmetry). EM seems to match well the distributed approach discussed earlier by suggesting strategies for weighting scores and predictions from different models. However, data features gathered at a variety of spatiotemporal resolutions are hard to integrate into interpretable predictive models. Quite evidently, there is strong need to identify and measure heterogeneity in spatial and time, and then cross-correlate all types of imaging features.

Structural symmetry is central to human brain functions and thus the functional activities of areas that are symmetric at an anatomic level can be strongly correlated, even if the areas are distant in space. Experiments in [9] studied symmetry in the human brain via coupled anatomical (DW-MRI data) and functional (fMRI) connectivity graphs defined on the same set of 90 cortical areas. Interestingly from a network standpoint the fact the relevance of anatomical symmetry in neural synchronization because determining correlated functional modules across distant locations. At one end anatomically symmetric regions obtained via clustering of nodes with phase at stationary state, at the other end a functional network with links indicating statistically significant correlations between the timeseries of cortical areas. For two nodes in spatially separated regions showing no anatomical connection it appeared symmetry and strong synchronization from fMRI. For other two nodes from spatially adjacent regions and anatomically connected a gap in phase emerged together with lack of fMRI synchronized patterns.

In an interesting study [37], non-contrast (nc) CT images with follow-up magnetic resonance diffusion-weighted (MR-DW) images have supported evidence on the differentiation between ischemic and healthy brain tissue through quantification of symmetry. Stroke spots were identified by spatially aligning MR-DW images to the corresponding ncCT images. An interesting observation is that three methods (AdaBoost, Support Vector Machines, Decision Trees) were used to classify, but a basic limitation in terms of accuracy remained when comparing the textures only between the stroke lesions while considering as normal the contralateral regions. Despite the symmetrical acute strokes would be uncommon, other ischemic lesions or pathologies might be present in the contralateral region and require selection of features *ad hoc* for differentiating stroke regions from normal ones.

In the above example networks were not used, but in general our ability to carry or block information through a network depends on its topology. Especially when the conductivity or information transfer is considered then spatial symmetry becomes central together with its density or average number of links per node and topological dimension [38]. Therefore, an alternative method of classification ideally fitting the experiments in would involve encoding equivariance in learning [39]. In order to approximate invariance to a class of transformations of the input, a neural network would need training via with data augmentation (see [40], and related references in this work). With enough capacity, the network may learn such invariances but without guarantee of generalization.

The specific patterns of synchronization detectable in a network provide information about the underlying couplings dynamics and when considered in all their relationships can well characterize the system under study. Even with non-linear dynamics interaction regimes become identifiable and the network structure inferable from similarity and connectivity. Co-existing complex synchronized dynamics male a network stable and robust against external factors (stressors/perturbations) and heterogeneous motifs are probes for functional connectivity, particularly when stable as they become an invariant feature (i.e., consistency of synchronization patterns). Thus, the relationship between network symmetries and the consistency of the synchronization patterns, particularly with coupling inducing global synchronization and resulting in symmetrical networks reflected onto functional ones, as shown in the brain context (see [11] for all details and related references).

DISCUSSION

We keep a final point for discussion. Identifying anomalies from unusual events, behaviors, patterns can be useful to spot network vulnerabilities and may reveal presence of symmetry breaking. Conversely, the control of large dynamical complex networks may depend on the identification of just a few input nodes or modules. Therefore, by keeping in mind what the targets are, the results from exerting control through networks can be quite efficiently achievable, although only approximately.

The identification of symmetries in a complex network remains important in order to decipher its organizational principles and rules. It is key to understand the role of symmetries in reconstructing or controlling network dynamics [41]. It is important to decompose a network in two possible ways: one into observable/controllable vs. unobservable/uncontrollable sub-networks, and another one into symmetry-driven vs. non-symmetry-driven sub-networks. Finally, it is key to study how network components synchronize or desynchronize because network functionalization depends on such coupled dynamics. Naturally enough, the problems are always much harder in non-linear networks as observability and controllability must deal with more complex dependence relationships.

The last considerations go to the application domain. Symmetry has several impacts on very transformative fields like robotics, computer vision, computer graphics, medical image analysis, radiomics all characterized by coupling artificial intelligence and machine learning with geometry, group theory, graphs, statistics etc. Recognizing symmetries is instrumental to the retrieval of structure from redundant noisy systems, therefore a statistical problem too, which brings in lots of other applications and many possible model frameworks (see for instance examples from network medicine applied to cancer data of various complexities [31]).

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary materials, further inquiries can be directed to the corresponding author/s.

AUTHOR CONTRIBUTIONS

The author confirms being the sole contributor of this work and has approved it for publication.

FUNDING

Support for study & research activity in network science is acknowledged from NIH AWD-005216 GRANT GR013673 and NSF 19-500 GRANT; DMS 1918925/1922843 (years 2019-22).

ACKNOWLEDGMENTS

The author thanks colleagues and scholars for deep discussion on these topics at participated seminars and talks. The author thanks two reviewers for their comments on a first draft of the paper. Support for study & research activity in network science is acknowledged from NIH AWD-005216 GRANT GR013673 NSF-NIH grant DMS-1922843.

REFERENCES

1. Nishihara R, Minka T, Tarlow D. Detecting parameter symmetries in probabilistic models. *arXiv [Preprint] arXiv:1312.5386v1* (2013).
2. Sevim V, Rikvold PA. Effects of preference for attachment to low-degree nodes on the degree distributions of a growing directed network and a simple food-web model. *Phys Rev E Stat Nonlin Soft Matter Phys.* (2006) 73(5 Pt 2):056115. doi: 10.1103/PhysRevE.73.056115
3. Xiao Y, MacArthur BD, Wang H, Xiong M, Wang W. Network quotients: structural skeletons of complex systems. *Phys Rev E.* (2008) 78:046102. doi: 10.1103/PhysRevE.78.046102
4. MacArthur BD, Sanchez-Garcia RJ. Spectral characteristics of network redundancy. *Phys Rev E.* (2009) 80:026117. doi: 10.1103/PhysRevE.80.026117
5. Wang H, Yan G, Xiao Y. Symmetry in world trade network. *J Syst Sci Complex.* (2009) 22:280–90. doi: 10.1007/s11424-009-9163-9
6. Xiao YH, Wu WT, Wang H, Xiong M, Wang W. Symmetry-based structure entropy of complex networks. *Phys A.* (2008) 387:2611–19. doi: 10.1016/j.physa.2008.01.027
7. Garrido A. Symmetry in complex networks. *Symmetry.* (2011) 3:1–15. doi: 10.3390/sym3010001
8. Garlaschelli D, Ruzzenenti F, Basosi R. Complex networks and symmetry I: a review. *Symmetry.* (2010) 2:1683–709. doi: 10.3390/sym2031683
9. Nicosia V, Valencia M, Chavez M, Diaz-Guilera A, Latora V. Remote synchronization reveals network symmetries and functional modules. *Phys Rev Lett.* (2013) 110:174102. doi: 10.1103/PhysRevLett.110.174102
10. Pecora LM, Sorrentino F, Hagerstrom AM, Murphy TE, Roy R. Cluster Synchronization and isolated desynchronization in complex networks with symmetries. *Nat Commun.* (2014) 5:4079. doi: 10.1038/ncomms5079
11. Malagarriga D, Villa AEP, Garcia-Ojalvo J, Pons AJ. Consistency of heterogeneous synchronization patterns in complex weighted networks. *Chaos.* (2017) 27:031102. doi: 10.1063/1.4977972
12. Kalman RE. Mathematical description of linear dynamical systems. *J SIAM Control Ser A.* (1963) 1:152–92. doi: 10.1137/0301010
13. Luenberger DG. *Introduction to Dynamic Systems: Theory, Models and Applications*. New york, NY: John Wiley & Sons, Inc., (1979). p. 446.

14. Liu YY, Slotine JJ, Barabasi AL. Controllability of complex networks. *Nature*. (2011) 473:167–73. doi: 10.1038/nature10011
15. Zhao C, Wang WX, Liu YY, Slotine JJ. Intrinsic dynamics induce global symmetry in network controllability. *Sci Rep*. (2015) 5:8422. doi: 10.1038/srep08422
16. Marras E, Travaglione A, Capobianco E. Sub-modular resolution analysis by network mixture models. *Stat Appl Genet Mol Biol*. (2010) 9:19. doi: 10.2202/1544-6115.1523
17. Garcia RJS. Exploiting symmetry in network analysis. *Commun Phys*. (2020) 3:87.
18. Aufderheide H, Rudolf L, Gross T. Mesoscale symmetries explain dynamical equivalence of food webs. *New J Phys*. (2012) 14:105014. doi: 10.1088/1367-2630/14/10/105014
19. Siddique AB, Pecora L, Hart JD. Symmetry and input cluster synchronization in networks. *Phys Rev E*. (2018) 97:042217. doi: 10.1103/PhysRevE.97.042217
20. Griffin LD. Symmetries of 1-D Images. *J Math Imaging Vis*. (2008) 31:157–64. doi: 10.1007/s10851-008-0078-1
21. Makkapati V, Patro A. *Enhancing Symmetry in GAN Generated Fashion Images*. Cambridge: Springer International Publishing (2017). p. 405–10. doi: 10.1007/978-3-319-71078-5_34
22. Dzhezyan G, Cecotti H. SymNet: symmetrical filters in convolutional neural networks. *arXiv*. (2019) 187:102786.
23. Hu SX, Zagoruyko S, Komodakis N. Exploring weight symmetry in deep neural networks. *Comput Vis Image Und*. (2019) 187:102786. doi: 10.1016/j.cviu.2019.07.006
24. Bloem-Reddy B, Teh YW. Probabilistic symmetry and invariant neural networks. *arXiv [Preprint] arXiv:1901.06082v1* (2020).
25. An N, Chen H, Ma C, Zhang H. Spontaneous symmetry breaking and discontinuous phase transition for spreading dynamics in multiplex networks. *New J Phys*. (2018) 20:125006. doi: 10.1088/1367-2630/aaf660
26. Hu C, El Fakhri G, Li Q. Evaluating structural symmetry of weighted brain networks via graph matching. *Med Image Comput Comput Assist Interv*. (2014) 17(Pt 2):733–40. doi: 10.1007/978-3-319-10470-6_91
27. Lambin P, Rios-Velazquez E, Leijenaar R, Carvalho S, van Stiphout RG, Granton P, et al. Radiomics: extracting more information from medical images using advanced feature analysis. *Eur J Cancer*. (2012) 48:441–6. doi: 10.1016/j.ejca.2011.11.036
28. Parekh V, Jacobs MA. Radiomics: a new application from established techniques. *Expert Rev Precis Med Drug Dev*. (2016) 1:207–26. doi: 10.1080/23808993.2016.1164013
29. Gillies RJ, Kinahan PE, Hricak H. Radiomics: images are more than pictures, they are data. *Radiology*. (2016) 278:563–77. doi: 10.1148/radiol.2015151169
30. Hosny A, Parmar C, Coreller TP, Grossmann P, Zeleznik R, Kumar A, et al. Deep learning for lung cancer prognostication: a retrospective multi-cohort radiomics study. *PLOS Med*. (2018) 15:e1002711. doi: 10.1371/journal.pmed.1002711
31. Capobianco E, Dominietto M. From medical imaging to radiomics: role of data science for advancing precision health. *J Pers Med*. (2020) 10:15. doi: 10.3390/jpm10010015
32. Sevakula RK, Singh V, Verma NK, Kumar C, Cui Y. Transfer learning for molecular cancer classification using deep neural networks. *IEEE/ACM Trans Comput Biol Bioinform*. (2019) 16:2089–100. doi: 10.1109/TCBB.2018.2822803
33. Kensen A, Harrison PJ, Spjuth O. Transfer learning with deep convolutional neural networks for classifying cellular morphological changes. *SLAS Discov*. (2019) 24:466–75. doi: 10.1177/2472555218818756
34. Xu Y, Hosny A, Zaleznik R, Parmar C, Coreller T, Franco I, et al. Deep learning predicts lung cancer treatment response from serial medical imaging. *Clin Can Res*. (2019) 25:11. doi: 10.1158/1078-0432.CCR-18-2495
35. Matos Jd, Britto AdS, Oliveira LES, Koerich AL. "Double Transfer Learning for Breast Cancer Histopathologic Image Classification," In: *2019 International Joint Conference on Neural Networks (IJCNN)*. Budapest (2019) 1–8. doi: 10.1109/IJCNN.2019.8852092.
36. Yang Y, Yan LF, Zhang X, Han Y, Nan HY, Hu YC, et al. Glioma grading on conventional MR images: a deep learning study with transfer learning. *Front Neurosci*. (2018) 12:804. doi: 10.3389/fnins.2018.00804
37. Peter R, Korfiatis, P, Blezek D, Beitia AO, Stepan-Buksakowska I, Horinek D, et al. A quantitative symmetry-based analysis of hyperacute ischemic stroke lesions in noncontrast computed tomography. *Med Phys*. (2017) 44:192–9. doi: 10.1002/mp.12015
38. Zhukov DO, Andrianova EG, Lesko SA. The influence of a network's spatial symmetry, topological dimension, and density on its percolation threshold. *Symmetry*. (2019) 11:920. doi: 10.3390/sym11070920
39. Sander Dieleman S, De Fauw J, Kavukcuoglu K. Exploiting cyclic symmetry in convolutional neural networks. *ICML'16: Proceedings of the 33rd International Conference on Machine Learning* (New York, NY). (2016) p. 1889–98.
40. Whalen AJ, Brennan SN, Sauer TD, Schift SJ. Observability and controllability of nonlinear networks: the role of symmetries. *Phys Rev X*. (2015) 5:011005. doi: 10.1103/PhysRevX.5.011005
41. Capobianco E. Next generation networks: featuring the potential role of emerging applications in translational oncology. *J Clin Med*. (2019) 8:E664. doi: 10.3390/jcm8050664

Conflict of Interest: The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Systems Neuroscience

ORIGINAL RESEARCH
published: 02 September 2020
doi: 10.3389/fnsys.2020.00043

EEG-Based Emotion Classification Using a Deep Neural Network and Sparse Autoencoder

Junxiu Liu[1,2], Guopei Wu[1,2], Yuling Luo[1,2], Senhui Qiu[1,2,3]*, Su Yang[4], Wei Li[5,6] and Yifei Bi[7,8]

[1] School of Electronic Engineering, Guangxi Normal University, Guilin, China, [2] Guangxi Key Lab of Multi-Source Information Mining & Security, Guangxi Normal University, Guilin, China, [3] Guangxi Key Laboratory of Wireless Wideband Communication and Signal Processing, Guilin, China, [4] Department of Computer Science and Software Engineering, Xi'an Jiaotong-Liverpool University, Suzhou, China, [5] Academy for Engineering & Technology, Fudan University, Shanghai, China, [6] Department of Electronic Engineering, The University of York, York, United Kingdom, [7] College of Foreign Languages, University of Shanghai for Science and Technology, Shanghai, China, [8] Department of Psychology, The University of York, York, United Kingdom

Emotion classification based on brain–computer interface (BCI) systems is an appealing research topic. Recently, deep learning has been employed for the emotion classifications of BCI systems and compared to traditional classification methods improved results have been obtained. In this paper, a novel deep neural network is proposed for emotion classification using EEG systems, which combines the Convolutional Neural Network (CNN), Sparse Autoencoder (SAE), and Deep Neural Network (DNN) together. In the proposed network, the features extracted by the CNN are first sent to SAE for encoding and decoding. Then the data with reduced redundancy are used as the input features of a DNN for classification task. The public datasets of DEAP and SEED are used for testing. Experimental results show that the proposed network is more effective than conventional CNN methods on the emotion recognitions. For the DEAP dataset, the highest recognition accuracies of 89.49% and 92.86% are achieved for valence and arousal, respectively. For the SEED dataset, however, the best recognition accuracy reaches 96.77%. By combining the CNN, SAE, and DNN and training them separately, the proposed network is shown as an efficient method with a faster convergence than the conventional CNN.

Keywords: EEG, emotion recognition, convolutional neural network, sparse autoencoder, deep neural network

OPEN ACCESS

Edited by:
Raina Robeva,
Randolph–Macon College,
United States

Reviewed by:
Lina Yao,
University of New South Wales,
Australia
Xiang Li,
National Supercomputer Center,
China

****Correspondence:***
Senhui Qiu
qiusenhui@mailbox.gxnu.edu.cn

Received: *29 March 2020*
Accepted: *12 June 2020*
Published: *02 September 2020*

Citation:
Liu J, Wu G, Luo Y, Qiu S, Yang S,
Li W and Bi Y (2020) EEG-Based
Emotion Classification Using a Deep
Neural Network and Sparse
Autoencoder.
Front. Syst. Neurosci. 14:43.
doi: 10.3389/fnsys.2020.00043

1. INTRODUCTION

The Brain–Computer Interface (BCI) directly connects human (or animal) brain activity with artificial effectors (Kübler et al., 2009), which provides an interactive pathway between the human brain and external devices for various applications. The process of such an interaction starts by recording the brain activity through the signal processing and analysis to detect the users' intent (Tabar and Halici, 2016). BCI systems and their various implementations have been subjects of ongoing study for decades, and one of the most appealing research directions is emotion recognition due to its potential applications in numerous scenarios. Both non-physiological and physiological signals could be employed for emotion detections. Non-physiological signals include facial expression images (Lane et al., 1997), voice signals (Scherer, 1995), and body gesture (Cheng and Liu, 2008). Compared to the non-physiological signals, physiological signals can be detected by some wearable devices, such as an electroencephalogram (EEG) (Zheng, 2017), electromyogram (Hiraiwa et al., 1989), electrocardiogram (Agrafioti et al., 2012), the galvanic skin response, blood volume pressure, and a photoplethysmogram. Among these physiological signals, EEG signals have been widely used for research into emotion recognition (Chi et al., 2012; Huang et al., 2015; Li et al., 2016; Liu et al., 2018c). Captured from the scalp by a number of EEG electrodes, emotion could be reflected immediately by an EEG signal once a subject receives the stimulations.

There are two conventional rules to follow when categorizing human emotions, namely, the discrete basic emotion description and the dimension approaches. According to the discrete basic emotion description approach, emotions can be classified into six basic emotions: sadness, joy, surprise, anger, disgust, and fear (van den Broek, 2013). For the dimension approach, the emotions can be classified into two (valence and arousal) or three dimensions (valence, arousal, and dominance) (Zheng and Lu, 2015). Among these dimensions, valence describes the level of positivity or negativity of one person, and arousal describes the level of excitement or apathy of emotion. The scale of dominance ranges from submissive (without control) to dominance (empowered). The emotion recognition is usually based on the dimension approach because of its simplicity compared to the discrete basic emotion description (Zheng and Lu, 2015).

Early works on emotion recognition through analysing EEG signal could be traced back to more than 50 years ago (Fink, 1969). Many new methods on feature extraction and classification have recently been proposed for emotion detection (Petrantonakis and Hadjileontiadis, 2010). For the feature extraction, two types of feature are commonly used to analyze EEG signals: time-domain and frequency-domain features. Time-domain features capture the temporal information of signals, such as the fractal dimension (Hjorth, 1970), Hjorth, and higher-order crossing features (Petrantonakis and Hadjileontiadis, 2010). The frequency-domain features can extract the useful information from the frequency perspective under different frequency bands. For instance, the EEG signal could be decomposed into δ (1–3 Hz), θ (4–7 Hz), α (8–13 Hz), β (14–30 Hz), and γ bands (31–50 Hz) (Hjorth, 1970; Li and Lu, 2009; Petrantonakis and Hadjileontiadis, 2010; Nie et al., 2011), where the features can be extracted from each of them. In addition, other features, such as Deep Forest (Zhou and Feng, 2017), Statistical Characteristics (SC), Differential Entropy (DE) feature (Zheng et al., 2014), Pearson Correlation Coefficient (PCC) feature (Lewis et al., 2007), and Principal Component Analysis (PCA) (Subasi and Gursoy, 2010), are also used in emotion recognitions.

In the meantime, various classification methods have been used for emotion recognition, such as k-Nearest Neighbor (Bahari and Janghorbani, 2013), Multi-Layer Perceptron (Orhan et al., 2011). A Support Vector Machine (SVM) and Linear Regression (LR) were used in Wang et al. (2019), but recognition accuracy can be improved. In recent years, deep neural networks (DNN) (Tripathi et al., 2017) has been developed into one of the most effective and popular methods in many research fields (Fu et al., 2017; Liu et al., 2018a,b, 2019; Luo et al., 2018). Convolutional Neural Networks (CNN) are widely used in computer vision, image classifications, visual tracking (Danelljan et al., 2016), segmentation, and object detections (Girshick et al., 2014). EEG emotion classification using the CNN method was also explored in the approaches of Tripathi et al. (2017). Cascade and parallel convolutional recurrent neural networks have been used for EEG human-intended movements classification tasks (Zhang et al., 2018). Additionally, before applying the CNN, EEG data could be converted to image representation after feature extraction (Tabar and Halici, 2016). However, the accuracy of emotion recognition by using only CNN is not high. In the work of Zhang et al. (2017), a deep learning framework consisting of the sparse autoencoder (SAE) and logistic regression was used to classify EEG emotion status. The sparse autoencoder was employed for feature extraction, and logistic regression was used to predict affective states. The SAE is an unsupervised machine learning algorithm. By calculating the error between the output of the SAE and original input, data could be reconstructed and useful features could be extracted for classification task. However, accuracy of that work is not high and there are no experiments for comparing to verify the work of the SAE.

In this work, a novel network model combining the CNN, SAE, and DNN to convert EEG time series into 2D images for a good emotion classification performance is proposed. The EEG signal is decomposed into several different bands. Based on frequency, time, and location information, the 2D features are extracted from EEG data. Then convolutional layers of the CNN are trained and used for further extracting features. The SAE is used for reconstructing data obtained from convolutional layers, and the DNN is used for classification. Compared to other approaches, the proposed neural network model, which leverages the benefits of convolutional layers of the CNN and sparsity of the SAE, demonstrates a good classification accuracy and fast convergence. The procedure of the proposed method is summarized in **Figure 1**. Original EEG data are pre-processed, and features are extracted for deep learning model. After training and testing on the model, final classification results are obtained.

The rest of this paper is organized as follows: the proposed neural network model is presented in section 2. Datasets and experimental results are provided in section 3. Section 4 summaries the work and discusses the future work.

2. DEEP LEARNING FRAMEWORK

In this section, fundamental principles and essential network modules are presented. The novel model is also introduced in detail.

2.1. Convolution Neural Network (CNN)

The features extracted from original EEG data are sent to the CNN first. The CNN model includes several convolution-pooling layer pairs and one output layer. Before sending to the CNN, features are concatenated into image form which is then convolved with several one-dimensional filters in convolution layers. After the pooling layer, the data are further subsampled to images with smaller size. Network weights and filters in the convolution layers are learned through back-propagation algorithm.

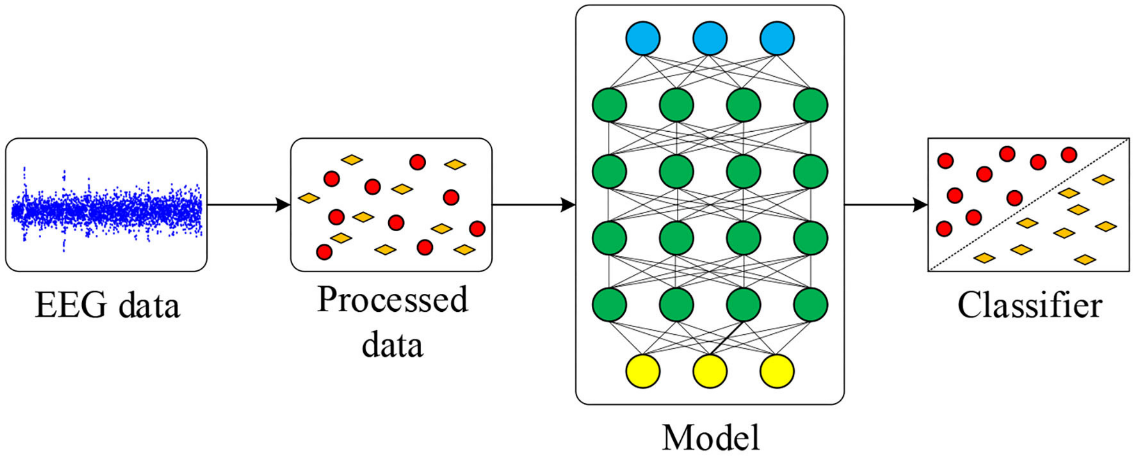


FIGURE 1 | Emotion classification procedure in this work.

In our experiments, data extracted from EEG signal are from four typical frequency bands, which include α (1–7 Hz), β (8–13 Hz), θ (14–30 Hz), and γ bands (30–45 Hz), using a Butterworth band-pass filter. After that, data are reformed into two-dimensional features, such as PCC, which are the input for CNN. Detailed methods of this procedure is presented in sections 3.3 and 3.4. It is worth noting that the two-dimensional features contain not only the frequency but also spatial location information of each electrode (Tabar and Halici, 2016). To preserve this information, one-dimensional filtering is applied in this work instead of two-dimensional filtering.

The CNN structure is relatively straightforward. Input vector is two-dimensional feature, which can be given by

$$x = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{pmatrix}, \tag{1}$$

where $m \times n$ is the shape of input vector x. The input two-dimensional feature is convolved with filters W_k at the convolution layer, which is given by

$$W_k = \begin{pmatrix} W_{11} \\ W_{21} \\ \dots \\ W_{i1} \end{pmatrix}, \tag{2}$$

where i is length of W_k and $i<m$ in Equation (1). After the image convolution, output map is formed and the feature map at the given layer is obtained by

$$f(\alpha) = f(W_k \times x + b_k), \tag{3}$$

where $W_k \in R^{i\times 1}$ is the weight matrix and b_k is the bias value, k denotes the filter, for $k = 1, 2, \dots, n$ and n denotes the total number of filtering in convolutional layer. The activation function is f, which is a rectified linear unit (*ReLU*) function in this work. Compared with the traditional neural network activation functions, such as *sigmoid* and *tanh*, *ReLU* is more efficient in avoiding gradient disappearance. *ReLU* function is defined by

$$f(\alpha) = ReLU(\alpha) = \ln(1 + e^{\alpha}), \tag{4}$$

where α is defined in Equation (3). At the max-pooling layer, the feature map is down sampled through the max-pooling function. Max-pooling is used because it is found that the maximum value from the selected values of a given feature map could be effectively extracted using this function.

After the last pooling layer, a fully connected layer follows in which output data from pooling layer is flattened. After that, fully connected layers named DNN are followed. In DNN, the activation function of each layer is also *ReLU*. For the output layer, because there are two classification tasks, including binary classification and multi-class classification, *sigmoid* and *softmax* are used, respectively. For the binary-classification task, *Adadelta* is used as an optimizer, and loss is calculated by binary crossentropy, which is given by

$$loss = -\sum_{n=1}^{N} \hat{y}_i log y_i + (1 - \hat{y}_i) log(1 - \hat{y}_i), \tag{5}$$

where N is number of samples, y_i is the value, which is a form of one-hot code, and $\hat{y}_i$ is the output from the output layer where *sigmoid* is used. For multi-class classification, such as three-class classification, *Adam* is used as an optimizer, and loss is calculated by categorical crossentropy, which is given by

$$loss = -\sum_{n=1}^{N} \hat{y}_{i1} log y_{i1} + \hat{y}_{i2} log y_{i2} + \hat{y}_{i3} log y_{i3}, \tag{6}$$

where N is number of samples, y_{i1}, y_{i2}, y_{i3} are values of the label, which is also a form of one-hot code, and $\hat{y}_{i1}, \hat{y}_{i2}$, and $\hat{y}_{i3}$ are three outputs from the output layer where *softmax* is used. Parameters

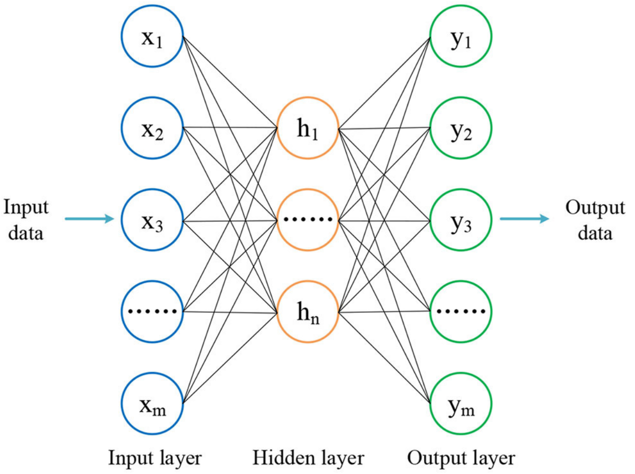


FIGURE 2 | The autoencoder includes one input, one hidden, and one output layer.

in the model are updated by using back-propagation algorithm. The error between the desired output and the actual output is computed and the gradient descent method is applied to update parameters in order to minimize the error. Functions to update the weight and bias are shown by

$$W_k = W_k - \eta \frac{\partial E}{\partial W_k}, \tag{7}$$

$$b_k = b_k - \eta \frac{\partial E}{\partial b_k}, \tag{8}$$

where W_k is the weight matrix, b_k is the bias, and η represents the learning rate, E is the error. E is equal to *loss* in Equations (5) and (6). The results obtained from this CNN will be used as a benchmark for the performance comparison in section 3.

2.2. Sparse Autoencoder (SAE)

An autoencoder is a network including one input, one hidden, and one output layer, which is used to preserve the essence of the input data as much as possible and remove the potential noise in an unsupervised manner. The output data are therefore simplified, and important information from the input data are retained, which is beneficial for classification.

The structure of autoencoder is shown in **Figure 2**. The whole data processing is divided into encoding and decoding phases. In the encoding phase, the dimension of input data are reduced in one layer. When the decoded data arrives at the hidden layer, the dimension of input data reaches the same as the number of neurons predefined for this layer. The encoding function of the hidden layer, h, is defined by

$$h = encoder(x) = f(W_k \times x + b_k), \tag{9}$$

where $W_k \in R^{m\times n}$ is the weight matrix between input layer and the next layer. As defined previously in CNN, b_k is also the bias vector, and f represents the output function. The output function used in this part is *ReLU*, which is similar to the activation of the CNN. Differently from the encoding phase, in the decoding phase, the same number of neurons in output layers should be set as that of layers in encoding phase, in order to guarantee the output data has the same dimension as the input data. The decoding function is shown by

$$y = decoder(x) = g(W_k \times x + b_k), \tag{10}$$

where $W_k \in R^{n\times m}$. After encoding and decoding phases, the model is trained, and the parameters could be obtained by minimizing the cost function, which is defined by

$$min \sum \left|E(x^i, y^i)\right|, \tag{11}$$

where y^i is output data and x^i is original input data. When the network is trained, output values are reconstructed, and the shape of which is equal to that of input data. Parameters of the model could be updated according to

$$W_k = W_k - \eta \frac{\partial E(x^i, y^i)}{\partial W_k}, \tag{12}$$

$$b_k = b_k - \eta \frac{\partial E(x^i, y^i)}{\partial b_k}, \tag{13}$$

where η denotes the learning rate of the network. E is an error in the SAE. For details of optimizer and E, they are the same as that in section 2.1 in binary-classification task.

In order to increase the generalization of the network and improve the training efficiency of the proposed network, a sparse constraint on the activity of the hidden representations is added in this work. Sparse constraint helps suppress activation of neurons in the hidden layer, and useful features can be extracted by autoencoder. Thus, the cost function in sparse autoencoder is described by

$$J_{sparse}(W, b) = J(W, b) + \beta \sum_{j=1}^{m} KL(\rho||\hat{\rho}_j), \tag{14}$$

where $\hat{\rho}_j$ is the average activation of hidden unit j, ρ is the sparsity level, and β is the weight of the sparsity penalty term. KL is the Kullback–Leibler divergence, which ensures the sparsity of neurons in hidden layer. KL is defined by

$$KL(\rho||\hat{\rho}_j) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho)\frac{1-\rho}{1-\hat{\rho}_j}, \tag{15}$$

$$\hat{\rho}_j = \frac{1}{m}\sum_{i=1}^{m} f_j(x^i), \tag{16}$$

where m denotes the number of samples at unite j in the hidden layer, and f_j denotes the activation of hidden neuron j.

2.3. Combined CNN-SAE-DNN

EEG signal is quite sensitive to a variety of factors during acquisition, such as environmental interference and the emotional fluctuations of humans. Therefore, EEG signals may be mixed with a variety of noise, which would undoubtedly influence the required brain patterns and the experimental results. In addition, in some experiments, subjects were unable to perform the emotion collection task successfully and the experimental results were deviated greatly. In order to overcome these problems, a deep learning network structure is proposed in this work. The structure of the proposed network is shown in **Figure 3**.

As shown by **Figure 3**, in the proposed network, the CNN structure consists of two convolutional layers and one max-pooling layer. Dropout connects to each convolutional layer. The SAE consists of one encode, one hidden, and one decode layer. In the DNN, there are three fully connected layers used for classification. Given features, such as PCC for input of the proposed network, the output of max-pooling layer is used as the input for the SAE. Finally, the output of the SAE is used as the input of the DNN for classification.

The training procedure is that the CNN with one fully-connected output layer are trained for some epochs using all samples and all features, and the output layer is abandoned after training. Then, by sending features to input the trained CNN, the output of the max-pooling layer can be obtained. The output is flattened to one-dimension data, and it is set as the input of SAE. After unsupervised learning of the SAE, data are reconstructed. The reconstructed data are divided for training and testing in the DNN, i.e., the CNN and SAE are trained separately. Thus, before data are classified in the DNN, training in the CNN and SAE can be seen as a part of feature extraction. It should be noticed that the DNN used for finally classification is not the fully-connected output layer abandoned from the CNN in the first step. The DNN is never trained before output of the SAE is obtained as input data for the DNN.

Another CNN with the same parameters and structure as the whole proposed network is set as comparison in order to test the performance of the proposed network fairly. When adding more layers into this CNN, accuracy does not improve and leads to an overfitting problem. For experiments on this CNN, features are split directly into 80% for training and the rest for testing.

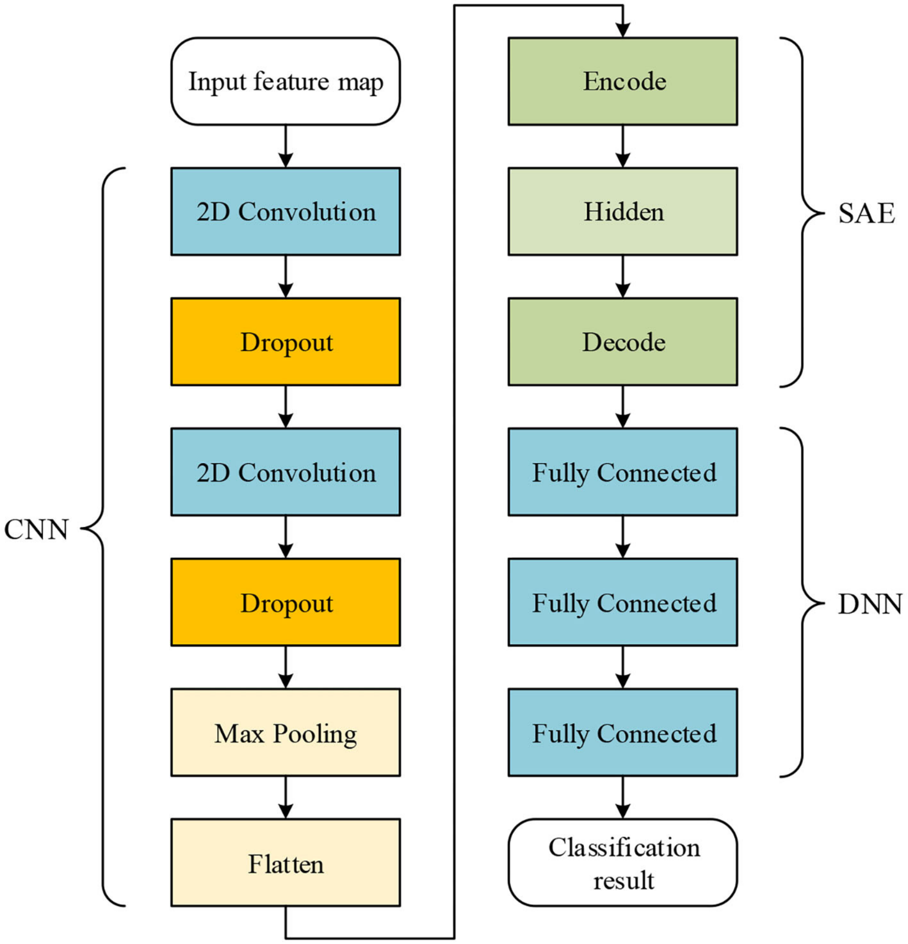


FIGURE 3 | The proposed network includes the CNN, SAE, and DNN; the CNN and SAE are used for feature extraction, and the DNN is used for classification.

3. DATASETS AND EXPERIMENTS

In this section, two datasets of DEAP (Koelstra et al., 2012) and SEED (Zheng and Lu, 2015) are used to evaluate the proposed network model. Data processing methods and experiment results are presented.

3.1. Emotional EEG Datasets

The DEAP dataset was collected from 32 subjects when they were watching 40 sets of 1-min music and video clips. The age of the subjects ranges between 19 and 37 years old, and half of them were males. During the 40 trials for each subject, various signals were recorded as 40-channel data, including EEG, electromyograms, breathing zone, plethysmographs, temperature, and so on (Koelstra et al., 2012). The EEG signal was recorded at 512 Hz. The data was segmented into trials of 60 s, and a bandpass frequency filter was applied after that. After each trial, the participants were asked to do a self-assessment about their emotional levels, including four different scales, such as valence, arousal, dominance, and liking.

The EEG signal is downsampled into 128 Hz for the experiments in this work, where the frequency of EEG data are from 4.0 to 45.0 Hz. Valence and arousal are the two scales chosen for this work. Each of them ranges from one (low) to nine (high), and scales are divided into two parts to construct our binary-classification tasks. Similarly to the work in Koelstra et al. (2012), valence is divided into high (ranging from five to nine) and low valence (range from one to five) according to the valence scale, and according to the arousal scale, arousal is divided into high (ranging from five to nine) and low arousal (ranging from one to five).

The SEED dataset was collected from 15 subjects (seven males) when they were asked to watch 15 film clips. The duration of each film clip was about 4 min, and each film as easily understood in order to elicit emotion of 15 subjects participating in the experiments effectively. There were 15 trials for each subject and each trial lasted for 305 s consisting of a hint of start for 5 s, a movie clip for 4 min, a self-assessment for 45 s, and a rest for 15 s. EEG data in SEED dataset was collected from 62 electrodes, which includes more information than the DEAP dataset. After collection, EEG data was downsampled to 200 Hz and applied with a bandpass filter from 0 to 75 Hz.

Similar to the DEAP dataset, in this dataset, the data are applied with a frequency filter from 4.0 to 45.0 Hz in order to equitably evaluate the proposed network. Negative, positive, and neutral are emotion labels in this dataset that represent the subjects' emotion states during each experiment. Label value of negative, positive and neutral is −1, 1, and 0, respectively. Thus, labels in the SEED dataset include three categories.

3.2. Experiment Setting

In order to test the efficacy of the proposed network, the CNN model and the proposed network are trained by using data obtained from two time windows of different lengths; in total, four groups of experiments were conducted. For experiments in the CNN used for comparison, after feature extraction of EEG data, 80% samples are used as training data and the rest samples are used as test data among all of the data. Average accuracy is calculated from accuracies of the last 10 epochs in each experiment. For the proposed network, before training data and testing data were divided, the CNN and SAE in the proposed network were trained using features. After that, features are sent to the input of the CNN, and the output data of SAE is obtained. The output data after feature extraction were divided into 80% for training and 20% for testing in the DNN. In this work, Keras and Tensorflow (Abadi et al., 2016) ere used for the proposed network implementation. For detailed free parameters in the proposed network, they are described in sections 3.3 and 3.4, respectively.

3.3. Experiments on the DEAP Dataset

Length of data in the DEAP dataset is 63 s, and the first 3 s are removed in the experiments. Then band pass filtering is then applied. Among 40 channels, EEG data are contained in 32 channels, which are chosen for experiments. After that, EEG signals are decomposed into α (1–7 Hz), β (8–13 Hz), θ (14–30 Hz), and γ bands (30–45 Hz). After band pass filtering, signal windowing on four frequency bands is applied. EEG signals are divided into short time frames in order to facilitate signal processing, thus time windows with different overlaps are applied to EEG data in order to increase samples for training. Two window sizes, 8 and 12 s, are used for evaluating the proposed network. From the start of each recorded EEG signal, data are segmented by a sliding time window with an overlap for each frequency band. For each trial of 60 s, 14 segments are obtained using an 8-s time window moving every 4 s, and seven segments are obtained using a 12-s time window moving every 8 s. Finally, from a total of 32 participants, 17,920 (14 segments × 40 trials × 32 participants) and 8,960 (seven segments × 40 trials × 32 participants) samples are obtained using time windows of 8 and 12 s, respectively. Segment labels are the same as the label of the original sample.

After that, three different features, namely PCC, PCA, and SC, are extracted to evaluate the proposed network. For PCC-based features, PCC of data in every two channels are calculated, and a 32 × 32 PCC matrix is constructed for one sample. For PCA-based features, dimension of data from each channel is reduced into 32, and features with the shape of 32 × 32 are obtained. For SC-based features, four different characteristics are extracted, including variance, mean, kurtosis, and skewness. These statistical characteristics of data are calculated together, and a 32 × 4 matrix is finally obtained. In the proposed work, the features are separately extracted in each of the frequency bands (α, β, θ, and γ bands). According to the work in Wang et al. (2018) and other similar researches, data of four frequency bands are used together in order to get the best results. After data are processed, for the data obtained using a time window of 8 s, the shapes of the above three different feature matrixes are 17,920 × 4 × 32 × 32, 17,920 × 4 × 32 × 32, and 17,920 × 4 × 32 × 4, respectively. For data obtained using a time window of 12 s, they are 8,960 × 4 × 32 × 32, 8,960 × 4 × 32 × 32, and 8,960 × 4 × 32 × 4, respectively. Detailed configuration of the proposed network for DEAP dataset is shown by **Figure 4**. For SC, input

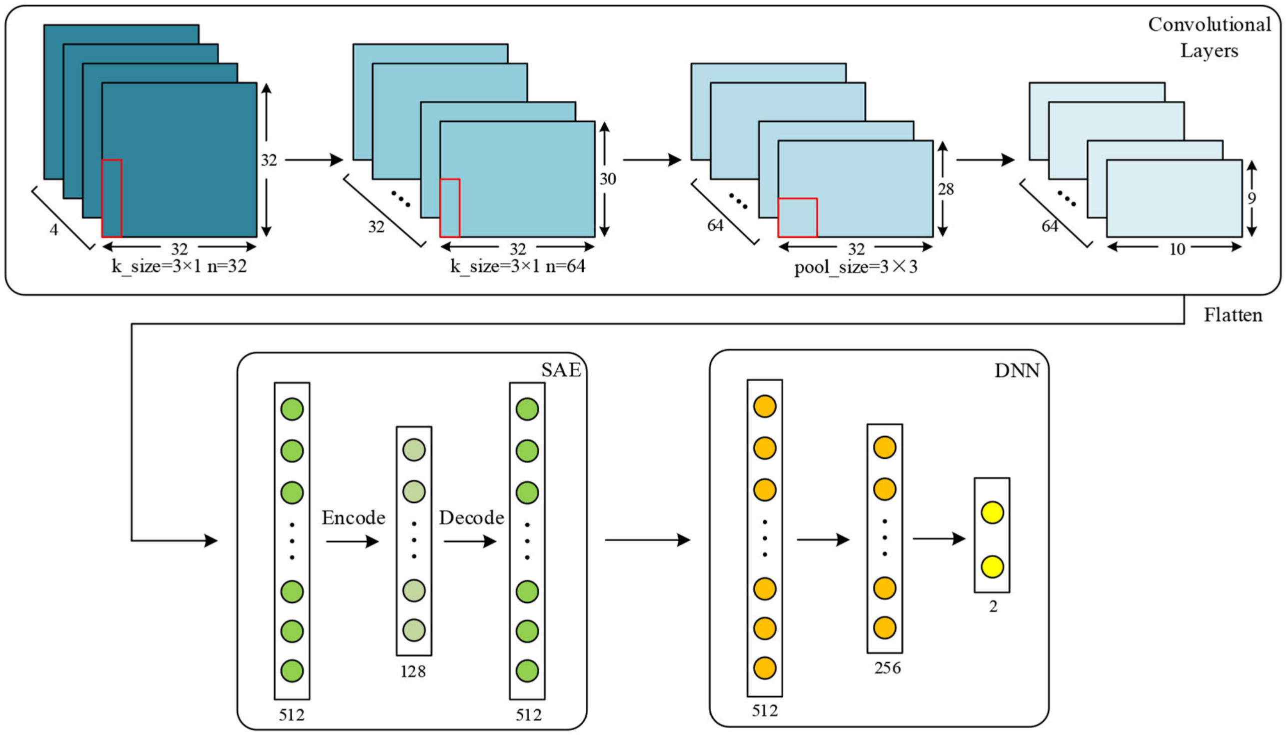


FIGURE 4 | Configuration of the proposed network for the DEAP dataset.

shape is 32 × 4. These features are two-dimensional, which are suitable inputs for the CNN and the proposed network.

As shown in **Figure 4**, for the DEAP dataset, two convolutional layers and one max-pooling layer are applied for the proposed network. Kernel size is set to 3 × 1, and pooling size is set to 3 × 3. The input data shape is 32 × 32. The numbers of kernels in convolutional layer are set to 32 and 64, respectively. In the SAE, the numbers of neurons in encode, hidden, and decode layers are set to 512, 128, and 512, respectively. In the DNN, the numbers of three fully connected layers are set to 512, 256, and 2, respectively. In the proposed network, the training epochs, batch size, and learning rate in the CNN are set to 50, 128, and 0.01. Epoch, batch size, and learning rate in the SAE are set to 100, 64, and 0.01, respectively. For those of the DNN, they are set to 100, 128, and 0.01, respectively.

In the proposed network, the training epochs are carried out in convolutional layers, and the SAE for features extraction, training, and testing epochs are carried out in the DNN for classification. Another CNN with the same parameters and structure as the proposed network served as a baseline method to evaluate the performance of the proposed network. The epoch, batch size, and learning rate of this CNN were set to 100, 128, and 0.01. Parameters in this CNN were the same as that of the proposed network. The data results of the experiments using a time window of 8 s are shown by **Table 1**.

TABLE 1 | Average accuracies comparisons of the DEAP dataset using different features extracted from the data with a length of 8 s between two networks.

Network	Labels	PCC (%)	PCA (%)	SC (%)
CNN	Valence	78.80	73.32	71.10
	Arousal	82.25	72.76	73.04
Proposed network	Valence	89.49	75.59	81.93
	Arousal	92.86	85.87	82.94

From **Table 1**, among all features extracted from EEG data, we can see the PCC feature was demonstrated to be better than most of the other features on both the CNN and the proposed network. The proposed network can reach a recognition accuracy of 92.86% on arousal by using PCC. Moreover, recognition accuracies of most experiments on the proposed network are better than the CNN (3.27–13.11% improvement). As described previously, this is due to the inclusion of SAE, which can not only reconstruct data from convolutional layers and pooling layer but can also extract features further and make the data easier to be recognized than the CNN.

Training for loss of SAE is shown in **Figure 5**; data reconstruction is achieved when the loss does not change sharply, and data reconstruction is fast during the training process of SAE. For other extracted features (except PCC), the recognition accuracy of each method is better than the work in Zhang et al. (2016a) (81.21% for valence and 81.26% for arousal).

Figures 6, **7** show the accuracies of the CNN and the proposed network. Red lines in figures denote the average accuracy of the last 10 epochs. It can be seen that the accuracy of the CNN gradually converges. For the proposed network, the accuracy converges rapidly at the beginning of the epoch after fewer than 10 epochs. This is because features are easy to recognize using output data obtained from the SAE before they are classified by the DNN. For features extracted by the PCC and other methods, the accuracy of a proposed network has a faster convergence than CNN.

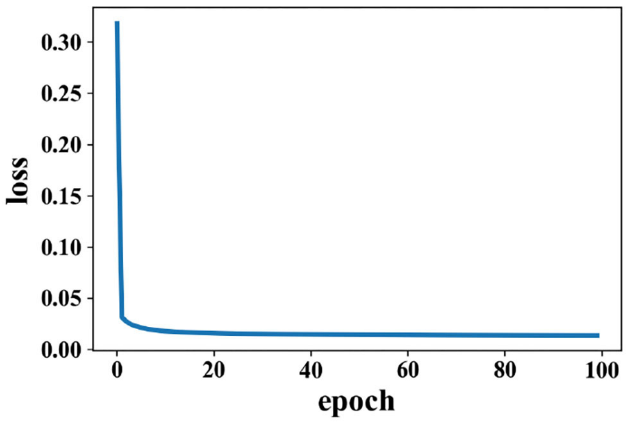


FIGURE 5 | Change of loss when data are reconstructed in the SAE on the DEAP dataset.

Similarly, results using data obtained from a time window of 12 s are shown in **Table 2**. From **Table 2**, we can see that the accuracy obtained using data with a length of 12 s is lower than

TABLE 2 | Average accuracy comparisons on the DEAP dataset using different features extracted from data with a length of 12 s between two networks.

Network	Labels	PCC (%)	PCA (%)	SC (%)
CNN	Valence Arousal	75.13 76.12	67.23 69.20	66.09 69.48
Proposed network	Valence Arousal	82.16 85.47	76.34 79.11	73.41 75.44

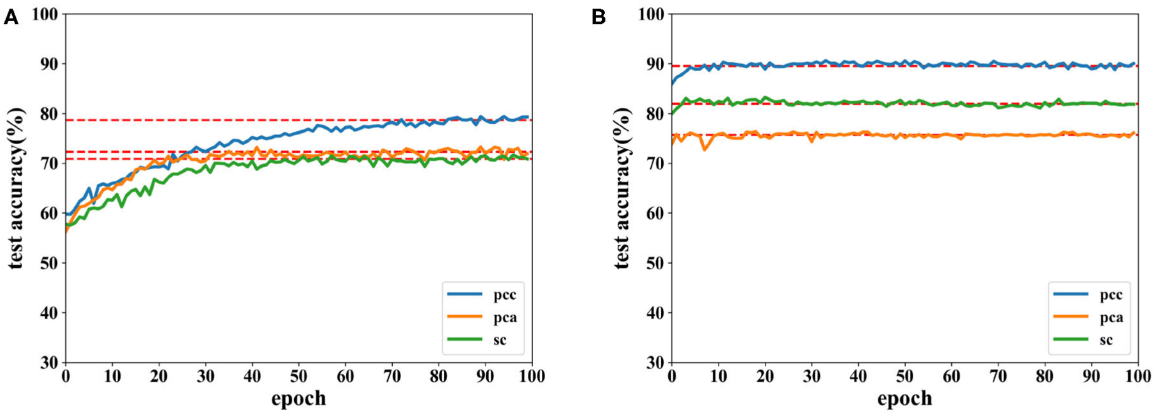


FIGURE 6 | Accuracy comparison of two networks on valence using data with a length of 8 s on the DEAP dataset in which **(A)** is result of the CNN and **(B)** the result of the proposed network.

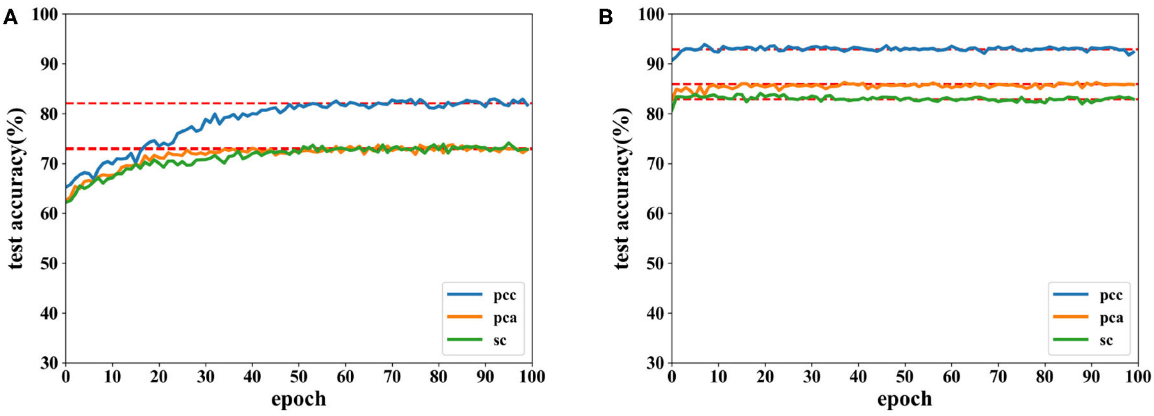


FIGURE 7 | Accuracy comparison of two networks on arousal using data with a length of 8 s on the DEAP dataset in which **(A)** is result of the CNN and **(B)** the result of the proposed network.

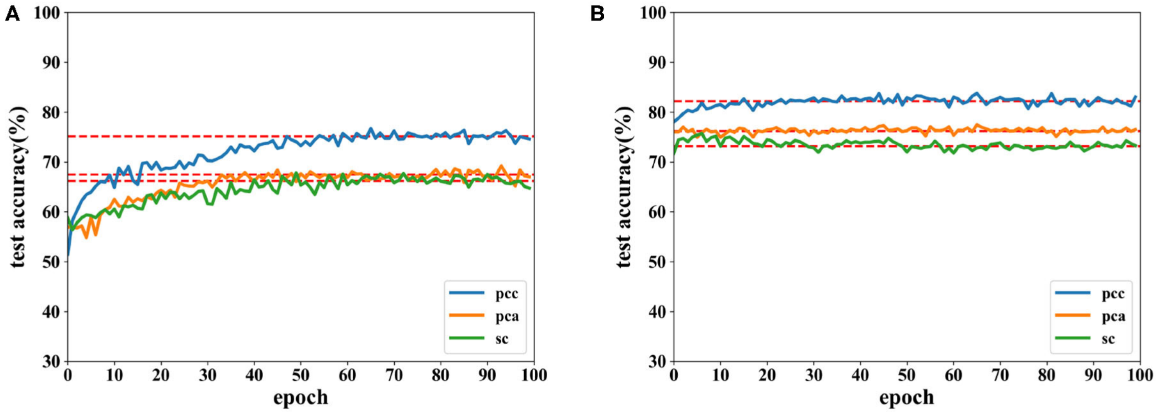


FIGURE 8 | Accuracy comparison of two networks on valence using data with a length of 12 s on DEAP dataset in which **(A)** is the result of the CNN and **(B)** the result of the proposed network.

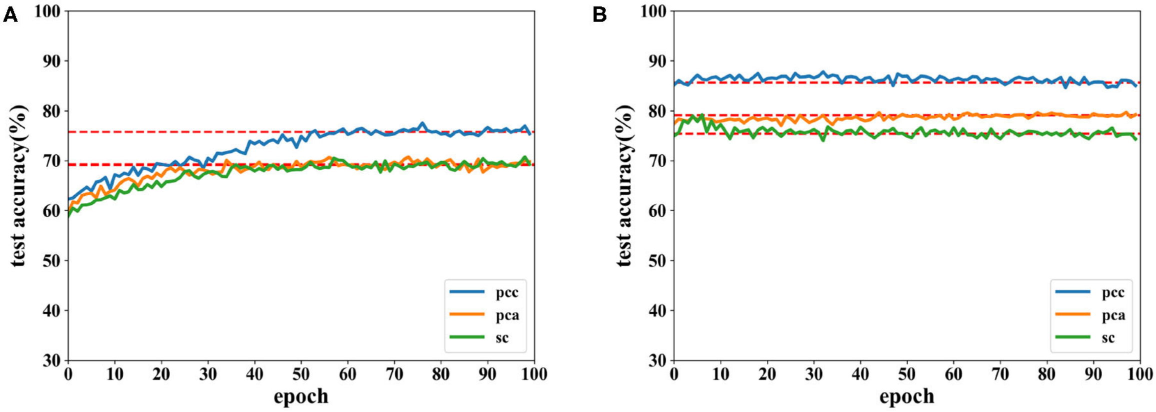


FIGURE 9 | Accuracy comparison of two networks on arousal using data with a length of 12 s on the DEAP dataset in which **(A)** is the result of the CNN and **(B)** the result of the proposed network.

that of 8 s. It is more difficult to collect emotion information when the stimulation time is increasing. Most studies related to the classification of EEG data was focused on a short length of time. In this experiment, higher classification accuracy is achieved on data of 12 s than that of shorter length in other studies; this is like the work in Zhang et al. (2017), which exhibits the effectiveness of the proposed network.

Accuracies for classification on data of 12 s on both CNN and the proposed network are shown by **Figures 8**, **9**. It can be found that higher recognition accuracy is obtained by the proposed network. Moreover, the classification accuracy of the proposed network has a faster convergence in each experiment.

The results in this subsection demonstrate that accuracies can reach 92.86% for data of 8 s and 85.47% for data of 12 s. When the same feature is used for comparison, the proposed network is more powerful in classifying the EEG emotion data than the CNN. Finally, the proposed network has a quicker convergence speed.

3.4. Experiments on SEED Dataset

There are a total of 675 trials in the SEED dataset. According to the work in Zheng and Lu (2015), the first sample of each subject was chosen, and a total of 225 samples were then obtained. Due to the different data length of each channel, the 80 s data segment was chosen to reduce the influence of unstable signals at the beginning and end of the whole signal; finally, data with the shape of 16,000 × 225 × 62 were obtained. Moreover, the data were processed as the same way as in the DEAP dataset: each sample was divided into different frames with different time windows. Two time windows, 8 and 12 s, were also used in the SEED dataset. A total of 19 and nine segments were obtained separately from data using a time window of 8 s moving every 4 s and a time window of 12 s moving every 8 s for each sample, respectively. Thus, in a total of 225 trials, 4,275 (19 segments × 15 trials × 15 participants) and 2,025 (9 segments × 15 trials × 15 participants) samples were obtained, respectively.

The detailed configuration of the proposed network for the SEED dataset is shown in **Figure 10**. The amount of data

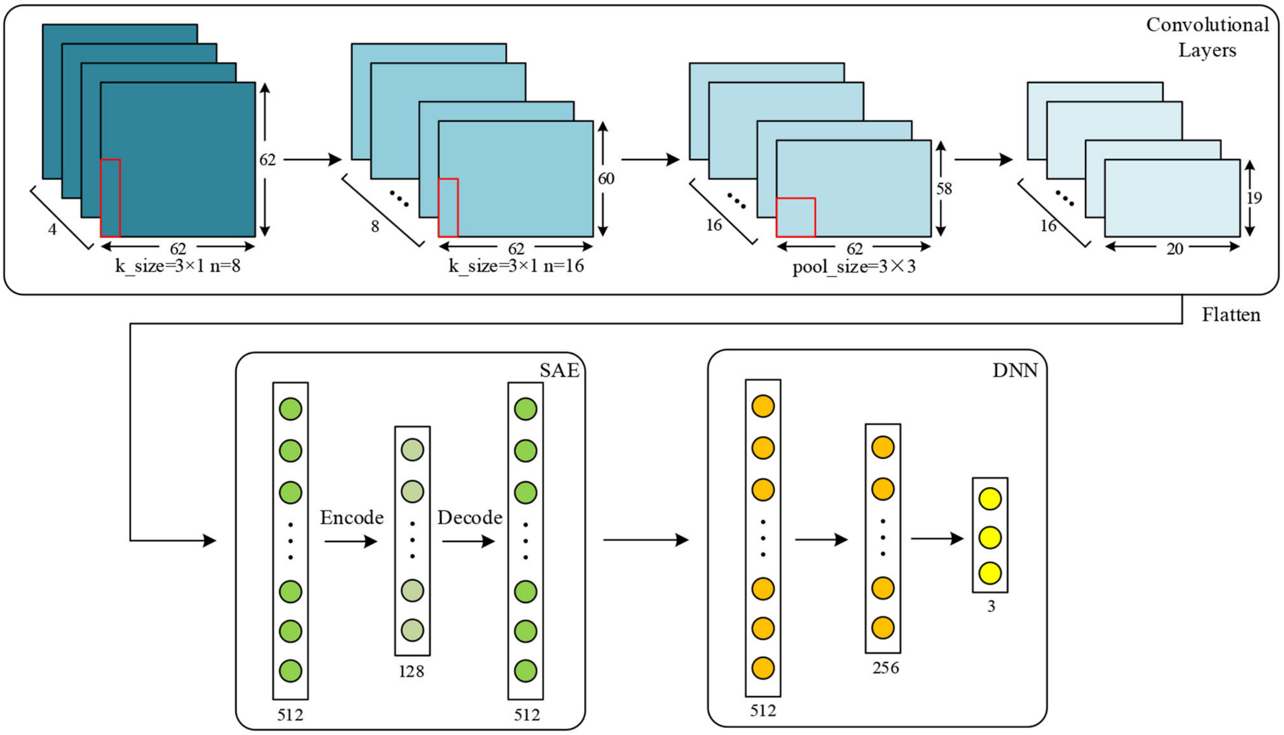


FIGURE 10 | Configuration of the proposed network for the SEED dataset.

TABLE 3 | Average accuracy comparisons on SEED dataset using different features extracted from data with a length of 8 s between two networks.

Network	Data length (s)	PCC (%)	PCA (%)	SC (%)
CNN	8 12	93.71 91.53	77.66 70.59	83.32 75.48
Proposed network	8 12	96.77 94.62	88.90 70.62	87.73 79.09

extracted from this dataset is much less than from the DEAP dataset, and the two classifiers used on this dataset are thus a little different. For PCC, input data are 62 × 62, and the numbers of kernels are separately set to eight and 16 in two convolutional layers. In the SAE and DNN, the number of each layer is set the same as that on the DEAP dataset except that the number of the output layer is set to three because this is a three-classification task on a SEED dataset. After training in the CNN and SAE for feature extraction, the DNN is used for the final classification. Similarly to the DEAP dataset, the same features are extracted for the SEED dataset. For PCA and SC, the input shape is 62 × 62 and 62 × 4, respectively. For the CNN used for comparison, parameters are also set as the same as the proposed network.

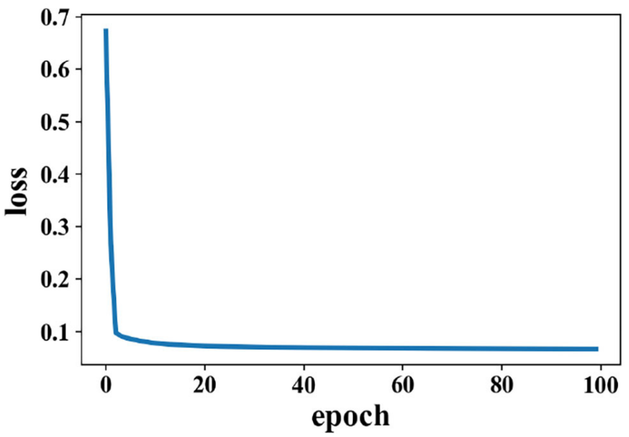


FIGURE 11 | Change of loss when data are reconstructed by the SAE on the SEED dataset.

The experiment results of data obtained from time windows of 8 and 12 s are shown in **Table 3**. The accuracy under this dataset is higher than the DEAP dataset. The highest average accuracy could reach 96.77%, which is better than the work in Wang et al. (2018), 90.2%. For the data obtained from time window of 12 s, the best accuracy could reach 94.62%, which shows that PCC-based features exhibit a better performance than others. The reconstruction of data by the SAE due to the change in loss on the SEED dataset is shown by **Figure 11**. Loss drops immediately following several epochs, i.e., the data reconstruction can be achieved quickly when the SAE is being trained.

Accuracies under different features extracted from data of 8 and 12 s are depicted in **Figures 12**, **13**. It is shown that recognition accuracies of the proposed network are better than the CNN for almost all features, especially the PCC-based features. The proposed network can achieve faster convergence on classification accuracy than the CNN on the SEED dataset. Experiments on these two datasets shows that the proposed network performs better than original the CNN in emotion recognition.

Moreover, EEG data divided by a fixed time window with different overlaps on the SEED dataset are tested. Besides a time window of 8 s with an overlap of 4 s, overlaps of 6 and 8 s are also tested. Due to the highest accuracy, PCC-based features are used in these experiments, and classification results are displayed in **Figure 14**.

As seen in **Figure 14**, recognition accuracy could reach the highest value while the overlap is 4 s. The shorter the overlap is, the more similar the neighboring data segments are, i.e., features could be learned better when similar information is included in each trial. However, when the overlap is too short, the number of data segments increases, which requires longer time for training. In this experiment, data with a length of 8 s and overlap of 4 s could achieve the best result.

In a short summary, the best recognition could reach 96.77% on the three-class classification. The proposed network is demonstrates to be more powerful in classifying EEG emotion data than the CNN on the SEED dataset. For the data with the same length, length of overlap has an impact on recognition accuracy where 4-s overlap obtained the best performance. In addition to this, the proposed network is also compared with other research works using the DEAP and SEED datasets, and the results can be seen in **Table 4**. For complexity analysis, the number of parameters are 7.55 × 10^5 and 7.50 × 10^5 for the networks used for DEAP and SEED, respectively.

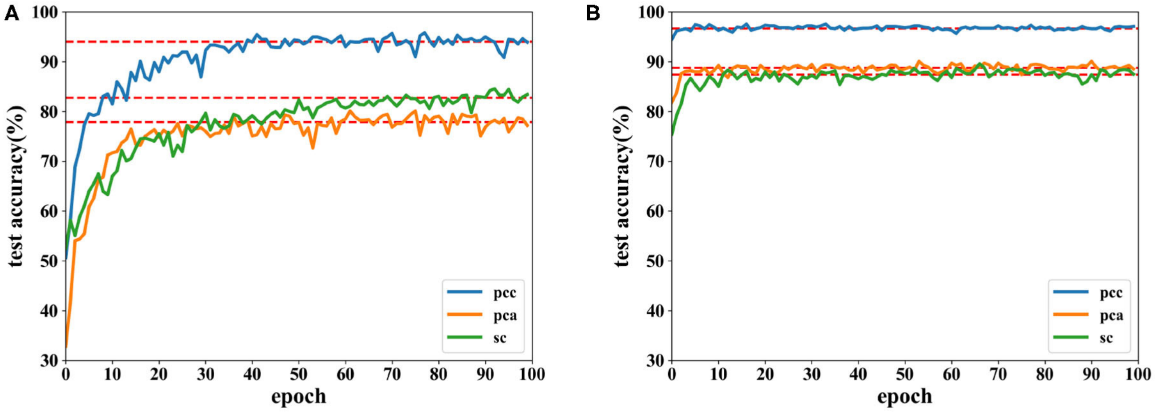


FIGURE 12 | Accuracy comparison of two networks using data with length of 8 s on the SEED dataset in which **(A)** is the result of the CNN and **(B)** the result of the proposed network.

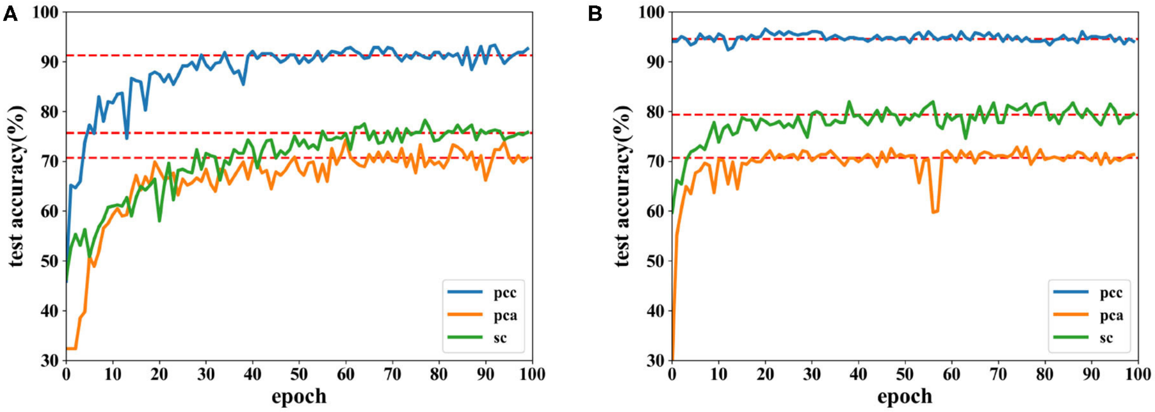


FIGURE 13 | Accuracy comparison of two networks using data with a length of 12 s on the SEED dataset in which **(A)** is the result of the CNN and **(B)** the result of the proposed network.

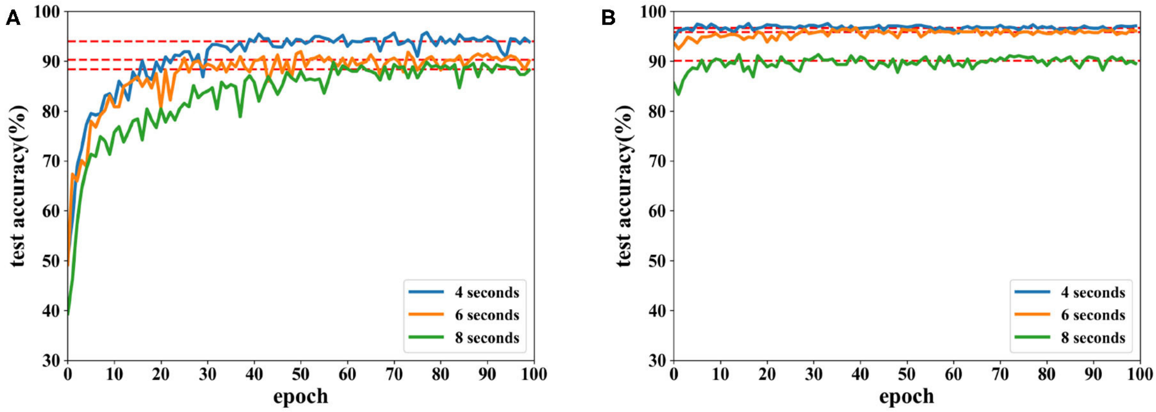


FIGURE 14 | Accuracy comparison of two networks using features extracted from different lengths of overlap on the SEED dataset in which **(A)** is the result of the CNN and **(B)** the result of the proposed network.

Table 4 shows the results of García et al. (2016) achieved 88.3% on valence and 90.6% on arousal. However, data used for experiments are limited, and the classification model is a better fit for classifying small amounts of data with high dimensions. The approach of Koelstra et al. (2012) used a Gaussian Bayes classifier, and experiment results proved that EEG signals are effective in emotion recognition of the DEAP dataset. The recent study (Tripathi et al., 2017) used extracted data for classification, and better accuracy results were obtained by using the CNN, where the classification accuracy of valence and arousal is 81.4 and 73.4%, respectively. The approaches of García et al. (2016) and Wang et al. (2018) used DE-based features and dynamical graph convolutional neural networks, and the accuracy achieved 93.7%. In the approach of Wang et al. (2019), BLSTM and other machine learning classifiers, such as SVM and LR were used for emotion recognition. BLSTM achieved the best accuracy of 94.96% on the SEED dataset, which is better than SVM and LR. In the approach of Soroush et al. (2019), phase space dynamics were introduced to classify emotions, achieving 87.42% on arousal and 84.59% on valence, respectively. A sparse discriminative ensemble was used for feature extraction in Ullah et al. (2019) and achieved 82.81% on valence and 74.53% on arousal, respectively. In this work, both the DEAP and SEED datasets are used for experiments, where accuracies achieve 89.49% and 92.86% in valence and arousal on the DEAP dataset, respectively, and 96.77% on the SEED dataset. Results demonstrate that the proposed network is more powerful than the CNN and other approaches.

TABLE 4 | Performance comparisons with other approaches.

Classification methods	DEAP dataset		SEED dataset (%)
	Valence (%)	Arousal (%)	
CNN + statistical methods (Tripathi et al., 2017)	81.4	73.4	/
Gaussian Bayes (Koelstra et al., 2012)	57.6	62.0	/
Deep SAE + RSP (Zhang et al., 2017)	73.1	80.8	/
BDGLS + DE (Wang et al., 2018)	/	/	93.7
DGCNN + DE (Zhang et al., 2016)	/	/	90.4
GP + LVM (García et al., 2016)	88.3	90.6	/
BLSTM + DE (Wang et al., 2019)	/	/	94.96
Physe Space Dynamics (Soroush et al., 2019)	84.6	87.4	/
SDEL + PCA (Ullah et al., 2019)	82.8	74.5	/
This work [PCC]	89.49	92.86	96.77

4. DISCUSSION AND CONCLUSION

There are some points worth discussing. First, the proposed model can be trained using an end-to-end method, which is different from this work. The end-to-end training method was tested, and it obtained a similar performance. However, the training model can be further investigated and optimized in a future work. Second, labels are used in feature extraction. It should be noted that many feature extraction algorithms use labels such Relief and ReliefF (Kira and Rendell, 1992), where feature weights are calculated according to samples in the same and different classes. Label information has been used in the feature extraction process (Bohgaki et al., 2014; Zhang et al., 2016b). Third, constructing an autoencoder-like structure is another method of emotion recognition, and this can be investigated in a future work.

In this work, a new deep network is proposed to classify EEG signals for emotion recognition. The CNN and the proposed network are applied for two different datasets, i.e., the DEAP and SEED datasets. In the proposed network, the CNN and SAE are trained for feature extraction in which, by combining supervised learning of the CNN and unsupervised learning of the SAE, more useful features are extracted. Experimental results show that the proposed network achieves a better performance than the CNN and other approaches. It also shows that when embedding an SAE structure into a CNN, the accuracy is better compared to a CNN with the same parameters and structure as the proposed network. In the proposed network, three different features are extracted for classifications. Results showed that, by using PCC-based features, the average recognition accuracy of the proposed network can reach 89.49% on valence and 92.86% on arousal for DEAP and 96.77% for SEED, where the proposed network has a faster convergence speed. In addition, overlap length also affects the performance, and results under the SEED dataset showed that data of 8 s with an overlap of 4 s can achieve the best result. It is also found that the data processed by the SAE is easily classified in the proposed network, which indicates that the SAE is effective in extracting features from EEG data. Future works will consider using the SAE and other classifiers to further improve the classification performance.

DATA AVAILABILITY STATEMENT

All datasets presented in this study are included in the article/supplementary material.

AUTHOR CONTRIBUTIONS

JL, GW, SQ, and YL developed, implemented, and evaluated the neural network algorithm. JL and GW wrote and revised the manuscript. SY, WL, and YB analyzed the performance of the proposed network and reviewed the manuscript. All authors contributed to the article and approved the submitted version.

FUNDING

This research is supported by the National Natural Science Foundation of China under Grant 61976063, the funding of Overseas 100 Talents Program of Guangxi Higher Education, research funds of Diecai Project of Guangxi Normal Univesity, Guangxi Key Lab of Multi-source Information Mining and Security (19-A-03-02) and Guangxi Key Laboratory of Wireless Wideband Communication and Signal Processing, the Young and Middle-aged Teachers' Research Ability Improvement Project in Guangxi Universities under Grant 2020KY02030, and the Innovation Project of Guangxi Graduate Education under Grant YCSW2020102.

REFERENCES

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., et al. (2016). "Tensorflow: a system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation* (Savannah, GA: USENIX Association), 265–283.

Agrafioti, F., Hatzinakos, D., and Anderson, A. K. (2012). ECG pattern analysis for emotion detection. *IEEE Trans. Affect. Comput.* 3, 102–115. doi: 10.1109/T-AFFC.2011.28

Bahari, F., and Janghorbani, A. (2013). "EEG-based emotion recognition using recurrence plot analysis and k nearest neighbor classifier," in *2013 20th Iranian Conference on Biomedical Engineering (ICBME)* (Tehran: IEEE), 228–233. doi: 10.1109/ICBME.2013.6782224

Bohgaki, T., Katagiri, Y., and Usami, M. (2014). Pain-relief effects of aroma touch therapy with citrus junos oil evaluated by quantitative EEG occipital alpha-2 rhythm powers. *J. Behav. Brain Sci.* 4, 11–22. doi: 10.4236/jbbs.2014.41002

Cheng, B., and Liu, G. (2008). "Emotion recognition from surface EMG signal using wavelet transform and neural network," in *2008 2nd International Conference on Bioinformatics and Biomedical Engineering* (Shanghai: IEEE), 1363–1366. doi: 10.1109/ICBBE.2008.670

Chi, Y. M., Wang, Y.-T., Wang, Y., Maier, C., Jung, T.-P., and Cauwenbe, G. (2012). Dry and noncontact EEG sensors for mobile brain–computer interfaces. *IEEE Trans. Neural Syst. Rehabil. Eng.* 20, 228–235. doi: 10.1109/TNSRE.2011.2174652

Danelljan, M., Robinson, A., Khan, F. S., and Felsberg, M. (2016). "Beyond correlation filters: learning continuous convolution operators for visual tracking," in *European Conference on Computer Vision*, eds B. Leibe, J. Matas, N. Sebe, and M. Welling (Amsterdam: Springer), 472–488. doi: 10.1007/978-3-319-46454-1_29

Fink, M. (1969). EEG and human psychopharmacology. *IEEE Trans. Inform. Technol. Biomed.* 9, 241–258. doi: 10.1146/annurev.pa.09.040169.001325

Fu, Q., Luo, Y., Liu, J., Bi, J., Qiu, S., Cao, Y., et al. (2017). "Improving learning algorithm performance for spiking neural networks," in *2017 IEEE 17th International Conference on Communication Technology (ICCT)* (Chengdu: IEEE), 1916–1919. doi: 10.1109/ICCT.2017.8359963

García, H. F., Álvarez, M. A., and Orozco, Á. A. (2016). "Gaussian process dynamical models for multimodal affect recognition," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (Orlando, FL: IEEE), 850–853. doi: 10.1109/EMBC.2016.7590834

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition* (Washington: IEEE), 580–587. doi: 10.1109/CVPR.2014.81

Hiraiwa, A., Shimohara, K., and Tokunaga, Y. (1989). "EMG pattern analysis and classification by neural network," in *Conference Proceedings, IEEE International Conference on Systems, Man and Cybernetics* (Cambridge, MA: IEEE), 1113–1115. doi: 10.1109/ICSMC.1989.71472

Hjorth, B. (1970). EEG analysis based on time domain properties. *Electroencephalogr. Clin. Neurophysiol.* 29, 306–310. doi: 10.1016/0013-4694(70)90143-4

Huang, Y.-J., Wu, C.-Y., Wong, A. M.-K., and Lin, B.-S. (2015). Novel active comb-shaped dry electrode for EEG measurement in hairy site. *IEEE Trans. Biomed. Eng.* 62, 256–263. doi: 10.1109/TBME.2014.2347318

Kira, K., and Rendell, L. A. (1992). "A practical approach to feature selection," in *Machine Learning Proceedings* (San Francisco, CA), 249–256. doi: 10.1016/B978-1-55860-247-2.50037-1

Koelstra, S., Muhl, C., Soleymani, M., Lee, J.-S., Yazdan, A., Ebrahimi, T., et al. (2012). Deap: a database for emotion analysis; using physiological signals. *IEEE Trans. Affect. Comput.* 3, 18–31. doi: 10.1109/T-AFFC.2011.15

Kübler, A., Furdea, A., Halder, S., Hammer, E. M., Nijboer, F., and Kotchoubey, B. (2009). A brain–computer interface controlled auditory event-related potential (p300) spelling system for locked-in patients. *Ann. N. Y. Acad. Sci.* 1157, 90–100. doi: 10.1111/j.1749-6632.2008.04122.x

Lane, R. D., Reiman, E. M., Bradley, M. M., Lang, P. J., Ahern, G. L., Davidson, R. J., et al. (1997). Neuroanatomical correlates of pleasant and unpleasant emotion. *Neuropsychologia* 35, 1437–1444. doi: 10.1016/S0028-3932(97)00070-5

Lewis, R. S., Weekes, N. Y., and Wang, T. H. (2007). The effect of a naturalistic stressor on frontal EEG asymmetry, stress, and health. *Biol. Psychol.* 75, 239–247. doi: 10.1016/j.biopsycho.2007.03.004

Li, M., and Lu, B.-L. (2009). "Emotion classification based on gamma-band EEG," in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (Minneapolis, MN: IEEE), 1223–1226.

Li, X., Hu, B., Sun, S., and Cai, H. (2016). EEG-based mild depressive detection using feature selection methods and classifiers. *Comput. Methods Prog. Biomed.* 136, 151–161. doi: 10.1016/j.cmpb.2016.08.010

Liu, J., Huang, Y., Luo, Y., Harkin, J., and McDaid, L. (2019). Bio-inspired fault detection circuits based on synapse and spiking neuron models. *Nerocomputing* 331, 473–482. doi: 10.1016/j.neucom.2018.11.078

Liu, J., Mcdaid, L. J., Harkin, J., Karim, S., Johnson, A. P., Millard, A. G., et al. (2018a). Exploring self-repair in a coupled spiking astrocyte neural network. *IEEE Trans. Neural Netw. Learn. Syst*. 30, 865–875. doi: 10.1109/TNNLS.2018.2854291

Liu, J., Sun, T., Luo, Y., Fu, Q., Cao, Y., Zhai, J., et al. (2018b). "Financial data forecasting using optimized echo state network," in *25th International Conference on Neural Information Processing (ICONIP)*, eds L. Cheng, A. C. S. Leung, and S. Ozawa (Cham: Springer), 138–149. doi: 10.1007/978-3-030-04221-9_13

Liu, Y.-J., Yu, M., Zhao, G., Song, J., Ge, Y., and Shi, Y. (2018c). Real–time movie–induced discrete emotion recognition from EEG signals. *IEEE Trans. Affect. Comput*. 9, 550–562. doi: 10.1109/TAFFC.2017.2660485

Luo, Y., Lu, Q., Liu, J., Fu, Q., Harkin, J., McDaid, L., et al. (2018). "Forest fire detection using spiking neural networks," in *Proceedings of the 15th ACM International Conference on Computing Frontiers* (New York, NY: ACM), 371–375. doi: 10.1145/3203217.3203231

Nie, D., Wang, X.-W., Shi, L.-C., and Lu, B.-L. (2011). "EEG-based emotion recognition during watching movies," in *2011 5th International IEEE/EMBS Conference on Neural Engineering* (Cancun: IEEE), 667–670. doi: 10.1109/NER.2011.5910636

Orhan, U., Hekim, M., and Ozer, M. (2011). EEG signals classification using the k-means clustering and a multilayer perceptron neural network model. *Expert Syst. Appl*. 38, 13475–13481. doi: 10.1016/j.eswa.2011.04.149

Petrantonakis, P. C., and Hadjileontiadis, L. J. (2010). Emotion recognition from EEG using higher order crossings. *IEEE Trans. Inform. Technol. Biomed*. 14, 186–197. doi: 10.1109/TITB.2009.2034649

Scherer, K. R. (1995). Expression of emotion in voice and music. *J. Voice* 9, 235–248. doi: 10.1016/S0892-1997(05)80231-0

Soroush, M. Z., Maghooli, K., Setarehdan, S. K., and Nasrabadi, A. M. (2019). A novel EEG-based approach to classify emotions through phase space dynamics. *Signal Image Video Process*. 13, 1149–1156. doi: 10.1007/s11760-019-01455-y

Subasi, A., and Gursoy, M. I. (2010). EEG signal classification using PCA, ICA, LDA and support vector machines. *Expert Syst. Appl*. 37, 8659–8666. doi: 10.1016/j.eswa.2010.06.065

Tabar, Y. R., and Halici, U. (2016). A novel deep learning approach for classification of EEG motor imagery signals. *J. Neural Eng*. 14, 90–100. doi: 10.1088/1741-2560/14/1/016003

Tripathi, S., Acharya, S., Sharma, R. D., Mittal, S., and Bhattacharya, S. (2017). "Using deep and convolutional neural networks for accurate emotion classification on deap dataset," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence* (Hawaiian, HI), 4746–4752.

Ullah, H., Uzair, M., Mahmood, A., Ullah, M., Khan, S. D., and Cheikh, F. A. (2019). Internal emotion classification using EEG signal with sparse discriminative ensemble. *IEEE Access* 7, 40144–40153. doi: 10.1109/ACCESS.2019.2904400

van den Broek, E. L. (2013). Ubiquitous emotion-aware computing. *Pers. Ubiquit. Comput*. 17, 53–67. doi: 10.1007/s00779-011-0479-9

Wang, X., Zhang, T., Xu, X., Chen, L., Xing, X., and Chen, C. L. P. (2018). "EEG emotion recognition using dynamical graph convolutional neural networks and broad learning system," in *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (Madrid: IEEE), 1240–1244. doi: 10.1109/BIBM.2018.8621147

Wang, Y., Qiu, S., Li, J., Ma, X., Liang, Z., Li, H., et al. (2019). "EEG-based emotion recognition with similarity learning network," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (Berlin: IEEE), 1209–1212. doi: 10.1109/EMBC.2019.8857499

Zhang, D., Yao, L., Zhang, X., Wang, S., Chen, W., and Boots, R. (2018). "Cascade and parallel convolutional recurrent neural networks on EEG-based intention recognition for brain computer interface," in *32nd AAAI Conference on Artificial Intelligence, AAAI 2018* (New Orleans, LA).

Zhang, J., Chen, M., Hu, S., Cao, Y., and Kozma, R. (2016). "PNN for EEG-based emotion recognition," in *2016 IEEE International Conference on Systems, Man, and Cybernetics* (Budapest), 2319–2323.

Zhang, J., Chen, M., Hu, S., Cao, Y., and Kozma, R. (2016a). "PNN for EEG-based emotion recognition," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (Budapest: IEEE), 002319-002323.

Zhang, J., Chen, M., Zhao, S., Hu, S., Shi, Z., and Cao, Y. (2016b). Relieff-based EEG sensor selection methods for emotion recognition. *Sensors* 16, 40144–40153. doi: 10.3390/s16101558

Zhang, Q., Chen, X., Zhan, Q., Yang, T., and Xia, S. (2017). Respiration-based emotion recognition with deep learning. *Comput. Ind*. 92–93, 84–90. doi: 10.1016/j.compind.2017.04.005

Zheng, W. (2017). Multichannel EEG-based emotion recognition via group sparse canonical correlation analysis. *IEEE Trans. Cogn. Dev. Syst*. 9, 281–290. doi: 10.1109/TCDS.2016.2587290

Zheng, W.-L., and Lu, B.-L. (2015). Investigating critical frequency bands and channels for EEG-based emotion recognition with deep neural networks. *IEEE Trans. Auton. Mental Dev*. 7, 162–175. doi: 10.1109/TAMD.2015.2431497

Zheng, W.-L., Zhu, J.-Y., Peng, Y., and Lu, B.-L. (2014). "EEG-based emotion classification using deep belief networks," in *2014 IEEE International Conference on Multimedia and Expo* (Chengdu: IEEE), 1–6. doi: 10.1109/ICME.2014.6890166

Zhou, Z. H., and Feng, J. (2017). "Deep forest: towards an alternative to deep neural networks," in *IJCAI International Joint Conference on Artificial Intelligence* (Melbourne, VIC), 3553–3559. doi: 10.24963/ijcai.2017/497

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Applied Mathematics and Statistics

ORIGINAL RESEARCH
published: 08 September 2020
doi: 10.3389/fams.2020.00041

Multi-Linear Population Analysis (MLPA) of LFP Data Using Tensor Decompositions

Justen Geddes[1], *Gaute T. Einevoll*[2,3,4], *Evrim Acar*[5] *and Alexander J. Stasik*[2,3]*

[1] Department of Mathematics, North Carolina State University, Raleigh, NC, United States, [2] Department of Physics, University of Oslo, Oslo, Norway, [3] Centre for Integrative Neuroplasticity, University of Oslo, Oslo, Norway, [4] Faculty of Science and Technology, Norwegian University of Life Sciences, Oslo, Norway, [5] Simula Metropolitan Center for Digital Engineering, Oslo, Norway

The local field potential (LFP) is the low frequency part of the extracellular electrical potential in the brain and reflects synaptic activity onto thousands of neurons around each recording contact. Nowadays, LFPs can be measured at several hundred locations simultaneously. The measured LFP is in general a superposition of contributions from many underlying neural populations which makes interpretation of LFP measurements in terms of the underlying neural activity challenging. Classical statistical analyses of LFPs rely on matrix decomposition-based methods, such as PCA (Principal Component Analysis) and ICA (Independent Component Analysis), which require additional constraints on spatial and/or temporal patterns of populations. In this work, we instead explore the multi-fold data structure of LFP recordings, e.g., multiple trials, multi-channel time series, arrange the signals as a higher-order tensor (i.e., multiway array), and study how a specific tensor decomposition approach, namely canonical polyadic (CP) decomposition, can be used to reveal the underlying neural populations. Essential for interpretation, the CP model provides uniqueness without imposing constraints on patterns of underlying populations. Here, we first define a neural network model and based on its dynamics, compute LFPs. We run multiple trials with this network, and LFPs are then analysed simultaneously using the CP model. More specifically, we design feed-forward population rate neuron models to match the structure of state-of-the-art, large-scale LFP simulations, but downscale them to allow easy inspection and interpretation. We demonstrate that our feed-forward model matches the mathematical structure assumed in the CP model, and CP successfully reveals temporal and spatial patterns as well as variations over trials of underlying populations when compared with the ground truth from the model. We also discuss the use of diagnostic approaches for CP to guide the analysis when there is no ground truth information. In comparison with classical methods, we discuss the advantages of using tensor decompositions for analyzing LFP recordings as well as their limitations.

Keywords: tensor decompositions, neuroscience, local field potential (LFP), population rate model, CANDECOMP/PARAFAC, independent component analysis (ICA), principal component analysis (PCA)

OPEN ACCESS

Edited by:
Axel Hutt,
Inria Nancy - Grand-Est Research Centre, France

Reviewed by:
Meysam Hashemi,
INSERM U1106 Institut de Neurosciences des Systèmes, France
Peter beim Graben,
Brandenburg University of Technology Cottbus-Senftenberg, Germany

****Correspondence:***
Alexander J. Stasik
a.j.stasik@fys.uio.no

Specialty section:
This article was submitted to Dynamical Systems, a section of the journal Frontiers in Applied Mathematics and Statistics

Received: *22 June 2020*
Accepted: *31 July 2020*
Published: *08 September 2020*

Citation:
Geddes J, Einevoll GT, Acar E and Stasik AJ (2020) Multi-Linear Population Analysis (MLPA) of LFP Data Using Tensor Decompositions. Front. Appl. Math. Stat. 6:41. doi: 10.3389/fams.2020.00041

1. INTRODUCTION

Most of what we know about the inner workings of the living brain has been learned from extracellular electrical recordings, that is, recordings of electrical potentials by sharp electrodes placed in the extracellular space between nerve cells. The high-frequency part of the signals, above a few hundred hertz, measures action potentials of neurons in the vicinity of the electrode contacts. The low-frequency part, the local field potential (LFP), is more difficult to interpret. While it is thought to mainly reflect processing of synaptic inputs by the neuronal dendrites by populations of pyramidal neurons, a simple rule-of-thumb interpretation in terms of the underlying neural activity like we have for spikes, is lacking [1, 2].

A traditional way to record LFPs in layered structures, such as hippocamapus or cortex is to record potentials with a linear multielectrode with many recording contacts crossing these laminarly organized brain structures. The resulting data typically have a two-way (matrix) structure (channels, time), and a standard analysis method has been to estimate the current-source density (CSD) [3]. The CSD measures the net volume density of electrical currents entering (sources) or leaving (sinks) the extracellular space. Several methods are available for CSD estimation, the standard CSD analysis assumes the neural activity to be constant in the horizontal directions in layered cortical or hippocampal structures [4], or other more recent methods like the iCSD [5] or the kCSD methods [6] which make different assumptions.

While the CSD is a more localized measure of neural activity than the LFP, it does not directly inform which neurons are active. An alternative is thus to try to decompose the measured LFPs into contributions from individual populations of neurons, for example, using matrix decomposition techniques like principal component analysis (PCA) [7] or independent component analysis (ICA) [8–10]. Here, data matrices are decomposed into outer products of pairs of vectors, and the vectors are interpreted as LFP contributions from individual neural populations. The decomposition of a matrix into outer products of vectors is not unique, and the methods involve additional assumptions like orthogonality (PCA) or statistical independence (ICA) of LFP contributions from the various populations. These mathematical assumptions cannot a priori be expected to be obeyed by neuronal populations in real brains. An alternative is to impose more physiological constraints in the decomposition [11]. An example is laminar population analysis (LPA) [12, 13] which, however, requires simultaneous recordings of action potentials and further physiological assumptions.

We here consider a new approach to LFP decomposition by considering three-way data (trial, channel, time) arranged as third-order tensors and performing a decomposition into outer products of triplets of vectors using an approach called CANDECOMP/PARAFAC (CP) [14, 15]. Unlike for the case with two-way data, this three-way decomposition is unique under mild conditions [16], and no strong assumptions, such as orthogonality or statistical independence on the components are needed. The underlying assumption in the CP model is that signals from each trial are a linear mixture of contributions from neural populations, and temporal and spatial signatures of neural populations stay the same across trials while each population's contribution to trials is scaled differently. Through the CP model, we assume that the LFP is *multi-linear*, thus linear in every argument (in our case tri-linear) since it assumes linearity in each mode. When matrix-based approaches, such as PCA and ICA are used on these signals from multiple trials by flattening the third-order LFP recording tensor, that structure cannot be maintained; therefore, additional assumptions, such as orthogonality are needed to ensure uniqueness.

Tensor decompositions are extensions of matrix decompositions, such as PCA to higher-order tensors (also referred to as multi-way arrays) and have proved useful in terms of finding underlying patterns in complex data sets in many domains including social network analysis, chemometrics, and signal processing [17–19]. As a result of its inherent uniqueness properties, among the tensor decomposition methods, the CP model has been quite popular since it can easily be interpreted. The CP model has also been successfully used in various neuroscience applications, e.g., the analysis of electroencephalography (EEG) signals [20], event related potential (ERP) estimation under the name topographic component analysis [21, 22], and more recently, capturing spatial, spectral, and temporal signatures of epileptic seizures [23, 24] and dynamics of learning [25].

In this study, we use the CP model to study LFP signals with the goal of disentangling individual neuronal populations. To the best of our knowledge, the CP model has not been used previously to analyze LFP recordings. In order to assess the performance of the CP model, we first create benchmarking data by simulating multi-channel LFP recordings across multiple trials based on a model of neuron populations. These simulated signals are arranged as a third-order tensor with modes: *trials*, *channels*, and *time*. Our numerical experiments demonstrate that the CP model can successfully reveal the underlying neuron populations by capturing their temporal and spatial signatures while more traditional ICA-based and PCA-based approaches fail to separate the populations. We also discuss advantages and limitations of the CP model in the presence of noise and different models of neuron populations.

2. METHODS

In the methods section we first describe the forward model used to compute the model-based benchmarking data. This forward modeling consists of two parts: (i) a population firing-rate model simulating neural dynamics in a multi-population feed-forward network model, and (ii) the computation of the local field potential (LFP) stemming from these neural dynamics. Second, we describe the tensor decomposition approach used in the inverse modeling to reveal the individual neuronal populations.

2.1. Forward Modeling of Benchmarking Data

2.1.1. Rate Model for Neuron Populations

A neuron is a cell that processes and transmits information. It receives inputs called action potentials via synapses from pre-synaptic neurons. This input will change the internal state of the neuron (e.g., its membrane potential), and the neuron may itself generate action potentials that is forwarded to other post-synaptic neurons. One option for modeling dynamics of neural networks is to explicitly model the train of action potentials of all neurons in the network. For large neural networks this not only becomes numerically cumbersome, the results also become difficult to interpret.

Firing-rate models thus offer an attractive alternative: if the number of neurons grows, further simplifications can be used. Given a subset of all neurons which have similar properties, receive very similar input, and project to similar groups of neurons, we can call them a *population*. Instead of taking every action potential into account, the population activity can be described by their instantaneous firing rate, that is, the average number of action potentials fired in a time window across the population. This presents a huge simplification since the relative timing of all action potentials are neglected, but gives a description of larger population dynamics. In many cases the temporal evolution of the firing rate can be modeled by means of ordinary differential equations (ODE). For an extensive discussion of rate models, see, e.g., Ermentrout and Terman [26] and Gerstner et al. [27]. A population rate model for I neuron populations can be described by

$$\boldsymbol{\tau} \frac{\mathrm{d}\mathbf{r}(t)}{\mathrm{d}t} = -\mathbf{r}(t) + \mathcal{F}\big(\mathbf{W}\mathbf{r}(t) + \mu(t)\big) \tag{1}$$

where $\mathbf{r}(t) \in \mathbb{R}^I_{\geq 0}$ is the vector of instantaneous firing rates of I populations (thus restricted to non-negativity), the vector $\tau \in \mathbb{R}^I_{\geq 0}$ contains the time constants for each population, $\mathbf{W} \in \mathbb{R}^{I\times I}$ is the coupling matrix describing the synaptic strength between populations, $\mu(t)$ is the instantaneous external stimulus received by each population and $\mathcal{F}$ is the column-wise response function of each population to input (both from other neuron population as from external sources). The response function can in principle take any form and in general adds a non-linearity to the system. For simplicity we assume if not stated otherwise that $\mathcal{F}$ is the identity function making the rate model fully linear.

The neural network is mainly characterized by the connection matrix $\mathbf{W}$. If several populations affect each other only in a sequential way, we call these populations a feed-forward network. If several populations drive each other in a loop, we call those recurrent. If two neural populations are completely independent, meaning that they do not affect each other directly or indirectly via intermediate populations or do not share a common input, we can regard them as separate networks. One can further distinguish between *excitatory* and *inhibitory* neuron populations. *Excitatory* populations increase firing rate of post-synaptic populations, while *inhibitory* populations inhibit the firing of post-synaptic populations. Thus, $\mathbf{W}_{i,j} > 0$ for all j if population i is *excitatory* and likewise $\mathbf{W}_{i,j} < 0$ for i being an *inhibitory* population.

By assuming initial conditions $\mathbf{r}(0)$ and external stimuli $\mu(t)$, we fully define the ODE problem, which is solved using the `ode45` package from `MatLab` [28]. We discretize dimensionless time between 0 and 1 in 1,000 steps and solve Equation (1) in this range. In this paper, we use a boxcar function

$$\mu_i(t) = c_i\Big(\Theta(t - t_i^{\text{start}}) - \Theta(t - t_i^{\text{end}})\Big) \tag{2}$$

as stimulus with c_i as the stimulus magnitude for each population i and t_i^{start} and t_i^{end} as the respective on and off-set for each population. Solving the rate model takes few seconds on a standard laptop and can thus be performed easily multiple times for different values of $\mathbf{W}$ and stimuli $\mu(t)$ to explore the model. Our choice of a linear model is a simplification which allows easier analysis, but has limitations, as, for example, it allows for (unphysical) negative firing rates.

2.1.2. Computation of LFP Signals

The local field potential (LFP) is the low-frequency part of extracellular potential, and *in vivo* it is generated from the superposition of the extracellular potential generated by many neurons [1]. The LFP Φ is a three-dimensional physical scalar field which can be continuously measured in time at any position, thus $\Phi(\mathbf{x}, t)$. The LFP is measured by electrodes recording the signal at one or, more typically, many discrete locations, typically called channels. Thus, the recording becomes a vector of time series. In this discrete case, each LFP recording can thus be written as a matrix $\Phi \in \mathbb{R}^{n_{\text{channels}} \times n_{\text{time steps}}}$ where the rows are the different channels and, the columns are the different discrete time points.

In this study, we mimic a situation where the LFP is recorded by a linear multielectrode at many positions through the depth of cortex. In this set-up, the recorded LFP will typically contain contributions from multiple populations of neurons with their somas (cell bodies) positioned at different depths (see, e.g., [12]). The multi-electrode probe records from multiple depth locations at the same time, and the spatial dependence $\mathbf{x}$ thus reduces to the scalar depth x, which simplifies further discussion. We do not aim to simulate any specific cortical system and instead keep things general. Therefore, we regard our model as a toy-model, which still incorporate key features of real LFPs. A method to efficiently compute LFPs is the so-called kernel method [29, 30].

The kernel method consists of several steps: (I) Biophysical, multi-compartment neuron models are used to generate a large number of neurons which then represent a set neuron populations. (II) All neurons in a single pre-synaptic population i are forced to emit an action potential at the same point in time. This mimics a δ-shaped firing rate. (III) The resulting LFP generated by each post-synaptic population j is stored separately for multiple locations, simulating a virtual probe with multiple channels. This provides a kernel H_{ij}, the expected LFP if population i fires and projects to population j. (IV) Steps II and III are repeated for each pre-synaptic population i, providing a full set of kernels H_{ij}. If $\mathbf{W}$ is constant, one can define the population kernel as the sum over all post-synaptic kernels $H_i = \sum_j H_{ij}$. However, it is advantageous to maintain all $H_{i,j}$ so that the contribution of individual populations to the total LFP can easily be computed.

The kernels $H_i(x, t)$ thus depend on space x and time t, in the discrete case they can be represented as a matrix as well with dimensions channels and time. The LFP contribution of the ith population $\Phi_i(x,t)$ is computed by a temporal convolution of the corresponding kernel $H_i(x,t)$ with the respective population firing rate $r_i(t)$:

$$\Phi_i(x,t) = (r_i * H_i)(x,t). \tag{3}$$

Since the electrical potential is additive, the total LFP is given by summation over all populations

$$\Phi(x,t) = \sum_{i=1}^{I} (r_i * H_i)(x,t) \tag{4}$$

where $*$ denotes the convolution operation in time and $r_i(t)$ is the firing rate of population i. In the discrete case, the result of the convolution is thus a matrix with the dimensions channel and time. This method has been developed as a part of the HybridLFPy software (see [29]). The kernels used in this work have been derived for a cat cortex model [31].

For each kernel H_i, there exists a rank-one approximation which allows to write the kernel as the outer product of a temporal k_i and a spatial function c_i:

$$H_i(x,t) \approx c_i(x) \cdot k_i^{\mathsf{T}}(t) \tag{5}$$

where k^{T} denotes the transposed vector. Since the projection of the neural population to the various channel locations is determined by their static morphology and the propagation of the electrical signal through neurons is fast, there is little delay at different locations. Therefore, the rank-one approximation is very accurate (variance explained above 95% for all studied cortical kernels, see **Appendix**). Due to this factorization of the kernel into spatial and temporal parts, the resulting LFP can also be factorized in a spatial and temporal part since the firing rate is only convolved with the kernel in the temporal dimension:

$$\Phi(x,t) \approx \sum_{i=1}^{I} c_i(x) \cdot \left(r_i * k_i\right)^{\mathsf{T}}(t), \tag{6}$$

where $*$ denotes the convolution operator. If we discretize, spatial and temporal factors of the kernel become vectors: $c_i(x) \rightarrow \mathbf{c}_i$, $k_i(t) \rightarrow \mathbf{k}_i$, and the kernel itself turns into a matrix $H_i(x,t) \rightarrow \mathbf{H}_i$. Since the population firing rate $\mathbf{r}(t)$ is also solved numerically, the continuous solution turns into a vector $\mathbf{r}_i(t) \rightarrow \mathbf{r}_i$. Thus, the LFP (Equation 6) can be written as a matrix $\Phi \in \mathbb{R}^{M\times N}$:

$$\Phi \approx \sum_{i=1}^{I} \mathbf{c}_i \cdot (\mathbf{r}_i * \mathbf{k}_i)^{\mathsf{T}} \tag{7}$$

with M as the number of channels and N as the number of time points. By repeating the simulation of Φ multiple (L) times, e.g., for different connection matrices $\mathbf{W}$ mimicking changes of the network over time, and stack the resulting LFPs, we get a three-way tensor $\mathcal{X} \in \mathbb{R}^{L\times M\times N}$ with modes: *trials*, *channels*, and *time*.

2.2. CANDECOMP/PARAFAC (CP) Tensor Model Used in Inverse Modeling

The CP model, also known as the canonical polyadic decomposition [32], is one of the most popular tensor decomposition approaches. Here, we use the CP model to analyze the three-way LFP tensor with modes: *trials*, *channels*, and *time*, and reveal temporal and spatial signatures of underlying neuron populations. For a third-order tensor (three-way array) $\mathcal{X} \in \mathbb{R}^{L\times M\times N}$, an R-component CP model approximates the tensor as the sum of R rank-one third-order tensors, as follows:

$$\mathcal{X} \approx \sum_{r=1}^{R} \mathbf{s}_r \circ \mathbf{c}_r \circ \mathbf{t}_r \tag{8}$$

where $\circ$ denotes the vector outer product following the notation in Kolda and Bader [17], $\mathbf{S} = [\mathbf{s}_1 \dots \mathbf{s}_R] \in \mathbb{R}^{L\times R}, \mathbf{C} = [\mathbf{c}_1 \dots \mathbf{c}_R] \in \mathbb{R}^{M\times R}, \mathbf{T} = [\mathbf{t}_1 \dots \mathbf{t}_R] \in \mathbb{R}^{N\times R}$ correspond to factor matrices in *trials*, *channels*, and *time* mode, respectively. Note that the outer product of three vectors is a third-order rank-one tensor, i.e., $\mathcal{Z} = \mathbf{u} \circ \mathbf{v} \circ \mathbf{w} \iff z_{ijk} = u_i v_j w_k$. The CP model is unique under mild conditions up to permutation and scaling [16, 17]; in other words, the same rank-one components, i.e., $\mathbf{s}_r \circ \mathbf{c}_r \circ \mathbf{t}_r$, for $r = 1, \dots, R$, are revealed by the model at the solution but the order of rank-one components is arbitrary and within each rank-one component, factor vectors have a scaling ambiguity, e.g., $2\mathbf{s}_r \circ {}^1\!/_2\mathbf{c}_r \circ \mathbf{t}_r$. By normalizing columns of the factor matrices and introducing an additional scalar, λ_r, for each rank-one component, we can rewrite Equation (8) as follows:

$$\mathcal{X} \approx \sum_{r=1}^{R} \lambda_r \cdot \bar{\mathbf{s}}_r \circ \bar{\mathbf{c}}_r \circ \bar{\mathbf{t}}_r \tag{9}$$

with $\bar{\mathbf{s}}_r$, $\bar{\mathbf{c}}_r$, and $\bar{\mathbf{t}}_r$ as unit-norm factor vectors.

The CP model is considered to be one way of extending Singular Value Decomposition (SVD) to higher-order data sets. As a result of its uniqueness properties, the CP model has the benefit of revealing underlying patterns without imposing strict and potentially unrealistic constraints, such as orthogonality or statistical independence as in the case of matrix decomposition-based approaches, such as PCA and ICA. When an R-component CP model is used to analyze an LFP tensor, it extracts R rank-one components.

Our motivation for using the CP model to analyze LFP tensors is as follows: If the population rate model is fully linear ($\mathcal{F}$ in Equation 1 is a linear function), then the solutions of equation (Equation 1) are only linearly dependent of the connection matrix $\mathbf{W}$ and/or the external stimulus $\boldsymbol{\mu}(t)$. If trials with variations of $\mathbf{W}$ and/or $\boldsymbol{\mu}(t)$ are performed, the resulting firing rates will thus only depend linearly on $\mathbf{W}$ and $\boldsymbol{\mu}(t)$. Furthermore, if we also assume that kernels are rank-one matrices, fulfilling Equation (5), a tensor consisting of multiple trials of multi-electrode LFP recordings will have an underlying CP structure (see Equations 7 and 8).

We claim that these extracted tensor components correspond to spatial and temporal signatures of neuron populations, i.e.,

$\mathbf{c}_r$ and $\mathbf{t}_r$ correspond to spatial and temporal signatures of the rth population (see Equation 7), due to the fact that the LFP of each population can be written as a bi-linear form and given the uniqueness of CP. Each component in the CP thus corresponds to a neural population.

Given the number of components R, in order to fit a CP model to a third-order tensor $\mathcal{X}$, we solve the following optimization problem:

$$\min_{\mathbf{S},\mathbf{C},\mathbf{T}} \left\| \mathcal{X} - \sum_{r=1}^{R} \mathbf{s}_r \circ \mathbf{c}_r \circ \mathbf{t}_r \right\|_F^2, \tag{10}$$

where $\|\cdot\|_F$ denotes the Frobenius norm for tensors, i.e., $\|\mathcal{X}\|_F^2 = \sum_{l=1}^{L}\sum_{m=1}^{M}\sum_{n=1}^{N} x_{lmn}^2$. There are various algorithmic approaches for solving this problem, such as alternating least squares (ALS) and all-at-once optimization methods. In this paper, we use a gradient-based all-at-once approach based on CP-OPT [33], that solves the problem for all factor matrices, **S**, **C**, **T**, simultaneously.

2.2.1. Diagnostics Tools

Determining the number of CP components (R) is a difficult problem. For the exact case, where there is an equality in Equation (8), R corresponds to the tensor rank and determining the rank of a tensor is NP-hard [17, 34]. In practice, this challenging problem has been mitigated using various diagnostic tools, such as core consistency diagnostic [35], split-half and residual analysis [36]. In this paper, we use the core consistency values and model fit to determine the number of components while modeling an LFP tensor using a CP model.

Core Consistency indicates whether an R-component CP model is an appropriate model for the data, and is defined as follows [35]:

$$\text{Core Consistency} = 100 \times \left(1 - \frac{\sum_{i=1}^{R}\sum_{j=1}^{R}\sum_{k=1}^{R}(g_{ijk} - t_{ijk})^2}{R}\right),$$

where $\mathcal{G}$ (of size $R \times R \times R$) is the estimated core array for a Tucker model [37] given the factor matrices of the R-component CP model, and $\mathcal{T}$ (of size $R \times R \times R$) is a super-diagonal core array for the CP model with non-zero entries only on the super-diagonal. Tucker is a more flexible tensor model; therefore, its core $\mathcal{G}$ can be a full core array. If the R-component CP model is a valid model, then off super-diagonal elements of $\mathcal{G}$ will be close to zero giving high core consistency values close to 100%. Low core consistency values indicate potentially an invalid CP model.

Model Fit is used to understand how much of the data is explained by the model. Given a tensor $\mathcal{X}$ and its CP approximation $\hat{\mathcal{X}} = \sum_{r=1}^{R} \mathbf{s}_r \circ \mathbf{c}_r \circ \mathbf{t}_r$, the fit can be defined as follows:

$$\text{Fit} = 100 \times \left(1 - \frac{\left\|\mathcal{X} - \hat{\mathcal{X}}\right\|_F^2}{\|\mathcal{X}\|_F^2}\right)$$

When the number of components extracted from the data increases, the fit also increases. However, the increase in additional complexity due to additional components should often be justified by an increase in model fit. In other words, we need to assess if we gain a significant increase in fit by extracting more components from the data.

To find the optimal number of components R, we examine model fit and core consistency values across different number of components. While heuristic in nature, these diagnostics are able to be employed effectively in this study.

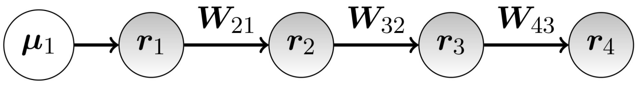


FIGURE 1 | Sketch of a four-population feed-forward neural model defined via connection matrix in Equation (11) .

3. NUMERICAL EXPERIMENTS

As an example, we assume the simple model of four neuron populations which are sequentially connected (see **Figure 1**). Population 1 receives the same stimulus in every trial l while the other populations only receive indirect input via projections from other populations ($\boldsymbol{\mu}(t) = (\mu_1(t), 0, 0, 0)^{\mathsf{T}}$). We further assume that the network is purely excitatory ($0 \leq W_{ij}$). We model multiple trials by assuming that due to plasticity, W_{21}, W_{32}, and W_{43} will change between trials. This mimics the situation where a subject is exposed to the same stimulus at different times and the neural network adapts and changes its connectivity due to learning. This plasticity is not part of our model, but externally enforced. We will show that the changes in weights across trials can be recovered using the CP model and thus will allow to study the learning process.

This network has the following sparse connectivity matrix:

$$\mathbf{W} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ W_{21} & 0 & 0 & 0 \\ 0 & W_{32} & 0 & 0 \\ 0 & 0 & W_{43} & 0 \end{pmatrix} \tag{11}$$

To indicate different trials, we use superscripts, e.g., $\mathbf{W}^l$ is the connection matrix for the lth trial. We define the experimental population strength $\mathbf{s}_r^l$ as the strength of the rth population in the lth trial. For the four population feed-forward model, we thus have the relation

$$\begin{aligned} \mathbf{s}_1^l &\propto \text{const} \\ \mathbf{s}_2^l &\propto W_{21}^l \\ \mathbf{s}_3^l &\propto W_{32}^l W_{21}^l \\ \mathbf{s}_4^l &\propto W_{43}^l W_{32}^l W_{21}^l. \end{aligned} \tag{12}$$

Since the convolution with the temporal factor of the kernel $\mathbf{k}_r$ (see Equation 7) is also a linear operation and the channel factor $\mathbf{c}_r$ does not change between trials, we expect that when multi-trial multi-electrode LFP signals from this model are analyzed using an R-component CP model, the analysis will reveal the following:

- The trial population strength matrix $(\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_R)$
- The channel factor matrix $(\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_R)$
- The temporal factor matrix $(\mathbf{r}_1 * \mathbf{k}_1, \mathbf{r}_2 * \mathbf{k}_2, \ldots, \mathbf{r}_R * \mathbf{k}_R)$

Note that it is only possible to recover (1) the experimental population strength $\mathbf{s}_r^l$ (see Equation 12), not the connection matrix entries $\mathbf{W}$, and (2) the time-convolved temporal components $(\mathbf{r}_r * \mathbf{k}_r = \mathbf{t}_r)$ instead of the raw time traces $\mathbf{r}_r$. In the following, we will refer to the time-convolved components simply as the time components. To perform numerical simulations, we impose learning by a controlled change of $\mathbf{W}$. Over a set of $L = 30$ trials, we modulate W_{21} with a sinusoidal and an offset, such that over the trials, it performs a quarter period ($W_{21}^l \propto \sin(\pi/2 \cdot l/L) + \text{const}$). W_{32} increases linearly ($W_{32}^l \propto \alpha \cdot l/L + \text{const}$) and W_{43} decreases linearly ($W_{43}^l \propto -\alpha \cdot l/L + \text{const}$. We arbitrary set $\boldsymbol{\tau} = (0.1, 0.3, 0.3, 0.2))^{\top}$ and stimulate with a boxcar (Equation 2) between 0 and 0.2 ($t^{\text{start}} = 0$, $t^{\text{end}} = 0.2$). Here, we simulate multi-electrode LFP recordings across multiple trials based on a simple population rate model, and arrange the data as an LFP tensor with modes: *trials*, *channels*, and *time*. We then analyze the LFP tensor using an R-component CP model to disentangle R populations by estimating their spatial signatures $(\mathbf{c}_r)$, temporal signatures $(\mathbf{t}_r)$ and relative contributions to trials $(\mathbf{s}_r)$, for $r = 1, \ldots, R$.

In order to demonstrate the performance of the CP model in terms of analyzing LFP tensors, we have constructed the following tensors (all in the form of 30 *trials* × 16 *channels* × 1,000 *time points*):

(*i*) *A noise-free data set using full kernels:* An LFP data set simulated by using the full kernels (Equation 4). No noise was added to the simulation.

(*ii*) *A noise-free simulation using rank-one approximation of kernels:* An LFP data set was simulated using the rank-one approximation of the kernels for each population (Equation 5). Let $\mathbf{H}_i = \mathbf{U}\Sigma\mathbf{V}^{\top} = \sum_{r=1}^{R} \sigma_r \mathbf{u}_r \mathbf{v}_r^{\top}$ be the SVD of kernel $\mathbf{H}_i \in \mathbb{R}^{M \times N}$ of rank R for population i. Its rank-one approximation is $\hat{\mathbf{H}}_i = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^{\top}$. The rank-one approximation of each selected kernel explains over 95% of the kernel matrix. No noise was added to the simulation.

(*iii*) *Noisy LFP with rank-one approximation of kernels:* LFP data set was simulated using the rank-one approximation of kernels and then added noise as follows: Let $\mathcal{X}$ denote the LFP tensor that is constructed using the rank-one kernel approximations. The noisy LFP tensor with noise level α is given by $\mathcal{X}_\alpha = \mathcal{X} + \alpha \frac{\|\mathcal{X}\|_F}{\|\mathcal{N}\|_F} \mathcal{N}$ where $\mathcal{N}$ is a tensor with entries randomly drawn from the standard normal distribution.

Constructed LFP tensors are then analyzed using CP with different number of components. We have also compared the performance of the CP model with ICA and PCA.

3.1. Implementation Details

For fitting the CP model, `cp_opt` from the Tensor Toolbox [38] using the non-linear conjugate gradient (NCG) algorithm, as implemented in the Poblano Toolbox [39] is used. Multiple initializations are used to fit each R-component CP model, and the solution with the best function value is reported. For computing core consistency values, we use the `corcond` function from the Nway Toolbox [40].

For ICA, we use two different algorithms: (i) FastICA [41] exploiting non-Gaussianity of the underlying sources, and (ii) the ERBM (entropy-rate bound minimization) [42] algorithm that takes into account both higher-order statistics and sample dependence to find the underlying sources. For ICA algorithms, again multiple initializations are used with R components and among all the runs, we report the one that matches the true factors best.

3.2. Performance Evaluation

In order to assess the performance of the CP model, we quantify the similarity between CP factors and true signatures of neuron populations in *trials*, *channels*, and *time* modes. We use the Factor Match Score (FMS) as the similarity measure defined as follows [33, 43]:

$$\text{FMS} = \frac{1}{R} \sum_{r=1}^{R} \frac{|\mathbf{s}_r^{\top} \hat{\mathbf{s}}_r|}{\|\mathbf{s}_r\| \|\hat{\mathbf{s}}_r\|} \times \frac{|\mathbf{c}_r^{\top} \hat{\mathbf{c}}_r|}{\|\mathbf{c}_r\| \|\hat{\mathbf{c}}_r\|} \times \frac{|\mathbf{t}_r^{\top} \hat{\mathbf{t}}_r|}{\|\mathbf{t}_r\| \|\hat{\mathbf{t}}_r\|}, \qquad (13)$$

where $\hat{\mathbf{s}}_r, \hat{\mathbf{c}}_r, \hat{\mathbf{t}}_r$ for $r = 1, \ldots, R$ denote the estimated components by the CP model while $\mathbf{s}_r, \mathbf{c}_r, \mathbf{t}_r$ for $r = 1, \ldots, R$ denote the true components, i.e., simulated *trials*, *channels*, and *time* mode factors.

3.3. Results

Using numerical experiments, we demonstrate that the CP model can successfully extract components revealing each population. The model succeeds in revealing the populations from the noise-free tensor constructed using full kernels and also from the noise-free tensor constructed using rank-one approximation of kernels. In the presence of noise, each population can still be unraveled while their signatures are distorted by the noise. For the first type of tensor constructed using full kernels, the 4-component CP model can successfully capture the true factors as shown in **Figure 2**. Note that even though the tensor is constructed with full kernels, the true components in *channels* and *time* modes are assumed to be the leading singular vectors of each kernel matrix under the assumption that kernels have a rank-one structure with noise. Under this assumption, when true components and CP components are compared quantitatively, the FMS is 0.9965 indicating the accurate recovery of true patterns using the CP model. Here, the optimal number of components is equivalent to the number of populations that contribute to the data; therefore $R = 4$. We have also analyzed the data using CP with different number of components. **Figure 3** shows how core consistency and model fit change with R. Since we expect high core consistency values for valid CP models, we increase the number of components until we see a drop in core consistency. Both $R = 4$ and $R = 5$ potentially look valid models; however, the model fit is already 100% for $R = 4$ and becomes flat after $R = 4$ indicating that the 4-component CP model is the right choice. Thus, we are able to identify the correct number of neural

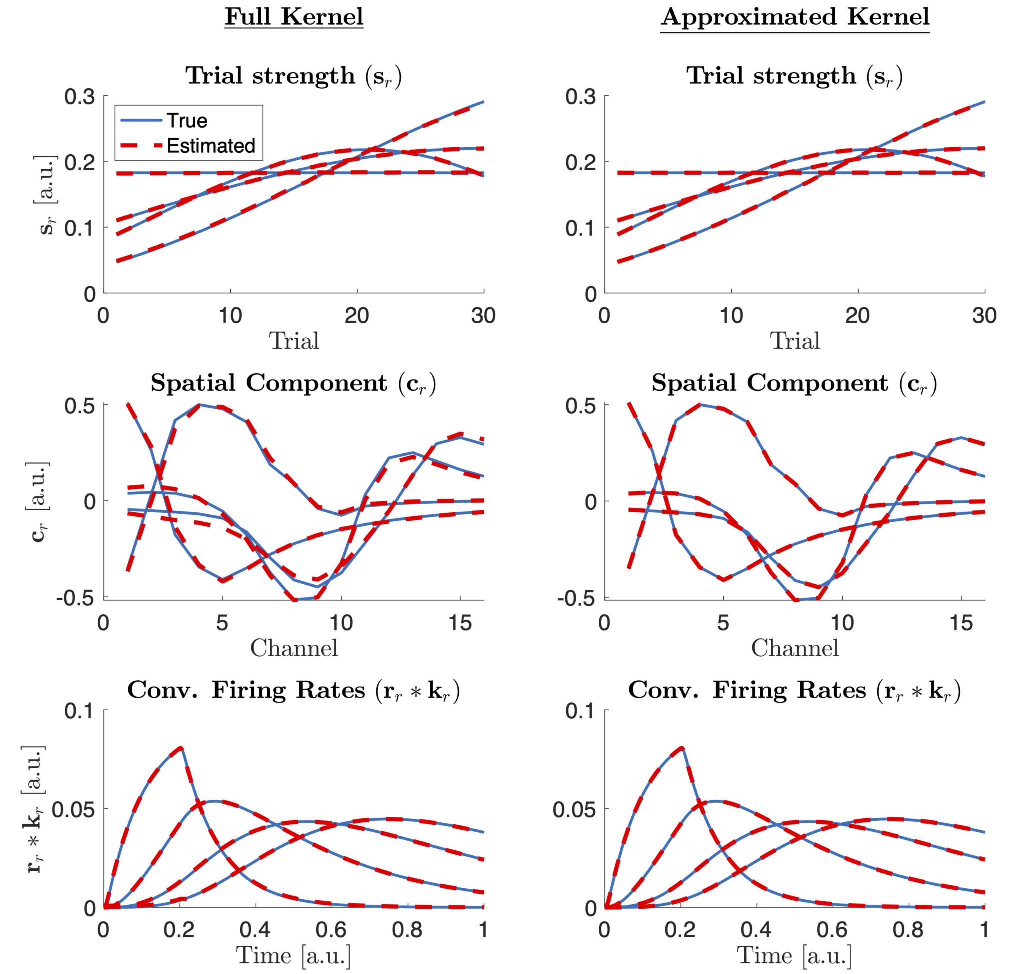


FIGURE 2 | True (simulated) components (blue solid line) vs. the factor vectors extracted by a 4-component CP model (red dashed) in *trials*, *channels*, and *time* modes from a tensor constructed via full kernels (left column) and rank one approximated kernels (right column). Note also that what is extracted by the CP model is $\hat{\mathbf{t}}_r$ which is then compared with $\mathbf{r}_r * \mathbf{k}_r$ from the benchmarking data.

populations based on the diagnostic tools used for determining the number of CP components.

In the second case, where a tensor is constructed via the best rank-one approximation of each kernel, the 4-component CP model can again unravel the four populations and their corresponding signatures in *trials*, *channels*, and *time* modes as shown in **Figure 2**. Core consistency and model fit values shown in **Figure 3** for different number of components and, in this case, indicate that $R = 4$ is the true number of components since the core consistency significantly drops after $R = 4$. Again, the true number of components could be identified using diagnostic tools.

When noise is added to the tensor constructed using rank-one approximated kernels, the CP model (with $R = 4$ components) can still reveal the true signatures of the populations; however, as we increase the noise level, we observe that factors become distorted as shown in **Figure 4**. **Table 1** shows that FMS values are still high for different levels of noise, and the model fit decreases as a result of the noise, as expected. Core consistency values, however, are much lower and therefore indicate noisy components.

3.4. Comparisons With ICA and PCA

Since ICA and PCA are more traditional approaches to analyze multi-channel electro-physiology data, we compare the performance of CP with both in terms of how well they recover the true temporal components. For ICA and PCA, we unfold the third-order LFP tensor in the *time* mode and arrange the data as a *trials-channels* by *time* matrix. In our comparisons, we use two different ICA algorithms: The FastICA algorithm [44] and the ERBM algorithm [42]. We study the noise-free case, analyze the LFP tensor using the correct number of underlying populations with CP, ICA ($R = 4$), and PCA, and compare the estimated sources in the *time* dimension in **Figure 5**. FastICA only finds two independent components, even if the correct larger number is given. The two reconstructed components do not mimic the actual firing rate components.

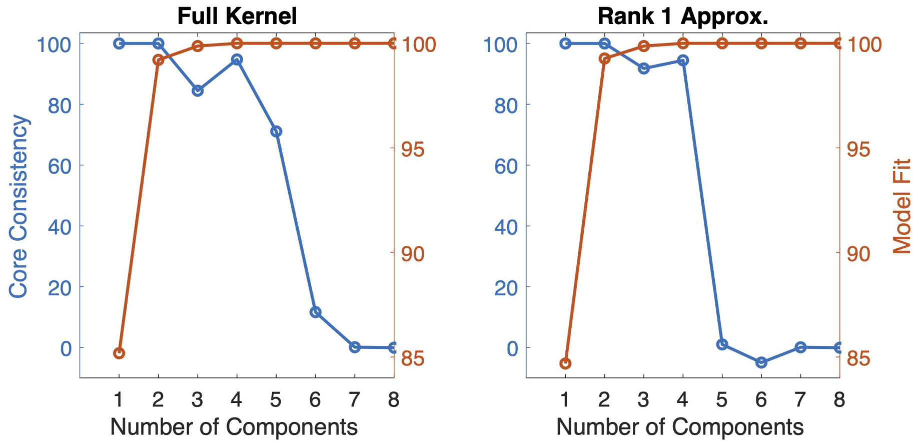


FIGURE 3 | Core consistency and model fit for CP models with $R = 1, 2, \ldots, 8$ components for tensors constructed with full kernels and the rank-one approximations of kernels.

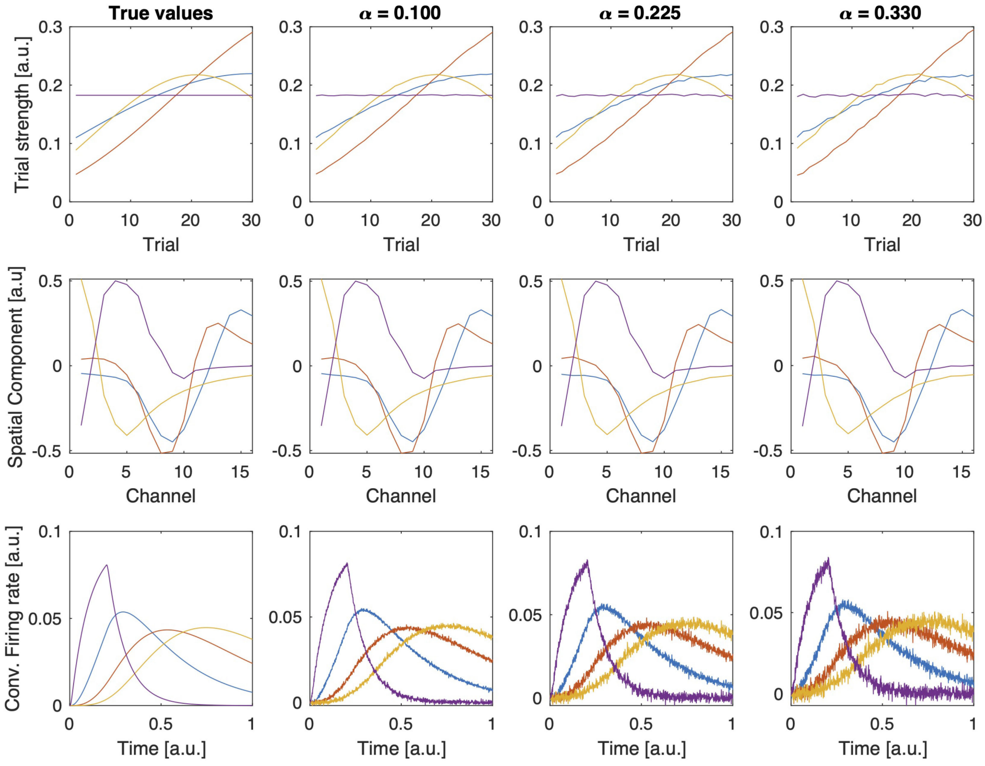


FIGURE 4 | Factor vectors in each mode of a 4-component CP model of tensors with different noise levels (α).

The ERBM algorithm was able to reconstruct two components which are similar to the ground truth, but the other two components do not match the ground truth. The unfolded noise-free tensor is of rank four, and when all four components were taken into account, PCA was able to recover the first component to some extent, but failed in the other components. The CP model, on the other hand, was able to recover all four components with high accuracy. This is a clear example that illustrates the advantage of using CP over classical matrix methods. Furthermore, the CP model also allows to estimate the components in *trials* and *channels* modes, whereas matrix-based methods like ICA or PCA require an unfolding of the tensor, therefore, failing to estimate components in all modes simultaneously.

3.4.1. Further Interpretation

We have shown that CP is able to blindly recover the location of each population ($\mathbf{c}_r$), but can only recover the population firing rate convolved with the temporal part of the corresponding kernel ($\mathbf{r}_r * \mathbf{k}_r$). There is no method available which blindly can deconvolve the kernel and firing rate. The potential kernel $\mathbf{k}_r$ can be constrained by values found in experimental studies. Performing standard deconvolution algorithms [45] can then be used to further estimate the actual population firing rate. Another limitation of the blind CP approach is that it can only recover the trial population strength $\mathbf{s}_r$, but not the underlying connectivity matrix $\mathbf{W}$ due to the relation in Equation (12). If one assumes a feed-forward network, a relation as described in Equation (12) can be assumed, but the order of components is arbitrary by CP. Inspection of the firing rate components (see for example **Figure 4**) can reveal a causal connection (purple → blue → red → yellow) and identify the order of components. Then, the elements of $\mathbf{W}$ can be recovered by the simple iteration $W^l_{r+1,r} = \mathbf{s}^l_{r+1}/\mathbf{s}^l_r$. Also, if two neuron populations have identical synaptic projection patterns, their LFP kernels will be also identical, making it impossible to distinguish them. Therefore, clearly identifying the spatial location of population LFPs relies on the assumption that their projection patterns are different.

TABLE 1 | FMS, core consistency, and model fit for 4-component CP models of noisy LFP tensors.

Noise level (α)	FMS	Core consistency (%)	Model fit (%)
0.100	0.9997	38	99
0.225	0.9985	32	95
0.330	0.9967	21	90

3.5. Application to a Non-linear Model

As stated in section 2.2, the assumption that a tensor consisting of multiple trials of multi-electrode LFP recordings will have an underlying CP structure relies on the assumption that $\mathcal{F}$ (Equation 1) is linear. However, in general $\mathcal{F}$ will be non-linear. To study the effects of introducing non-linearity, we use a hyperbolic tangent function as suggested by beim Graben and Kurths [46] instead of the identity function. To study the gradual increase of non-linearity, our function is set to be in the form of

$$\mathcal{F}(x) = \frac{1}{\beta}\tanh\big(\beta(x-a)\big) + b$$

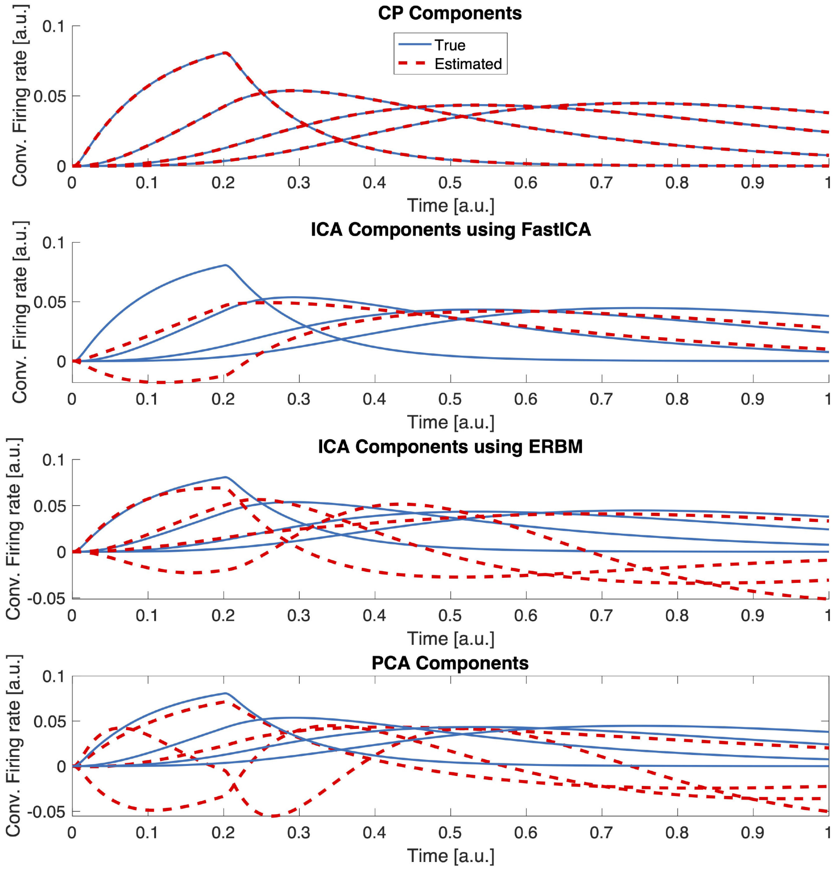


FIGURE 5 | Comparing CP, FastICA, ERBM, and PCA estimates of temporal components. Data was generated without additional noise and the correct number of components was provided to the algorithms.

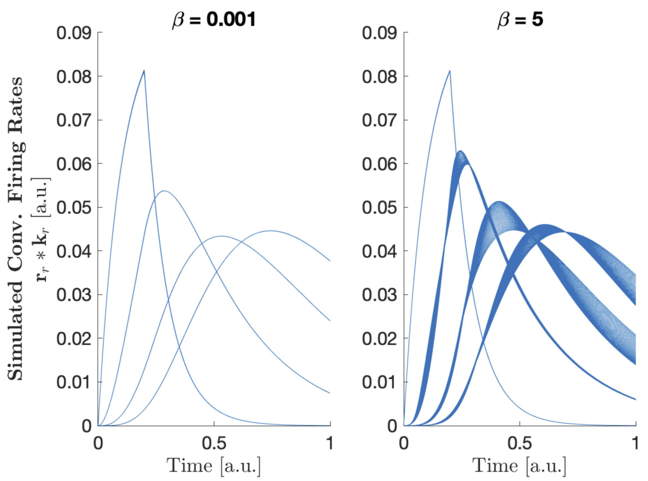


FIGURE 6 | Normalized convolved firing rates from all trials with $\mathcal{F}(x) = \frac{1}{\beta}\tanh\left(\beta(x-a)\right) + b$ and $\beta = 0.001$ (left) and $\beta = 5$ (right).

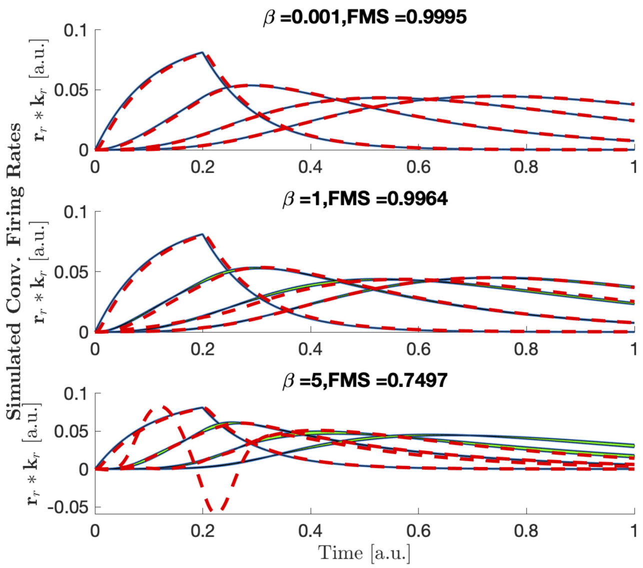


FIGURE 7 | The mean ± 3 standard deviations of each population's convolved firing rate across trials (blue) with area between the two curves shaded in green and CP reconstruction plotted as dashed red lines for increasing amounts of non-linearity. Larger green regions represent larger shape changes in the normalized firing rates.

with $a = 0.5$ so that the inflection is within the range of inputs, and $b = -\frac{1}{\beta}\tanh(-\beta \cdot a)$ to ensure that $\mathcal{F}(0) = 0$. In this form, the parameter β can be thought to represent the strength of the non-linearity of the system. For small values of β, $\mathcal{F}$ is approximately the identity function. As the value of β increases $\mathcal{F}$ becomes more non-linear. Effects of the parameter β on the convolved firing rates can be seen in **Figure 6**. Results of CP decomposition of the LFP tensor generated with increasing non-linearity (β) can be seen in **Figure 7** with associated factor match scores. Factor match scores are computed assuming that the mean of the firing rates of each population across trials is the ground truth. We observe FMS of 0.9995, 0.9964, and 0.7497 for $\beta = 0.001, 1,$ and 5, respectively. Note that with a weak non-linearity, CP is able to recover components that are close to the mean of the trials, thus the non-linearity can be regarded as a small perturbation. In this case CP is stable and can produce reasonable results. If the non-linearity becomes too strong, CP will, as expected, eventually fail, as shown in **Figure 7**. Thus, CP will provide a linear approximation of the data.

Also note that CP without non-negativity constraints reveal temporal components with negative values. In the Appendix (**Figure A3**), we also show the temporal factors obtained using a CP model with non-negativity constraints in the *time* and *trial* mode, for the different strength of non-linearity. We observe that for small non-linearities, where CP gives reasonable (and already non-negative) components, non-negativity constraint has almost no effect. In cases of strong non-linearity where CP fails, the constraint results in very different results.

4. DISCUSSION

4.1. Summary of Findings

In this paper we have applied the CP tensor decomposition approach to disentangle different neural populations in LFP recordings from multiple trials. The idea behind the CP model is that signals from each trial are a mixture of signals from several neural populations with specific temporal and spatial patterns. While these population-specific patterns stay the same across trials, contributions of the populations are scaled differently from one trial to another. By jointly analyzing signals from multiple trials, the CP model can uniquely reveal the underlying neural populations and capture population-specific temporal and spatial signatures without imposing *ad hoc* constraints on those patterns like what is required for using PCA and ICA. Physiologically, we have made several assumptions. We first assume that if a neural population receives the same input but with different magnitude in multiple trials, than its firing will also be the same, but with a different magnitude proportional to the input magnitude, thus we assume linear response of the population. Second, we assume that the location, and thus the recording channels a certain population projects to, is constant over time and also does not change between trials. Last, we assume that our input dominates the network activity and background processes ongoing in the network can be regarded as noise.

We used a linear feed-forward firing-rate toy model with four populations in this study. The LFP was computed by means of a kernel method, which allowed simple and fast generation of LFP given the population activity of neurons in the model. By repeating the model with different synaptic weights and the same stimulus, we generated a multi-trial, multi-channel time series tensor of LFP recordings. For this toy-model, we found that CP works very well and is able to recover the temporal, spatial and trials modes of the model as shown in **Figure 2**. This decomposition is unique, which allows interpretation of those components as the neural populations. We also found that the method is robust against noise. Even if the noise is reflected in the components, the components still cover the main feature of the model as shown in **Figure 4**. We were able to uncover the true number of components using the core consistency diagnostic (see **Figure 3**). For applications with experimental recordings where the number of components is not known, the core consistency diagnostic is often an effective way to estimate the correct number of components.

4.2. Comparison With Classical Matrix Factorization Methods

Different ICA methods and PCA were not able to fully recover the true components corresponding to individual populations. ICA relies on the strong assumption of statistical independence of components, even if that is relaxed in the ERBM approach. In a biological network, this assumption is unlikely to be met since it is expected to process and integrate information, thus correlated patterns are expected. We suspect that ICA methods failed because of highly correlated component vectors in the *time* mode. PCA tried to minimize the correlation between components, which is unlikely to be met in a neural network. On the other hand, CP does not rely on the assumption of independence and is, therefore, able to recover the components successfully. Also, ICA and PCA methods work on signals arranged as matrices, which requires unfolding the tensor. Unfolding destroys the multi-linear structure the CP model benefits from, and ICA/PCA cannot make use of that structure.

4.3. Extensions of CP Model for Different Models of Neuronal Populations Activity

In our toy feed-forward model, the data had a low-rank CP structure and a CP decomposition was able to fully recover it. We also studied the result of introducing non-linear neural response and showed that, for a small amount of non-linearity, CP is able to identify populations. However, for larger non-linearities we see the failure of the CP method as the data violates the CP assumptions to a larger extent. In a more realistic model, we can introduce recurrent connections, multiple simultaneous inputs and non-linear response of neural populations. In such cases, in particular for the second case, using a CP model would lead to uniqueness issues due to linearly dependent components. For instance, this can occur when two different stimuli are injected in the network, thus each stimulus in each neural population would correspond to a single component, but all components related to the same neural population would have the same channel factor. Generalizations of the CP model, e.g., the Tucker model [17], are flexible enough to cover this. While not unique by construction, additional constraints, such as non-negativity, sparsity or the structure of the Tucker core can allow for uniqueness.

When CP is applied to experimental LFP recordings like in Verleger et al. [22], it is unlikely that a low-rank approximation will fully explain the recording due to multiple external sources, contributions from other brain regions, noise effects and non-linearity of the system. However, even if CP is not capable to model the entire data using a low-rank approximation, CP will still pick up low-rank parts, such as a feed-forward structure and thus make it possible to understand sub-structures of the network.

4.4. Measurements Other Than LFP Signal

Other multi-electrode recordings like EEG, electrocorticography (ECoG) or magnetoencephalography (MEG) are very similar to LFP in the sense that they also consist of contributions of multiple sources. Due to the linearity of electromagnetism, these contributions are also additive. Recent work [47] allows to simulate not only LFP but also other electrophysiological signals in the same framework. Thus, the kernel method can also be applied to these other observables. If the kernels have a good rank-one approximation, this work would also apply to EEG, ECoG, or MEG. Since all those observables are linked by the same underlying population activity, it is also possible to combine several observables in a fusion framework jointly analyzing multiple tensors [48].

DATA AVAILABILITY STATEMENT

The simulation and analysis scripts for this study can be found here https://github.com/CINPLA/MultiLinearPopulationAnalysis-MLPA. We put the final code here.

AUTHOR CONTRIBUTIONS

EA and AS conceived and conceptualized the project. JG, EA, and AS wrote and executed all the codes for simulating and analysing the data. JG, EA, GE, and AS wrote the paper. All authors contributed to the article and approved the submitted version.

FUNDING

JG has been funded by the National Science Foundation under the award NSF/DMS(RTG) #1246991. GE, EA, and AS have been funded by the Research Council of Norway (NFR) through the following projects: GE [project# 250128 (COBRA)], EA [project# 300489 (IKTPLUSS)], and AS [project# 250128 (COBRA) and 300504 (IKTPLUSS)].

ACKNOWLEDGMENTS

We would like to thank Espen Hagen for providing the kernels for this study.

REFERENCES

1. Einevoll GT, Kayser C, Logothetis NK, Panzeri S. Modelling and analysis of local field potentials for studying the function of cortical circuits. *Nat Rev Neurosci*. (2013) **14**:770–85. doi: 10.1038/nrn3599
2. Pesaran B, Vinck M, Einevoll GT, Sirota A, Fries P, Siegel M, et al. Investigating large-scale brain dynamics using field potential recordings: analysis and interpretation. *Nat Neurosci*. (2018) **21**:903–19. doi: 10.1038/s41593-018-0171-8
3. Pettersen KH, Lindén H, Dale AM, Einevoll GT. Extracellular spikes and CSD. In: Brette R, Destexhe A, editors. *Handbook of Neural Activity Measurement*. Cambridge: Cambridge University Press (2012). p. 92–135. doi: 10.1017/CBO9780511979958.004
4. Nicholson C, Freeman JA. Theory of current source-density analysis and determination of conductivity tensor for anuran cerebellum. *J Neurophysiol*. (1975) **38**:356–68. doi: 10.1152/jn.1975.38.2.356
5. Pettersen KH, Devor A, Ulbert I, Dale AM, Einevoll GT. Current-source density estimation based on inversion of electrostatic forward solution: effects of finite extent of neuronal activity and conductivity discontinuities. *J Neurosci Methods*. (2006) **154**:116–33. doi: 10.1016/j.jneumeth.2005.12.005
6. Helena. Kernel current source density method. *Neural Comput*. (2012) **24**:541–75. doi: 10.1162/NECO_a_00236
7. Barth DS, Di S. Laminary excitability cycles in neocortex. *J Neurophysiol*. (1991) **65**:891–8. doi: 10.1152/jn.1991.65.4.891
8. Leski S, Kublik E, Swiejkowski Da, Wróbel A, Wójcik DK. Extracting functional components of neural dynamics with independent component analysis and inverse current source density. *J Comput Neurosci*. (2010) **29**:459–73. doi: 10.1007/s10827-009-0203-1
9. Makarov VA, Makarova J, Herreras O. Disentanglement of local field potential sources by independent component analysis. *J Comput Neurosci*. (2010) **29**:445–57. doi: 10.1007/s10827-009-0206-y
10. Głçbska H, Potworowski J, Łçski S, Wójcik DK. Independent components of neural activity carry information on individual populations. *PLoS ONE*. (2014) **9**:e105071. doi: 10.1371/journal.pone.0105071
11. Gratiy SL, Devor A, Einevoll GT, Dale AM. On the estimation of population-specific synaptic currents from laminar multielectrode recordings. *Front Neuroinform*. (2011) **5**:32. doi: 10.3389/fninf.2011.00032
12. Einevoll GT, Pettersen KH, Devor A, Ulbert I, Halgren E, Dale AM. Laminar population analysis: estimating firing rates and evoked synaptic activity from multielectrode recordings in rat barrel cortex. *J Neurophysiol*. (2007) **97**:2174–90. doi: 10.1152/jn.00845.2006
13. Glabska HT, Norheim E, Devor A, Dale AM, Einevoll GT, Wojcik DK. Generalized laminar population analysis (gLPA) for interpretation of multielectrode data from cortex. *Front Neuroinform*. (2016) **10**:1. doi: 10.3389/fninf.2016.00001
14. Harshman RA. Foundations of the PARAFAC procedure: models and conditions for an explanatory multi-modal factor analysis. *UCLA Work Pap Phonet*. (1970) **16**:1–84.
15. Carroll JD, Chang JJ. Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. *Psychometrika*. (1970) **35**:283–319. doi: 10.1007/BF02310791
16. Kruskal JB. Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear Algebra Appl*. (1977) **18**:95–138. doi: 10.1016/0024-3795(77)90069-6
17. Kolda TG, Bader BW. Tensor decompositions and applications. *SIAM Rev*. (2009) **51**:455–500. doi: 10.1137/07070111X
18. Acar E, Yener B. Unsupervised multiway data analysis: a literature survey. *IEEE Trans Knowl Data Eng*. (2009) **21**:6–20. doi: 10.1109/TKDE.2008.112
19. Papalexakis EE, Faloutsos C, Sidiropoulos ND. Tensors for data mining and data fusion: models, applications, and scalable algorithms. *ACM Trans Intell Syst Technol*. (2016) **8**:Article 16. doi: 10.1145/2915921
20. Cole HW, Ray JW. EEG correlates of emotional tasks related to attentional demands. *Int J Psychophysiol*. (1985) **3**:33–41. doi: 10.1016/0167-8760(85)90017-0
21. Möcks J. Topographic components model for event-related potentials and some biophysical considerations. *IEEE Trans Biomed Eng*. (1988) **35**:482–4. doi: 10.1109/10.2119
22. Verleger R, Paulick C, Möcks J, Smith JL, Keller K. Parafac and go/no-go: disentangling CNV return from the P3 complex by trilinear component analysis. *Int J Psychophysiol*. (2013) **87**:289–300. doi: 10.1016/j.ijpsycho.2012.08.003
23. Acar E, Bingol CA, Bingol H, Bro R, Yener B. Multiway analysis of epilepsy tensors. *Bioinformatics*. (2007) **23**:i10–8. doi: 10.1093/bioinformatics/btm210
24. De Vos M, Vergult A, De Lathauwer L, De Clercq W, Van Huffel S, Dupont P, et al. Canonical decomposition of ictal scalp EEG reliably detects the seizure onset zone. *Neuroimage*. (2007) **37**:844–54. doi: 10.1016/j.neuroimage.2007.04.041
25. Williams AH, Kim TH, Wang F, Vyas S, Ryu SI, Shenoy KV, et al. Unsupervised discovery of demixed, low-dimensional neural dynamics across multiple timescales through tensor component analysis. *Neuron*. (2018) **98**:1099–115. doi: 10.1016/j.neuron.2018.05.015
26. Ermentrout GB, Terman DH. *Mathematical Foundations of Neuroscience*. Vol. 35. Springer Science & Business Media (2010).
27. Gerstner W, Kistler WM, Naud R, Paninski L. *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*. Cambridge: Cambridge University Press (2014). doi: 10.1017/CBO9781107447615
28. MATLAB. *Version 9.5.0.1067069 (R2018b) Update 4'*. Natick, MA: The MathWorks Inc. (2018).
29. Hagen E, Dahmen D, Stavrinou ML, Lindén H, Tetzlaff T, van Albada SJ, et al. Hybrid scheme for modeling local field potentials from point-neuron networks. *Cereb Cortex*. (2016) **16**:P67. doi: 10.1186/1471-2202-16-S1-P67
30. Skaar JEW, Stasik AJ, Hagen E, Ness TV, Einevoll GT. Estimation of neural network model parameters from local field potentials (LFPs). *PLoS Comput Biol*. (2020) **16**:e1007725. doi: 10.1371/journal.pcbi.1007725
31. Potjans T, Diesmann M. The cell-type specific cortical microcircuit: relating structure and activity in a full-scale spiking network model. *Cereb Cortex*. (2012) **24**:785–806. doi: 10.1093/cercor/bhs358
32. Hitchcock FL. The expression of a tensor or a polyadic as a sum of products. *J Math Phys*. (1927) **6**:164–89. doi: 10.1002/sapm192761164
33. Acar E, Dunlavy DM, Kolda TG. A scalable optimization approach for fitting canonical tensor decompositions. *J Chemometr*. (2011) **25**:67–86. doi: 10.1002/cem.1335
34. Håstad J. Tensor rank is NP-complete. *J Algorithms*. (1990) **11**:644–54. doi: 10.1016/0196-6774(90)90014-6
35. Bro R, Kiers HAL. A new efficient method for determining the number of components in PARAFAC models. *J Chemometr*. (2003) **17**:274–86. doi: 10.1002/cem.801
36. Bro R. PARAFAC. Tutorial and applications. *Chemometr Intell Lab Syst*. (1997) **38**:149–71. doi: 10.1016/S0169-7439(97)00032-4
37. Tucker LR. Some mathematical notes on three-mode factor analysis. *Psychometrika*. (1966) **31**:279–311. doi: 10.1007/BF02289464
38. Bader BW, Kolda TG. *MATLAB Tensor Toolbox Version 3.1*. (2019). Available online at: https://www.tensortoolbox.org
39. Dunlavy DM, Kolda TG, Acar E. *Poblano v1.0: A Matlab Toolbox for Gradient-Based Optimization*. Albuquerque, NM; Livermore, CA: Sandia National Laboratories (2010).
40. Andersson CA, Bro R. The N-way toolbox for MATLAB. *Chemometr Intell Lab Syst*. (2000) **52**:1–4. doi: 10.1016/S0169-7439(00)00071-X
41. Hyvärinen A. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Trans Neural Netw*. (1999) **10**:626–34. doi: 10.1109/72.761722
42. Li XL, Adali T. Blind spatiotemporal separation of second and/or higher-order correlated sources by entropy rate minimization. In: *ICASSP'2010: Proceedings of IEEE International Conference on Acoustics, Speech, Signal Processing*. Dallas, TX (2010). p. 1934–7. doi: 10.1109/ICASSP.2010.5495311
43. Tomasi G, Bro R. A comparison of algorithms for fitting the PARAFAC model. *Comput Stat Data Anal*. (2006) **50**:1700–34. doi: 10.1016/j.csda.2004.11.013
44. Hyvärinen A, Oja E. Independent component analysis: algorithms and applications. *Neural Netw*. (2000) **13**:411–30. doi: 10.1016/S0893-6080(00)00026-5
45. Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat Methods*. (2020) **17**:261–72. doi: 10.1038/s41592-019-0686-2

46. beim Graben P, Kurths J. Simulating global properties of electroencephalograms with minimal random neural networks. *Neurocomputing*. (2008) **71**:999–1007. doi: 10.1016/j.neucom.2007.02.007

47. Hagen E, Næss S, Ness TV, Einevoll GT. Multimodal modeling of neural network activity: computing LFP, ECoG, EEG, and MEG signals with LFPy 2.0. *Front Neuroinform*. (2018) **12**:92. doi: 10.3389/fninf.2018.00092

48. Acar E, Schenker C, Levin-Schwartz Y, Calhoun VD, Adali T. Unraveling diagnostic biomarkers of schizophrenia through structure-revealing fusion of multi-modal neuroimaging data. *Front Neurosci*. (2019) **13**:416. doi: 10.3389/fnins.2019.00416

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

APPENDIX

Kernels for LFP Approximation

The kernels used in this work were computed as described in Hagen et al. [29] for a cat cortex model [31]. We selected a random sequence of only excitatory kernels where we made sure that the first kernel projects from talamic input. We used the following kernels: thalamus → layer 4, layer 4 → layer 6, layer 6 → layer 5, and layer 5 → layer 2/3. Kernels were computed with ±20ms around the δ input. Due to causality, no response is expected before the pulse. **Figure A1** shows the selected kernels as well as their rank one approximation. As it can be seen, a rank one approximation approximates the original kernel well.

This is shown more systematically in **Figure A2**. This plot shows the variance explained for all kernels that have been computed for the cat cortex model [31], that is all combinations from the four layers, both excitatory and inhibitory [$(4+4)^2$ kernels] as well as talamic connections to this layers (8). More than 90% of those kernels can be approximated with more than 90% variance explained by a rank 1 approximation.

Non-linear Case: Analysis Using CP With Non-negativity Constraints

Figure A3 demonstrates the temporal components captured by the CP model with non-negativity constraints. Here, LFP tensors are generated using different strength of non-linearity (β), and analyzed using CP models with non-negativity constraints in the *time* and *trial* mode.

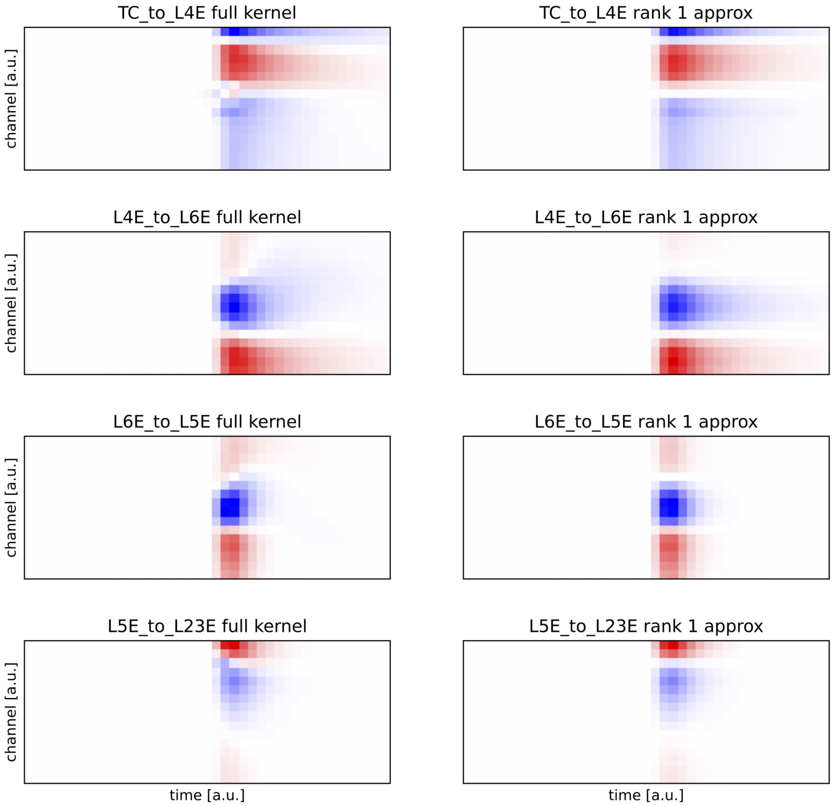


FIGURE A1 | Comparison of the full kernels (left side) and their rank 1 approximation (right side). Plots have equal scale.

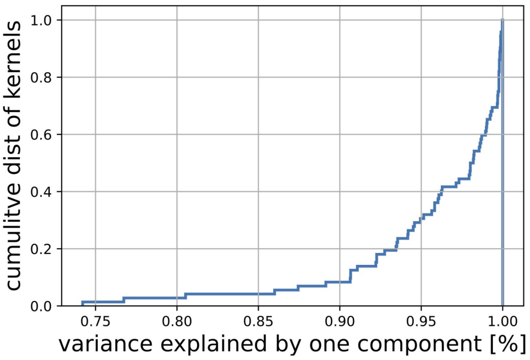


FIGURE A2 | The plot shows the variance explained by a rank 1 approximation for all kernels that have been computed form the cat cortex model [31].

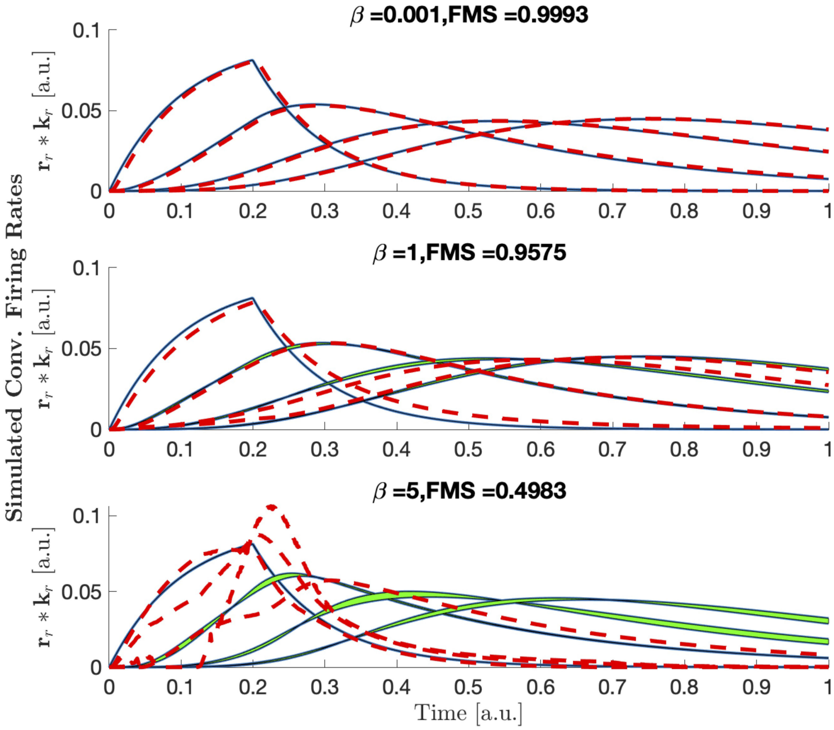


FIGURE A3 | Same as **Figure 7**, but with bounded CP to positive values in the firing rate and trial components. Note that for small non-linearities, the temporal components are almost identical for constrained and unconstrained CP, while for strong non-linearities, where CP fails, the components are very different.

frontiers
in Medicine

ORIGINAL RESEARCH
published: 07 October 2020
doi: 10.3389/fmed.2020.573522

Deep Learning Improves Osteonecrosis Prediction of Femoral Head After Internal Fixation Using Hybrid Patient and Radiograph Variables

Wanbo Zhu[1,2†], *Xianzuo Zhang*[1†], *Shiyuan Fang*[1*], *Bing Wang*[3*] *and Chen Zhu*[1*]

[1] *Department of Orthopedics, The First Affiliated Hospital of USTC, Division of Life Sciences and Medicine, University of Science and Technology of China, Hefei, China,* [2] *Department of Orthopedics, Affiliated Anhui Provincial Hospital of Anhui Medical University, Hefei, China,* [3] *School of Electrical and Information Engineering, Anhui University of Technology, Ma'anshan, China*

OPEN ACCESS

Edited by:
Axel Hutt,
Inria Nancy - Grand-Est Research Centre, France

Reviewed by:
Tobias Winkler,
Charité - University Medicine Berlin, Germany
Lynne Christine Jones,
Johns Hopkins Medicine, United States

***Correspondence:**
Shiyuan Fang
fangshiyuan2008@126.com
Bing Wang
wangbing@ustc.edut
Chen Zhu
zhuchena@ustc.edu.cn

[†] *These authors have contributed equally to this work*

Specialty section:
This article was submitted to Translational Medicine, a section of the journal Frontiers in Medicine

Received: *17 June 2020*
Accepted: *01 September 2020*
Published: *07 October 2020*

Citation:
Zhu W, Zhang X, Fang S, Wang B and Zhu C (2020) Deep Learning Improves Osteonecrosis Prediction of Femoral Head After Internal Fixation Using Hybrid Patient and Radiograph Variables. Front. Med. 7:573522. doi: 10.3389/fmed.2020.573522

Femoral neck fractures (FNFs) are a great public health problem that leads to a high incidence of death and dysfunction. Osteonecrosis of the femoral head (ONFH) after internal fixation of FNF is a frequently reported complication and a major cause for reoperation. Early intervention can prevent osteonecrosis aggravation at the preliminary stage. However, at present, failure to diagnose asymptomatic ONFH after FNF fixation hinders effective intervention at early stages. The primary objective of this study was to develop a predictive model for postoperative ONFH using deep learning (DL) methods developed using plain X-ray radiographs and hybrid patient variables. A two-center retrospective study of patients who underwent closed reduction and cannulated screw fixation was performed. We trained a convolutional neural network (CNN) model using postoperative pelvic radiographs and the output regressive radiograph variables. A less experienced orthopedic doctor, and an experienced orthopedic doctor also evaluated and diagnosed the patients using postoperative pelvic radiographs. Hybrid nomograms were developed based on patient and radiograph variables to determine predictive performance. A total of 238 patients, including 95 ONFH patients and 143 non-ONFH patients, were included. A CNN model was trained using postoperative radiographs and output radiograph variables. The accuracy of the validation set was 0.873 for the CNN model, and the algorithm achieved an area under the curve (AUC) value of 0.912 for the prediction. The diagnostic and predictive ability of the algorithm was superior to that of the two doctors, based on the postoperative X-rays. The addition of DL-based radiograph variables to the clinical nomogram improved predictive performance, resulting in an AUC of 0.948 (95% CI, 0.920–0.976) and better calibration. The decision curve analysis showed that adding the DL increased the clinical usefulness of the nomogram compared with a clinical approach alone. In conclusion, we constructed a DL facilitated nomogram that incorporated a hybrid of radiograph and patient variables, which can be used to improve the prediction of preoperative osteonecrosis of the femoral head after internal fixation.

Keywords: osteonecrosis, femoral neck fracture, clinical prediction, artificial intelligence, nomogram

INTRODUCTION

Hip fracture is a significant public health concern that affects 4.5 million people worldwide each year and this number is expected to increase to 21 million in the next 40 years (1, 2). Femoral neck fracture (FNF) is one of the most common types of hip fracture, accounting for 49–80% of all hip fractures (3, 4). Despite the availability of multiple effective internal fixation procedures, ∼10–48.8% femoral neck fractures require reoperation (5–7). Osteonecrosis of the femoral head (ONFH) is a major cause of reoperation for FNF (8). Joint disfunction, pain, disability, and mental anguish caused by ONFH result in great suffering for patients (9–11). End-stage ONFH often inevitably requires artificial joint replacement surgery, an invasive and economically costly technique. Early diagnosis can facilitate the application of interventions that can avoid or delay arthroplasty to a certain extent (12–14). However, misdiagnoses and delayed diagnoses are common due to the lack of preliminary symptoms, typical features, and internal fixation interference on radiographs (14). Different diagnostic criteria or simple visual estimates are used by radiologists for practical imaging diagnosis, resulting in unsatisfactory levels of diagnostic consistency and accuracy (15). Therefore, early accurate and consistent prediction of ONFH in patients after FNF internal fixation may hold the key for improving patient outcomes.

Deep learning (DL) using radiographs has a proven ability of classifying bone structures and features in specific sites with expert-level accuracy (16, 17). Convolutional Neural Networks (CNNs) are the most suitable models for image recognition of DL, and have been widely used for the orthopedic diagnosis of wrists and ankles (18, 19). Gale et al. developed a hip fracture detector using DL and achieved an AUC of 0.994 (20). Cheng et al. reported on a deep convolutional neural network (DCNN) for the detection and localization of hip fractures using pelvic radiographs, which achieved an AUC of 0.98 for the identification of hip fractures (21). Recently, Chee et al. made a breakthrough discovery for the diagnosis of early ONFH using radiography through deep learning (22). This model achieved an AUC of 0.93 and sensitivity and specificity that were not inferior to the diagnosis made by both the less experienced and experienced radiologists. Their study indicated the potential of DL for the diagnosis and prediction of ONFH, especially for X-ray imaging. However, the implementation of DL for the diagnosis of postoperative ONFH using digital radiography remains unexplored. Postoperative X-rays are highly affected by interference, such as that of internal fixation devices, which cause difference between the images on radiographs and the original appearance of the femoral neck and femoral head. Since postoperative X-rays are the most common method used for early examination, a consistent diagnosis based on postoperative X-rays made using DL may improve the prediction of postoperative ONFH for better prognosis. In this study, we designed and assessed the diagnostic performance of a DL algorithm based on the CNN network model using postoperative X-rays. We also compared the accuracy of the diagnosis of postoperative ONFH between this DL model and assessments made by two orthopedic doctors of different levels of experience.

In previous studies, a large number of research studies have indicated that patient and interventional variables, including demography, fracture classification, laboratory examination, reduction quality, and initial postoperative rehabilitation, are significantly associated with postoperative ONFH (23–26). However, intraoperative, and postoperative factors, especially radiographic variables, including intraoperative reduction and fracture healing, have yet to be incorporated into routine clinical postoperative ONFH prediction. In this study, a DL facilitated predictive model using a hybrid of patient and artificial intelligence (AI) radiographic variables, was also developed. Comparisons were made with a single clinical prediction model was performed to estimate whether DL could improve the prediction of postoperative ONFH.

MATERIALS AND METHODS

Study Population

Data were obtained from two urban tertiary hospitals, The First Affiliated Hospital of University of Science and Technology of China (FAH) and the Southern Branch of the First Affiliated Hospital of University of Science and Technology of China (SBH). One hundred thirty-nine FAH patients and 99 SBH patients who had received closed reduction and cannulated screw fixation from June 2013 to January 2015 were enrolled in this study. The patient inclusion criteria were as follows: (i) Patients over 18 years of age with fresh FNFs; (ii) Postoperative pelvic radiographs obtained 6 months after surgery; (iii) Continuous follow-up for a minimum of 5 years with the clinical characteristics available. The exclusion criteria were as follows: (i) Pathological fractures and bilateral fractures; (ii) Long-term hormone use. The treatment standard and strategy used for femoral neck fracture was the cannulated compression screws fixation technique, based on American Academy of Orthopedic Surgeons guidelines (27). Postoperative ONFH was diagnosed using pelvic MRIs or co-diagnosis by three experienced orthopedic surgeons based on the pelvic radiograph obtained at the last follow-up. This study was approved by the Ethics Committees of both hospitals. Exemption of the informed consent, the information disclosure, and a negative opportunity are guaranteed in the Ethical approval (20-P-049).

Demographics, comorbidities, smoking status, alcohol use, blood tests, preoperative Garden classification, Pauwels angle, preoperative interval from injury, operation associated data, postoperative Garden index, preoperative interval to weight bearing and other baseline patient and clinical data were derived from medical and follow-up records. The data were de-identified after patient variables were collected.

Imaging Studies

Image acquisition and retrieval procedures were conducted using Picture Archiving and Communication Systems (PACS) on FAH and SBH patients. Digital radiographs of the hip were obtained using Digital Diagnostics (Philips Healthcare) on FAH patients and Discovery XR656 (GE Healthcare) on SBH patients. The size of the stored images varied from 2,128 × 2,248 pixels to 2,688 × 2,688 pixels, with 8-bit grayscale color. Each radiograph was labeled based on the final diagnosis of postoperative ONFH. Geometric, smooth, concave, bandlike low-signal intensity lesions at the femoral head on the T1-weighted images were regarded as pathognomonic MRI findings of ONFH. For MRI data not obtained at the last follow-up (45/238, 18.9%), diagnosis was based on pelvic plain radiographs obtained at the last follow-up and was set as a reference for labeling. The Association Research Circulation Osseous (ARCO) classification system was used as the diagnostic standard for ONFH (28).

Radiographic image files were loaded for processing using a MATLAB library (version 2017b, MathWorks, USA). The 7 × 7 cm images centered on the bilateral femoral heads were cropped. The center coordinates were manually recorded in advance. Radiographs were standardized to a common size and pixel intensity distribution. The images were down-sampled and padded to a final size of 120 × 120 pixels. Mean pixel intensity and standard deviation of each image was normalized.

Algorithm Development and Extraction of Image Variables

For the development of a deep learning algorithm, we used MATLAB (version 2017b, MathWorks, USA) to implement a CNN model to compute abstract image features from input image pixel arrays. The design of the CNN model is shown in **Table 1**. The CNN model consisted of three convolutional blocks, a dropout and full connection layers. Each convolutional block comprised of convolutional operation, batch normalization, relu, and average pooling. The input used was Pixel values were set at 120*120 using a digital image. Cubic convolution and pooling were performed on each layer to adjust the weights of the neural network, using the difference between the output and true labels.

The patients in the dataset were assigned to different groups as follows: 149 (63%) for training, 17 (7%) for validation and 72 (30%) for testing. The output results underwent regression analysis. The network output was a probability distribution for the continuous variables of the regression coefficient from 0 to 1.25, which was divided at 0.25 intervals into classified labels, 1–5. Higher label values were more likely to be considered to more strongly predict postoperative ONFH. In this study, this output label was referred to as the AI index classification.

TABLE 1 | The design of CNN model.

Type	Operations	Filter shape	Input size
Conv1	Conv	8 × 7 × 7 × 1	120*120
	batchnorm		
	relu		
	avgpool		8 × 120*120
Conv2	Conv	7 × 5 × 5 × 8	8 × 60*60
	batchnorm		
	relu		
	avgpool		7 × 60*60
Conv3	Conv	5 × 3 × 3 × 7	7 × 30*30
	batchnorm		
	relu		
	avgpool		5 × 30*30
Dropout	Dropout	1*1	5 × 15*15
FC	Fully connected	1,125*1	5 × 15*15
Regression	Regression output	1*1	1*1

Algorithm Evaluation

Seventy-two independent datasets were used to test the trained predictive model to evaluate its accuracy for postoperative ONFH prediction. The probability of the diagnosis being postoperative ONFH generated by the model was evaluated using the receiver operating characteristic (ROC) curve and the area under the curve (AUC). The sensitivity, accuracy, recall and specificity of the radiographs for the prediction of ONFH were measured using a cutoff level probability of 0.5. A training curve was used to determine root mean squared error (RMSE) and loss, while a precision-recall curve was used to determine precision and recall.

Image Predictive Variable Evaluation

We compared the AI index with the predictive measurement scores assigned by the two orthopedic surgeons of different levels of experience with the results of the DL algorithm based on the same X-rays to evaluate the performance of the algorithm. Radiographs obtained 6-months after anteroposterior hip operations were randomly divided into two IPAC sequences by the study coordinator. A less experienced orthopedic doctor (Doctor A, 3rd year of residency in orthopedics) and an experienced orthopedic doctor (Doctor B, 18 years in orthopedics) participated in the reading session. Both doctors were not involved in surgery, data collection or reference labeling. A score based on the subjective prediction of the doctors using the postoperative X-ray to determine the most likely outcome at final follow-up was assigned using a 1–5 grading system. One indicated that the development of ONFH was considered to be impossible, while 5 indicated that the development of ONFH was considered to be certain. Each doctor independently graded the predictive variables for ONFH. Comparison between the performance of the AI index and the evaluation made by the two doctors was conducted through calibration and ROC analysis.

Development of Prediction Models

A multivariable logistic regression analysis was used to develop the clinical predication model based on patient and clinical variables. AI index classification was applied as a candidate predictor for univariate and multivariable logistic regression analyses for the construction of a DL-based postoperative ONFH prediction model using hybrid variables. A clinical prediction nomogram and a DL-based nomogram were then constructed based on multivariate logistic regression models. The work flowchart of this study is presented in **Figure 1**.

Assessment of Nomogram Performance

AI-based nomogram and clinical nomogram calibration were assessed using a calibration curve. The discrimination performance of both the AI-based nomogram and clinical nomogram were quantified using the AUC.

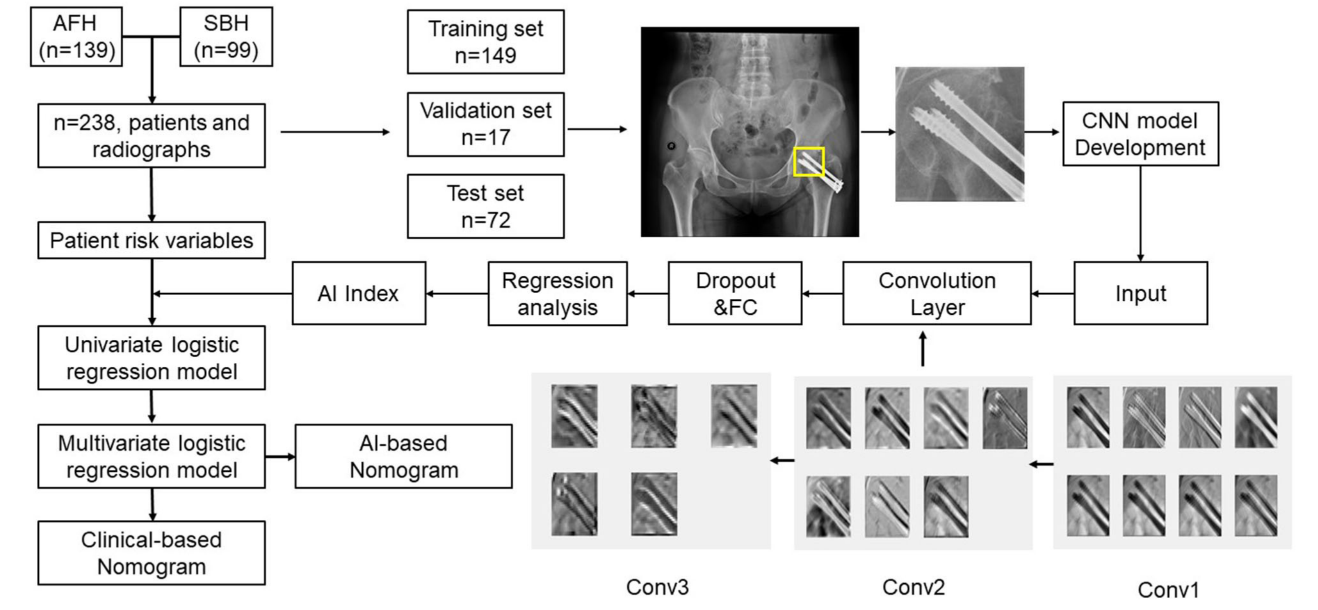


FIGURE 1 | Flowchart of hybrid nomogram construction.

Clinical Use

Decision curve analysis (DCA) was performed by calculating the net benefits for a range of threshold probabilities to estimate the clinical utility of the nomogram.

Statistical Analysis

Median and mean standard deviation (SD) were used to describe continuous variables. Categorical variables were presented as frequencies and percentages. Statistical comparisons between groups were performed using the Mann-Whitney *U*-test and Chi-square test. R software version 3.0.1 was used to construct the nomogram. The "pROC" package was used to plot ROC curves. Nomogram construction and calibration plot creation were performed using the "rms" package. DCA was performed using the "dca.R" package. Model selection was based on the forward–backward step-wise method using the likelihood ratio test with Akaike's information criterion as the stopping rule. The model with the smallest Akaike Information Criterion was selected as the final model. The statistical significance levels reported are all two-sided, with statistical significance set at a *P*-value of 0.05.

RESULTS

Patient and Radiograph Characteristics

Postoperative radiographs of a total of 238 patients, including 95 ONFH patients and 143 normal patients were used for the development of the DL model and construction of the predictive nomogram. Imaging feature variables were extracted from each radiograph and were referred to as the AI index of all patients. **Table 2** shows the baseline characteristics of the patients. Significant differences were found in BMI, Charlson comorbidity index, Injury Severity Score (ISS), d-dimer, timing of reduction, Garden classification and AI index between patients with ONFH and those without ONFH (**Table 2**).

Performance of the CNN Model

A CNN model was established for the extraction of radiograph variables. The precision-recall curve of the test set is shown in **Figure 2A**, while the threshold value at the break-even point was 0.425. This point was set as the highest sum of sensitivity and specificity. Training accuracy values at this threshold for the training set was 0.903 and 0.873 for the test set. The change in RMSE and loss during the training process are shown in **Figure 2B**. Deviation of the RMSE in the training set and test set gradually decreased and the two curves leveled off (upper diagram) along with the increase of iterations. Similarly, as the number of iterations increased the deviation in loss between the training set and test set gradually decreased.

Performance of the Predictive Radiograph AI Variables

The calibration curve of the AI index for the prediction of postoperative ONFH demonstrated good agreement between prediction and actual observations, compared with that of Doctor A and Doctor B (**Figure 3A**). The sensitivity value was 0.910 (95% CI, 0.871–0.949) for the AI index, 0.657 (95% CI, 0.591–0.724) for the less experienced Doctor A and 0.827 (95% CI, 0.776–0.879) for experienced Doctor B (**Figure 3B**). The DCA curves shown in **Figure 3C** indicate that when the threshold probability for a doctor or a patient was within the range of 0.09–0.96, the AI index added more net benefits for the prediction, than that of Doctor A or Doctor B.

TABLE 2 | Patients baseline characteristics stratified by ONFH.

	All patients	**Non-ONFH group**	**ONFH group**	***p***
	N* = *238	***N* = *143***	***N* = *95***	
Age	46.4 ± 12.7	45.6 ± 13.3	47.6 ± 11.7	0.215
Sex				0.167
Female	106 (44.5%)	58 (40.6%)	48 (50.5%)	
Male	132 (55.5%)	85 (59.4%)	47 (49.5%)	
BMI	22.7 ± 2.88	22.4 ± 2.82	23.2 ± 2.93	0.048
Smoking				0.875
No	148 (62.2%)	90 (62.9%)	58 (61.1%)	
Yes	90 (37.8%)	53 (37.1%)	37 (38.9%)	
Alcohol use				0.696
No	165 (69.3%)	101 (70.6%)	64 (67.4%)	
Yes	73 (30.7%)	42 (29.4%)	31 (32.6%)	
WIC	1.34 ± 1.40	1.10 ± 1.21	1.68 ± 1.59	0.003
CVD				0.097
No	220 (92.4%)	136 (95.1%)	84 (88.4%)	
Yes	18 (7.56%)	7 (4.90%)	11 (11.6%)	
ISS score				0.029
≤16	210 (88.2%)	132 (92.3%)	78 (82.1%)	
>16	28 (11.8%)	11 (7.69%)	17 (17.9%)	
WBC	7.42 ± 2.44	7.51 ± 2.51	7.28 ± 2.34	0.484
RBC	4.30 ± 0.57	4.31 ± 0.59	4.28 ± 0.55	0.638
Hb	130 ± 16.4	130 ± 16.5	131 ± 16.3	0.871
PLT	181 ± 58.2	178 ± 56.6	185 ± 60.6	0.387
ALB	40.9 ± 3.18	41.1 ± 3.18	40.7 ± 3.19	0.360
D-dimer	4.40 ± 5.59	5.16 ± 6.40	3.27 ± 3.86	0.005
Causes of injury				0.192
High energy trauma	63 (26.5%)	33 (23.1%)	30 (31.6%)	
Low energy trauma	175 (73.5%)	110 (76.9%)	65 (68.4%)	
Timing of reduction				<0.001
<72 h	100 (42.0%)	72 (50.3%)	28 (29.5%)	
72–120 h	97 (40.8%)	58 (40.6%)	39 (41.1%)	
>120 h	41 (17.2%)	13 (9.09%)	28 (29.5%)	
ASA grade				0.223
Grade 1	118 (49.6%)	76 (53.1%)	42 (44.2%)	
Grade 2–3	120 (50.4%)	67 (46.9%)	53 (55.8%)	
Garden classification				0.014
Type 2	19 (7.98%)	17 (11.9%)	2 (2.11%)	
Type 3	116 (48.7%)	63 (44.1%)	53 (55.8%)	
Type 4	103 (43.3%)	63 (44.1%)	40 (42.1%)	
Pauwels angle	53.2 ± 14.8	53.9 ± 15.4	52.1 ± 13.8	0.346
Garden index				0.130
1	43 (18.1%)	29 (20.3%)	14 (14.7%)	
2	61 (25.6%)	34 (23.8%)	27 (28.4%)	
3	70 (29.4%)	36 (25.2%)	34 (35.8%)	
4	64 (26.9%)	44 (30.8%)	20 (21.1%)	
Interval to part weightbearing				0.393
<1 m	16 (6.72%)	10 (6.99%)	6 (6.32%)	
1–3 m	89 (37.4%)	58 (40.6%)	31 (32.6%)	
3–6 m	122 (51.3%)	67 (46.9%)	55 (57.9%)	
>6 m	11 (4.62%)	8 (5.59%)	3 (3.16%)	
Interval to full weightbearing				0.474
<3 m	25 (10.5%)	15 (10.5%)	10 (10.5%)	
3–6 m	161 (67.6%)	93 (65.0%)	68 (71.6%)	
>6 m	52 (21.8%)	35 (24.5%)	17 (17.9%)	
AI index	0.48 ± 0.39	0.24 ± 0.24	0.83 ± 0.29	<0.001

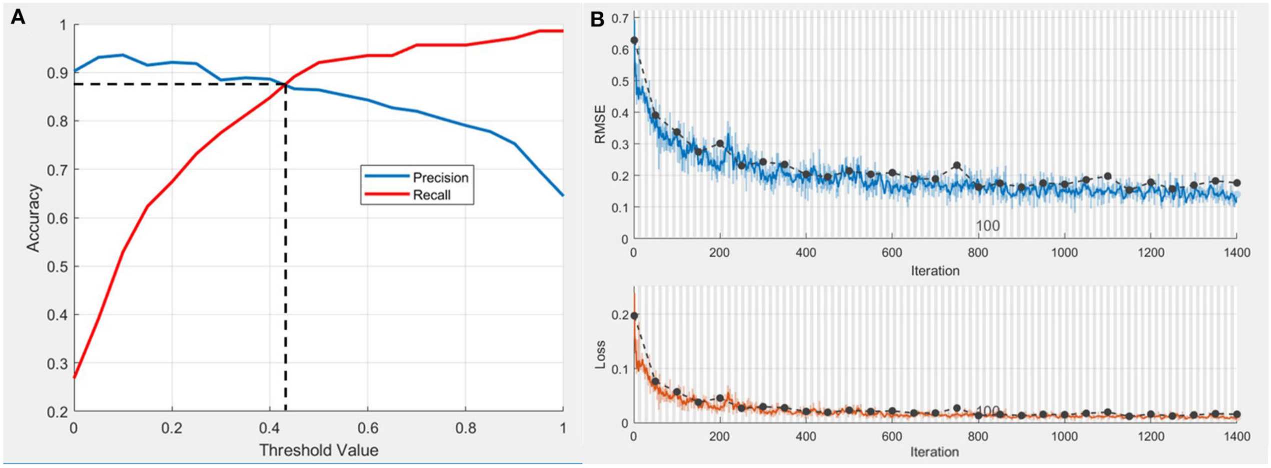


FIGURE 2 | Performance of CNN model in postoperative ONFH prediction. **(A)** Precision-recall curve of test set. The threshold value at Break-Even point is 0.425 and the accuracy at this threshold set is 0.873. **(B)** The change of root mean square error (RMSE) and loss during the training process. Dotted line, RMSE and loss of the training set. Blue wave, RMSE of the validation set. Red wave, loss of the validation set.

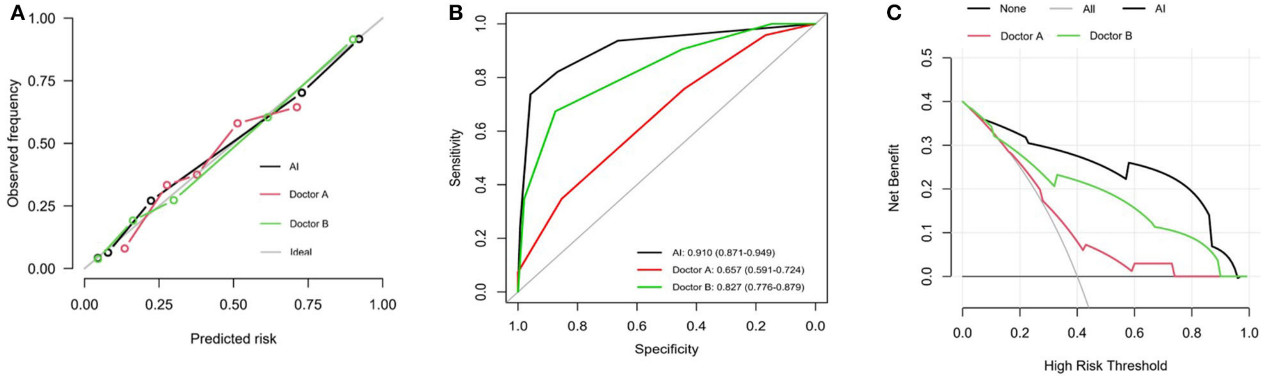


FIGURE 3 | Performance of predictive value of AI index. **(A)** Calibration plots for prediction of AI, Doctor A and Doctor B. Calibration curves depict the calibration of the nomogram in terms of agreement between the predicted risk and outcomes. The 45° gray ideal line represents a perfect Prediction. The closer the dotted line fit is to the ideal line, the better the predictive accuracy of the diagnosis and nomogram is. **(B)** ROC curves for prediction of AI, Doctor A and Doctor B. **(C)** DCA analysis curves for radiodiagnosis of AI, Doctor A and Doctor B. It showed that if the threshold probability is between 0.09 and 0.96, then using the AI index adds more benefit than testing either all or no patients.

Development of a Hybrid Prediction Model

In the univariate logistic regression analysis, BMI, Injury Severity Score (ISS), timing of reduction, Garden classification and AI index were found to be significant factors associated with ONFH in the training cohort (all $P < 0.05$; **Table 2**). In the final multivariate logistic regression model, BMI (HR 0.471, 95% CI 0.187–1.147, $P = 0.101$), ISS (HR 3.427, 95% CI 0.919–13.05, $P = 0.068$), timing of reduction (72 h-120 h: HR 1.533, 95% CI 0.564–4.253, $P = 0.403$; >120 h: HR 9.464, 95% CI 2.471–40.38, $P = 0.002$), Garden classification (Type 3: HR 0.336, 95% CI 0.050–3.315, $P = 0.292$; Type 4: HR 1.344, 95% CI0.243–12.98, $P = 0.745$) and AI index (HR 6.043, 95% CI 4.071–9.717, $P < 0.001$) were identified as hybrid independent predictors of ONFH (**Table 3**). We then created a prediction nomogram that incorporated the above independent predictors and presented it as a hybrid nomogram (**Figure 4A**). A clinical nomogram was also constructed based on independent predictors excluded from the AI index (**Figure 4B**).

Performance of the Hybrid Nomogram

The calibration curve of the hybrid nomogram for the prediction of postoperative ONFH demonstrated good agreement between prediction and actual observations, compared with that of the clinical nomogram (**Figure 5A**). The AUC of the AI-based nomogram was 0.948 (95% CI, 0.920–0.976), while the AUC for the clinical nomogram was 0.696 (95% CI, 0.629–0.763) (**Figure 5B**). The difference was statistically significant, which indicated that the hybrid nomogram showed better discrimination and prediction ability for the diagnosis of ONFH.

TABLE 3 | The results of univariate and step-wise multivariate analyses of confounding variables.

Variable	Univariate model		Multivariate model	
	HR (95% CI)	***P***	**HR (95% CI)**	***P***
Age	1.013 (0.992–1.035)	0.227	–	–
Sex, male	0.668 (0.395–1.126)	0.131	–	–
BMI, ≤24	0.618 (0.361–1.054)	0.077	0.471 (0.187–1.147)	0.101
Smoking, yes	1.083 (0.633–1.847)	0.769	–	–
Alcoholism, yes	1.164 (0.663–2.036)	0.593	–	–
Causes of injury	1.431 (0.851–2.419)	0.147	–	–
ASA grade, grade 2-3	1.412 (0.851–2.419)	0.178	–	–
WIC	1.348 (1.116–1.643)	0.002	Not selected	–
CVD, yes	2.544 (0.964–7.155)	0.063	Not selected	–
ISS score, >16	2.615 (1.178–6.028)	0.020	3.427 (0.919–13.05)	0.068
WBC	0.962 (0.862–1.071)	0.488	–	–
RBC	0.897 (0.567–1.414)	0.640	–	–
PLT	1.002 (0.998–1.007)	0.379	–	–
Hb	1.001 (0.986–1.018)	0.871	–	–
Alb	0.962 (0.885–1.044)	0.358	–	–
D2D	1.411 (0.839–2.382)	0.195	–	–
Timing of reduction			–	–
<72 h	Reference			
72–120 h	1.729 (0.956–3.159)	0.072	1.533 (0.564–4.253)	0.403
>120 h	5.538 (2.562–12.53)	<0.001	9.464 (2.471–40.38)	0.002
Garden classification				
Type 2	Reference		Reference	
Type 3	5.397 (1.443–35.20)	0.029	0.336 (0.050–3.315)	0.292
Type 4	7.150 (1.932–46.41)	0.011	1.344 (0.243–12.98)	0.745
Pauwells angle	0.992 (0.974–1.009)	0.355	–	–
Garden index			Not selected	–
1	Reference		–	–
2	1.645 (0.736–3.774)	0.231	–	–
3	1.956 (0.896–4.400)	0.097	–	–
4	0.942 (0.412–2.181)	0.887	–	–
Interval to part weightbearing			–	–
<1 m	Reference		–	–
1–3 m	0.891 (0.301–2.831)	0.837	–	–
3–6 m	1.368 (0.477–4.241)	0.567	–	–
>6 m	0.625 (0.105–3.207)	0.581	–	–
Interval to full weightbearing			–	–
<3 m	Reference		–	–
3–6 m	1.098 (0.469–2.662)	0.833	–	–
>6 m	0.728 (0.271–1.987)	0.529	–	–
AI index (per 0.25 increase)	4.594 (3.365–6.572)	<0.001	6.043 (4.071–9.717)	<0.001

Clinical Use

The DCA for the hybrid nomogram and for the clinical nomogram are presented in **Figure 5C**. The DCA indicated that when the threshold probability for a doctor or a patient was within the range of 0–0.98, the hybrid nomogram added more net benefits than "treat all" or "treat none" strategies. The range for the clinical nomogram was from 0.2 to 0.7, revealing that use of the hybrid nomogram to predict postoperative ONFH was more beneficial.

DISCUSSION

Early detection and identification of ONFH after femoral neck fracture fixation has been a long-term concern in clinical practice. In this study, we developed and trained a DL model that could use postoperative pelvic radiographs to predict ONFH. The output values of the CNN model successfully stratified patients based on their risk of developing postoperative ONFH, which was referred to as AI index classification for prediction. The predictive

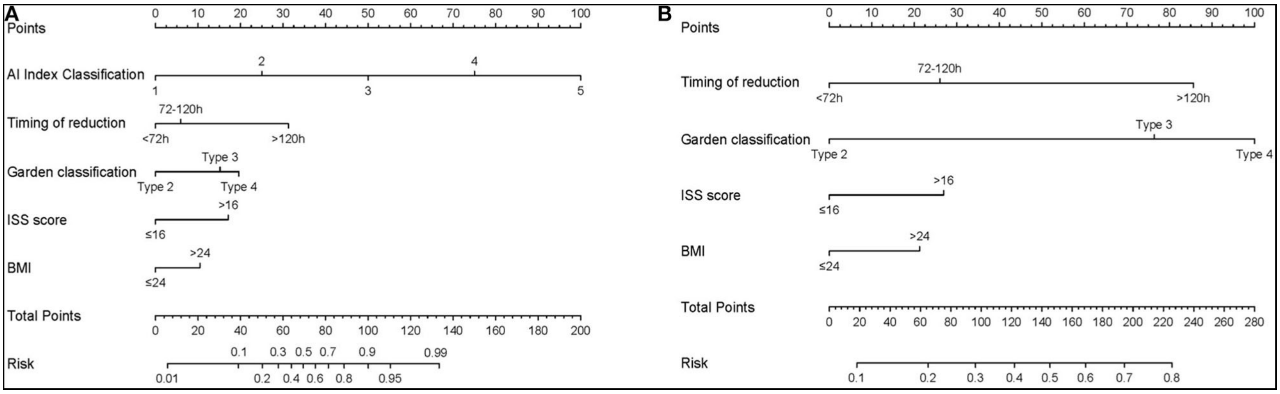


FIGURE 4 | The nomogram for the operative prediction of ONFH. **(A)** Hybrid AI-based nomogram incorporated hybrid independent radiograph and patient variables. **(B)** Clinical-based nomogram constructed based on independent predictors excluded AI index.

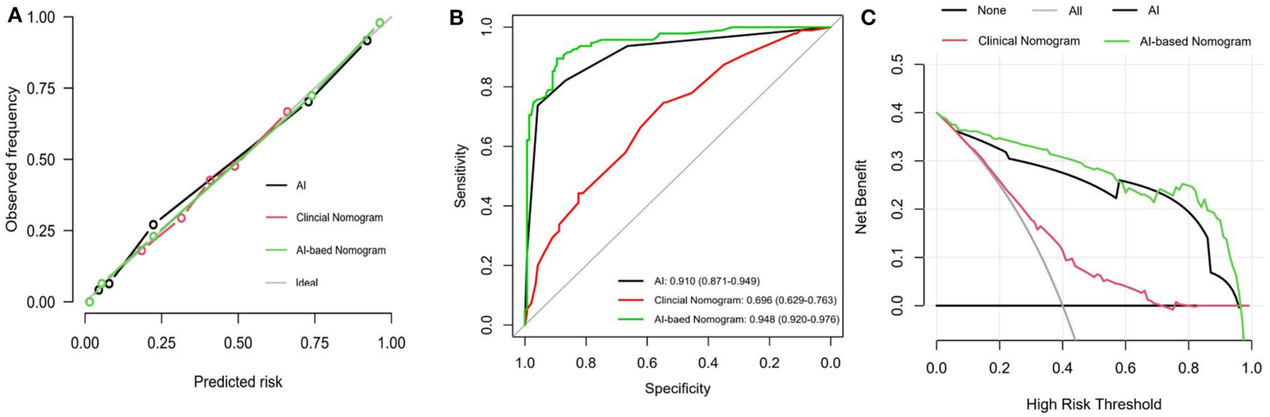


FIGURE 5 | Performance of the hybrid predictive model. **(A)** Calibration plots for AI index, AI-based nomogram and Clinical nomogram. **(B)** ROC curves for prediction of AI index, AI-based nomogram and Clinical nomogram. **(C)** DCA analysis curves for AI radiodiagnosis, AI-based nomogram and Clinical nomogram. The y-axis measures the net benefit. The blue line represents the hybrid AI-based nomogram. The green line represents the clinical nomogram. The gray line represents the assumption that all patients have postoperative ONFH. Thin black line represents the assumption that no patients have postoperative ONFH. The x-axis represents the threshold probability. The threshold probability is where the expected benefit of treatment is equal to the expected benefit of avoiding treatment. It showed that if the threshold probability is between 0 and 0.98, then using the AI-based nomogram adds more benefit in predicting ONFH than testing either all or no patients.

performance of the AI index was significantly superior to the predictive performance of a less experienced orthopedic doctor and non-inferior to that of an experienced orthopedic doctor. A combination of patient and radiograph variables were used to construct an AI-based nomogram for postoperative ONFH prediction. The hybrid nomogram showed better performance for the postoperative prediction of ONFH than a single clinical nomogram, indicating its potential in predicting and targeting ONFH during clinical follow-up to provide a decision base for orthopedic doctors.

Hip pain is the most common postoperative symptom after FNF surgery. It may be associated with fractures, surgery, implant irritation, and early ONFH that should be identified during follow-up. Postoperative X-rays are the most common and readily available imaging examination used for routine clinical follow-up after internal fixation. The detection of sclerotic abnormalities and trabecular interruptions of the femoral head for the diagnosis of postoperative ONFH are subjective and depend on the level of experience and diagnostic criteria used by each doctor. Only radiologists who are rich in experience, may be able to accurately predict ONFH using postoperative X-rays. Even then, objectivity and consistency may be difficult to be achieved. The increased workload of radiologists worldwide has already had a significant impact on the diagnostic performance of radiologists (29, 30). Therefore, DL can be used as a potential auxiliary diagnostic tool for orthopedic diagnoses to obtain stable and accurate diagnoses (16, 31). In this study, we trained a DL model to read postoperative X-rays to predict ONFH. The accuracy and consistency of the DL model was significantly better than that of an orthopedic doctor with less experience. The DL model was similar in accuracy but better in consistency, compared with the experienced orthopedic doctor. This indicated the potential of the use of the DL model for the diagnosis and prediction of postoperative ONFH. Previous studies have indicated that an important feature of the DL model is its ability to detect key features of images through cyclic learning undergone by neural networks, which may be different from the existing understanding and research on image features in black box models. This makes it possible for the diagnostic path of the DL model to differ from existing known diagnostic and prediction criteria, resulting in a positive difference in the diagnostic accuracy of the DL model, compared with that of orthopedic doctors. The DL model created in Chee's study showed a high level of sensitivity and accuracy for the diagnosis of pre-collapse ONFH (22). When we applied the CNN network obtained from this non-traumatic ONFH prediction model to our postoperative ONFH prediction, internal fixation of the postoperative X-ray was found to be one of the major differences between the two models. Recent studies have suggested that different fixation constructs, such as cannulated screws or dynamic hip screws, produce different fracture fixation outcomes. The location differences under the implemented operations standard for the same fixation construct do not significantly affect outcomes (32). During training, we found that the output of the DL model could still reflect prediction efficiency and showed good calibration, even though the positions of the metal internal fixations were not exactly the same and occupied the recognition area in the finite image pixel.

Existing studies using clinical risk factors, such as demographic data, fracture classification, and preoperative interval, to make preoperative predictions for surgical decisions (33–35). Due to the lack of the incorporation of all perioperative variables, especially the intraoperative and postoperative radiograph variables, the preoperative prediction models in these studies have shown difficulties in achieving an ideal predictive ability. For example, the clinical nomogram constructed in our study achieved an AUC of 0.696 (95% CI, 0.629–0.763), which is similar to the AUC of 0.746 obtain by the Naive Bayes Classifier constructed by Cui et al. (36). The predictive ability of a preoperative model is limited for patients who have received certain internal fixation, for example dynamic hip screws and cannulated compression screws (34, 36). The hybrid nomogram showed better prediction performance after the incorporation of patient and radiograph variables, compared with conventional clinical nomograms and the simple radiographic-based DL model for postoperative ONFH prediction. In this study, the hybrid classifier achieved an AUC of 0.948 (95% CI, 0.920–0.976). The variables we included after multivariate regression analysis of all risk factors were similar to that of conventional preoperative clinical prediction models. High-risk factors generally include fracture patterns, preoperative interval, and BMI. Inclusion of the DL model-based imaging prediction significantly improved the ONFH predictive ability of the traditional prediction models, indicating the value of using a combination of variables. The predictive model using hybrid variables more closely mimicked the diagnostic and predictive processes of orthopedic doctors, who are better at interpreting images based on the clinical status of patients (37). The addition of a combination of patient and hospital process variables associated with routine clinical care improved the ability of a DL model trained by Badgeley et al. to predict hip fractures (38). One explanation for this improvement was the presence of non-biological signals on radiographs that are predictive of diseases (39). Although multiple regression analyses were performed for risk factors, including intraoperative reduction, and postoperative weight bearing, the variables included in the single clinical nomogram were all preoperative variables. Among them, Garden classification showed the most assigned value, which was similar to the results of previous studies that found that fracture patterns are crucial for the prediction of postoperative ONFH (7, 40). When the postoperative AI index was included, the attribution of Garden classification decreased significantly, which may be because the AI index already included certain manually incorporated graded variables from the images. The information was considered as a non-biological signal and contributed to the classification. The DL-based prediction model that incorporated a combination of patient and radiograph variables showed a significantly higher ability of prediction postoperative ONFH, and can be used to provide second opinions and a base for doctors to make decisions during clinical follow-up.

In the DCA curves analysis, prediction and diagnosis based on the DL model were found to be non-inferior to that of the two orthopedic doctors, while that of the AI-based nomogram using hybrid variables was superior to imaging prediction alone, allowing for more accurate diagnosis and prediction during clinical follow-up. There is no doubt that the gold standard imaging modality for the preliminary stages of ONFH is MRI (41, 42). However, MRI is not the most common test used to evaluate treatment options and ONFH during postoperative FNF follow-up. MRIs are affected by metal implants, which may cause potential internal fixation losses and thermal effect (43). MRI tests are more expensive, take longer, and require the radiologist to have a higher level of diagnostic experience. Nomograms based on the DL model and clinical variables can improve the ability of positive diagnostic screening and provide doctors the opportunity of obtaining a second opinion.

The AI-based nomogram using hybrid variables may potentially assist in decision making during clinical follow-up as patients with early-stage ONFH may benefit from timely interventions (44). Although the definitive method of treatment for traumatic ONFH remains controversial, certain early interventions have been widely used during post-operative clinical follow-up. For patients with a high probability of developing ONFH, interventions for hip preservation or delayed joint replacement, including platelet-rich plasma (PRP)-incorporated autologous granular and free vascularized fibular, have been proven to be safe and effective procedures for postoperative ONFH (45, 46). Extracorporeal shock wave therapy and alendronate administration can also be potentially performed on patients with a moderate probability of a risk of developing ONFH (47–49). We assessed whether the AI-based nomogram assisted decisions that would improve patient outcomes to justify its clinical usefulness. Our study showed that if the threshold probability was between 0.06 and 0.96, as shown by the constructed decision curves, the AI-based nomogram could predict postoperative ONFH compared with treating either all or no patients. This indicated that early postoperative prediction using this hybrid of patient and radiograph variables can be useful for the application of early interventions that may even allow for a reasonable delay of the onset of arthroplasty (50). Substantial positive rehabilitation can be applied after accurate predictions are obtained after the operation for patients with a lower prediction probability, which will also relieve patient anxiety (51).

This study has some limitations. First, it was conducted on a retrospective cohort study, and is therefore likely to have been affected by selection bias. Second, due to the rarity of the disease, our study included only 238 images in the CNN model. The performance of the CNN model can be improved by using a larger multicenter sample size. Third, our diagnostic criteria for postoperative ONFH was based on follow-up MRIs and typical pelvic radiographs without the use of histopathological confirmation. Therefore, false-negative and false-positive values would not have been avoided due to the subjectivity of the imaging diagnosis method. At the same time, transverse comparison was not conducted with gold standard MRI when postoperative X-rays were included 6 months after surgery. The reason was that, as a retrospective study, MRIs had been performed on only 197 patients, probably due to their high cost. In the future, prospective clinical studies using larger cohorts should be preplanned to investigate strategies that can be used for ONFH prediction of patients after internal fixation.

CONCLUSION

In conclusion, this study presents a DL facilitated nomogram that incorporates hybrid radiograph and patient variables, shows favorable predictive accuracy for preoperative osteonecrosis of femoral head in patients with femoral neck fractures after internal fixation.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

ETHICS STATEMENT

The studies involving human participants were reviewed and approved by the First Affiliated Hospital of USTC. The patients/participants provided their written informed consent to participate in this study.

AUTHOR CONTRIBUTIONS

WZ and XZ conceived and designed the study, and wrote the manuscript. WZ collected the data. CZ, BW, and SF read, corrected, and approved the final manuscript. All authors read and approved the final manuscript.

FUNDING

This work was supported by the National Natural Science Foundation of China (Grant No. 81871788), the project for Science and Technology leader of Anhui Province (Grant No. 2018H177), the Scientific Research Fund of Anhui Education (Grant No. 2017jyxm1097), the Anhui Provincial Postdoctoral Science Foundation (Grant No. 2019B302), and Key Research and Development Plan of Anhui Province (Grant No. 912278014064).

REFERENCES

1. Gullberg B, Johnell O, Kanis JA. World-wide projections for hip fracture. *Osteoporos Int.* (1997) 7:407–13. doi: 10.1007/PL00004148
2. Parker M, Johansen A. Hip fractur. *BMJ.* (2006) 333:27–30. doi: 10.1136/bmj.333.7557.27
3. Thorngren KG, Hommel A, Norrman PO, Thorngren J, Wingstrand H. Epidemiology of femoral neck fractures. *Injury.* (2002) 33(Suppl. 3):C1–7. doi: 10.1016/S0020-1383(02)00324-8
4. Johnell O, Kanis JA. An estimate of the worldwide prevalence, mortality and disability associated with hip fracture. *Osteoporos Int.* (2004) 15:897–902. doi: 10.1007/s00198-004-1627-0
5. Bhandari M, Devereaux PJ, Swiontkowski MF, Tornetta P III, Obremskey W, Koval KJ, et al. Internal fixation compared with arthroplasty for displaced fractures of the femoral neck. A meta-analysis. *J Bone Joint Surg Am.* (2003) 85:1673–81. doi: 10.2106/00004623-200309000-00004
6. Nauth A, Creek AT, Zellar A, Lawendy A-R, Dowrick A, Gupta A, et al. Fracture fixation in the operative management of hip fractures (FAITH): an international, multicentre, randomised controlled trial. *Lancet.* (2017) 389:1519–27. doi: 10.1016/S0140-6736(17)30066-1
7. Do LND, Kruke TM, Foss OA, Basso T. Reoperations and mortality in 383 patients operated with parallel screws for Garden I-II femoral neck fractures with up to ten years follow-up. *Injury.* (2016) 47:2739–42. doi: 10.1016/j.injury.2016.10.033
8. Xu DF, Bi FG, Ma CY, Wen ZF, Cai XZ. A systematic review of undisplaced femoral neck fracture treatments for patients over 65 years of age, with a focus on union rates and avascular necrosis. *J Orthop Surg Res.* (2017) 12:28. doi: 10.1186/s13018-017-0528-9
9. Leonardsson O, Rolfson O, Hommel A, Garellick G, Akesson K, Rogmark C. Patient-reported outcome after displaced femoral neck fracture: a national survey of 4467 patients. *J Bone Joint Surg Am.* (2013) 95:1693–9. doi: 10.2106/JBJS.L.00836
10. He D, Xue Y, Li Z, Tang Y, Ding H, Yang Z, et al. Effect of depression on femoral head avascular necrosis from femoral neck fracture in patients younger than 60 years. *Orthopedics.* (2014) 37:e244–51. doi: 10.3928/01477447-20140225-56
11. Zielinski SM, Bouwmans CA, Heetveld MJ, Bhandari M, Patka P, Van Lieshout EM, et al. The societal costs of femoral neck fracture patients treated with internal fixation. *Osteoporos Int.* (2014) 25:875–85. doi: 10.1007/s00198-013-2487-2
12. Zhang C, Fang X, Huang Z, Li W, Zhang W, Lee G C. Addition of bone marrow stem cells therapy achieves better clinical outcomes and lower rates of disease progression compared with core decompression alone for early stage osteonecrosis of the femoral head: a systematic review and meta-analysis. *J Am Acad Orthop Surg.* (2020). doi: 10.5435/JAAOS-D-19-00816. [Epub ahead of print].
13. Pan J, Ding Q, Lv S, Xia B, Jin H, Chen D, et al. Prognosis after autologous peripheral blood stem cell transplantation for osteonecrosis of the femoral head in the pre-collapse stage: a retrospective cohort study. *Stem Cell Res Ther.* (2020) 11:83. doi: 10.1186/s13287-020-01595-w
14. Chee CG, Cho J, Kang Y, Kim Y, Lee E, Lee JW, et al. Diagnostic accuracy of digital radiography for the diagnosis of osteonecrosis of the femoral head, revisited. *Acta Radiol.* (2019) 60:969–76. doi: 10.1177/0284185118808083

15. Lee GC, Khoury V, Steinberg D, Kim W, Dalinka M, Steinberg M. How do radiologists evaluate osteonecrosis? *Skeletal Radiol.* (2014) 43:607–14. doi: 10.1007/s00256-013-1803-4
16. Olczak J, Fahlberg N, Maki A, Razavian AS, Jilert A, Stark A, et al. Artificial intelligence for analyzing orthopedic trauma radiographs. *Acta Orthop.* (2017) 88:581–6. doi: 10.1080/17453674.2017.1344459
17. Lindsey R, Daluiski A, Chopra S, Lachapelle A, Mozer M, Sicular S, et al. Deep neural network improves fracture detection by clinicians. *Proc Natl Acad Sci USA.* (2018) 115:11591–6. doi: 10.1073/pnas.1806905115
18. Thian Y L, Li Y, Jagmohan P, Sia D, Chan VEY, Tan RT. Convolutional neural networks for automated fracture detection and localization on wrist radiographs. *Radiology.* (2019) 1:e180001. doi: 10.1148/ryai.2019180001
19. Kitamura G, Chung CY, Moore BE II. Ankle fracture detection utilizing a convolutional neural network ensemble implemented with a small sample, *de novo* training, and multiview incorporation. *J Digit Imaging.* (2019) 32:672–77. doi: 10.1007/s10278-018-0167-7
20. Gale W, Oakden-Rayner L, Carneiro G, Bradley AP, Palmer LJ. *Detecting Hip Fractures with Radiologist-Level Performance Using Deep Neural Networks*. (2017). Available online at: https://arxiv.org/abs/1711.06504 (accessed November 17, 2017).
21. Cheng CT, Ho TY, Lee TY, Chang CC, Chou CC, Chen CC, et al. Application of a deep learning algorithm for detection and visualization of hip fractures on plain pelvic radiographs. *Eur Radiol.* (2019) 29:5469–77. doi: 10.1007/s00330-019-06167-y
22. Chee CG, Kim Y, Kang Y, Lee KJ, Chae HD, Cho J, et al. Performance of a deep learning algorithm in detecting osteonecrosis of the femoral head on digital radiography: a comparison with assessments by radiologists. *AJR Am J Roentgenol.* (2019) 213:155–62. doi: 10.2214/AJR.18.20817
23. Rawall S, Bali K, Upendra B, Garg B, Yadav CS, Jayaswal A. Displaced femoral neck fractures in the young: significance of posterior comminution and raised intracapsular pressure. *Arch Orthop Trauma Surg.* (2012) 132:73–9. doi: 10.1007/s00402-011-1395-1
24. Lapidus LJ, Charalampidis A, Rundgren J, Enocson A. Internal fixation of garden I and II femoral neck fractures: posterior tilt did not influence the reoperation rate in 382 consecutive hips followed for a minimum of 5 years. *J Orthop Trauma.* (2013) 27:386–90; discussion: 390–1. doi: 10.1097/BOT.0b013e318281da6e
25. Riaz O, Arshad R, Nisar S, Vanker R. Serum albumin and fixation failure with cannulated hip screws in undisplaced intracapsular femoral neck fracture. *Ann R Coll Surg Engl.* (2016) 98:376–9. doi: 10.1308/rcsann.2016.0124
26. Campenfeldt P, Hedstrom M, Ekstrom W, Al-Ani AN. Good functional outcome but not regained health related quality of life in the majority of 20-69 years old patients with femoral neck fracture treated with internal fixation: a prospective 2-year follow-up study of 182 patients. *Injury.* (2017) 48:2744–53. doi: 10.1016/j.injury.2017.10.028
27. Ly HV. Treatment of femoral neck fractures in young adults. *J Bone Joint Surg Am.* (2008) 90:2254–66.
28. Yoon BH, Mont MA, Koo KH, Chen CH, Cheng EY, Cui Q, et al. The 2019 revised version of association research circulation osseous staging system of osteonecrosis of the femoral head. *J Arthroplasty.* (2020) 35:933–40. doi: 10.1016/j.arth.2019.11.029
29. Lu YZS, Chu PW, Arenson Rl. An update survey of academic radiologists' clinical productivity. *J Am Coll Radiol.* (2008) 5:817–26. doi: 10.1016/j.jacr.2008.02.018
30. Berlin L. Liability of interpreting too many radiographs. *AJR Am J Roentgenol.* (2000) 175:17–22. doi: 10.2214/ajr.175.1.1750017
31. Urakawa T, Tanaka Y, Goto S, Matsuzawa H, Watanabe K, Endo N. Detecting intertrochanteric hip fractures with orthopedist-level accuracy using a deep convolutional neural network. *Skeletal Radiol.* (2019) 48:239–44. doi: 10.1007/s00256-018-3016-3
32. Li J, Wang M, Zhou J, Zhang H, Li L. Finite element analysis of different screw constructs in the treatment of unstable femoral neck fractures. *Injury.* (2020) 51:995–1003. doi: 10.1016/j.injury.2020.02.075
33. Ai ZS, Gao YS, Sun Y, Liu Y, Zhang CQ, Jiang CH. Logistic regression analysis of factors associated with avascular necrosis of the femoral head following femoral neck fractures in middle-aged and elderly patients. *J Orthop Sci.* (2013) 18:271–6. doi: 10.1007/s00776-012-0331-8
34. Gregersen M, Krogshede A, Brink O, Damsgaard EM. Prediction of reoperation of femoral neck fractures treated with cannulated screws in elderly patients. *Geriatr Orthop Surg Rehabil.* (2015) 6:322–7. doi: 10.1177/2151458515614369
35. Florschutz AV, Langford JR, Haidukewych GJ, Koval KJ. Femoral neck fractures: current management. *J Orthop Trauma.* (2015) 29:121–9. doi: 10.1097/BOT.0000000000000291
36. Cui S, Zhao L, Wang Y, Dong Q, Ma J, Wang Y, et al. Using Naive Bayes Classifier to predict osteonecrosis of the femoral head with cannulated screw fixation. *Injury.* (2018) 49:1865–70. doi: 10.1016/j.injury.2018.07.025
37. Titano JJ, Badgeley M, Schefflein J, Pain M, Su A, Cai M, et al. Automated deep-neural-network surveillance of cranial images for acute neurologic events. *Nat Med.* (2018) 24:1337–41. doi: 10.1038/s41591-018-0147-y
38. Badgeley MA, Zech JR, Oakden-Rayner L, Glicksberg BS, Liu M, Gale W, et al. Deep learning predicts hip fracture using confounding patient and healthcare variables. *NPJ Digit Med.* (2019) 2:31. doi: 10.1038/s41746-019-0105-1
39. Zech JR, Badgeley MA, Liu M, Costa AB, Titano JJ, Oermann EK. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study. *PLoS Med.* (2018) 15:e1002683. doi: 10.1371/journal.pmed.1002683
40. Stockton DJ, O'hara LM, O'hara NN, Lefaivre KA, O'brien PJ, Slobogean GP. High rate of reoperation and conversion to total hip arthroplasty after internal fixation of young femoral neck fractures: a population-based study of 796 patients. *Acta Orthop.* (2019) 90:21–25. doi: 10.1080/17453674.2018.1558380
41. Microsurgery Department of the Orthopedics Branch of the Chinese Medical Doctor A, Group from The O, Bone Defect Branch of the Chinese Association of R, Reconstructive S, Microsurgery, Reconstructive Surgery Group of the Orthopedics Branch of the Chinese Medical A. Chinese guideline for the diagnosis and treatment of osteonecrosis of the femoral head in adults. *Orthop Surg.* (2017) 9:3–12. doi: 10.1111/os.12302
42. Larson E, Jones LC, Goodman SB, Koo KH, Cui Q. Early-stage osteonecrosis of the femoral head: where are we and where are we going in year 2018? *Int Orthop.* (2018) 42:1723–8. doi: 10.1007/s00264-018-3917-8
43. Kumar NM, Netto CDC, Schon LC, Fritz J. Metal artifact reduction magnetic resonance imaging around arthroplasty implants: the negative effect of long echo trains on the implant-related artifact. *Investigative Radiology.* (2017) 52:310–6. doi: 10.1097/RLI.0000000000000350
44. Atilla B, Bakircioglu S, Shope AJ, Parvizi J. Joint-preserving procedures for osteonecrosis of the femoral head. *EFORT Open Rev.* (2019) 4:647–58. doi: 10.1302/2058-5241.4.180073
45. Zhang CQ, Sun Y, Chen SB, Jin DX, Sheng JG, Cheng XG, Xu J, Zeng BF. Free vascularised fibular graft for posttraumatic osteonecrosis of the femoral head in teenage patients. *J Bone Joint Surg Br.* (2011). 93:1314–19. doi: 10.1302/0301-620X.93B10.26555
46. Xian H, Luo D, Wang L, Cheng W, Zhai W, Lian K, et al. Platelet-Rich plasma-incorporated autologous granular bone grafts improve outcomes of post-traumatic osteonecrosis of the femoral head. *J Arthroplasty.* (2020) 35:325–30. doi: 10.1016/j.arth.2019.09.001
47. Algarni A D, Al Moallem H M. Clinical and radiological outcomes of extracorporeal shock wave therapy in early-stage femoral head osteonecrosis. *Adv Orthop.* (2018) 2018:7410246. doi: 10.1155/2018/7410246
48. Wang CJ, Wang FS, Huang CC, Yang KD, Weng LH, Huang HY. Treatment for osteonecrosis of the femoral head: comparison of extracorporeal shock waves with core decompression and bone-grafting. *J Bone Joint Surg Am.* (2005) 87:2380–7. doi: 10.2106/00004623-200511000-00002
49. Yu X, Zhang D, Chen X, Yang J, Shi L, Pang Q. Effectiveness of various hip preservation treatments for non-traumatic osteonecrosis of the femoral head:

a network meta-analysis of randomized controlled trials. *J Orthop Sci.* (2018) 23:356–64. doi: 10.1016/j.jos.2017.12.004

50. Jo WL, Lee YK, Ha YC, Kim TY, Koo KH. Delay of total hip arthroplasty to advanced stage worsens post-operative hip motion in patients with femoral head osteonecrosis. *Int Orthop.* (2018) 42:1599–603. doi: 10.1007/s00264-018-3952-5
51. Chen SB, Hu H, Gao YS, He HY, Jin DX, Zhang CQ. Prevalence of clinical anxiety, clinical depression and associated risk factors in chinese young and middle-aged patients with osteonecrosis of the femoral head. *PLoS ONE.* (2015) 10:e0120234. doi: 10.1371/journal.pone.0120234

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Applied Mathematics and Statistics

ORIGINAL RESEARCH
published: 06 November 2020
doi: 10.3389/fams.2020.553000

Data-Driven Supervised Learning for Life Science Data

Maximilian Münch[1,2], *Christoph Raab*[1,3], *Michael Biehl*[2] *and Frank-Michael Schleif*[1,4,5]*

[1]Department of Computer Science, University of Applied Sciences Wuerzburg-Schweinfurt, Wuerzburg, Germany, [2]Bernoulli Insititute for Mathematics, Computer Science and Artificial Intelligence, University of Groningen, Groningen, Netherlands, [3]Bielefeld University, CITEC Centre of Excellence, Bielefeld, Germany, [4]University of Applied Sciences Mittweida, Computational Intelligence Research Group, Mittweida, Germany, [5]The University of Birmingham, Edgbaston, Birmingham, United Kingdom

Life science data are often encoded in a non-standard way by means of alpha-numeric sequences, graph representations, numerical vectors of variable length, or other formats. Domain-specific or data-driven similarity measures like alignment functions have been employed with great success. The vast majority of more complex data analysis algorithms require fixed-length vectorial input data, asking for substantial preprocessing of life science data. Data-driven measures are widely ignored in favor of simple encodings. These preprocessing steps are not always easy to perform nor particularly effective, with a potential loss of information and interpretability. We present some strategies and concepts of how to employ data-driven similarity measures in the life science context and other complex biological systems. In particular, we show how to use data-driven similarity measures effectively in standard learning algorithms.

Keywords: similarity based learning, non-metric learning, kernel methods, indefinite learning, gershgorin circles

OPEN ACCESS

Edited by:
Andre Gruning,
University of Surrey, United Kingdom

Reviewed by:
Anastasiia Panchuk,
Institute of Mathematics (NAN Ukraine), Ukraine
Axel Hutt,
Inria Nancy - Grand-Est Research Centre, France

****Correspondence:***
Frank-Michael Schleif
frank-michael.schleif@fhws.de

Specialty section:
This article was submitted to Dynamical Systems, a section of the journal Frontiers in Applied Mathematics and Statistics

Received: *17 April 2020*
Accepted: *24 September 2020*
Published: *06 November 2020*

Citation:
Münch M, Raab C, Biehl M and Schleif F-M (2020) Data-Driven Supervised Learning for Life Science Data. Front. Appl. Math. Stat. 6:553000. doi: 10.3389/fams.2020.553000

INTRODUCTION

Life sciences comprise a broad research field with challenging questions in domains such as (bio-) chemistry, biology, environmental research, or medicine. Not only recent technological developments allow the generation of large, high dimensional and very complex data sets in these fields, but also, the structure of the measured data representing an object of interest is often challenging. The data may be compositional, such that classical vectorial functions are not easy to apply and could also be very heterogeneous by combining different measurement sources. Accordingly, new strategies and algorithms are needed to cope with the complexity of life science applications. In general, it is a promising way to reflect characteristic data properties in the employed data processing pipeline. This typically leads to increased performance in tasks such as clustering, classification, and non-linear regression, which are commonly addressed by machine learning methods. One possible way to achieve this is to adapt the used metric according to the underlying data properties and application, respectively [1]. Basically, all machine learning and data analysis algorithms employ the comparison of objects referred to as similarities or dissimilarities, or more general as proximities. Hence, the representation of these proximities is a crucial part. These measures enter the modeling algorithm either by means of distance measures, e.g., in the standard k-means algorithm or by inner products as employed in the famous support vector machine (SVM) [2]. The calculation of these proximities is typically based on a vectorial representation of the input data. If the used machine learning approach is solely based on proximities, a vectorial representation is in general not needed, but the pairwise proximity values are sufficient. This approach is referred to as similarity-based learning, where the data are represented by metric pairwise similarities only.

TABLE 1 | List of commonly used non-metric proximity measures in various domains.

Measure	Application field
Dynamic Time Warping (DTW) (6)	Time series or spectral alignment
Inner distance (7)	Shape retrieval e.g., in robotics
Compression distance (8)	Generic used also for text analysis
Smith Waterman Alignment (5)	Bioinformatics
Divergence measures (9)	Spectroscopy and audio processing
Generalized Lp norm (10)	Time series analysis
Non-metric modified Hausdorff (11)	Template matching
(Domain-specific) alignment score (12)	Mass spectrometry

We can distinguish similarities, indicating how close or similar two items are to each other and dissimilarities in the opposite sense. In the following, we expect that these proximities are at least symmetric, but do not necessarily obey metric properties. See e.g., [3] for an extended discussion.

Non-metric measures are common in many disciplines and occasionally entail so-called non positive semi-definite (non-psd) kernels if a similarity measure is used. This is particularly interesting because many classical learning algorithms can be kernelized [4], but are still expecting a psd measure. As we will outline in this paper, we can be more flexible in the use of a proximity measure as long as some basic assumptions are fulfilled. In particular, it is not necessary, for many real-world life science data, to restrict the analysis pipeline to a vectorial Euclidean representation of the data.

In the various domains like spectroscopy, high throughput sequencing, or medical image analysis, domain-specific measures have been designed and effectively used. Classical sequence alignment functions (e.g., Smith-Waterman [5]) produce non-metric proximity values. There are many more examples and use cases, as listed in **Table 1** and detailed later on.

Multiple authors argue that the non-metric part of the data contains valuable information and should not be removed [13, 14]. In this work, we highlight recent achievements in the field of similarity-based learning for non-metric measures and provide conceptual and experimental evidence on a variety of scenarios that non-metric measures are legal and effective tools in analyzing such data. We argue that a restriction to *mathematically* more convenient, but from the *data perspective* unreliable, measures are not needed anymore.

Along this line, we first provide an introduction to similarity-based learning in non-metric spaces. Then we provide an outline and discussion of preprocessing techniques, which can be used to implement a non-metric similarity measure within a classical analysis pipeline. In particular, we highlight a novel advanced shift correction approach. Here we extend prior work published by the authors in 15, which is substantially extended by novel theoretical findings (**Section 2.4**, in particular, the eigenvalue approximation via Gershgorin), experimental results (**Section 3**, with additional experiments and datasets), and an extended discussion. The highlights of this paper:

- We provide a broad study of life science data encoded by proximities only.
- We reveal the limitations of former encodings used to enable standard kernel methods.
- We derive a novel encoding concept widely preserving the data's desired properties while showing considerable performance.
- We improve the efficiency of the encodings using an approximation concept not considered so far with almost no loss of performance in the classification process.

In the experiments, we show the effectiveness of appropriately preprocessed non-metric measures in a variety of real-life use cases. We conclude by a detailed discussion and provide practical advice in applying non-metric proximity measures in the analysis of life science data.

MATERIALS AND METHODS

Notation and Basic Concepts

Given a set of N data items (like N spectral measurements or N sequences), their pairwise proximity (similarity or dissimilarity) measures can be conveniently summarized in a $N \times N$ proximity matrix. These proximities can be very generic in practical applications, but most often come either in the form of symmetric similarities or dissimilarities only. Focusing on one of the respective representation forms is not a substantial restriction. As outlined in 16, a conversion from dissimilarities to similarities is cheap regarding to computational costs. Also, an out of sample extension can be easily provided. In the following, we will refer to similarity and dissimilarity type proximity matrices as S and D, respectively. These notions enter into models by means of proximity or score functions $f(x, x') \in \mathbb{R}$ where x and x' are the compared objects (both are data items). The objects x, x' may exist in a d-dimensional vector space, so that $x \in \mathbb{R}^d$, but can also be given without an explicit vectorial representation, e.g., as biological sequences.

As outlined in 17, the majority of analysis algorithms are applicable only in a tight mathematical setting. In particular, it is expected that $f(x, x')$ obeys a variety of properties. If $f(x, x')$ is a dissimilarity measure, it is often assumed to be a metric measure. Many algorithms become invalid or do not converge if $f(x, x')$ does not fulfill metric properties.

For example, the support vector machine formulation [18] no longer leads to a convex optimization problem [19] when the given input data is non-metric. Prominent solvers, such as sequential minimization (SMO), will converge to only a local optimum [20, 21] and other kernel algorithms may not converge at all. Accordingly, dedicated strategies for non-metric data are very desirable.

The score function $f(x, x')$ could violate the metric properties to different degrees. In general it is at least expected that $f(x, x')$ obeys the symmetry property such that $f(x, x') = f(x', x)$. In general, this property is a fundamental condition, because a large number of algorithms become meaningless for asymmetric data. We will also make this assumption. In the considered cases, the proximities are either already symmetric or can be symmetrized without expecting a negative impact. While symmetry is a reasonable assumption, the triangle inequality is frequently violated, proximities become negative, or self-dissimilarities are not zero. Such violations can be attributed to noise as addressed in 22 or are a natural property of the proximity function *f*.

If noise is the source, often a simple eigenvalue correction [23] can be used, although this can become costly for large datasets. As we will see later on, the noise may cause eigenvalue contributions close to zero. A simple way to eliminate these contributions is to calculate a low-rank approximation of the matrix, which can be realized with small computational cost [24, 25]. In particular, the small eigenvalues could become negative, also leading to problems in the use of classical learning algorithms. A recent analysis of the possible sources of negative eigenvalues is provided in 26. Such an analysis is particularly helpful in selecting the appropriate eigenvalue correction method applied to the proximity matrix. Non-metric proximity measures are part of the daily work in various domains [27]. An area, frequently applying such such non-metric proximity measures, is the field of bioinformatics, spectroscopy, or alike, where classical sequence alignment algorithms (e.g., Smith-Waterman - [5]) produce non-metric proximity values. For such data, some authors argue that the non-metric part of the data contains valuable information and should not be removed [13]. In particular, this is the motivation for our work. Evaluating such data with machine learning models typically asks for discriminative models. In particular, for classification tasks, a separating plane has to be determined in order to separate the given data according to their classes. However, in practice, a linear plane in the original feature space is rarely separating two classes of such complexity. A common generalization is to map the training vectors x_i into a higher dimensional space by the function ϕ. In this space, it is expected that the machine learning model finds a linear separating hyperplane with a maximal margin. The principle behind such a so-called kernel function is explained in more detail in **Section 2.1.1**. In our setting, the mapping is provided by some data-driven similarity function, which, however, may not lead to a psd kernel and hence has to be preprocessed (for more details, see **Section 2.1.4**). As a primal representation, we will focus on similarities because the wide majority of algorithms is specified in the kernel space. A brief introduction is given in the following section.[1]

Kernels and Kernel Functions

Let $\mathcal{X}$ be a collection of N objects x_i, $i = 1, 2, \ldots, N$, in some input space. Further, let $\phi : \mathcal{X} \mapsto \mathscr{H}$ be a mapping of patterns from $\mathcal{X}$ to a high-dimensional or infinite-dimensional Hilbert space $\mathscr{H}$ equipped with the inner product $\langle \cdot, \cdot \rangle_{\mathscr{H}}$. The transformation ϕ is, in general, a non-linear mapping to a high-dimensional space $\mathscr{H}$ and may commonly not be given in an explicit form. Instead of this, a kernel function $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is given which encodes the inner product in $\mathscr{H}$. The kernel k is a positive (semi) definite function such that $k(x, x') = \langle \phi(x)^\top, \phi(x') \rangle$ for any $x, x' \in \mathcal{X}$. The matrix $K := \Phi^\top \Phi$ is an $N \times N$ kernel matrix derived from the training data, where $\Phi : [\phi(x_1), \ldots, \phi(x_N)]$ is a matrix of images (column vectors) of the training data in $\mathscr{H}$. The motivation for such an embedding comes with the hope that the non-linear transformation of input data into higher dimensional $\mathscr{H}$ allows for using linear techniques in $\mathscr{H}$. Kernelized methods process the embedded data points in a feature space utilizing only the inner products $\langle \cdot, \cdot \rangle_{\mathscr{H}}$ (kernel trick) [28], without the need to calculate ϕ explicitly. The specific kernel function can be very generic, but in general, the kernel is expected to fulfill Mercer conditions [28]. Most prominent are the linear kernel with $k(x, x') = x^\top x'$ as the Euclidean inner product or the RBF kernel $k(x, x') = \exp(-(\|x - x'\|^2/2\sigma^2))$, with σ as a free parameter.

Support Vector Machine

In this paper, we address data-driven supervised learning; accordingly, our focus is primal on a domain-specific representation of the data by means of a generic similarity measure. There are many approaches for similarity-based learning and, in particular, kernel methods [28]. We will evaluate our data-driven encodings employing the support vector machine (SVM) as a state of the art supervised kernel method.

Let $x_i \in X, i \in \{1, \ldots, N\}$ be training points in the input space X , with labels $y_i \in \{-1, 1\}$, representing the class of each point.[2] The input space X is often considered to be $\mathbb{R}^d$ but can be any suitable space due to the kernel trick. For a given positive penalization term C, the SVM is the minimum of the following regularized empirical risk functional.

$$\min_{\omega, \xi, b} \frac{1}{2} \omega^\top \omega + C \sum_{i=1}^{M} \xi_i \tag{1}$$

subject to $y_i(\omega^\top \phi(x_i) + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$. Here ω is the parameter vector of a separating hyperplane and b a bias term. The variables ξ are so-called slack variables. The goal is to find a hyperplane that correctly separates the data while maximizing the sum of distances to the closest positive and negative points (the margin). The parameter C controls the weight of the classification errors ($C = \infty$ in the separable case). Details can be found in 28.

In case of a positive semi-definite kernel function without metric violations, the underlying optimization problem is easily solved using, e.g., the Sequential Minimal Optimization Algorithm [20]. The objective of a SVM is to derive a model from the training set, which predicts class labels of unclassified feature sets in the test data. The decision function is given as:

$$f(x) = \sum_{i=1}^{N} y_i \alpha_i k(x_i, x) + b,$$

where the α_i are the optimized Lagrange parameters of the dual formulation of **Eq. 1**. In case of a non-psd kernel function, the optimization problem of a SVM is no longer convex, but only a local optimum is obtained [19, 21]. As a result, the trained SVM model can become inaccurate and incorrect. However, as we will

[1] For data given as dissimilarity matrix, the associated similarity matrix can be obtained, in a non-destructive way, by double centering (17) of the dissimilarity matrix. $S = -JDJ/2$ with $J = (I - 11^\top/N)$, identity matrix I and vector of ones ?.

[2] In case of more than two classes we use the one vs all approach.

see in **Section 2.1.4**, there are several methods to handle non-psd kernel matrices within a classical SVM.

Representation in the Krein Space

A Krein space is an *indefinite* inner product space endowed with a Hilbertian topology. Let $\mathcal{K}$ be a real vector space. An inner product space with an indefinite inner product $\langle \cdot, \cdot \rangle_{\mathcal{K}}$ on $\mathcal{K}$ is a bi-linear form where all $f, g, h \in \mathcal{K}$ and $\alpha \in \mathbb{R}$ obey the following conditions:

- Symmetry : $\langle f, g \rangle_{\mathcal{K}} = \langle g, f \rangle_{\mathcal{K}}$;
- linearity : $\langle \alpha f + g, h \rangle_{\mathcal{K}} = \alpha \langle f, h \rangle_{\mathcal{K}} + \langle g, h \rangle_{\mathcal{K}}$;
- $\langle f, g \rangle_{\mathcal{K}} = 0$ implies $f = 0$

An inner product is positive semi definite if $\forall f \in \mathcal{K}$, $\langle f, f \rangle_{\mathcal{K}} \geq 0$, negative definite if $\forall f \in \mathcal{K}$, $\langle f, f \rangle_{\mathcal{K}} < 0$, otherwise it is indefinite. A vector space $\mathcal{K}$ with inner product $\langle \cdot, \cdot \rangle_{\mathcal{K}}$ is called an inner product space.

An inner product space $(\mathcal{K}, \langle \cdot, \cdot \rangle_{\mathcal{K}})$ is a Krein space if we have two Hilbert spaces $\mathscr{H}_+$ and $\mathscr{H}_-$ spanning $\mathcal{K}$ such that $\forall f \in \mathcal{K}$ we have $f = f_+ + f_-$ with $f_+ \in \mathscr{H}_+$ and $f_- \in \mathscr{H}_-$ and $\forall f, g \in \mathcal{K}$, $\langle f, g \rangle_{\mathcal{K}} = \langle f_+, g_+ \rangle_{\mathscr{H}_+} - \langle f_-, g_- \rangle_{\mathscr{H}_-}$.

As outlined before, indefinite kernels are typically observed by means of domain-specific non-metric similarity functions (such as alignment functions used in biology [29]), by specific kernel functions - e.g., the Manhattan kernel $k(x, x') = -\|x - x'\|_1$, tangent distance kernel [30] or divergence measures, plugged into standard kernel functions [9]. A finite-dimensional Krein-space is a so-called pseudo-Euclidean space.

Given a symmetric *dissimilarity* matrix with zero diagonal, an embedding of the data in a pseudo-Euclidean vector space determined by the eigenvector decomposition of the associated similarity matrix S is always possible [31] - as mentioned above, e.g., by a prior double centering. Given the eigendecomposition of S: $S = U \Lambda U^\top$, we can compute the corresponding vectorial representation V in the pseudo-Euclidean space by

$$V = U_{p+q+z} \left| \Lambda_{p+q+z} \right|^{1/2} \tag{2}$$

where Λ_{p+q+z} consists of p positive, q negative non-zero eigenvalues and z zero eigenvalues. U_{p+q+z} consists of the corresponding eigenvectors. The triplet (p, q, z) is also referred to as the signature of the pseudo-Euclidean space. A detailed presentation of similarity and dissimilarity measures and mathematical aspects of metric and non-metric spaces is provided in 17, 32, 33.

Indefinite Proximity Functions

Proximity functions can be very generic but are often restricted to fulfill metric properties to simplify the mathematical modeling and especially the parameter optimization. In 32, a large variety of such measures was reviewed and basically most common methods nowadays make still use of metric properties. While this appears to be a reliable strategy, researchers in the field of e.g., psychology [34, 35], vision [14, 26, 36, 37] and machine learning [13, 38] have criticized this restriction as inappropriate in multiple cases. In fact, in 38 was shown that many real-life problems are better addressed by proximity measures, which are not restricted to be metric.

The triangle inequality is frequently violated, if we consider object comparisons in daily life problems, like the comparisons of text documents, biological sequence data, spectral data or graphs [23, 39, 40]. These data are inherently compositional and a representation as explicit (vectorial) features leads to information loss. As an alternative, tailored dissimilarity measures such as pairwise alignment functions, kernels for structures, or other domain-specific similarity and dissimilarity functions can be used as an interface to the data [41, 42]. Also for vectorial data, non-metric proximity measures are quite common in some disciplines. An example of this type is the use of divergence measures [9, 43, 44] which are very popular for spectral data analysis in chemistry, geo- and medical sciences [45–49], and are not metric in general. Also the popular Dynamic Time Warping (DTW) [6] algorithm provides a non-metric alignment score, which is often used as a proximity measure between two one-dimensional functions of different lengths. In image processing and shape retrieval, indefinite proximities are often obtained by means of the inner distance. This measure specifies the dissimilarity between two objects, which are represented by their shape only. Thereby, several seeding points are used and the shorted paths *within* the shape are calculated in contrast to the Euclidean distance between the landmarks. Further examples can be found in physics where problems of the special relativity theory or other research topics naturally lead to indefinite spaces [50].

A list of non-metric proximity measures is provided in **Table 1** and some are exemplarily illustrated in **Figures 1** and **2**. Most of these measures are very popular but often violate the symmetry or triangle inequality condition or both. Hence many standard proximity-based machine learning methods like kernel methods are not easily accessible for these data.

Eigenspectrum Corrections

Although native models for indefinite learning are available (see e.g., [27, 51, 52]), they are not frequently used. This is mainly due to three reasons: 1) the proposed algorithms have in general, quadratic or cubic complexity [53], 2) the obtained models are non-sparse [54], and 3) the methods are complicated to implement [27, 55]. Considering the wide spread of machine learning frameworks, it would be very desirable to use the therein implemented algorithms - like an efficient support vector machine, instead of having the burden to implement another algorithm, and in general another numerical solver. Therefore, we focus on eigenspectrum corrections, which can be effectively done in a large number of frameworks without much effort.

A natural way to address the indefiniteness problem and to obtain a psd similarity matrix is to correct the eigenspectrum of the original similarity matrix S. Popular strategies include eigenvalue correction by *flipping*, *clipping*, *squaring*, and *shifting*. The non-psd similarity matrix S is decomposed by an eigendecomposition: $S = U \Lambda U^\top$, where U contains the eigenvectors of S and Λ contains the corresponding eigenvalues λ_i. Now, the eigenvalues in Λ can be manipulated to eliminate all negative parts. After the correction, the matrix can be reconstructed, now being psd.

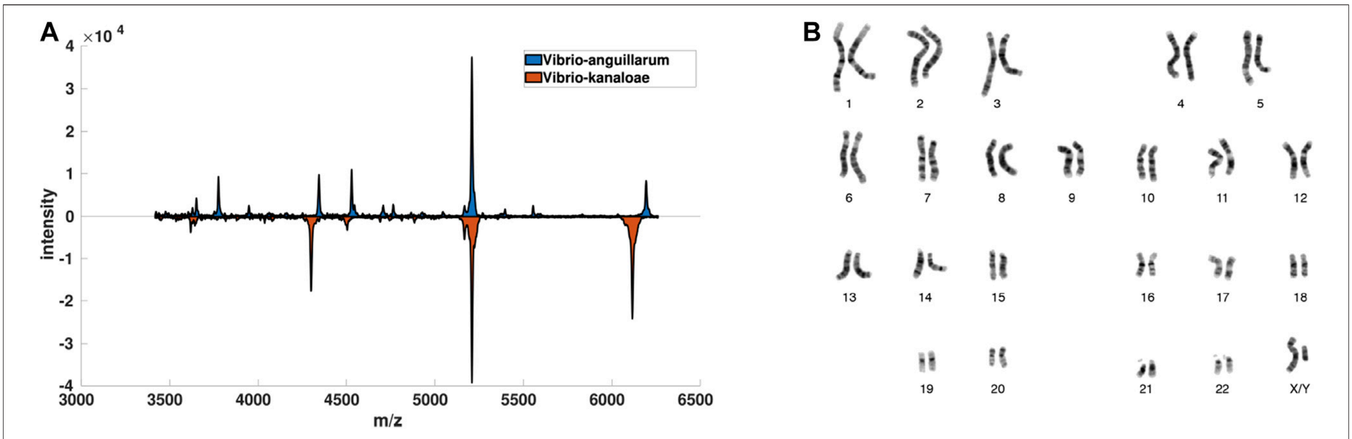


FIGURE 1 | Visualization of data-driven data description scenarios. In **(A)** for some Vibrio bacteria and in **(B)** for Chromosome data. Both datasets are used in the experiments.

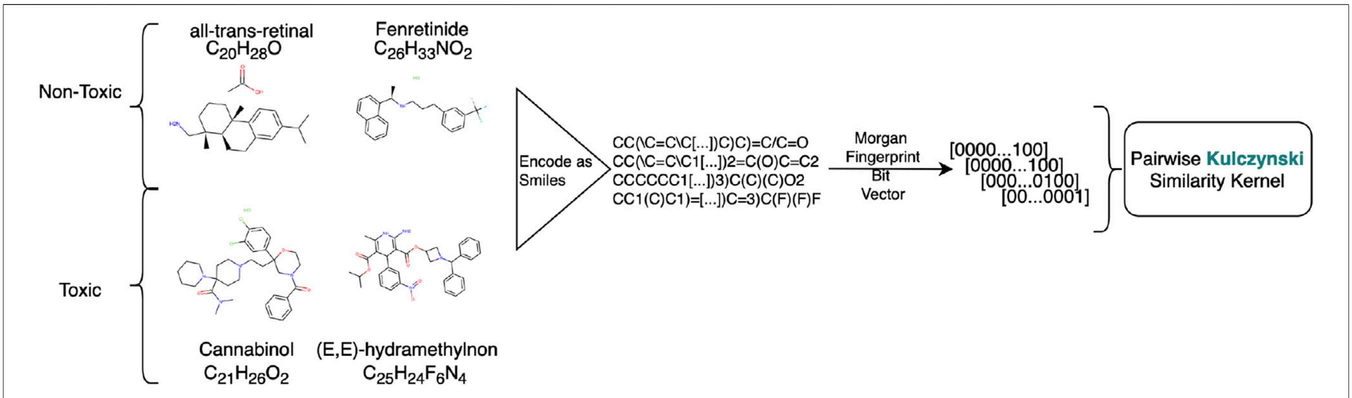


FIGURE 2 | Preprocessing workflow for creating the Tox-21 datasets. Chemicals represented as SMILE codes are translated to Morgan Fingerprints. The kernel is created by using an application related pairwise similarity measure on the Morgan Fingerprints, in this case so-called *Kulczynski*.

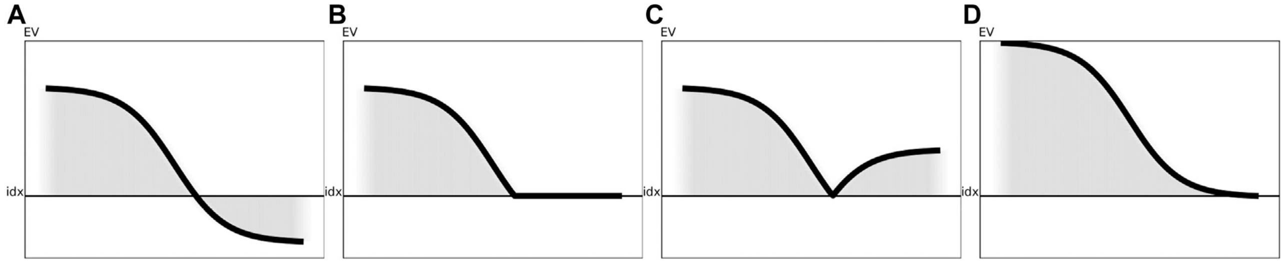


FIGURE 3 | Visualization of the various preprocessing techniques for a generic eigenspectrum as obtained from a generic similarity matrix. The black line illustrates the impact of the respective correction method on the eigenspectrum without reordering of the eigenvalues. (A) Visualization of a sample eigenspectrum with pos./neg. eigenvalues. (B) Preprocessing of the eigenspectrum from **Figure 3A** using clip. (C) Preprocessing of the eigenspectrum from **Figure 3A** using flip. (D) Preprocessing of the eigenspectrum from **Figure 3A** using shift.

Clip Eigenvalue Correction

All negative eigenvalues in Λ are set to 0 (see **Figure 3B**). The spectrum clip leads to the nearest psd matrix S in terms of the Frobenius norm [56]. Such a correction can be achieved by an eigendecomposition of the matrix S, a clipping operator on the eigenvalues, and the subsequent reconstruction. This operation has a complexity of $\mathcal{O}(N^3)$. The complexity might be reduced by either a low-rank approximation or the approach shown by 22 with roughly quadratic complexity.

Flip Eigenvalue Correction

All negative eigenvalues in Λ are set to $\lambda_i := |\lambda_i| \, \forall i$, which at least keeps the absolute values of the negative eigenvalues and keeps potentially relevant information [17]. This operation can be calculated with $\mathcal{O}(N^3)$ or $\mathcal{O}(N^2)$ if low-rank approaches are used. Flip is illustrated in **Figure 3C**.

Square Eigenvalue Correction

All negative eigenvalues in Λ are set to $\lambda_i := \lambda_i^2 \, \forall i$ which amplifies large and very small eigenvalues. The square eigenvalue correction can be achieved by matrix multiplication [57] with $\approx \mathcal{O}(N^{2.8})$.

Classical Shift Eigenvalue Correction

The shift operation was already discussed earlier by different researchers [58] and modifies Λ such that $\lambda_i := \lambda_i - \min_{ij}\Lambda \, \forall i$. The classical shift eigenvalue correction can be accomplished with linear costs if the smallest eigenvalue $\lambda_{\min}$ is known. Otherwise, some estimator for $\lambda_{\min}$ is needed. A few estimators for this purpose have been suggested: analyzing the eigenspectrum on a subsample, making a reasonable guess, or using some low-rank eigendecomposition. In our approach, we suggest employing a power iteration method, for example the *von Mises* approach, which is fast and accurate [59] or using the Gershgorin circle theorem [60, 61].

A spectrum shift enhances all the self-similarities and, therefore, the eigenvalues by the amount of λ_{min} and does not change the similarity between any two different data points. However, it may also increase the intrinsic dimensionality of the data space and amplify noise contributions, as shown in **Figure 3D**. As already mentioned by 23, small eigenvalue contributions could be linked to noise in the original data. If now an eigencorrection step amplifies tiny eigenvalues, this can be considered as a noise amplification.

Limitations

Multiple approaches have been suggested to correct a similarity matrix's eigenspectrum to obtain a psd matrix [17, 27]. Most approaches modify the eigenspectrum in a radical way and are also costly due to an involved cubic eigendecomposition. In particular, the flip, square and clip operator have an apparent strong impact. The flip operator affects all negative eigenvalues by changing the sign and this will additionally lead to a reorganization of the eigenvalues. The square operator is similar to flip but additionally emphasizes large eigencontributions while fading out eigenvalues below 1. The clip method is useful in case of noise; it may also remove valuable contributions. The clip operator only *removes* eigenvalues, but generally keeps the majority of the eigenvalues unaffected. The classical shift is another alternative operator changing only the diagonal of the similarity matrix leading to a shift of the whole eigenspectrum by the provided offset. This may also lead to reorganizations of the eigenspectrum due to new non-zero eigenvalue contributions. While this simple approach seems to be very reasonable, it has the significant drawback that all (!) eigenvalues are shifted, which also affects small or even 0 eigenvalue contributions. While 0 eigenvalues have no contribution in the original similarity matrix, they are artificially upraised by the classical shift operator. This may introduce a large amount of noise in the eigenspectrum, which could potentially lead to substantial numerical problems for employed learning algorithms, for example, kernel machines. If we consider the number of non-vanishing eigenvalues as a rough estimate of the intrinsic dimension of the data, a classical shift will increase this value. This may accelerate the curse of dimension problem on this modified data [62].

Advanced Shift Correction

To address the aforementioned challenges, we suggest an alternative formulation of the shift correction, subsequently referred to as advanced shift. In particular, we would like to keep the original eigenspectrum structure and aim for a sub-cubic eigencorrection. As mentioned in **Section 2.3** the classical shift operator introduces noise artifacts for small eigenvalues. In the advanced shift procedure, we will remove these artificial contributions by a null space correction. This is particularly effective if non-zero, but small eigenvalues are also taken into account. Accordingly, we apply a low-rank approximation of the similarity matrix as an additional preprocessing step. The procedure is summarized in **Algorithm 1**.

The first part of the algorithm applies a low-rank approximation on the input similarities S using a restricted SVD or other technique [63]. If the number of samples $N \leq 1000$, then the rank parameter $k = 30$, otherwise $k = 100$.[3] The shift parameter λ is calculated on the low-rank approximated matrix, using a *von Mises* or power iteration [59] to determine the respective largest negative eigenvalue of the matrix. As shift parameter, we use the absolute value of λ for further steps. This procedure provides an accurate estimate of the largest negative eigenvalue, instead of making an educated guess as frequently suggested [51]. This is particularly relevant because the scaling of the eigenvalues can be very different between the various datasets, which may lead to an ineffective shift (still with negative eigenvalues left) if the guess is incorrect. The basis **B** of the nullspace is calculated, again by a restricted SVD. The nullspace matrix **N** is obtained by calculating a product of **B**. Due to the low-rank approximation, we ensure that small eigenvalues, which are indeed close to 0 due to noise, are shrunk to 0 [64]. In the final step, the original S or the respective low-rank approximated matrix $\hat{S}$ is shifted by the largest negative eigenvalue λ that is determined by *von Mises* iteration. By combining the shift with the nullspace matrix **N** and the identity matrix I, the whole matrix will be affected by the shift and not only the diagonal matrix. Finally, the doubled shift factor 2 ensures that the largest negative eigenvalue $\hat{\lambda}^*$ of the new matrix $\hat{S}^*$ will not become 0, but are kept as a contribution.

Complexity: The advanced shift approach shown in **Algorithm 1** is comprised of various subtasks with different complexities. The low-rank approximation can be achieved with $\mathcal{O}(N^2)$ as well as the nullspace approximation. The shift parameter is calculated by *von Mises* iteration with $\mathcal{O}(N^2)$. Since **B** is a rectangular $N \times k$ matrix, the matrix **N** can be calculated with $\mathcal{O}(N^2)$. The final

[3] The settings for k are taken as a rule of thumb without further fine-tuning.

Algorithm 1 Advanced shift eigenvalue correction.

```
Advanced_shift($S, k$)
if approximate to low rank then
    $S$ := LowRankApproximation($S, k$)
end if
$\lambda$ := |ShiftParameterDetermination($S$)|
$\mathbf{B}$ := NullSpace($S$)
$\mathbf{N}$ := $\mathbf{B} \cdot \mathbf{B}'$
$S^{*}$ := $S + 2 \cdot \lambda \cdot (I - \mathbf{N})$
return $S^{*}$
```

eigenvalue correction to obtain $\widehat{S}^{*}$ is also $\mathcal{O}(N^2)$. In summary, the low-rank advanced shift eigenvalue correction can be achieved with $\mathcal{O}(N^2)$ operations.

Efficient Approximation of the Smallest Eigenvalue

An alternative method to accelerate the estimation of the shift parameter λ is to approximate the region in which the smallest eigenvalue can be found. The identification of this region can be efficiently achieved by the Gershgorin circle theorem [60, 61]. Let $S = (s_{ij})$ be a square matrix ($N \times N$) and $r_i = \sum_{j \neq i} |s_{ij}|$ the row sums of this matrix. Then, within the Gershgorin circle theorem, one may define a disc D_i in the complex plane with center s_{ii} and radius r_i. In 61, it is shown why this can be employed to obtain a valid estimate of the eigenvalues of S. With $D_i = \{z \in \mathbb{C} \quad |z - s_{ii}| \leq r_i\}$, we obtain ranges that contain the eigenvalues of S: $[s_{ii} - r_i, s_{ii} + r_i]$. Hence one only has to calculate N row-sums and to evaluate the main diagonal of S. The obtained results can be used to find the minimum eigenvalue of S.

As an example, consider the following 3×3 matrix for S:

$$S = \begin{pmatrix} -6 & 1 & -1 \\ 1 & -2 & 5 \\ -1 & 5 & 10 \end{pmatrix} \tag{3}$$

The matrix is symmetric, so all eigenvalues are real. For each row in S, there is one Gershgorin circle defined by its center and its radius:

- D_1 with the center point $c_1 = s_{11} = -6$ and $r_1 = |1| + |-1| = 2$
- D_2 with the center point $c_2 = s_{22} = -2$ and $r_2 = |1| + |5| = 6$
- D_3 with the center point $c_3 = s_{33} = 10$ and $r_3 = |-1| + |5| = 6$

This implicates, all eigenvalues of S must lie in one of the ranges

$$[s_{11} - r_1, s_{11} + r_1] = [-8, -4], \quad [s_{22} - r_2, s_{22} + r_2] = [-8, 4], \quad [s_{33} - r_3, s_{33} + r_3] = [4, 16].$$

Performing the numerical computation shows that the eigenvalues are approximately $\{-6.6, -3.2, 11.8\}$, all inside the determined ranges. Using the Gershgorin circle approach, we see that the minimum eigenvalue cannot be smaller than the minimum border value, in this example −8, while the right value is ≈ -6.6. **Figure 4** shows that all eigenvalues (green dots) of our matrix are within at least one of the circles.

Since in a squared matrix, all centers of the circle are already given by their diagonals and the calculation of the radius only covers the summation of the elements in the respective row, this variant of the *ShiftParameterDetermination* in Algorithm 1 has a complexity of $\mathcal{O}(N)$. In the experiments, we apply the advanced shift correction on a low-rank approximation of S.

Structure preservation

In this context, the term *structure preservation* refers to the structure of the eigenspectrum with the requirement that those eigenvalues with a contribution in the original spectrum should keep their contribution in the new (but psd) spectrum. Those parts of the eigenspectrum that have no need for correction to construct a psd matrix should be kept unchanged. As illustrated by a synthetic example above in 3a - 3d, the various correction methods differently modify the eigenspectrum and some of them fundamentally change the structure of the eigenspectrum. Those modifications to the eigenvalues (and implicitly on the contribution to the matrix) are: changing the sign of an eigenvalue, changing its magnitude, removing the impact of an eigenvalue, adding artificial contribution to eigenvalues that had zero contribution in the original matrix, or changing the position of the eigenvalue with respect to the original ranking causing a profound reorganization of the eigenspectrum. Especially the last one is highly relevant in learning models that make use of only a few eigenvalues/eigenvectors such as kernel PCA or similar methods that reduce the dimensionality or make use of only the most meaningful eigenvalues and eigenvectors.

In order to illustrate the effects of the various correction methods, **Figure 5** shows the impact of the most relevant correction methods on the properties of the eigenspectrum of a real-world dataset, here the protein dataset is used (see **Section 2.5** for more details about this dataset).

Here, the x-axis represents the index of the eigenvalue, while the y-axis illustrates the contribution value (or impact) of the eigenvalue. The left column of **Figure 5** (Subfigures 5a, 5c, 5e, 5g, 5i) shows the eigenspectra without a low-rank representation, the right column (Subfigures 5b, 5d, 5f, 5h, 5j) comprises the low-rank version of the eigenspectrum: **Figure 5A** illustrates the eigenspectrum of the original dataset without any modification. The red rectangle (solid line) highlights the negative parts of the eigenvalues for which their contribution must be preserved in the data. The orange rectangle (dashed line) represents those eigenvalues that are close to zero or zero. The values of particularly these eigenvalues should be kept untouched

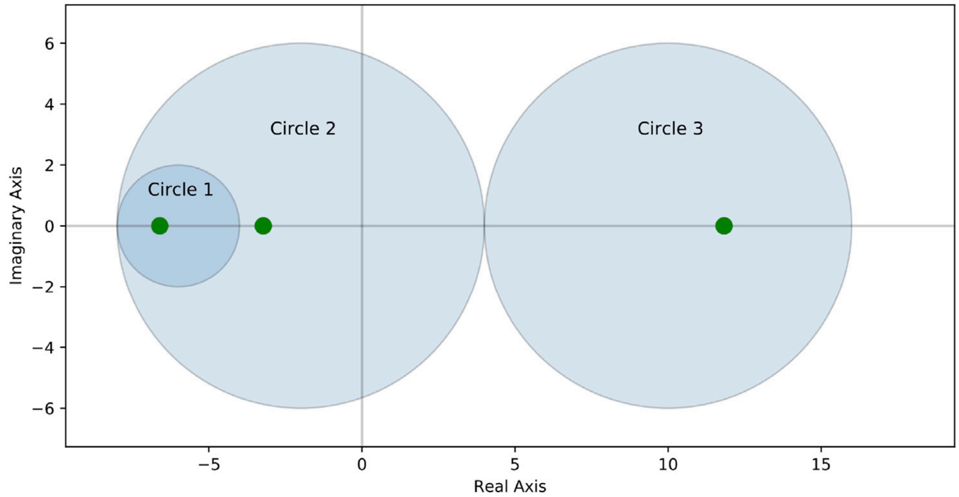


FIGURE 4 | Visualization of Gershgorin's circle theorem on an exemplary matrix.

such that their contribution is still irrelevant after the correction. The green rectangle (dotted line) highlights the positive parts of the eigenvalues which contribution should also be kept unchanged in order not to manipulate the eigenspectrum too aggressively. **Figure 5B** shows the low-rank representation of the original data of 5a. Here, the major negative and major positive eigenvalues (red/solid and green/dotted rectangle) are still present, but many eigenvalues that have been close to zero before, have now been set to exactly 0 (black/dashed rectangle).

Figures 5 C and D show the eigenvalues after applying the clip operator to the eigenvalues shown in **Figures 5 A and B**. In both cases, the major positive eigenvalues (green/dotted rectangle) remain unchanged, as well as the positive values close to 0 and exactly 0. However, the negative eigenvalues close to 0 (parts of the orange/dashed rectangle) and, in particular, the major negative eigenvalues (red/solid rectangle) are all set to exactly 0. By using the clip operator, the contribution to the eigenspectrum of both major negative and slightly negative eigenvalues is completely eliminated.

In contrast to clipping, the flip corrector preserves the contribution of the negative and slightly negative eigenvalues, shown in **Figures 5 E and F**. When using the flip corrector, only the negative sign of the eigenvalue is changed; thus, only the diagonal of the matrix is changed and not the rest. Since the square operator behaves almost analogously to the flip operator and only squares the negative eigenvalues in addition to flipping them, it was not listed separately here. Squaring the values of a matrix drastically increases the impact of the major eigenvalues compared to the minor eigenvalues. If an essential part of the data's information is located in the small eigenvalues, this part gets a proportionally reduced contribution against the significantly increased major eigenvalues.

The modified eigenspectra after applications of the classical shift operator are presented in **Figures 5 G and H**: by increasing all eigenvalues of the spectrum, the part with the larger negative eigenvalues (red/solid rectangle) that had a higher impact now only remains with zero or close to zero contribution. Furthermore, a higher contribution was assigned to those eigenvalues that previously had no or nearly no effect on the eigenspectrum (orange/dashed rectangle). As a result, the classical shift increases the number of non-zero eigencontributions by introducing artificial noise into the data. The same is also evident for the advanced shift without low-rank approximation depicted in **Figure 5I**. Since there are many eigenvalues close to zero but not exactly zero in this data set, all these eigenvalues are also increased in the advanced shift, but can be cured in the low-rank approach.

Unlike the advanced shift approach without low-rank approximation, depicted in **Figure 5I**, a low-rank representation of the data leads to a shifting of only those eigenvalues that had relevant contributions before (red/solid rectangle). Eigenvalues with previously slightly zero contribution (orange/dashed rectangle), derive a contribution of exactly zero by the approximation and are therefore not shifted in the advanced shift method.

Considering the description of *structure preservation* outlined in 2.4, we observe that only the flip and the advanced shift correction (only with low-rank approximation) widely preserve the structure of the given eigenspectrum. For all other methods, the eigenspectrum is substantially modified in particular contributions are removed, amplified, or artificially introduced. In particular, this also holds for the clip or the classical shift corrector, which, however, are frequently recommended in the literature. Although this section contained results exclusively for the protein dataset, we observed similar findings for other indefinite datasets as well. Our findings show that a more sophisticated treatment of the similarity matrix is needed to obtain a suitable psd matrix. This makes our method more appropriate compared to simpler approaches such as the classic shift or clip.

Materials & Experimental Setup

This section contains a series of experiments to highlight the effectiveness of our approach in combination with a low-rank

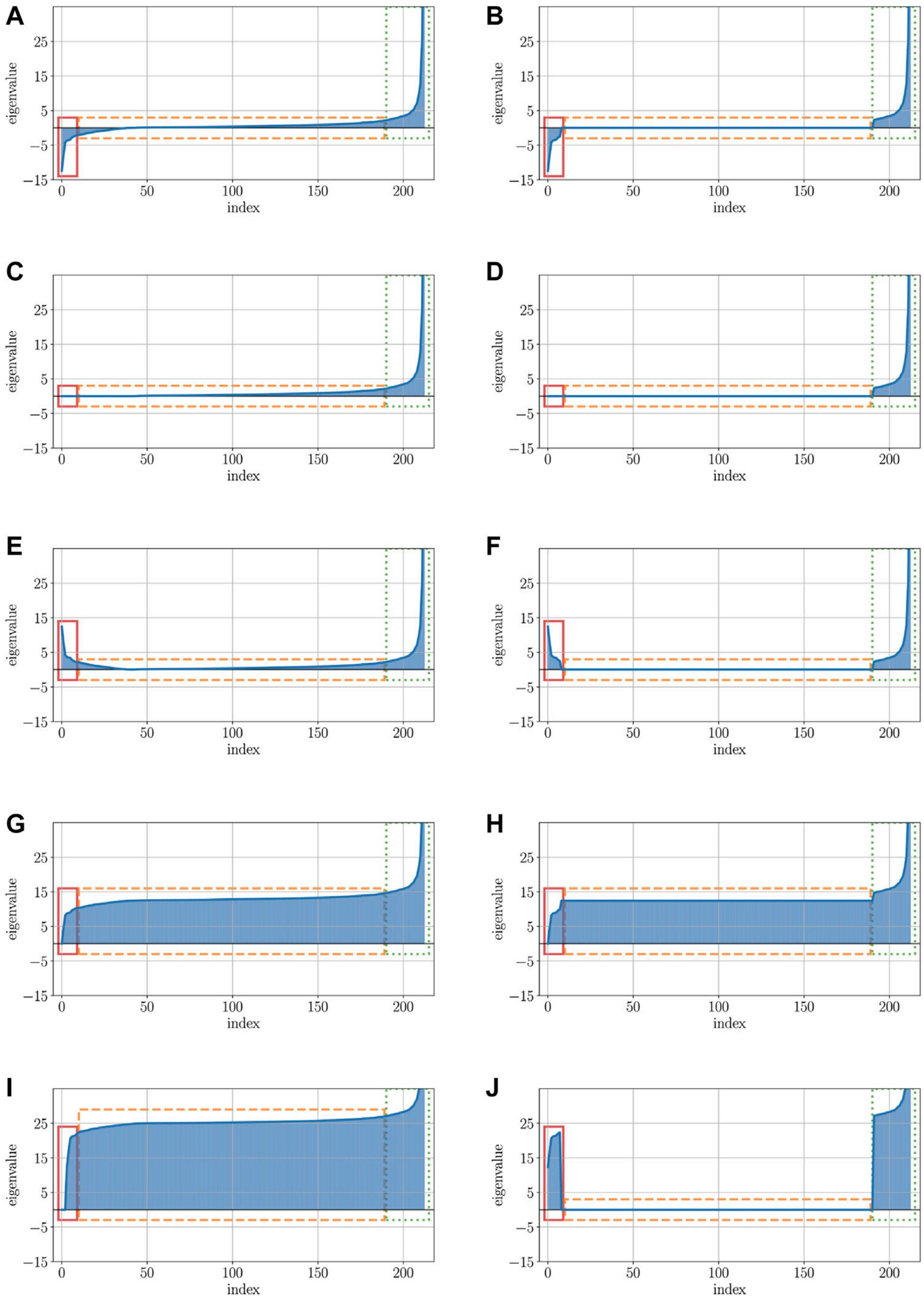


FIGURE 5 | Visualizations of the protein data's eigenspectra after applying various correction methods. **(A)** Visualization of the original eigenspectrum with pos. and neg. eigenvalues of the protein dataset. **(B)** Low-rank representation of the original eigenspectrum from **Figure 5A**. **(C)** Visualization of the original eigenspectrum of **Figure 5A** after clipping all neg. eigenvalues. **(D)** Visualization of the low-rank approximated eigenspectrum after clipping all neg. eigenvalues. **(E)** Visualization of the original eigenspectrum of **Figure 5A** after flipping all neg. eigenvalues. **(F)** Visualization of the low-rank approximated eigenspectrum after flipping all neg. eigenvalues. **(G)** Visualization of the original eigenspectrum of **Figure 5A** after shifting all neg. eigenvalues. **(H)** Visualization of the low-rank approximated eigenspectrum after shifting all neg. eigenvalues. **(I)** Visualization of the original eigenspectrum of **Figure 5A** after advanced shift. **(J)** Visualization of the low-rank approximated eigenspectrum of **Figure 5B** after advanced shift.

approximation. We evaluate the algorithm for a set of benchmark data that are typically used in the context of proximity-based learning. The data are briefly described in the following and summarized in **Table 2**, with details given in the references. After a brief overview of the datasets used for the evaluation, the experimental setup, and the performance of the different eigenvalue correction methods on the benchmark datasets are presented and discussed in this section.

TABLE 2 | Overview of the different datasets. Details are given in the textual description.

Dataset	#samples	#classes	signature
Chromosomes	4,200	21	(2258, 1899, 43)
Flowcyto-1	612	3	(538, 73, 1)
Flowcyto-2	612	3	(26, 73, 582)
Flowcyto-3	612	3	(541, 70, 1)
Flowcyto-4	612	3	(26, 73, 582)
Prodom	2604	53	(1502, 680, 422)
Protein	213	4	(170, 40, 3)
SwissProt	10,988	30	(8487, 2500, 1)
Tox-21: AllBit similarity	14484	2	(2049, 0, 12435)
Tox-21: Assymetric similarity	14484	2	(1888, 3407, 9189)
Tox-21: Kulczynski similarity	14484	2	(2048, 2048, 10388)
Tox-21: McConnaughey similarity	14484	2	(2048, 2048, 10388)
Vibrio	1100	49	(851, 248, 1)

Datasets:

In the experiments, all datasets exhibit indefinite spectral properties and are commonly characterized by pairwise distances or (dis-)similarities. As mentioned above, if the data are given as dissimilarities, a corresponding similarity matrix can be obtained by double centering [17]: $S = -JDJ/2$ with $J = (I - 11^{\top}/N)$, with identity matrix I and vector of ones $\mathbf{1}$. These datasets constitute typical examples of non-Euclidean data. In particular, the focus is on proximity-based data from the life science domain. We consider a broad spectrum of domain-specific data: from sequence analysis, mass spectrometry, chemical structure analysis to flow cytometry. In particular, the later one of flow cytometry [65] could also be important in the analysis of viral data like SARS-CoV-2 [66]. In all cases, dedicated preprocessing steps and (dis-)similarity measures for structures were used by the domain experts to create this data with respect to an appropriate proximity measure. The (dis-)similarity measures are inherently non-Euclidean and cannot be embedded isometrically in a Euclidean vector space. The datasets used for the experiments are described in the following and summarized in **Table 2**, with details given in the references.

1. **Chromosomes:** The Copenhagen chromosomes data set constitutes a benchmark from cytogenetics [67] with a signature (2258, 1899, 43). Karyotyping is a crucial process to classify chromosomes into standard classes and the results are routinely used by the clinicians to diagnose cancers and genetic diseases. A set of 4,200 human chromosomes from 21 classes (the autosomal chromosomes) are represented by grey-valued images. These are transferred to strings measuring the thickness of their silhouettes. These strings are compared using edit distance with insertion/deletion costs 4.5 [40].

2. **Flowcyto** This dissimilarity dataset is based on 612 FL3-A DNA flow cytometer histograms from breast cancer tissues in 256 resolution. The initial data were acquired by M. Nap and N. van Rodijnen of the Atrium Medical Center in Heerlen, The Netherlands, during 2000-2004, using tubes 3, 4,5, and 6 of a DACO Galaxy flowcytometer. Overall, this data set consists of four datasets, each representing the same data, but with different proximity measure settings. Histograms are labeled in 3 classes: aneuploid (335 patients), diploid (131), and tetraploid (146). Dissimilarities between normalized histograms are computed using the L1 norm, correcting for possible different calibration factors [68].

3. **Prodom**: the ProDom dataset with signature (1502,680,422) consists of 2604 protein sequences with 53 labels. It contains a comprehensive set of protein families and appeared first in the work of [69]. The pairwise structural alignments were computed by 69. Each sequence belongs to a group labeled by experts; here, we use the data as provided in 68.

4. **Protein**: the Protein data set has sequence-alignment similarities for 213 proteins and is used for comparing and classifying protein sequences according to its four classes of globins: heterogeneous globin (G), hemoglobin-A (HA), hemoglobin-B (HB) and myoglobin (M). The signature is (170,40,3), where class one through four contains 72, 72, 39, and 30 points, respectively [70].

5. **SwissProt**: the SwissProt data set (SWISS), with a signature (8487,2500,1), consists of 10,988 points of protein sequences in 30 classes taken as a subset from the popular SwissProt database of protein sequences [71]. The considered subset of the SwissProt database refers to the release 37. A typical protein sequence consists of a string of amino acids, and the length of the full sequences varies between 30 to more than 1000 amino acids depending on the sequence. The ten most common classes such as Globin, Cytochrome b, Protein kinase st, etc. provided by the Prosite labeling [72] were taken, leading to 5,791 sequences. Due to this choice, an associated classification problem maps the sequences to their corresponding Prosite labels. These sequences are compared using Smith-Waterman, which computes a local alignment of sequences [5]. This database is the standard source for identifying and analyzing protein sequences such that an automated classification and processing technique would be very desirable.

6. **Tox-21**: The initial intention of the Tox-21 challenges is to predict whether certain chemical compounds have the potential to disrupt processes in the human body that may lead to adverse health effects, i. e. are toxic to humans [73]. This version of the dataset contains 14484 molecules encoded as Simplified Molecular Input Line Entry Specification (SMILE) codes. SMILE codes are ASCII-strings to encode complex chemical structures. For example, Lauryldiethanolamine has the molecular formula of C16H35NO2 and is encoded as CCCCCCCCCCCCN(CCO)CCO. Each smile code is described as a morgan fingerprint [74, 75] and encoded as a bit-vector with a length of 2048 via the RDKit[4] framework. The molecules are compared to each other using the non-psd binary similarity metrics AllBit, Kulczynski, McConnaughey, and Asymmetric provided by the RDKIT. The similarity matrix is constructed based on these pairwise similarities. According to the applied similarity metrics, the resulting matrices are varying in their signatures: AllBit (2049, 0, 12435), Asymmetric (1888, 3407, 9189), Kulczynski (2048, 2048, 10388), McConnaughey (2048, 2048,10388). The task of the dataset is binary classification, which is either toxic or non-toxic for every given molecule and should be predicted by a machine learning algorithm. Note that also graph-based representations for smile data are possible [76].

[4] https://www.rdkit.org/

7. **Vibrio:** Bacteria of the genus Vibrio are Gram-negative, primarily facultative anaerobes, forming motile rods. Contact with contaminated water and consumption of raw seafood are the primary infection factors for Vibrio-associated diseases. Vibrio parahaemolyticus, for instance, is one of the leading causes of foodborne gastroenteritis worldwide. The Vibrio data set consists of 1,100 samples of Vibrio bacteria populations characterized by mass spectra. The spectra encounter approximately 42,000 mass positions. The full data set consists of 49 classes of vibrio-sub-species. The mass spectra are preprocessed with a standard workflow using the BioTyper software [12]. As usual, mass spectra display strong functional characteristics due to the dependency of subsequent masses, such that problem-adapted similarities such as described in 12, 77 are beneficial. In our case, similarities are calculated using a specific similarity measure as provided by the BioTyper software [12] with a signature (851,248,1).

RESULTS

In this section, we evaluate our strategy of data-driven proximity-based analysis and highlight the performance of the proposed advanced shift correction on the previously mentioned datasets against other eigenvalue correction methods using a standard SVM classifier. For this purpose, the correction approaches ensure that the input similarity, herein used as a kernel matrix, is psd. This is particularly important for kernel methods to keep expected convergence properties. During the experiments, we measured the algorithm's mean accuracy and its standard deviation in a ten-fold cross-validation. Additionally, we captured the complexity of the model based on the number of necessary support vectors for the SVM. Therefore, we track the percentage of training data points, the SVM model needs as support vectors to indicate the model's complexity.

In each experiment, the parameter C has been selected for each correction method by a grid search on independent data not used during the tests. For better comparability of the considered methods, the results presented here refer exclusively to the use of the low-rank approximated matrices in the SVM. Only when employing the original data for the SVM, no low-rank approximation was implemented to ensure that small negative eigenvalues were not inadvertently removed if they were of low-rank. Please note, that a low-rank approximation only, does not lead to a psd matrix. Accordingly, convergence problems and uncontrolled information loss, by means of discrimination power, may still occur. Furthermore, both proposed methods for the determination of the shift parameter proposed in **section 2.4** were tested on the low-rank approximated datasets against the other eigenvalue correction methods. The results for the classification performance for the advanced shift methods against the other correction methods are shown in **Table 3**. In column *Adv. Shift*, we show the classification performance for the advanced shift with the exact determination of the smallest eigenvalue, whereas column *Adv.-GS* contains the classification performance of the advanced shift, which applied the Gershgorin theorem to approximate the smallest eigenvalue. For the Prodom data, it is known from 27 that the SVM has convergence problems (not converged - subsequently n.c.) on the indefinite input matrix.

In general, the accuracies of the various correction methods are quite similar and rarely differ significantly. As expected, a correction step is needed and the plain use of uncorrected data is suboptimal, often with a clear drop in the performance or may fail. Also, the use of the classical shift operator can not be recommended due to suboptimal results in various cases. In summary, the presented Advanced Shift with the exact determination of the shift parameter performed best, followed by the flip corrector. The results in **Table 3** also show that the accuracy of the Gershgorin shift variant is not substantially lower compared to the other methods.

In most cases, the Gershgorin advanced shift performs as well as the clip and the square correction method. Compared to the classic shift, our Gershgorin advanced shift consistently results in much better accuracies. The reason for this is the appropriate preservation of the structure of the eigenspectrum, as shown in **Section 2.4**. It becomes evident that not only the dominating eigenvalues have to be kept, but the preservation of the entire structure of the eigenspectrum is important to obtain reliable results in general. As the application of the low-rank approximation to similarity matrices leads to a large number of truly zero eigenvalues, both variants of the advanced shift corrections become more effective. Both proposed approaches benefit from eigenspectra with many close to zero eigenvalues, which occurs in many practical data, especially in complex domains like life sciences. Surprisingly, the classical shift operator is still occasionally preferred in the literature [51, 58, 78], despite its reoccurring limitations. The herein proposed advanced shift outperforms the classical shift in almost every experimental setup. In fact, many datasets have an intrinsic low-rank nature, which we employ in our approach but which is not considered in the classical eigenvalue shift. In any case, the classical shift increases the intrinsic dimensionality, also if many eigenvalues have already been of zero contribution in the original matrix. This leads to substantial performance loss in the classification models, as seen in the results. Considering the results of **Table 3**, the advanced shift correction is preferable in most scenarios.

Additionally to the accuracy of the different correction methods, the number of support vectors of each SVM model was gathered. **Table 4** shows the complexity of the generated SVM models in terms of their required support vectors. Thus, the number of support vectors is set in relation to the number of all the available training data points required to build a solid decision boundary. The higher this percentage, the more data points were needed to create the separation plane, leading to a more complex model. As explained in 79 or 80, the run time complexity can become considerably higher with an increasing number of support vectors.

Compared to the original SVM without the low-rank approximation, it becomes evident that our approach generally requires fewer and occasionally significantly fewer support vectors and is therefore considerably less complex. Furthermore, in comparison to the classic shift corrector, the advanced shift is significantly superior in both accuracy and required support vectors. However, compared to clip, flip, and square, things are slightly different: **Table 4** shows, the advanced shift can keep up with the clipping and flipping but has a higher percentage of support vectors compared to the square correction method. Considering the slightly better accuracy and the lower computational cost from **Section 2.2** than clip and flip, the

TABLE 3 | Prediction accuracy (mean ± standard-deviation) for the various data sets and methods in comparison to the advanced shift method. Column *Adv. Shift* shows the performance of the advanced shift method and column *Adv.-GS* provides the performance of the advanced shift using the Gershgorin approach to estimate the minimum eigenvalue.

Dataset	Adv.-GS	Adv. Shift	Original	Shift	Clip	Flip	Square
Chromosomes	96.90 ± 0.61	97.02 ± 0.86	96.83 ± 0.83	71.38 ± 9.34	97.00 ± 0.69	**97.05** ± **1.02**	96.45 ± 0.91
Flowcyto-1	69.62 ± 5.28	69.28 ± 5.10	63.74 ± 6.50	66.02 ± 5.45	69.93 ± 6.31	70.26 ± **5.41**	**70.58** ± 6.09
Flowcyto-2	70.59 ± 4.62	**72.4** ± **5.85**	62.09 ± 5.36	65.69 ± 6.44	71.39 ± 4.96	70.42 ± 3.84	71.08 ± 2.86
Flowcyto-3	71.25 ± 5.75	70.26 ± 3.58	62.09 ± 0.44	64.55 ± 5.61	70.74 ± 5.70	**71.10** ± **4.67**	70.75 ± 3.03
Flowcyto-4	70.10 ± 4.68	70.43 ± 6.12	59.88 ± 0.58	63.54 ± 6.97	**71.10** ± **4.92**	70.25 ± 5.31	68.29 ± 5.68
Prodom	99.77 ± 0.19	**99.85** ± **0.25**	n.c.	99.77 ± 0.26	99.77 ± 0.31	99.77 ± 0.25	99.65 ± 0.47
Protein	98.12 ± 2.31	**99.07** ± **2.12**	60.40 ± 1.13	58.23 ± 9.91	98.10 ± 3.16	99.02 ± 1.86	98.59 ± 2.15
SwissProt	97.55 ± 0.36	**97.50** ± **0.31**	96.46 ± 0.63	96.52 ± 0.37	96.47 ± 0.84	96.53 ± 0.60	97.42 ± 0.39
Tox-21: - AllBit -	97.22 ± 0.31	97.36 ± 0.49	97.37 ± 0.47	97.38 ± 0.44	97.33 ± 0.52	**97.38** ± **0.30**	97.35 ± 0.38
Tox-21: - Asymmetric -	97.33 ± 0.43	**97.46** ± **0.44**	90.40 ± 2.01	95.28 ± 0.64	96.96 ± 0.46	97.33 ± 0.35	97.18 ± 0.48
Tox-21: - Kulczynski -	97.34 ± 0.56	**97.36** ± **0.39**	92.81 ± 2.16	95.28 ± 0.54	97.20 ± 0.26	97.29 ± 0.37	97.30 ± 0.31
Tox-21: - McConnaughey-	97.31 ± 0.44	**97.34** ± **0.41**	92.08 ± 2.02	94.97 ± 0.56	97.15 ± 0.50	97.33 ± 0.32	97.15 ± 0.54
Vibrio	**100.0** ± **0.00**	**100.0** ± **0.00**	**100.0** ± **0.00**	**100.0** ± **0.00**	**100.0** ± **0.00**	**100.0** ± **0.00**	**100.0** ± 0.00

TABLE 4 | Average percentage of data points that are needed by the SVM models for building a well-fitting decision hyperplane.

Dataset	Adv.-GS	Adv. Shift	Original	Shift	Clip	Flip	Square
Chromosomes	45.4%	39.7%	43.9%	99.8%	30.3%	30.6%	**24.0%**
Flowcyto-1	**59.4%**	60.6%	63.8%	99.7%	63.6%	63.6%	62.9%
Flowcyto-2	59.6%	59.1%	69.5%	96.7%	**57.6%**	58.3%	57.7%
Flowcyto-3	58.6%	59.3%	65.1%	99.3%	**57.8%**	58.5%	59.4%
Flowcyto-4	61.2%	59.9%	65.5%	99.5%	59.3%	**59.2%**	62.7%
Prodom	46.6%	18.7%	n.c.	18.7%	18.7%	18.8%	**12.9%**
Protein	38.6%	39.6%	80.3%	99.8%	22.9%	23.6%	**14.7%**
SwissProt	14.1%	13.9%	48.9%	13.9%	13.9%	13.9%	**12.2%**
Tox-21: AllBit	5.5%	5.5%	5.8%	7.4%	6.5%	7.2%	**4.6%**
Tox-21: Assymetric	4.7%	5.4%	7.3%	10.0%	7.6%	7.1%	**4.6%**
Tox-21: Kulczynski	5.3%	5.9%	8.0%	10.0%	7.2%	7.1%	**5.3%**
Tox-21: McConnaughey	5.1%	5.6%	8.4%	8.3%	7.6%	7.5%	**4.2%**
Vibrio	99.9%	99.6%	100.0%	99.5%	99.6%	99.6%	**92.0%**

advanced shift is preferable to clip and flip eigenvalue correction and competitive to the square correction.

In summary, as pointed out also in previous work, there is no simple solution for handling non-psd matrices or the correction of eigenvalues. The results make evident that the proposed variants of the advanced shift correction are especially useful if the negative eigenvalues are meaningful and a low-rank approximation of the similarity matrix preserves the relevant eigenvalues. The analysis also shows that domain-specific measures by means of a data-driven analysis are effectively possible and keep relevant information. The presented strategies allow the use of standard machine learning approaches, like kernel methods without much hassle.

DISCUSSION

In this paper, we addressed the topic of data-driven supervised learning by general proximity measures. In particular, we presented an alternative formulation of the classical eigenvalue shift, *preserving the structure of the eigenspectrum of the data*, such that the inherent data properties are kept. For this advanced shift method, we also presented a novel strategy that approximates the shift parameter based on the Gershgorin circles theorem.

Furthermore, we pointed to the limitations of the classical shift induced by the shift of all eigenvalues, including those with small or zero eigenvalue contributions. Surprisingly, the classical shift eigenvalue correction is nevertheless frequently recommended in the literature, pointing out that only a suitable offset needs to be applied to shift the matrix to psd. However, it is rarely mentioned that this shift affects the entire eigenspectrum and thus increases the contribution of eigenvalues that had no contribution in the original matrix.

As a result of our approach, the eigenvalues that had vanishing contribution before the shift remain irrelevant after the shift. Those eigenvalues with a high contribution keep their relevance, leading to the preservation of the eigenspectrum but with a positive (semi-)definite matrix. In combination with the low-rank approximation, our approach was, in general, better compared to the classical methods. Moreover, also the approximated version of the advanced shift via Gershgorin circles theorem performed as well as the classical methods.

We analyzed the effectiveness of data-driven learning on a broad spectrum of classification problems from the life science domain. The use of domain-specific proximity measures originally caused a number of challenges for practitioners, but with the recent work on indefinite learning, substantial improvements are available. In fact, our experiments with eigenvalue correction methods, especially the advanced shift approach, which keeps the eigenspectrum intact, have shown promising results on many real-life problems. In this way, domain-specific non-standard proximity measures allow the effective analysis of life science data in a data-driven way.

Future work on this subject will include the reduction of the computational costs using advanced matrix approximation and decomposition techniques in the different sub-steps. Another field of interest is a possible adoption of the advanced shift to unsupervised scenarios.

Finally, it remains to be said that the analysis of life science data offers tremendous potential for understanding complex processes in domains such as (bio)chemistry, biology, environmental research, or medicine. Many challenges have already been tackled and solved, but there are still many open issues in these areas where the analysis of complex data can be a key component in understanding these processes.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: https://bitbucket.fiw.fhws.de:8443/users/popp/repos/proximitydatasetbenchmark/browse.

AUTHOR CONTRIBUTIONS

MM, CR and FMS contributed conception and design of the study; CR preprocessed and provided the Tox-21 database; MM performed the statistical analysis; MM and FMS wrote the first draft of the manuscript; MM, CR, FMS and MB wrote sections of the manuscript. All authors contributed to manuscript revision, read and approved the submitted version.

FUNDING

FMS, MM are supported by the ESF program WiT-HuB/2014-2020, project IDA4KMU, StMBW-W- IX.4-170792. FMS, CR are supported by the FuE program of the StMWi, project OBerA, grant number IUK-1709- 0011// IUK530/010.

ACKNOWLEDGMENTS

We thank Gaelle Bonnet-Loosli for providing support with indefinite learning and R. Duin, Delft University for various support with DisTools and PRTools. We would like to thank Dr. Markus Kostrzewa and Dr. Thomas Maier for providing the Vibrio data set and expertise regarding the biotyping approach and Dr. Katrin Sparbier for discussions about the SwissProt data (all Bruker Corp.). A related conference publication by the same authors was published at ICPRAM 2020 see [15] - copyright related material is not affected.

REFERENCES

1. Biehl M, Hammer B, Schneider P, Villmann T. Metric learning for prototype-based classification. In: M Bianchini, M Maggini, F Scarselli, LC Jain, editors. *Innovations in Neural Information Paradigms and Applications. Studies in Computational Intelligence*, **Vol. 247**: Springer (2009) p. 183–99
2. Hastie T, Tibshirani R, Friedman J. *The Elements of Statistical Learning*. New York: Springer (2001)
3. Nebel D, Kaden M, Villmann A, Villmann T. Types of (dis-)similarities and adaptive mixtures thereof for improved classification learning. *Neurocomputing* (2017) **268**:42–54. doi:10.1016/j.neucom.2016.12.091
4. Schölkopf B, Smola A. *Learning with Kernels*. MIT Press (2002)
5. Gusfield D. *Algorithms on Strings, trees, and sequences: Computer science and computational biology*. Cambridge University Press (1997)
6. Sakoe H, Chiba S. Dynamic programming algorithm optimization for spoken word recognition, *IEEE Trans Acoust Speech Signal Process* (1978) **26**:43–49. doi:10.1109/tassp.1978.1163055
7. Ling H, Jacobs DW. Using the inner-distance for classification of articulated shapes. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR 2005), 20-26 June 2005*. San Diego, CA, USA: IEEE Computer Society (2005) p 719–26.
8. Cilibrasi R, Vitányi PMB. Clustering by compression. *IEEE Trans Inform Theory* (2005) **51**:1523–45. doi:10.1109/tit.2005.844059
9. Cichocki A, Amari S-I. Families of alpha- beta- and gamma- divergences: Flexible and robust measures of similarities. *Entropy* (2010) **12**:1532–68. doi:10.3390/e12061532
10. Lee J, Verleysen M. Generalizations of the lp norm for time series and its application to self-organizing maps. In: M. Cottrell, editor. *5th Workshop on Self-Organizing Maps*. **Vol. 1** (2005) p 733–40.
11. Dubuisson MP, Jain A. A modified hausdorff distance for object matching. In Pattern recognition, 1994. Vol. 1–conference A: Computer vision amp; Image processing., proceedings of the 12th IAPR international conference. **Vol.** (1994) p. 566–568.
12. Maier T, Klebel S, Renner U, Kostrzewa M. Fast and reliable maldi-tof ms–based microorganism identification. *Nature Methods* (2006) **3**:1–2. doi:10.1038/nmeth870.
13. Pekalska E, Duin RPW, Günter S, Bunke H. On not making dissimilarities euclidean. In *SSPR&SPR 2004* (2004) p. 1145–1154.
14. Scheirer WJ, Wilber MJ, Eckmann M, Boult TE. Good recognition is non-metric. *Patt Recog* (2014) **47**:2721–2731. doi:10.1016/j.patcog.2014.02.018
15. Münch M., Raab C., Biehl M, Schleif F. Structure preserving encoding of non-euclidean similarity data. In *Proceedings of the 9th international conference on pattern recognition applications and methods–Volume 1: ICPRAM,. INSTICC (SciTePress)* (2020) p 43–51. doi:10.5220/0008955100430051
16. Gisbrecht A, Schleif FM. Metric and non-metric proximity transformations at linear costs. *Neurocomputing* (2015) **167**:643–57. doi:10.1016/j.neucom.2015.04.017
17. Pekalska E, Duin R. *The dissimilarity representation for pattern recognition*. World Scientific (2005)
18. Vapnik V. *The nature of statistical learning theory. Statistics for engineering and information science*. Springer (2000)
19. Ying Y, Campbell C, Girolami M. Analysis of svm with indefinite kernels. In: Y Bengio, D Schuurmans, J D Lafferty, CKI Williams, A Culotta, editors *Advances in neural information processing systems 22*: Curran Associates, Inc. (2009) p 2205–13.
20. Platt JC. Fast training of support vector machines using sequential minimal optimization. In: *Advances in kernel methods: Support vector learning*. Cambridge, MA: MIT Press (1999) p 185–208.
21. Lin H, Lin C. A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods. *Neural Comput* (2003) 1–32. doi:10.1.1.14.6709
22. Luss R, d'Aspremont A. Support vector machine classification with indefinite kernels. *Math Prog Comp* (2009) **1**:97–118. doi:10.1007/s12532-009-0005-5
23. Chen Y, Garcia E, Gupta M, Rahimi A, Cazzanti L. Similarity-based classification: concepts and algorithms. *J Mac Learn Res* (2009) **10**:747–76.
24. Indyk P, Vakilian A, Yuan Y. *Learning-based low-rank approximations*. In: HM Wallach, H Larochelle, A Beygelzimer, F d'Alché-Buc, EB Fox, R Garnett editors. Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada (2019) p. 7400–10.
25. Williams CKI, Seeger MW. Using the nyström method to speed up kernel machines. In: TK Leen, TG Dietterich, V Tresp editors *Advances in neural information processing systems 13, Papers from neural information processing systems (NIPS) 2000* Denver, CO: MIT Press (2000) p 682–688.
26. Xu W, Wilson R, Hancock E. Determining the cause of negative dissimilarity eigenvalues. *LNCS 6854*: LNCS (2011) p 589–597.

27. Schleif FM, Tiño P. Indefinite proximity learning: A review. *Neural Computation* (2015) **27**:2039–96. doi:10.1162/neco_a_00770
28. Shawe-Taylor J, Cristianini N. *Kernel methods for pattern analysis and discovery*. Cambridge University Press (2004)
29. Smith TF, Waterman MS. Identification of common molecular subsequences. *J Mol Biol* (1981) **147**:195–197. doi:10.1016/0022-2836(81)90087-5
30. Haasdonk B, Keysers D. Tangent distance kernels for support vector machines. *ICPR* (2002) (2). 864–868. doi:10.1109/icpr.2002.1048439
31. Goldfarb L. A unified approach to pattern recognition. *Patt Recog* (1984) **17**: 575–582. doi:10.1016/0031-3203(84)90056-6
32. Deza M, Deza E. *Encyclopedia of Distances*. Springer (2009)
33. Ong CS, Mary X, Canu S, Smola AJ. Learning with non-positive kernels. In: CE Brodley, editor. Machine learning, proceedings of the twenty-first international conference (ICML 2004), Banff, Alberta, Canada, July 4-8, 2004. ACM international conference proceeding series: ACM, **Vol. 69** (2004) p 81. doi:10.1145/1015330.1015443
34. Hodgetts CJ, Hahn U. Similarity-based asymmetries in perceptual matching. *Acta Psychologica* (2012) **139**:291–299. doi:10.1016/j.actpsy.2011.12.003
35. Hodgetts CJ, Hahn U, Chater N. Transformation and alignment in similarity. *Cognition* (2009) **113**:62–79. doi:10.1016/j.cognition.2009.07.010
36. Kinsman T, Fairchild M, Pelz J. *Color is not a metric space implications for pattern recognition, machine learning, and computer vision*. In Proceedings of Western New York image processing workshop, WNYIPW 2012 (2012) p. 37–40.
37. Van Der Maaten L, Hinton G. Visualizing non-metric similarities in multiple maps. *Mac Learn* (2012) **87**:33–55. doi:10.1007/s10994-011-5273-4
38. Duin RPW, Pekalska E. *Non-euclidean dissimilarities: causes and informativeness*. In *SSPR&SPR 2010* (2010) p. 324–33.
39. Kohonen T, Somervuo P. How to make large self-organizing maps for nonvectorial data. *Neural Networks* (2002) **15**:945–52. doi:10.1016/s0893-6080(02)00069-2
40. Neuhaus M, Bunke H. Edit distance-based kernel functions for structural pattern classification. *Patt Recog* (2006) **39**:1852–63. doi:10.1016/j.patcog.2006.04.012
41. Gärtner T, Lloyd JW, Flach PA. Kernels and distances for structured data. *Mac Learn* (2004) **57**:205–32. doi:10.1023/B:MACH.0000039777.23772.30
42. Poleksic A. Optimal pairwise alignment of fixed protein structures in subquadratic time. *J Bioinform Comput Biol* (2011) **9**:367–82. doi:10.1142/s0219720011005562
43. Zhang Z, Ooi BC, Parthasarathy S, Tung AKH. Similarity search on Bregman divergence. *Proc VLDB Endow* (2009) **2**:13–24. doi:10.14778/1687627.1687630
44. Schnitzer D, Flexer A, Widmer G. A fast audio similarity retrieval method for millions of music tracks. *Multimed Tools Appl* (2012) **58**:23–40. doi:10.1007/s11042-010-0679-8
45. Mwebaze E, Schneider P, Schleif FM. Divergence based classification in learning vector quantization. *Neurocomputing* (2010) **74**:1429–35. doi:10.1016/j.neucom.2010.10.016
46. Nguyen NQ, Abbey CK, Insana MF. Objective assessment of sonographic: Quality ii acquisition information spectrum. *IEEE Trans Med Imag* (2013) **32**: 691–98. doi:10.1109/tmi.2012.2231963
47. Tian J, Cui S, Reinartz P. Building change detection based on satellite stereo imagery and digital surface models. *IEEE Trans Geosci Remote Sens* (2013) **52**(1):406–417. doi:10.1109/tgrs.2013.2240692
48. van der Meer F. The effectiveness of spectral similarity measures for the analysis of hyperspectral imagery. *Int J Appl Earth Obser Geoinf* (2006) **8**:3–17. doi:10.1016/j.jag.2005.06.001
49. Bunte K, Haase S, Biehl M, Villmann T. Stochastic neighbor embedding (SNE) for dimension reduction and visualization using arbitrary divergences. *Neurocomputing* (2012) **90**:23–45. doi:10.1016/j.neucom.2012.02.034
50. Mohammadi M, Petkov N, Bunte K, Peletier RF, Schleif FM. Globular cluster detection in the GAIA survey. *Neurocomputing* (2019) **342**:164–171. doi:10.1016/j.neucom.2018.10.081
51. Loosli G. *Trik-svm: an alternative decomposition for kernel methods in krein spaces*. In: M Verleysen editor. Proceedings of the 27th european symposium on artificial neural networks (ESANN) 2019. Evere, Belgium: D-side Publications (2019) p. 79–94
52. Mehrkanoon S, Huang X, Suykens JAK. Indefinite kernel spectral learning. *Patt Recog* (2018) **78**:144–153. doi:10.1016/j.patcog.2018.01.014
53. Schleif F, Tiño P, Liang Y. *Learning in indefinite proximity spaces - recent trends*. In: 24th european symposium on artificial neural networks, ESANN 2016; 2016 April 27-29; Bruges, Belgium. (2016) p. 113–122.
54. Loosli G, Canu S, Ong CS. Learning SVM in Kreĭn spaces. *IEEE Trans Patt Anal Mach Intell* (2016) **38**:1204–16. doi:10.1109/tpami.2015.2477830
55. Schleif FM, Tiño P. Indefinite core vector machine. *Patt Recog* (2017) **71**: 187–195. doi:10.1016/j.patcog.2017.06.003.
56. Higham NJ. Computing a nearest symmetric positive semidefinite matrix. *Linear Algebra and Its Applications* (1988) **103**:103–118. doi:10.1016/0024-3795(88)90223-6
57. Strassen V. Gaussian elimination is not optimal. *Numer Math* (1969) **13**: 354–356. doi:10.1007/bf02165411
58. Filippone M. Dealing with non-metric dissimilarities in fuzzy central clustering algorithms. *International Journal of Approximate Reasoning* (2009) **50**:363–384. doi:10.1016/j.ijar.2008.08.006
59. Mises RV, Pollaczek-Geiringer H. Praktische Verfahren der Gleichungsauflösung. *Z Angew Math Mech* (1929) **9**:152–164. doi:10.1002/zamm.19290090206
60. Gerschgorin S. Ueber die abgrenzung der eigenwerte einer matrix. *Izvestija Akademii Nauk SSSR, Serija Matematika* (1931) **7**:749–54.
61. Varga RS. Geršgorin and his circles. In: *Springer series in computational mathematics*. Springer Berlin Heidelberg (2004)
62. Verleysen M, François D. The curse of dimensionality in data mining and time series prediction. In: J Cabestany, A Prieto, FS Hernández, editors. Computational intelligence and bioinspired systems, 8th international work-conference on artificial neural networks, IWANN 2005. Lecture Notes in Computer Science; 2005 June 8-10; Vilanova i la Geltrú, Barcelona, Spain, Proceedings: **Vol. 3512**: Springer (2005) p. 758–770.
63. Sanyal A, Kanade V, Torr PHS. Low rank structure of learned representations. *CoRR* (2018) doi:CoRR abs/1804.07090
64. Ilic M, Turner IW, Saad Y. Linear system solution by null-space approximation and projection (SNAP). *Numer Linear Algebra Appl* (2007) **14**:61–82.
65. Aghaeepour N, Finak G, Hoos H, Mosmann TR, Brinkman R, Gottardo R, et al. Critical assessment of automated flow cytometry data analysis techniques. *Nat Methods* (2013) **10**:228–238. doi:10.1038/nmeth.2365
66. Ou X, Liu Y, Lei X, Li P, Mi D, Ren L. Characterization of spike glycoprotein of sars-cov-2 on virus entry and its immune cross-reactivity with sars-cov. *Nat Commun* (2020) **11**:1620. doi:10.1038/s41467-020-15562-9
67. Lundsteen C, Phillip J, Granum E. Quantitative analysis of 6985 digitized trypsin g-banded human metaphase chromosomes. *Clin Genet* (1980) **18**: 355–370 doi:10.1111/j.1399-0004.1980.tb02296.x
68. Duin RP [Dataset]: PRTools (2012)
69. Roth V, Laub J, Buhmann JM, Müller KR. Going metric: denoising pairwise data. In: *NIPS* (2002) p. 817–824.
70. Hofmann T, Buhmann JM. Pairwise data clustering by deterministic annealing. *IEEE Trans Patt Anal Machine Intell* (1997) **19**:1–14. doi:10.1109/34.566806
71. Boeckmann B, Bairoch A, Apweiler R, Blatter MC, Estreicher A, Gasteiger E. The swiss-prot protein knowledgebase and its supplement trembl in 2003. *Nucleic Acids Res* (2003) **31**:365–370. doi:10.1093/nar/gkg095
72. Gasteiger E, Gattiker A, Hoogland C, Ivanyi I, Appel R, Bairoch A. Expasy: the proteomics server for in-depth protein knowledge and analysis. *Nucleic Acids Res* (2003) **31**. doi:10.1093/nar/gkg563
73. Huang R, Xia M, Nguyen DT, Zhao T, Sakamuru S, Zhao J. Tox21challenge to build predictive models of nuclear receptor and stress response pathways as mediated by exposure to environmental chemicals and drugs. *Front Environ Sci* (2016) **3**:85. doi:10.3389/fenvs.2015.00085
74. Figueras J. Morgan revisited. *J Chem Inf Model* (1993) **33**, 717–718. doi:10.1021/ci00015a009
75. Ralaivola L, Swamidass SJ, Saigo H, Baldi P. Graph kernels for chemical informatics. *Neural Networks* (2005) **18**:1093–1110. doi:10.1016/j.neunet.2005.07.009
76. Bacciu D, Lisboa P, Martín JD, Stoean R, Vellido A. *Bioinformatics and medicine in the era of deep learning*. In: 26th european symposium on artificial

neural networks, ESANN 2018; 2018 April 25-27; Bruges, Belgium (2018) p. 345–354.

77. Barbuddhe SB, Maier T, Schwarz G, Kostrzewa M, Hof H, Domann E. Rapid identification and typing of listeria species by matrix-assisted laser desorption ionization-time of flight mass spectrometry. *Appl Environ Microbiol* (2008) **74**: 5402–5407. doi:10.1128/aem.02689-07
78. Chakraborty J. *Non-metric pairwise proximity data*. [PhD thesis]: Berlin Institute of Technology (2004)
79. Burges CJC. Simplied support vector decision rules. *Icml* (1996)
80. Osuna E, Girosi F. Reducing the run-time complexity of support vector machines. International conference on pattern recognition (1998)

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Systems Neuroscience

CLINICAL TRIAL
published: 26 November 2020
doi: 10.3389/fnsys.2020.527757

BrainWave Nets: Are Sparse Dynamic Models Susceptible to Brain Manipulation Experimentation?

Diego C. Nascimento[1,2]*, *Marco A. Pinto-Orellana*[3], *Joao P. Leite*[4], *Dylan J. Edwards*[5,6], *Francisco Louzada*[1†] *and Taiza E. G. Santos*[4†]

[1] Institute of Mathematical Science and Computing, University of São Paulo, Sao Carlos, Brazil, [2] Departamento de Matemática, Universidad de Atacama de Chile, Copiapo, Chile, [3] Institutt for maskinelektronikk og kjemi, Oslo Metropolitan University, Oslo, Norway, [4] Ribeirao Preto Medical School, University of São Paulo, Ribeirao Preto, Brazil, [5] Moss Rehabilitation Research Institute, Elkins Park, PA, United States, [6] School of Medical and Health Sciences, Edith Cowan University, Joondalup, WA, Australia

OPEN ACCESS

Edited by:
Alex Hansen,
Norwegian University of Science and Technology, Norway

Reviewed by:
Aimin Jiang,
Hohai University, China
Xinlong Wang,
University of Texas at Arlington, United States

Correspondence:
Diego C. Nascimento
dnstata@gmail.com

†These authors have contributed equally to this work and share last authorship

Received: *17 January 2020*
Accepted: *19 August 2020*
Published: *26 November 2020*

Citation:
Nascimento DC, Pinto-Orellana MA, Leite JP, Edwards DJ, Louzada F and Santos TEG (2020) BrainWave Nets: Are Sparse Dynamic Models Susceptible to Brain Manipulation Experimentation? Front. Syst. Neurosci. 14:527757. doi: 10.3389/fnsys.2020.527757

Sparse time series models have shown promise in estimating contemporaneous and ongoing brain connectivity. This paper was motivated by a neuroscience experiment using EEG signals as the outcome of our established interventional protocol, a new method in neurorehabilitation toward developing a treatment for visual verticality disorder in post-stroke patients. To analyze the [complex outcome measure (EEG)] that reflects neural-network functioning and processing in more specific ways regarding traditional analyses, we make a comparison among sparse time series models (classic VAR, GLASSO, TSCGM, and TSCGM-modified with non-linear and iterative optimizations) combined with a graphical approach, such as a Dynamic Chain Graph Model (DCGM). These dynamic graphical models were useful in assessing the role of estimating the brain network structure and describing its causal relationship. In addition, the class of DCGM was able to visualize and compare experimental conditions and brain frequency domains [using finite impulse response (FIR) filter]. Moreover, using multilayer networks, the results corroborate with the susceptibility of sparse dynamic models, bypassing the false positives problem in estimation algorithms. We conclude that applying sparse dynamic models to EEG data may be useful for describing intervention-relocated changes in brain connectivity.

Keywords: state space models, multilayer networks, high-dimensional time series model, transcranial direct current stimulation, dynamic graphical model

1. INTRODUCTION

In the area of neuroscience, work related to the brain network structure, as well as its dynamics, has increased due to technological developments (high resolution and storage capacity). Notwithstanding, the field aims to understand "how" and "why" the effects/events occur based on learning probabilistic connection structures to assume some feasible causal inference (Pearl, 2014). There is thus an immediate urge to map its complex organization, and two types of connectivity are commonly studied: functional and dynamic. Functional connectivity is a statistical measure of the correlation within observations in the same time-lapse, and dynamic connectivity is the relationship among the measurements compared with their previous value impact.

Thus, the links among anatomical parcellations of the brain are described by their similarity patterns; for instance, a channel represents the activity of a group of neurons, and it is measured according to its space relation, time, and frequency domains. Statistical significance tests are often conducted to estimate the existence of those links in order to project an estimated topology regarding the interaction among this observed group of neurons. For example, brain dynamics are measured as biosignals through an electroencephalogram (EEG), functional magnetic resonance imaging (fMRI), diffusion tensor imaging (DTI), and Doppler ultrasound. Most recently, effective brain network connectivity changes following non-invasive transcranial stimulation has been investigated using fMRI (Fiori et al., 2018), fNIRS (Cao et al., 2018), and EEG (Baxter et al., 2017).

Biosignals are often presented as time-indexed values in which their modeling requires components that may also vary over time; the dynamic factor models, together with graphical representation, can help this demand. Time-varying Bayesian dynamic models were introduced, and variations were then developed, such as the Gaussian graphical model and usage of splines (for further details, please see Quintana and West, 1987; Queen and Smith, 1993; Carvalho et al., 2007; Anacleto et al., 2017). Nevertheless, this approach is always suitable for multivariate series whose component univariate series are similar and share a common structure.

Network modeling is a mathematical framework, part of graph theory, used to represent and analyze relationships in multivariate data. Recent advances in network estimation have moved the emphasis of the analysis from single-layer networks to multilayer structures facilitating the interpretation of multivariate relationships (Kivelä et al., 2014). This paradigm shift expands the possibilities of extracting information about complex systems, and conducts a multilayer network estimation of biosignals that can incorporate the change in time and/or different frequencies.

Multilayer analysis can reveal the complexity of the human brain, and investigations can thus show effective functional roles in brain region activation and visual representation (De Domenico, 2017; Gratton et al., 2018). In this context, two main approaches are often seen, multimodal connectivity or structural-functional relationships (different layers represent replicated nodes and their interaction) and time-varying networks (evolution of the temporal snapshots).

The concept of sparse multivariate time series with multiplex networks benefits the analysis of brain dynamic activation by using the frequency-domain approaches as physiologically applicable biosignal denoising. Decomposition methods in the frequency domain are generally used in conjunction with graphical models; for example, Bach and Jordan (2004) presented this methodology for stationary Gaussian time series, which complement the results obtained from the time domain. Moreover, sparse models deal directly with the limitations of complex high-frequency time series, such as complex structural and computational constraints.

In this paper, the main contribution was the description of a statistical methodological plot adopting the time domain series in the frequency domain combined with some dynamic spatial models, targeting a more in-depth understanding of an applied neuroscience research question. We demonstrated the validity and feasibility of this sequence of statistical approaches that could reveal a pattern toward brain activation, comparing the brain dynamic before and after a transcranial neuromodulation stimulation. The data were acquired following a systematic randomized controlled clinical trial protocol (Santos et al., 2018), using a sample of the EEG signals collected before applying high-definition transcranial direct current stimulation (HD-tDCS) over the temporal-parietal junction, under the polarity anode center condition and *post* the 2 mA current intensity in a single young healthy subject.

The motivation stems from the need to understand neuro-activation across different brain areas to analyze the effects of a focal transcranial brain stimulation and establish an innovative and effective neurorehabilitation strategy to treat verticality disorder after brain lesions (post-stroke). Moreover, the impact of this study will extend to the entire neuroscience/medical field that needs to adopt dynamic modeling for complex data; sparse models enable the use of big data demanding a low computational cost (shrinking the number of parameters in the model).

2. METHODS

The paper is organized as follows. In subsection 2.1, we present an overview of the adopted experimental protocol. In subsection 2.2, we present the theoretical background for dynamic linear models, sparse estimation, sparsity in modeling, multilayer networks, network inference, and time series from a frequency-domain approach. In section 3, we discuss the empirical clinical results comparing different sparse estimations to distinguish patterns among different brain wavebands. Finally, some final comments are given in section 4.

2.1. Protocol Rational and Data Characterization

Neural systems' imbalance and degeneration related to postural control have led to new research regarding their origin and pathophysiology (Winter, 1995). In humans, different sensory information is used as pathways in the brain to maintain posture in the upright position (Day and Cole, 2002), and postural imbalance is one of the most common disorders after stroke. However, it has not been well-documented in the literature (Chern et al., 2010; Baggio et al., 2016). Hence, increasing knowledge about the effects of this strategy is essential for developing more effective rehabilitation protocols.

Non-invasive techniques of brain stimulation are current therapeutic resources related to the pathophysiology and behavior of the mechanisms that guide the human mind. Transcranial direct current electrical stimulation (tDCS) is a non-invasive neuromodulation technique that can model the cerebral function with a safe profile (Edwards et al., 2013). tDCS consists of electrodes unleashing weak electrical currents over the scalp, inducing cortical changes; it increases or decreases the local network excitability depending on the electrical current polarity.

At the neuronal level, tDCS affects polarization of the resting membrane potential, and this effect may acutely impact cortical excitability (Priori et al., 1998). Another effect may be related to the electrical dynamics of the neuronal membrane potential, as well as its change by at least 1 h (Nitsche et al., 2003). In addition, changes in the effectiveness of synaptic connections may last during the stimulation period. Studies on peripheral nerve and spinal cord stimulation have shown that direct current effects are also non-synaptic, with transient changes in the density of protein channels below the stimulation area (Ardolino et al., 2005; Cogiamanian et al., 2008). High definition tDCS (HD-tDCS) is a contemporary way of transcranial electrical stimulation, which promotes more focal stimulation than the conventional tDCS methods (please see Edwards et al., 2013).

In addition to these tDCS direct effects, "indirect" consequences come from connective-driven alterations of distant cortical and sub-cortical areas (Brunoni et al., 2012). Lang et al. (2005) revealed that stimulating the right frontopolar cortex (M1) with tDCS also activates several connected regions. Changes in brain activity, after the tDCS session, were also measured related to regions concerning blood flow using the sequential H1520 PET scan. In addition, by observing the stimulus area, the activation of "several motor areas" was observed, including "the caudal portion of the anterior cingulate cortex, cerebellum and superior temporal sulcus." This could be due to a modulation of the functional interaction between M1 and these areas via cortico-cortical and cortico-subcortical connections.

Other studies using transcranial magnetic stimulation (TMS), also as a non-invasive neuromodulation technique, described the increased activity of the homologous area, contralateral to the stimuli (Siebner et al., 2000; Lee et al., 2003). Moreover, cerebral hemisphere interaction is commonly observed in the literature (Gilio et al., 2003; Plewnia et al., 2003).

These "indirect" changes on cerebral function are fundamental issues regarding the objective of the present study, which evaluated the effects of tDCS in the temporoparietal junction, the area related to postural control in humans (Winter, 1995). Inter-hemispheric interactions may contribute to defining the temporal and spatial features of voluntary movements, and consequently postural control (Meyer et al., 1998). There is a balance between these inter-hemispheric interactions, where each human cortex exerts inhibitory influences on the opposite motor cortex in normal conditions (Ferbert et al., 1992). Therefore, developing non-invasive techniques that modulate this balance will be a significant advance in the rehabilitation setting of stroke patients and other postural control disorders after more profound knowledge is gained of the technique's effects on the human brain.

The current study was derived from a randomized double-blinded sham-controlled clinical trial that aimed to investigate a polarity and intensity-dependent shift in high-density EEG signals, following an intervention using high-definition transcranial direct current stimulation applied over the temporo-parietal junction in healthy subjects (Santos et al., 2018). The study protocol consisted of an HD-tDCS application over the right temporoparietal junction area, using a Soterix® NY-USA HD-tDCS with a constant current anode (active control). Four electrodes were used; the central electrode was placed over the circumcenter of P4-C4-T8 EEG coordinates, and the three peripheral electrodes were placed at a distance of 3 centimeters from the central electrode (over the EEG coordinates P4, C4, and T8). EEG recordings were made before and after each stimulation period, thus detecting ongoing changes in the raw EEG signals in response to tDCS (**Figure 1**). The total duration was 5 min of resting-state baseline condition added by 1.5 min of stimulation plus 5 min of accommodation post-stimulus, as shown in **Figure 2** (for protocol details, please see Santos et al., 2018).

A dense array EEG signal was acquired using a 256-channel sensor net from Electrical Geodesics Inc. during the aforementioned electrical stimulation conditions. All channels were referenced to the vertex with reduced electrical impedance. The EEG was recorded continuously before and after the stimulation, excluding ramp-up and ramp-down periods (1.5 min total). The full trial experimentation lasted ∼120 min. Previously, we discussed (Nascimento et al., 2019) some variations toward the Cathodal against the Active Control (Anodal) at the 2 mA condition; in this work we aimed to discuss an innovative statistical analyses of only one sample of the protocol experimentation compared to its reference (baseline).

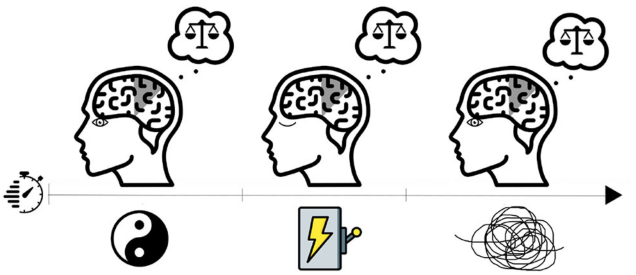


FIGURE 1 | Visual representation of a resting-state baseline condition (left illustration; eyes open) in addition to stimulation stage (eyes closed) and accommodation post-stimulus (right illustration; eyes open). The main interest in the study is to compare the resting-state vs. accommodation post-stimulus.

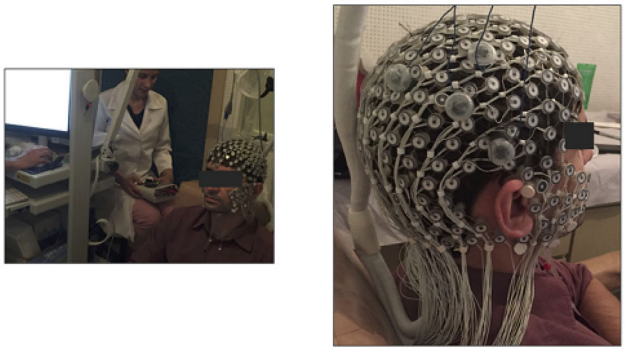


FIGURE 2 | Photograph of the experimental trial (left-hand side) and the EEG cap (right-hand side) with small electrode array covering the scalp, while the large electrodes identifiable as a triangle configuration (four electrodes total) represent the tDCS stimulating electrodes (this image was previously published by Santos et al., 2018, under Open Access and Creative Commons Attribution License).

Thus, in the present study, we analyzed and discussed the data set of a single healthy adult male participant during the resting state (baseline condition) and 45 s after an electrical stimulation. Each period (before and after stimulation) contains 5 min of observation, whereas the EEG sample rate was 500 Hz (500 observations per second), representing a total of 300,000 observations.

2.2. The Model

Dynamic structure modeling may be considered as an alternative to estimate brain connectivity; additionally, it is natural to aggregate its estimated parameters into a graphical representation. Nonetheless, the dynamic model class is overparametrized (West et al., 1985; West and Harrison, 1989), especially in the time-varying approach, demanding some shrinkage of the parameter space (i.e., by adding sparsity to the parameter vector estimation process). A word of caution must be mentioned here; search patterns in small dimensions may deal with great noise (Nakao, 2016), added by limitations toward how to generalize the low-dimensional reduction approach (Rodrigues et al., 2016) and, for instance, brainwaves present a highly active process which comes with much noise (Natarajan et al., 2004). Therefore, filtering preprocessing is suggested to break the observed/raw time series signal into the frequency domain and then using the finite impulse response (FIR) filter. These elements are presented next and visually summarized in **Figure 3**.

2.2.1. Dynamic Linear Model

The state space model is a flexible learning linear/non-linear dynamical system. As a particular case, the state transition and observation functions, known as a Dynamic Factor Model (DFM), may be expressed as a Gaussian linear process, often called a Dynamic linear model (DLM). For instance, consider a p-dimensional State Vector and m-dimensional observations, both normally distributed. At the initial time, $(t = 0)$ presents the mean μ_0 and variance σ_0^2,

$$\theta_0 \sim N_p(\mu_0, \sigma_0^2)$$

then for the time $t \geq 1$,

$$\underbrace{Y_t = F_t\theta_t + \upsilon_t}_{\text{observation equation}}, \quad \upsilon_t \sim N_m(0, V_t),$$

$$\underbrace{\theta_t = G_t\theta_{t-1} + \omega_t}_{\text{state equation}}, \; \omega_t \sim N_p(0, W_t)$$

where matrices G_t (dimension $p \times p$) and F_t $(m \times p)$ are known, followed by independent Gaussian random vectors υ_t and ω_t with mean equal to zero and known variance matrices V_t and W_t.

Considering an $\mathbb{R}^p$-valued and $\mathbb{R}^m$-valued time series, we have the following: (i) (θ_t) is a Markov chain and (ii) the observed time series (Y_t), conditioned to (θ_t). They are independent among the other time series and depends only on the associated state (θ_t).

Moreover, this class of models is flexible given the possibility of incorporating more complex structures (locally they are linear, but globally perform as non-linear dynamic), by allowing the time-varying parameters, that is, compounding a latent variable in the estimation process. The estimation toward the state vector uses the conditional density $\pi(\theta_k \mid Y)$, where $t = 1, ..., T$ and Y are the observed values. Furthermore, k represents the recursive period and t the current period, where estimation problems are filtering $(k = t)$, smoothing $(k < t)$, and state prediction $(k > t)$.

Filtering is a procedure that aims to update the current estimates as new data are observed $\pi(\theta_t \mid Y_{1:t})$. Smoothing is a retrospective analysis, already containing all the observations in the series, which computes the conditional distribution θ represented by $\pi(\theta_t \mid Y_{1:T})$, starting from $\pi(\theta_T \mid Y_{1:T})$ back

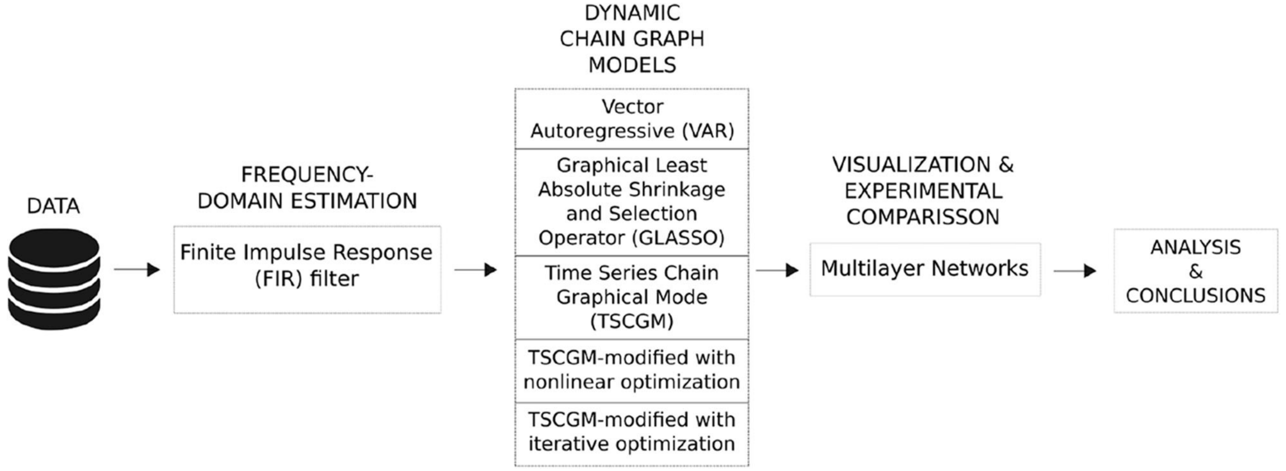

DATA
FREQUENCY-DOMAIN ESTIMATION
Finite Impulse Response (FIR) filter
DYNAMIC CHAIN GRAPH MODELS
Vector Autoregressive (VAR)
Graphical Least Absolute Shrinkage and Selection Operator (GLASSO)
Time Series Chain Graphical Mode (TSCGM)
TSCGM-modified with nonlinear optimization
TSCGM-modified with iterative optimization
VISUALIZATION & EXPERIMENTAL COMPARISSON
Multilayer Networks
ANALYSIS & CONCLUSIONS

FIGURE 3 | Visual summary of the methodological framework. Only one participant/trial was selected, and its biosignals were extracted using EEG during baseline period and 2 mA post-stimulation in order to compare the brain dynamic responses. As a pre-processing phase, an FIR filter was applied in the raw EEG signal aiming to estimate the brain frequency-domain phases. Then, five types of dynamic chain graph models were tested and compared, for instance, illustrated only with the filtered EEG alpha band. Later, all the bandpower were compared, given the outperformed model comparing resting-state vs. port-stimulation electrical brain dynamic.

to front. Prediction is a forecast procedure that estimates the next observation based on the data $\pi(\theta_{t+1} \mid Y_{1:t})$. Further details on Bayesian Forecasting and Dynamic models can be found in West and Harrison (1989) and Petris et al. (2009).

In contrast, the Vector Autoregressive (VAR) model is widely used in the literature (Krystal et al., 1999; Prado et al., 2006; Schlögl and Supp, 2006; Garrison et al., 2015) and recognize the non-linear dependencies between different brain regions, although may present limitations toward the curse of dimensionality. It is possible to impose restrictions on a VAR to make it "similar" to a factor model, i.e., such as DFM.

2.2.2. Sparse Estimation Framework

Recent discoveries, related to time series modeling, discuss the challenge of estimating the model's dependency order, that is, related to the measure of complexity to high-dimensionality resolution. For instance, it enables the eigenvalues and eigenvectors to rotate in the state-space parameter dimension, given restrictions in the parameter vector space imposing some parameters to be equal to zero. Therefore, the main question may concern identifying the "best" and "simplest" approximation (without losing relevant information) that corresponds to the dynamic process.

This definition of "the best" is non-trivial given the lack of knowledge regarding the joint function related to the data and parameter associated with the phenomenon under study. The only available information is from the observed data as an information base in the estimation process. Several inferential methods may be adopted; among them, the most popular are maximum likelihood and ordinary least squares.

The sparse approach is equivalent to creating a bias toward sparsity in the maximum likelihood estimator (MLE), which may reduce the minimum square error. Thus, it sets conditions in the least squares aiming to minimize the l_1-norm producing sparsity in the parameter vector θ. Additionally, prior knowledge can be incorporated, targeting only a subset of the parameter vector; to minimize a specific parameterization (θ_0) problem, then

$$\underset{s}{min}\|\theta_0 + s\|_1$$

truncating a NP hard problem (Chickering, 1996) into a linear programming (LP) problem in standard form (Zeemering, 2015). In general terms, adding vectorial assumptions concerning the reparatrization of the model associated with the parameter vector θ will impact the adjustment of the model and will be represented as an error vector [$(e(\theta))$, which can be calculated according to a criterion, for example, least squares] that depends only on θ.

The search space is limited by models, some of them equivalent, which produce the same error vector value and least squares error (Tibshirani et al., 2012). That is, shrinkage may be applied through a singular value decomposition (SVD) to the matrix, which associates the number of constraints kernel of the Jacobian ($J(\theta)$) or Hessian ($H(\theta)$) matrices.

The non-linear least squares minimization method search direction ($s(\theta)$) to refine the parameters by successive iterations may be adopted, such as a Newton method, described as

$$s(\theta) = -\alpha H(\theta)^{-1} J(\theta)' e(\theta).$$

Based on the SVD results, values that assume a value equal to zero can be determined, thus setting a threshold if needed. A word of caution regarding the threshold; low values may bound the search space (then exclude valid directions to search for sparsity) and high values may change the model's behavior.

In contrast, other solutions may be obtained by the dual or primal linear programming (LP) problem. Deviation toward the search direction accuracy during the optimization procedure, through setting up a threshold, determines the quality of the maximization procedure. An application in the medical field, Zeemering (2015) used regression and state space classes of models in order to add sparse estimation to atrial fibrillation research.

The models adopted were classical VAR, Graphical Least Absolute Shrinkage and Selection Operator (GLASSO), Time Series Chain Graphical Model (TSCGM) and TSCGM-modified using Non-linear optimization over log-likelihood and Iterative optimizing the log-likelihood. The modified TSCGM, adopted in this work, considered an optimization option that uses the proportion of parameters equal to zero in relation to the total number of parameters of the model with a bias toward sparsity, in the MLE, whose minimization will occur through the l_1-norm of the parameter vector and the Smoothly Clipped Absolute Deviation (SCAD).

2.2.3. Sparsity in Modeling

The classical method for estimating connectivity matrices often uses the Vector Autoregressive (VAR) Model, which is a particular case of DLM when the parameters are invariant in time. For instance, consider a vector of observed variables Y, where I is an identity matrix, Matrices X represent Y lagged dependence, Γ_j are autoregressive parameters, and u is the error vector with covariance matrix Σ, using an ordinary least squares (OLS) standard estimation procedure equation by equation. Its vectorized form would be expressed as

$$vec(Y) = (I_m \otimes X)\Gamma + vec(u),\ where\ vec(u) \sim N(0, \Sigma \otimes I_t)$$

where the matrix of coefficients Γ presents $m \times [\#\,\text{lagged variables} + 1]$ dimension, which is the dynamic connectivity (also called effective connectivity), and the matrix of coefficients Σ represents the functional connectivity, where t represents the length of the Y series. The OLS estimation process can be translated by

$$\text{log-likelihood}(\hat{\Gamma}, \hat{\Sigma}|\text{observed data}) = \underset{\Gamma,\Sigma}{argmin}\left[\frac{1}{t}tr((Y - X\Gamma)\Sigma^{-1}(Y - X\Gamma)') - log|\Sigma^{-1}|\right].$$

However, as the graph model also includes small linear dependencies, implying a number of larger links, it results in an exponential increase in relation to the number of channels, jointly impacting the interpretation of complexity and the processing/interpretation of results. Therefore, it is usual to use a data-dependent threshold to remove the weak connections, but selecting an appropriate value can be different according to the experiment setting and goals (Garrison et al., 2015).

An alternative approach is to reduce the number of links during the connectivity matrix estimation, using sparse time series models. One widely adopted model is the GLASSO, used as a sparse VAR and proposed by Friedman et al. (2008); the method takes into account the sparsity toward the estimation on the functional connectivity. Inherently, the estimated connectivity matrices often have few links, but, despite maximizing the likelihood of the observed biosignals regarding the proposed theoretical model, they can lead to a distinct dynamic/effective connectivity estimation.

For instance, consider N multivariate normal observations of dimension p, with mean μ, and covariance Σ. Using the empirical covariance matrix, the problem is to penalize the negative log likelihood,

$$\text{log-likelihood}(\hat{\Gamma}, \hat{\Sigma}|\text{observed data}) =$$
$$\underset{\Gamma,\Sigma}{argmin}[\frac{1}{t}tr((Y - X\Gamma)\Sigma^{-1}(Y - X\Gamma)') - log|\Sigma^{-1}| +$$
$$\lambda_1 \sum_{i=1}^{G} \|\gamma_i\|_2 + \lambda_2 \sum_{k \neq k'} \|\Sigma^{-1}_{kk'}\|]$$

with λ_1 and λ_2 penalty parameters, γ_i is a subvector of Γ, $G = q^2$ total number of groups and k block coordinate descent derived from Σ (that is, shrinking only in part of the covariance matrix).

A generalization of this model is found in the TSCGM, proposed by Abegaz and Wit (2013), where sparse estimations of both effective and functional connectivity matrices are obtained. In this method, both matrices are estimated interactively: first, a sparse functional connectivity estimate is calculated with a non-sparse non-concave penalty (smoothly clipped absolute deviation, SCAD); and, later, sparse effective connectivity using the previous estimation as an initial value. This cycle is performed until it reaches convergence. For further details, please see Abegaz and Wit (2013).

TSCGM has been successfully applied to genetic data, and when applied to electroencephalograms, numerical experiments have shown a considerable reduction in the number of estimated connections. However, TSCGM also distorts the strength of some links, creating connections that were not present using a VAR model, because it relies on GLASSO to estimate the functional connectivity in each iteration.

The approach behind TSCGM is remarkable for increasing the sparsity of the estimations. Since the algorithmic implementation presented some issues during its application with biosignals, we introduced some adjustments. We also used a TSCGM-modified model that estimates the effective and functional connectivity that maximizes the loglikelihood of the model simultaneously using a Newton-type numerical optimization method. These methods are the non-linear optimization and iterative optimization. For more in-depth discussions toward sparsity profile, please see Benson et al. (2003), Wipf and Nagarajan (2008), and Rakotomamonjy (2011).

2.2.4. Multilayer Networks

Graph models are useful for describing and exploring patterns of dynamic/effective and functional/contemporaneous interactions of a given phenomenon. In human neuroscience experimentation, brain network connectivity activation can be recorded from the electrical impulse aiming to highlight interaction among areas.

Given the complexity of the brain, multilayer networks incorporate the multivariate and multi-scale information scheme (De Domenico, 2017). In general, multilayer networks can be seen as a collection of several distinct classic networks, which separately encode a specific type of information about the system as a layer, thus composing a multilayer network at the end. Those layers quantify some elements of similarities, such as (i) activity in different frequency bands, (ii) time-varying activity, (iii) activity of different tasks, and (iv) structural and functional connectivity.

Alongside this information, two important concepts about brain networks are essential; first the *functional connectivity*, which expresses the statistical correlation within a time step, also interpreted as contemporaneous interactions, and the second concept is related to *effective connectivity* in which it describes the dynamics of the current time in relation to previous times (this is the dynamics of the present response in relation to the lagged responses) (Friston, 2011).

2.2.5. Inferential Network Analyses

Let us start discussing the concept of conditional independence. It should be mentioned that part of this subsection was inspired by Højsgaard et al. (2012). Consider a collection of random variables $(X_\nu)\nu \in V$ associated along with a joint density, where V is a finite node set. Now, let us arbitrarily select three subsets of V (suppose A, B, and C); $X_A = (X_\nu)\nu \in A$ as well as for X_B and X_C. The statement X_A and X_B is said to be conditionally independent given X_C (that is, $A \perp\!\!\!\perp B \mid C$) if for each observation x_C of X_C, X_A, and X_B are independent in the conditional distribution given $X_C = x_c$. In this context, a generic probability function, $\pi(\ )$, defines the characterization $A \perp\!\!\!\perp B \mid C$ as

$$\pi(x_A, x_B \mid x_C) = \pi(x_A \mid x_C)\pi(x_B \mid x_C),$$

and rewriting as two functions $g(\ )$ and $h(\ )$, then

$$\pi(x_A, x_B, x_C) = g(x_A, x_C)h(x_B, x_C). \quad (1)$$

Whenever possible to describe the joint density as a product of functions, as in Equation 1, adopting the conditional independence approach, this is known as the factorization criterion. Hence $(X_\nu)\nu \in V$ can be represented as a set of joint densities, for instance, described as a parametric model, enabling us to use the factorization form, adopting the conditional independence relations between the variables. Often described as an undirected graph, conditional independence models unravel patterns out of a complex application. Suppose that $\mathcal{G} = (V, E)$ is an undirected graph with cliques (maximal complete subset) $C_1, \ldots, C_k$. The factorization form occurs if the joint density $\pi()$ of the variables in V is

$$\pi(x_\nu) = \prod_{i=1}^{k}(g_i(x_{C_i}))$$

where functions $g_1() \dots g_k()$ depend on x only through x_{Cj} according to the condition that $\pi()$ factorizes according to $\mathcal{G}$.

The global Markov property ensures that through the model it factorizes in all densities given $\mathcal{G}$, then $\mathcal{G}$ encodes the model's structure through the conditional independence; that is, whenever sets are separated by another in the graph, it is said that conditional independence happens under the model.

Nevertheless, there is not a unique equivalence/representation corresponding to patterns of conditional independences represented by a chain graph $\mathcal{G}$, guarded by the Markov properties. A chain graph is a combination of no bidirected edges and no semi-directed cycle graphs and may be seen as a natural generalization of undirected graphs and directed graphs that is acyclic (DAG).

For instance, the Markov properties can be described as two-step factorization; the first step represents the joint density as sub-parts; similar to a DAG, the search for the separation that maximizes the information is described as a graph form.

$$\pi(x_V) = \prod_{C \in \mathcal{C}} \pi(x_C \mid x_{pa(C)})$$

where C is the set of components of $\mathcal{G}$. Each conditional density $\pi(x_C \mid x_{pa(C)})$ will be based according to an undirected constructed graph;

That is, the form of subgraph $\mathcal{G}$ is induced by $C \cup$ pa(C), disregarding the directions, in relation to all possible pa (C). A hierarchy should be considered since some variable sets $pa(\nu)\nu \in V$, in relation to the variables in pa(ν) precede v. It is worth mentioning that the vertices of the graph represent the random variables, enabling us to identify the sets pa(ν) with the parents (descendent) of ν in the DAG.

Let us consider a chain graph (or complex network) for a given network defined by a set of vertices V and a set of edges E order in pairs, then each point is represented as $\mathcal{G} = (V, E)$. The interpretation of edges (also called links) can be also dynamic, as they are indexed in time, which represents the evolution of the interaction between pairs of vertices.

Time series data modeling can combine dynamic graphical models, which enables us to incorporate sparsity, aiming to estimate statistical causality and correlation across series. For the sake of simplicity, let us consider Markovian dynamics (time t relates only to time $t - 1$), which are similar to VAR(1), as

$$(a,\ b) \in V_t \times V_{t-1} \Leftrightarrow \Gamma_{ab} \neq 0$$

where effective connectivity is represented by the link between area a and b at consecutive time steps related to an element from Γ (points across time). Similarly, functional connectivity is represented by the estimated links associated with the effects corresponding to the precision matrix Σ (correlation within the same time period); this is related to the models' errors as

$$(a,\ b) \in V_t \times V_t \Leftrightarrow \Sigma_{ab} \neq 0.$$

Thus, a multivariate time series can be translated into a learning probabilistic connection network structure (as a graph model), aiming to estimate brain connectivity networks. The Dynamic Chain Graph Model (DCGM) creates a multivariate dynamic linear model for each chain component, and Wermuth and Lauritzen (1990) discuss the class of dynamic graphical models that enables us to estimate different signal phases and compare their structural relations. For instance, the dynamic/contemporaneous interactions between brain regions, presented by Costa et al. (2017), as a particular case of its theory in the neuroscience field.

2.2.6. TS Frequency Domain Approach

Brain activity can be collected as biosignals, composing the information flow from a group of connected neurons (called a neural circuit). These activities may seem at first to be pure noise, but between specific ranges, they may distinguish hidden patterns (Prado and West, 2010; Scheffer-Teixeira et al., 2013). Moreover, different frequency bands can contribute toward the brain mapping functionality by maximizing the information flow through the brain regions (according to the observed and latent components).

The literature presents changes in the frequency cuts (Fransson, 2005; Su et al., 2013), and those hubs might be very different when measured at different frequency bands. The findings concern the topological information measured from components at different frequencies (in hertz unit—Hz). Thus, such an enriched representation (decomposed TS signal) is more valuable than other aggregated representations (raw TS signal). For instance, some pass band ripple filters are Butterworth, Chebyshev, Elliptic or Cauer, and Finite Impulse Response (FIR) filter (for further details, please see Parks and Burrus, 1987).

Moreover, results presented in the literature (Newson and Thiagarajan, 2018; Wojcik et al., 2018) suggest that a healthy human brain operates at a transition point between independent and highly dependent frequency bands (e.g., represented as functional layers). EEG raw signals enable us to establish encoding the connectivity between the neural circuit, and are described within five frequency bands. It is reasonable to adopt the delimitation of biosignals in frequency bands theta (0.01–4 Hz), delta (4–8 Hz), alpha (8–16 Hz), beta (16–32 Hz), and gamma (32–49 Hz).

De Domenico (2017) suggests that brain activity may be represented in functional layers, without acting independently between them, adopting existing mechanisms for integration and segregation across different frequency bands. Thus, adopting multilayer techniques is shown to be potential in biomarkers as it integrates the whole concept of interdependence and is applicable in neurological and mental studies.

Thus, this work adopted the finite impulse response (FIR) filter, used to filter the limit of the signal coefficients given some order and frequency cutoff. Additionally, we added a correction using a Forward and Reverse filter applied to the FIR obtained signal to correct the phase distortion introduced by a one-pass filter, although this approach exerts a magnitude in the process in which it is equivalent to square responses. Both tools are implemented in R (Octave Forge, 2007), presented in the package signal.

The multiplex sparse dynamic model framework enables us to map the network connections, across different layers encoded as frequency bands (although integrated as De Domenico, 2017 suggests). Furthermore, the irreducibility of the multilayer functional representation of the human brain increases the need for multilayer analysis of the underlying architecture, targeting the identification of hubs.

3. RESULTS

Neuroscientists have attempted to understand brain connectivity through the functional and effective connectivity among brain areas, using biosignals, such as Electroencephalogram (EEG) or functional Magnetic Resonance Imaging (fMRI). This work tried to fathom the brain manipulation task related to the perception of verticality and posturography as a novelty targeting the development of a therapeutic approach for post-stroke patients.

A previous study performed the recording of high-density EEG together with the evaluation of visual vertical (VV). The authors mapped the high-density evoked potential with the evoked potential analysis discriminating the location of brain activation during VV evaluation. The authors verified brain activity during the task with a focus on the right lateral tempo-occipital cortex (Lopez et al., 2011). These physiological findings reaffirm the hypothesis of the dominance of the right cerebral hemisphere in the control of vertical perception. They also highlighted the right temporoparietal junction (TPJ) as a key point in the judgment of vertical orientation (Dieterich et al., 2003; Karnath and Dieterich, 2006; Pérennou et al., 2008; Baier et al., 2012).

Our group developed a promising brain stimulation protocol applied on right TPJ using a bipolar mount with conventional transcranial direct current stimulation (tDCS) and right hemisphere high-definition tDCS (HD-tDCS). We verified the efficacy and safety of this protocol in VV manipulation in healthy individuals.

For instance, Santos and Edwards (2019) pointed out that investigations toward the influence of cortical activity using non-invasive electromagnetic brain stimulation (NIBS) suggests understanding and treating verticality disorders as a neurorehabilitation. Thereby, Santos et al. (2018) implemented a protocol toward human verticality manipulation, using neuromodulation, on healthy participants aiming to understand the recovery of this intentional artificial brain lesions, briefly introduced in section 2.1.

Randomly selecting a single participant, **Figure 4** illustrates 5 min of brain response in each panel (raw EEG signals), selecting only seven channels (out of 256), and compares the signals during the resting state (top panel) vs. post-2 mA stimulation (bottom panel). Most of the selected EEG channels were located in the motor cortex; three channels were derived from the right hemisphere (164, 173, and 183) and located nearby the region placed the tDCS four electrodes. Then, three other channels were derived from the left hemisphere (66, 71, and 72) in which they are physiologically related to those selected from the right hemisphere; additionally the EEG channel 143 was placed in the parietal cortical region.

It can be observed in **Figure 4** that post-stimulation of the brain response amplitude from the raw EEG signals increased, which is more related to the hemisphere side to channels 183, 164, and 173 (related with the tDCS placed region). In addition, channel 66 has had its signal shifted up, which is physiologically explicable due to the polarity dependence created by the applied stimulus (directly related to channel 164, through the anodal input current electrode). According to Ombao and Ho (2006), Prado and West (2010), and De Domenico (2017), studies provide traces that brain connectivity may be better understood using frequency band decomposition limiting the influence of

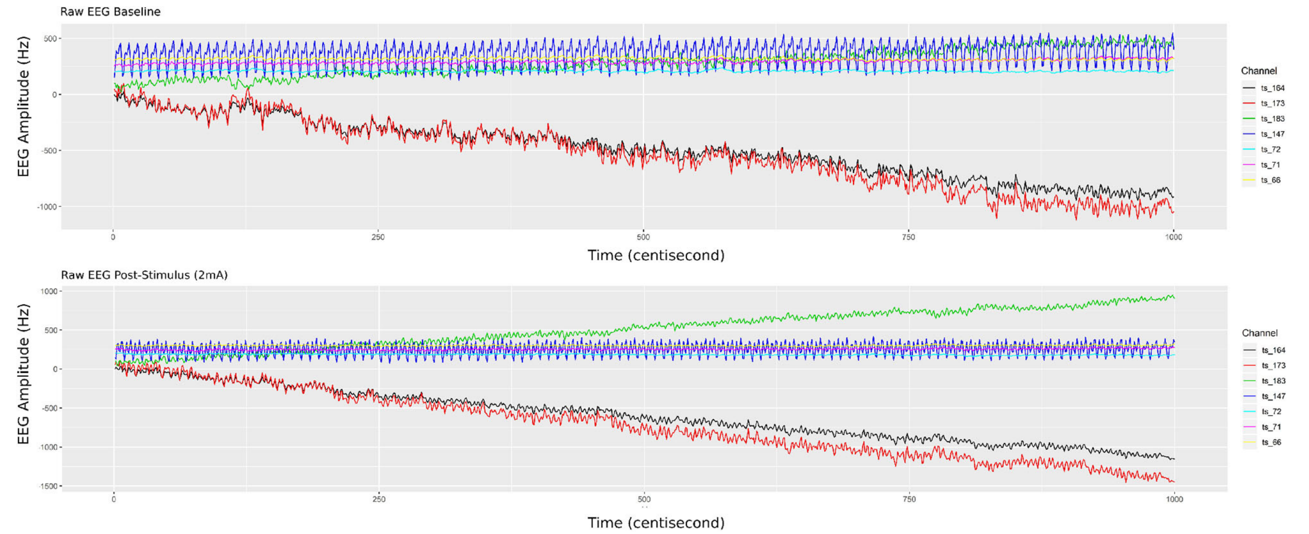


FIGURE 4 | Single participant EEG raw signals from 5 min of recorded biosignals (top panel) during the resting state and (bottom panel) after HD-tDCS–Anodal Center 2 mA–stimulation. The brain responses' amplitude (on the y-axis), from the raw EEG signals, increased after the stimulation.

noise in the brain signal and describing different brain tasks as oscillatory bands.

Initially, we filtered the raw EEG signals, adopting the FIR with pass-band filter, utilizing five fundamental bands of brain waves (alpha, beta, delta, gamma, and theta). **Figure 5** shows only the filtered signals related to the post-stimulation period, whereas elucidating the difference in band oscillation (signal phases) for each channel.

The channels located on the same brain hemisphere side as the neuromodulation (tDCS), presented greater oscillation. Thus, this dynamic may be translated/associated with the electrical transferred activity (energetic dissipation). This activity is expected given the rise of entropy through electrical synergy in this area (Nascimento et al., 2019).

The study of the human brain has been developing and generates an enormous amount of data, however, revealing the information extracted from this complex system is not trivial and, often, aggregating this information may lead to erroneous results (Fiecas and Ombao, 2011; Castruccio et al., 2016; Shen et al., 2016). Alternatively, the multilayer network approach provides a mathematical background to model and analyze complex data with multivariate and multi-scale information (Kivelä et al., 2014). Multiplex network shapes can be formatted using (i) activity in different frequency bands, (ii) time-varying activity, (iii) activity with respect to different tasks, and (iv) structural and functional connectivity.

Thus, estimations regarding the representation of a joint distribution of random variables are needed (the network structure). This procedure seeks to describe the causal relations across the brain regions. The Vector Autoregression (VAR) model would be appropriate to describe a brain connectivity network, nonetheless, it may present a high curse of dimensionality in large sets. This class of models presents a significant number of parameters to be estimated. Additionally,

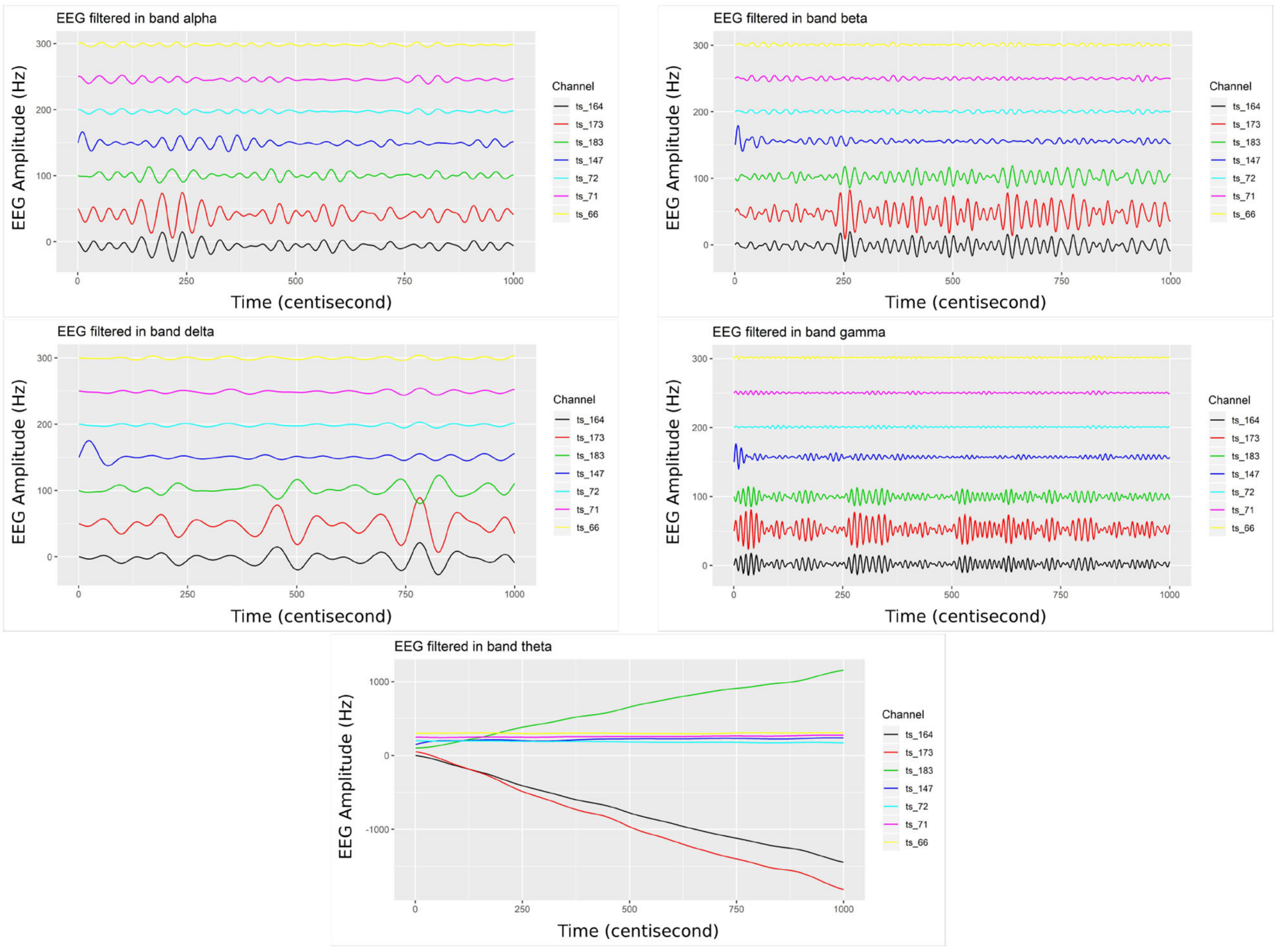


FIGURE 5 | Bandpower from the filtered EEG signals (top left) considering the alpha band, (top right) filter in the beta band, (middle left) in the delta band, (middle right) in the gamma band, and (bottom center) filter in the theta band. The EEG electrodes placed on the right-side brain hemisphere present higher dynamic/variation (channels 164, 173, and 183), related to post-2 mA stimulation.

shrinkage either in the data (such as PCA) or parameter spaces (like GLASSO and TSCGM) is not straightforward and may lead to misleading information.

The graphical LASSO (GLASSO) model, proposed by Friedman et al. (2008), estimates that matrices tended to be different from those determined by a classical VAR method. It was noticeable that non-sparse VAR estimation not only increased the sparsity of the effective connectivity matrix but also "created links" that did not appear before (based on our empirical analysis). These models present a high sensitivity to non-stationary series and might mislead the estimation point connections (given the shrinkage on the covariance matrix–Contemporaneous Effect–,thus changing the dynamic interactions).

Alternatively, TSCGM and TSCGM-modified was performed using a non-linear optimization over the log-likelihood, and iterative optimizing the log-likelihood (with l_1-norm and SCAD penalization, not only in the covariance matrix) (Abegaz and Wit, 2013). **Figure 6** shows the supra-adjacency matrix related with the functional connectivity, across seven EEG channels, comparing seven estimation methods (classic VAR, GLASSO, TSCGM, TSCGM non-linear l_1-norm, TSCGM non-linear SCAD, TSCGM-iterative l_1-norm, and TSCGM-iterative SCAD), for instance, only the performance of a single band (alpha).

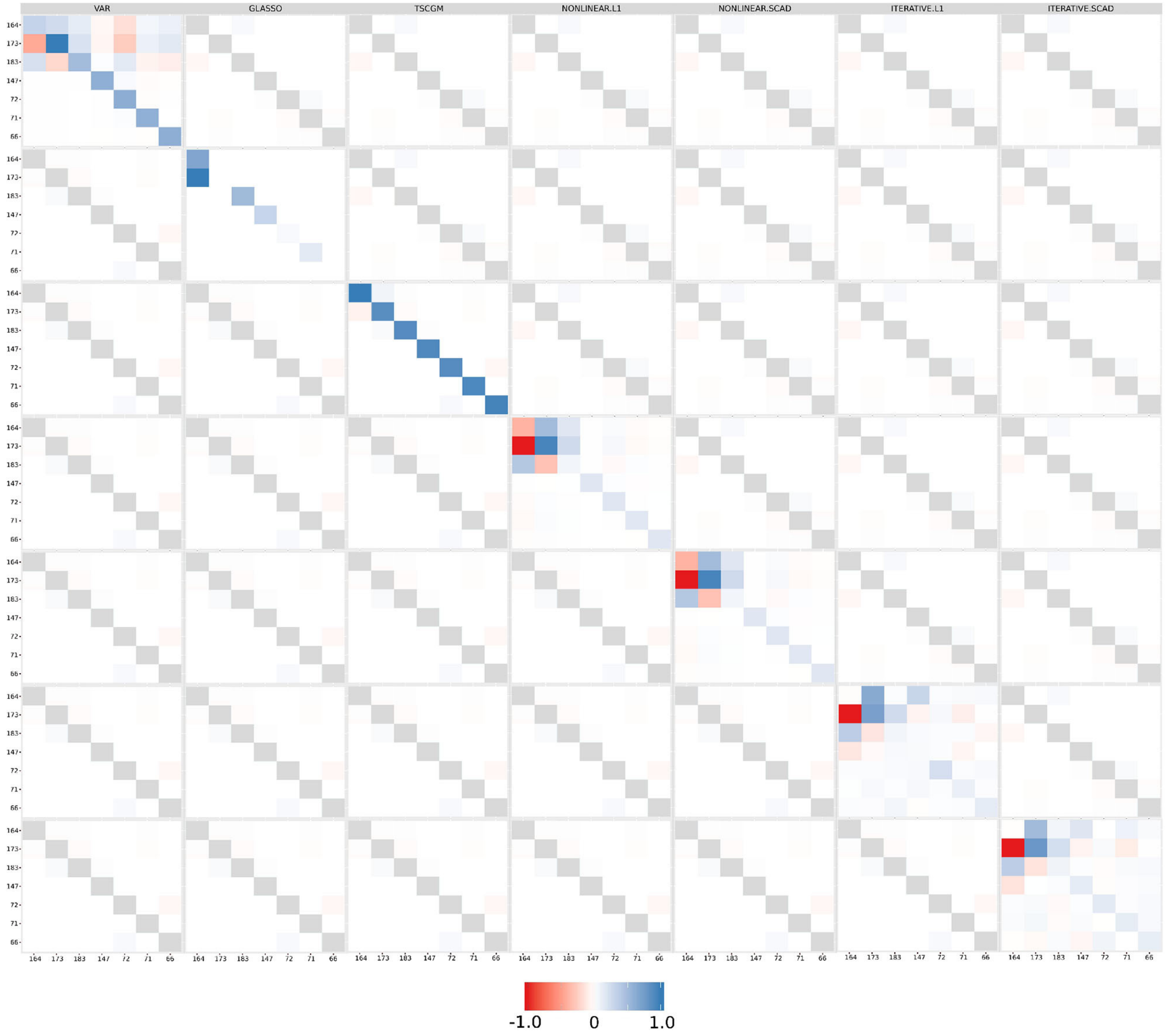


FIGURE 6 | Functional connectivity as the supra-adjacency matrix in which rows and columns form groups from the seven filtered EEG alpha frequency-band signals, throughout the methods (VAR, GLASSO, TSCGM, TSCGM non-linear l_1-norm and SCAD, and TSCGM-iterative l_1-norm and SCAD). The VAR method is the reference, whereas the target is to maintain the strong links and remove the weak using sparsity. The TSCGM non-linear provided a competitive insight preserving the structure and function of the human brain.

The VAR model includes weak linear dependencies, as mentioned in section 2.2, and it is desirable to use a data-dependent threshold to remove the weak connections without losing information. GLASSO and TSCGM led to different interpretations, compared to the VAR-estimated matrix. Nevertheless, TSCGM-modified with non-linear optimization using both l_1-norm and SCAD penalization maintained the strong links presented in the VAR but also eliminated the weak ones, therefore suggesting a competitive performance among the others. The same cannot be said for the TSCGMs-modified with iterative optimization.

Figure 7 shows the estimated brain dynamic/effective connectivity among the seven filtered channels (top figures) during the resting state and (bottom figures) post-stimulation, adopting the performance of TSCGM non-linear optimization using SCAD. That is, the brain illustrates with the correlation matrices the neuronal information floating connectivity (in different frequency-band signals).

No visual modification can be observed through the analysis of the alpha, beta, and delta bands, according to **Figure 7**. Gamma and theta bands show a slight change (considering the new estimated coefficient intensity during post-stimulus). In agreement with the present findings, previous results showed gamma band change after brain stimulation (Santos et al., 2018).

The results were similar to the findings observed in patients after stroke. Our data thus indicate that the proposed approach may be a promising tool for methodological-analysis toward the treatment of verticality error in stroke patients (Santos-Pontelli et al., 2016; Santos et al., 2018). In previous studies, Nascimento et al. (2019) compared HD-tDCS dose-response, adopting the

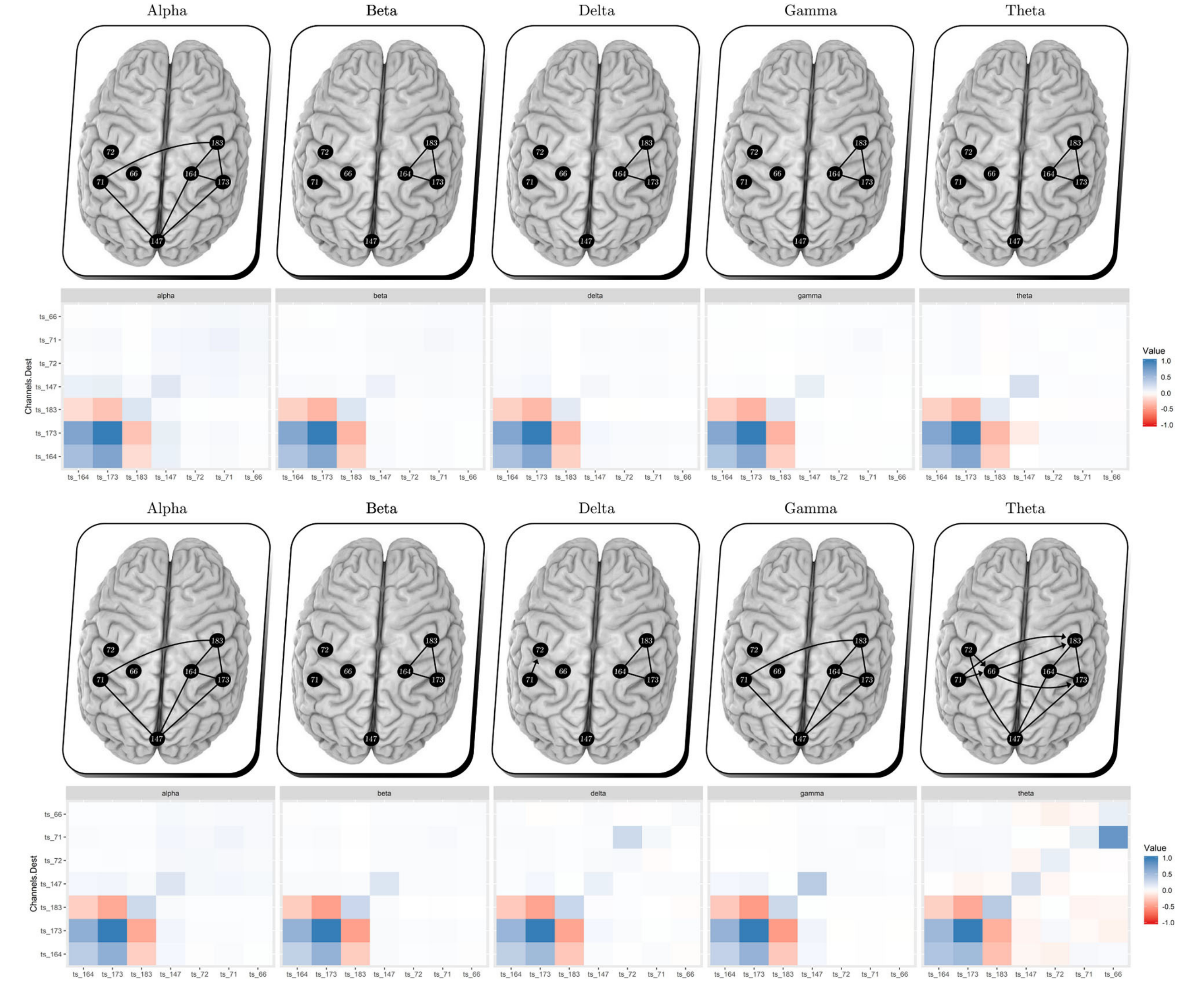


FIGURE 7 | Multiplex EEG signals, per bandpower, (top panels) resting state and (bottom panels) after unilateral HD-tDCS–Anodal Center 2 mA–stimulation. Comparing the resting state vs. post-stimulation, we obtained additional links within the gamma and theta bands during post-stimulus, which suggests an outgrowth in the electrical brain dynamic.

same protocol study, which included the placebo/sham HD-tDCS trail response, that by its statistical results, it helped to validate the sparse dynamic models' feasibility effects search of HD-tDCS and its pure effect.

4. FINAL REMARKS

This study aimed to implement and discuss the comparison of sparse methods toward parameter dimension shrinkage. Nevertheless, preserving information from empirical data is necessary to develop elements for brain manipulation intervention related to the perception of verticality and posturography as a novelty aimed at the recovery of post-stroke patients. The multilayer network approach enabled us to integrate the information retained given the electrical post-stimulus synergy (through different frequency bands).

The findings obtained in this paper contribute to the process of estimating the neuronal circuit connections, with robust inference and computational feasibility. Estimating a network structure can be a non-trivial (Chickering, 1996), highly complex task (Rodrigues et al., 2016), despite the fact that these sparse models showed to be promising, bypassing the false positives link estimation (results in **Figure 6**).

As demonstrated in the present work, the sparse models (using a dynamic linear model) combined with the frequency domain approach represented as the multilayer network implement to the neuroscience field the capability of interpreting/estimating the dynamic of the neural circuits based on EEG data in a comprehensive way. Moreover, we aimed to contribute with more in-depth data analysis toward the protocol (Santos et al., 2018), discussing its feasibility, enlightening the human manipulation intervention response dynamic.

This work is limited given that conclusions are based on a single participant response, whereas future works intend to extend this modeling using hierarchical models and interpretation of the entire sample and protocol. Liao et al. (2017) showed that the modular structures of brain networks completely vary across individuals. Thus, hierarchical modeling is required in the form of a set of state vectors for each chain component, as an exchangeable sample with the common mean. Therefore, future work shall explore the time-varying parameters, enclosed by the dynamic linear models, in a hierarchical version, suitable for interventions, such as those presented here, indexed in time.

DATA AVAILABILITY STATEMENT

The datasets generated for this study are available on request to the corresponding author.

ETHICS STATEMENT

The studies involving human participants were reviewed and approved by Joao P. Leite—Department of Neuroscience and Behavioral Sciences, Ribeirao Preto Medical School, University of São Paulo. The patients/participants provided their written informed consent to participate in this study.

AUTHOR CONTRIBUTIONS

DN and MP-O: statistical analysis, computational modeling, interpretation of the data, and manuscript writing. JL: study concept, interpretation of the data, and critical revision of the manuscript. DE: study concept and design of the clinical trial, interpretation of computational modeling, data analysis, and critical revision of the manuscript. FL: supervision of the statistical analysis, interpretation of the data, and critical revision of the manuscript. TS: study concept and design of the clinical trial, project management, data acquisition, supervision of the data analysis, data interpretation, and manuscript writing. All authors contributed to the article and approved the submitted version.

FUNDING

This research was partially supported by CNPq, FAPESP, and CAPES from Brazil. This research was carried out using the computational resources of the Center for Mathematical Sciences Applied to Industry (CeMEAI) funded by FAPESP (grant 2013/07375-0).

ACKNOWLEDGMENTS

The authors would like to acknowledge the help from Fabio Felix for **Figure 2**, from Evandro Ortigossa for the **Figure 6** and from Diego Cintra for the **Figure 7**, which they helped on part of the graphical development. Also, thanks to Diandra Favoretto for the clinical discussions.

REFERENCES

Abegaz, F., and Wit, E. (2013). Sparse time series chain graphical models for reconstructing genetic networks. *Biostatistics* 14, 586–599. doi: 10.1093/biostatistics/kxt005

Anacleto, O., Queen, C., et al. (2017). Dynamic chain graph models for time series network data. *Bayes. Anal.* 12, 491–509. doi: 10.1214/16-BA1010

Ardolino, G., Bossi, B., Barbieri, S., and Priori, A. (2005). Non-synaptic mechanisms underlie the after-effects of cathodal transcutaneous direct current stimulation of the human brain. *J. Physiol.* 568, 653–663. doi: 10.1113/jphysiol.2005.088310

Bach, F. R., and Jordan, M. I. (2004). Learning graphical models for stationary time series. *IEEE Trans. Signal Process.* 52, 2189–2199. doi: 10.1109/TSP.2004.831032

Baggio, J. A., Mazin, S. S., Alessio-Alves, F. F., Barros, C. G., Carneiro, A. A., Leite, J. P., et al. (2016). Verticality perceptions associate with postural control and functionality in stroke patients. *PLoS ONE* 11:e0150754. doi: 10.1371/journal.pone.0150754

Baier, B., Suchan, J., Karnath, H.-O., and Dieterich, M. (2012). Neural correlates of disturbed perception of verticality. *Neurology* 78, 728–735. doi: 10.1212/WNL.0b013e318248e544

Baxter, B. S., Edelman, B. J., Sohrabpour, A., and He, B. (2017). Anodal transcranial direct current stimulation increases bilateral directed brain connectivity during motor-imagery based brain-computer interface control. *Front. Neurosci.* 11:691. doi: 10.3389/fnins.2017.00691

Benson, H. Y., Shanno, D. F., and Vanderbei, R. J. (2003). "A comparative study of large-scale nonlinear optimization algorithms," in *High Performance Algorithms and Software for Nonlinear Optimization. Applied Optimization, Vol. 82*, eds G. Di Pillo and A. Murli (Boston, MA: Springer). doi: 10.1007/978-1-4613-0241-4_5

Brunoni, A. R., Nitsche, M. A., Bolognini, N., Bikson, M., Wagner, T., Merabet, L., et al. (2012). Clinical research with transcranial direct current stimulation (TDCS): challenges and future directions. *Brain Stimul.* 5, 175–195. doi: 10.1016/j.brs.2011.03.002

Cao, J., Wang, X., Liu, H., and Alexandrakis, G. (2018). Directional changes in information flow between human brain cortical regions after application of anodal transcranial direct current stimulation (TDCS) over Broca's area. *Biomed. Opt. Express* 9, 5296–5317. doi: 10.1364/BOE.9.005296

Carvalho, C. M., West, M., et al. (2007). Dynamic matrix-variate graphical models. *Bayes. Anal.* 2, 69–97. doi: 10.1214/07-BA204

Castruccio, S., Ombao, H., and Genton, M. G. (2016). A multi-resolution spatio-temporal model for brain activation and connectivity in fMRI data. *arXiv[Preprint].arXiv:1602.02435*.

Chern, J.-S., Lo, C.-Y., Wu, C.-Y., Chen, C.-L., Yang, S., and Tang, F.-T. (2010). Dynamic postural control during trunk bending and reaching in healthy adults and stroke patients. *Am. J. Phys. Med. Rehabil.* 89, 186–197. doi: 10.1097/PHM.0b013e3181c56287

Chickering, D. M. (1996). "Learning Bayesian Networks is NP-Complete," in *Learning from Data. Lecture Notes in Statistics*, Vol. 112, eds D. Fisher and H. J. Lenz (New York, NY: Springer), 121-130. doi: 10.1007/978-1-4612-2404-4_12

Cogiamanian, F., Vergari, M., Pulecchi, F., Marceglia, S., and Priori, A. (2008). Effect of spinal transcutaneous direct current stimulation on somatosensory evoked potentials in humans. *Clin. Neurophysiol.* 119, 2636–2640. doi: 10.1016/j.clinph.2008.07.249

Costa, L., Nichols, T., Smith, J. Q., et al. (2017). Studying the effective brain connectivity using multiregression dynamic models. *Braz. J. Probab. Stat.* 31, 765–800. doi: 10.1214/17-BJPS375

Day, B. L., and Cole, J. (2002). Vestibular-evoked postural responses in the absence of somatosensory information. *Brain* 125, 2081–2088. doi: 10.1093/brain/awf212

De Domenico, M. (2017). Multilayer modeling and analysis of human brain networks. *Giga Sci.* 6:gix004. doi: 10.1093/gigascience/gix004

Dieterich, M., Bense, S., Lutz, S., Drzezga, A., Stephan, T., Bartenstein, P., et al. (2003). Dominance for vestibular cortical function in the non-dominant hemisphere. *Cereb. Cortex* 13, 994–1007. doi: 10.1093/cercor/13.9.994

Edwards, D., Cortes, M., Datta, A., Minhas, P., Wassermann, E. M., and Bikson, M. (2013). Physiological and modeling evidence for focal transcranial electrical brain stimulation in humans: a basis for high-definition tdcs. *Neuroimage* 74, 266–275. doi: 10.1016/j.neuroimage.2013.01.042

Ferbert, A., Priori, A., Rothwell, J., Day, B., Colebatch, J., and Marsden, C. (1992). Interhemispheric inhibition of the human motor cortex. *J. Physiol.* 453, 525–546. doi: 10.1113/jphysiol.1992.sp019243

Fiecas, M., and Ombao, H. (2011). The generalized shrinkage estimator for the analysis of functional connectivity of brain signals. *Ann. Appl. Stat.* 1102–1125. doi: 10.1214/10-AOAS396

Fiori, V., Kunz, L., Kuhnke, P., Marangolo, P., and Hartwigsen, G. (2018). Transcranial direct current stimulation (TDCS) facilitates verb learning by altering effective connectivity in the healthy brain. *Neuroimage* 181, 550–559. doi: 10.1016/j.neuroimage.2018.07.040

Fransson, P. (2005). Spontaneous low-frequency bold signal fluctuations: an fMRI investigation of the resting-state default mode of brain function hypothesis. *Hum. Brain Mapp.* 26, 15–29. doi: 10.1002/hbm.20113

Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* 9, 432–441. doi: 10.1093/biostatistics/kxm045

Friston, K. J. (2011). Functional and effective connectivity: a review. *Brain Connect.* 1, 13–36. doi: 10.1089/brain.2011.0008

Garrison, K. A., Scheinost, D., Finn, E. S., Shen, X., and Constable, R. T. (2015). The (in) stability of functional brain network measures across thresholds. *Neuroimage* 118, 651–661. doi: 10.1016/j.neuroimage.2015.05.046

Gilio, F., Rizzo, V., Siebner, H. R., and Rothwell, J. C. (2003). Effects on the right motor hand-area excitability produced by low-frequency rtms over human contralateral homologous cortex. *J. Physiol.* 551, 563–573. doi: 10.1113/jphysiol.2003.044313

Gratton, C., Laumann, T. O., Nielsen, A. N., Greene, D. J., Gordon, E. M., Gilmore, A. W., et al. (2018). Functional brain networks are dominated by stable group and individual factors, not cognitive or daily variation. *Neuron* 98, 439–452. doi: 10.1016/j.neuron.2018.03.035

Højsgaard, S., Edwards, D., and Lauritzen, S. (2012). *Graphical Models With R*. Boston, MA: Springer Science & Business Media.

Karnath, H.-O., and Dieterich, M. (2006). Spatial neglect—a vestibular disorder? *Brain* 129, 293–305. doi: 10.1093/brain/awh698

Kivelä, M., Arenas, A., Barthelemy, M., Gleeson, J. P., Moreno, Y., and Porter, M. A. (2014). Multilayer networks. *J. Complex Netw.* 2, 203–271. doi: 10.1093/comnet/cnu016

Krystal, A. D., Prado, R., and West, M. (1999). New methods of time series analysis of non-stationary EEG data: eigenstructure decompositions of time varying autoregressions. *Clin. Neurophysiol.* 110, 2197–2206. doi: 10.1016/S1388-2457(99)00165-0

Lang, N., Siebner, H. R., Ward, N. S., Lee, L., Nitsche, M. A., Paulus, W., et al. (2005). How does transcranial dc stimulation of the primary motor cortex alter regional neuronal activity in the human brain? *Eur. J. Neurosci.* 22, 495–504. doi: 10.1111/j.1460-9568.2005.04233.x

Lee, L., Siebner, H. R., Rowe, J. B., Rizzo, V., Rothwell, J. C., Frackowiak, R. S., et al. (2003). Acute remapping within the motor system induced by low-frequency repetitive transcranial magnetic stimulation. *J. Neurosci.* 23, 5308–5318. doi: 10.1523/JNEUROSCI.23-12-05308.2003

Liao, X., Cao, M., Xia, M., and He, Y. (2017). Individual differences and time-varying features of modular brain architecture. *Neuroimage* 152, 94–107. doi: 10.1016/j.neuroimage.2017.02.066

Lopez, C., Mercier, M., Halje, P., and Blanke, O. (2011). Spatiotemporal dynamics of visual vertical judgments: early and late brain mechanisms as revealed by high-density electrical neuroimaging. *Neuroscience* 181, 134–149. doi: 10.1016/j.neuroscience.2011.02.009

Meyer, B.-U., Röricht, S., and Woiciechowsky, C. (1998). Topography of fibers in the human corpus callosum mediating interhemispheric inhibition between the motor cortices. *Ann. Neurol.* 43, 360–369. doi: 10.1002/ana.410430314

Nakao, H. (2016). Phase reduction approach to synchronisation of nonlinear oscillators. *Contemp. Phys.* 57, 188–214. doi: 10.1080/00107514.2015.1094987

Nascimento, D. C., Depetri, G., Stefano, L. H., Anacleto, O., Leite, J. P., Edwards, D. J., et al. (2019). Entropy analysis of high-definition transcranial electric stimulation effects on eeg dynamics. *Brain Sci.* 9:208. doi: 10.3390/brainsci9080208

Natarajan, K., Acharya, R., Alias, F., Tiboleng, T., and Puthusserypady, S. K. (2004). Nonlinear analysis of EEG signals at different mental states. *Biomed. Eng. Online* 3:7. doi: 10.1186/1475-925X-3-7

Newson, J. J., and Thiagarajan, T. C. (2018). EEG frequency bands in psychiatric disorders: a review of resting state studies. *Front. Hum. Neurosci.* 12:521. doi: 10.3389/fnhum.2018.00521

Nitsche, M. A., Liebetanz, D., Antal, A., Lang, N., Tergau, F., and Paulus, W. (2003). Modulation of cortical excitability by weak direct current stimulation-technical, safety and functional aspects. *Suppl. Clin. Neurophysiol.* 56, 255–276. doi: 10.1016/S1567-424X(09)70230-2

Octave Forge (2007). *Octave Forge Sourceforge Project*. Available online at: https://octave.sourceforge.io/ (accessed May 10, 2019).

Ombao, H., and Ho, M. R. (2006). Time-dependent frequency domain principal components analysis of multichannel non-stationary signals. *Comput. Stat. Data Anal.* 50, 2339–2360. doi: 10.1016/j.csda.2004.12.011

Parks, T. W., and Burrus, C. S. (1987). *Digital Filter Design*. New York, NY: Wiley-Interscience.

Pearl, J. (2014). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, CA: Elsevier.

Pérennou, D., Mazibrada, G., Chauvineau, V., Greenwood, R., Rothwell, J., Gresty, M., et al. (2008). Lateropulsion, pushing and verticality perception in hemisphere stroke: a causal relationship? *Brain* 131, 2401–2413. doi: 10.1093/brain/awn170

Petris, G., Petrone, S., and Campagnoli, P. (2009). *Dynamic Linear Models with* R. New York, NY: Springer.

Plewnia, C., Lotze, M., and Gerloff, C. (2003). Disinhibition of the contralateral motor cortex by low-frequency rtms. *Neuroreport* 14, 609–612. doi: 10.1097/00001756-200303240-00017

Prado, R., Molina, F., and Huerta, G. (2006). Multivariate time series modeling and classification via hierarchical var mixtures. *Comput. Stat. Data Anal.* 51, 1445–1462. doi: 10.1016/j.csda.2006.03.002

Prado, R., and West, M. (2010). *Time Series: Modeling, Computation, and Inference*. Hoboken, NJ: CRC Press.

Priori, A., Berardelli, A., Rona, S., Accornero, N., and Manfredi, M. (1998). Polarization of the human motor cortex through the scalp. *Neuroreport* 9, 2257–2260. doi: 10.1097/00001756-199807130-00020

Queen, C. M., and Smith, J. Q. (1993). Multiregression dynamic models. *J. R. Stat. Soc. B Methodol.* 55, 849–870. doi: 10.1111/j.2517-6161.1993.tb01945.x

Quintana, J. M., and West, M. (1987). An analysis of international exchange rates using multivariate DLM's. *J. R. Stat. Soc. D Stat.* 36, 275–281. doi: 10.2307/2348524

Rakotomamonjy, A. (2011). Surveying and comparing simultaneous sparse approximation (or group-lasso) algorithms. *Signal Process.* 91, 1505–1526. doi: 10.1016/j.sigpro.2011.01.012

Rodrigues, F. A., Peron, T. K. D., Ji, P., and Kurths, J. (2016). The kuramoto model in complex networks. *Phys. Rep.* 610, 1–98. doi: 10.1016/j.physrep.2015.10.008

Santos, T. E., and Edwards, D. (2019). Non-invasive brain stimulation to treat disorders of human verticality. *Neurol. Rehabil.* 25, 50–53. doi: 10.14624/NR1904011

Santos, T. E., Favoretto, D. B., Toostani, I. G., Nascimento, D. C., Rimoli, B. P., Bergonzoni, E., et al. (2018). Manipulation of human verticality using high-definition transcranial direct current stimulation. *Front. Neurol.* 9:825. doi: 10.3389/fneur.2018.00825

Santos-Pontelli, T. E., Rimoli, B. P., Favoretto, D. B., Mazin, S. C., Truong, D. Q., Leite, J. P., et al. (2016). Polarity-dependent misperception of subjective visual vertical during and after transcranial direct current stimulation (TDCS). *PLoS ONE* 11:e0152331. doi: 10.1371/journal.pone.0152331

Scheffer-Teixeira, R., Belchior, H., Leao, R. N., Ribeiro, S., and Tort, A. B. (2013). On high-frequency field oscillations (>100 Hz) and the spectral leakage of spiking activity. *J. Neurosci.* 33, 1535–1539. doi: 10.1523/JNEUROSCI.4217-12.2013

Schlögl, A., and Supp, G. (2006). Analyzing event-related eeg data with multivariate autoregressive parameters. *Prog. Brain Res.* 159, 135–147. doi: 10.1016/S0079-6123(06)59009-0

Shen, Y., Baingana, B., and Giannakis, G. B. (2016). Nonlinear structural vector autoregressive models for inferring effective brain network connectivity. *arXiv[Preprint].arXiv:1610.06551*.

Siebner, H., Peller, M., Willoch, F., Minoshima, S., Boecker, H., Auer, C., et al. (2000). Lasting cortical activation after repetitive tms of the motor cortex: a glucose metabolic study. *Neurology* 54, 956–963. doi: 10.1212/WNL.54.4.956

Su, L., Wang, L., Shen, H., Feng, G., and Hu, D. (2013). Discriminative analysis of non-linear brain connectivity in schizophrenia: an fmri study. *Front. Hum. Neurosci.* 7:702. doi: 10.3389/fnhum.2013.00702

Tibshirani, R., Bien, J., Friedman, J., Hastie, T., Simon, N., Taylor, J., et al. (2012). Strong rules for discarding predictors in lasso-type problems. *J. R. Stat. Soc. B Stat. Methodol.* 74, 245–266. doi: 10.1111/j.1467-9868.2011.01004.x

Wermuth, N., and Lauritzen, S. L. (1990). On substantive research hypotheses, conditional independence graphs and graphical chain models. *J. R. Stat. Soc. B Methodol.* 52, 21–50. doi: 10.1111/j.2517-6161.1990.tb01771.x

West, M., and Harrison, J. (1989). *Bayesian Forecasting and Dynamic Models*. London: Springer Science & Business Media.

West, M., Harrison, P. J., and Migon, H. S. (1985). Dynamic generalized linear models and bayesian forecasting. *J. Am. Stat. Assoc.* 80, 73–83. doi: 10.1080/01621459.1985.10477131

Winter, D. A. (1995). Human balance and posture control during ding and walking. *Gait Posture* 3, 193–214. doi: 10.1016/0966-6362(96)82849-9

Wipf, D., and Nagarajan, S. (2008). *Iterative Reweighted 1 and 2 Methods for Finding Sparse Solutions*. Technical Report, UC San Francisco, San Francisco, CA, United States.

Wojcik, G. M., Masiak, J., Kawiak, A., Kwasniewicz, L., Schneider, P., Polak, N., et al. (2018). Mapping the human brain in frequency band analysis of brain cortex electroencephalographic activity for selected psychiatric disorders. *Front. Neuroinformatics* 12:27. doi: 10.3389/fninf.2018.00027

Zeemering, S. (2015). *Sparse estimation: applications in atrial fibrillation* (Ph.D. thesis), Maastricht University, Maastricht, Netherlands.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Applied Mathematics and Statistics

ORIGINAL RESEARCH
published: 17 February 2021
doi: 10.3389/fams.2020.616658

Exploiting Multiple Timescales in Hierarchical Echo State Networks

Luca Manneschi[1], *Matthew O. A. Ellis*[1], *Guido Gigante*[2], *Andrew C. Lin*[3,4], *Paolo Del Giudice*[2†] *and Eleni Vasilaki*[1*†]

[1]*Department of Computer Science, The University of Sheffield, Sheffield, United Kingdom,* [2]*National Center for Radiation Protection and Computational Physics, Italian Institute of Health, Rome, Italy,* [3]*Department of Biomedical Science, The University of Sheffield, Sheffield, United Kingdom,* [4]*Neuroscience Institute, The University of Sheffield, Sheffield, United Kingdom*

Echo state networks (ESNs) are a powerful form of reservoir computing that only require training of linear output weights while the internal reservoir is formed of fixed randomly connected neurons. With a correctly scaled connectivity matrix, the neurons' activity exhibits the echo-state property and responds to the input dynamics with certain timescales. Tuning the timescales of the network can be necessary for treating certain tasks, and some environments require multiple timescales for an efficient representation. Here we explore the timescales in hierarchical ESNs, where the reservoir is partitioned into two smaller linked reservoirs with distinct properties. Over three different tasks (NARMA10, a reconstruction task in a volatile environment, and psMNIST), we show that by selecting the hyper-parameters of each partition such that they focus on different timescales, we achieve a significant performance improvement over a single ESN. Through a linear analysis, and under the assumption that the timescales of the first partition are much shorter than the second's (typically corresponding to optimal operating conditions), we interpret the feedforward coupling of the partitions in terms of an effective representation of the input signal, provided by the first partition to the second, whereby the instantaneous input signal is expanded into a weighted combination of its time derivatives. Furthermore, we propose a data-driven approach to optimise the hyper-parameters through a gradient descent optimisation method that is an online approximation of backpropagation through time. We demonstrate the application of the online learning rule across all the tasks considered.

Keywords: reservoir computing (RC), echo state network (ESN), timescales, hyperparameter adaptation, backpropagation through time

OPEN ACCESS

Edited by:
Andre Gruning,
University of Surrey, United Kingdom

Reviewed by:
Petia D. Koprinkova-Hristova,
Institute of Information and Communication Technologies (BAS), Bulgaria
Igor Farkaš,
Comenius University, Slovakia

Correspondence:
Eleni Vasilaki
e.vasilaki@sheffield.ac.uk

[†]*These authors share senior authorship*

Specialty section:
This article was submitted to Dynamical Systems, a section of the journal Frontiers in Applied Mathematics and Statistics

Received: *12 October 2020*
Accepted: *29 December 2020*
Published: *17 February 2021*

Citation:
Manneschi L, Ellis MOA, Gigante G, Lin AC, Del Giudice P and Vasilaki E (2021) Exploiting Multiple Timescales in Hierarchical Echo State Networks. Front. Appl. Math. Stat. 6:616658. doi: 10.3389/fams.2020.616658

1 INTRODUCTION

The high inter-connectivity and asynchronous loop structure of Recurrent Neural Networks (RNNs) make them powerful techniques for processing temporal signals [1]. However, the complex inter-connectivity of RNNs means that they cannot be trained using the conventional back-propagation (BP) algorithm [2] used in feed-forward networks, since each neuron's state depends on other neuronal activities at previous times. A method known as Back-Propagation-Through-Time (BPTT) [3], which relies on an unrolling of neurons' connectivity through time to propagate the error signal to earlier time states, can be prohibitively complex for large networks or time series. Moreover, BPTT is not considered biologically plausible as neurons must retain memory of their activation over the length of the input and the error signal must be propagated backwards with symmetric synaptic weights [4].

Many of these problems can be avoided using an alternative approach: reservoir computing (RC). In the subset of RC networks known as Echo State networks, a fixed "reservoir" transforms a temporal input signal in such a way that only a single layer output perceptron needs to be trained to solve a learning task. The advantage of RC is that the reservoir is a fixed system that can be either computationally or physically defined. Since it is fixed it is not necessary to train the reservoir parameters through BPTT, making RC networks much simpler to train than RNNs. Furthermore, the random structure of a RC network renders the input history over widely different time-scales, offering a representation that can be used for a wide variety of tasks without optimising the recurrent connectivity between nodes.

Reservoirs have biological analogues in cerebellum-like networks (such as the cerebellum, the insect mushroom body and the electrosensory lobe of electric fish), in which input signals encoded by relatively few neurons are transformed via "expansion re-coding" into a higher-dimensional space in the next layer of the network, which has many more neurons than the input layer [5–8]. This large population of neurons (granule cells in the cerebellum; Kenyon cells in the mushroom body) acts as a reservoir because their input connectivity is fixed and learning occurs only at their output synapses. The principal neurons of the "reservoir" can form chemical and electrical synapses on each other (e.g., Kenyon cells: [9–11]), analogous to the recurrent connectivity in reservoir computing that allows the network to track and transform temporal sequences of input signals. In some cases, one neuronal layer with recurrent connectivity might in turn connect to another neuronal layer with recurrent connectivity; for example, Kenyon cells of the mushroom body receive input from olfactory projection neurons of the antennal lobe, which are connected to each other by inhibitory and excitatory interneurons [12, 13]. Such cases can be analogised to hierarchically connected reservoirs. In biological systems, it is thought that transforming inputs into a higher-dimensional neural code in the "reservoir" increases the associative memory capacity of the network [5]. Moreover, it is known that for the efficient processing of information unfolding in time, which requires networks to dynamically keep track of past stimuli, the brain can implement ladders of neural populations with hierarchically organised "temporal receptive fields" [14].

The same principles of dimensional expansion in space and/or time apply to artificial RC networks, depending on the non-linear transformation of the inputs into a representation useful for learning the task at the single linear output layer. We focus here on a popular form of RC called Echo State Networks [15], where the reservoir is implemented as a RNN with a fixed, random synaptic connection matrix. This connection matrix is set so the input "echoes" within the network with decaying amplitude. The performance of an Echo State Network depends on certain network hyper-parameters that need to be optimised through grid search or explicit gradient descent. Given that the dependence of the network's performance on such hyper-parameters is both non-linear and task-dependent, such optimisation can be tedious.

Previous works have studied the dependence of the reservoir properties on the structure of the random connectivity adopted, studying the dependence of the reservoir performance on the parameters defining the random connectivity distribution, and formulating alternatives to the typical Erdos-Renyi graph structure of the network [16–18]. In this sense, in [17] a model with a regular graph structure has been proposed, where the nodes are connected forming a circular path with constant shortest path lengths equal to the size of the network, introducing long temporal memory capacity by construction. The memory capacity has been studied previously for network parameters such as the spectral radius (ρ) and sparsity; in general memory capacity is higher for ρ close to one and low sparsity, but high memory capacity does not guarantee high prediction [19, 20]. ESNs are known to perform optimally when at the "edge of criticality" [21], where low prediction error and high memory can be achieved through network tuning.

More recently, models composed of multiple reservoirs have gathered the attention of the community. From the two ESNs with lateral inhibition proposed in [22], to the hierarchical structure of reservoirs first analyzed by Jaeger in [23], these complex architectures of multiple, multilayered reservoirs have shown improved generalisation abilities over a variety of tasks [23–25]. In particular, the works [26, 27] have studied different dynamical properties of such hierarchical structures of ESNs, while [28] have proposed hierarchical (or deep) ESNs with projection encoders between layers to enhance the connectivity of the ESN layers. The partitioning (or modularity) of ESNs was studied by [29], where the ratio of external to internal connections was varied. By tuning this partitioning performance can be increased on memory or recall tasks. Here we demonstrate that one of the main reasons to adopt a network composed by multiple, pipelined sub-networks, is the ability to introduce multiple timescales in the network's dynamics, which can be important in finding optimal solutions for complex tasks. Examples of tasks that require such properties are in the fields of speech, natural language processing, and reward driven learning in partially observable Markov decision processes [30]. A hierarchical structure of temporal kernels [31], as multiple connected ESNs, can discover higher level features of the input temporal dynamics. Furthermore, while a single ESN can be tuned to incorporate a distribution of timescales with a prefixed mode, optimising the system hyper-parameters to cover a wide range of timescales can be problematic.

Here, we show that optimisation of hyper-parameters can be guided by analysing how these hyper-parameters are related to the timescales of the network, and by optimising them according to the temporal dynamics of the input signal and the memory required to solve the considered task. This analysis improves performance and reduces the search space required in hyper-parameter optimisation. In particular, we consider the case where an ESN is split into two sections with different hyper-parameters resulting in separate temporal properties. In the following, we will first provide a survey of timescales in ESNs before presenting the comparative success of these hierarchical ESNs on three different tasks. The first is the non-linear auto-regressive moving average 10 (NARMA10) task which requires both memory and fast non-linear transformation of the input. Second, we explore the performance of the network in a reconstruction and state "perception" task with different levels of external white noise applied on the input signal. Finally, we apply the hierarchical ESN to a permuted sequential MNIST classification task, where the usual MNIST hand written digit database is serialised and permuted as a 1 dimensional time-series.

2 SURVEY OF TIMESCALES IN ECHO STATE NETWORKS

We begin by describing the operations of an ESN and present a didactic survey of their inherent timescales, which will be drawn upon in later sections to analyze the results.

As introduced in the previous section, an ESN is a recurrent neural network and the activity, $\mathbf{x}(t)$, of the neurons due to a temporal input signal $\mathbf{s}(t)$ is given by

$$\mathbf{x}(t+\delta t) = (1-\alpha)\mathbf{x}(t) + \alpha f(\mathbf{h}(t)), \tag{1}$$

$$\mathbf{h}(t) = \gamma \mathbf{W}_{\text{in}}\mathbf{s}(t) + \rho \mathbf{W}\mathbf{x}(t), \tag{2}$$

where $\mathbf{W}$ is a possibly sparse random matrix defining the connectivity of the network, $\mathbf{W}_{\text{in}}$ defines the input adjacency matrix, and γ is a rescaling factor of the input weights. $\alpha = \delta t/\tau$ is the leakage term of the node, and ρ is a scaling factor for the spectral radius of the connectivity matrix and will be discussed in more detail in the following. $f()$ is a non-linear function, which in this work we define as the hyperbolic tangent. To ensure that the network exhibits the Echo-State property, and so that the activity does not saturate, the initial random connectivity matrix, $\mathbf{W}$, is rescaled by its maximum eigenvalue magnitude (spectral radius), $|\lambda_{\mathbf{W}}^{\max}| = \max|\text{eig}(\mathbf{W})|$, thus ensuring a unitary spectral radius which can be tuned using ρ as a hyper-parameter. In practice, $\mathbf{W}$ is constructed from a matrix of Normally distributed random numbers and the sparseness is enforced by randomly setting to zero a fixed proportion of these elements. Typically 10 non-zero connections per node are retained in $\mathbf{W}$.

The timescales of this dynamical system are closely linked to the specific structure of $\mathbf{W}$ and to the two hyper-parameters; α and ρ. Since α is the leakage rate, it directly controls the retention of information from previous time steps, while ρ specifies the maximum absolute magnitude of the eigenvalues and as such tunes the decay time of internal activity of the network. Thus, the basic hyper-parameters that need to be set are γ, α and ρ. Considering the nonlinear dependence of the network performance on these values and the task-dependent nature of an efficient parameterisation, this process can be challenging. Such hyper-parameters are commonly optimised through a grid search or through explicit gradient descent methods in online learning paradigms [32]. However, the fine tuning procedure can be guided, and the searchable space reduced, using a simple analysis of the hyper-parameters' relation to the timescales of the network, the external signal's temporal dynamics, and the memory required to solve the considered task.

Considering that the eigenvalues $\lambda_{\mathbf{W}}$ of the connectivity matrix are inside the imaginary unit circle due to the normalisation procedure described previously, and that α is a constant common to all neurons, the eigenvalues of the linearised system given by **Eq. 1** are

$$\lambda = 1 - \alpha(1 - \rho\lambda_{\mathbf{W}}). \tag{3}$$

This corresponds to a rescaling of value $\alpha\rho$ and to a translation of value $1-\alpha$ across the real axis of the original $\lambda_{\mathbf{W}}$. This operation on the eigenvalues of $\mathbf{W}$ is depicted in **Figure 1A**. Thus, considering that each eigenvalue λ_i can be decomposed in its corresponding exponential decaying part $\exp(-\delta t/\tau_i)$ and its oscillatory imaginary component, the timescales of the linearised system are

$$\tau = \frac{\delta t}{1 - \text{Re}(\lambda)} \tag{4}$$

$$= \frac{\delta t}{\alpha(1 - \rho\text{Re}(\lambda_{\mathbf{W}}))} \tag{5}$$

When the connectivity matrix, $\mathbf{W}$, is given by a sparse matrix with non-zero elements drawn randomly from a uniform distribution with the range $[-1, 1]$, then the corresponding eigenvalues will be uniformly distributed within a circle with a radius of $\max(|\lambda_{\mathbf{W}}|)$ in the complex plane [33]. These eigenvalues are then re-scaled by $\max(|\lambda_{\mathbf{W}}|)$ to ensure they are within the unit circle. The distribution of the eigenvalues then reveals the distribution of timescales of the linearised system. Indeed, given $p(\text{Re}(\lambda), \text{Im}(\lambda))$, the distribution of timescales can be found through computation of the marginal $p(\text{Re}(\lambda)) = \int p(\text{Re}(\lambda), \text{Im}(\lambda)) d\text{Im}(\lambda)$ and the change of variable defined in **Eq. 5**, giving

$$p(\tau) = \frac{2\delta t^2}{\pi\alpha^2\rho^2\tau^2}\sqrt{\alpha^2\rho^2 - (\alpha - \delta t/\tau)^2} \tag{6}$$

Importantly we note that while the eigenvalues are uniformly distributed over the unit circle, the timescales are not due to the inverse relationship between them. The resulting distribution of the linearised system, shown in **Figure 1B** (red line), is in excellent agreement with the numerically computed distribution for a single ESN (black points + shaded area).

The analytical form of the distribution, together with **Eq. 5**, allows us to explicitly derive how changes in α and ρ affect the network timescales. Notably we can obtain analytical expression for the minimum, maximum and most probable (peak of the distribution) timescale:

$$\tau_{\min} = \frac{\delta t}{\alpha(1+\rho)}, \tag{7}$$

$$\tau_{\max} = \frac{\delta t}{\alpha(1-\rho)}, \tag{8}$$

$$\tau_{\text{peak}} = \frac{5\delta t}{4\alpha(1-\rho^2)}\left(1 - \sqrt{1 - \frac{24}{25}(1-\rho^2)}\right) \tag{9}$$

where **Eqs. 8** and **7** can be derived directly from **Eq. 5**, while **Eq. 9** follows from maximisation of **Eq. 6**. As expected, α strongly

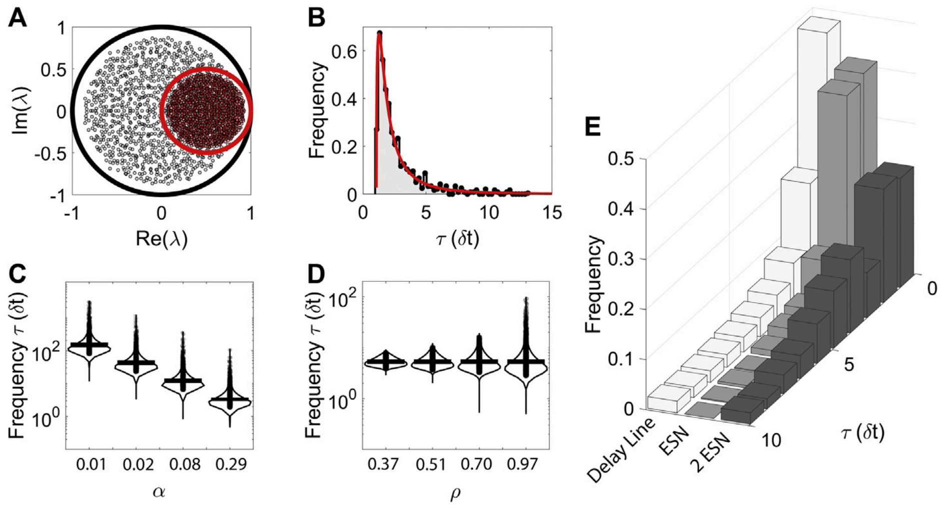


FIGURE 1 | The analysis of the timescales of the system in the linear regime can guide the search for the optimal values of the hyper-parameters α and ρ. **(A)**: Translation and scaling of the eigenvalues of the system due to the presence of the leakage factor. **(B)**: Example of distribution of timescales, computed analytically (red line) and computationally (black points) estimated from the eigenvalues of **W**. **(C)**: Pirate plot of the distributions of timescales as α increases. Both axes are logarithmic. Higher α values correspond to longer timescales and to a more compressed range of timescales (logarithmic *y*-axis). **(D)**: Pirate plot of the distributions of timescales: as ρ increases, the range of timescales expands. Again, both axes are logarithmic. **(E)**: Example distributions of timescales for reservoirs with different connectivity structure. From left to right, a delay line, single ESN, 2 ESNs (connected and unconnected, see text for the reason why the timescales for these two structures are the same in the linear regime). The higher complexity of the models reported is reflected in a richer distribution of timescales.

affects all these three quantities; interestingly, though, α does not influence the relative range of the distribution, $\tau_{\max}/\tau_{\min} = (1+\rho)/(1-\rho)$. Indeed α plays the role of a unit of measure for the τ's, and can then be used to scale the distribution in order to match the relevant timescales for the specific task. On the other hand, ρ does not strongly affect the shape of the distribution, but determines how dispersed the τ's are. Given the finite number of τ's expressed by a finite ESN, the hyper-parameter ρ can be used to balance the raw representation power of the network (how wide the range of timescales is) with the capacity to approximate any given timescale in that range. **Figures 1C,D** give a more detailed view of how the distribution of timescales changes as α and ρ, respectively, vary; note the logarithmic scale on the *y*-axis, that makes the dependence on α linear. The link between the eigenvalues and the reservoir dynamics can be shown through the analysis of the network response to an impulsive signal, shown in **Section 2 Supplementary Material** where the experimental activities are compared with the theoretical ones expected from the linearised system.

2.1 Hierarchical Echo-State Networks

Different studies have proposed alternatives to the random structure of the connectivity matrix of ESNs, formulating models of reservoirs with regular graph structures. Examples include a delay line [17], where each node receives and provides information only from the previous node and the following one respectively, and the concentric reservoir proposed in [18], where multiple delay lines are connected to form a concentric structure. Furthermore, the idea of a hierarchical architecture of ESNs, where each ESN is connected to the preceding and following one, has attracted the reservoir computing community for its capability of discovering higher level features of the external signal [34]. **Figure 2** schematically shows the architecture for **(A)** a single ESN, **(B)** 2 sub-reservoir hierarchical ESN for which the input is fed into only the first sub-reservoir which in turn feeds into the second and **(C)** a parallel ESN, where two unconnected sub-reservoirs receive the same input. These hierarchical ESNs are identical to the 2 layer DeepESN given by [27]. A general ensemble of interacting ESNs can be described by

$$\mathbf{x}^{(k)}(t+\delta t) = \left(1-\alpha^{(k)}\right)\mathbf{x}^{(k)} + \alpha^{(k)} f\left(\mathbf{h}^{(k)}(t)\right), \tag{10}$$

$$\mathbf{h}^{(k)}(t) = \gamma^{(k)}\mathbf{W}_{\text{in}}^{(k)}\mathbf{s}^{(k)}(t) + \sum_{l}^{N_{\text{ESN}}} \rho^{(kl)}\mathbf{W}^{(kl)}\mathbf{x}^{(l)}(t), \tag{11}$$

where the parameters have the similar definitions as in the case of a single ESN in **Eq. 1**. The index *k* indicates the network number and N_{ESN} is the total number of networks under consideration. In a hierarchical structure of ESNs $\mathbf{W}^{(kl)} \neq \mathbf{0}$ for $k = l$ or $k = l + 1$ only, and $\mathbf{W}^{(kl)}$ can be drawn from any desirable distribution thanks to the absence of feedback connections to higher-order reservoirs. Indeed, in this case, the necessary condition for the Echo-State network property is that all the inner connectivity matrices $\mathbf{W}^{(kk)}$ have eigenvalues with an absolute value less than one. Furthermore, in the typical hierarchical structure proposed in previous works [23–25, 27, 35], the input is fed to the first network only, and $\mathbf{W}_{\text{in}}^{(k)} \neq \mathbf{0}$ if $k = 1$ only. We emphasise that the values of $\alpha^{(k)}$ and $\rho^{(k)}$, which are closely related to the timescales and repertoire of dynamics of network number *k* (and, in the case of hierarchical reservoirs, also to all subsequent networks), do not have to be equal for

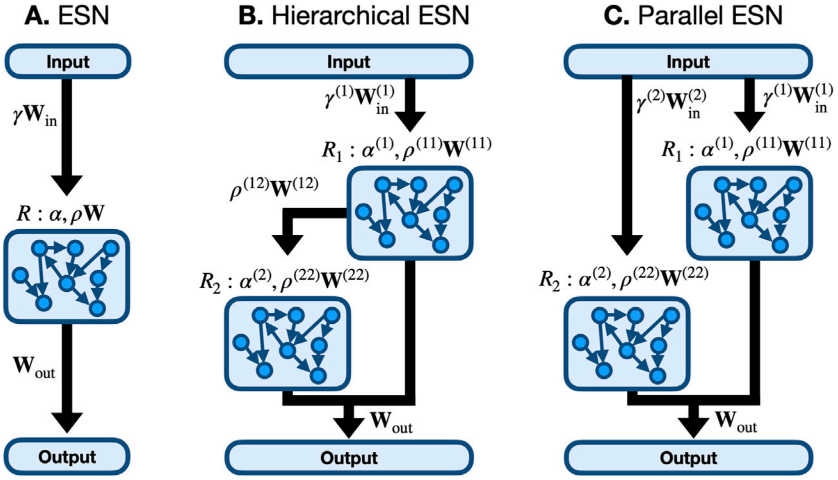

A. ESN
B. Hierarchical ESN
C. Parallel ESN
Input
$\gamma\mathbf{W}_{\text{in}}$
$R: \alpha, \rho\mathbf{W}$
$\mathbf{W}_{\text{out}}$
Output
$\gamma^{(1)}\mathbf{W}_{\text{in}}^{(1)}$
$R_1: \alpha^{(1)}, \rho^{(11)}\mathbf{W}^{(11)}$
$\rho^{(12)}\mathbf{W}^{(12)}$
$R_2: \alpha^{(2)}, \rho^{(22)}\mathbf{W}^{(22)}$
$\gamma^{(2)}\mathbf{W}_{\text{in}}^{(2)}$

FIGURE 2 | Single and hierarchical echo-state network (ESN) architectures. **(A)**: A single ESN with internally connected nodes with a single set of hyper-parameters α and ρ. **(B)**: A hierarchical ESN composed of 2 connected reservoirs where the input is fed into reservoir one only and the connection is unidirectional from R1 to R2, which is identical to the 2 layer DeepESN of [27]. **(C)**: A parallel (or unconnected hierarchical) ESN where the network is partitioned into 2 reservoirs, R1 and R2, which each receive the input and provide output but have distinct hyper-parameters.

each ESN, but can be chosen differently to fit the necessity of the task. In particular, some tasks could require memory over a wide range of timescales that could not effectively be covered by a single ESN.

In **Figure 1E** we show examples of the timescale distributions of the corresponding linearised dynamical systems for different ESN structures, from the simple delay line model to the higher complexity exhibited from two hierarchical ESNs. In order from left to right, the histograms of timescales are for a delay line, a single ESN, and two ESNs (whether hierarchically connected or unconnected; see below for clarification). All the models share an ESN with $\rho = 0.9$ and $\alpha = 0.9$; where present, the second reservoir has $\alpha = 0.2$. By construction, the richness and range of timescales distributions reported increases with the complexity of the models. However, we note how a simple delay line could exhibit longer temporal scales than the other structures analyzed thanks to its constant and high value of minimum path length between any pairs of nodes. Nevertheless, its limited dynamics restricts its application to simple tasks. The cases with two ESNs show a bimodal distribution corresponding to the two values of α.

Yet, the spectrum of the eigenvalues of the linearised system is only partially informative of the functioning and capabilities of an ESN. This is clearly demonstrated by the fact that a hierarchical and a parallel ESN share the same spectrum in the linear regime. Indeed, for a hierarchical ESN, whose connectivity matrix of the linearised dynamics is given by:

$$W = \begin{bmatrix} \mathbf{W}^{(11)} & 0 \\ \mathbf{W}^{(21)} & \mathbf{W}^{(22)} \end{bmatrix}, \tag{12}$$

it is easy to demonstrate that every eigenvalue of $\mathbf{W}^{(11)}$ and $\mathbf{W}^{(22)}$ is also an eigenvalue of W, irrespective of $\mathbf{W}^{(21)}$, not unlike what happens for a parallel ESN (where $\mathbf{W}^{(21)} = \mathbf{0}$, and hence the demonstration follows immediately). Nonetheless, as we will see in the next sections, the hierarchical ESN has better performance on different tasks compared to the other structures considered, including the parallel ESN.

It is interesting to note, in this respect, that the success of the hierarchical ESN is generally achieved when the leakage term of the first reservoir is higher than the leakage term of the second (or, in other words, when the first network has much shorter timescales). Such observation opens the way to an alternative route to understand the functioning of the hierarchical structure, as the first reservoir expanding the dimensionality of the input and then feeding the enriched signal into the second network. Indeed, in the following, we will show how, in a crude approximation and under the above condition of a wide separation of timescales, the first ESN extracts information on the short term behavior of the input signal, notably its derivatives, and the second ESN integrates such information over longer times.

We begin with the (continuous time) linearized dynamics of a Hierarchical ESN given by

$$\dot{\boldsymbol{x}}^{(1)}(t) = -\mathbf{M}^{(1)}\mathbf{x}^{(1)}(t) + \mathbf{W}_{\text{in}}^{(1)}\mathbf{s}(t), \tag{13}$$

$$\dot{\boldsymbol{x}}^{(2)}(t) = -\mathbf{M}^{(2)}\mathbf{x}^{(2)}(t) + \mathbf{W}^{(21)}\mathbf{x}^{(1)}(t), \tag{14}$$

where, for simplicity, we have reabsorbed the $\rho^{(kl)}$ and $\gamma^{(k)}$ factors into the definitions of $\mathbf{W}^{(kl)}$ and $\mathbf{W}_{\text{in}}^{(k)}$ respectively, and the new constants can be derived with reference to **Eqs 1,2**; for example:

$$\mathbf{M}^{(k)} = \frac{\alpha^{(k)}}{\delta t}\left[\mathbf{I} - f'(0)\ \mathbf{W}^{(kk)}\right]. \tag{15}$$

The neuron activity can be projected on to the left eigenvector of each of the $\mathbf{M}^{(i)}$ matrices. As such we define the eigenvector matrices, $\mathbf{V}^{(i)}$, where each row is a left eigenvector and so satisfies the equation $\mathbf{V}^{(i)}\mathbf{M}^{(i)} = \mathbf{\Lambda}^{(i)}V^{(i)}$. $\mathbf{\Lambda}^{(1)}$ and $\mathbf{\Lambda}^{(22)}$ are the diagonal matrices of the eigenvalues of the two $\mathbf{M}$ matrices. Using these we can define $\mathbf{y}^{(k)} \equiv \mathbf{V}^{(k)}\mathbf{x}^{(\mathbf{k})}$, and so the dynamical equations can be expressed as

$$\dot{\boldsymbol{y}}^{(1)}(t) = -\boldsymbol{\Lambda}^{(1)}\mathbf{y}^{(1)}(t) + \tilde{\mathbf{W}}_{\text{in}}^{(1)}\mathbf{s}(t), \tag{16}$$

$$\dot{\boldsymbol{y}}^{(2)}(t) = -\boldsymbol{\Lambda}^{(2)}\mathbf{y}^{(2)}(t) + \tilde{\boldsymbol{W}}^{(21)}\mathbf{y}^{(1)}(t), \tag{17}$$

where $\tilde{\boldsymbol{W}}_{\text{in}}^{(1)} = \mathbf{V}^{(1)}\mathbf{W}_{\text{in}}^{(1)}$ and $\tilde{\boldsymbol{W}}^{(21)} = \mathbf{V}^{(2)}\mathbf{W}^{(21)} \ \ (\mathbf{V}^{(1)})^{-1}$ are the input and connection matrices expanded in this basis. Taking the Fourier transform on both sides of **Eq. 16**, such that $FT[\mathbf{y}^{(1)}(t)] = \tilde{\boldsymbol{y}}^{(1)}(\omega)$ and $FT[\dot{\boldsymbol{y}}^{(1)}(t)] = -i\omega\tilde{\boldsymbol{y}}^{(1)}(\omega)$, where i is the imaginary unit. The transform $\tilde{\boldsymbol{y}}^{(2)}(\omega)$ of $\mathbf{y}^{(2)}(t)$ can now be expressed as a function of the transform of the signal $\tilde{\mathbf{s}}(\omega)$ giving

$$\left(\boldsymbol{\Lambda}^{(1)} - i\omega\mathbf{I}\right)\tilde{\boldsymbol{y}}^{(1)}(\omega) = \tilde{\boldsymbol{W}}_{\text{in}}^{(1)}\tilde{\mathbf{s}}(\omega) \tag{18}$$

where $\mathbf{I}$ is the identity matrix of the same size as $\boldsymbol{\Lambda}^{(1)}$. If the second ESN's timescale are much longer than that of the first one (i.e., $\boldsymbol{\Lambda}^{(1)} \gg \boldsymbol{\Lambda}^{(2)}$), then we can expand the inverse of the $\tilde{\boldsymbol{y}}^{(1)}$ coefficient on the LHS of **Eq. 18** when $\boldsymbol{\Lambda}^{(1)} \to \infty$ as

$$\left(\boldsymbol{\Lambda}^{(1)} - i\omega\mathbf{I}\right)^{-1} = \left(\boldsymbol{\Lambda}^{(1)}\right)^{-1}\left(1 - i\omega\left(\boldsymbol{\Lambda}^{(1)}\right)^{-1}\right)^{-1}, \tag{19}$$

$$\approx \left(\boldsymbol{\Lambda}^{(1)}\right)^{-1}\sum_{n=0}^{\infty}\left(i\omega\left(\boldsymbol{\Lambda}^{(1)}\right)^{-1}\right)^{n}, \tag{20}$$

By applying this approximation to **Eq. 18**, and by defining the diagonal matrix of characteristic times $\mathbf{T}^{(1)} \equiv -\,(\boldsymbol{\Lambda}^{(1)})^{-1}$, the relation between the activity of reservoir one and the input in Fourier space is given by

$$\tilde{\boldsymbol{y}}^{(1)}(\omega) = -\mathbf{T}^{(1)}\sum_{n=0}^{\infty}\left(-i\omega\mathbf{T}^{(1)}\right)^{n}\tilde{\mathbf{W}}_{\text{in}}^{(1)}\tilde{\mathbf{s}}(\omega). \tag{21}$$

The coefficients of this series are equivalent to taking successive time derivatives in Fourier space, such that $(-i\omega)^{n}\tilde{\mathbf{s}} = d^{(n)}\tilde{\mathbf{s}}/dt^{(n)}$. So by taking the inverse Fourier transform we find the following differential equation for $\boldsymbol{y}^{(1)}$

$$\mathbf{y}^{(1)}(t) = -\mathbf{T}^{(1)}\sum_{n=0}^{\infty}\left(\mathbf{T}^{(1)}\right)^{n}\tilde{\mathbf{W}}_{\text{in}}^{(1)}\frac{d^{(n)}\mathbf{s}(t)}{dt^{(n)}}, \tag{22}$$

which can be inserted into **Eq. 17** to give

$$\dot{\boldsymbol{y}}^{(2)} = \boldsymbol{\Lambda}^{(2)}\mathbf{y}^{(2)} - \tilde{\boldsymbol{W}}^{(21)}\boldsymbol{T}^{(1)}\left[\tilde{\boldsymbol{W}}_{\text{in}}^{(1)}\mathbf{s}(t) + \sum_{n=1}^{\infty}\left(\mathbf{T}^{(1)}\right)^{n}\tilde{\boldsymbol{W}}_{\text{in}}^{(1)}\frac{d^{(n)}\mathbf{s}(t)}{dt^{(n)}}\right]. \tag{23}$$

Thus the second ESN integrates the signal with a linear combination of its derivatives. In other words, the first reservoir expands the dimensionality of the signal to include information regarding the signal's derivatives (or, equivalently in discretized time, the previous values assumed by the signal). In this respect, **Eq. 23** is key to understanding how the hierarchical connectivity between the two reservoirs enhances the representational capabilities of the system. The finite-difference approximation of the time derivatives appearing in **Eq. 23** implies that a combination of past values of the signal appears, going back in time as much as the retained derivative order dictates.

2.2 Online Learning of Hyper-Parameter

Selecting the hyper-parameters of such systems can be challenging. Such selection process can be informed by the knowledge of the natural timescales of the task/signal at hand. Alternatively one can resort to a learning method to optimise the parameters directly. The inherent limitation of these methods is the same as learning the network weights with BPTT: the whole history of network activations is required at once. One way to by-pass this issue is to approximate the error signal by considering only past and same-time contributions, as suggested by Bellec *et al.* [4] in their framework known as *e-prop* (see also [36]), and derive from this approximation an online learning rule for the ESN hyper-parameters. Following their approach, we end up with a novel learning rule for the leakage terms of connected ESNs that is similar to the rule proposed by Jaeger *et al.* [32] but extended to two hierarchical reservoirs. The main learning rule is given by:

$$\frac{dE}{d\alpha^{(i)}}(t) = \sum_{k=1}^{N_{\text{ESN}}}\frac{\partial E}{\partial\mathbf{x}^{(k)}(t)}\mathbf{e}^{(ki)}(t), \tag{24}$$

where $\mathbf{e}^{(ki)}(t) = d\mathbf{x}^{(k)}(t)/d\alpha^{(i)}$ is known as the eligibility trace which tracks the gradient of neuron activities in the reservoir number k with respect to the ith leakage rate. Given the closed form for the hierarchical ESNs in **Eqs 10,11** these terms can be readily calculated. For our N_{ESN} sub-reservoirs in the hierarchical structure there will be N_{ESN}^{2} eligibility traces to track how each sub-reservoir depends on the other leakage rates. In the hierarchical case of a fixed feed-forward structure some of these traces will be zero, and the number of non-zero eligibility traces would be $N(N+1)/2$. Since the update of the neuron's activity depends on its previous values, so do the eligibility traces; therefore, they can be calculated recursively through

$$\begin{aligned}\mathbf{e}^{(ki)}(t+\delta t) = {} & \left(1-\alpha^{(k)}\right)\mathbf{e}^{(ki)}(t) + \delta_{ki}\left(f\left(\mathbf{h}^{(k)}(t)\right) - \mathbf{x}^{(k)}(t)\right) \\ & + \alpha^{(k)}f'\left(\mathbf{h}^{(k)}(t)\right)\sum_{l\neq k}\rho^{(kl)}\mathbf{W}^{(kl)}\mathbf{e}^{(li)}(t),\end{aligned} \tag{25}$$

where $\delta_{ki} = 1$ if $k = i$ and 0 otherwise, i. e the Kronecker delta. The update of **Eq. 25** for each k-i pair needs to follow the order of dependencies given by the structure of connected reservoirs considered. The eligibility trace is an approximation that only includes same-time contributions to the gradient but has the advantage that is can be easily computed online. A complete description of our method is given in the **Supplementary Material**. For an example where the mean squared error function $E(t) = \frac{1}{2}[\tilde{y}(t) - y(t)]^{2}$ is used in a regression task and a structure composed by two reservoirs, the updating equations on the leakage terms are

$$\begin{aligned}\alpha^{(1)} &\leftarrow \alpha^{(1)} - \eta_{\alpha}[\tilde{y}(t) - y(t)]\mathbf{W}_{\text{out}}\begin{pmatrix}\mathbf{e}^{(11)}(t)\\ \mathbf{e}^{(12)}(t)\end{pmatrix}\\ \alpha^{(2)} &\leftarrow \alpha^{(2)} - \eta_{\alpha}[\tilde{y}(t) - y(t)]\mathbf{W}_{\text{out}}\begin{pmatrix}\mathbf{e}^{(21)}(t)\\ \mathbf{e}^{(22)}(t)\end{pmatrix}\end{aligned} \tag{26}$$

where η_α is the learning rate on the leakage terms and $(\mathbf{e}^{(k1)}(t), \mathbf{e}^{(k2)}(t))$ $(k = 1, 2$ in this case with two reservoirs) is a vector composed by the juxtaposition of the eligibility traces, which can be computed through **Eq. 25**. Of course, the gradient can be combined with existing gradient learning techniques, among which we adopt the Adam optimiser, described in the **Supplementary Material**. In all online learning simulations, training is accomplished through minibatches with updates at each time step. Training is stopped after convergence. When learning α's and the output weights simultaneously, the learning rates corresponding to these hyper-parameters need to be carefully set, since the weights need to adapt quickly to the changing dynamic of the network, but a fast convergence of $\mathbf{W}_{\text{out}}$ can trap the optimisation process around sub-optimal values of the leakage terms. For a reservoir with trained and converged output weights, a further variation of α's, even in the right direction, could correspond to an undesirable increase in the error function. We found that this problem of local minimum can be avoided by applying a high momentum in the optimisation process of α and randomly re-initialising the output weights when the α's are close to convergence. The random re-initialisation functions to keep the output weights from being too close to convergence. Thus, we defined the convergence of the algorithm for α's as when the α's do not change considerably after re-initialisation. When this happens, it is possible to turn off the learning on the leakage terms and to optimise the read-out only. More details about online training can be found in the discussions related to each task.

3 RESULTS

The following sections are dedicated to the study of the role of timescales and the particular choices of α and ρ in various tasks, with attention on networks composed by a single ESN, 2 unconnected ESNs and 2 hierarchical ESNs. The number of trainable parameters in each task for the different models will be preserved by using the same total number of neurons in each model. The results analyzed will be consequently interpreted through the analysis of timescales of the linearised systems.

3.1 NARMA10

A common test signal for reservoir computing systems is the non-linear auto-regressive moving average sequence computed with a 10 step time delay (NARMA10) [37, 38]. Here we adopt a discrete time formalism where $n = t/\delta t$ and the internal state of the reservoir is denoted as $\mathbf{x}_n = \mathbf{x}(n\delta t)$. The input, s_n, is a uniformly distributed random number in the range $[0, 0.5]$ and the output time-series is computed using

$$y_n = y_{n-1}\left(a + b\sum_{k=1}^{D} y_{n-k}\right) + cs_{n-1}s_{n-D} + d, \tag{27}$$

where $D = 10$ is the memory length, $a = 0.3$, $b = 0.05$, $c = 1.5$, and $d = 0.1$. The task for the network is to predict the NARMA10 output y_n given the input s_n. We have adapted this to also generate a NARMA5 task where $D = 5$ but the other parameters are unchanged. This provides an almost identical task but with different timescales for comparison.

The task of reconstructing the output of the NARMA10 sequence can be challenging for a reservoir as it requires both a memory (and average) over the previous 10 steps and fast variation with the current input values to produce the desired output. A typical input and output signal is shown in **Figure 3A** and the corresponding auto-correlation function of the input and output in B. Since the input is a random sequence it does not exhibit any interesting features but for the output the auto-correlation shows a clear peak at a delay of 9 δt in accordance with the governing equation. For a reservoir to handle this task well it is necessary to include not only highly non-linear dynamics on a short timescale but also slower dynamics to handle the memory aspect of the task.

This regression task is solved by training a set of linear output weights to minimise the mean squared error (MSE) of the network output and true output. The predicted output is computed using linear output weights on the concatenated network activity $(\mathbf{x}_n = (\mathbf{x}_n^{(1)}, \mathbf{x}_n^{(2)})^T)$, such that

$$\tilde{y}_n = \mathbf{x}_n^T \mathbf{W}_{\text{out}} \tag{28}$$

where $\mathbf{W}_{\text{out}}$ is the weight vector of length N+1 when an additional bias unit is included. The MSE is minimised by using the ridge regression method [39] such that the weights are computed using

$$\mathbf{W}_{\text{out}} = \left(\mathbf{x}^T\mathbf{x} - \lambda\mathbf{I}\right)^{-1}\mathbf{x}^T\mathbf{y} \tag{29}$$

where $\mathbf{x}$ is a matrix formed from the activation of the internal states with a shape of number of samples by number of neurons, $\mathbf{y}$ is the desired output vector, λ is the regularisation parameter that is selected using a validation data set and $\mathbf{I}$ the identity matrix. To analyze the performance of the ESNs on the NARMA10 task we use the normalised root mean squared error as

$$\text{NRMSE} = \sqrt{\frac{1}{N_s}\frac{\sum_{n}^{N_s}\left(\tilde{y}_n - y_n\right)^2}{\text{Var}(\mathbf{y})}}, \tag{30}$$

where $\tilde{y}_n$ is the predicted output of the network and y_n is the true output as defined by **Eq. 27**.

To test the effectiveness of including multiple time-scales in ESNs, we simulate first a single ESN with $N = 100$ neurons and vary both α and ρ to alter the time-scale distribution. Secondly, we simulate a hierarchical ESN split into 2 reservoirs each with $N = 50$ neurons, where we vary $\alpha^{(1)}$ and $\alpha^{(2)}$ with $\rho^{(1)} = \rho^{(2)} = 0.95$. The input factor was set as $\gamma^{(1)} = 0.2$ and $\gamma^{(2)} = 0$ for the connected hierarchical ESN but when they are unconnected the input is fed into both, such that $\gamma^{(1)} = \gamma^{(2)} = 0.2$. In all cases the NRMSE is computed on an unseen test set and averaged over 20 initialisations of the ESN with a running median convolution is applied to the error surfaces to reduce outliers. In parallel to this we have also applied the online training method for the α hyper-parameters. The hyper-parameters used for the gradient descent learning are summarised in **Table 1**.

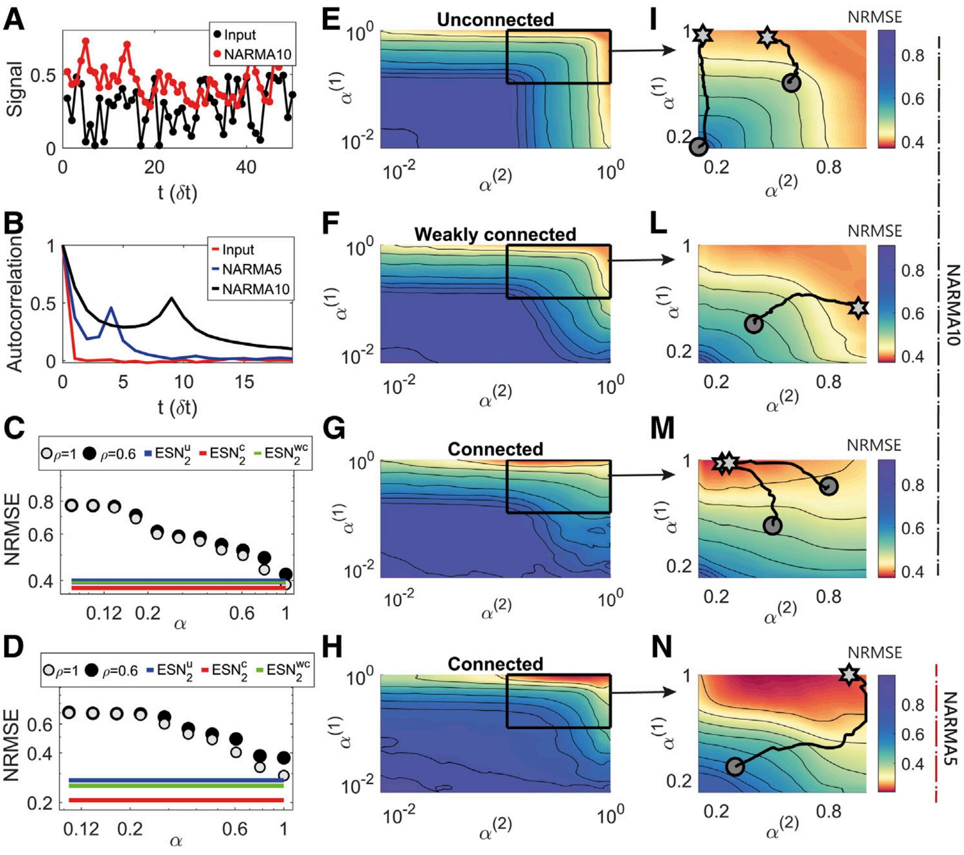


FIGURE 3 | Performance of single or hierarchical ESNs on the NARMA10 and NARMA5 task. **(A)**: Example input signal (black) and desired output (red) for the NARMA10 task. **(B)**: The auto-correlation function of the (black) input (red) NARMA10 and (blue) NARMA5 desired output signals, showing a second peak at about 9 delay steps for the NARMA10 and 4 for the NARMA5. **(C)**: The NRMSE for a single ESN for with $\rho = 1.0$ and 0.63 over a range of α. The NRMSE is lower for $\rho \approx 1$ and $\alpha = 1$. The solid lines show the minimum NRMSE for the unconnected (blue line) and connected (red line); for the unconnected case the minimum NRMSE is similar to the single ESN while the connected case has a smaller NRMSE by about 10%. **(D)**: Average NRMSE of a single ESN for various α compared to the hierarchical ESNs for the NARMA5 task. **(E–K):** The average NRMSE surface using a hierarchical ESN computed for varying the leakage rates $\alpha^{(k)}$ of both the reservoir components for **(E)** and **(I)** (no coupling, $\rho^{(21)} = 0$), **(F)** and **(J)** (weak coupling, $\rho^{(21)} = 0.1$), and **(G)** and **(K)** (strong coupling, $\rho^{(21)} = 1$). Panels (**I-L**) show a close up in region for the range $\alpha^{(k)} = [0.1, 1]$ to highlight the changing behaviors. The lines on these panels show the trajectory of the $\alpha^{(k)}$ values trained directly using the online method. For each case of the coupling the online learning trends toward the approximate error minimum. **(H)** shows the NRMSE surface for the NARMA5 task using a strongly connected hierarchical ESN, with **(N)** again showing a zoom of the $\alpha = [0.1, 1]$ region. The region of best performance is with $\alpha^{(2)} \approx 0.5$ which matches the shorter timescale demonstrated in the auto-correlation in **(B)**.

Figures 3E–G and I–M show the NRMSE depending on $\alpha^{(1)}$ and $\alpha^{(2)}$ for 3 variations of the hierarchical ESN connection strength on the NARMA10 task. In the unconnected case ($\rho^{(21)} = 0$, panels E and I), we find that the NRMSE drops by increasing both leakage rates but the minimum is when one of the leakage rates is ≈ 0.5. This is in agreement with the online learning method for the α's in shown in I but the error minimum is shallow and prone to noise in the signal or ESN structure. For the weakly connected hierarchical ESN ($\rho^{(21)} = 0.1$, panels F and J) we find again that when the sub-reservoirs have different timescales the NRMSE is reduced. In comparison to the unconnected case the error surface is asymmetric with a minimum at approximately $\alpha^{(1)} = 1.0$ and $\alpha^{(2)} \approx 0.5$. As the strength of the connection is increased ($\rho^{(21)} = 1.0$, Panel G and K), the minimum error moves to a lower leakage rate in the second reservoir ($\alpha^{(2)} \approx 0.2$) which reflects a better separation of the timescale distributions. This is a gradual effect with respect to the connection strength since stronger connection allows for a relative increase of the expanded input from the first reservoir compared to the base input signal. Since the input feeds into reservoir 1, a high α provides a transformation on the input over short time-scales, expanding the dimensionality of the signal, offering a representation that preserves much of the dynamic of the driving input and that is fed to the second reservoir. Then, since the latter does not have a direct connection to the input it performs a longer timescale transformation of the internal states of reservoir 1. In this way the reservoirs naturally act on different parts of the task, i.e., reservoir one provides a fast non-linear transformation of the input while reservoir 2 follows the slower varying 10-step average of the signal, and thus returning a lower NRMSE. As a side note, we can demonstrate the validity of the theoretical analysis in **Section 2.1** by replacing the first reservoir by **Eq. 23** on the NARMA task (see

TABLE 1 | Table of the hyper-parameters adopted in the online learning process. η is the learning rate in each case, while β_1, β_2 and ϵ are parameters for the Adam optimiser (further details are given in the **Supplementary Material**).

Learning hyper-parameters

	NARMA/Telegraph	**psMNIST**
Network size N	100	1200
Minibatch size N_{batch}	10	50
	Learning $\mathbf{W}_{\text{out}}$	
η_W	10^{-3}	10^{-3} $[10^{-4}]^{a}$
β_1	0.9	0.9
β_2	0.999	0.999
ϵ	10^{-8}	10^{-8}
	Learning α	
η_α	5×10^{-6}	10^{-3}
β_1	0.99	0.999
β_2	0.999	0.999
ϵ	10^{-8}	10^{-8}

[a] *symbol indicates that the learning rate 10^{-3} is for the case with 4 hidden states, while the learning rate $[10^{-4}]$ is for the case with 28 hidden states. This decrease of η is due to the increase in the dimensionality of the representation for the latter case in comparison to the situation where the read-out is composed by four concatenated values of activity. Furthermore, such learning rates are 10 times higher than the case in which only the read-out is trained (only in the psMNIST task). Thus, the high learning rate adopted has the purpose to introduce noise in the learning process and to avoid local minima in the complex case where α and $\mathbf{W}_{out}$ are optimised simultaneously.*

Section 3 Supplementary Material), resulting in a similar landscape as in **Figure 3G** and a similar optimal value for $\alpha^{(2)}$.

Figure 3C shows the relative performance of the single ESN to the minimum values for the unconnected (ESN_2^u) and connected (ESN_2^c) hierarchical reservoirs. The single ESN shows the similar decrease in NRMSE with increasing α and reaches a similar minimum NRMSE as the unconnected case. In comparison with the connected cases the multiple timescales provides a more optimised result. If we consider the analysis of the timescales discussed in the previous section the choice of these hyper-parameters becomes more evident. With $\alpha = 1$ the timescale distribution of the network is sharply peaked close to the minimum timescale of one discrete step while when $\alpha = 0.1$ this peak is broader and the peak of the distribution is closer to the second peak present in the auto-correlation function shown in Panel **B**. We note that while the most likely timescale is $\tau_{\text{peak}} \approx 6$ for $\alpha = 0.1, \rho = 0.95$ which is lower than the natural timescale of the problem, the increased width of the distribution increases the number of timescales at $\tau = 10$ dramatically which maybe why a lower α is not necessary.

To further investigate the effect of the inherent timescale of the task on the timescales we performed a similar analysis on the NARMA5 task. **Figures 3H,L** show the NRMSE surface for the strongly connected case. The minimum error occurs at $\alpha^{(1)} \approx 1.0$ (similar to the NARMA10 results in G and K) but $\alpha^{(2)} \approx 0.5$ (as opposed to ≈ 0.2 for NARMA10). This is due to the shorter timescales required by the NARMA5 task and the peak timescale for these values is much closer to the peak in the auto-correlation shown in **B**. Panel D shows the performance of the single ESN where again the optimal leakage rate is $\alpha = 1$ and similar to the unconnected cases but the NRMSE is higher than the connected cases.

In this theoretical task where the desired output is designed *a priori*, the memory required and the consequent range of timescales necessary to solve the task are known. Consequently, considering the mathematical analysis in **Section 2.1**, and that for hierarchical ESNs the timescales of the first ESN should be faster than those of the second **Figure 3**), the best-performing values of the leakage terms can be set *a priori* without the computationally expensive grid search reported in **Figures 3E–L**. However, it can be difficult to guess the leakage terms in the more complex cases where the autocorrelation structure of the signal is only partially informative of the timescales required.

This problem can be solved using the online learning approach defined through **Eq. 24**. In this case, learning is accomplished through minibatches and the error function can be written explicitly as

$$E(t) = \frac{1}{2\text{N}_{\text{batch}}} \sum_{m=1}^{\text{N}_{\text{batch}}} \left[\tilde{y}(t,m) - y(t,m)\right]^2 \tag{31}$$

where N_{batch} is the minibatch size and m is its corresponding index. A minibatch is introduced artificially by dividing the input sequence into N_{batch} signals or by generating different NARMA signals. Of course, the two methods lead to equivalent results if we assure that the N_{batch} sequences are temporally long enough. A learning rate $\eta_\alpha/\eta_W \approx 10^{-2} - 10^{-3}$ was adopted. The optimiser used for this purpose is Adam, with the suggested value of $\beta_1 = 0.9$ adopted for the output weights and a higher first momentum $\beta_1 = 0.99$ adopted for the leakage terms. Instead, we set $\beta_2 = 0.999$ of the second momentum for both types of parameters (See **Section 2.2** for a description of the updating rules). Panels I–L show a zoomed in region of the error surface with the lines showing the online training trajectory of the α hyper-parameters. In each case the trajectory is moving toward the minimum NRMSE of the α phase space.

3.2 A Volatile Environment

We now turn to study the reservoir performance on a task of a telegraph process in a simulated noisy environment. The telegraph process $s^{(1)}(t)$ has two states that we will call *up* 1) and *down* (0), where the probability of going from a *down* state to an *up* state $p(s = 1|s = 0)$ (or the opposite $p(s = 0|s = 1)$) is fixed for any time step. The environment is also characterised by a telegraph process $s^{(2)}(t)$, but the transition probability is much lower and controls the transition probability of the first signal. To simplify the notation in the following we denote the probability of the signal i transitioning from state a to state b as $p(s^{(i)}(t) = a|s^{(i)}(t - \delta t) = b) = p_{ab}^{(i)}(t)$. The signal taken under consideration is then composed by a fast telegraph process with probabilities $p_{01}^{(1)}(t)$ and $p_{10}^{(1)}(t)$, whose values are interchanged by following the dynamic of a slower telegraph process $s^{(2)}(t)$. Every time the slower environment signal changes its state, the probabilities of the first signal are changed, i. e., $p_{01}^{(1)}(t) \leftrightarrow p_{10}^{(1)}(t)$. The resulting signal is then characterised by

$$p_{10}^{(1)}(t) = \begin{cases} p_1, & \text{if } s^{(2)}(t) = 0 \\ p_2, & \text{if } s^{(2)}(t) = 1 \end{cases} \tag{32}$$

$$p_{01}^{(1)}(t) = \begin{cases} p_2, & \text{if } s^{(2)}(t) = 0 \\ p_1, & \text{if } s^{(2)}(t) = 1 \end{cases} \tag{33}$$

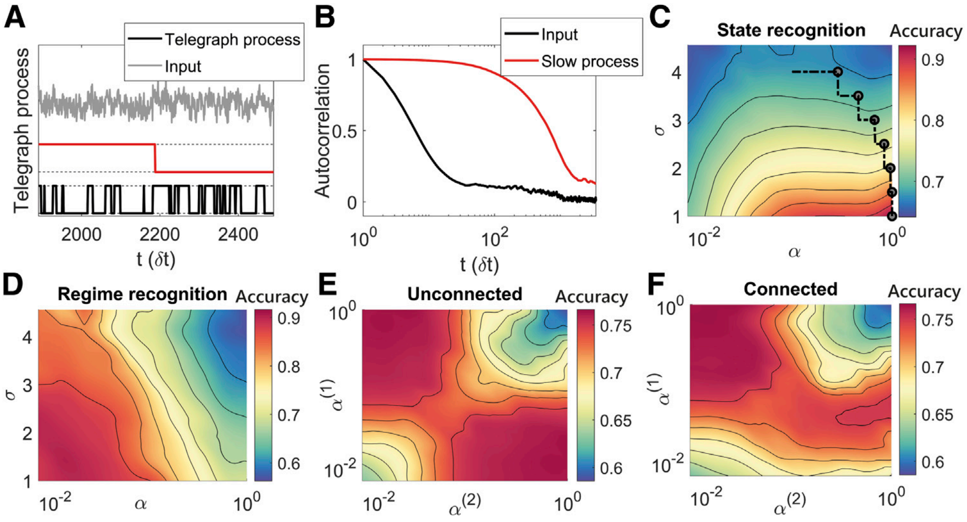


FIGURE 4 | The best structure and parameters of the model depend on the specific environment considered, that is different values of the additive noise in the input signal, and on the specific desired output. **(A)**: Example of input signal and of its generative processes, which have a faster and a slower dynamic respectively. When the slower process (red line) is *up* (*down*), the other signal is in a regime where the average time in the zero (one) state is greater than the average time spent in the other state. The input signal (gray line) corresponds to the faster process (black line) with additional white noise. **(B)**: Auto-correlation structure of the two generative processes. **(C)**: The accuracy surface for a single ESN on the state recognition sub-task for varying level of noise (σ) and leakage rate of the network showing that for increasing levels of noise a lower leakage rate is needed to determine the state. The line shows the trajectory of α using the online learning method when the strength of the noise is changed. **(D)**: The accuracy for a single ESN on the regime recognition sub-task for varying noise and leakage rate. In this case the low leakage rate is preferred for all values of noise. **(E)**: Accuracy surface for the state recognition sub-task for an unconnected hierarchical ESN showing how either of the leakage rates must be low while the other is high. **(F)**: Accuracy surface for the regime recognition sub-task for a hierarchical ESN showing the first reservoir must have a high leakage rate and the second a low leakage rate.

The transition probabilities of the second signal are fixed and symmetric such that

$$p_{01}^{(2)}(t) = p_{10}^{(2)}(t) = p_3, \tag{34}$$

The probabilities p_1, p_2 and p_3 are fixed parameters of the signal that define the process. Given that the second signal controls the probabilities of the first telegraph process, we say that it defines the regime of the input, while we refer to the *up* and *down* values of the first process simply as states. Thus, the reconstruction of $s^{(1)}(t)$ from the input will be called state reconstruction, while reconstruction of $s^{(2)}(t)$ will be called regime reconstruction. These reconstructions can be considered separately or as a joint task requiring the system to be modeled on long and short timescales simultaneously. Due to the probability transition caused by $s^{(2)}(t)$, both states and regime will be equally present over a infinitely long signal. The values adopted for the simulation are $p_1 = 0.05$, $p_2 = 0.1$ and $p_3 = 0.0005$.

The input signal corresponds to $s^{(1)}(t) + \sigma\mathcal{N}(0,1)$, that is the faster telegraph process with additional white noise. The input signal constructed is a metaphor of a highly stochastic environment with two states and two possible regimes that define the probability of switching between the two states. The reservoir will be asked to understand in which state ($s^{(1)}(t) = 1$ or 0) and/or regime ($s^{(2)}(t) = 1$ or 0) it is for each time t, measuring the understanding of the model to estimate the state of the input signal. The input signal and telegraph processes is shown in **Figure 4A**, while B shows the corresponding auto-correlation structure of the processes. The auto-correlation shows that the input has a temporal structure of around 10 δt while the slow ‘environment’ process has a structure close to 1000 δt. This corresponds directly to the timescales defined by the probabilities of the signals.

Panels C and D of **Figure 4** show the performance of a single ESN when it is tasked to reconstruct the processes $s^{(1)}(t)$ (state recognition) and $s^{(2)}(t)$ (regime recognition) respectively. In this simulation, learning is always accomplished online and the error function is the same as **Eq. 31**. First, panel C demonstrates how the leakage term, α, must be tuned to the level of noise of the environment, and how lower values of α are desirable for noisier signals, in order to solve the state recognition problem. Indeed, the need to smooth the fluctuations of the input signal increases with σ, while for low values of noise the network should simply mimic the driving input. Second, panel D shows how the desirable values of α must be lower in the case where the network is asked to reproduce the slower dynamic of $s^{(2)}(t)$ independently of having to output the fast signal, in order to solve the regime recognition problem. This result exemplifies how the timescales of the network must be tuned depending on the desired output. It demonstrates that, even in this relatively simple environment, it is crucial to adopt multiple timescales in the network to obtain results that are robust with respect to a variation of the additional white noise σ.

Finally, panels E and F of **Figure 4** show the accuracy of two unconnected (E) and connected (F) reservoirs when the network

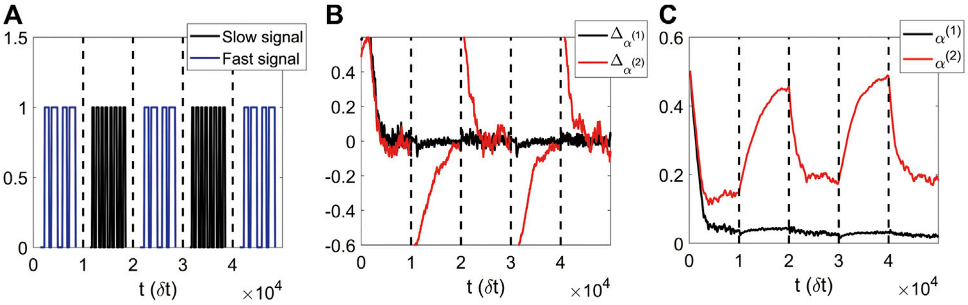


FIGURE 5 | The online training of the leakage terms can adapt to the changing environment, that is the signal probabilities are increased or decreased periodically. **(A)**: Scheme of the change of the values of probabilities, where high probabilities of switching are referred to as fast phase of the telegraph process, while low probabilities as slow phase. **(B)**: Running average of the gradients of $\alpha^{(1)}$ and $\alpha^{(2)}$ as time varies. **(C)**: Online adaptation of the leakage terms.

has to classify the state and the regime of the input signal at the same time. In this case, the desired output corresponds to a four dimensional signal that encodes all the possible combinations of states and regimes; for instance, when the signal is in the state one and in the regime one, we would require the first dimension of the output to be equal to one and all other dimensions to be equal to zero, and so on. The best performance occurs when one leakage term is high and the other one is low and in the range of significant delays of the auto-correlation function. This corresponds to one network solving the regime recognition and the other network solving the state recognition. For the unconnected reservoirs, it does not matter which reservoir has high vs. low leakage terms, reflected by the symmetry of **Figure 4E**, while for the connected reservoirs, the best performance occurs when the first reservoir has the high leakage term and the second the low leakage terms, see **Figure 4F**, similar to **Figure 3**. Both two-reservoir networks can achieve accuracy 0.75, but the single ESN can not solve the task efficiently, since it cannot simultaneously satisfy the need for high and low αs, reporting a maximum performance of about 0.64.

The path reported in panel C of **Figure 4** and all panels in **Figure 5** show the application of the online training algorithm in this environment. The values of the hyper-parameters adopted in the optimisation process through the Adam optimiser are the same as in **Section 3.1**, where we used a slower learning rate and a higher first momentum on the leakage terms in comparison to the values adopted for the output weights. The line of panel C (**Figure 4**) shows the online adaptation of α for a simulation where the external noise increases from one to four with six constant steps of 0.5 equally spaced across the computational time of the simulation. The result shows how the timescales of the network decrease for each increase in σ, depicted with a circle along the black line. The path of online adaptation reports a decrease of the α value for noisier external signals. This result occurs because as the signal becomes noisier (σ rises), it becomes more important to dampen signal fluctuations. This result also shows that the online algorithm can adapt in environments with varying signal to noise ratio. **Figure 5** shows the online training of $\alpha^{(1)}$ and $\alpha^{(2)}$ for an environment composed by a faster and a slower composition of telegraph processes. This specific simulation is characterised by the alternation of two signals defined by **Eqs 32**, **33** and **34**, each with different values of p_1 and p_2. In particular, while $p_1 = 0.5$ and $p_2 = 0.1$ for the 'fast' phase of the external signal, $p_1 = 0.1$ and $p_2 = 0.05$ for the "slow" phase. In contrast, the slower timescale of the task defined by $p_3 = 0.0005$ remains invariant across the experiment. Panel C shows the adaptation of the leakage terms for this task in the case of a hierarchical structure of ESNs. While $\alpha^{(2)}$ adapts to the change of p_1 and p_2 following the transition between the two phases of the external signals, the relatively constant value of $\alpha^{(1)}$ indicates how the first network sets its timescales to follow the slower dynamic of the signal, characterised by the constant value of p_3. Thus, the composed network exploits the two reservoirs separately, and the first (second) reservoir is used to represent the information necessary to recognise the regime (state) of the external signal.

3.3 Permuted Sequential MNIST

The Permuted Sequential MNIST (psMNIST) task is considered a standard benchmark for studying the ability of recurrent neural networks to understand long temporal dependencies. The task is based on the MNIST dataset, which is composed of 60,000 handwritten digits digitised to 28 × 28 pixel images. In the standard MNIST protocol every pixel is presented at the same temporal step so a machine has all the information of the image available at once and needs to classify the input into one out of ten classes. In contrast, in the psMNIST task, the model receives each pixel sequentially once at a time, so that the length of the one dimensional input sequence is 784. Thus, the machine has to rely on its intrinsic temporal dynamic and consequent memory ability to classify the image correctly. Furthermore, each image in the dataset is transformed through a random permutation of its pixels in order to include temporal dependencies over a wide range of input timescales and to destroy the original images' structure. Of course, the same permutation is applied on the entire dataset. The performance of ESNs on the MNIST dataset, where each columns of pixels in a image is fed to the network sequentially (each image corresponds to a 28

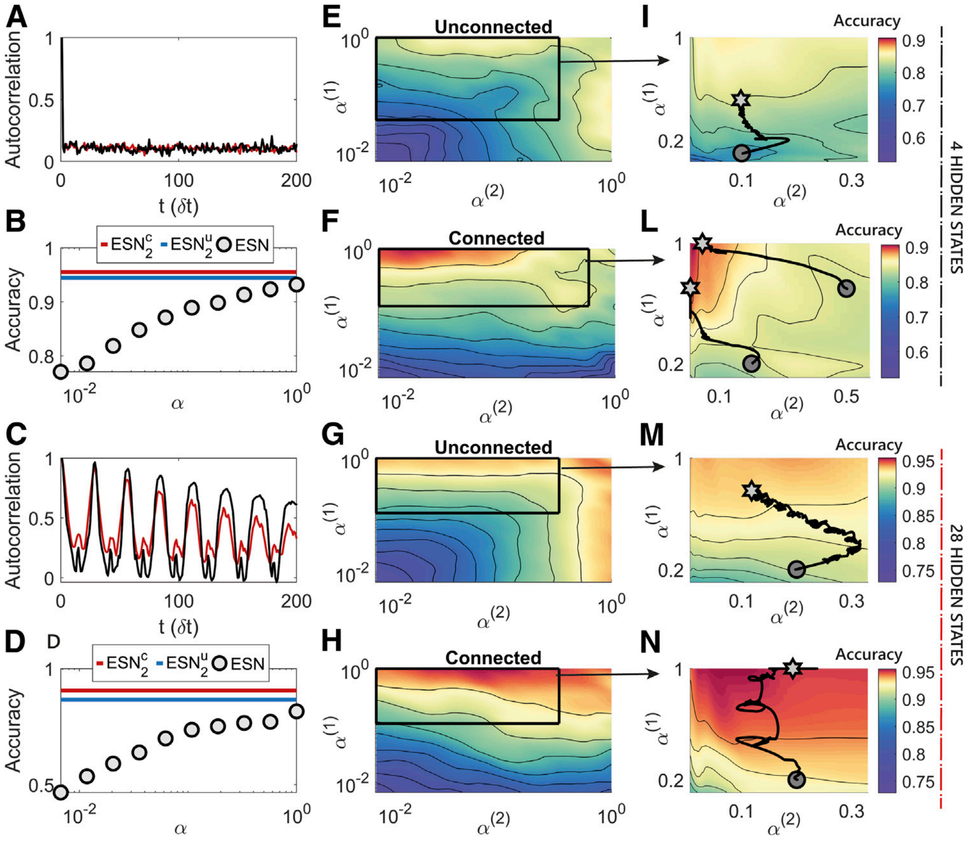


FIGURE 6 | The additional non linearity added by the hierarchical reservoir structure is responsible for a relevant modification and increase of the performance surface. **(A,C)**: Auto-correlation structure of the MNIST dataset for two examples of digits, where each pixel is presented one after the other **(C)**, and auto-correlation structure of the data after the random permutation **(A)**. The oscillatory trend in **C** reflects the form of the written digits, when this is seen one pixel after the other. The auto-correlation function of the permuted data is low, but not negligible, for all the temporal steps, showing the necessity to have a wide repertoire of timescales in the interval corresponding to the image size. **(B,D)**: Accuracy of a single ESN for various α values compared the maximum accuracy of the hierarchical ESNs with 4 hidden states **(B)** or 28 hidden states **(D)**. **(E–F)**: case with low sampling frequency of the ESNs which corresponds to a higher demand of internal memory in the reservoir. While the best region of accuracy for the unconnected reservoirs is characterised by intermediate values of the leakage factors, the hierarchically connected network structure reports the best performance when the second network has slower dynamics. **(G–H)**: The utilisation of a high sampling frequency alleviates the need for long term memory, and the reservoirs prefer the regions with fast timescales. In both cases analyzed, the additional complexity of the hierarchical model leads to a considerable boost in performance. **(I–N)**: Paths (black line, starting from the circle and ending in the star) that describe the online changes of the leakage terms achieved through the online training algorithm in a zoomed region of the performance surface of $\alpha^{(1)}$ and $\alpha^{(2)}$. The paths are smoothed through a running average.

dimensional signal of length 28 time steps), has been analyzed in [40] and in [41]. In [40] the original dataset was preprocessed through reshaping and rotating the original image to enhance the network's ability to understand high level features of the data. In this case, the original dataset is used. In [41], the addition of thresholds and the introduction of sparse representation in the read-out of the reservoir was used to improve the performance of the network in the online learning of the standard MNIST task through reservoir computing. This section is focused on the analysis of the performance of ESNs on the psMNIST task and on their dependence on the range of timescales available in the network, i.e. the values of α and ρ chosen. In contrast to the previous sections where ESNs are trained through ridge regression, we have applied an online gradient descent optimisation method. The cost function chosen to be minimised is the cross entropy loss

$$E = -\frac{1}{N_{\mathrm{batch}}}\sum_{m=1}^{N_{\mathrm{batch}}}\sum_{j=1}^{N_{\mathrm{class}}}\left[y_j(m)\log\left(\tilde{y}_j(m)\right)\right. \left. + \left(1-y_j(m)\right)\log\left(1-\tilde{y}_j(m)\right)\right], \tag{35}$$

where m is the minibatch index, N_{batch} corresponds to the minibatch size and N_{class} is the number of classes. For this task the desired output, y_j, is a one-hot encoded vector of the correct classification while the desired output is a sigmoid function of the readout of the reservoir nodes. Furthermore, instead of reading out the activity of the reservoir from the final temporal step of each sequence only, we have expanded the reservoir representation by using previous temporal activities of the network. In practice, given the sequence of activities $\mathbf{x}(0), \mathbf{x}(\delta t), \ldots, \mathbf{x}(\delta t T)$ $(T = 784)$ that defines the whole temporal dynamic of the network subjected to an example input sequence, we trained the network by reading out from the expanded vector $\mathbf{X} = [\mathbf{x}(M\delta t), \mathbf{x}(2M\delta t), \ldots, \mathbf{x}(T\delta t)]$, where M defines the "time frame" used to sample the activities of the evolution of the system across time.

$$\tilde{y} = \mathrm{sigm}\left(\sum_{n=1}^{T/M} \mathbf{W}_{\mathrm{out}}^{(n)} \mathbf{x}(nM\delta t)\right) \tag{36}$$

where sigm stands for sigmoid activation function. We then repeat the simulation for two different time frames of sampling for each different model, that is a single ESN and a pair of unconnected or connected ESNs, as in the previous sections.

The two values of M used are 28 and 196, corresponding to a sampling of 28 and 4 previous representations of the network respectively. Of course, a higher value of M corresponds to a more challenging task, since the network has to exploit more its dynamic to infer temporal dependencies. We note, however, that none of the representation expansions used can guarantee a good understanding of the temporal dependencies of the task, or in other words, can guarantee that the system would be able to discover higher order features of the image, considering that these features depend on events that could be distant in time.

In **Figure 6** we again analyze the performance of two connected or unconnected ESNs varying $\alpha^{(1)}$ and $\alpha^{(2)}$ for both $M = 28$ and 196. In contrast to the previous sections, we now use gradient descent learning on the output weights instead of ridge regression and increase the total number of neurons in each model to $N = 1200$ due to the complexity of the task. The Adam optimiser is used; its parameters, for both the output weights and α learning, are in **Table 1**. As previously, we have trained the output weights over a range of fixed αs and report the performance on an unseen test data set. In parallel to this we have trained both the output weights and α values which, as shown by the lines on the contour plots, converge toward the minimum computed using the fixed α's.

As in the other simulations, we found that the values of ρ corresponding to the best performance was approximately one, which maximises the range of timescales and the memory available in the network. **Figures 6E,F** shows the case with $M = 28$, while **Figures 6G,H** reports the accuracy for the simulation with $M = 196$ where E and G are unconnected and F and H connected reservoirs. The accuracy surface demonstrates how, in the case of the unconnected ESNs with a fast sampling rate in panel G, the best performance is achieved when at least one of the two values of α is close to one. The result is due to the fast changing dynamic of the temporal sequence that is introduced through the random permutation of the pixels. On the contrary, in the case of the unconnected ESNs with a slow sampling rate in panel E the best accuracy is in a range of intermediate timescales since both partitions must respond to both fast and slow timescales.

This relatively simple behavior of the dependence of the accuracy on the setting of the hyper-parameters changes in the cases of two connected ESNs, whose additional complexity corresponds to a considerable increase in the performance. **Figure 6H** reports how the network prefers a regime with a fast timescale in the first reservoir and a intermediate timescale in the second, which acts as an additional non-linear temporal filter of the input provided by the first network. The need of memory of events distant in time is emphasised in 6F, where the best performing network is composed by reservoirs with fast and slow dynamics respectively. The performance boost from the panels E–G to the ones F-H has only two possible explanations: first, the timescales of the second network are increased naturally thanks to the input from the first reservoir; second, the connections between the two reservoirs provide an additional non-linear filter of the input that can be exploited to discover higher level features of the signal. Thus, we can conclude once again that achieving high performance in applying reservoir models requires 1) additional non-linearity introduced through the interconnections among the reservoirs and 2) an appropriate choice of timescales, reflecting the task requirements in terms of external signal and memory.

Panels I, L, M and N show the application of the online training of αs for the various cases analyzed. In the psMNIST task we found that the major difficulties in the application of an iterative learning rule on the leakage terms are: the possibility to get trapped in local minima, whose abundance can be caused by the intrinsic complexity of the task, the intrinsic noise of the dataset, the randomness of the reservoir and of the applied permutation; the high computational time of a simulation that exploits an iterative optimisation process on αs arising from a practical constraint in the implementation. Indeed, while the activities of the reservoir can be computed once across the whole dataset and then saved in the case of untrained values of αs, the activities of the nodes need to be computed every time the leakage terms change in the online learning paradigm. However, we found that using a higher learning rate η_W on the output weights, compared to the value adopted in the paradigm where the leakage terms are not optimised (as in Panels E, F, G and H), can introduce beneficial noise in the learning process and help to avoid local minima. Furthermore, a higher value of the learning rate on the output weights corresponds to an increased learning rate on the thresholds, as shown from **Supplementary Equation S7** and from the dependence of the updating equations on $\mathbf{W}_{\mathrm{out}}$. As in the previous simulations of **Sections 3.1 and 3.2**, the output weights are randomly reinitialised after the convergence of αs, helping the algorithm to avoid an undesirable quick convergence of weights. The online process is then ended when the leakage terms remain approximately constant even after the re-initialisation. Following this computational recipe, it possible to avoid the difficulties found and train the leakage terms efficiently.

Finally, we note how the best accuracy of 0.96 reached throughout all the experiments on the psMNIST is comparable to the results obtained by recurrent neural networks trained with BPTT, whose performance on this task are analyzed in [42] and can vary from 0.88 to 0.95. In comparison to recurrent structures trained through BPTT, a network with two interacting ESNs provide a cheap and easily trainable model. However, this comparison is limited by the necessity of recurrent neural networks to carry the information from the beginning to the end of the sequence, and to use the last temporal state only or to adopt attention mechanisms.

4 CONCLUSION

In summary, ESNs are a powerful tool for processing temporal data, since they contain internal memory and time-scales that can be adjusted via network hyper-parameters. Here we have highlighted that multiple internal time-scales can be accessed by adopting a split network architecture with differing hyper-parameters. We have explored the performance of this architecture on three different tasks: NARMA10, a benchmark composed by a fast-slow telegraph process and PSMNIST. In each task, since multiple timescales are present the hierarchical ESN performs better than a single ESN when the two reservoirs have separate slow and fast timescales. We have demonstrated how choosing the optimal leakage terms of a reservoir can be aided by the theoretical analysis in the linear regime of the network, and by studying the auto-correlation structure of the input and/or desired output and the memory required to solve the task. The theoretical analysis developed needs to be considered as a guide for the tuning of the reservoir hyper-parameters, and in some specific applications it could be insufficient because of the lack of information about the nature of the task. In this regard, we showed how to apply a data-driven online learning method to optimise the timescales of reservoirs with different structures, demonstrating its ability to find the operating regimes of the network that correspond to high performance and to the best, task-dependent, choice of timescales. The necessity of adopting different leakage factors is emphasised in the case of interactive reservoirs, whose additional complexity leads to better performance in all cases analyzed. Indeed, the second reservoir, which acts as an additional non linear filter with respect to the input, is the perfect candidate to discover higher temporal features of the signal, and it consequently prefers to adopt longer timescales in comparison to the first reservoir, which has instead the role of efficiently representing the input. We believe such hierarchical architectures will be useful for addressing complex temporal problems and there is also potential to further optimise the connectivity between the component reservoirs by appropriate adaptation of the online learning framework presented here.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: http://yann.lecun.com/exdb/mnist/. The code for this study is available from: https://github.com/LucaManneschi/EchoStatesNetwork.

AUTHOR CONTRIBUTIONS

All the authors contributed to the paper conceptually and to the writing of the article. EV proposed the original concept of the paper. LM, ME, GG, PD, EV contributed to the design of the case studies and to the analysis of the results. LM and ME designed and performed the simulations.

FUNDING

EV and AL acknowledge the support from a Google Deepmind Award. EV and ME were funded by the Engineering and Physical Sciences Research Council (Grant No. EP/S009647/1). EV was also supported by EPSRC (Grant Nos. EP/S030964/1 and EP/P006094/1). PD and GG were supported by the European Union Horizon 2020 Research and Innovation program under the FET Flagship Human Brain Project (SGA2 Grant agreement No. 785907 and SGA3 Grant agreement No. 945539). AL was supported by the European Research Council (639489) and the Biotechnology and Biological Sciences Research Council (BB/S016031/1).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fams.2020.616658/full#supplementary-material.

REFERENCES

1. Ludik J, Prins W, Meert K, Catfolis T. A comparative study of fully and partially recurrent networks. *Proc Int Conf Neural Netw* (1997) 1(ICNN'97):292–7. doi:10.1109/ICNN.1997.611681
2. Rumelhart DE, Hinton GE, Williams RJ. Technical Report. Learning internal representations by error propagation. California Univ San Diego La Jolla Inst for Cognitive Science (1985)
3. Werbos PJ. Backpropagation through time: what it does and how to do it. *Proc IEEE* (1990) 78:1550–60. doi:10.1109/5.58337
4. Bellec G, Scherr F, Subramoney A, Hajek E, Salaj D, Legenstein R, et al. A solution to the learning dilemma for recurrent networks of spiking neurons. *Nat Commun* (2020) 11:3625. doi:10.1038/s41467-020-17236-y
5. Marr D. A theory of cerebellar cortex. *J Physiol* (1969) 202:437–70. doi:10.1113/jphysiol.1969.sp008820
6. Farris SM. Are mushroom bodies cerebellum-like structures? *Arthropod Struct Dev* (2011) 40:368–79. doi:10.1016/j.asd.2011.02.004
7. Laurent G. Olfactory network dynamics and the coding of multidimensional signals. *Nat Rev Neurosci* (2002) 3:884–95. doi:10.1038/nrn964
8. Warren R, Sawtell NB. A comparative approach to cerebellar function: insights from electrosensory systems. *Curr Opin Neurobiol* (2016) 41:31–7. doi:10.1016/j.conb.2016.07.012
9. Takemura SY, Aso Y, Hige T, Wong A, Lu Z, Xu CS, et al. A connectome of a learning and memory center in the adult Drosophila brain. *eLife* (2017) 6:5643. doi:10.7554/eLife.26975
10. Zheng Z, Lauritzen JS, Perlman E, Robinson CG, Nichols M, Milkie D, et al. A complete electron microscopy volume of the brain of adult *Drosophila melanogaster*. *Cell* (2018) 174:730–43. doi:10.1016/j.cell.2018.06.019
11. Liu Q, Yang X, Tian J, Gao Z, Wang M, Li Y, et al. Gap junction networks in mushroom bodies participate in visual learning and memory in Drosophila. *eLife* (2016) 5:e13238. doi:10.7554/eLife.13238
12. Shang Y, Claridge-Chang A, Sjulson L, Pypaert M, Miesenböck G. Excitatory local circuits and their implications for olfactory processing in the fly antennal lobe. *Cell* (2007) 128:601–12. doi:10.1016/j.cell.2006.12.034
13. Olsen SR, Wilson RI. Lateral presynaptic inhibition mediates gain control in an olfactory circuit. *Nature* (2008) 452:956–60. doi:10.1038/nature06864
14. Yeshurun Y, Nguyen M, Hasson U. Amplification of local changes along the timescale processing hierarchy. *Proc Natl Acad Sci U S A* (2017) 114:9475–80. doi:10.1073/pnas.1701652114

15. Jaeger H. *The "echo state" approach to analysing and training recurrent neural networks-with an erratum note*. Bonn, Germany: German National Research Center for Information Technology GMD Technical Report (2001) 148:13.
16. Deng Z, Zhang Y. Collective behavior of a small-world recurrent neural system with scale-free distribution. *IEEE Trans Neural Netw* (2007) 18:1364–75. doi:10.1109/tnn.2007.894082
17. Rodan A, Tino P. Minimum complexity echo state network. *IEEE Trans Neural Netw* (2010) 22:131–44. doi:10.1109/TNN.2010.2089641
18. Bacciu D, Bongiorno A. Concentric esn: assessing the effect of modularity in cycle reservoirs. In: 2018 International Joint Conference on Neural Networks (IJCNN); 2018 July 8–13; Rio, Brazil . (IEEE) (2018), 1–8.
19. Farkaš I, Bosák R, Gergeľ P. Computational analysis of memory capacity in echo state networks. *Neural Netw* (2016) 83:109–20. doi:10.1016/j.neunet.2016.07.012
20. Marzen S. Difference between memory and prediction in linear recurrent networks. *Phys Rev E* (2017) 96:032308. doi:10.1103/PhysRevE.96.032308
21. Livi L, Bianchi FM, Alippi C. Determination of the edge of criticality in echo state networks through Fisher information maximization. *IEEE Trans Neural Netw Learn Syst* (2018) 29:706–17. doi:10.1109/TNNLS.2016.2644268
22. Xue Y, Yang L, Haykin S. Decoupled echo state networks with lateral inhibition. *Neural Netw* (2007) 20:365–76. doi:10.1016/j.neunet.2007.04.014
23. Jaeger H. Technical Report. Discovering multiscale dynamical features with hierarchical echo state networks. Bremen, Germany: Jacobs University Bremen (2007)
24. Gallicchio C, Micheli A, Pedrelli L. Deep echo state networks for diagnosis of Parkinson's disease (2018a) arXiv preprint. Available from: https://arxiv.org/abs/1802.06708 (Accessed February 19, 2018).
25. Malik ZK, Hussain A, Wu QJ. Multilayered echo state machine: a novel architecture and algorithm. *IEEE Trans Cybernetics* (2016) 47:946–59. doi:10.1109/TCYB.2016.2533545
26. Gallicchio C, Micheli A. Echo state property of deep reservoir computing networks. *Cogn Comp* (2017) 9:337–50. 10.1007/s12559-017-9461-9
27. Gallicchio C, Micheli A, Pedrelli L. Design of deep echo state networks. *Neural Netw* (2018b) 108:33–47. doi:10.1016/j.neunet.2018.08.002
28. Ma Q, Shen L, Cottrell GW. Deepr-esn: a deep projection-encoding echo-state network. *Inf Sci* (2020) 511:152–71. doi:10.1016/j.ins.2019.09.049
29. Rodriguez N, Izquierdo E, Ahn YY. Optimal modularity and memory capacity of neural reservoirs. *Netw Neurosci* (2019) 3:551–66. doi:10.1162/netn_a_00082
30. Szita I, Gyenes V, Lőrincz A. Reinforcement learning with echo state networks. In: International Conference on Artificial Neural Networks; December 5–8, 2006; Athens, Greece (Springer) (2006) 830–9.
31. Hermans M, Schrauwen B. Recurrent kernel machines: computing with infinite echo state networks. *Neural Comput* (2012) 24:104–33. doi:10.1162/NECO_a_00200
32. Jaeger H, Lukoševičius M, Popovici D, Siewert U. Optimization and applications of echo state networks with leaky-integrator neurons. *Neural Netw* (2007) 20:335–52. doi:10.1016/j.neunet.2007.04.016
33. Girko VL. Circular law. *Theory Probab Appl* (1985) 29:694–706. doi:10.1038/s41586-019-1763-5
34. Gallicchio C, Micheli A, Pedrelli L. Deep reservoir computing: a critical experimental analysis. *Neurocomputing* (2017) 268:87–99. doi:10.1016/j.neucom.2016.12.089
35. Sun X, Li T, Li Q, Huang Y, Li Y. Deep belief echo-state network and its application to time series prediction. *Knowl Based Syst* (2017) 130:17–29. doi:10.1016/j.knosys.2017.05.022
36. Manneschi L, Vasilaki E. An alternative to backpropagation through time. *Nat Mach Intell* (2020) 2:155–6. doi:10.1002/mp.14033
37. Atiya AF, Parlos AG. New results on recurrent network training: unifying the algorithms and accelerating convergence. *IEEE Trans Neural Networks* (2000) 11:697–709. doi:10.1109/72.846741
38. Goudarzi A, Banda P, Lakin MR, Teuscher C, Stefanovic D. A comparative study of reservoir computing for temporal signal processing (2014) arXiv preprint. Available from: https://arxiv.org/abs/1401.2224 (Accessed January 10, 2014).
39. Lukoševičius M, Jaeger H. Reservoir computing approaches to recurrent neural network training. *Computer Science Review* (2009) 3:127–49. doi:10.1016/j.cosrev.2009.03.005
40. Schaetti N, Salomon M, Couturier R. Echo state networks-based reservoir computing for mnist handwritten digits recognition. In: 2016 IEEE Intl Conference on Computational Science and Engineering (CSE) and IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC) and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering (DCABES); 2016 August 24–26; Paris, France (IEEE) (2016) 484–91.
41. Manneschi L, Lin AC, Vasilaki E. Sparce: sparse reservoir computing (2019) arXiv preprint. Available at: https://arxiv.org/abs/1912.08124 (Accessed December 4, 2019).
42. Chandar S, Sankar C, Vorontsov E, Kahou SE, Bengio Y. Towards non-saturating recurrent units for modelling long-term dependencies. *Proc AAAI Conf Artif Intell* (2019) 33:3280–7. doi:10.1609/aaai.v33i01.33013280

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers
in Artificial Intelligence

ORIGINAL RESEARCH
published: 23 February 2021
doi: 10.3389/frai.2021.618372

Generalizable Machine Learning in Neuroscience Using Graph Neural Networks

Paul Y. Wang[1,2]*, *Sandalika Sapra*[1,3], *Vivek Kurien George*[1,4] and *Gabriel A. Silva*[1,4,5]*

[1]Center for Engineered Natural Intelligence, University of California San Diego, La Jolla, CA, United States, [2]Department of Physics, University of California San Diego, La Jolla, CA, United States, [3]Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, CA, United States, [4]Department of Bioengineering, University of California San Diego, La Jolla, CA, United States, [5]Department of Neurosciences, University of California San Diego, La Jolla, CA, United States

Although a number of studies have explored deep learning in neuroscience, the application of these algorithms to neural systems on a microscopic scale, i.e. parameters relevant to lower scales of organization, remains relatively novel. Motivated by advances in whole-brain imaging, we examined the performance of deep learning models on microscopic neural dynamics and resulting emergent behaviors using calcium imaging data from the nematode *C. elegans*. As one of the only species for which neuron-level dynamics can be recorded, *C. elegans* serves as the ideal organism for designing and testing models bridging recent advances in deep learning and established concepts in neuroscience. We show that neural networks perform remarkably well on both neuron-level dynamics prediction and behavioral state classification. In addition, we compared the performance of structure agnostic neural networks and graph neural networks to investigate if graph structure can be exploited as a favourable inductive bias. To perform this experiment, we designed a graph neural network which explicitly infers relations between neurons from neural activity and leverages the inferred graph structure during computations. In our experiments, we found that graph neural networks generally outperformed structure agnostic models and excel in generalization on unseen organisms, implying a potential path to generalizable machine learning in neuroscience.

Keywords: calcium imaging, graph neural network, deep learning, *C elegans*, motor action classification

OPEN ACCESS

Edited by:
Raina Robeva,
Randolph–Macon College,
United States

Reviewed by:
Arnaud Fadja Nguembang,
University of Ferrara, Italy
Jean-Roch,
Vlimant, California Institute of
Technology, United States

****Correspondence:***
Paul Y. Wang
pywang@ucsd.edu
Gabriel A. Silva
gsilva@ucsd.edu

Specialty section:
This article was submitted to
Machine Learning and Artificial
Intelligence,
a section of the journal
Frontiers in Artificial Intelligence

Received: *16 October 2020*
Accepted: *12 January 2021*
Published: *23 February 2021*

Citation:
Wang PY, Sapra S, George VK and Silva GA (2021) Generalizable Machine Learning in Neuroscience Using Graph Neural Networks.
Front. Artif. Intell. 4:618372.
doi: 10.3389/frai.2021.618372

1 INTRODUCTION

Constructing generalizable models in neuroscience poses a significant challenge because systems in neuroscience are typically complex in the sense that dynamical systems composed of numerous components collectively participate to produce emergent behaviors. Analyzing these systems can be difficult because they tend to be highly non-linear in how they interact, can exhibit chaotic behaviors and are high-dimensional by definition. As such, indistinguishable macroscopic states can arise from numerous unique combinations of microscopic parameters i.e., parameters relevant to lower scales of organization. Thus, bottom-up approaches to modeling neural systems often fail since a large number of microscopic configurations can lead to the same observables (Golowasch et al., (2002); Prinz et al., (2004)).

Because neural systems are highly degenerate and complex, their analysis is not amenable to many conventional algorithms. For example, observed correlations between individual neurons and behavioral states of an organism may not generalize to other organisms or even to repeated trials in the same individual (Frégnac (2017); Churchland et al., (2010); Goldman et al., (2001)). Hence, individual variability of neural dynamics remains poorly understood and a fundamental obstacle to model development as evaluation on unseen individuals often leads to subpar results. Nevertheless, neural systems exhibit universal behavior: organisms behave similarly. Motivated by the need for robust and generalizable analytical techniques, researchers recently applied tools from dynamical systems analysis to simple organisms in hopes of discovering a universal organizational principle underlying behavior. These studies, made possible by advances in whole-brain imaging, reveal that neural dynamics live on low-dimensional manifolds which map to behavioral states [Prevedel et al., (2014); Kato et al., (2015)]. This discovery implies that although microscopic neural dynamics differ between organisms, a macroscopic/global universal framework may enable generalizable algorithms in neuroscience. Nevertheless, the need for significant hand-engineered feature extraction in these studies underscores the potential of deep learning models for scalable analysis of neural dynamics.

In this work, we examine the performance and generalizability of deep learning models applied to the neural activity of *C. elegans* (round worm/nematode). In particular, *C. elegans* is a canonical species for investigating microscopic neural dynamics because it remains the only organism whose connectome (the mapping of all 302 neurons and their synaptic connections) is completely known and well studied [White et al., (1986); Bargmann and Marder (2013); Varshney et al., (2011); Cook et al., (2019)]. Furthermore, the transparent body of these worms allows for calcium imaging of whole brain neural activity which remains the only imaging technique capable of spatially resolving the dynamics of individual neurons (Wen and Kimura, 2020). Leveraging these characteristics and insight gained from previous studies, we developed deep learning models that bridge recent advances in neuroscience and deep learning. Specifically, we first demonstrate state-of-the-art performance for classifying motor action states-e.g., forward and reverse crawling-of *C. elegans* from calcium imaging data acquired in previous works. Next, we examine the generalization performance of our deep learning models on unseen worms both within the same study and in worms from a separate study published years later. We then show that graph neural networks exhibit a favourable inductive bias for analyzing both higher-order function and microscopic/neuron-level dynamics in *C. elegans*.

2 BACKGROUND

In this section we discuss recent advances in neuroscience and machine learning upon which we build our model and experiments.

2.1 Universality/Generalizability in Neuroscience

The motor action sequence of *C. elegans* is one of the only systems for which experiments on whole-brain microscopic neural activity may be performed and readily analyzed. As such, numerous efforts have focused on building models that can accurately capture the hierarchical nature of neural dynamics and resulting locomotive behaviors [Sarma et al., (2018); Gleeson et al., (2018)]. Taking advantage of this, Kato et al., (2015) investigated neural dynamics corresponding to a pirouette, a motor action sequence in which worms switch from forward to backward crawling, turn, and then continue forward crawling. Their analysis showed that most variations (~ 65%) in neural dynamics can be expressed by three components found through principal component analysis (PCA) and that neural dynamics in the resulting latent space trace cyclical trajectories on well-defined low dimensional manifolds corresponding to the motor action sequence (**Supplementary Figure S1**). By identifying individual neurons, an experimental feat, these authors further determined that these topological structures in latent space were universally found among all five worms imaged in their study.

Following Kato et al., (2015), the authors published several studies focusing on global organizational principles of C. Elegant behavior [Nichols et al., (2017); Kaplan et al., (2020); Skora et al., (2018)]. Building on two of these works, Brennan and Proekt (2019) found consistent differences between each individual's neural dynamics, precluding the use of established dimensional reduction techniques. For example, among 15 neurons uniquely identified among all 5 worms, only 3 neurons displayed statistically consistent behavior (**Figure 1D**). Examples of inconsistent behavior for unequivocally identified neurons (ALA and RIML) are shown in **Figure 1C** where the average of ALA's activity fails to resemble the behavior of any worm and where RIML's activity is consistent among all animals during dorsal turns, but inconsistent during reverse crawling. Resulting from these discrepancies, topological structures identified by performing PCA on each worm's neural activity were no longer observed when data from all worms was pooled together.

To address this issue, Brennan and Proekt (2019) introduced a new algorithm, Asymmetric Diffusion Map Modeling (ADMM), which maps the neural activity of any worm to an universal manifold (**Figure 2**). To achieve this, ADMM first performs time-delay embedding of neural activity into phase space. Next, a transition probability matrix is constructed by calculating distances between points in phase space using a Gaussian kernel centered on the subsequent timestep. Finally, this asymmetric diffusion map is used to construct a manifold representative of neural activity. Contrasting conventional dimensional reduction techniques, ADMM allowed quantitative modeling by mapping neural activity from the manifold, and enabled the prediction of motor action states up to 30s ahead. Despite its success, the algorithm heavily relies on hyperparameters, such as embedding parameters, which are difficult to justify and tune.

2.2 Graph Neural Networks

Graph Neural Networks (GNNs) are a class of neural networks that explicitly use graph structure during computations through message passing algorithms where features are passed along edges between nodes and then aggregated for each node [Scarselli et al.,

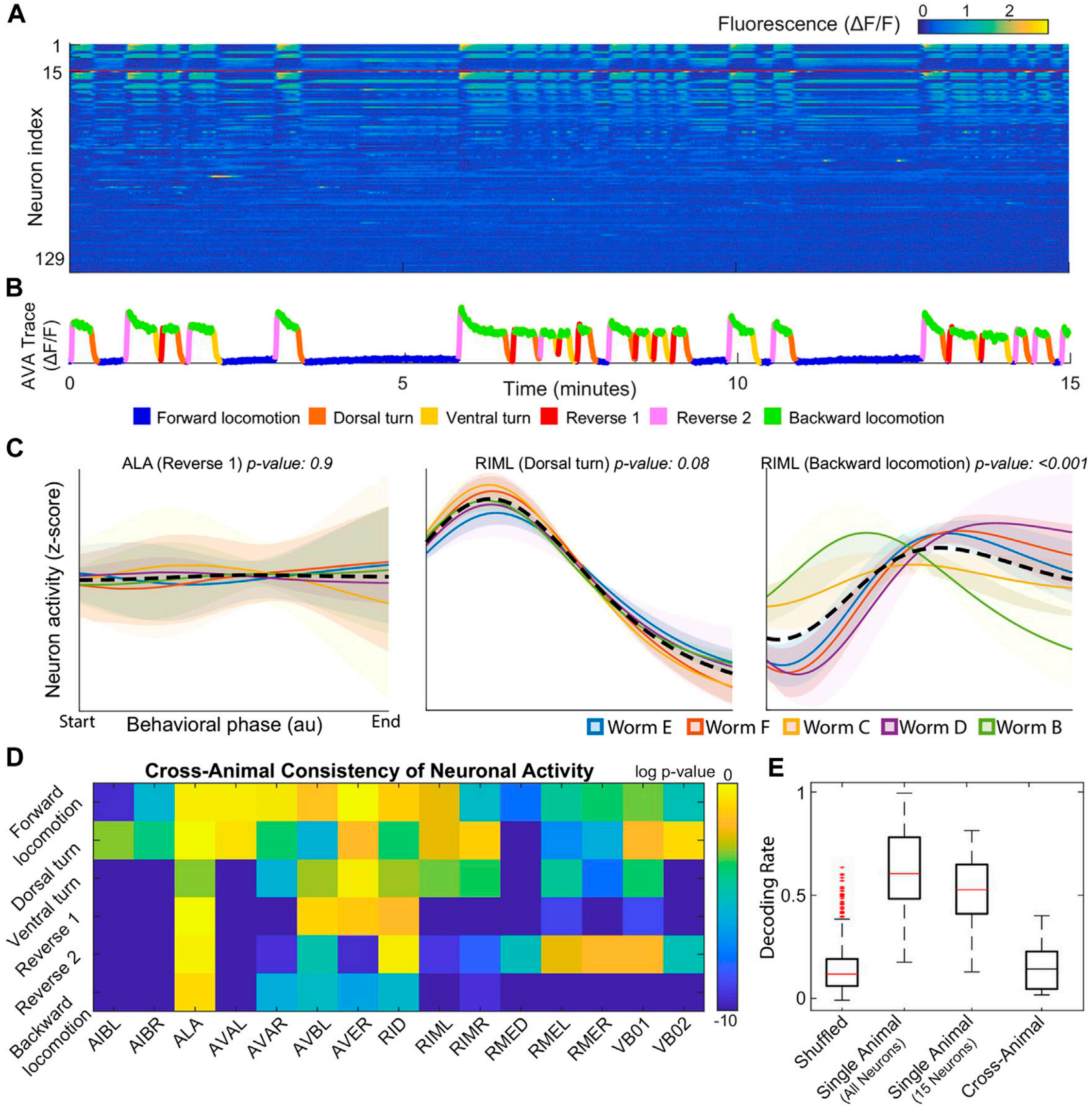


FIGURE 1 | (A) Calcium signals recorded in one animal for ∼ 15 min by Kato et al., (2015). Each row represents a single neuron. The top 15 rows (above the red line) correspond to neurons unambiguously identified in all animals (shared neurons) **(B)** Sample trace with corresponding behavioral state colored **(C)** Neural dynamics of two neurons for specific behavior states. Colored solid lines are the mean activity for each animal, and the black dashed line is the mean activity for all animals. Shaded colored regions show 95% confidence intervals **(D)** Probabilities that neural dynamics from different individuals were drawn from the same distribution **(E)** Attempt by Brennan and Proekt (2019) to decode onset of backwards locomotion using neural dynamics for each animal and averaged neural dynamics across other four animals. Reproduced with permission from Brennan and Proekt (2019).

(2009); Gilmer et al., (2017)]. These networks were inspired by the success of convolutional neural networks in the domain of two-dimensional image processing and failures when extending conventional convolutional networks to non-euclidean domains Battaglia et al., (2018). In essence, because graphs can have arbitrary structure, the inductive bias of convolutional neural networks [equivariance to translational transformations (Cohen and Welling, 2016)] often breaks down when applied to graphs. Addressing this issue, an early work on GNNs showed that one-hop message passing approximates spectral convolutions on graphs [Kipf and Welling (2016)]. Subsequent works have examined the representational power of GNNs in relation to the Weisfeiler-Lehman isomorphism test Xu et al., (2018) and limitations of GNNs when learning graph moments [Dehmamy et al., (2019)]. From an applied perspective, GNNs have been widely successful in a wide variety of domains including relational inference [Kipf et al., (2018); Löwe et al., (2020); Raposo et al., (2017)], node classification Kipf and Welling (2016) Hamilton et al., (2017), point cloud segmentation (Wang et al.,, 2019), and traffic forecasting Yu et al., (2018); Li et al., (2018).

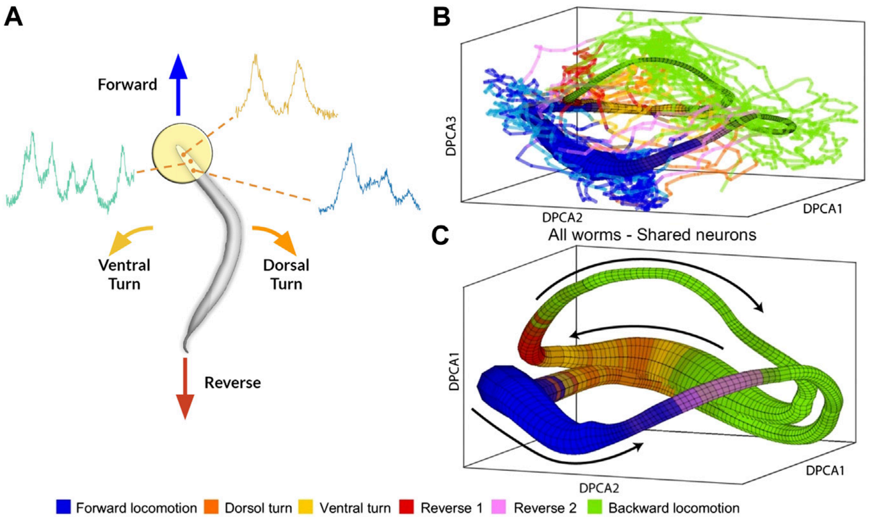

A
Forward
Ventral Turn
Dorsal Turn
Reverse
B
DPCA3
DPCA2
DPCA1
C
All worms - Shared neurons
DPCA1
DPCA2
DPCA1
🟦 Forward locomotion 🟧 Dorsol turn 🟨 Ventral turn 🟥 Reverse 1 🟪 Reverse 2 🟩 Backward locomotion

FIGURE 2 | (A) Rendering of calcium imaging experiment where activity of neurons in the head of the worm is recorded. Colored arrows show main motor action behavioral states **(B)** and **(C)** Resulting manifold from Brennan and Proekt (2019) **(B)** Manifold constructed from activity of four worms with colored lines indicating neural activity of fifth worm **(C)** Manifold constructed from neural activity of uniquely identified neurons ($n = 15$) shared among all five worms. Black arrows correspond to cyclical transition of motor action sequence and colors correspond to motor action states. Modified with permission from Brennan and Proekt (2019).

2.3 Relational Inference

Relational inference remains a longstanding challenge with early works in neuroscience seeking to quantify correlations between neurons Granger (1969). Modern approaches to relational inference employ graph neural networks as their explicit reliance on graph structure forms a relational inductive bias [Battaglia et al., (2016); Battaglia et al., (2018)]. In particular, our model is inspired by the Neural Relational Inference model (NRI) which uses a variational autoencoder for generating edges and a decoder for predicting trajectories of each object in a system [Kipf et al., (2018)]. By inferring edges, the NRI model explicitly captures interactions between objects and leverages the resulting graph as an inductive bias for various machine learning tasks. This model was successfully used to predict the trajectories of coupled Kuramoto oscillators, particles connected by springs, the pick and roll play from basketball, and motion capture visualizations. Subsequently, the authors developed Amortized Causal Discovery, a framework based on the NRI model which infers causal relations from time-dependent data Löwe et al., (2020).

2.4 Deep Learning in Neuroscience

With the success of convolutional neural networks, researchers successfully applied deep learning to numerous domains in neuroscience Glaser et al., (2019) including MRI imaging Lundervold and Lundervold (2019) and connectomes Brown and Hamarneh (2016) where algorithms can predict disorders such as autism Brown et al., (2018). Further leveraging the explicit graph structure of neural systems, several studies have successfully applied GNNs on various tasks such as annotating cognitive state Zhang and Bellec, 2019, and several frameworks based on graph neural networks have been proposed for analyzing fMRI data [Li and Duncan (2020); Kim and Ye (2020)].

Similarly, brain-computer interfaces (BCI) are a well-studied field related to our work as they focus on decoding macroscopic variables from measurements of neural activity. These studies generally involve fMRI or EEG data, which characterize neural activity on a population level, to varying amounts of success [Bashivan et al., (2015); Kwak et al., (2017); Mensch et al., (2017); Makin et al., (2020)]. Regardless, a challenge for the field is developing generalizable algorithms to individuals unseen during training Zhang et al., (2019).

3 MODEL

In this section, we first present the general framework of our behavioral state classification and trajectory prediction models. Next, we detail the implementation of our neural network models.

3.1 Framework

We define the set of trajectories (calcium imaging traces) for each worm as $\boldsymbol{X}_\alpha = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n, \ldots, \boldsymbol{x}_N\}_\alpha$ where α denotes the label of the individual, n the name of the neuron, N the total number of neurons, and $\boldsymbol{x}_n$ the feature vector of the neuron. In our case, $\boldsymbol{x}_n \in \mathbb{R}^{T\times 2}$ corresponds to time-dependent normalized calcium traces and their derivatives for each neuron where T is the total number of timesteps. Likewise, $x_{n,t} \in \mathbb{R}^2$ corresponds to the features of neuron n at timestep t. Finally, the behavioral states of an individual are encoded as $\boldsymbol{a}_\alpha = (a_1, \ldots, a_t, \ldots, a_T)_\alpha$ where a behavioral state a is assigned for each timestep t.

Separate models were developed for each task: behavioral state classification and trajectory prediction. In both cases, data from a worm α is structured as a temporal graph $\mathcal{G}_\alpha =$

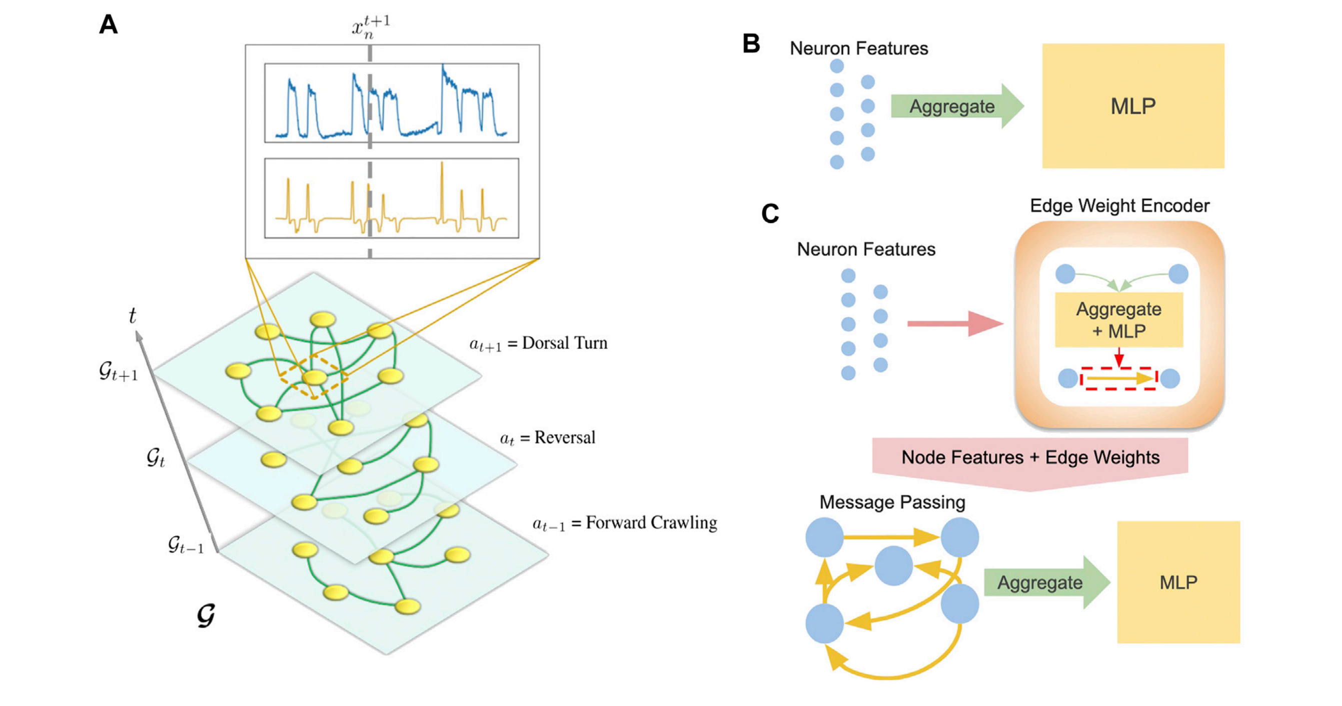


FIGURE 3 | (A) Visualization of temporal graph. Inset shows $\boldsymbol{x}_n$ plotted against t where the top is the calcium trace, and the bottom is its derivative. The dashed line intercepts the feature vectors at $t_\prime = t + 1$ and denotes x_n^{t+1} **(B)** and **(C)** are simplified visualizations of the MLP and GNN models respectively.

$(\mathcal{G}_1, \ldots, \mathcal{G}_t, \ldots, \mathcal{G}_T)_\alpha$ (**Figure 3A**) where each timestep is represented by a static graph whose nodes correspond to neurons. Following the notation above for worm α, the trajectories of each neuron's calcium traces are encoded as node features $\boldsymbol{x}_n$, and the behavioral state of the worm is interpreted as a graph feature a_t. For behavioral state classification, our model consists of the following:

$$\boldsymbol{H}_{\alpha,t} = f(\boldsymbol{X}_{\alpha,t}), \tag{1}$$

$$\boldsymbol{p}_{\alpha,t} = softmax(\boldsymbol{H}_{\alpha,t}), \tag{2}$$

$$\widehat{a}_{\alpha,t} = argmax(\boldsymbol{p}_{\alpha,t}). \tag{3}$$

where $\boldsymbol{X}_{\alpha,t}$ corresponds to node feature vectors for worm α at timestep t, f is an universal approximator/neural network model (described in the next section), $\boldsymbol{H}_{\alpha,t} \in \mathbb{R}^k$ corresponds to embedded features, $\boldsymbol{p}_{\alpha,t}$ is the probability that the worm is in one of k motor states (**Figure 4D**), and $\widehat{a}_{\alpha,t}$ is the most probable/predicted state.

For trajectory prediction, we developed a Markovian model for inferring trajectories of a consecutive timestep:

$$\boldsymbol{H}_{\alpha,t} = f(\boldsymbol{X}_{\alpha,t}), \tag{4}$$

$$\widehat{\boldsymbol{X}}_{\alpha,t+1} = \boldsymbol{X}_{\alpha,t} + \boldsymbol{H}_{\alpha,t}, \tag{5}$$

where f is the same as before, $\boldsymbol{H}_{\alpha,t}$ is the predicted change of the trajectory and can be interpreted as $\Delta\widehat{\boldsymbol{X}}_{\alpha,t}$, and $\widehat{\boldsymbol{X}}_{\alpha,t+1}$ is the predicted value of the subsequent timestep. When predicting multiple timesteps, the predicted value of the previous timestep is substituted for $\boldsymbol{X}_{\alpha,t}$. We also experimented with non-Markovian models (RNNs) for which a hidden state is included for each timestep.

The structure of our framework allows us to substitute various models for f. While we include results from several neural networks, we focus on two representative models: a multi-layer perceptron (MLP) agnostic to graph structure (**Figure 3B**) and a graph neural network (GNN) which explicitly computes on an inferred graph (**Figure 3C**).

3.2 Neural Network Models *f*: MLP and GNN

Our MLP model aggregates (sums or concatenates) the features of a graph and feeds the aggregated features into a 2-layer MLP neural network:

$$\boldsymbol{H}_{out} = g_{graph:mlp}(\text{aggregation}(x_1, \ldots, x_n, \ldots, x_N)), \tag{6}$$

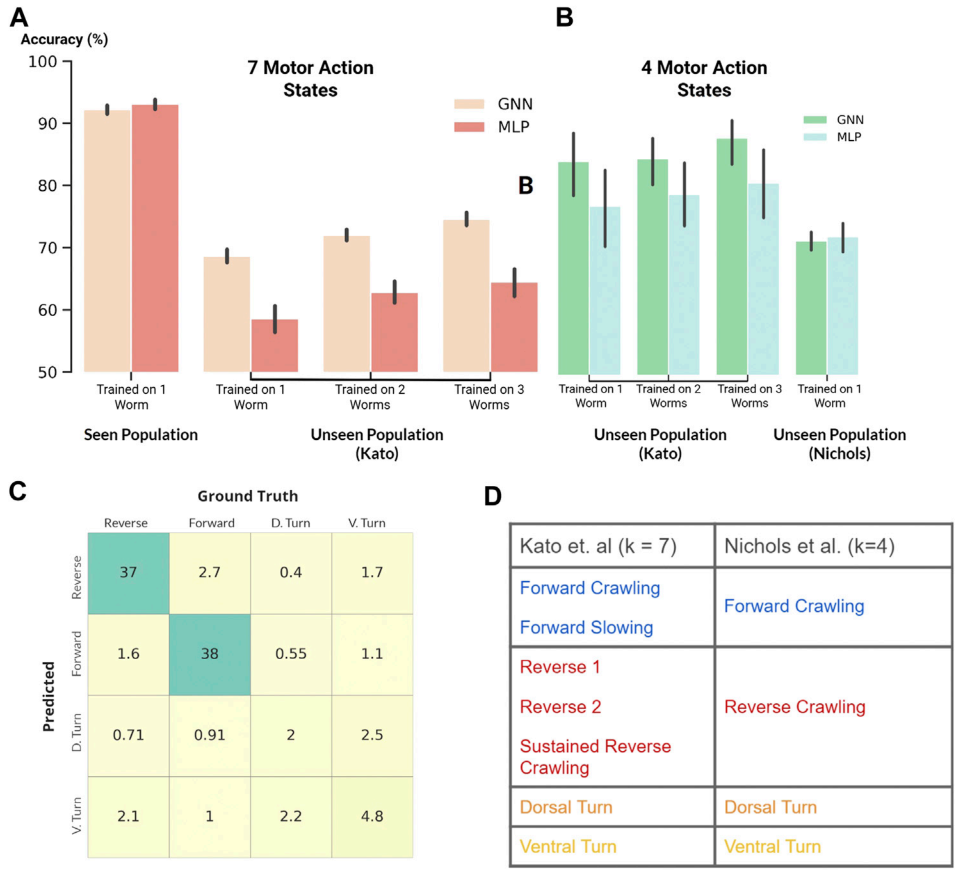
where $g_{graph:mlp}$ is a 2-layer MLP. Contrasting the MLP model, our GNN relies on message passing between connected nodes and contains an encoder for edge weights A_{ij}:

$$\boldsymbol{V} = g_{node}(\boldsymbol{X}), \tag{7}$$

$$E_{ij} = g_{edge}(\text{aggregation}(\boldsymbol{v}_i, \boldsymbol{v}_j)), \tag{8}$$

$$A_{ij} = sigmoid(E_{ij}), \tag{9}$$

where in **Eq. 7**, $\boldsymbol{V} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n, \ldots, \boldsymbol{v}_N)$ corresponds to the embedding of each node's features through the MLP g_{node}. Next, the edge embedding E_{ij} is computed by aggregating all pairs of node embeddings followed by the MLP g_{edge}. Finally, applying the sigmoid function to the edge embedding E_{ij} produces edge weights A_{ij} normalized between 0 and 1. $\boldsymbol{A}$ can be interpreted as an inferred weighted adjacency matrix where A_{ij} denotes the edge weight between nodes i and j such that $i = j$



	Reverse	Forward	D. Turn	V. Turn
Reverse	37	2.7	0.4	1.7
Forward	1.6	38	0.55	1.1
D. Turn	0.71	0.91	2	2.5
V. Turn	2.1	1	2.2	4.8

Kato et. al (k = 7)	Nichols et al. (k=4)
Forward Crawling Forward Slowing	Forward Crawling
Reverse 1 Reverse 2 Sustained Reverse Crawling	Reverse Crawling
Dorsal Turn	Dorsal Turn
Ventral Turn	Ventral Turn

FIGURE 4 | (A and B) Classification accuracy of our GNN and MLP models where black vertical lines show statistical spread **(A)**: Classification of seven motor action states within the Kato dataset **(B)**: Classification of four motor action states on both the Kato and Nichols datasets **(C)** Confusion matrix. Percent occurrence of predicted states against labeled states when evaluating on the Nichols dataset **(D)** Mapping of behavioral states between the Kato and Nichols dataset.

denotes a self edge. The edge weights either dynamically change in each timestep's inferred graph $\mathcal{G}_t$ or remain fixed for the whole temporal graph $\mathcal{G}$ of an individual worm. If the edges are static for the temporal graph, the aggregation step in **Eq. 8** also averages hidden features across all timesteps such that $\boldsymbol{V} = \frac{1}{T}\sum_{t=1}^{T} g_{node}(\boldsymbol{X}_t)$. Note that in this case, the edge encoder is given all timesteps $\boldsymbol{X}_\alpha$ in **Eqs 1, 4** instead of just one timestep.

After edges are encoded, the GNN performs a message passing **Eq. 10** and aggregation step **Eq. 11**:

$$\boldsymbol{M} = \boldsymbol{A}\boldsymbol{X}, \tag{10}$$

$$\boldsymbol{H}_{out} = g_{graph:gnn}(\text{aggregation}(\boldsymbol{M})) \tag{11}$$

As mentioned before, our MLP and GNN models can be subsituted for f in **Eqs 1, 4**. Depending on the task, the dimension of $\boldsymbol{H}_{out}$ for the MLP **Eq. 6** and GNN **Eq. 11** models differs. For behavioral state classification, $\boldsymbol{H}_{out} \in \mathbb{R}^k$ whereas for trajectory prediction, $dim(\boldsymbol{H}_{out}) = dim(\boldsymbol{X}_{\alpha,t})$ such that $\boldsymbol{H}_{out} \in \mathbb{R}^{N\times 2}$.

Theoretically, an arbitrary number of message passing steps can be implemented; however, we did not find any improvements when using more than one step. In addition, we find that performance improves when using concatenation instead of summation during the aggregation step.

4 EXPERIMENTS AND DATA

Our experiments were performed with data acquired in Kato et al., (2015) and Nichols et al., (2017). We summarize various details about the data in this section; however, we direct the reader to each respective publication for specific experimental details.

4.1 Calcium Imaging

Kato et al., (2015) showed that neural activity corresponding to the motor action sequence lives on low dimensional manifolds. To record neuron level dynamics, they performed whole-brain genetically encoded Ca^{2+} imaging with single-cell-resolution and measured ~ 100 neurons for around 18 min. They then normalized each calcium trace by peak fluorescence and identified neurons using spatial position and previous literature (Altun et al.,, 2002–2020). Aside from imaging freely moving worms, the authors also examined robustness of topological features to sensory stimuli changes, hub neuron silencing, and immobilization. For simplicity, we limited our experiments to data collected on freely moving worms.

Nichols et al., (2017) focused on differences in neural activity of *C. elegans* while awake or asleep and studied two different strains of worms, n2 (11 total worms) and npr1 (10 total worms). Because experiments in both studies were performed by the same group, most experimental procedures were similar, allowing us to easily process data to match the Kato dataset. While this dataset includes imaging data of each worm during quiescence, for consistency with the Kato dataset, we only included data before sleep was induced. Furthermore, we pooled results for both strains of worms as we did not notice any statistically relevant differences between them.

4.2 Dataset Enlargement

Although our data for each worm is relatively small (~ 3,000–4,000 timesteps), our datasets contained calcium traces from numerous worms. In total, 5 worms were measured in Kato et al., (2015) and 21 worms were measured in Nichols et al., (2017). Taking advantage of the large number of worms measured, we experimented with dataset enlargement where our models were trained on pooled data from different numbers of worms in the Kato dataset. Similarly, we pooled data from all 21 worms from the Nichols dataset; however, we use this dataset only during evaluation-i.e., the model never sees this dataset in training. In this way, we define the "seen" population as worms whose data was seen in training and the "unseen" population as worms the model did not see during training. More details about how datasets were used in our experiments can be found in **Section 4.2**.

To perform dataset enlargement, we separately trained the models on each worm in the seen population for each epoch. In other words, we independently optimized the loss function for each worm in every epoch. We followed this procedure such that batch normalization was separately performed on each worm's features. This technique was motivated by experiments where batch normalization on data from individual worms improved both test set and generalization accuracy. In contrast, performing batch normalization on pooled data from all worms greatly decreased model performance.

4.3 Data Processing

We normalized the calcium trace and its derivative of each neuron to [0,1]. Normalization was performed for the entire recorded calcium trace of a worm instead of within each batch because the relative magnitudes of the traces have been found to contain graded information about the worm's behavioral state, (e.g. crawling speed).

For the seen population, we separated each calcium trace of approximately 3,000–4,000 timesteps into batches of 8 timesteps where each timestep corresponds to roughly 1/3 of a second. We chose batch sizes of 8 timesteps because visualization of calcium traces showed that most local variations occur within this time frame. Moreover, 8 timesteps roughly corresponds to 3 s which is about the amount of time a worm needs to execute a behavioral change. Finally, the batches were shuffled before being divided into 10 folds later used for cross-validation, ensuring that each fold is representative across the whole dataset.

When evaluating on the unseen population, we treat the data differently for each task. For behavioral classification, we infer the behavioral state of the system using data from one timestep. As such, we do not split the data and simply run the model separately on each timestep of the worm's calcium traces. In contrast, for trajectory prediction, we split the calcium traces into batches of 16 timesteps and evaluate the model on all batches.

To compare with previous works, we performed our experiments on uniquely identified neurons between the datasets that we investigated. Identifying specific neurons is an experimental challenge, and as such, only a small fraction of neurons were unequivocally labeled. A total of 15 neurons were uniquely identified between all 5 worms measured in the Kato dataset: (AIBL, AIBR, ALA, AVAL, AVAR, AVBL, AVER, RID, RIML, RIMR, RMED, RMEL, RMER, VB01, VB02). In addition, the Nichols dataset contained data from 21 worms with 3 uniquely identified neurons shared among all worms in both datasets: (AIBR, AVAL, VB02).

5 RESULTS

Following Brennan and Proekt (2019), we used data from Kato et al., (2015) for training/evaluating our models and data from Nichols et al., (2017) as an extended evaluation set. Because whole brain imaging is incredibly difficult, our datasets were relatively small. To address this, we experimented with dataset enlargement (**Section 4.1.2**) by combining data from multiple worms in the Kato dataset during model training. For all experiments, we performed 10-fold cross validation on all permutations of worms in our training set. More details, along with supplemental experiments, can be found in the Supplementary Information.

5.1 Behavioral State Classification

Our first experiment compared the performance of our models to state-of-the-art results reported in Brennan and Proekt (2019). Specifically, this experiment involved the classification of only two motor action states, forward and reverse crawling. Along with our models described above, we also experimented with a support vector machine (SVM) and a GNN which computes with edges derived from the physical connectome (White et al.,, 1986). In particular, we incorporated the connectome into our model to investigate whether physical/structural connections between neurons can serve as a favourable inductive bias for our GNN. Our results are shown in **Table 1** where "Seen Population" denotes test set accuracy after training on the same worm and "Unseen Population" denotes evaluation/generalization accuracy on worms unseen during training.

Our deep learning models clearly outperformed the SVM and state-of-the-art results, demonstrating the ability of our models to successfully classify behavioral states and generalize to other worms. Interestingly, the SVM matched the performance of our deep learning models on the seen population; however, its generalization performance on unseen individuals was significantly worse than our deep learning models. As such, the SVM distinctly illustrates challenges of individual

TABLE 1 | Classification accuracy of forward and reverse crawling.

	Seen population	Unseen population (kato)	Unseen population (nichols)
Brennan and Proekt (2019)	83	81	—
SVM	98.8 ± 0.4	82.8 ± 7.6	79.0 ± 11.7
MLP	99.3 ± 0.6	93.9 ± 10.3	88.9 ± 11.4
GNN (connectome)	99.5 ± 0.6	96.8 ± 4.3	85.5 ± 12.9
GNN	99.5 ± 0.5	97.7 ± 3.1	95.5 ± 6.1

variability for model development in neural systems despite the simplicity of our experiments which involve the same set of unequivocally identified neurons. Similarly, our GNN using edges derived from the connectome performed well on the seen population but generalized worse than when using inferred edges. We hypothesize that the detrimental effect of using the connectome may be attributed to the distinction between inferred/functional and structural connectivity. In particular, the connectome maps physical connections between neurons which is generally conserved between different individuals. In contrast, individual variability of neural activity implicitly implies that the inferred/functional connectivity is unique to individuals (**Supplementary Section S1.4.3**).

Following the previous experiment, we applied our MLP and GNN models to the harder task of classifying all behavioral states labeled in the Kato dataset (**Figure 4A**). Within this dataset, 7 states were labeled: Forward Crawling, Forward Slowing, Reverse 1, Reverse 2, Sustained Reverse Crawling, Dorsal Turn, and Ventral Turn. In comparison to the Kato dataset, only 4 states were labeled in the Nichols dataset: reverse crawling, forward crawling, ventral turn, and dorsal turn. For compatibility, we mapped the 7 states of the Kato dataset to 4 states of the Nichols dataset when using the Nichols dataset as an extended evaluation set (**Figure 4D**).

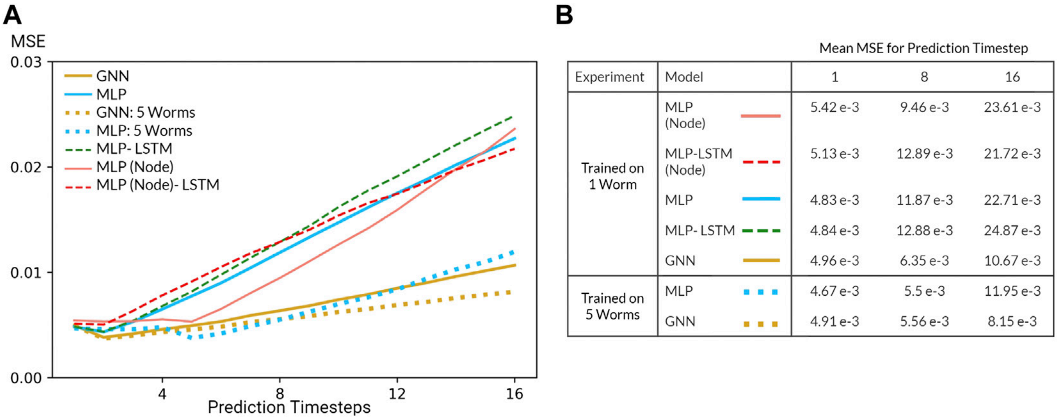
Despite the harder task of classifying 7 states, our models achieved a classification accuracy of ~92% on the same worm (**Figure 4A**). Moreover, our GNN trained on three worms in the Kato dataset generalized with an accuracy of 87% (**Figure 4B**) when classifying 4 states on the remaining unseen worms. This substantially exceeds the performance of our MLP model and Brennan and Proekt (2019) who report a 81% cross-animal accuracy on two states. Nevertheless, both MLP and GNN models generalized equally well (~ 70%) to the 21 unseen worms of the Nichols dataset. These experiments consistently demonstrate that our GNN exceeds the performance of state-of-the-art techniques and also often exceeds the performance of our baseline MLP model.

5.2 Neuron-Level Trajectory Prediction

For trajectory prediction, we predicted each neuron's calcium trace and its derivative (normalized to [0,1]) for 8 timesteps during training (seen population) and 16 timesteps during evaluation/validation (unseen population). While training our Markovian models, scheduled sampling was performed to minimize the accumulation of error (Bengio et al.,, 2015). When evaluating on the unseen population, the model was given one timestep as the initial condition after which the model predicts 16 timesteps. In addition to our Markovian models, we also experimented with RNN implementations trained with burn-in periods of four timesteps (12 timesteps during training and 20 timesteps during evaluation). Our experiments primarily focused on generalization performance of our models on the extended evaluation/Nichols dataset (**Figure 5**).

Predicting neuron-level trajectory using deep learning is fairly novel since advances in whole-brain imaging are recent and limited to few organisms. Nevertheless, neural systems generically fall under the category of dynamical systems where each neuron is described by a differential equation such that neural activity can be modeled as a system of coupled differential equations. Under this formulation, the task of trajectory prediction involves learning the underlying physical laws in order to predict the time evolution of the system. To quantify the predictive power of our models, we evaluated the mean squared error (MSE) of each prediction timestep relative to the true trajectory. In the context of our Markovian model, this metric measures the error of the predicted transition matrix which time evolves the state of the system and, by extension, demonstrates the ability of our models to learn the underlying physical laws of the dynamical system.

Several challenges limited the predictive power of our models. Most prominently, our system is inherently non-linear and potentially chaotic, a fact further exasperated by the nature of calcium imaging which is notoriously noisy and an indirect measurement of neural activity. In addition, our datasets are relatively small in spite of our dataset enlargement technique. Resulting from these challenges, the performance of our model is poor, especially in comparison to that of models in data assimilation which leverage a priori knowledge of the dynamical system (). Nevertheless, inspecting the MSE as a function of prediction step (**Figure 5**) reveals that our models are able to learn how the system transitions up to a short timescale. Moreover, increasing the number of worms included during training (dataset enlargement) also improved generalization performance of our MLP and GNN models. Perhaps most surprising, our Markovian GNN outperformed all MLP models and their derived RNN variants. We attribute this result to the largely deterministic nature of neural dynamics, characterized by sparse bifurcations on the latent manifold, and the inductive bias of GNNs. As a result, given 1 timestep, our GNN outperformed all other models including RNN variants which were given 4 burn-in timesteps. Therefore, we conclude that our GNN



Experiment	Model		Mean MSE for Prediction Timestep		
			1	8	16
Trained on 1 Worm	MLP (Node)	—	5.42 e-3	9.46 e-3	23.61 e-3
	MLP-LSTM (Node)	- - -	5.13 e-3	12.89 e-3	21.72 e-3
	MLP	—	4.83 e-3	11.87 e-3	22.71 e-3
	MLP- LSTM	- - -	4.84 e-3	12.88 e-3	24.87 e-3
	GNN	—	4.96 e-3	6.35 e-3	10.67 e-3
Trained on 5 Worms	MLP	▪ ▪ ▪	4.67 e-3	5.5 e-3	11.95 e-3
	GNN	▪ ▪ ▪	4.91 e-3	5.56 e-3	8.15 e-3

FIGURE 5 | (A) Mean squared error (MSE) of the GNN and various MLP models evaluated on the Nichols dataset. All models were trained using data from one worm or five worms in the Kato Dataset **(B)** Table of mean MSE values for all models for 1, 8, and 16 timesteps.

displays a favourable inductive bias in contrast to graph-agnostic models on the task of predicting microscopic dynamics.

6 DISCUSSION

For both tasks, our GNN consistently matched or exceeded our MLP model which we accredit to its favourable inductive bias. Kato et al., (2015) established that projecting neural dynamics onto three principal components for each worm reveals universal topological structures; however, attempts to project neural dynamics onto shared principal components of all worms failed to display any meaningful structure. Thus, variability in each worm's neural activity, corresponding to low dimensional manifolds in latent space, is represented by different linear combinations of neurons. In other words, relevant topological structures in latent space are loosely related by linear transformations of node features. We speculate that our GNN's performance stems from its explicit structure of message passing along inferred edges which is analogous to learning linear transformations of node features (**Eq. 10**). Based on our experimental results, we further speculate that this inductive bias proves favourable on both microscopic and macroscopic machine learning tasks in neural systems.

Interestingly, our model's performance was not significantly impacted by using 3 neurons (~ 1% of all neurons) instead of 15 (~ 5% of all neurons). This is not surprising because neurons strongly coupled to the motor action sequence retain most information (Gao and Ganguli, 2015), a fact consistent with Brennan and Proekt (2019) who found that strategically choosing 1 neuron retains ~ 75% of the information contained in the larger set of 15 neurons.

Finally, as a critical question, we ask whether our model's performance stems from choosing a stereotyped organism that is well studied and biologically simple, or if our results imply a path toward generalizable/universal machine learning in neural systems. While the neurophysiology of *C. elegans* is quite complex, the motor action sequence we studied is relatively simple, especially in comparison to other organisms and cognitive functions. Moreover, organisms are adaptive and capable of learning new behavior, a fact not represented in our dataset. However, a recent astounding study Gallego et al., (2020) measured neural dynamics in monkeys trained to perform action sequences and determined that learned latent dynamics live in low-dimensional manifolds that were conserved throughout the length of the study. By aligning latent dynamics, their model accurately decoded the action of monkeys up to two years after the model was trained despite changes in biology, (e.g. neuron turnover, adaptation to implants). Consequently, we posit that techniques similar to those used in our model may broadly apply to more complex organisms and functions.

7 CONCLUSION

In this study, we examined the ability of neural networks to classify higher-order function and predict neuron level dynamics. In addition, inspired by global organizational principles of behavior discovered in previous studies, we demonstrated the ability of neural networks to generalize to unseen organisms. Specifically, we first showed that our models exceed the performance of previous studies in behavioral state classification of *C. elegans*. Next, we found that a simple MLP performs remarkably well on unseen organisms. Nevertheless, our graph neural network, which explicitly learns linear transformations of node features, matched or exceeded the performance of graph agnostic models in all experiments. These experiments demonstrate that our models are capable of successful evaluation on unseen organisms, both within the same study, and in a separate experiment spaced years apart. Finally, our results show that dataset enlargement through the inclusion of more individuals can significantly improve generalization performance in microscopic neural systems.

We note that our results of generalization on both higher-order functions and neuron-level dynamics (macroscopic and microscopic) suggests wide applicability of our technique to numerous machine learning tasks in neuroscience and hierarchical dynamical systems. A promising research direction is the hierarchical relationship between neuron-level and population-level dynamics. Breakthroughs in this direction may inform machine learning models working with population-level functional and imaging techniques, such as EEG or fMRI, which are readily available and widespread. In addition, in this study, we only focused on simple machine learning tasks and imaging data taken under similar experimental conditions. Further studies may involve more complex tasks such as those involving graded information in neural dynamics, changes in sensory stimuli, acquisition of learned behaviors, and higher-order functions comprised of complicated sequences of behavior. From a machine learning perspective, the development of a recurrent graph neural network for the edge encoder with a suitable attention mechanism may aid model generalization. Additional work is also needed in examining and improving model performance on arbitrary sets of neurons as neuron identification is experimentally challenging and limited to small systems.

DATA AVAILABILITY STATEMENT

The original data used in the analysis in our paper can be found in original citations and associated repositories on OSF: https://osf.io/2395t/ (Kato et al.,); https://osf.io/kbf38/ (Nichols et al.,). Additional inquiries can be directed to the corresponding authors.

AUTHOR CONTRIBUTIONS

Experiments and models were conceived by PW. SS assisted with the implementation of various algorithms. The manuscript was written and revised after numerous iterations by all the authors.

FUNDING

This work was supported by unrestricted funds to the Center for Engineered Natural Intelligence.

ACKNOWLEDGMENTS

The authors thank the authors of Brennan and Proekt (2019) for graciously allowing reproductions of their figures. In addition, the authors thank the Zimmer Lab for making their data available online (data from Kato et al., (2015) and Nichols et al., (2017) can be found here). PW is grateful to Ilya Valmianski for insightful discussion and guidance. Finally, the authors greatly appreciate the dedication and effort of the reviewers whose comments have been invaluable.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frai.2021.618372/full#supplementary-material.

REFERENCES

Altun, Z. F., Herndon, L. A., Wolkow, C. A., Crocker, C., Lints, R., and Hall, D. H. (2002–2020). Worm atlas. Wormbase, WBPaper00012319

Bargmann, C. I., and Marder, E. (2013). From the connectome to brain function. *Nat. Methods* 10, 483. doi:10.1038/nmeth.2451

Bashivan, P., Rish, I., Yeasin, M., and Codella, N. (2015). Learning representations from eeg with deep recurrent-convolutional neural networks. arXiv preprint: https://arxiv.org/abs/1511.06448 (Accessed November 19, 2015).doi:10.1109/spmb.2015.7405422

Battaglia, P., Hamrick, J. B. C., Bapst, V., Sanchez, A., Zambaldi, V., Malinowski, M., et al. (2018). Relational inductive biases, deep learning, and graph networks. arXiv preprint:https://arxiv.org/pdf/1806.01261.pdf. (Accessed October 17, 2018).

Battaglia, P., Pascanu, R., Lai, M., Rezende, D. J., et al. (2016). Interaction networks for learning about objects, relations and physics. Neural information processing systems, 4502–4510. (Accessed December 1, 2016).

Bengio, S., Vinyals, O., Jaitly, N., and Shazeer, N. (2015). "Scheduled sampling for sequence prediction with recurrent neural networks," in Proceedings of the 28th International Conference on Neural Information Processing Systems, December 2015, Montreal, Canada (California, CA: Neural Information Processing Systems), 1171–1179.

Brennan, C., and Proekt, A. (2019). A quantitative model of conserved macroscopic dynamics predicts future motor commands. *Elife.* 8, e46814. doi:10.7554/eLife.46814

Brown, C. J., and Hamarneh, G. (2016). Machine learning on human connectome data from mri. arXiv preprint:https://arxiv.org/abs/1611.08699 (Accessed November 26, 2016).

Brown, C. J., Kawahara, J., and Hamarneh, G. (2018). "Connectome priors in deep neural networks to predict autism," in 15th international symposium on biomedical imaging (ISBI 2018), April 2018, 110–113. IEEE.

Churchland, M. M., Cunningham, J. P., Kaufman, M. T., Ryu, S. I., and Shenoy, K. V. (2010). Cortical preparatory activity: representation of movement or first cog in a dynamical machine? *Neuron* 68, 387–400. doi:10.1016/j.neuron.2010.09.015

Cohen, T., and Welling, M. (2016). "Group equivariant convolutional networks," in International conference on machine learning, June, 2016, New York, NY, USA. 2990–2999.

Cook, S. J., Jarrell, T. A., Brittin, C. A., Wang, Y., Bloniarz, A. E., Yakovlev, M. A., et al. (2019). Whole-animal connectomes of both caenorhabditis elegans sexes. *Nature* 571, 63–71. doi:10.1038/s41586-019-1352-7

Dehmamy, N., Barabási, A.-L., and Yu, R. (2019). Understanding the representation power of graph neural networks in learning graph topology. arXiv preprint:https://arxiv.org/abs/1907.05008 (Accessed July 11, 2019), 15413–15423.

Frégnac, Y. (2017). Big data and the industrialization of neuroscience: a safe roadmap for understanding the brain? *Science* 358, 470–477. doi:10.1126/science.aan8866

Gallego, J. A., Perich, M. G., Chowdhury, R. H., Solla, S. A., and Miller, L. E. (2020). Long-term stability of cortical population dynamics underlying consistent behavior. *Nat. Neurosci.* 23, 260–270. doi:10.1038/s41593-019-0555-4

Gao, P., and Ganguli, S. (2015). On simplicity and complexity in the brave new world of large-scale neuroscience. *Curr. Opin. Neurobiol.* 32, 148–155. doi:10.1016/j.conb.2015.04.003

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. (2017). "Neural message passing for quantum chemistry," in Proceedings of the 34th

International Conference on Machine Learning, August 6-11, 2017, Sydney, Australia 70, 1263–1272.

Glaser, J. I., Benjamin, A. S., Farhoodi, R., and Kording, K. P. (2019). The roles of supervised machine learning in systems neuroscience. *Progress in neurobiology.* 175, 126–137. doi:10.1016/j.pneurobio.2019.01.008

Gleeson, P., Lung, D., Grosu, R., Hasani, R., and Larson, S. D. (2018). c302: a multiscale framework for modeling the nervous system of caenorhabditis elegans. *Phil. Trans. R. Soc. B.* 373, 20170379. doi:10.1098/rstb.2017.0379

Goldman, M., Golowasch, J., Marder, E., and Abbott, L. (2001). Global structure, robustness, and modulation of neuronal models. *J. Neurosci.* 21, 5229–5238. doi:10.1523/jneurosci.21-14-05229.2001

Golowasch, J., Goldman, M. S., Abbott, L., and Marder, E. (2002). Failure of averaging in the construction of a conductance-based neuron model. *J. Neurophysiol.* 87, 1129–1131. doi:10.1152/jn.00412.2001

Granger, C. W. J. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37, 424–438. doi:10.2307/1912791

Hamilton, W., Ying, Z., and Leskovec, J. (2017). Inductive representation learning on large graphs-Advances in neural information processing. arXiv preprint: https://arxiv.org/abs/1706.02216 (Accessed July 7, 2017), 1024–1034.

Kaplan, H. S., Salazar Thula, O., Khoss, N., and Zimmer, M. (2020). Nested neuronal dynamics orchestrate a behavioral hierarchy across timescales. *Neuron* 105, 562–e9. doi:10.1016/j.neuron.2019.10.037

Kato, S., Kaplan, H. S., Schrödel, T., Skora, S., Lindsay, T. H., Yemini, E., et al. (2015). Global brain dynamics embed the motor command sequence of caenorhabditis elegans. *Cell.* 163, 656–669. doi:10.1016/j.cell.2015.09.034

Kim, B. H., and Ye, J. C. (2020). Understanding graph isomorphism network for rs-fMRI functional connectivity analysis. *Front. Neurosci.* 14, 630. doi:10.3389/fnins.2020.00630

Kipf, T., Fetaya, E., Wang, K.-C., Welling, M., and Zemel, R. (2018). Neural relational inference for interacting systems. arXiv preprint:https://arxiv.org/abs/1802.04687 (Accessed February 13, 2018), 2688–2697.

Kipf, T. N., and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint:https://arxiv.org/abs/1609.02907 (Accessed September 6, 2016),

Kwak, N. S., Müller, K. R., and Lee, S. W. (2017). A convolutional neural network for steady state visual evoked potential classification under ambulatory environment. *PloS One.* 12, e0172578, doi:10.1371/journal.pone.0172578

Li, X., and Duncan, J. (2020). Braingnn: interpretable brain graph neural network for fmri analysis. bioRxiv:https://www.biorxiv.org/content/10.1101/2020.05.16.100057v1 (Accessed May 22, 2020).

Li, Y., Yu, R., Shahabi, C., and Liu, Y. (2018). Diffusion convolutional recurrent neural network: data-driven traffic forecasting. arXiv preprint:https://arxiv.org/abs/1707.01926 (Accessed July 6, 2017).

Löwe, S., Madras, D., Zemel, R., and Welling, M. (2020). Amortized causal discovery: learning to infer causal graphs from time-series data. arXiv preprint:https://arxiv.org/abs/2006.10833 (Accessed June 18, 2020).

Lundervold, A. S., and Lundervold, A. (2019). An overview of deep learning in medical imaging focusing on mri. *Z. Med. Phys.* 29, 102–127. doi:10.1016/j.zemedi.2018.11.002

Makin, J. G., Moses, D. A., and Chang, E. F. (2020). Machine translation of cortical activity to text with an encoder–decoder framework. *Nat. Neurosci.* 23, 575–582. doi:10.1038/s41593-020-0608-8

Mensch, A., Mairal, J., Bzdok, D., Thirion, B., and Varoquaux, G. (2017). "Learning neural representations of human cognition across many fmri studies Advances in neural information processing systems. Available at:https://arxiv.org/abs/1710.11438 (Accessed October 31, 2017), 5883–5893.

Nichols, A. L. A., Eichler, T., Latham, R., and Zimmer, M. (2017). A global brain state underlies c. elegans sleep behavior. *Science.* 356. doi:10.1126/science.aam6851

Prevedel, R., Yoon, Y. G., Hoffmann, M., Pak, N., Wetzstein, G., Kato, S., et al. (2014). Simultaneous whole-animal 3d imaging of neuronal activity using light-field microscopy. *Nat. Methods.* 11, 727–730. doi:10.1038/nmeth.2964

Prinz, A. A., Bucher, D., and Marder, E. (2004). Similar network activity from disparate circuit parameters. *Nat. Neurosci.* 7, 1345–1352. doi:10.1038/nn1352

Raposo, D., Santoro, A., Barrett, D., Pascanu, R., Lillicrap, T., and Battaglia, P. (2017). Discovering objects and their relations from entangled scene representations. Available at:https://arxiv.org/abs/1702.05068 (Accessed February 16, 2017).

Sarma, G. P., Lee, C. W., Portegys, T., Ghayoomie, V., Jacobs, T., Alicea, B., et al. (2018). Openworm: overview and recent advances in integrative biological simulation of caenorhabditis elegans. *Phil. Trans. R. Soc. B.* 373, 20170382. doi:10.1098/rstb.2017.0382

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. (2009). The graph neural network model. *IEEE Trans. Neural Network.* 20, 61–80. doi:10.1109/TNN.2008.2005605

Skora, S., Mende, F., and Zimmer, M. (2018). Energy scarcity promotes a brain-wide sleep state modulated by insulin signaling in c. elegans. *Cell Rep.* 22, 953–966. doi:10.1016/j.celrep.2017.12.091

Varshney, L. R., Chen, B. L., Paniagua, E., Hall, D. H., and Chklovskii, D. B. (2011). Structural properties of the caenorhabditis elegans neuronal network. *PLoS Comput. Biol.* 7, e1001066. doi:10.1371/journal.pcbi.1001066

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., and Solomon, J. M. (2019). Dynamic graph cnn for learning on point clouds. *ACM Trans. Graph.* 38, 1–12. doi:10.1145/3326362

Wen, C., and Kimura, K. D. (2020). How do we know how the brain works?—analyzing whole brain activities with classic mathematical and machine learning methods. *Jpn. J. Appl. Phys.* 59, 030501. doi:10.35848/1347-4065/ab77f3

White, J. G., Southgate, E., Thomson, J. N., and Brenner, S. (1986). The structure of the nervous system of the nematode caenorhabditis elegans. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* 314, 1–340. doi:10.1098/rstb.1986.0056

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. (2018). How powerful are graph neural networks. Available at:https://arxiv.org/abs/1810.00826 (Accessed October 1, 2018).

Yu, B., Yin, H., and Zhu, Z. (2018). Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. Available at:https://arxiv.org/abs/1709.04875 (Accessed September 14, 2017), 3634–3640.

Zhang, X., Yao, L., Wang, X., Monaghan, J., Mcalpine, D., and Zhang, Y. (2019). A survey on deep learning based brain computer interface: recent advances and new frontiers. arXiv preprint: https://arxiv.org/abs/1905.04149(Accessed May 10, 2019)

Zhang, Y., and Bellec, P. (2019). "Functional annotation of human cognitive states using graph convolution networks", in 33rd Conference on Neural Information Processing Systems, December 8-14, 2019 Vancouver, Canada.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

ORIGINAL RESEARCH
published: 18 March 2021
doi: 10.3389/fams.2020.616584

Reconstructing Complex Cardiac Excitation Waves From Incomplete Data Using Echo State Networks and Convolutional Autoencoders

Sebastian Herzog[1,2], *Roland S. Zimmermann*[3,4], *Johannes Abele*[1,4], *Stefan Luther*[1,5,6] and *Ulrich Parlitz*[1,4,6]*

[1]Max Planck Institute for Dynamics and Self-Organization, Göttingen, Germany, [2]Third Institute of Physics and Bernstein Center for Computational Neuroscience, University of Göttingen, Göttingen, Germany, [3]Tübingen AI Center, University of Tübingen, Tübingen, Germany, [4]Institute for the Dynamics of Complex Systems, University of Göttingen, Göttingen, Germany, [5]Institute of Pharmacology and Toxicology, University Medical Center Göttingen, Göttingen, Germany, [6]German Center for Cardiovascular Research (DZHK), Partner Site Göttingen, Göttingen, Germany

OPEN ACCESS

Edited by:
Axel Hutt,
Inria Nancy - Grand-Est Research Centre, France

Reviewed by:
Petia D. Koprinkova-Hristova,
Institute of Information and Communication Technologies (BAS), Bulgaria
Meysam Hashemi,
INSERM U110 6 Institut de Neurosciences des Systèmes, France

****Correspondence:***
Ulrich Parlitz
ulrich.parlitz@ds.mpg.de

Specialty section:
This article was submitted to Dynamical Systems, a section of the journal Frontiers in Applied Mathematics and Statistics

Received: *12 October 2020*
Accepted: *07 December 2020*
Published: *18 March 2021*

Citation:
Herzog S, Zimmermann RS, Abele J, Luther S and Parlitz U (2021) Reconstructing Complex Cardiac Excitation Waves From Incomplete Data Using Echo State Networks and Convolutional Autoencoders. Front. Appl. Math. Stat. 6:616584. doi: 10.3389/fams.2020.616584

The mechanical contraction of the pumping heart is driven by electrical excitation waves running across the heart muscle due to the excitable electrophysiology of heart cells. With cardiac arrhythmias these waves turn into stable or chaotic spiral waves (also called rotors) whose observation in the heart is very challenging. While mechanical motion can be measured in 3D using ultrasound, electrical activity can (so far) not be measured directly within the muscle and with limited resolution on the heart surface, only. To bridge the gap between measurable and not measurable quantities we use two approaches from machine learning, echo state networks and convolutional autoencoders, to solve two relevant data modelling tasks in cardiac dynamics: Recovering excitation patterns from noisy, blurred or undersampled observations and reconstructing complex electrical excitation waves from mechanical deformation. For the synthetic data sets used to evaluate both methods we obtained satisfying solutions with echo state networks and good results with convolutional autoencoders, both clearly indicating that the data reconstruction tasks can in principle be solved by means of machine learning.

Keywords: reservoir computing, convolutional autoencoder, image enhancement, cross-prediction, cardiac arrhythmias, excitable media, electro-mechanical coupling, cardiac imaging

1 INTRODUCTION

Cardiac arrhythmias, such as ventricular or atrial fibrillation, are electro-mechanical dysfunctions of the heart that are associated with complex, chaotic spatio-temporal excitation waves within the heart muscle resulting in incoherent mechanical contraction and a significant loss of pump function [1–3]. Ventricular fibrillation (VF) is the most common deadly manifestation of a cardiac arrhythmia and requires immediate defibrillation using high-energy electric shocks. Atrial fibrillation (AF) is the most common form of a cardiac arrhythmia, affecting 33 million patients worldwide [62]. While not immediately life-threatening, AF is considered to be responsible for 15% of strokes if left untreated [63, 64]. The structural substrate and functional mechanisms that underlie the onset and perpetuation of VF and AF are not fully understood. It is generally agreed that imaging of the cardiac electrical and mechanical function is key to an improved mechanistic understanding of cardiac disease and the development of novel diagnosis and therapy. This has motivated the development of non-invasive and invasive electrophysiological measurement and imaging modalities. Electrical activity of the heart can (so far) non-invasively be measured on its surface, only. Direct measurements can be made *in vivo* inside the heart using so-called basket catheters with typically 64 electrodes or in *ex-vivo* experiments, where an extracted heart in a Langendorff perfusion set-up is kept beating and the cell membrane voltage on the epicardial surface is made visible using fluorescent dyes (a method also known as optical mapping) [4]. A method for indirect observation of electrical excitation waves is ECG imaging where an array of EEG-electrodes is placed on the body surface and an (ill-posed) inverse problem is solved to estimate the potential on the surface of the heart. Mechanical contraction and deformation of the heart tissue can be studied in full 3D using ultrasound, in 2D using real-time MRT [5] or (using optical mapping) by motion tracking in Langendorff experiments.

The reconstruction of patterns of action potential wave propagation in cardiac tissue from ultrasound has been introduced by Otani et al. [6, 7]. They proposed to use ultrasound to visualize the propagation of these waves through the mechanical deformations they induce and to reconstruct action potential-induced active stress from the deformation. Provost et al. [8] introduced electromechanical wave imaging to map the mechanical deformation of cardiac tissue at high temporal and spatial resolutions. The observed deformations resulting from the electrical activation were found to be closely correlated with electrical activation sequences. The cardiac excitation-contraction-coupling (ECC) [9] has also been studied in optical mapping experiments in Langendorff-perfused isolated hearts [10–12]. Using electromechanical optical mapping [12], it was shown that during ventricular tachyarrhythmias electrical rotors introduce corresponding rotating mechanical waves. These co-existing electro-mechanical rotors were observed on the epicardial surface of isolated Langendorff-perfused intact pig and rabbit hearts using optical mapping [13]. Using high-resolution ultrasound, these mechanical rotors were also observed inside the ventricular wall during ventricular tachycardia and fibrillation [13].

All these measurement modalities are limited, in particular those suitable for *in vivo* applications. Measurements with basket catheters are effectively undersampling the spatio-temporal wave pattern. Inverse ECGs suffer from ill-posedness and require regularization that may lead to loss of spatial resolution and blurring. Limited spatial resolution is also an issue with ultrasound measurements, but they are currently the only way to “look inside” the heart, albeit measuring only mechanical motion. Electrical excitation waves inside the heart muscle are so far not accessible by any measurement modality available.

These limitations motivated the search for algorithms to reconstruct electro-mechanical wave dynamics in cardiac tissue from measurable quantities. Berg et al. [14] devised synchronization-based system identification of extended excitable media, in which model parameters are estimated by minimizing the synchronization error. Using this approach, Lebert and Christoph [15] demonstrated that electro-mechanic wave dynamics of excitable-deformable media can be recovered from a limited set of observables using a synchronization-based data assimilation approach. Hoffman et al. reconstructed electrical wave dynamics using ensemble Kalman filters [16, 17]. In another approach, it was shown that echo state networks [18] and deep convolutional neural networks [19, 20] provide excellent cross estimation results for different variables of a mathematical model describing complex electrical excitation waves during cardiac arrhythmias. Following this approach, Christoph and Lebert [21] demonstrated the reconstruction of electrical excitation and active stress from deformation using a simulated deformable excitable medium. To continue this research and to address the general challenge of missing or impaired observations we consider in this article two tasks: (i) recovering electrical excitation patterns from noisy, blurred or undersampled observations and (ii) reconstructing electrical excitation waves from mechanical deformation. To solve the corresponding data processing and cross-prediction tasks two machine learning methods are employed and evaluated: echo state networks and convolutional autoencoders. Both algorithms are applied to synthetical data generated by prototypical models for electrophysiology and electromechanical coupling.

2 METHODS

In this section we will first introduce in **Sections 2.1 and 2.2** the mathematical models describing cardiac dynamics which were used to generate the example data for the two tasks to be solved: (i) recovering electrical wave pattern from impaired observations and (ii) cross-predicting electrical excitation from mechanical deformation. Then in **Section 2.3** both machine learning methods used for solving these tasks, echo state networks (**Section 2.3.1**) and convolutional autoencoders (**Section 2.3.2**), will be briefly introduced.

2.1 Recovering Complex Spatio-Temporal Wave Patterns From Impaired Observations

For motivating, illustrating, and evaluating the employed methods for dealing with incomplete or distorted observations we shall use spatio-temporal time series generated with the Bueno-Orovio-Cherry-Fenton (BOCF) model [22] describing complex electrical excitation patterns in the heart during cardiac arrhythmias. The BOCF model is a set of partial differential equations (PDEs) with four variables and will be introduced in **Section 2.1.1**. In **Section 2.1.2** a formal description of the data recovery tasks will be given.

TABLE 1 | TNNP model parameter values for the BOCF model [22].

u_o	0	τ_{v2}^-	1150	τ_{fi}	0.11	τ_{s1}	2.7342	τ_{s2}	3	τ_{o1}	6	τ_{o2}	6
u_u	1.58	τ_v^+	1.4506	τ_{w1}^-	70	τ_{w2}^-	20	τ_{so1}	43	τ_{so2}	0.2	τ_{si}	2.8723
θ_v	0.3	$\tau_{w\infty}$	0.07	τ_{v1}^-	60	τ_w^+	280	k_s	2.0994	w_∞^*	0.94	θ_w	0.015
u_s	0.9087	θ_v^-	0.015	k_w^-	65	θ_o	0.006	u_w^-	0.03	k_{so}	2	u_{so}	0.65

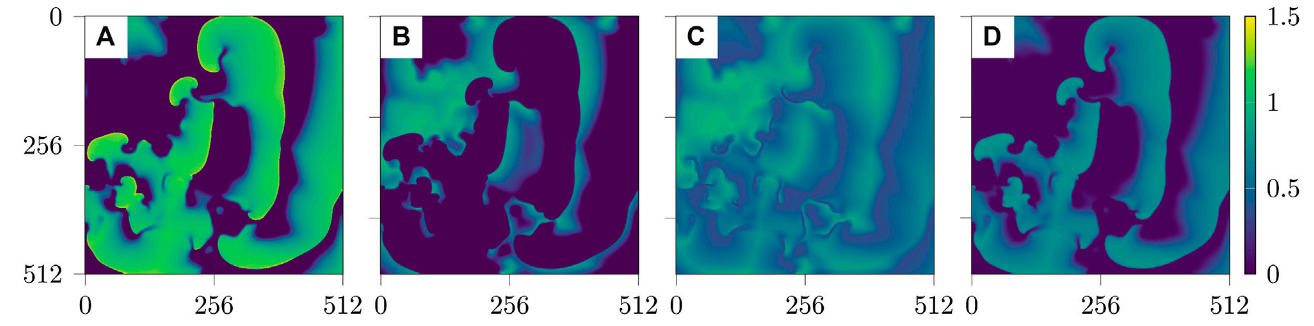


FIGURE 1 | Snapshots of the four fields u, v, w, s of the BOCF model Eq. (1) (from left to right).

2.1.1 Bueno–Orovio–Cherry–Fenton Model

Cardiac dynamics is controlled by electrical excitation waves triggering mechanical contractions of the heart. In the case of cardiac arrhythmias like lethal ventricular fibrillation, wave break-up and complex chaotic wave patterns occur resulting in significantly reduced pump performance of the heart. From the broad range of mathematical models describing this spatio-temporal dynamics [23] we chose the Bueno–Orovio–Cherry–Fenton (BOCF) model [22] to generate spatio-temporal time series that are used as a benchmark to validate our approaches for reconstructing complex wave patterns in excitable media from incomplete data. The BOCF model consists of four system variables whose evolution is given by four (partial) differential equations

$$\begin{aligned}
\frac{\partial u}{\partial t} &= D\cdot\nabla^2 u - \left(J_{si} + J_{fi} + J_{so}\right)\\
\frac{\partial v}{\partial t} &= \frac{1}{\tau_v^-}\left(1 - H(u-\theta_v)\right)(v_\infty - v) - \frac{1}{\tau_v^+}H(u-\theta_v)v\\
\frac{\partial w}{\partial t} &= \frac{1}{\tau_w^-}\left(1 - H(u-\theta_w)\right)(w_\infty - w) - \frac{1}{\tau_w^+}H(u-\theta_w)w\\
\frac{\partial s}{\partial t} &= \frac{1}{2\tau_s}\left((1 + \tanh(k_s(u-u_s))) - 2s\right).
\end{aligned} \tag{1}$$

The variable u represents the continuum limit representation of the membrane voltage of cardiac cells and the variables v, w, and s are gating variables controlling ionic transmembrane currents J_{si}, J_{fi} and J_{so} given by the equations

$$\begin{aligned}
J_{si} &= -\frac{1}{\tau_{si}}H(u-\theta_w)ws\\
J_{fi} &= -\frac{1}{\tau_{fi}}vH(u-\theta_v)(u-\theta_v)(u_u - u)\\
J_{so} &= \frac{1}{\tau_o}(u-u_o)(1 - H(u-\theta_w)) + \frac{1}{\tau_{so}}H(u-\theta_w).
\end{aligned} \tag{2}$$

Here $H(\cdot)$ denotes the Heaviside function and the currents depend on the following seven voltage controlled variables

$$\begin{aligned}
\tau_v^- &= (1 - H(u-\theta_v^-))\tau_{v1}^- + H(u-\theta_v^-)\tau_{v2}^-\\
\tau_w^- &= \tau_{w1}^- + \frac{1}{2}(\tau_{w2}^- - \tau_{w1}^-)(1 + \tanh(k_w^-(u-u_w^-)))\\
\tau_{so}^- &= \tau_{so1} + \frac{1}{2}(\tau_{so2} - \tau_{so1})(1 + \tanh(k_{so}(u-u_{so})))\\
\tau_s &= (1 - H(u-\theta_w))\tau_{s1} + H(u-\theta_w)\tau_{s2}\\
\tau_o &= (1 - H(u-\theta_o))\tau_{o1} + H(u-\theta_o)\tau_{o2}\\
v_\infty &= \begin{cases} 1, & \text{if } u \le \theta_v^-\\ 0, & \text{if } u \ge \theta_v^- \end{cases}\\
w_\infty &= (1 - H(u-\theta_o))\left(1 - \frac{u}{\tau_{w\infty}}\right) + H(u-\theta_o)w_\infty^*.
\end{aligned} \tag{3}$$

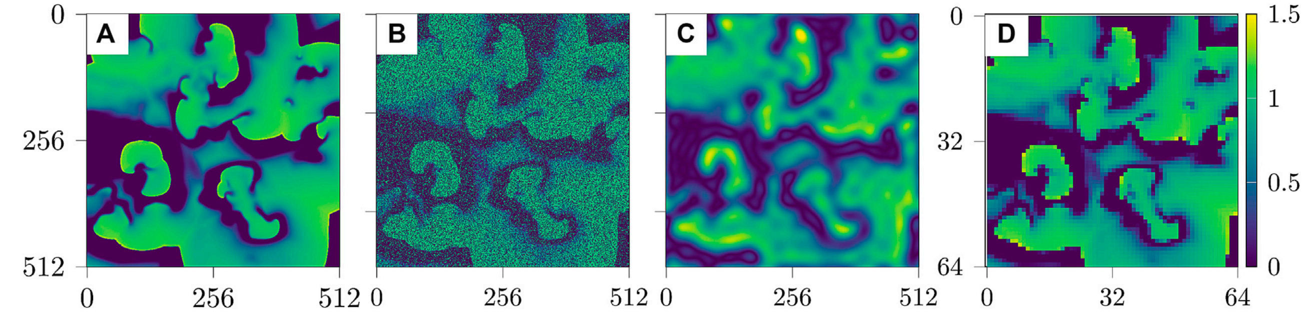


FIGURE 2 | Snapshots of the three cases of impaired data based on *u* [from left to right: **(A)** reference data *u*, **(B)** noisy data, **(C)** blurred data, and **(D)** undersampled data].

For simulating the dynamics we used the set of parameters given in **Table 1** for which the BOCF model was found [22] to exhibit excitation wave dynamics similar to the *Ten Tusscher–Noble–Noble–Panfilov* (TNNP) model [24] describing human heart tissue.

Typical snapshots of the four variables during a chaotic evolution are shown in **Figure 1**. The spatio-temporal chaotic dynamics of this system is actually transient chaos whose lifetime grows exponentially with system size [25, 26]. To obtain chaotic dynamics with a sufficiently long lifetime the system has been simulated on a domain of 512×512 grid points with a grid constant of $\Delta x = 1.0$ space units and a diffusion constant $D = 0.2$. Furthermore, an explicit Euler stepping in time with $\Delta t = 0.1$, a 5 point approximation of the Laplace operator, and no-flux boundary conditions were used for solving the PDEs.

2.1.2 Reconstruction Tasks

Experimental measurements of the dynamics of a system of interest often allow only the observation of some state variables (e.g., the membrane voltage) and may provide only incomplete or distorted information about the measured observable. Typical limitations are (additive) measurement noise and low-spatial resolution (due to the experimental conditions and/or the available hardware). Formally, measurements impaired due to noise, blurring or undersampling can be described as follows: Let $\mathbf{X}_n \in \mathbb{R}^{r\times c}$ be the measured data (here: snapshots of the field u) where r and c specify the two spatial dimensions. Each sample $\mathbf{X}_n$ with $n = 1, \ldots, N$ corresponds to a true system output $\mathbf{X}_n' \in \mathbb{R}^{r'\times c'}$ that is assumed to be known only during the training phase in terms of a training set $\mathcal{D} = \{\mathbf{Z}_1 = (\mathbf{X}_1, \mathbf{X}_1'), \ldots, \mathbf{Z}_N = (\mathbf{X}_N, \mathbf{X}_N')\}$. Note that with coarse graining $r \le r'$ and $c \le c'$. The task is to predict the true system output $\mathbf{X}'$ from impaired observations $\mathbf{X}$ which belong to one of the following three cases:

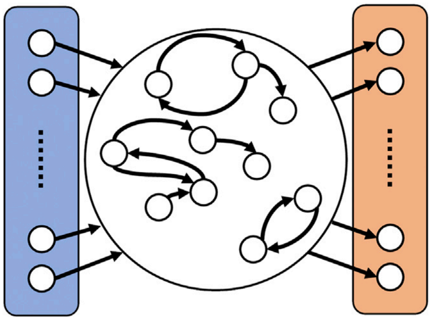


FIGURE 4 | Schematic representation of an ESN. On the left side (colored in blue) is the input layer where the input signal $\vec{u}_n$ and a constant bias b_{in} are fed in. The reservoir is represented as the large circle in the middle, where the small circles are the nodes. The output layer on the right (colored in orange) provides the reservoir signals $\vec{s}_n$ that are part of the vector $\vec{x}_n = [b_{out}; \vec{s}_n; \vec{u}_n]$ used for computing the output $\vec{y}_n = \mathbf{W}_{\text{out}}\vec{x}_n$.

1. *Noisy data*: To add noise each element of $\mathbf{X}'$ is replaced with probability p by 0 or 1 drawn from a Bernoulli distribution $\mathcal{B}(0.5)$ (note that in our case $\mathbf{X}'$ is given by the variable u of the BOCF model which has a range of $[0, 1]$). To simulate different levels of noise different probabilities $p = 0.1, 0.2, \ldots, 0.9$ are used to generate noisy data sets $\{\mathbf{X}_n\}$. In the following p is called the noise level.
2. *Blurred data*: Date with reduced spatial resolution are obtained as Fourier low-pass filtered data $\mathbf{X} = \mathcal{F}^{-1}(\mathcal{P}^m(\mathcal{F}(\mathbf{X}')))$ where $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the Fourier transform and its inverse, respectively, and $\mathcal{P}^m$ is a projection where frequencies outside a radius $m \in [2, 4, 8, \ldots, 18]$ (Manhattan distance) centered at frequency zero are set to zero.
3. *Undersampled data*: To generate undersampled date $\mathbf{X}'$ is down-sampled $\mathbb{R}^{r'\times c'} \to \mathbb{R}^{r\times c}$ with $r < r'$ and $c < c'$ by accessing every 2^i-th value of $\mathbf{X}'$, where $i \in [1, 7]$.

Figure 2 shows examples of the three types of impaired observations.

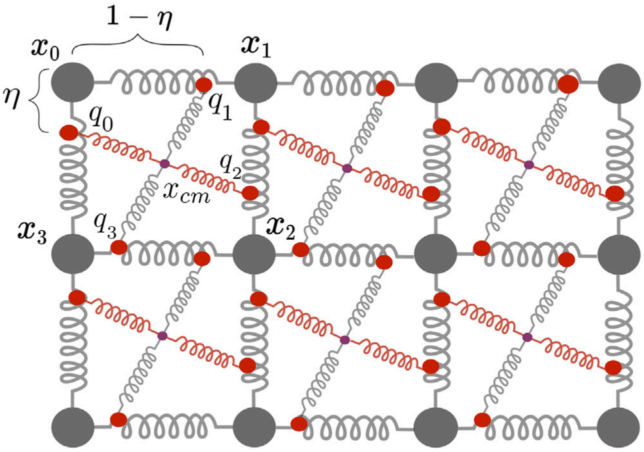


FIGURE 3 | Two dimensional mass-spring damper system with one active spring modelling fibre orientation (red) and one passive spring (gray), the centre of mass x_{cm}, the four points of attachment q_i to the structural springs and the orientation parameter η.

2.2 Predicting Electrical Excitation From Mechanical Contraction

To learn the relation between mechanical deformation and electrical excitation *inverse modelling data* were generated by a conceptual electro-mechanical model consisting of an Aliev–Panfilov model describing the electrical activity and a driven mass-spring-system [15].

2.2.1 Aliev–Panfilov Model

Specifically developed to mimic cardiac action potentials in the myocardium, the Aliev–Panfilov model is a modification of the FitzHugh–Nagumo model, which reproduces the characteristic shape of electric pulses occurring in the heart [27]. It is given by a set of two differential equations,

$$\frac{\partial u}{\partial t} = \nabla(\mathbf{D}\cdot\nabla u) - ku(u-a)(u-1) - uv \qquad (4)$$

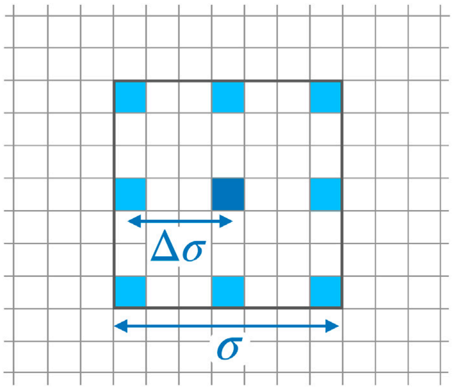


FIGURE 5 | Stencil for locally sampling data used as input of the ESN operating at the location of the dark blue pixel in the center. The stencil is characterized by its width σ and the spatial separation Δσ of sampling points.

$$\frac{\partial v}{\partial t} = \epsilon(u, v) \cdot (-v - ku(u - b - 1))$$
$$\epsilon(u, v) = \epsilon_0 + \frac{\mu_1 v}{\mu_2 + u} \tag{5}$$

in which u and v are the normalized membrane voltage and the recovery variable, respectively, and a, b and k are model parameters. The term $\nabla(\mathbf{D} \cdot \nabla u)$ accounts for the diffusion, in which the tensor $\mathbf{D}$ can be used to model anisotropies in the myocardial tissue. In addition, the term $\epsilon(u, v)$ is introduced to adjust the shape of the restitution curve by modulating the parameters μ_1 and μ_2. The computational advantage of the Aliev–Panfilov model lies in its simplicity over other ion-flow-based models which allows shorter runtimes and combined with the elastomechanical model, keeps computational costs fairly reasonable. For this reason, the Aliev–Panfilov model was chosen for generating synthetic data from complex chaotic electromechnical wave dynamics.

Within the heart muscle, the myocardium, cells contract upon electrical excitation through a passing action potential. At this point it is important to note that muscle fibre contracts along its principal orientation which has to be considered during the implementation of the mechanical part of the simulation. To couple the mechanical contraction of the muscle fibre to electrical excitation of a cell, as an extension to the Aliev–Panfilov model the *active stress* T_a was introduced by Nash and Panfilov [28] which leads to contraction in the principal orientation of the muscle fibre. The change of the active stress is described by

$$\frac{\partial T_a}{\partial t} = \epsilon_T(u) \cdot (k_T u - T_a), \tag{6}$$

where k_T controls the strength of the build-up of active stress. The term $\epsilon_T(u)$ regulates the influence of u on T_a for large u. In our simulations we use a smooth function introduced by Göktepe and Kuhl [29] given by

$$\epsilon_T(u) = \epsilon_{T,0} + (\epsilon_\infty - \epsilon_{T,0}) \cdot \exp(-\exp(-\xi_T \cdot (u - u_0))). \tag{7}$$

Here, ξ_T controls the steepness of the transition between ϵ_∞ and $\epsilon_{T,0}$ and u_0 denotes the potential threshold for the activation of the active stress, with $\epsilon_\infty < \epsilon_{T,0}$ to achieve a physiological time course [30].

2.2.2 Mass-Spring Damper System

The elasto-mechanical properties of the cardiac muscle fibre were implemented using a modified two-dimensional mass-spring damper system [31]. For the current study the mass-spring system was implemented in two dimensions because this allows shorter runtimes and primarily serves as a proof-of-principle for the evaluated reconstruction approach. In its two-dimensional form this mechanical model might correspond best to a cut-out of the atrium's wall, since there the muscle tissue is less than 4 mm thick. However, this mass-spring system can easily be expanded to three dimensions (see [15]).

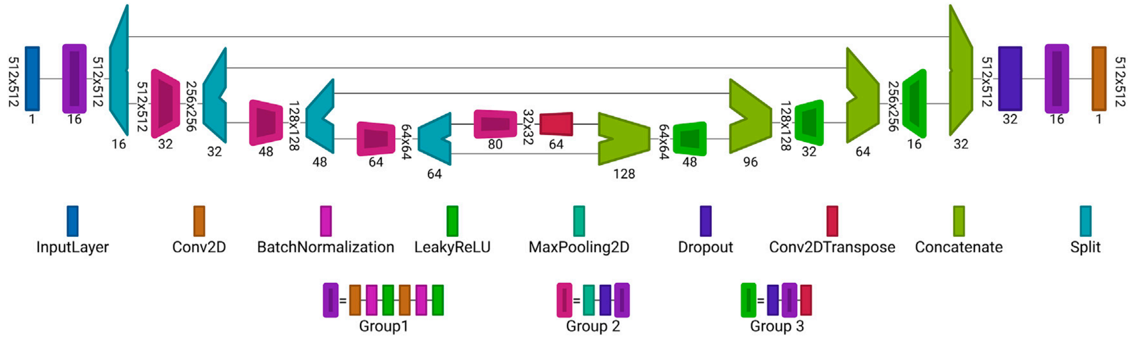


FIGURE 6 | Proposed autoencoder architecture for reconstruction of data from noisy or blurred input. Each block is a set of layers. The values written vertically describe the dimension of the input for each layer, e.g., for noisy and blurred data $r = 512$, $c = 512$ and for the inverse modelling data $r = 100$, $c = 100$. The horizontally written values at the layers are the number of channels or number of filters. Group 1 is an combination of layers, consisting of: Conv2D, BatchNormlization, LeakyReLU, Conv2D, BatchNormlization and LeakyReLU layers. Group 2 is an extension of Group 1 where a MaxPooling2D and Dropout layer are placed before Group 1. Similar applies to Group 3, it consists of a Dropout layer followed by the layers from Group 1 and finalized by a Conv2DTranspose layer follows. The architecture was visualized with *Net2Vis* [55].

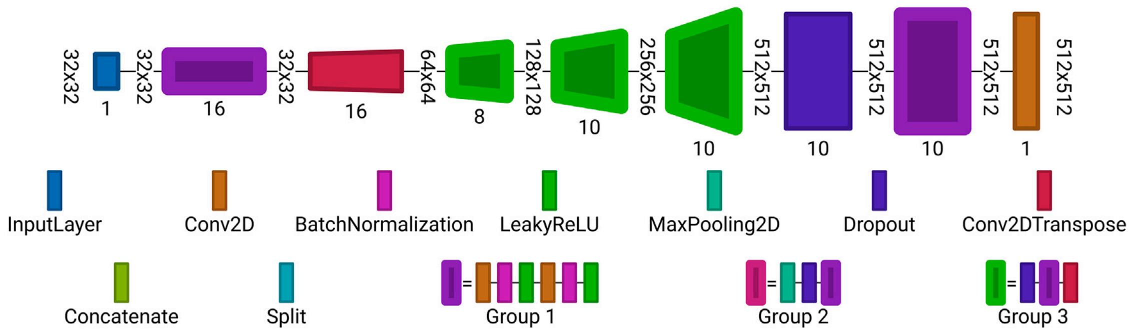


FIGURE 7 | Autoencoder architecture used for reconstruction from undersampled observations. Each block is a set of layers. The layer labeling is the same as in **Figure 6**. Visualized with *Net2Vis* [55].

TABLE 2 | The examined set of hyperparameters σ and Δσ for the local states.

σ	Δσ	σ	Δσ	σ	Δσ	σ	Δσ	σ	Δσ	σ	Δσ	σ	Δσ	σ	Δσ
25	2	29	2	33	2	37	2	41	2	45	2	49	4	101	10
25	4	29	7	33	8	37	4	41	4	45	4	49	8	101	20
25	8	29	14	33	16	37	9	41	8	45	11	49	16	101	25
25	24	29	28	33	32	37	12	41	20	45	22	49	24	101	50
						37	18	41	40	45	44	49	48		
						37	36								

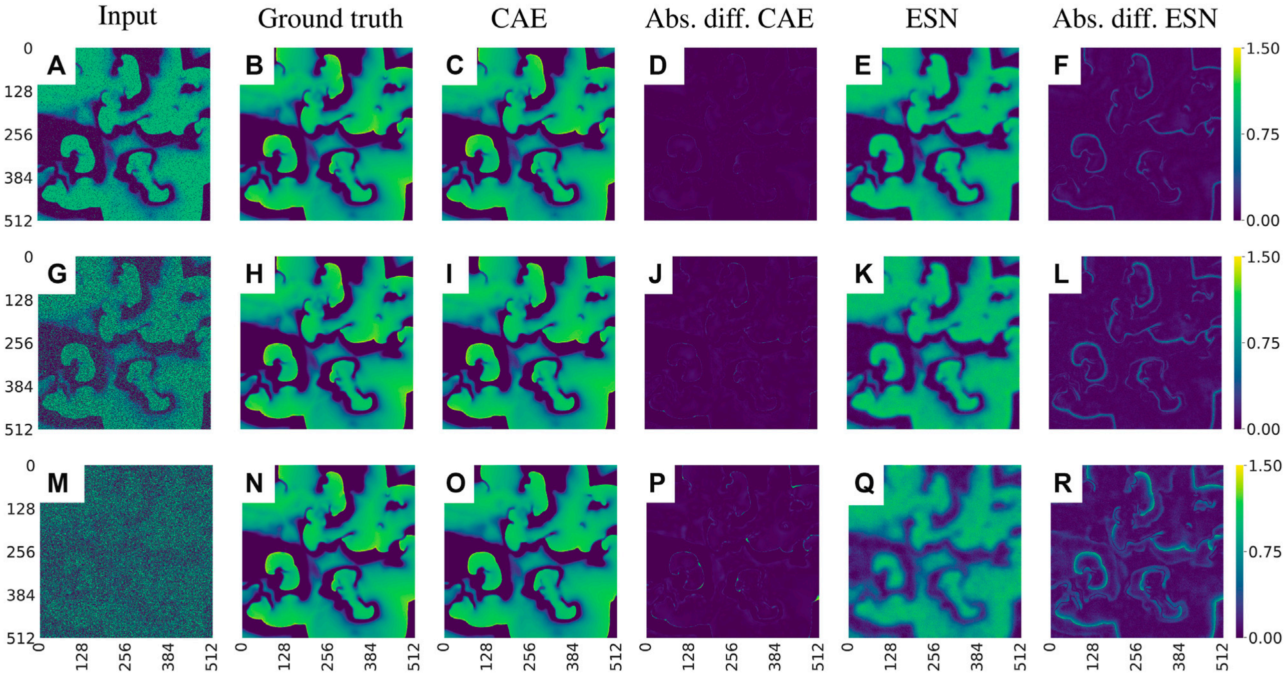


FIGURE 8 | Exemplary visualization of the input and output for both networks for data with different noise levels p: **(A)–(F)** $p = 0.1$, **(G)–(L)** $p = 0.5$, and **(M)–(R)** $p = 0.9$. Comparing the absolute differences between the prediction and the ground truth [**(D), (J), (P)** for the CAE and **(F), (L), (R)** for the ESN] one can see that the CAE is less sensitive to noise. Note that the errors develop primarily on the fronts of the waves.

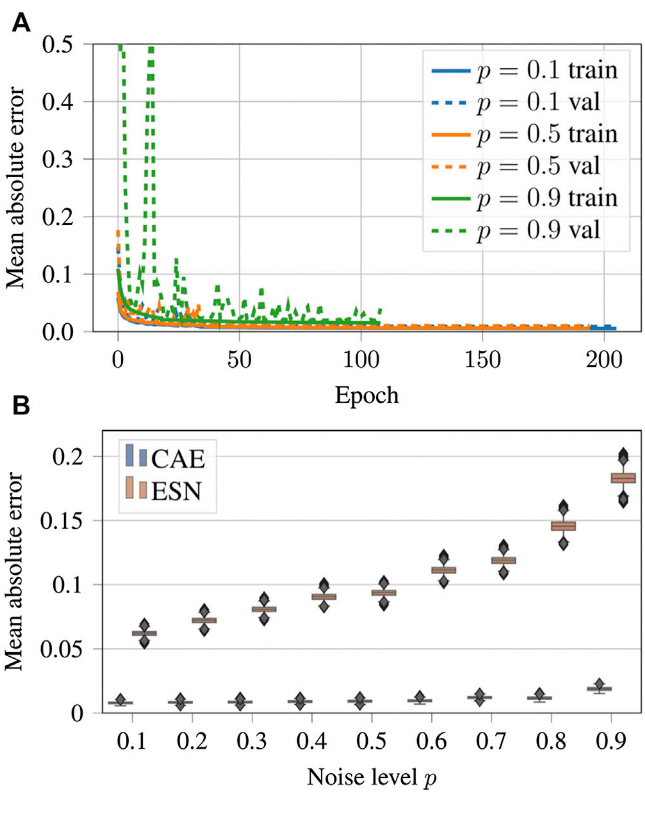


FIGURE 9 | (A) Evolution of the loss function values over the epochs for noisy input data generated with noise levels $p = 0.1$, $p = 0.5$, and $p = 0.9$ (compare **Figure 8**). It can be seen that the training always ran up to the point where early stopping, as defined in **Section 3.1**, terminated it. The solid lines are the values of the loss function during training on the training data, while the dotted lines are the values of the loss function obtained when the trained model is applied to the validation data. One epoch trained approximately 110 s on a GTX 1080 Ti. **(B)** Comparison between CAE and ESN performance with noisy input data showing boxplots of mean absolute errors (18) for different noise levels $p \in [0.1, 0.2, \ldots, 0.9]$. Each discrete value on the x-axis is assigned to the boxes of the CAE and ESN, where the ESN boxplots are colored in orange and the CNN boxplots are colored in blue. Note that for better visibility the CAE boxes and the ESN boxes a slightly shifted to the left and to the right, respectively. A tabular overview of the values can be found in **Table 3**).

Placed on a regular lattice, one mechanical cell is made up of four particles x_i at the corners connected by *structural springs* and two sets of orthogonal springs connecting the centre of mass $\vec{x}_{\mathrm{cm}}$ to each side of the cell (see **Figure 3**). The springs in the middle of the cell are called *axial springs*, of which one is made to be active (red). Here it is important to point out that one cell in the electrical model corresponds to one cell in the mechanical mass-spring system. For setting the fibre orientation through the active axial spring, the orientation parameter $\eta \in [0, 1]$ has been introduced, with which the four points of attachment q_i can be computed easily. This parameter can be set individually for each cell, so that various fibre orientations can be modelled.

Using $\vec{x}_{\mathrm{cm}} = \frac{1}{4}\sum_{i=0}^{3}\vec{x}_i$ the forces from the passive spring $\vec{f}_j$ and the active spring $\vec{f}_a$ are obtained as.

$$\vec{f}_j = -k_j\left(\|\vec{q}_j - \vec{x}_{\mathrm{cm}}\| - l_{j,0}\right)\cdot\vec{e}_j, \tag{8}$$

$$\vec{f}_a = -k_a\left(\|\vec{q}_a - \vec{x}_{\mathrm{cm}}\| - \frac{l_{a,0}}{1 + c_a \cdot T_a}\right)\cdot\vec{e}_a. \tag{9}$$

Here $l_{j,0}$, k_j and $l_{a,0}$, k_a denote the resting lengths and spring constants of the passive and active spring, respectively. From **Eq. (9)** it can be seen that, upon a rise in active stress T_a from **Eq. (6)**, the active spring contracts and an inward force is generated. The parameter c_a represents a scaling factor to modulate the influence of the active stress. Through the orientation parameter the forces from the active and passive spring can be redistributed to the corresponding particles at the corners. For example for $\vec{q}_1$, the force on x_0 would be $\vec{f}_0 = \eta\vec{f}_{q_1}$ and on x_1 it would amount to $\vec{f}_1 = (1-\eta)\vec{f}_{q_1}$.

In addition, the mechanical grid is held together by structural forces between the corner particles, which can be computed using

$$\vec{f}_{ij} = -k_{ij}\left(\|\vec{x}_i - \vec{x}_j\| - l_{ij}\right)\cdot\vec{e}_{ij}, \tag{10}$$

$$\vec{f}_{ji} = -\vec{f}_{ij}, \tag{11}$$

with l_{ij} being the resting length between particle x_i and x_j.

Finally, with all the above forces acting on particle x_i with mass m_i, its motion is determined according to

$$m_i\frac{\mathrm{d}^2\vec{x}_i}{\mathrm{d}t^2} = \sum_{\{a\}\{j\}\{ij\}}\vec{f_k} - \nu\frac{\mathrm{d}\vec{x}_i}{\mathrm{d}t}, \tag{12}$$

with the sum $\sum_{\{a\}\{j\}\{ij\}}\vec{f_k}$ of all relevant springs pulling or pushing the particle. The damping constant ν sets the strength of the damping to increase the stability of the mechanical system as a whole.

The area of each cell was calculated with a simple formula for a general quadrilateral using the positions of its four corners. As a measure of contraction, the relative change of area

$$\Delta A(t) = \frac{A(t)}{A_{\mathrm{undeformed}}} - 1 \tag{13}$$

has been used. The numerical algorithm for solving the full set of electro-mechanical ODEs is summarized in the Appendix.

2.2.3 Reconstruction Task

The inverse modelling data are generated by forward modelling $M: u \mapsto \Delta A$ using the output of **Equations (4)** and **(13)**. The task is to train an ESN or CAE to approximate $M^{-1}: \Delta A \mapsto u$. To fulfill this task we use the membrane voltages and the local deformations at all $r \times c$ grid points sampled at times t_n. The training data set $\mathcal{D} = \{\mathbf{Z}_1 = (\mathbf{X}_1, \mathbf{X}_1'), \ldots, \mathbf{Z}_N = (\mathbf{X}_N, \mathbf{X}_N')\}$ thus consists of snapshots $\mathbf{X}_n \in \mathbb{R}^{r \times c}$ and $\mathbf{X}_n' \in \mathbb{R}^{r \times c}$ of the relative mechanical deformation $\Delta A(t_n)$ and the membrane voltage $u(t_n)$, respectively, and we aim at approximating $M^{-1}: \mathbf{X}_n \mapsto \mathbf{X}_n'$ with $r, c = 100$.

TABLE 3 | Comparison of the MAE obtained when applying the CAE method and the ESN method to the test data set.

CAE (MAE ± STD)

Case	Blurred data	Noisy data	Undersampled data
1	0.01644 ± 0.00136	0.00794 ± 0.00096	0.00432 ± 0.00020
2	0.02076 ± 0.00170	0.00835 ± 0.00097	0.00782 ± 0.00053
3	0.02667 ± 0.00227	0.00856 ± 0.00104	0.01613 ± 0.00119
4	0.03450 ± 0.00318	0.00900 ± 0.00096	0.04727 ± 0.00393
5	0.04532 ± 0.00407	0.00919 ± 0.00099	0.12190 ± 0.01061
6	0.06137 ± 0.00585	0.00961 ± 0.00110	0.27821 ± 0.02823
7	0.08913 ± 0.00864	0.01210 ± 0.00103	0.42401 ± 0.02800
8	0.14018 ± 0.01261	0.01156 ± 0.00120	-
9	0.24689 ± 0.01898	0.01873 ± 0.00136	-
10	0.37214 ± 0.02408	-	-

ESN (MAE ± STD)

Case	Blurred data	Noisy data	Undersampled data
1	0.05220 ± 0.00347	0.06193 ± 0.00264	0.00362 ± 0.00031
2	0.05910 ± 0.00365	0.07193 ± 0.00288	0.01682 ± 0.00110
3	0.06245 ± 0.00394	0.08070 ± 0.00299	0.03516 ± 0.00242
4	0.07318 ± 0.00469	0.09052 ± 0.00312	0.08491 ± 0.00561
5	0.08476 ± 0.00536	0.09344 ± 0.00341	0.20439 ± 0.01105
6	0.09959 ± 0.00644	0.11136 ± 0.00370	n.A.
7	0.12325 ± 0.00813	0.11889 ± 0.00391	n.A.
8	0.18129 ± 0.01323	0.14548 ± 0.00596	-
9	0.27925 ± 0.01628	0.18259 ± 0.00720	-
10	0.39927 ± 0.01717	-	-

2.3 Machine Learning Methods

In this section we will introduce the two machine learning approaches, echo state networks (ESN) [32] and convolutional autoencoders (CAE) [33], that will be applied to solve the reconstruction tasks defined in **Section 2.1.2**.

2.3.1 Echo State Network

Echo state networks have been introduced in 2001 by Jaeger [32] as a simplified type of recurrent neural network, in which the weights describing the strength of the connections within the network are fixed. In its general composition an ESN subdivides into three sections [32], as illustrated in **Figure 4**. First of all, there is the input layer into which the input signal $\vec{u}_n \in \mathbb{R}^{N_u}$ and a constant bias b_{in} are fed. Secondly, the intermediate reservoir consists of N nonlinear units and its state is given by $\vec{s}_n \in \mathbb{R}^N$. And lastly, the output layer provides the output signal $\vec{y}_n \in \mathbb{R}^{N_y}$. Here, n denotes the discrete time steps $n = 1, \ldots, T$.

The concatenated bias-input vector $[b_{in}; \vec{u}_n]$ is fed into the reservoir through the input matrix $\mathbf{W}_{in} \in \mathbb{R}^{N\times(1+N_u)}$. Inside the reservoir connections are given by the weight matrix $\mathbf{W} \in \mathbb{R}^{N\times N}$, where N is the reservoir size. Together with the input matrix it is possible to determine the state of the reservoir at time n through the update rule

$$\vec{s}_n = (1-\alpha)\vec{s}_{n-1} + \alpha f_{in}\left(\mathbf{W}_{in}\left[b_{in}; \vec{u}_n\right] + \mathbf{W}\vec{s}_{n-1}\right), \tag{14}$$

in which $[\cdot;\cdot]$ denotes a concatenated vector. The input bias b_{in}, as well as the later introduced output bias b_{out} were both set to 1 in the following. The parameter $\alpha \in (0, 1]$ in **Eq. (14)** represents the *leaking rate* which controls how much of a neuron's activation is carried over to the next time step and can be used as a parameter to enhance predictions. As for the transfer function $f_{in}(\cdot)$ we use $\tanh(\cdot)$ and the network dynamics used has no feedback loop. Only the weights $\mathbf{W}_{out}$ providing the output signal

$$\vec{y}_n = \mathbf{W}_{out}\vec{x}_n \quad \text{with} \quad \vec{x}_n = \left[b_{out}; \vec{s}_n; \vec{u}_n\right] \tag{15}$$

are adapted during the training process by minimizing the cost function [34]

$$C(\mathbf{W}_{out}) = \sum_n \left\|\vec{y}_n^{\,true} - \mathbf{W}_{out}\vec{x}_n\right\|^2 + \lambda \mathrm{Tr}\left(\mathbf{W}_{out}\mathbf{W}_{out}^T\right) \tag{16}$$

where Tr denotes the trace of a matrix and λ controls the impact of the regularization term that prevents overfitting [35]. The final output matrix is given by the minimum of the cost function at $\mathbf{W}_{out} = \mathbf{Y}\mathbf{X}^{\mathrm{T}}(\mathbf{X}\mathbf{X}^{\mathrm{T}} + \lambda\mathbb{1})^{-1}$ where $\mathbf{X}$ and $\mathbf{Y}$ are matrices whose columns are given by the vectors $\vec{x}_n$ and $\vec{y}_n^{\,true}$, respectively.

Both matrices $\mathbf{W}_{in}$ and $\mathbf{W}$, are initialized with random values from the interval $[-0.5, 0.5]$. Since in experiments it turned out that more diverse dynamics could be modelled using networks in which only a small percentage ϵ of weights inside the reservoir remained non-zero [32], the weight matrix $\mathbf{W}$ is made sparse with only a portion ϵ of its values remaining non-zero. Furthermore, it is scaled by a factor $\frac{\rho}{|\mu_{max}|}$ where $|\mu_{max}|$ denotes here the largest eigenvalue of $\mathbf{W}$ and ρ is a hyperparameter for optimizing the performance (by ensuring the so-called echo state property [36]). To reduce the probability of drawing an dysfunctional set of matrix entries the randomly generated matrices $\mathbf{W}_{in}$ and $\mathbf{W}$ were selected from four different realisations. To optimize the performance of the ESN five hyperparameters ($N, \epsilon, \rho, \alpha$ and λ) are tuned.

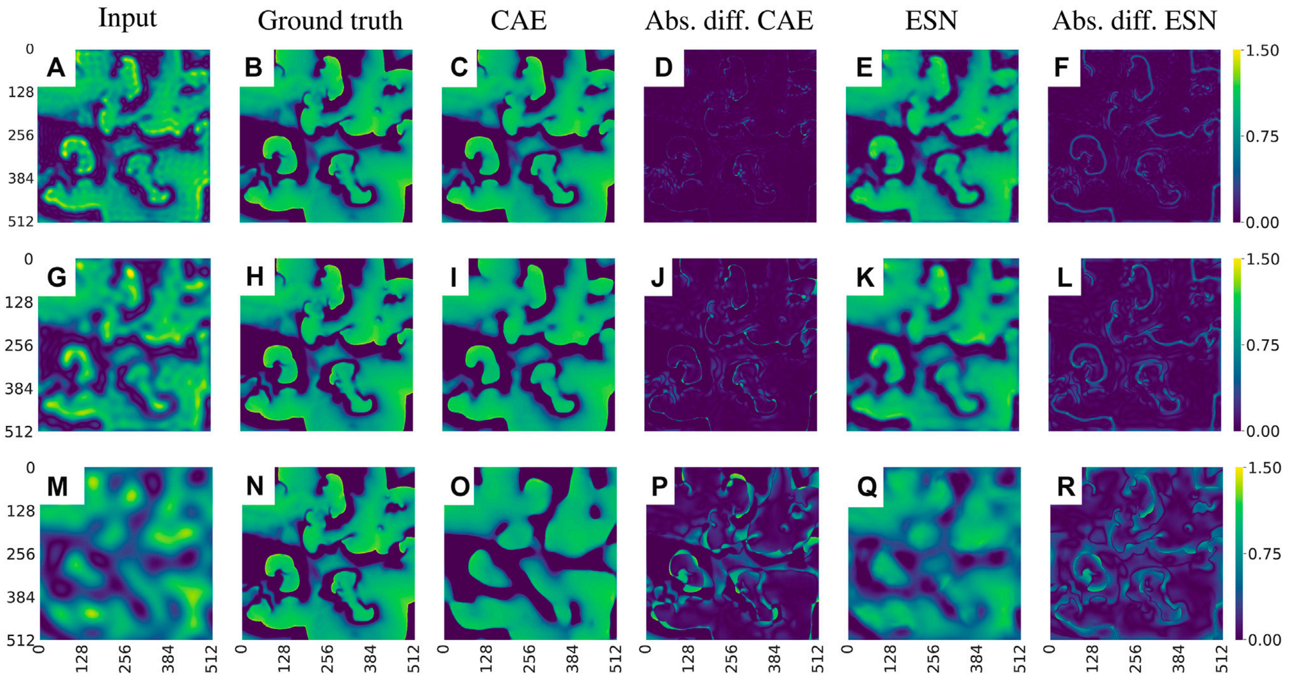


FIGURE 10 | Exemplary visualization of the input and output for both networks, CAE and ESN, when recovering the original data from blurred measurements. **(A)–(F)** corresponds to case one, where $m = 20$, **(G)–(L)** to case six with $m = 14$ and **(M)–(R)** to seven, $m = 8$. Comparing the absolute differences between the prediction and the ground truth [**(D), (J), (P)** for the CNN and **(F), (L), (R)** for the ESN] one recognizes that the CAE and ESN exhibit different patterns. The errors of the CAE are rather pointwise distributed at some locations [see **(D), (J)**] on the front while they are more evenly distributed when using ESNs **(F), (L).** This pattern is even more pronounced in **(P)** vs. **(R)**.

Reservoir computing using ESNs for predicting chaotic dynamics has already been demonstrated in 2004 by Jaeger and Haas [37]. Since then many studies appeared analyzing and optimizing this approach (see, for example [38–44], and references cited therein). In particular, it has been pointed out how reservoir computing exploits generalized synchronization of uni-directionally coupled systems [45, 46].

Recently, applications of ESNs to spatio-temporal time series have been presented [18, 47] employing many networks operating in parallel at different spatial locations based on the concept of (reconstructed) local states [48]. In particular, using this mode of reservoir computing it was possible to perform a cross-prediction between the four different variables of the BOCF model [18]. Therefore, for the current task of reconstructing data from impaired observations we build on the previous ESN design and modelling procedure. For each pixel an ESN is trained receiving input from neighboring pixels, only, representing the local state at the location of the reference pixel as illustrated in **Figure 5**. This design introduces two new hyperparameters σ and Δσ to the default ESN, where σ is the size of the stencil to define the local state and Δσ specifies the spatial distance of adjacent pixels included in the local state. Optimal values for all hyperparameters are determined by a grid search.

2.3.2 Convolutional Autoencoder

A convolutional autoencoder [33] is a special architecture of a feed forward network (FFN) with convolutional layers similar to convolutional neural networks (CNNs) [49]. Generally a CAE learns a representation of the training set $\mathcal{D}$ with the purpose of dimensionality reduction. For each pair $\mathbf{Z_i} = (\mathbf{X_i}, \mathbf{X_i'}) \in \mathcal{D}$ the CAE is trained to perform a nonlinear transformation from the input representation of $\mathbf{X}_i$ to the output representation of $\mathbf{X}_i'$. Like CNNs a CAE is a partially locally connected feed forward network, which is typically composed of the following layers:

Convolutional layers: Convolution of the input by a kernel sliding over the input. The number of rows and columns of the kernel are hyper-parameters, in this work they are set to be 3 × 3.
Batch normalization layer: Normalization of the activations of the previous layer during training and for each batch. Batch normalization allows the use of higher learning rates, being computationally more efficient, and also acts as a regularizer [50].
Leaky ReLU [51] layer: Leaky version of a rectified linear unit (ReLU) [52], such that:

$$\nu(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0. \end{cases}$$

Max pooling layer: Sample-based operation for discretization based on a kernel that slides over the input like the convolutional operator but only the maximum value of the kernel is passed to the next layer. Width and height of the kernel are hyper-parameters (in this work 2 × 2). In contrast to the convolutional layer a pooling layer is not trainable.

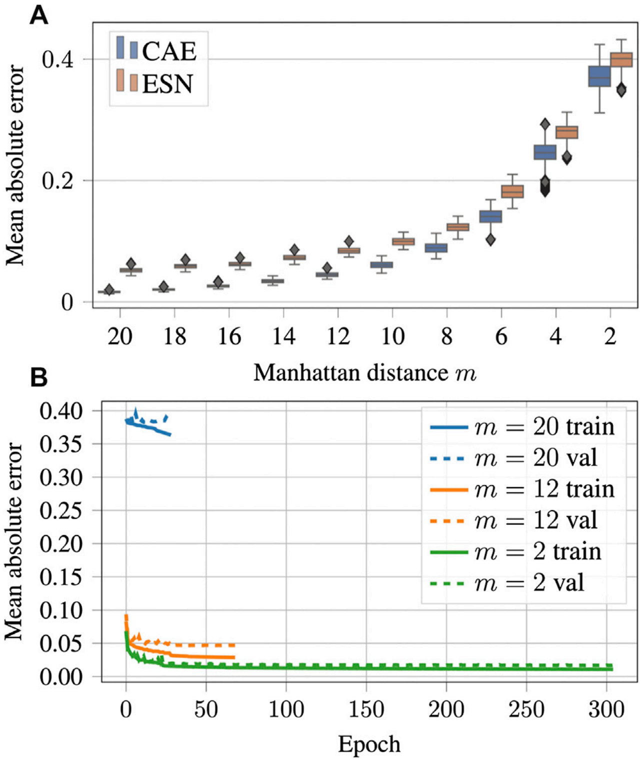


FIGURE 11 | (A) Mean absolute errors of reconstructions using CAE or ESN from data blurred with different values of the low-pass filter parameter *m*. Like in **Figure 9B** boxes are horizontally shifted for better visibility. A tabular overview of the values can be found in **Table 3**. **(B)** Evolution of the loss function values over the training epochs for blurred data. Training was always terminated by reaching the early stopping criterion, defined in **Section 3.1**. One epoch trained approximately 108 s on a GTX 1080 Ti.

Dropout layer: Regularization method to prevent overfitting where during training some weights are set randomly to zero [53]. In this work the probability of setting the weights to zero is 0.05.

The eponymous part of the CAEs are the convolutional layers, a convolution of $A = (a_{ij}) \in K^{n\times n}$ with a kernel $F = (f_{ij}) \in K^{k\times k}$, where $k < n$, is given by:

$$(A*F)_{xy} = \sum_{i=-k/2}^{k/2} \sum_{j=-k/2}^{k/2} a_{xy} f_{(i-x)\,(j-y)}, \tag{17}$$

with $x, y \in 1, \ldots, n$. If $i - x-$ or $j - y$ exceeds the range of A zero-padding is applied [54].

In this work two architectures are used. The first one employed to reconstruct the data from noisy, blurred and inverse modelling data is illustrated in **Figure 6**. The architecture is the same for the tasks in **Section 2.1.2** and **Section 2.2.3** but the sizes of $\mathbf{X}$ and $\mathbf{X}'$ are different, with $\mathbf{X}, \mathbf{X}' \in \mathbb{R}^{512\times 512}$ and $\mathbf{X}, \mathbf{X}' \in \mathbb{R}^{100\times 100}$, respectively. Due to the smaller input size, the data for the inverse modelling reconstruction is transformed into a latent space with the size of 25 × 25. The second architecture is sketched in **Figure 7** and deals with the undersampled data reconstruction.

3 RESULTS

In the following both machine learning methods will be applied to two tasks: (i) Reconstructing electrical excitation waves from noisy blurred and under sampled data (**Section 3.1**) and (ii) Predicting electrical excitation from mechanical contraction (**Section 3.2**).

3.1 Recovering Complex Spatio-Temporal Wave Patterns From Impaired Observations

To benchmark both reconstruction methods, using ESNs and CAEs, we use time series generated by the BOCF model introduced in **Section 2.1.1**. The same data were used for both methods, consisting of 5,002 samples in the training data set, 2,501 samples in the validation data set, and 2,497 samples in the test data set. The sampling time of all time series equalled $10\Delta t = 1$. We considered nine cases of noisy data (with different noise levels), ten cases of (differently) blurred data and seven examples of (spatially) undersampled time series.

For the implemention of the ESN we used the software package *easyesn* [56]. To determine the optimal ESN hyperparameters a grid search is performed as described in [18] using the training and validation subsets of the data. This search consists of two stages: first, for each combination of the local states' hyperparameters σ and $\Delta\sigma$ as listed in **Table 2** a grid search is performed to find the optimal five hyperparameters of the ESN resulting in 37 sets of optimal hyperparameters. To make these grid searches more feasible, they were performed just for a single input patch (area covered by the stencil, see **Figure 5**) in the spatial center of the training set and thus not using the full spatial data.

In the second stage, for each of the 37 sets of optimal hyperparameters determined before (for each combination of σ and $\Delta\sigma$), an ESN is trained on a larger subset of the training data and not just on a single patch. Ideally, this step should be performed on the entire spatial domain of the training set, however, as we did not notice significant differences in the results when the ESNs were trained on a spatial subset of size 250 × 250 to speed up the training process. Following the same methodology as in [18], for each pixel from this spatial subset a single ESN is trained and then the obtained output matrices $\mathbf{W}_{out}$ of these ESNs are averaged over all pixels. Compared to the procedure used in [18], the handling of boundary values has been changed. As for boundary pixels fewer adjacent pixels exist than for those inside, the creation of local states is obstructed, and boundary pixels require special treatment. In our previous work [18] individual ESNs have been trained for the boundary pixels using local states of lower dimensionality. In the following we use an alternative approach based on padding the boundary pixels by mirroring their values (motivated by the no-flux boundary conditions used). In this way, local states can be formally defined for boundary pixels in the same way as for inner pixels.

Next, the different optimal ESNs obtained for different stencils $(\sigma, \Delta\sigma)$ were evaluated by comparing their performance on the validation subset. In this way optimal values for σ and $\Delta\sigma$ were selected by choosing the combination $(\sigma, \Delta\sigma)$ with the lowest ℓ_2 difference between the prediction and ground truth on the validation set. This process yields an ESN whose

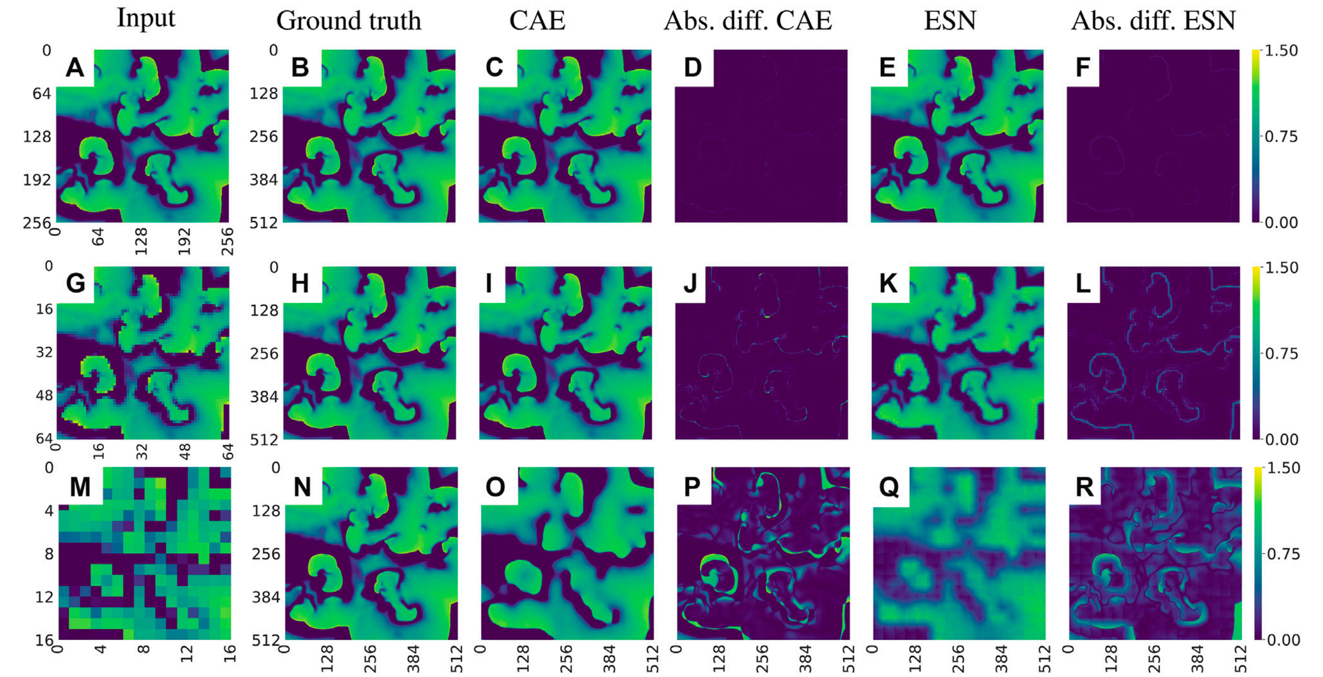


FIGURE 12 | Exemplary visualization of the reconstruction of the *u*-field of the BOCF model from undersampled data. **(A)–(F)** corresponds to a sampling parameter $i = 1$ resulting in $\mathbf{X} \in \mathbb{R}^{256\times256}$, **(G)–(L)** to case $i = 3$, $\mathbf{X} \in \mathbb{R}^{64\times64}$ and **(M)–(R)** to $i = 5$, $\mathbf{X} \in \mathbb{R}^{16\times16}$. With downsampling by a factor of 2^1 reconstructions by both networks, CAEs and ESNs, are both very successful and the absolute differences shown in **(D)** and **(F)** are nearly zero. Similar to reconstructions from noisy or blurred data errors occur mainly at the fronts of the waves, and reconstruction errors of the CAE appear to be more localized compared to ESN results.

hyperparameters and weights are optimized to yield minimal ℓ_2 error. Finally, without training the network again on the entire training set, the optimal ESN found before is used to perform the prediction on the entire test set. As a pre-processing step, both the input and target data of the training, validation and test set are rescaled with min-max scaling, where the minimal and maximal value are determined over all pixels of the training set.

The CAE was trained using the ADAM optimizer [57], implemented with *TensorFlow* [58] in version 2.3, with early stopping when the validation loss has not improved at least by 10^{-6} for 20 epochs. The learning rate was reduced by a factor of 0.2 when the loss metric stopped improving at least by 10^{-5} for ten epochs. Dropout was set to be 0.05 in all cases. As loss function the mean absolute error (MAE) was chosen:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|\widehat{\mathbf{X}}_i - \mathbf{X}_i'\right|, \tag{18}$$

where N is the number of elements in the data set, $\widehat{\mathbf{X}}_i$ the network output, $\mathbf{X}_i'$ desired ground-truth and $|.|$ stands for the absolute values.

3.1.1 Noisy Data

Figure 8 shows snapshots of the noisy input data, the corresponding ground truth, the outputs provided by the CAE and the ESN, respectively, and the absolute values of their prediction errors with respect to the ground truth.

The evolution of the loss function during the training epochs is shown in **Figure 9A**. In all cases the error decreases and the training converges, but the duration of the training depends on the complexity of the case.

Figures 9B shows a comparison of the performance of the CAE and the ESN for noisy data with nine different noise levels $p = 0.1, \ldots, 0.9$. While the mean absolute error of the CAE remains below 0.02, the reconstruction error of the ESN increases from 0.06 for $p = 0.1$ to 0.18 for $p = 0.9$, the associated ESN hyperparameter can be found in **Table 4**.

3.1.2 Blurred Data

To evaluate the performance of CAE and ESN for recovering full resolution (ground truth) data from blurred observations we consider nine cases where the radius m of Fourier low-pass filtering ranges from $m = 2$ to $m = 18$ (in steps of 2). **Figure 10** shows snapshots of reconstructions of the u-variable of the BOCF model using CAE and ESN with filter parameters $m = 20$ (A-F), $m = 14$ (G-L), and $m = 8$ (M-R). Similar to **Figure 8** the errors are largest at fronts of the excitation waves, but in contrast to noisy images the performances of CAE and ESN differ not much for blurred data. This observation is also confirmed by a systematic comparison of the mean absolute errors of both methods for different manhatten distances m given in **Figure 11A**. The errors decrease with m because the larger m the less blurred are the input data of the CAE or ESN (for hyperparameter see **Table 5**). **Figure 11B** shows the evolution of the loss function during training of the CAE.

TABLE 4 | Selected ESN hyperparameters for the case of noisy data.

ESN hyperparameters for the case of noisy data					
Case	**N**	**ρ**	**α**	**ε**	**L2 regularisation**
1	500	1.25	0.5	0.05	10
2	500	0.50	0.2	0.05	10
3	250	1.00	0.2	0.2	10
4	250	0.05	0.2	0.2	10
5	250	3.00	0.2	0.1	10
6	250	1.25	0.2	0.1	10
7	250	3.00	0.2	0.05	10
8	500	1.25	0.05	0.2	10
9	500	3.00	0.05	0.05	10

TABLE 5 | Selected ESN hyperparameters for the case of blurred data.

ESN hyperparameters for the case of noisy data					
Case	**N**	**ρ**	**α**	**ε**	**L2 regularisation**
1	250	0.30	0.01	0.1	10
2	250	1.25	0.05	0.05	10
3	500	2.00	0.05	0.1	10
4	500	0.75	0.2	0.2	5
5	250	1.50	0.2	0.2	10
6	250	1.50	0.2	0.2	10
7	250	1.25	0.2	0.2	10
8	500	0.90	0.2	0.2	5
9	250	1.50	0.5	0.2	10
10	500	0.75	0.7	0.2	10

3.1.3 Undersampled Data

Figure 12 shows examples of data reconstructed from undersampled data. In total we considered seven cases of undersampling by 2^i pixels, where i ranges from $i = 1$ to $i = 7$. For $i = 1$ input images have a resolution of $\mathbf{X} \in \mathbb{R}^{256\times256}$ and for $i = 7$, $\mathbf{X} \in \mathbb{R}^{4\times4}$. In all cases the desired output (ground truth) $\mathbf{X}'$ has a size of 512 × 512 pixels. The used hyperparameter for the ESN can be found in **Table 6**). **Figure 13B** shows the evolution of the corresponding loss function.

3.2 Predicting Electrical Excitation From Mechanical Contraction

Echo state networks as well as convolutional autoencoders have been trained with time series generated by the electromechanical model introduced in **Section 2.2** to predict the membrane voltage $u(x,t)$ **Eq. (4)** from the local contraction $\Delta A(x,t)$ given in **Eq. (13)**. The sampling time of all time series equalled $6\Delta t = 0.48$. Since for periodically rotating spiral waves this cross-prediction task is quite straightforward we focus here on the much more demanding case of an example exhibiting spatio-temporal chaos (corresponding to atrial or ventricular fibrillation). **Figure 14** shows at three instants of time snapshots of the observed contraction ΔA (first column), the ground truth of the voltage u (second column), the prediction of the CAE u_{CAE} and its absolute error $|u - u_{\mathrm{CAE}}|$ (third and fourth column) and the prediction of the ESN u_{ESN} and the corresponding absolute error $|u - u_{\mathrm{ESN}}|$ (columns five and six, respectively). Both, the ESN and the CAE were trained and tested with the same spatio-temporal time series (lengths of training, validation and test sets are 15,000, 2000 and 2000 samples, respectively). Hyperparameters of the ESN are $N = 600, \alpha = 0.5, \rho = 1.1, \epsilon = 0.05, \lambda = 5 \cdot 10^{-3}, \sigma = 7$ and $\Delta\sigma = 1$. To make the ESN more robust normally distributed noise with zero mean and a variance of 10^{-4} was added to the arguments of the activation function.

TABLE 6 | Selected ESN hyperparameters for the case of undersampled data.

ESN hyperparameters for the case of noisy data					
Case	**N**	**ρ**	**α**	**ε**	**L2 regularisation**
1	250	0.30	0.01	0.1	5
2	500	0.05	0.05	0.2	0.5
3	250	1.50	0.05	0.2	10
4	250	2.00	0.5	0.2	10
5	500	0.30	0.05	0.1	5
6	500	0.75	0.2	0.05	10
7	250	1.50	0.05	0.1	10

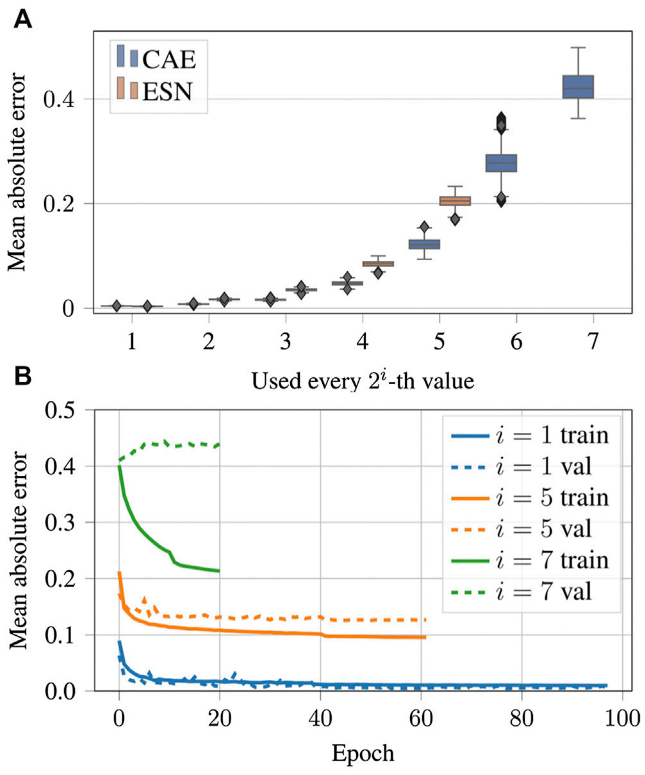


FIGURE 13 | (A) Same as **Figures 9B** but for the case of the undersampled data. A tabular overview of the values can be found in **Table 3. (B)** Evolution of the loss function values over the epochs for undersampled input data.

As illustrated in **Figure 14** both networks can solve the inverse problem and reconstruct the electrical potential field u from **Eq. (4)**. However, the reconstruction of the CAE is more precise, which is particularly noticeable at the edges of the reconstructed electrical potential field. Considering the median of the MAE over the entire test data the ESN

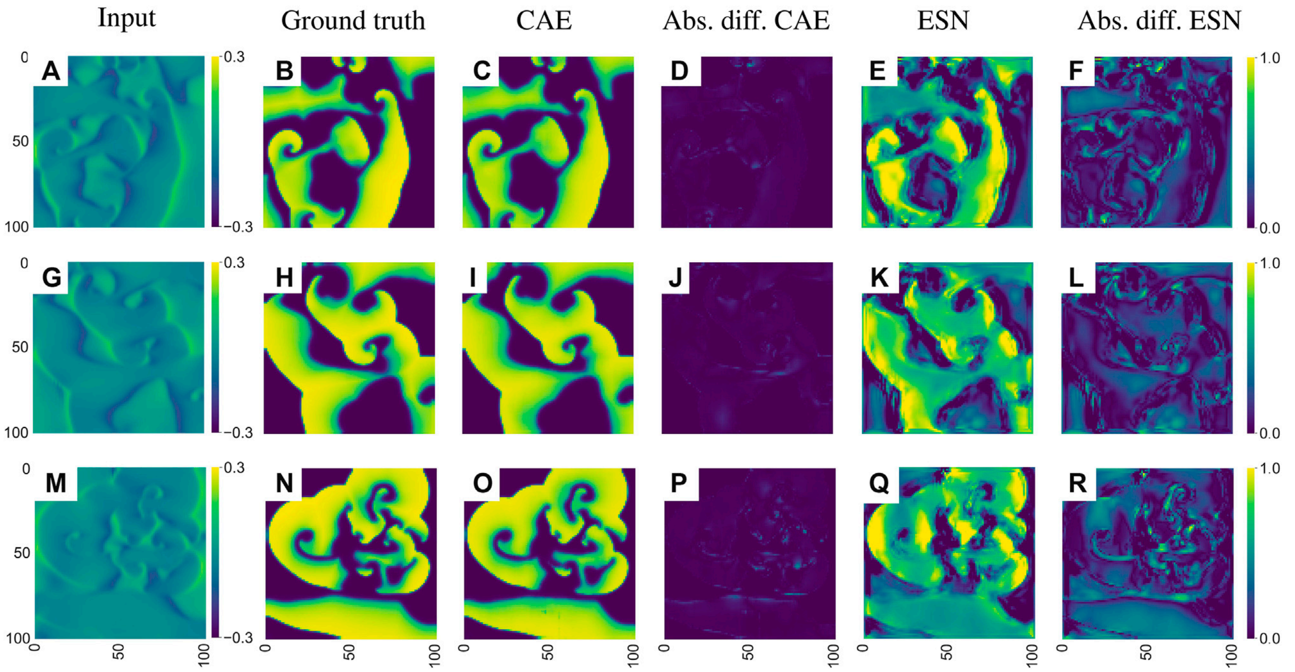


FIGURE 14 | Exemplary visualization of the input and output for both networks for the inverse reconstruction of the membrane voltage u **Eq. (4)** based on the mechanical deformation ΔA **Eq. (13)**. **(A)–(F)** corresponds to $t = 1,000$, **(G)–(L)** to $t = 1,500$ and **M-R** to $t = 2,000$ of the test data set. The input is the mechanical deformation given by **Eq. (13)** caused by the electrical potential u from **Eq. (4)**.

approach achieves an error of 0.1963 ± 0.0260 while the median error of the CAE equals 0.0164 ± 0.0028.

4 CONCLUSION

Using synthetic data generated with conceptual models describing complex cardiac dynamics we have demonstrated possible applications of machine learning methods to complete and enhance experimental observations. It was shown that echo state networks as well as convolution autoencoders provide promising results, where the latter turned out to be the method of choice in terms of more faithful reconstructions. At this point, however, we would like to stress, that we didn't try to fully optimize the algorithms employed. One could, for example, increase the size of the ESNs used or extend and refine the grid search of hyperparameters. Also with the CAE several options exist to improve the performance even more. Instead of the MAE in the loss function one could use an adaptive robust loss function [59] or the Jensen-Shannon divergence [19]. The weights of the CAE could be optimized with a stochastic gradient descend approach instead of the ADAM algorithm [57]. But we expect such modifications would show only minor improvements (if at all) [60, 61].

A comparison of the computing times with ESNs and CAEs is unfortunately not immediately possible. For the CAE the training time depends on the convergence, as illustrated in **Figures 9A**, **11B** and **13B**. In contrast to this, for ESNs training times depend strongly on the search for optical hyperparameters, especially the size of the reservoir, and this search is strongly dependent on the search space size and the number of parameters. For the task where electrical excitation is predicted from mechanical contraction our computations took 3,382 s on two Intel Xeon CPU E5-4620. While the CAE simulations have been run on GPUs the training and application of the ESN was on CPUs which makes a direct comparison difficult. Furthermore, the runtime of the programmes used is highly dependent on the libraries used and how well they have been adapted to special system architectures. In general we would estimate that in this work the effort to train and search hyperparameters for an ESN was a bit less demanding, in the sense of computational resources, compared to the training of the CAE. However, we would not consider the difference big enough to be an advantage for the ESN approach. Once trained both approaches need comparable execution times when applied to new data and executed on a CPU. In future work, using more realistic numerical simulations (and experimental data) such an optimization should be performed to achieve the best possible result for the intended medical application. Since here we used only data from conceptual models we refrained from fully optimizing the machine learning methods applied. The fact that already a straight-forward application of known algorithms and architectures provided very good results for the considered reconstruction tasks is very promising and encourages to address in future work extended tasks (including other variables, like calcium concentration, mechanical stress and strain, etc.) and reconstruction tasks with more realistic synthetic data (from 3D models, for example) combined with experimental measurements.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation. The source code will be available at https://gitlab.gwdg.de/sherzog3/reconstructing-cardiac-excitation-waves.git.

AUTHOR CONTRIBUTIONS

SH performed the CAE simulations and RZ and JA did the ESN modelling. All authors designed the study, analyzed the results, and wrote the manuscript.

FUNDING

SH acknowledges funding by the International Max Planck Research Schools of Physics of Biological and Complex Systems. UP and SL acknowledge funding by the Max Planck Society, the German Center for Cardiovascular Research (DZHK) partner site Goettingen, and the SFB 1002 Modulatory Units in Heart Failure (Project C03).

ACKNOWLEDGMENTS

SH and UP thank Florentin Wörgötter and Thomas Lilienkamp for continuous support and inspiring scientific discussions.

REFERENCES

1. Gray RA, Pertsov AM, Jalife J. Spatial and temporal organization during cardiac fibrillation. *Nature* (1998) 392:75–8. doi:10.1038/32164
2. Davidenko JM, Pertsov AV, Salomonsz R, Baxter W, Jalife J. Stationary and drifting spiral waves of excitation in isolated cardiac muscle. *Nature* (1992) 355: 349–51. doi:10.1038/355349a0
3. Witkowski FX, Leon LJ, Penkoske PA, Giles WR, Spano ML, Ditto WL, et al. Spatiotemporal evolution of ventricular fibrillation. *Nature* (1998) 392:72–82. doi:10.1038/32170
4. Efimov IR, Nikolski VP, Salama G. Optical imaging of the heart. *Circ Res* (2004) 95:21–33. doi:10.1161/01.RES.0000130529.18016.35
5. Rosenzweig S, Scholand N, Holme HCM, Uecker M. Cardiac and respiratory self-gating in radial mri using an adapted singular spectrum analysis (ssa-fary). *IEEE Trans Med Imag* (2020) 39:3029–3041.
6. Otani N, Luther S, Singh R, Gilmour R. Transmural ultrasound-based visualization of patterns of action potential wave propagation in cardiac tissue. *Ann Biomed Eng* (2010a) 38:3112–23.
7. Otani N, Luther S, Singh R, Gilmour R. *Methods and systems for functional imaging of cardiac tissueInternational Patent*. London: World Intellectual Property Organization (2010b)
8. Provost J, Lee WN, Fujikura K, Konofagou EE. Imaging the electromechanical activity of the heart *in vivo*. *Proc Natl Acad Sci USA* (2011) 108:8565–70. doi:10.1073/pnas.1011688108
9. Bers DM. Cardiac excitation-contraction coupling. *Nature* (2002) 415:198–205. doi:10.1038/415198a
10. Bourgeois EB, Bachtel AD, Huang J, Walcott GP, Rogers JM. Simultaneous optical mapping of transmembrane potential and wall motion in isolated, perfused whole hearts. *J Biomed Optic* (2011) 16:096020. doi:10.1117/1.3630115
11. Zhang H, Iijima K, Huang J, Walcott GP, Rogers JM. Optical mapping of membrane potential and Epicardial deformation in beating hearts. *Biophys J* (2016) 111:438–51. doi:10.1016/j.bpj.2016.03.043
12. Christoph J, Schröder-Schetelig J, Luther S. Electromechanical optical mapping. *Prog Biophys Mol Biol* (2017) 130:150–69. doi:10.1016/j.pbiomolbio.2017.09.015
13. Christoph J, Chebbok M, Richter C, Schroeder-Schetelig J, Stein S, Uzelac I, et al. Electromechanical vortex filaments during cardiac fibrillation. *Nature* (2018) 555:667–72. doi:10.1038/nature26001
14. Berg S, Luther S, Parlitz U. Synchronization based system identification of an extended excitable system. *Chaos: An Interdisciplinary Journal of Nonlinear Science* (2011) 21:033104. doi:10.1063/1.3613921
15. Lebert J, Christoph J. Synchronization-based reconstruction of electromechanical wave dynamics in elastic excitable media. *Chaos: An Interdisciplinary Journal of Nonlinear Science* (2019) 29:093117. doi:10.1063/1.5101041
16. Hoffman MJ, LaVigne NS, Scorse ST, Fenton FH, Cherry EM. Reconstructing three-dimensional reentrant cardiac electrical wave dynamics using data assimilation. *Chaos Interdiscipl J Nonlinear Sci* (2016) 26:013107. doi:10.1063/1.4940238
17. Hoffman MJ, Cherry EM. Sensitivity of a data-assimilation system for reconstructing three-dimensional cardiac electrical dynamics. *Phil Trans Math Phys Eng Sci* (2020) 378:20190388. doi:10.1098/rsta.2019.0388
18. Zimmermann RS, Parlitz U. Observing spatio-temporal dynamics of excitable media using reservoir computing. *Chaos Interdiscipl J Nonlinear Sci* (2018) 28: 043118. doi:10.1063/1.5022276
19. Herzog S, Wörgötter F, Parlitz U. Data-driven modeling and prediction of complex spatio-temporal dynamics in excitable media. *Front Appl Math Stat* (2018) 4:60. doi:10.3389/fams.2018.00060
20. Herzog S, Wörgötter F, Parlitz U. Convolutional autoencoder and conditional random fields hybrid for predicting spatial-temporal chaos. *Chaos* (2019) 29: 123116. doi:10.1063/1.5124926
21. Christoph J, Lebert J. *Inverse mechano-electrical reconstruction of cardiac excitation wave patterns from mechanical deformation using deep learning*. Chaos (2020) 30:123134. doi:10.1063/5.0023751
22. Bueno-Orovio A, Cherry EM, Fenton FH. Minimal model for human ventricular action potentials in tissue. *J Theor Biol* (2008) 253:544–60. doi:10.1016/j.jtbi.2008.03.029
23. Clayton R, Bernus O, Cherry E, Dierckx H, Fenton F, Mirabella L, et al. Mathematical and modelling foundations, models of cardiac tissue electrophysiology: progress, challenges and open questions *Prog Biophys Mol Biol* (2011) 104:22–48. doi:10.1016/j.pbiomolbio.2010.05.008
24. Ten Tusscher K, Noble D, Noble P, Panfilov AV. A model for human ventricular tissue. *Am J Physiol Heart Circ Physiol* (2004) 286: H1573–H1589.
25. Strain MC, Greenside HS. Size-dependent transition to high-dimensional chaotic dynamics in a two-dimensional excitable medium. *Phys Rev Lett* (1998) 80:2306–9. doi:10.1103/PhysRevLett.80.2306
26. Lilienkamp T, Christoph J, Parlitz U. Features of chaotic transients in excitable media governed by spiral and scroll waves. *Phys Rev Lett* (2017) 119:054101.
27. Aliev RR, Panfilov AV. A simple two-variable model of cardiac excitation. *Chaos Solit Fractals* (1996) 7:293–301. doi:10.1016/0960-0779(95)00089-5
28. Nash MP, Panfilov AV. Electromechanical model of excitable tissue to study reentrant cardiac arrhythmias. *Prog Biophys Mol Biol* (2004) 85:501–22. doi:10.1016/j.pbiomolbio.2004.01.016
29. Göktepe S, Kuhl E. Electromechanics of the heart: a unified approach to the strongly coupled excitation–contraction problem. *Comput Mech* (2009) 45: 227–43. doi:10.1007/s00466-009-0434-z
30. Eriksson TSE, Prassl A, Plank G, Holzapfel G. Influence of myocardial fiber/sheet orientations on left ventricular mechanical contraction. *Math Mech Solid* (2013) 18:592–606. doi:10.1177/1081286513485779
31. Bourguignon D, Cani MP. *Controlling anisotropy in mass-spring systems. Eurographics*. Vienna: Springer (2000) p. 113–23. doi:10.1007/978-3-7091-6344-3_9
32. Jaeger HGMD Report. *The 'echo state' approach to analysing and training recurrent neural networks—with an erratum note*. Kaiserslautern: German National Research Institute for Computer Science (2001) p. 43.

33. Cheng Z, Sun H, Takeuchi M, Katto J. *Deep convolutional autoencoder-based lossy image compressionPicture Coding Symposium, PCS 2018—Proceedings*. London: Institute of Electrical and Electronics Engineers Inc. (2018) p. 253–7. doi:10.1109/PCS.2018.8456308
34. Lukoševičius M. *A practical guide to applying echo state networks. Lecture notes in computer science*. Berlin: Springer (2012) p. 659–86. doi:10.1007/978-3-642-35289-8_36
35. Lukoševičius M, Jaeger H. Reservoir computing approaches to recurrent neural network training. *Comput Sci Rev* (2009) 3:127–49. doi:10.1016/j.cosrev.2009.03.005
36. Yildiz IB, Jaeger H, Kiebel SJ. Re-visiting the echo state property. *Neural Netw* (2012) 35:1–9. doi:10.1016/j.neunet.2012.07.005
37. Jaeger H, Haas H. Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. *Science* (2004) 304:78–80. doi:10.1126/science.1091277
38. Lu Z, Pathak J, Hunt B, Girvan M, Brockett R, Ott E. Reservoir observers: model-free inference of unmeasured variables in chaotic systems. *Chaos Interdiscipl J Nonlinear Sci* (2017) 27:041102. doi:10.1063/1.4979665
39. Carroll TL, Pecora LM. Network structure effects in reservoir computers. *Chaos Interdiscipl J Nonlinear Sci* (2019) 29:083130. doi:10.1063/1.5097686
40. Griffith A, Pomerance A, Gauthier DJ. Forecasting chaotic systems with very low connectivity reservoir computers. *Chaos Interdiscipl J Nonlinear Sci* (2019) 29:123108. doi:10.1063/1.5120710
41. Thiede LA, Parlitz U. Gradient based hyperparameter optimization in echo state networks. *Neural Network* (2019) 115:23–9. doi:10.1016/j.neunet.2019.02.001
42. Haluszczynski A, Aumeier J, Herteux J, Rth C. Reducing network size and improving prediction stability of reservoir computing. *Chaos Interdiscipl J Nonlinear Sci* (2020) 30:063136. doi:10.1063/5.0006869
43. Carroll TL. Dimension of reservoir computers. *Chaos Interdiscipl J Nonlinear Sci* (2020) 30:013102. doi:10.1063/1.5128898
44. Fan H, Jiang J, Zhang C, Wang X, Lai YC. Long-term prediction of chaotic systems with machine learning. *Phys Rev Res* (2020) 2:012080. doi:10.1103/PhysRevResearch.2.012080
45. Parlitz U, Hornstein A. Dynamical prediction of chaotic time series. *Chaos Complex Lett* (2005) 1:135–44.
46. Lu Z, Hunt BR, Ott E. Attractor reconstruction by machine learning. *Chaos Interdiscipl J Nonlinear Sci* (2018) 28:061104. doi:10.1063/1.5039508
47. Pathak J, Hunt B, Girvan M, Lu Z, Ott E. Model-free prediction of large spatiotemporally chaotic systems from data: a reservoir computing approach. *Phys Rev Lett* (2018) 120. doi:10.1103/physrevlett.120.024102
48. Parlitz U, Merkwirth C. Prediction of spatiotemporal time series based on reconstructed local states. *Phys Rev Lett* (2000) 84:1890–3. doi:10.1103/physrevlett.84.1890
49. LeCun Y, Boser BE, Denker JS, Henderson D, Howard RE, Hubbard WE, et al. Handwritten digit recognition with a back-propagation network. In: DS Touretzky, editor. *Advances in neural information processing systems, 2* Burlington, MA: Morgan-Kaufmann (1990) p. 396–404.
50. Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. *Proc Int Conf Mach Learn* (2015) 37:448456. doi:10.1609/aaai.v33i01.33011682
51. Maas AL, Hannun AY, Ng AY. "Rectifier nonlinearities improve neural network acoustic models Speech and Language Processing." In ICML workshop on deep learning for audio (2013) p. 3.
52. Hahnloser RH, Sarpeshkar R, Mahowald MA, Douglas RJ, Seung HS. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature* (2000) 405:947.
53. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res* (2014) 15:1929–58. doi:10.1109/iwcmc.2019.8766500
54. Dumoulin V, Visin F. *A guide to convolution arithmetic for deep learning* (2016) arXiv e-printsarXiv:1603.07285
55. Bäuerle A, Ropinski T. *Net2vis: transforming deep convolutional networks into publication-ready visualizations* (2019) arXiv preprint arXiv:1902.04394.
56. Zimmermann R, Thiede L. *easyesn* (2020) Available at: https://github.com/kalekiu/easyesn.
57. Kingma DP, Ba J. *Adam: a method for stochastic optimization* (2014) arXiv e-printsarXiv:1412.6980.
58. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, et al. *TensorFlow: large-scale machine learning on heterogeneous systems* (2015) Software available from tensorflow.org.
59. Barron JT. *A general and adaptive robust loss function. Berlin*: Springer (2019)
60. Rosser JB. Nine-point difference solutions for poisson's equation. *Comput Math Appl* (1975) 1:351–60. doi:10.1016/0898-1221(75)90035-8
61. Scherer POJ. *Computational physics: simulation of classical and quantum systems*. Berlin: Springer (2010) p. 147.
62. Chugh SS, Havmoeller R, Narayanan K, Singh D, Rienstra M, Benjamin EJ, et al. Worldwide epidemiology of atrial fibrillation: a Global Burden of Disease 2010 Study. Circulation (2014) 129:837–847. doi:10.1161/CIRCULATIONAHA.113.005119
63. Wolf PA, Dawber TR, Thomas HE, Kannel WB. Epidemiologic assessment of chronic atrial fibrillation and risk of stroke: the Framingham study. Neurology (1978) 28:973–977.
64. Manolio TA, Kronmal RA, Burke GL, O'Leary DH, Price TR. Short-term predictors of incident stroke in older adults. The Cardiovascular Health Study. Stroke (1996) 27:1479–1486.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

APPENDIX 1: SOLUTION OF THE ELECTRO-MECHANICAL DYNAMICS

The differential equations from the extended Aliev-Panfilov model in **Eqs. A1–A3** have been integrated using the forward Euler method $y_i^{t+\Delta t} = y_i^t + \Delta t \cdot f(y) + \mathcal{O}(\Delta t^2)$, in which y is a place holder for the model variables u, v and T_a; $f(y)$ represents the right-hand side of the respective equation. The diffusion tensor **D** in Equation (**4**) was set to a scalar constant D and the diffusion term was approximated with a nine-point stencil [60]

$$\begin{aligned}\nabla\left(\mathbf{D} \cdot \nabla u_{ij}\right) = D \cdot \nabla^2 u_{ij} = \frac{D}{6h^2}\,&(4u_{i+1,j} + 4u_{i-1,j} \\ &+ u_{i+1,j+1} + u_{i+1,j-1} \quad + 4u_{i,j+1} + 4u_{i,j-1} \\ &+ u_{i-1,j+1} + u_{i-1,j-1} - 20u_{ij}) + O(h^4) \qquad \text{(A1)}\end{aligned}$$

where i, j are the indices of the grid points and h denotes the spacing constant between the cells.

For the excitation variable u in the electrical part of the simulation, no-flux boundary conditions have been used which were imposed by setting the two outermost cells to the same value, i.e., $u_{0,j} = u_{1,j}$. In the mechanical part of the simulation, numerical calculations were carried out according to the following scheme for each time step: (1) update of the position of the centre of mass x_{cm}, (2) calculation of all four points of attachment q_i for each cell, (3) computing of forces from structural and axial springs for each particle x_i, (4) update of the positions of all particles using the Verlet method and (5) determine change of area for each cell. Here the Verlet method refers to the standard Verlet algorithm which is given as [61]

$$\vec{x}_i^{\,t+\Delta t} = 2\vec{x}_i^{\,t} - \vec{x}_i^{\,t-\Delta t} + \frac{1}{m_i}\vec{F}_i^{\,t}\Delta t^2 + \mathcal{O}\left(\Delta t^4\right), \qquad \text{(A2)}$$

with the total force $\vec{F}_i^{\,t}$ acting on the particle. Because the total force includes the damping term, it is convenient to rewrite **Eq. A2** with $\vec{F}_i = \vec{f}_i - \nu\frac{\mathrm{d}\vec{x}_i}{\mathrm{d}t}$ to

$$\vec{x}_i^{\,t+\Delta t} = \frac{2\vec{x}_i^{\,t} - \vec{x}_i^{\,t-\Delta t}\left(1 - \frac{\nu}{2m_i}\Delta t\right) + \frac{1}{m_i}\vec{f}_i^{\,t}\Delta t^2}{1 + \frac{\nu}{2m_i}\Delta t} + \mathcal{O}\left(\Delta t^4\right). \qquad \text{(A3)}$$

A padding layer of ten electrically inactive cells was implemented outside the electrical grid to account for boundary effects in the mechanical network. In addition, the active stress variable T_a from Eq. (**6**) of the last row of cells just at the edge of the simulation grid was mirrored to the two padding layers just outside the simulation grid which proved to dramatically reduce mechanical boundary effects. This is likely due to the fact that a proper contraction of an electrically active cell is not guaranteed if one of its sides is connected to an inactive cell. Lastly, the outermost padding cell's positions were fixed to prevent the grid as a whole from moving away from its original position.

To improve numerical accuracy for each time step Δt of the electrical equations (Euler method) five time steps of size $\Delta t/5$ were computed for the mechanical system (3). All computations have been performed on a spatial grid of 100 × 100 elements. The parameter values of the dynamical equations used are given in **Table A1**.

TABLE A1 | Parameters of the electro-mechanical model.

u_0	0.1	a	0.05	b	0.05	μ_1	0.2	μ_2	0.3	k	8	ϵ_0	0.002
D	0.22	Δt	0.08	k_T	3	k_{ij}	13	k_{ij}^{pad}	23	k_j	2	k_a	9
k_a^{pad}	23	c_a	10	m_i	0.2	ν	6.86	$\epsilon_{T,0}$	1	ϵ_∞	0.1	ξ_T	30

frontiers in Systems Neuroscience

MINI REVIEW
published: 20 May 2021
doi: 10.3389/fnsys.2021.685387

Minireview of Epilepsy Detection Techniques Based on Electroencephalogram Signals

*Guangda Liu, Ruolan Xiao, Lanyu Xu and Jing Cai**

College of Instrumentation and Electrical Engineering, Jilin University, Changchun, China

Epilepsy is one of the most common neurological disorders typically characterized by recurrent and uncontrollable seizures, which seriously affects the quality of life of epilepsy patients. The effective tool utilized in the clinical diagnosis of epilepsy is the Electroencephalogram (EEG). The emergence of machine learning promotes the development of automated epilepsy detection techniques. New algorithms are continuously introduced to shorten the detection time and improve classification accuracy. This minireview summarized the latest research of epilepsy detection techniques that focused on acquiring, preprocessing, feature extraction, and classification of epileptic EEG signals. The application of seizure prediction and localization based on EEG signals in the diagnosis of epilepsy was also introduced. And then, the future development trend of epilepsy detection technology has prospected at the end of the article.

Keywords: epilepsy, neurological disorder, EEG, machine learning, detection

OPEN ACCESS

Edited by:
Thomas Hartung,
Johns Hopkins University,
United States

Reviewed by:
Olivier Darbin,
University of South Alabama,
United States
Marco Cambiaghi,
University of Verona, Italy

****Correspondence:***
Jing Cai
caijing1979@jlu.edu.cn

Received: *25 March 2021*
Accepted: *20 April 2021*
Published: *20 May 2021*

Citation:
Liu G, Xiao R, Xu L and Cai J (2021) Minireview of Epilepsy Detection Techniques Based on Electroencephalogram Signals. Front. Syst. Neurosci. 15:685387. doi: 10.3389/fnsys.2021.685387

INTRODUCTION

Epilepsy is a neurological disorder caused by the sudden abnormal discharge of brain neurons. The typical characteristics of epilepsy are recurrent, unconscious body movements, and so on (Sra et al., 2019). Uncontrollable seizures are more likely to induce Depression, Cardiovascular disease, and other diseases making patients and their families miserable (Supriya et al., 2020). The World Health Organization (WHO) report manifests that approximately 50 million people have epilepsy worldwide (Liu et al., 2020). Knowing the precursors of epilepsy can allow patients to avoid the pain of epileptic seizures through drug control, so there are an urgent need for simple, fast and effective epilepsy detection methods.

EEG is a commonly used non-invasive auxiliary method in the clinical diagnosis of epilepsy. However, it is a highly tedious, laborious, time-consuming, and costly task for neurologists to identify seizures from EEG for a long time (Yao et al., 2021). Therefore, it is necessary to develop a reliable epilepsy automatic detection system, which can significantly improve the quality of life of epilepsy patients (Solaija et al., 2018). Prior et al. (1973) introduced the Cerebral Function Monitor (CFM) that monitors the long-time EEG, which can record the number of seizures. Subsequently, Gotman (1982) selectively recorded the EEG signals of epileptic in the interictal and ictal as samples and used its amplitude, period, and other characteristics to distinguish whether the samples were in the state of epileptic seizures. Martinerie et al. (1998) realized the epileptic seizure detection by extracting the non-linear indicators of the EEG signal around seizure onset. The above work uses computers to automatically collect EEG data of patients with epilepsy and try to extract the characteristics of the lesions, but it cannot fully realize the prediction of epileptic seizures, and it is of limited help to medical staff and patients. While the emergence of machine learning promotes the development of epilepsy detection techniques (Aayesha et al., 2021), it provides the possibility to automatically detect epilepsy, thus attracting more and more researchers to join it. Chen et al. (2014) proposed an epilepsy detection framework based on machine learning to realize epileptic seizure detection. Later Craley et al. (2021) developed an end-to-end deep learning model for automatic seizure detection in multichannel EEG recording. Their outstanding research has made machine learning a step toward success in the field of automatic epilepsy detection. At present, most patients with epilepsy can be treated with drugs (Wang et al., 2021). Liu et al. (2018) and Ito et al. (2021) used EEG observing epileptic discharges to verify anti-epileptic drugs' reliability. For patients with drug-resistant epilepsy, surgical treatment, such as Temporal Lobectomy (TL) is necessary to control seizures (Anoop et al., 2021). And for patients with refractory epilepsy, Vagus nerve stimulation (VNS) has a significant therapeutic effect (Shimogawa et al., 2021). Under such circumstances, using machine learning algorithms in EEG signals to realize epileptic detection, thus realizing the treatment effect evaluation, will help clinicians treat epileptic (Assi et al., 2017; Al-Hadeethi et al., 2020).

The machine learning algorithm mainly compares the abnormal time-frequency domain characteristics of the EEG signal of patients with epileptic seizures to detect epileptic seizures. In recent years, seizure detection has also promoted the development of seizure prediction and location. This minireview introduced the decisive steps of epileptic seizure detection shown in **Figure 1**, including the acquisition, preprocessing, feature extraction, and classification of epilepsy EEG signals, besides the application of seizure prediction and localization in the diagnosis of epilepsy and trend of future seizure detection techniques was also given here.

ACQUISITION OF EEG AND PREPROCESSING

There are many devices for obtaining EEG signals, such as brain-computer interface (BCI) equipment from Neurosky and portable EEG acquisition equipment by COMPUMEDICS NeuroScan. The EEG signals can be acquired by placing EEG electrodes on the scalp of patients with epilepsy. EEG electrodes can be placed on the whole brain according to the international 10–20 EEG system for EEG electrode placement (Herwig et al., 2003).

Before detecting the collected epilepsy EEG signals, it is a regulation method to use the publicly available epilepsy EEG signal data set (e.g., epilepsy EEG dataset of Children Hospital Boston, Massachusetts Institute of Technology (CHB-MIT), epilepsy EEG dataset of The Freiburg, epilepsy EEG dataset of Bonn University) to establish and verify the EEG detection model. In addition to using publicly available datasets, some researchers used clinical epilepsy EEG data by clinicians to verify the reliability of the epilepsy detection model (Iesmantas and Alzbutas, 2020). Furthermore, some researchers evaluated the model's reliability through cross-database (Sun et al., 2019). Raghu et al. (2020) used five different epilepsy EEG datasets for the first time to verify the generalization capability of seizure detection models. What's more, the collected epilepsy data needs to undergo preprocessing, including artifacts removing and noise filtering, to obtain a clean epilepsy EEG signal for the next step, feature extraction (Acharya et al., 2018a).

FEATURE EXTRACTION

Feature extraction is an essential step in epileptic seizure detection, which is used to establish an epilepsy detection model via standard epilepsy data, and epilepsy detection from actual collected EEG signal data. The effect of feature extraction is closely related to the accuracy of epilepsy detection, so it is imperative to improve feature extraction. And research shows that different dimensionality reduction methods can improve the saliency of features. Al-Hadeethi et al. (2020) proposed for the first time to use the covariance matrix for reducing EEG signals dimensionality and extract its statistical features, and use non-parametric tests to obtain the set that has the most distinguishing features, which can be used as the input of Adaptive Boosting Least Square-Support Vector Machines (AB-LS-SVM) model to achieve satisfactory results (>99% accuracy). Vicnesh and Hagiwara (2019) extracted the non-linear features from the EEG data, selected them, and then fed them into the decision tree (DT) to classify the different epilepsy classes.

EEG signals are non-linear and non-stationary time signals. Using wavelet transform to re-express EEG signals is a commonly used method of dimensionality reduction. Ott et al. (2010) extracted the standard deviation, variance, and higher-order moments after wavelet transform and used them as the input of linear discriminant analysis (LDA) and k-nearest neighbor (KNN) classifiers. On CHB-MIT, the method yielded a classification accuracy of 99.45% using the KNN classifier. Wang et al. (2017) presented a three-class classification system based on discrete wavelet transform (DWT) and the non-linear sparse extreme learning machine (SELM). Three-level lifting DWT using Daubechies order four wavelets was introduced to decompose the Bonn University EEG dataset, and the maximum and standard deviation values of each subband were computed. The experiment obtained a classification accuracy of 98.4%. DWT was also used by Amin et al. (2020) for differentiating epileptic seizures for standard signals.

In addition to wavelet transform, the transformation of epilepsy EEG signals can also use empirical mode decomposition (EMD), wavelet packet decomposition (WPD), etc. A new method was presented by Ahmet and Aydin (2018) to analyze intrinsic mode functions (IMFs) decomposed by EMD. The method showed 97.89% accuracy by using the Bonn University EEG dataset. Ailckovic et al. (2018) applied EMD, DWT, WPD to process the Freiburg and CHB-MIT EEG dataset and extracted six statistical features, after that putting them into Random Forest (RF), Support vector machine (SVM), Multilayer perceptron (MLP), and KNN classifiers. This method could discriminate

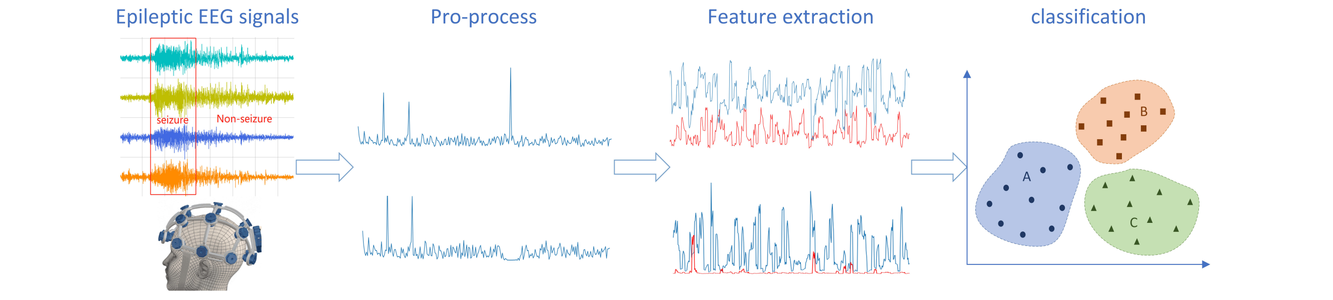


FIGURE 1 | The key steps of epilepsy detection.

between inter-ictal and pre-ictal EEG states with an accuracy of 99.70%. Hassan and Subasi (2016) proposed a new signal processing scheme for EEG signal segments, namely complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN). Six spectral moments were extracted from the CEEMDAN mode functions and then inputted into linear programming boosting (LPBoost) classifier. This model got 100% accuracy. Tao et al. (2017) applied a fusion method of variational mode decomposition (VMD) and autoregression (AR) for feature extraction. Statistical features of the best AR model coefficients were calculated and fed into RF classifier for classification. Finally, the test showed 97.352% accuracy. Another critical step after feature extraction is classification, which will give the final epilepsy detection results from the extracted features.

CLASSIFICATION

In recent years, the application of machine learning in the classification of epilepsy diagnosis has attracted more and more researchers. And some machine learning algorithms used in epilepsy detection and classification are summarized in **Table1**, mainly including SVM, convolutional neural networks (CNN), extreme learning machines (ELM), and other algorithms.

TABLE 1 | Summary of machine learning methods for epilepsy detection.

Author	Dataset	Model	Accuracy (%)
Janjarasjitt and Suparerk	CHB-MIT	SVM	96.87
Chen et al.	BONN	LS-SVM	99.5
Al-Hadeethi et al.	BONN	AB-LS-SVM	99
Qi et al.	BONN	ELM	96.5
Li et al.	BONN	M-ELM	100
Song et al.	BONN	FF-ELM-SD	97.53
Wang et al.	BONN	SELM	97.6
Acharya et al.	BONN	CNN	88.67
Wei et al.	CHB-MIT	CNN	90.57
Nogay and Adeli	CHB-MIT	DRNN	100
Choubey and Pandey	BONN	ANN + KNN	KNN:98 ANN:94
Yuan et al.	CHB-MIT	BLDA	95.74
Zeng et al.	BONN	GRP-DNet	100
Juarez-Guerra et al.	BONN	MRW-FFWNN	95.0

SVM is a commonly used classifier, and the classification results of which can be changed using different kernel functions and different cross-validation multiples. Janjarasjitt (2017) applied SVM to classify single-channel scalp EEG data features times. Moreover, the test got an average classification accuracy rate of 96.87% using 10-fold cross-validation. Besides, many researchers combine different algorithms with SVM to obtain better classification accuracy and detection efficiency. Makaram et al. (2020) extracted the time domain characteristics and signal complexity. Further, they used the Support Vector Machine-Error-Correcting Output Codes (SVM-ECOC) to train the classification algorithm, and the improvement in classification accuracy had been obtained. Ramakrishnan and Murugavel (2019) proposed a new seizure detection model using layered directed acyclic graph SVM (LDAG-SVM), which improved classification accuracy and reduced detection time compared to existing methods. After performing DWT, Chen et al. (2019) extracted the non-linear features of each sub-band and inputted them into six different classifiers for training. Finally, they increased the classification accuracy of Least Square-Support Vector Machines (LS-SVM) to 99.5%, which was better than five other classifiers. Based on LS-SVM, Al-Hadeethi et al. (2020) further applied the AB-LS-SVM model for epilepsy detection.

In 2006, Huang et al. (2006) improved Backward Propagation (BP) to improve learning efficiency, simplify learning parameters, and proposed ELM. Then Qi et al. (2011) extracted non-linear features and applied ELM for an epilepsy diagnosis. The classification accuracy was improved to 96.5%, which was better than BP and SVM in both classification accuracy and training time. To make ELM better application, Li et al. (2016) proposed a ternary classification system based on the Multiplicative Extreme Learning Machine (M-ELM), with a maximum classification accuracy of 100%. Song et al. (2016) designed a novel fusion feature and integrated the fusion feature and ELM. Experimental results demonstrated 97.35% classification accuracy. Wang et al. (2017) applied SELM for epilepsy detection. Liu et al. (2017) proposed Kernel ELM and introduced Cholesky decomposition to reduce the computation of out weights. The experimental results showed that the method can achieve an average classification accuracy of 96.5%. On this basis, Zhang et al. (2019) proposed Expectation Kernel ELM (EKELM) to further improve ELM classification abilities.

In 2018, Acharya et al. (2018b) applied CNN to the study of EEG signals for the first time and realized a 13-layer deep convolutional neural network for epilepsy detection without separate feature extraction and feature selection. The proposed technique achieved an accuracy of 88.67%. Iesmantas and Alzbutas (2020) extracted different features from clinical epilepsy EEG signals and applied CNN for training data. Wei et al. (2019) used the increasing and decreasing sequences (MIDS) merger to highlight the characteristic of waveforms and a data augmentation method for increasing the sample diversity and EEG information. Furthermore, they applied CNN classifier for epilepsy detection to get 90.57% accuracy. Nogay and Adeli (2020) proposed a machine learning method for seizure detection using the pre-trained deep two-dimensional CNN and transfer learning concept that achieved 100% accuracy for binary classification and ternary classification for epileptic seizure detection.

With the continuous development of machine learning, new algorithms are constantly being introduced into seizure detection. Akyol (2020) proposed a new deep neural network for seizure detection that successfully obtained an average accuracy of 97.17%. Choubey and Pandey (2020) used Artificial Neural Network (ANN) and KNN to achieve seizure detection. Yuan et al. (2018) applied a Bayesian linear discriminant analysis (BLDA) classifier to classify the CHB-MIT scalp EEG dataset and achieved an average classification accuracy of 95.74%. Zeng et al. (2021) combined gray recurrence plot (GRP) and densely connected convolutional network (DenseNet) for epilepsy detection and even achieved 100% excellent classification accuracy in each classification experiment. Mouleeshuwarapprabu and Kasthuri (2020) proposed a Non-linear Vector Decomposed Neural Network (NVDN) detect epileptic seizures and obtained 95.60% effective epilepsy detection results. Sharma et al. (2020) described a computationally fast seizure classification algorithm using non-linear higher-order statistics and deep neural network algorithms. This technique could capture weak information related to epilepsy EEG signals and achieved 100% seizure classification accuracy. Juarez-Guerra et al. (2020) proposed a new epilepsy seizure detection method for classifying epilepsy seizures, namely Multidimensional Radial Wavelons Feed-Forward Wavelet Neural Network (MRW-FFWNN). The experiment showed that the accuracy of the three classifications was 93.33%. From the above research results, it is not difficult to find that the research on epilepsy detection has been fruitful, and even some epilepsy detection algorithms have reached 100% accuracy. However, scientists' research on epilepsy does not stop there. The goal they really want to achieve is to prevent it before it happens, in other words, to predict epilepsy.

SEIZURE PREDICTION AND LOCALIZATION

In the 1970s, seizure prediction has become a hot research topic (Assi et al., 2017). MohanBabu et al. (2020) focused on the seizure prediction obtained from the CHB-MIT scalp EEG dataset using an optimized deep learning network model (ODLN), and the experiment by them provided 100% accuracy of seizure prediction. Zhang and Parhi (2016) extracted 44 features every 2 s for each channel and then ranked and selected them in a specific way. The selected features were processed by the Kalman filter and then inputted into the SVM classifier. This algorithm could achieve 100% sensitivity on the Freiburg EEG dataset. Daoud and Bayoumi (2019) applied deep learning to achieve epileptic seizure prediction, achieving epileptic seizure prediction while attaining 99.9% accuracy of epileptic seizure prediction.

Identifying epileptogenic zones prior to surgery is an indispensable step for patient before surgery. Alshebeili et al. (2020) proposed a framework that uses DWT and SVM to solve the problem of focus positioning. The framework used the best frequency band characteristics and wavelet coefficient characteristics, and its positioning accuracy could reach 88.0%. Sriraam and Raghu (2017) extracted 26 features from focal and non-focal EEG, then they used Wilcoxon rank sum test to select significant features and used an optimized SVM classifier with 10-fold cross-validation to perform important functions classification. This method achieved an accuracy of 92.15% and could be used to identify focal EEG signals to locate epileptic areas. Myers et al. (2020) also proposed a novel method for automatic localization of seizure on the scalp from clinical EEG data, which could get 93.3% accuracy and 100% sensitivity. Wu et al. (2021) presented a new localization method for epileptic seizure onset zones (SOZs), an unsupervised clustering method based on the combination of adaptive-genetic-algorithm-based matching pursuit (AGA-MP) and k-medoids clustering method. Moreover, compared with several existing methods, this method had certain advantages in sensitivity and specificity.

OTHER APPLICATIONS OF MACHINE LEARNING

Machine learning algorithms applied to EEG signals have also shined in other fields. Seal et al. (2021) used deep CNN to detect Depression and the detection accuracy of this algorithm was 99.37%. While Zhang et al. (2021) used CNN for motor imagery (MI) classification, and the average accuracy of the model reached over 88.4%. Huang (2021) recognized different psychological emotions via improved SVM, whose classification accuracy was as high as 85.9%. Raurale et al. (2021) develop an automated system combining quadratic time-frequency distribution (TFD) with CNN to identifying the severity of hypoxic-ischemic encephalopathy injury (HIE), which could assist clinical decision-making for neonates with HIE.

CONCLUSION

Since the beginning of the twenty-first century, the rapid development of artificial intelligence and machine learning, epilepsy detection techniques based on EEG signals has attracted more and more attention from researchers. This minireview briefly introduced the basic idea of epilepsy detection techniques based on EEG signals. From epilepsy EEG data and preprocessing to feature extraction and classification, the research progress of epilepsy automatic detection techniques in recent years were reviewed. Due to the random nature of epileptic seizures, fast and convenient seizure detection is essential for the immediate treatment of epilepsy patients. There is still much room for the development of epilepsy detection techniques. Here are a few points about the future development trend of epilepsy detection techniques based on EEG signals.

1. Seizure prediction and localization are still one of the future development directions of epilepsy detection techniques. Seizure prediction can effectively improve the quality of epilepsy patients, and non-invasive epilepsy focus localization can better assist clinicians in epilepsy diagnosis time and save costs.
2. Epilepsy detection is related to the patient's age, region, and other things, but the publicly available epilepsy EEG datasets are limited. Therefore, many epilepsy clinical EEG data from different countries and different countries and different age groups need to be improved.
3. With the development of machine learning, more and more new methods are applied to the feature extraction and classification of epilepsy EEG signals. The emergence of deep learning may gradually replace machine learning as the mainstream epilepsy diagnosis method in the future.
4. With wireless transmission technology development, seizure detection may get rid of wired transmission in the future and realize remote epilepsy detection.

In recent years, more and more new methods have begun to be applied to the automatic detection of epilepsy. The development of faster and more accurate epilepsy detection models will contribute to epilepsy detection techniques in clinical diagnosis and the development of portable and integrated epilepsy detection equipment. Therefore, a concise and efficient epilepsy detection model will become an inevitable development trend in the future.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct and intellectual contribution to the work, and approved it for publication.

REFERENCES

Aayesha, Qureshi, M. B., Afzaal, M., Qureshi, M. S., and Fayaz, M. (2021). Machine learning-based EEG signals classification model for epileptic seizure detection. *Multimed. Tools Appl.* (in press). doi: 10.1007/s11042-021-10597-6

Acharya, U. R., Hagiwara, Y., and Adeli, H. (2018a). Automated seizure prediction. *Epilepsy Behav.* 88, 251–261. doi: 10.1016/j.yebeh.2018.09.030

Acharya, U. R., Oh, S. L., Hagiwara, Y., Tan, J. H., and Adeli, H. (2018b). Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals. *Comput. Biol. Med.* 100, 270–278. doi: 10.1016/j.compbiomed.2017.09.017

Ahmet, M., and Aydin, A. (2018). Seizure onset detection based on frequency domain metric of empirical mode decomposition. *Signal Image Video Process.* 12, 1489–1496. doi: 10.1007/s11760-018-1304-y

Ailckovic, E., Kevric, J., and Subasi, A. (2018). Performance evaluation of empirical mode decomposition, discrete wavelet transform, and wavelet packed decomposition for automated epileptic seizure detection and prediction. *Biomed. Signal Process. Control.* 39, 94–102. doi: 10.1016/j.bspc.2017.07.022

Akyol, K. (2020). Stacking ensemble based deep neural networks modeling for effective epileptic seizure detection. *Expert Syst. Appl.* 148:113239. doi: 10.1016/j.eswa.2020.113239

Al-Hadeethi, H., Abdulla, S., Diykh, M., Deo, R. C., and Green, J. H. (2020). Adaptive boost LS-SVM classification approach for time-series signal classification in epileptic seizure diagnosis applications. *Expert Syst. Appl.* 161:113676. doi: 10.1016/j.eswa.2020.113676

Alshebeili, S. A., Sedik, A., El-Rahiem, B., Alotaiby, T. N., Banby, G. M., El-Khobby, H. A., et al. (2020). Inspection of EEG signals for efficient seizure prediction. *Appl. Acoust.* 166:107327. doi: 10.1016/j.apacoust.2020.107327

Amin, H. U., Yusoff, M. Z., and Ahmad, R. F. (2020). A novel approach based on wavelet analysis and arithmetic coding for automated detection and diagnosis of epileptic seizure in EEG signals using machine learning techniques. *Biomed. Signal Process. Control.* 56:101707. doi: 10.1016/j.bspc.2019.101707

Anoop, K., Charchit, G., Shvet, D., and Chirag, M. (2021). Perampanel in cases of refractory temporal lobe epilepsy – A report of two post-operative and two eligible for temporal lobectomy. *Interdiscip. Neurosurg.* 24:101094. doi: 10.1016/j.inat.2021.101094

Assi, E. B., Nguyen, D. K., Rihana, S., and Sawan, M. (2017). Towards accurate prediction of epileptic seizures: a review. *Biomed. Signal Process. Control.* 34, 144–157. doi: 10.1016/j.bspc.2017.02.001

Chen, L. L., Zhang, J., Zou, J. Z., Zhao, C. J., and Wang, G. S. (2014). A framework on wavelet-based nonlinear features and extreme learning machine for epileptic seizure detection. *Biomed. Signal Process. Control.* 10, 1–10. doi: 10.1016/j.bspc.2013.11.010

Chen, S., Zhang, X., Chen, L., and Yang, Z. (2019). Automatic diagnosis of epileptic seizure in electroencephalography signals using nonlinear dynamics features. *IEEE Access* 7, 61046–61056. doi: 10.1109/ACCESS.2019.2915610

Choubey, H., and Pandey, A. (2020). A combination of statistical parameters for the detection of epilepsy and EEG classification using ANN and KNN classifier. *Signal Image Video Process.* 15, 475–483. doi: 10.1007/s11760-020-01767-4

Craley, J., Johnson, E., Jouny, C., and Venkataraman, A. (2021). Automated inter-patient seizure detection using multichannel convolutional and recurrent neural networks. *Biomed. Signal Process. Control.* 64:102360. doi: 10.1016/j.bspc.2020.102360

Daoud, H., and Bayoumi, M. A. (2019). Efficient epileptic seizure prediction based on deep learning. *IEEE Trans. Biomed. Circuits Syst.* 13, 804–813.

Gotman, J. (1982). Automatic recognition of epileptic seizures in the EEG. *Electroencephalogr. Clin. Neurophysiol.* 54, 530–540. doi: 10.1016/0013-4694(82)90038-4

Hassan, A. R., and Subasi, A. (2016). Automatic identification of epileptic seizures from EEG signals using linear programming boosting. *Comput. Meth. Programs Biomed.* 136, 65–77. doi: 10.1016/j.cmpb.2016.08.013

Herwig, U., Satrapi, P., and Schonfeldt-Lecuona, C. (2003). Using the international 10-20 EEG system for positioning of transcranial magnetic stimulation. *Brain Topogr.* 16, 95–99. doi: 10.1023/B:BRAT.0000006333.93597.9d

Huang, C. Y. (2021). Recognition of psychological emotion by EEG features. *Netw. Model. Anal. Health Inform. Bioinform.* 10:12. doi: 10.1007/s13721-020-00283-2

Huang, G. B., Zhu, Q. Y., and Siew, C. K. (2006). Extreme learning machine: theory and applications. *Neurocomputing* 70, 489–501. doi: 10.1016/j.neucom.2005.12.126

Iesmantas, T., and Alzbutas, R. (2020). Convolutional neural network for detection and classification of seizures in clinical data. *Med. Biol. Eng. Comput.* 58, 1919–1932. doi: 10.1007/s11517-020-02208-7

Ito, S., Nagumo, K., Nishikawa, A., Oguni, H., and Nagata, S. (2021). Low-dose phenobarbital for epilepsy with myoclonic absences: a case report. *Brain Dev.* 43, 666–668. doi: 10.1016/j.braindev.2020.12.018

Janjarasjitt, S. (2017). Epileptic seizure classifications of single-channel scalp EEG data using wavelet-based features and SVM. *Med. Biol. Eng. Comput.* 55, 1743–1761. doi: 10.1007/s11517-017-1613-2

Juarez-Guerra, E., Alarcon-Aquino, V., Gomez-Gil, P., Ramirez-Cortes, J. M., and Garcia-Trevino, E. S. (2020). A new wavelet-based neural network for classification of epileptic-related states using EEG. *J. Signal Process. Syst.* 92, 187–211. doi: 10.1007/s11265-019-01456-7

Li, D. Z., Xie, Q., Jin, Q., and Hirasawa, K. (2016). A sequential method using multiplicative extreme learning machine for epileptic seizure detection. *Neurocomputing* 214, 692–707. doi: 10.1016/j.neucom.2016.06.056

Liu, Q., Zhao, X. G., Hou, Z. G., and Liu, H. G. (2017). Epileptic seizure detection based on the kernel extreme learning machine. *Technol. Health Care* 25, S399–S409. (in press). doi: 10.3233/THC-171343

Liu, Y., Jiang, B., Feng, J., Hu, J. Z., and Zhang, H. B. (2020). Classification of EEG signals for epileptic seizures using feature dimension reduction algorithm based on LPP. *Multimed. Tools Appl.* doi: 10.1007/s11042-020-09135-7

Liu, Y. L., Hou, B. C., Zhang, Y. S., Fan, Y. T., Peng, B. W., Liu, W. H., et al. (2018). Anticonvulsant agent DPP4 inhibitor sitagliptin downregulates CXCR3/RAGE pathway on seizure models. *Exp. Neurol.* 307, 90–98. doi: 10.1016/j.expneurol.2018.06.004

Makaram, N., Ellenrieder, N. V., Tanaka, H., and Gotman, J. (2020). Automated classification of five seizure onset patterns from intracranial electroencephalogram signals. *Clin. Neurophysiol.* 131, 1210–1218. doi: 10.1016/j.clinph.2020.02.011

Martinerie, J., Adam, C., Quyen, M., Baulac, M., Clemenceau, S., Renault, B., et al. (1998). Epileptic seizures can be anticipated by non-linear analysis. *Nat. Med.* 4, 1173–1176. doi: 10.1038/2667

MohanBabu, G., Anupallavi, S., and Ashokkumar, S. R. (2020). An optimized deep learning network model for eeg based seizure classification using synchronization and functional connectivity measures. *J. Ambient Intell. Humaniz. Comput.* (in press). doi: 10.1007/s12652-020-02383-3

Mouleeshuwarapprabu, R., and Kasthuri, N. (2020). Nonlinear vector decomposed neural network based EEG signal feature extraction and detection of seizure. *Microprocess. Microsyst.* 76:103075. doi: 10.1016/j.micpro.2020.103075

Myers, M. H., Padmanabha, A., Bidelman, G. M., and Wheless, J. W. (2020). Seizure localization using EEG analytical signals. *Clin. Neurophysiol.* 131, 2131–2139. doi: 10.1016/j.clinph.2020.05.034

Nogay, H. S., and Adeli, H. (2020). Detection of epileptic seizure using pretrained deep convolutional neural network and transfer learning. *Eur. Neurol.* 83, 602–614. doi: 10.1159/000512985

Ott, D., Siddarth, P., Gurbani, S., Koh, S., and Caplan, R. (2010). Behavioral disorders in pediatric epilepsy: unmet psychiatric need. *Epilepsia* 44, 591–597. doi: 10.1046/j.1528-1157.2003.25002.x

Prior, P. F., Virden, R. S. M., and Maynard, D. E. (1973). An EEG device for monitoring seizure discharges. *Epilepsia* 14, 367–372. doi: 10.1111/j.1528-1157.1973.tb03975.x

Qi, Y., Zhou, W., Li, S., and Cai, D. (2011). Epileptic EEG classification based on extreme learning machine and nonlinear features. *Epilepsy Res.* 96, 29–38. doi: 10.1016/j.eplepsyres.2011.04.013

Raghu, S., Sriraam, N., Gommer, E. D., Hilkman, D., and Kubben, P. L. (2020). Cross-database evaluation of EEG based epileptic seizures detection driven by adaptive median feature baseline correction. *Clin. Neurophysiol.* 131, 1567–1578. doi: 10.1016/j.clinph.2020.03.033

Ramakrishnan, S., and Murugavel, A. (2019). Epileptic seizure detection using fuzzy-rules-based sub-band specific features and layered multi-class SVM. *Pattern Anal. Appl.* 22, 1161–1176. doi: 10.1007/s10044-018-0691-6

Raurale, S. A., Boylan, G. B., Mathieson, S. R., Marnane, W. P., and O'Toole, J. M. (2021). Grading hypoxic-ischemic encephalopathy in neonatal EEG with convolutional neural networks and quadratic time-frequency distributions. *J. Neural Eng.* 18:046007. doi: 10.1088/1741-2552/abe8ae

Seal, A., Bajpai, R., Agnihotri, J., Yazidi, A., Herrera-Viedma, E., and Krejcar, O. (2021). Deprnet: a deep convolution neural network dramework for detecting depression using EEG. *IEEE Trans. Instrum. Meas.* 70:2505413. doi: 10.1109/TIM.2021.3053999

Sharma, R., Pachori, R. B., and Sircar, P. (2020). Seizures classification based on higher order statistics and deep neural network. *Biomed. Signal Process. Control.* 59:101921. doi: 10.1016/j.bspc.2020.101921

Shimogawa, T., Mukae, N., Morioka, T., Tanaka, S., Sakata, A., Uehara, T., et al. (2021). Possible relationship between vagus nerve stimulation and ictal discharges revealed by long-term electroencephalographic and electrocorticographic monitoring in a non-responsive patient. *Interdiscip. Neurosurg.* 24:101066. doi: 10.1016/j.inat.2020.101066

Solaija, M., Sajid, S., Khawar, K., Ali, H. S., and Kamboh, K. (2018). Dynamic mode decomposition based epileptic seizure detection from scalp EEG. *IEEE Access* 6, 38683–38692. doi: 10.1109/ACCESS.2018.2853125

Song, J. L., Hu, W., and Zhang, R. (2016). Automated detection of epileptic EEGs using a novel fusion feature and extreme learning machine. *Neurocomputing* 175, 383–391. doi: 10.1016/j.neucom.2015.10.070

Sra, B., Ns, B., Yt, C., Svr, D., Ash, E., and Plk, C. (2019). Performance evaluation of dwt based sigmoid entropy in time and frequency domains for automated detection of epileptic seizures using SVM classifier. *Comput. Biol. Med.* 110, 127–143. doi: 10.1016/j.compbiomed.2019.05.016

Sriraam, N., and Raghu, S. (2017). Classification of focal and Non focal epileptic seizures using multi-features and SVM classifier. *J. Med. Syst.* 41:160. doi: 10.1007/s10916-017-0800-x

Sun, C., Cui, H., Zhou, W., Nie, W., and Yuan, Q. (2019). Epileptic seizure detection with EEG textural features and imbalanced classification based on EasyEnsemble learning. *Int. J. Neural Syst.* 29:1950021. doi: 10.1142/S0129065719500217

Supriya, S., Siuly, S., Wang, H., and Zhang, Y. (2020). Automated epilepsy detection techniques from electroencephalogram signals: a review study. *Health Inf. Sci. Syst.* 8:33. doi: 10.1007/s13755-020-00129-1

Tao, Z., Chen, W., and Li, M. (2017). AR based quadratic feature extraction in the VMD domain for the automated seizure detection of EEG using random forest classifier. *Biomed. Signal Process. Control.* 31, 550–559. doi: 10.1016/j.bspc.2016.10.001

Vicnesh, J., and Hagiwara, Y. (2019). Accurate detection of seizure using nonlinear parameters extracted from EEG signals. *J. Mech. Med. Biol.* 19:1940004. doi: 10.1142/S0219519419400049

Wang, J., Huang, J., Li, Y. Q., Yao, S., and Sun, X. D. (2021). Neuregulin 1/ErbB4 signaling contributes to the anti-epileptic effects of the ketogenic diet. *Cell Biosci.* 11:29. doi: 10.1186/s13578-021-00536-1

Wang, Y., Li, Z., Feng, L., Zheng, C., and Zhang, W. (2017). Automatic detection of epilepsy and seizure using multiclass sparse extreme learning machine classification. *Comput. Math. Method Med.* 2017:6849360. doi: 10.1155/2017/6849360

Wei, Z., Zou, J., Zhang, J., and Xu, J. (2019). Automatic epileptic EEG detection using convolutional neural network with improvements in time-domain. *Biomed. Signal Process. Control.* 53:101551. doi: 10.1016/j.bspc.2019.04.028

Wu, M., Wan, T., Wan, X., Fang, Z., and Du, Y. (2021). A new localization method for epileptic seizure onset zones based on time-frequency and clustering analysis. *Pattern Recognit.* 111:107687. doi: 10.1016/j.patcog.2020.107687

Yao, X., Li, X., Ye, Q., Huang, Y., and Zhang, G. Q. (2021). A robust deep learning approach for automatic classification of seizures against non-seizures. *Biomed. Signal Process. Control.* 64:102215.

Yuan, S., Liu, J., Shang, J., Kong, X., Qi, Y., and Ma, Z. (2018). The earth mover's distance and bayesian linear discriminant analysis for epileptic seizure detection in scalp EEG. *Biomed. Eng. Lett.* 8, 373–382. doi: 10.1007/s13534-018-0082-3

Zeng, M., Zhang, X., Zhao, C., Lu, X., and Meng, Q. (2021). GRP-DNet: a gray recurrence plot-based densely connected convolutional network for classification of epileptiform EEG. *J. Neurosci. Methods* 347:108953. doi: 10.1016/j.jneumeth.2020.108953

Zhang, C., Kim, Y. K., and Eskandarian, A. (2021). EEG-inception: an accurate and robust end-to-end neural network for EEG-based motor imagery classification. *J. Neural Eng.* 18:046014. doi: 10.1088/1741-2552/abed81

Zhang, W., Zhang, Z., Wang, L., Chao, H. C., and Zhou, Z. (2019). Extreme learning machines with expectation kernels. *Pattern Recognit.* 96:106960. doi: 10.1016/j.patcog.2019.07.005

Zhang, Z., and Parhi, K. K. (2016). Low-complexity seizure prediction from iEEG/sEEG using spectral power and ratios of spectral power. *IEEE Trans. Biomed. Circuits Syst.* 10, 693–706. doi: 10.1109/TBCAS.2015.2477264

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Predicting Motion Patterns Using Optimal Paths

Mads Fromreide[1,2] *and Alex Hansen*[3]*

[1]NORCE Norwegian Research Centre AS, Kristiansand, Norway, [2]Department of Applied Mathematics, University of Santiago de Compostela, Santiago de Compostela, Spain, [3]PoreLab, Department of Physics, Norwegian University of Science and Technology, Trondheim, Norway

The ability to navigate safely and efficiently through a given landscape is relevant for any intelligent moving object. Examples range from robotic science and traffic analysis, to the behavior within an ecosystem. Many objects tend to move in patterns depending on their nature. By establishing models of patterns of motion one may estimate the future motion within an area. We propose here a method for detecting regular patterns of motion by modeling the environment as an energy landscape, and locating optimal paths through it. As an example, we use maritime position Automatic Identification System (AIS) data as input to work out optimal routes between different start and end points when these are not located along the standard shipping lanes. These initial tests show that the method has potential for analyzing and determining regular patterns of motion.

Keywords: optimal paths, machine learning, motion prediction, energy landscape, Dijkstra algorithm

OPEN ACCESS

Edited by:
Antonio F. Miguel,
University of Evora, Portugal

Reviewed by:
Gyei-Kark Park,
Mokpo National Maritime University, South Korea
Mathias Anneken,
Fraunhofer-Institut für Optronik, Systemtechnik und Bildauswertung (IOSB), Germany

Correspondence:
Alex Hansen
alex.hansen@ntnu.no

Specialty section:
This article was submitted to Interdisciplinary Physics, a section of the journal Frontiers in Physics

Received: *20 January 2021*
Accepted: *02 June 2021*
Published: *21 June 2021*

Citation:
Fromreide M and Hansen A (2021) Predicting Motion Patterns Using Optimal Paths. Front. Phys. 9:656296. doi: 10.3389/fphy.2021.656296

1 INTRODUCTION

Imagine a large town square. The square has fixed structures such as fountains etc. that block direct pathways across it. It is a busy place where many pass across it in all directions. Cameras have been set up that record the motion of people across the town square. The question we will address in this paper is the following: Based on the recordings, is it possible to predict the motion of a single person from some start point A to some end point B across the town square, even when the points A and B are not located along the typical paths that people use across the square. This question has since long been posed in different contexts [1]. Examples range from the behavior of ecosystems to robotic navigation and traffic systems. A predator needs to account for the future motion of its prey in order to catch it, just as a ship needs to consider the future positions of other ships to avoid collisions [2].

Moving objects are influenced by both the landscape in which they move, as well as other objects, moving or not, within the same area [3–5]. There are several different ways to approach motion prediction. The most straight forward approach is to predict the motion of each object in a system individually, by assigning to each object a position as a function of time [6]. However, for large systems, this method would produce a large number of coupled equations. Hence, this approach would be unproductive in this case. A better approach is to exploit the fact that objects tend to move in patterns [3, 7]. Depending on their nature and surroundings, moving objects tend to move in regular patterns. By establishing a model of these motion-patterns in a given area, one may use the pattern itself when predicting future motion. This is the core idea of our approach.

When applying methods of pattern recognition to motion prediction, the process typically operates in two stages. The first stage is the actual pattern recognizing, which learns the regular patterns of motion using a set of training data. The next stage applies the learned pattern to predict the future motion. Further, this two-stage process may be grouped into two main groups of techniques; Grid-based techniques and cluster-based techniques [3].

The grid-based techniques are derived from the occupancy grid concept [8]. That is, the landscape is modeled as a grid and transition probabilities between the cells are calculated from the training data. The grid is then used directly for motion prediction. Grid-based techniques are frequently used in robot navigation systems [9–11].

Cluster-based techniques on the other hand, apply statistical decision tools in order to group similar trajectories into representative clusters. Several different clustering techniques exists, the Expectation-Maximization approach [12] is considered to be the state of the art [3]. Future motion of a moving object is then estimated as the representative cluster which the given route is most likely to belong to.

In this paper, we propose a dynamic grid-based technique for learning motion patterns by mapping it onto the optimal paths in a disordered landscape problem [13, 14]. We describe this problem as follows. Imagine a plane and that $\vec{x}$ is a point on this plane. There is a stochastic field $e(\vec{x})$ associated with the plane. We choose a path $\mathcal{P}$ through the plane starting at point $\vec{x}_A$ and ending at point $\vec{x}_B$. We integrate the field $e(\vec{x})$ along the path $\mathcal{P}$,

$$E_{\mathcal{P}} = \int_{\vec{x}\in\mathcal{P}} e\left(\vec{x}\right) \mathrm{d}\vec{x}. \tag{1}$$

The optimal path is found by the minimization

$$E_{\mathcal{O}} = \min_{\mathcal{P}} E_{\mathcal{P}} = \min_{\mathcal{P}} \int_{\vec{x}\in\mathcal{P}} e\left(\vec{x}\right) \mathrm{d}\vec{x}. \tag{2}$$

This problem has produced a large body of work within the statistical physics community. It is also closely related to the optimal path problem which is central in a large number of applications and fields [15–18].

The central idea we present in this paper is to relate the function $e(\vec{x})$ to the inverse of the density earlier paths raised to some power. We then identify the optimal path from $\vec{x}_A$ to $\vec{x}_B$ through this landscape.

We apply this idea to vessel traffic, using marine automatic identification system coordinates. We transform the coordinates to a dimensionless area and introduce a grid over the area. We associate each grid point with the local density of AIS coordinates. We implement the optimal paths through the area using the iterative algorithm of Hansen and Kertész [19, 21], but any other algorithms may be used, e.g., the Bellman-Ford or the Dijkstra algorithms [22–24].

We emphasize that we are not attempting here to present a fully implementable algorithm ready to be used on ships. Rather, this is a feasibility study testing whether the central ideas may work.

We note that optimal paths have been used earlier in connection with marine motion prediction [20]. However, the paths in this case are optimized with respect to length. This is a very different concept than what we present here.

We organize this paper as follows. **Section 2** describes the method we propose. In **Section 3**, we implement the method for marine AIS data. We end by a brief summary and discussion.

2 ALGORITHM

We now describe how we transform the AIS coordinates, given in terms of continuous longitude and latitude, into grid points. We then go on to describe the concept of optimal paths in this context and the algorithm used to extract it. Lastly, we describe how we turn this into path prediction.

2.1 From Automatic Identification System Coordinates to Grid

The automatic tracking system AIS uses tranceivers to allow ships to view surrounding marine traffic and to be seen themselves. It provides, among other services, a record of the position as a function of time for the equipped vessels passing through the area. This includes most large vessels.

We define our area of interest as the rectangle defined by the corners given by the longitudinal and latitudinal coordinates $\mathrm{long}_{\mathrm{min}}$, $\mathrm{long}_{\mathrm{max}}$, $\mathrm{lae}_{\mathrm{min}}$ and $\mathrm{lae}_{\mathrm{max}}$. A given ship at a given time is at position $(\mathrm{long}_k, \mathrm{lae}_k)$ where the subscript refers to the position record (i.e., which ship and at what time). We introduce dimensionless Cartesian coordinates to describe its position, $x_k \in \{0, N-1\}$ and $y_k \in \{0, N-1\}$, where N is an integer, given by

$$x_k = \frac{\mathrm{long}_k - \mathrm{long}_{\mathrm{min}}}{\mathrm{long}_{\mathrm{max}} - \mathrm{long}_{\mathrm{min}}} (N-1), \tag{3}$$

and

$$y_k = \frac{\mathrm{lae}_k - \mathrm{lae}_{\mathrm{min}}}{\mathrm{lae}_{\mathrm{max}} - \mathrm{lae}_{\mathrm{min}}} (N-1). \tag{4}$$

The position of a given ship at a given time k at (x_k, y_k) is then located within grid cell defined by the corner nodes $i_k^1 \mathrm{mod}(x_k)$, $i_k^2 \mathrm{mod}(x_k) + 1$, $j_k^1 \mathrm{mod}(y_k)$ and $j_k^2 \mathrm{mod}(y_k) + 1$. In order to construct a path density defined at the nodes (i, j), $\rho_{i,j}$, we assign a weight to the position of ship position and time k, (x_k, y_k) distributed among the four corned nodes (i_k^1, j_k^1), (i_k^2, j_k^1), (i_k^2, j_k^2) and (i_k^1, j_k^2). If $r_{i_k^1, j_k^1}$ is the distance between (x_k, y_k) and (i_k^1, j_k^1), and likewise for $r_{i_k^2, j_k^1}$, $r_{i_k^2, j_k^2}$ and $r_{i_k^1, j_k^2}$, we define

$$R_{x_k, y_k} = r_{i_k^1, j_k^1} + r_{i_k^2, j_k^1} + r_{i_k^2, j_k^2} + r_{i_k^1, j_k^2}. \tag{5}$$

We define the weights

$$\begin{aligned} W_{i_k^1, j_k^1} &= \frac{1}{3R_{x_k, y_k}} \left(R_{x_k, y_k} - r_{i_k^1, j_k^1} \right), \\ W_{i_k^2, j_k^1} &= \frac{1}{3R_{x_k, y_k}} \left(R_{x_k, y_k} - r_{i_k^2, j_k^1} \right), \\ W_{i_k^2, j_k^2} &= \frac{1}{3R_{x_k, y_k}} \left(R_{x_k, y_k} - r_{i_k^2, j_k^2} \right), \\ W_{i_k^1, j_k^2} &= \frac{1}{3R_{x_k, y_k}} \left(R_{x_k, y_k} - r_{i_k^1, j_k^2} \right). \end{aligned} \tag{6}$$

The path density may then be defined as

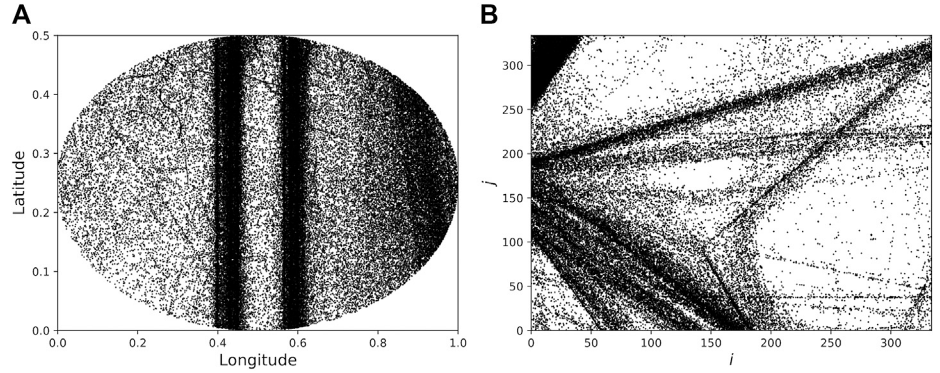


FIGURE 1 | (A) Data set *A* showing AIS position recordings with positions in degrees relative to the origin. **(B)** A selected subset of area *A* spanning 0.3° in both the longitudinal and latitudinal directions, transformed into a grid of dimensions 334 × 334 with grid size 100 × 100 m^2.

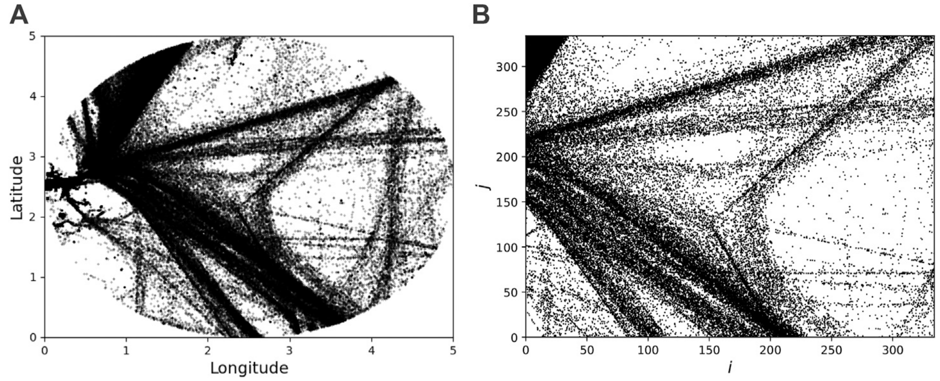


FIGURE 2 | (A) Data set *B* showing AIS position recordings with positions in degrees relative to the origin. **(B)** A selected subset of area *B* spanning 3° in both the longitudinal and latitudinal directions, transformed into a grid of dimensions 334 × 334 with cell size 1000 × 1000 m^2.

$$\rho_{i,j} = \sum_{k=1}^{K} W_{i_k,j_k}, \tag{7}$$

when K is the number of position recordings within the grid cell.

We now associate a weight with node (i, j),

$$e_{i,j} = \begin{cases} \rho_{i,j}^{-\alpha}, & \text{if } \rho_{i,j} > 0, \\ m, & \text{if } \rho_{i,j} = 0, \\ M, & \text{if } (i,j) \text{ is on land,} \end{cases} \tag{8}$$

where α is an adjustable parameter controlling the magnitude of the fluctuations of $e_{i,j}$: if $\alpha \to 0$ the fluctuations are smoothened out and disappear when $\alpha = 0$ as all nodes then are assigned the same weight. The parameter m is chosen so that there is a balance between the tendency for a path to follow the normal shipping lanes (where $\rho_{i,j}$ is large) and a path being as short as possible. The other value $M \gg m$ ensures that no paths crosses land.

The last step in setting up the system is to assign weights to the *links* between neighboring nodes. Let the node (i_{nn}, j_{nn}) be one of the four nearest neighbors of node (i, j). Then, the link between them is given the weight

$$e_{i,j;i_{nn},j_{nn}} = \frac{1}{2}\left(e_{i,j} + e_{i_{nn},j_{nn}}\right). \tag{9}$$

We also allow for diagonal paths. The link between node (i, j) and its diagonal neighbors (i_{nn}, j_{nn}) as

$$e_{i,j;i_{nn},j_{nn}} = \frac{\sqrt{2}}{2}\left(e_{i,j} + e_{i_{nn},j_{nn}}\right), \tag{10}$$

where the factor $\sqrt{2}$ is introduced to take into account the additional length of the diagonal edges.

2.2 Optimal Path Construction

We define a path $\mathcal{P}$ between two nodes A at (i_A, j_A) and B at (i_B, j_B) as a continuous chain of neighboring links linking the two nodes. We associate a weight of the path in the same way as in **Eq. 1**,

$$E_{\mathcal{P}} = \sum_{(i,j)\in\mathcal{P}} e_{i,j;i_{nn},j_{nn}}. \tag{11}$$

The optimal path is then

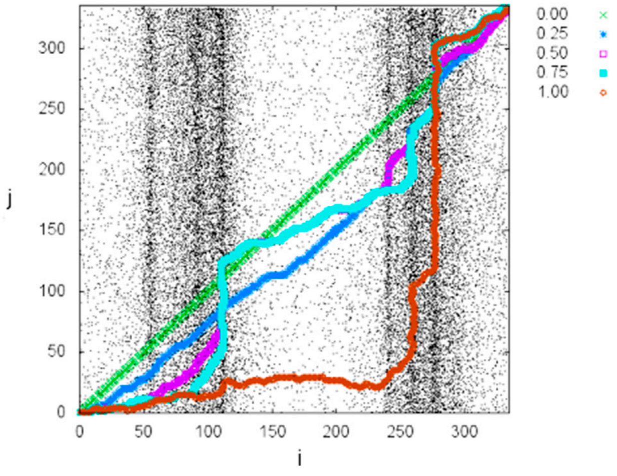


FIGURE 3 | Optimal paths between the lower left and upper right corner of the grid in **Figure 1B** for $\alpha = 0$, 0.25, 0.50, 0.75 and 1. The parameter β, defined in **Eq. 15** was set to zero.

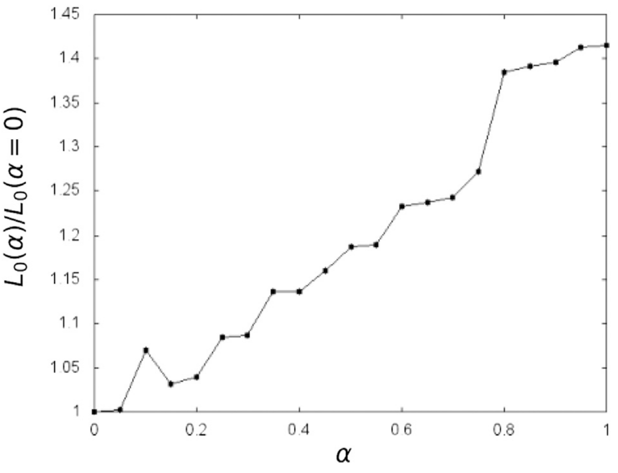


FIGURE 4 | The length of the optimal paths, $L_{\mathcal{O}}$ between the lower left and upper right corner of III as a function of α, while keeping $\beta = 0$.

$$E_{\mathcal{O}} = \min_{\mathcal{P}} E_{\mathcal{P}} = \min_{\mathcal{P}} \sum_{(i,j)\in\mathcal{P}} e_{i,j;i_{nn},j_{nn}}. \tag{12}$$

We will in the following assume that both nodes A and B lie on the edges of the grid. In order to identify the optimal path, we use the algorithm of Hansen and Kertész [19]. It consists of two main steps; first an initialization and then an updating process. A variable $e_{i,j}$ is assigned to each node. For the nodes on the edges of the grid, the values $e_{i,j}$ stay fixed, while it is updated for the internal nodes iteratively. The iteration algorithm for the internal nodes is

$$e_{i,j} \rightarrow e_{i,j} = \min_{i_{nn},j_{nn}}\left(e_{i,j;i_{nn},j_{nn}} + e_{i_{nn},j_{nn}}\right). \tag{13}$$

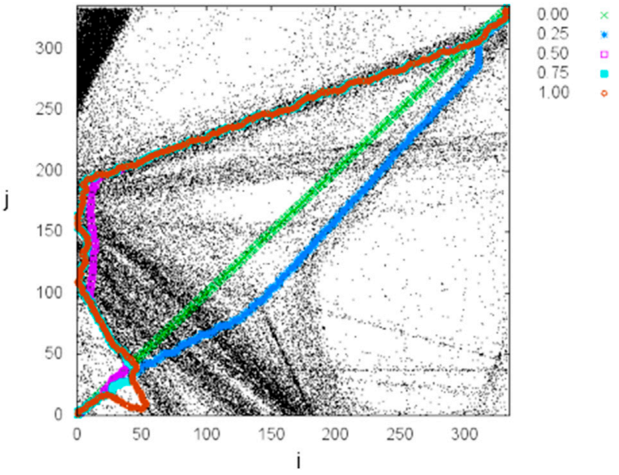


FIGURE 5 | Optimal paths between the lower left and upper right corner of the grid in **Figure 2B** for $\alpha = 0$, 0.25, 0.50, 0.75 and 1. The parameter β, defined in **Eq. 15** was set to zero.

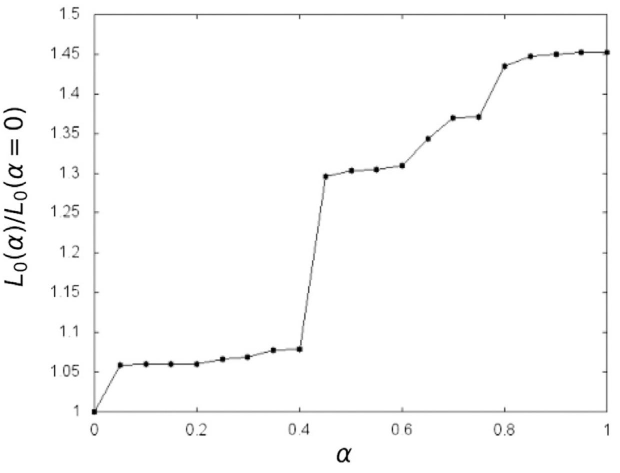


FIGURE 6 | The length of the optimal paths, $L_{\mathcal{O}}$ between the lower left and upper right corner of V as a function of α, while keeping $\beta = 0$.

After M iterations, the variable $e_{i,j}$ will contain the sum of the weights along the optimal path starting at node (i,j) of length M. The end point of the optimal path is not specified. Furthermore, the optimal path may curl up on itself, creating a tadpole configuration.

Consider now a node (i_A, j_A) on the boundary of the grid. In order to find the optimal path from an internal node (i,j) to (i_A, j_A), we set the value e_{i_A,j_A} to zero, while for the remaining boundary nodes the value e_{i_c,j_c} is set to a very large positive value. The updating process for the internal nodes is carried out according to **Eq. 13**, until all values $e_{i,j}$ no longer change. At this point, the value of $e_{i,j}$ contains the value of $E_{\mathcal{O}} = E_{i,j;i_A,j_A}$ along the optimal path between nodes (i,j) and (i_A, j_A).

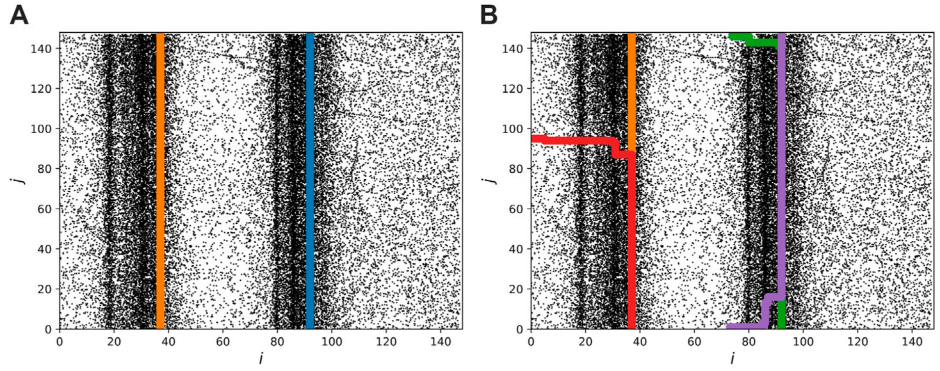


FIGURE 7 | Area *A*: **(A)** The two most optimal paths between the intervals along different edges. **(B)** The five most optimal paths between different edges. Each edge has been divided into two intervals, and fixing $\alpha = 0.8$, and $\beta = 0.8$.

We now choose another boundary node (i_B, j_B) as end point for the optimal paths. Hence, we fix $e_{i_B,j_B} = 0$ and fix all the other boundary nodes to a large positive value, including boundary node (i_A, j_A). The internal node values are initially set to zero, $e_{i,j} = 0$. We then iterate according to **Eq. 13**. When numbers no longer change, $e_{i,j}$ will contain $E_{i,j;i_B,j_B} = E_O$ for the optimal path between nodes (i, j) and (i_B, j_B).

We may now combine the optimal paths starting at boundary node (i_A, j_A) and ending at internal node (i, j) with the optimal path starting at internal node (i, j) and ending at boundary node (i_B, j_B). The optimal weight E_O for this combined path is then given by,

$$E_{i_A,j_A;i,j;i_B,j_B} = \min_{i_{nn},j_{nn}}\Big(E_{i_A,j_A;i,j} + e_{i,j;i_{nn},j_{nn}} + E_{i_{nn},j_{nn};i_B,j_B}, E_{i_A,j_A;i_{nn},j_{nn}} + e_{i_{nn},j_{nn};i,j} + E_{i,j;i_B,j_B}\Big). \tag{14}$$

Associating each internal node (i, j) with the value $e_{i,j} = E_{i_A,j_A;i,j;i_B,j_B}$ leads to the construction of a pathscape [21]. The optimal path between edge nodes (i_A, j_A) and (i_B, j_B) is the sequence of nodes associated with the smallest $e_{i,j}$ values. There will then be a sequence of nodes having the second smallest values $e_{i,j}$. This sequence will branch out from the globally optimal path as some node, to rejoin it at a different node along the path. Then there will be sequence with the third smallest values $e_{i,j}$ branching off and rejoining nodes belonging to the two paths containing the two smaller $e_{i,j}$—and so on. Each internal node will belong to some path in this hierarchy.

2.3 Predicting Paths

We now focus our attention on pathscapes where boundary nodes (i_A, j_A) and (i_B, j_B) are placed along different edges. There are $6N^2$ possible combinations. It may be convenient to coarse grain the end points of the optimal paths. Hence, we divide each edge into intervals of length L_I. This means that we set the weight of all the edge nodes nodes (i'_A, j'_A) belonging to the interval, $e_{i_{A'},j_{A'}} = 0$. The pathscape will then consist of all optimal paths starting somewhere in the first interval, (i'_A, j'_A), passing through internal node (i, j) and then ending at a node $(i_{B'}, j_{B'})$ somewhere in the end interval. Hence, the number of pathscapes is then reduced from $6N^2$ to $6n^2$, where $n = N/L_I$.

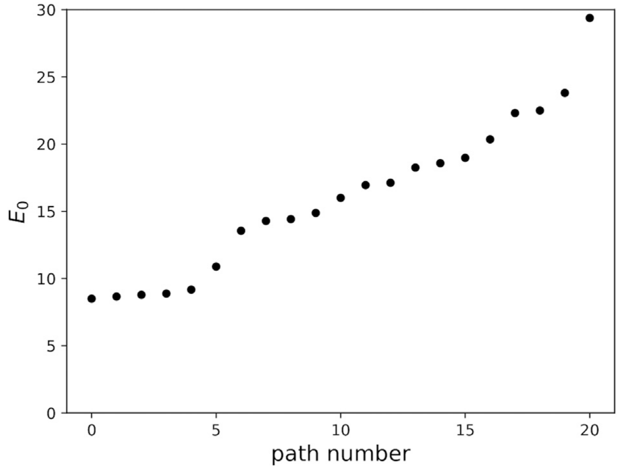


FIGURE 8 | The weight E_O ranging from the most optimal (No. 0) to the least optimal (No. 20) for paths between intervals on different edges as shown in **Figure 7**.

Suppose the optimal path $(i_{A'}, j_{A'}; i, j; i_{B'}, j_{B'})$ has a length $L_{i_{A'},j_{A'};i,j;i_{B'},j_{B'}}$ and a weight $E_{i_{A'},j_{A'};i,j;i_{B'},j_{B'}}$. Using the weight alone in predicting paths does not function well since a short path through a high-weight region may be as optimal as a longer path through a low-weight region. We therefore renormalize the weights, $E_{i_{A'},j_{A'};i,j;i_{B'},j_{B'}} \rightarrow E'_{i_{A'},j_{A'};i,j;i_{B'},j_{B'}}$ by setting

$$E'_{i_{A'},j_{A'};i,j;i_{B'},j_{B'}} = C_p \frac{E_{i_{A'},j_{A'};i,j;i_{B'},j_{B'}}}{\left(L_{i_{A'},j_{A'};i,j;i_{B'},j_{B'}}\right)^{\beta}}, \tag{15}$$

where β is an adjustable parameter. The constant C_p is introduced to further separate between different optimal paths. We note that higher β makes longer paths more favorable.

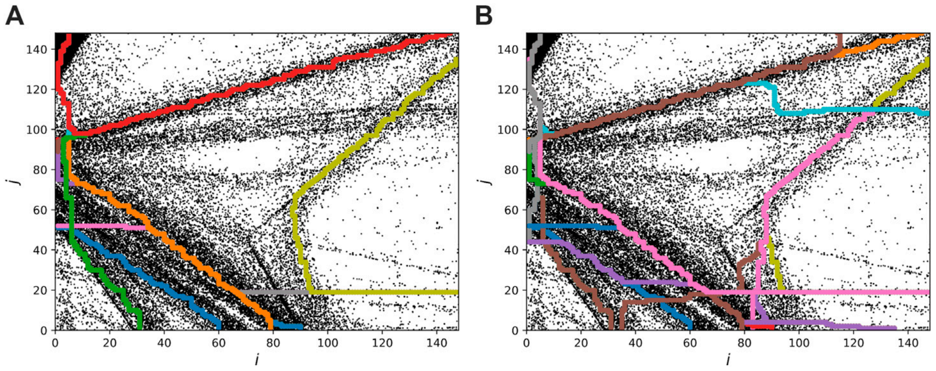


FIGURE 9 | Area *B*: Here the grid has been divided into three segments along each edge. In **(A)** we show the 24 most optimal paths and in **(B)** we show the 48 most optimal paths. We fixed the parameter values to $\alpha = 0.08$ and $\beta = 0.8$.

3 ANALYSIS OF TWO AUTOMATIC IDENTIFICATION SYSTEM PATTERNS

We denote the two AIS sets we consider in the following *A* and *B*. **Figures 1**, **2** show the two areas and the subsets that we use in our analysis. Both subsets, shown in **Figures 1B**, **2B**, have size 100×100 and where each grid block has size $100 \times 100\ \text{m}^2$ (area *A*) and $1000 \times 1000\ \text{m}^2$ (area *B*). We see that *A* has a simpler structure than *B*, consisting of two vertical clusters, while *B* includes multiple clusters with different orientation. By “cluster” we mean an area with high density of position recordings.

We now consider area *A*. Setting parameter $\beta = 0$ in **Eq. 15**, we show in **Figure 3** the optimal path between the lower left and upper right corners of the grid 1b for different values of α.

We show in **Figure 4** the length of the optimal path between the lower left and upper right corner in **Figure 3** for different values of α and with $\beta = 0$, L_O. We note that L_O is approximately linear in α for $\alpha < 0.75$, at which there is a jump.

Turning to area *B*, we show in **Figure 5** the optimal paths starting from the lower left corner and ending at the upper right corner of the grid shown in **Figure 2B** as a fuction of α while keeping $\beta = 0$.

We show in **Figure 6** the length of the optimal path between the lower left and upper right corner in **Figure 5** for different values of α and $\beta = 0$. As in **Figure 4** for area *A*, we find a jump in the length of the optimal paths for a given value of α, here $\alpha \approx 0.4$. However, there are clearly defined plateaus in the optimal path length, e.g., for $0.05 < \alpha < 0.4$.

We now introduce intervals L_I as described in Section C. We consider first a more coarse grained section of area *A*, covered by a grid of size 148×148 with grid size $100 \times 100\ \text{m}^2$. We divide each edge into two sections. **Figure 7A** shows the two most optimal paths in this system. In **Figure 7B**, we show the five most optimal paths. We see in **Figure 7B** that several of the five optimal paths overlap considerably, creating an impression that there are fewer paths in the figure than there is in reality.

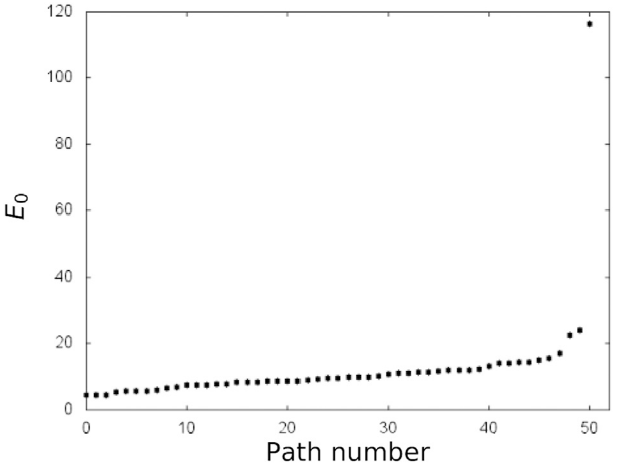


FIGURE 10 | The weight E_O ranging from the most optimal (No. 0) to the least optimal (No. 50) for paths between intervals on different edges as shown in **Figure 9**.

Figure 8 shows the ordered sequence of weights E'_O for the optimal paths in **Figure 7**. We see that the weights of the first five paths is quite similar, whereas from the sixth and onwards, it is significantly higher. Generating this flat region is accomplished by adjusting β and it signifies that these optimal paths are equally good.

We do the same construction as in **Figure 7** for area *B* in **Figure 9**. We divide the edges into three intervals and choose the values $\alpha = 0.08$ and $\beta = 0.8$ for the two adjustable parameters.

Figure 10 shows the ascending sequence of renormalized weights E'_O for the two cases shown in **Figures 9A,B**. We find 48 optimal paths with slowly increasing weights before it jumps to a much higher value.

Figure 9 shows the same area as in **Figure 5**. We have here divided the edges into three intervals. The weight of the 50 most optimal paths is shown if **Figure 10**. As in the much simpler pucture in **Figure 7**, there is also in this case considerable overlap between the optimal paths.

4 DISCUSSION AND CONCLUSION

We have in this paper introduced a method to predict motion in an area based on earlier motion in the same region. That is, given the history of traffic in the area, what would be the most likely path a new traveler would take between positions *A* and *B*, even if these positions are outside the usual routes of travel in the area. The method is based on the concept of optimal paths through a landscape formed by the paths taken earlier. It is a dynamic method as each new trajectory taken in the area is added to the history.

We have tested the method on marine Automatic Identification System (AIS) data. From a visual point of view, the method locates the motion patterns efficiently in both the simple case we studied (*A*) and in the more complex traffic picture (*B*). However, a proper performance test has not been performed. Further, the results showed that sectioning the edges into only a few intervals, were enough to generate a good estimate of the pattern. As all the grid nodes along the edges, may be represented by only a few intervals, with a short running time of the algorithm. The grid dimension and size of the cells does not seem to influence the results that are found.

This work shows that the method we propose manages to identify sensible paths that optimize between path length and frequency of use—two seemingly very different quantities. In order to turn this into a *practical* method, much more work is needed, e.g. with respect to the cluster identification, grid construction, type of vessel (if it is to be implemented as a marine tool).

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding author.

AUTHOR CONTRIBUTIONS

MF did the numerical work based on an idea by AH. MF write the first draft of the manuscript. AH wrote the second version.

FUNDING

This work was partly supported by the Research Council of Norway through its Centers of Excellence funding scheme, project number 262644.

REFERENCES

1. Hirakawa T, Yamashita T, Tamaki T, Fujiyoshi H, Umezu Y, Takeuchi I, et al. Can AI Predict Animal Movements? Filling Gaps in Animal Trajectories Using Inverse Reinforcement Learning. *Ecosphere* (2018) 9:e02447. doi:10.1002/ecs2.2447
2. Pallotta G, Vespe M, and Bryan K. Vessel Pattern Knowledge Discovery from AIS Data: A Framework for Anomaly Detection and Route Prediction. *Entropy* (2013) 15:2218–45. doi:10.3390/e15062218
3. Vasquez D, and Fraichard T. Motion Prediction for Moving Objects: A Statistical Approach. *Proc ICRA'04, IEEE* (2004) 4:3931–6. doi:10.1109/ROBOT.2004.1308883
4. Martinez J, Black MJ, and Romero J. On Human Motion Prediction Using Recurrent Neural Networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017). p. 2891–900.
5. González D, Pérez J, Milanés V, and Nashashibi F. A Review of Motion Planning Techniques for Automated Vehicles. *IEEE Trans Intell Transport Syst* (2016) 17:1135–45. doi:10.1109/TITS.2015.2498841
6. Zhu Q. A Stochastic Algorithm for Obstacle Motion Prediction in Visual Guidance of Robot Motion. In: IEEE International Conference on Systems Engineering. IEEE (1990). doi:10.1109/ICSYSE.1990.203136
7. Weiming Hu W, Xuejuan Xiao X, Zhouyu Fu Z, Xie D, Tieniu Tan T, and Maybank S. A System for Learning Statistical Motion Patterns. *IEEE Trans Pattern Anal Mach Intell* (2006) 28:1450–64. doi:10.1109/TPAMI.2006.176
8. Elfes A. Using Occupancy Grids for mobile Robot Perception and Navigation. *Computer* (1989) 22:46–57. doi:10.1109/2.30720
9. Tadokoro S, Hayashi M, Manabe Y, Nakami Y, and Takamori T. Motion Planner of mobile Robots Which Avoid Moving Human Obstacles on the Basis of Stochastic Prediction. *Intell Syst 21st Century* (1995) 4:3286–91. doi:10.1109/ICSMC.1995.538292
10. Kruse E, Gutsche R, and Wahl FM. Estimation of Collision Probabilities in Dynamic Environments for Path Planning with Minimum Collision Probability. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS'96, 3 (1996). p. 1288–95. doi:10.1109/IROS.1996.568983
11. Tanaka K. Detecting Collision-free Paths by Observing Walking People. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, 1 (2002). p. 55–60. doi:10.1109/IRDS.2002.1041362
12. Bennewitz M, Burgard W, and Thrun S. Learning Motion Patterns of Persons for mobile Service Robots. In: Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292), 4 (2002). p. 3601–6. doi:10.1109/ROBOT.2002.1014268
13. Halpin-Healy T, and Zhang Y-C. Kinetic Roughening Phenomena, Stochastic Growth, Directed Polymers and All that. Aspects of Multidisciplinary Statistical Mechanics. *Phys Rep* (1995) 254:215–414. doi:10.1016/0370-1573(94)00087-J
14. Alava M, Duxbury P, Moukarzel C, and Rieger H. Combinatorial Optimization and Disordered Systems. In: C Domb and JL Lebowitz, editors. *Phase Transitions and Critical Phenomena*, Vol. 18. Amsterdam: Elsevier (2000).
15. Alur R, La Torre S, and Pappas GJ. Optimal Paths in Weighted Timed Automata. In: *International Workshop on Hybrid Systems: Computation and Control*, 49–62. Berlin: Springer (2001). p. 49–62. doi:10.1007/3-540-45351-2_8
16. Xia Q. Optimal Paths Related to Transport Problems. *Commun Contemp Math* (2003) 05:251–79. doi:10.1142/S021919970300094X
17. Gendreau M, Ghiani G, and Guerriero E. Time-dependent Routing Problems: A Review. *Comput Operations Res* (2015) 64:189–97. doi:10.1016/j.cor.2015.06.001
18. Yu J, and LaValle SM. Planning Optimal Paths for Multiple Robots on Graphs. In: IEEE International Conference on Robotics and Automation (2013). p. 3612–7. doi:10.1109/ICRA.2013.6631084
19. Hansen A, and Kertész J. Phase Diagram of Optimal Paths. *Phys Rev Lett* (2004) 93:040601. doi:10.1103/PhysRevLett.93.040601

20. Soleimani BH, De Souza EN, Hilliard C, and Matwin S. Anomaly Detection in Maritime Data Based on Geometrical Analysis of Trajectories. In: 18th International Conference on Information Fusion (Fusion) (2015). p. 1100–5.
21. Talon L, Auradou H, Pessel M, and Hansen A. Geometry of Optimal Path Hierarchies. *EPL* (2013) 103:30003. doi:10.1209/0295-5075/103/30003
22. Ford LR, Jr. Network Flow Theory, *No. P–923*. Rand Corp Santa Monica Ca (1956).
23. Bellman R. On a Routing Problem. *Quart Appl Math* (1958) 16:87–90. doi:10.1090/qam/102435
24. Dijkstra EW. A Note on Two Problems in Connexion with Graphs. *Numer Math* (1959) 1:269–71. doi:10.1007/bf01386390

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Medicine

BRIEF RESEARCH REPORT
published: 02 August 2021
doi: 10.3389/fmed.2021.655686

Developing Machine Learning Algorithms to Predict Pulmonary Complications After Emergency Gastrointestinal Surgery

Qiong Xue[1†], Duan Wen[1†], Mu-Huo Ji[1,2], Jianhua Tong[1], Jian-Jun Yang[1] and Cheng-Mao Zhou[1*]*

[1] *Department of Anesthesiology, Pain and Perioperative Medicine, The First Affiliated Hospital of Zhengzhou University, Zhengzhou, China,* [2] *Department of Anesthesiology, The Second Affiliated Hospital, Nanjing Medical University, Nanjing, China*

OPEN ACCESS

Edited by:
Axel Hutt,
Inria Nancy - Grand-Est Research Centre, France

Reviewed by:
Ulrich Parlitz,
Max Planck Society (MPG), Germany
Meysam Hashemi,
INSERM U1106 Institut de Neurosciences des Systèmes, France

Correspondence:
Cheng-Mao Zhou
zhouchengmao187@foxmail.com
Jian-Jun Yang
yjyangjj@126.com

[†] *These authors have contributed equally to this work*

Specialty section:
This article was submitted to Translational Medicine, a section of the journal Frontiers in Medicine

Received: *19 January 2021*
Accepted: *12 July 2021*
Published: *02 August 2021*

Citation:
Xue Q, Wen D, Ji M-H, Tong J, Yang J-J and Zhou C-M (2021) Developing Machine Learning Algorithms to Predict Pulmonary Complications After Emergency Gastrointestinal Surgery. Front. Med. 8:655686. doi: 10.3389/fmed.2021.655686

Objective: Investigate whether machine learning can predict pulmonary complications (PPCs) after emergency gastrointestinal surgery in patients with acute diffuse peritonitis.

Methods: This is a secondary data analysis study. We use five machine learning algorithms (Logistic regression, DecisionTree, GradientBoosting, Xgbc, and gbm) to predict postoperative pulmonary complications.

Results: Nine hundred and twenty-six cases were included in this study; 187 cases (20.19%) had PPCs. The five most important variables for the postoperative weight were preoperative albumin, cholesterol on the 3rd day after surgery, albumin on the day of surgery, platelet count on the 1st day after surgery and cholesterol count on the 1st day after surgery for pulmonary complications. In the test group: the logistic regression model shows AUC = 0.808, accuracy = 0.824 and precision = 0.621; Decision tree shows AUC = 0.702, accuracy = 0.795 and precision = 0.486; The GradientBoosting model shows AUC = 0.788, accuracy = 0.827 and precision = 1.000; The Xgbc model shows AUC = 0.784, accuracy = 0.806 and precision = 0.583. The Gbm model shows AUC = 0.814, accuracy = 0.806 and precision = 0.750.

Conclusion: Machine learning algorithms can predict patients' PPCs with acute diffuse peritonitis. Moreover, the results of the importance matrix for the Gbdt algorithm model show that albumin, cholesterol, age, and platelets are the main variables that account for the highest pulmonary complication weights.

Keywords: machine learning, pulmonary complications, diffuse peritonitis, predict, AUC

INTRODUCTION

Complex intra-abdominal infections may result in localized or diffuse peritonitis (1). Thus, early prognostic assessment and testing for diffuse peritonitis is essential for assessing disease severity and optimizing treatment (2). Studies have shown that the mortality rate for patients with diffuse peritonitis is 9% (3).

Postoperative pulmonary complications (PPCs) are a major cause of morbidity after upper abdominal surgery, as they lengthen hospital stays and increase medical costs (4). PPC refers to postoperative pulmonary abnormalities with clinical manifestations and negative effects on disease progression with an incidence of 10–30%. Examples of PPCs include atelectasis, pulmonary infections, pleural effusion, and pulmonary thromboembolism (5–7). In clinical application, vital capacity is used for risk assessment of pulmonary complications, but critical state of lung function cannot predict complications. Moreover, there are a variety of interventions which prevent pulmonary complications, including a pre-emptive strategy to optimize respiratory physiology, and interventions during and after surgery. However, due to multiple factors pertaining to strategy, it is impossible to confirm which part of the intervention is the most important. Therefore, there is an urgent need for more effective measures and new technologies for predicting and preventing postoperative pulmonary complications.

Recent years have seen a growing body of research on machine learning and perioperative medicine (8–11). Machine Learning (ML) methods can predict clinical outcomes better than traditional statistical methods (12). For example, Fei et al. (13) used clinical data on severe acute pancreatitis (SAP) to construct Artificial Neural Network (ANN) and logistic regression models. Nijbroek et al. have argued that machine learning can support the development of more powerful PPC prediction models (14). However, studies have shown that machine learning has no performance advantage over logistic regression for clinical prediction models (15).

The present study explores the use of machine learning to improve the prediction of postoperative pulmonary complications in patients with diffuse peritonitis.

MATERIALS AND METHODS

Patients

Ethics Committee Approval and Consent to Participate

This is a secondary data analysis using database data. Data are available from the BioStudies (public) database (https://www.ebi.ac.uk/biostudies/studies?query=S-EPMC6034864).

In accordance with local laws and institutional requirements, ethical review and approval was not required for this study on human participants. In accordance with national laws and institutional requirements, written informed consent was not required from patients to participate in this study.

The data included medical records from critically ill patients who had received emergency gastrointestinal surgery for diffuse peritonitis.

Perioperative Variables

The following variables are included in the analysis: body mass index, sex, age, ASA score, lesion location, diagnosis, perioperative shock, preoperative laboratory findings, postoperative complications (3), and type of surgery.

Machine Learning Algorithms

The aim of classification by logistic regression is to establish a regression formula to classify boundary lines based on existing data. Logistic regression is a linear fit of a response variable to a logarithmic probability ratio. The coefficients are obtained by maximum likelihood estimation. The intuitive meaning of the maximum likelihood is that a pair of estimates for B0 and B1 is needed to predict the probability of the observations they produce, as close as possible to the actual observation of Y (the likelihood). A linear regression model is expressed as an equation that calculates a particular weight (i.e., coefficient b) for the input variable, and then describes a straight line that best fits the relationship between the input variable (x) and the output variable (y). For example: $y = B0 + B1 * x$.

Decision tree learning is a decision model that incorporates data attributes into a tree structure. Decision trees are often constructed based on a given dataset (16). Decision tree algorithm is a method of approximating discrete function values. It is a typical classification method, which first processes the data, generates readable rules and decision trees using inductive algorithms, and then uses the decisions to analyze the new data. In short, decision tree is a process for classifying data based on a series of rules.

The Gradient Boosting Decision Tree (GBDT) method (17) is used for data bulletins, to create M models (such as classification). This model is simple, and it is referred to as a weak learner. For each classification, the weight of the data incorrectly divided the previous time is increased one point before classification. In this way, the final classifier can produce good results, for both test data and training data.

Lightgbm (gbm) is another implementation of GBDT (18). Based on GBDT, it adopts two new strategies.

Our analysis was conducted with R version 3.1.3 (http://www.R-project) and Python version 3.6 (Python Software Foundation). We used five machine learning algorithms (Logistic regression, DecisionTree, GradientBoosting, Xgbc, and gbm) (19, 20) to predict postoperative pulmonary complications. We randomly divided all samples into training and test groups at a ratio of 7:3 using 5-fold cross-validation. We performed the 5-fold cross-validation in the training group, and then obtained its optimal model and parameters, and applied them to the test group. Five-fold cross-validation is a data splitting strategy for cross-validation, that is, the data set is split into A data set and B data set. The principle is: First, the whole training data set is divided into 5-folds, where 4-folds are used as the A data set to train the model, and the remaining 1-fold is used as the B data set to score the model, and the above process is repeated five times. In the weighted correlation analysis, we ranked the variables from highest to lowest weighted scores accounting for pulmonary complications. We did this using Pearson correlation analysis. The variables' missing values are supplemented by multiple imputation. The values were normalized and scaled 0–1. ROC is an abbreviation for "receiver operating characteristic." The ROC curve's area is the AUC (Area Under the Curve). AUC (area under the ROC curve), i.e., the area under the ROC curve, the larger the better, indicating that the model had higher prediction value: (1) AUC≈1.0: the most ideal test index; (2) AUC is within 0.7–0.9:

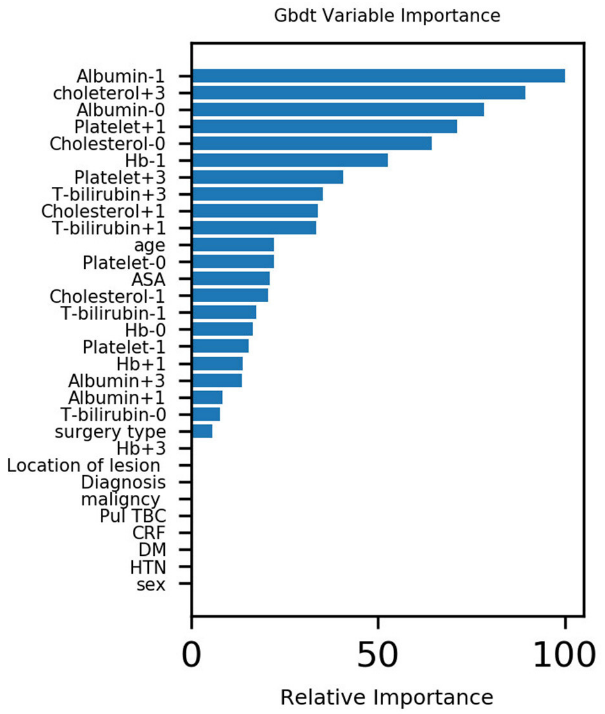


FIGURE 1 | Variable importance of features included in machine learning algorithms for predicting pulmonary complications.

the model has high accuracy; (3) AUC ≤ 0.5: the model has no predictive value. We can only provide code that runs out of the results portion of the algorithm because of the patent application issues involved. See **Appendix 1** for specific codes.

RESULTS

Nine hundred and twenty-six cases were included in this study; 187 cases (20.19%) had postoperative pulmonary complications. The average age of the patients with postoperative pulmonary complications was 65.6 (± 14.5) years old. One hundred and twenty-six (67.4%) were males with postoperative pulmonary complications, and 61 (32.6%) were females with postoperative pulmonary complications (see **Supplementary Table 1**).

The five most important variables for the postoperative weight were preoperative albumin, cholesterol on the 3rd day after surgery, albumin on the day of surgery, platelet count on the 1st day after surgery and cholesterol count on the 1st day after surgery for pulmonary complications. The correlation heat map showed that platelets, cholesterol, and albumin were negatively correlated with pulmonary complications (see **Figures 1**, **2**).

Supplementary Table 2 and **Figure 3** present the results of the machine learning algorithm in the training group. The logistic regression model shows that AUC = 0.836, accuracy = 0.826 and precision = 0.625; Decision tree shows AUC = 0.782, accuracy = 0.821 and precision = 0.563; The GradientBoosting model shows AUC value = 0.853, accuracy = 0.824 and precision = 0.947; The Xgbc model shows AUC = 0.835, accuracy = 0.833 and precision = 0.897. The Gbm model shows AUC = 0.856, accuracy = 0.816 and precision = 0.929.

Supplementary Table 2 and **Figure 4** present the results of the machine learning algorithm in the test group. The logistic regression model shows AUC = 0.808, accuracy = 0.824 and precision = 0.621; Decision tree shows AUC = 0.702, accuracy = 0.795 and precision = 0.486; The GradientBoosting model shows AUC = 0.788, accuracy = 0.827 and precision = 1.000; The Xgbc model shows AUC = 0.784, accuracy =0.806 and precision = 0.583. The Gbm model shows AUC = 0.814, accuracy = 0.806 and precision = 0.750.

DISCUSSION

Postoperative pulmonary complications (PPCs) often occur after major surgery (18). Any PPC, even if it is "mild," is associated with increased long-term hospitalization and hospital mortality (21). Thus, PPC predictions have the potential to optimize care for individual patients, normalize the use of scarce resources, and may even enrich research populations for testing PPC treatments' effects. Machine learning, using methods such as "unbiased cluster analysis" and biophenotypic analysis, likely improves PPC prediction models (22). The results of the present study show that machine learning algorithms can predict the postoperative pulmonary complications of patients with acute diffuse peritonitis.

Serum albumin levels represent a patient's nutritional status. Studies have shown that preoperative hypoproteinemia increases the incidence of postoperative abdominal complications (23). Other studies have shown that perioperative changes in serum albumin are predictors of lung complications in patients with lung cancer and laparoscopic gastrectomy (24, 25). Our results also indicate that changes in perioperative albumin are associated with changes in postoperative pulmonary complications. In addition, when ASA grade ≥ 3 and BMI is low, the incidence of pulmonary complications after early lung cancer radical surgery may increase (26). The present study's results also indicate that the ASA score is directly proportional to the occurrence of pulmonary complications.

Elevated levels of high-density lipoprotein cholesterol may be associated with decreased lung function in healthy male adolescents (27). Moreover, lower serum cholesterol levels are a poor prognostic factor in patients with severe community-acquired pneumonia (28). High cholesterol/high fat diet-induced hypercholesterolemia may result in lower respiratory inflammation associated with TLRs/NFκB pathway in C57BL/6J mice (29). Our study also supports this view, in that there is an inverse relationship between cholesterol and postoperative pulmonary complications.

Red blood cell distribution width and platelet count are biomarkers of pulmonary hypertension in patients with connective tissue disease (30). Also, platelet activating factor receptor regulates lung inflammation caused by colitis by NLRP3 inflammation (31). Platelets are factors in lung development in mice through Clec-2/podoplanin interactions (32). Our study

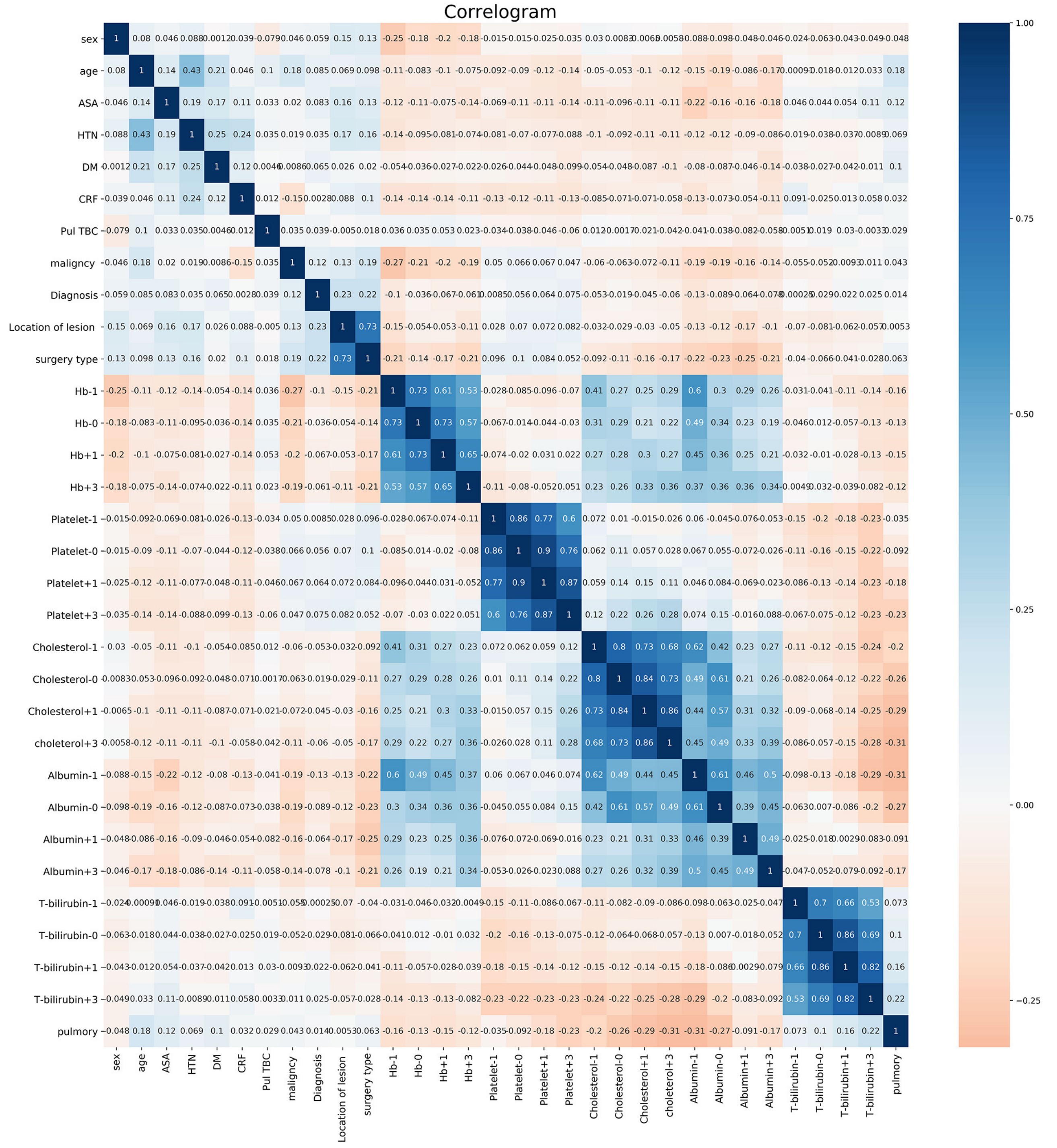


FIGURE 2 | Correlation analysis of various factors.

also shows that perioperative platelet changes are a factor in the occurrence of PPCs.

Hemoglobin $\leq$ 100 g/L is an independent risk factor for postoperative pulmonary complications (33). Similarly, studies have suggested that serum albumin reduction on the 1st day after surgery can be a predictor of PPCs in patients with lung cancer after thoracoscopic anatomy (24). Our results also suggest that hemoglobin is an important contributor to postoperative complications, and that the two are inversely proportional.

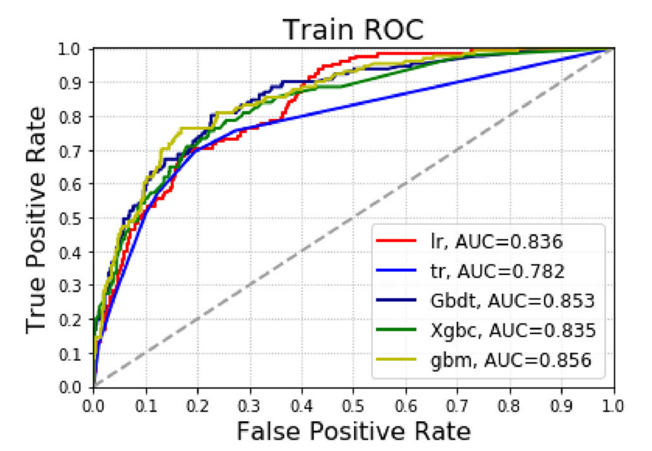


FIGURE 3 | AUC for predicting pulmonary complications using various machine learning algorithms in the training group.

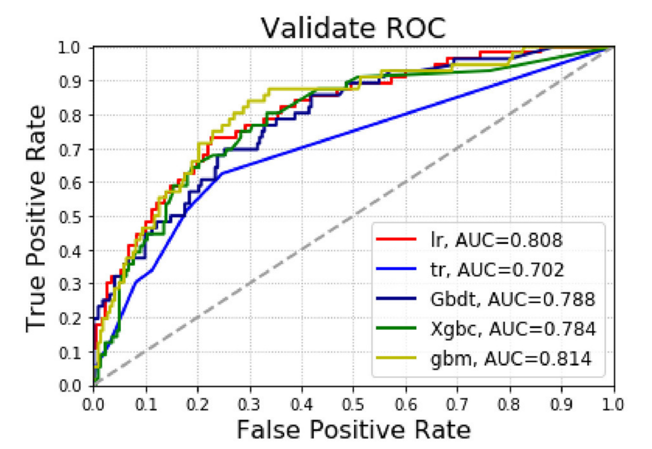


FIGURE 4 | The AUC for predicting pulmonary complications using various machine learning algorithms in the test group.

This study has several limitations. First, because it was a retrospective study, the selected patient entry training and test datasets did not meet the predictions for the "future" cohort results. Thus, we needed to build a stable model to predict future postoperative pulmonary complications. In addition, the variables involved in this study may be insufficient. Future research should incorporate etiology and include more relevant influencing factors for analysis and research. Moreover, due to the limited predictive utility in our study, especially the ML algorithm's lower recall rate, there were several difficulties in applying the ML model in a clinical setting. However, with improved accuracy, this study's results are still reliable. Finally, the training sample size is still limited, as the cohort is from only one center. A multi-center prospective study is needed for training and validation in the future.

In sum, machine learning algorithms can predict the PPCs of patients with acute diffuse peritonitis. In future studies, specific machine learning models could be trained with a larger cohort of patients with acute diffuse peritonitis.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/**Supplementary Material**, further inquiries can be directed to the corresponding author/s.

ETHICS STATEMENT

Ethical review and approval was not required for the study on human participants in accordance with the local legislation and institutional requirements. Written informed consent for participation was not required for this study in accordance with the national legislation and the institutional requirements.

AUTHOR CONTRIBUTIONS

QX, DW, C-MZ, and J-JY were major contributors in writing the manuscript. All authors analyzed the data.

ACKNOWLEDGMENTS

We are grateful to Professor Lee for disclosing his data in the BioStudies (public) database (3).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fmed.2021.655686/full#supplementary-material

REFERENCES

1. Sartelli M, Catena F, Balogh Z, Bendinelli C, Gupta S, Kluger Y, et al. Complicated intra-abdominal infections in a worldwide context: an observational prospective study (CIAOW Study) | NOVA. The University of Newcastle's Digital Repository. *World J Emerg Surg.* (2013) 8:1–7. doi: 10.1186/1749-7922-8-1
2. Sartelli M. A focus on intra-abdominal infection. *World J Emerg Surg.* (2010) 5:9. doi: 10.1186/1749-7922-5-9
3. Lee SH, Lee JY, Hong TH, Kim BO, Lee YJ, Lee JG. Severe persistent hypocholesterolemia after emergency gastrointestinal surgery predicts in-hospital mortality in critically ill patients with diffuse peritonitis. *PLoS ONE.* (2018) 13:e0200187. doi: 10.1371/journal.pone.0200187
4. Smetana GW, Lawrence VA, Cornell JE. Preoperative pulmonary risk stratification for noncardiothoracic surgery: systematic review for the American College of Physicians. *Ann Intern Med.* (2006) 144:581–95. doi: 10.7326/0003-4819-144-8-200604180-00009

5. Overend TJ, Anderson CM, Lucy SD, Bhatia C, Jonsson BI, Timmermans C. The effect of incentive spirometry on postoperative pulmonary complications: a systematic review. *Chest.* (2001) 120:971–8. doi: 10.1378/chest.120.3.971
6. Canet J, Gallart L, Gomar C, Paluzie G, Vallès J, Castillo J, et al. Prediction of postoperative pulmonary complications in a population-based surgical cohort. *Anesthesiology.* (2010) 113:1338–50. doi: 10.1097/ALN.0b013e3181fc6e0a
7. Weingarten TN, Kor DJ, Gali B, Sprung J. Predicting postoperative pulmonary complications in high-risk populations. *Curr Opin Anaesthesiol.* (2013) 26:116–25. doi: 10.1097/ACO.0b013e32835e21d2
8. Neto AS, Bos LD, Campos P, Hemmes S, Schultz MJ. Association between pre-operative biological phenotypes and postoperative pulmonary complications: an unbiased cluster analysis. *Eur J Anaesthesiol.* (2018) 35:1. doi: 10.1097/EJA.0000000000000846
9. Shameer K, Johnson KW, Glicksberg BS, Dudley JT, Sengupta PP. Machine learning in cardiovascular medicine: are we there yet? *Heart.* (2018) 104:1156–64. doi: 10.1136/heartjnl-2017-311198
10. Bibault JE, Giraud P. Burgun Anita. Big Data and machine learning in radiation oncology: state of the art and future prospects. *Cancer Lett.* (2016) 382:110–7. doi: 10.1016/j.canlet.2016.05.033
11. Zhou CM, Xue Q, Wang Y, Tong J, Ji M, Yang JJ. Machine learning to predict the cancer-specific mortality of patients with primary non-metastatic invasive breast cancer. *Surg Today Surg Today.* (2021) 51:756–63. doi: 10.1007/s00595-020-02170-9
12. Zhou ZH. *Ensemble Methods - Foundations and Algorithms*. Boca Raton, FL: Taylor & Francis Group (2012) p. 77–9. doi: 10.1201/b12207
13. Fei Y, Gao K, Li WQ. Artificial neural network algorithm model as powerful tool to predict acute lung injury following to severe acute pancreatitis. *Pancreatology*. (2018) 18:892–9. doi: 10.1016/j.pan.2018.09.007
14. Nijbroek SG, Schultz MJ, Hemmes SNT. Prediction of postoperative pulmonary complications. *Curr Opin Anaesthesiol.* (2019) 32:443–51. doi: 10.1097/ACO.0000000000000730
15. Christodoulou E, Ma J, Collins GS, Steyerberg EW, Verbakel JY, Van CB, et al. systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models. *J Clin Epidemiol.* (2019) 110:12–22. doi: 10.1016/j.jclinepi.2019.02.004
16. Praagman J. Classification and regression trees: Leo Breiman, Jerome H. Friedman, Richard A. Olshen and Charles J. Stone. The Wadsworth Statistics/Probability Series, Wadsworth, Belmont, 1984, x + 358 pages. *Eur J Operation Res*. (1985) 19:144. doi: 10.1016/0377-2217(85)90321-2
17. Liao Z, Yong H, Yue X, Lu H, Ying J. *In silico* prediction of gamma-aminobutyric acid type-a receptors using novel machine-learning-based SVM and GBDT approaches. *Biomed Res Int.* (2016) 2016:2375268. doi: 10.1155/2016/2375268
18. Zhang J, Mucs D, Norinder U, Svensson F. LightGBM: an effective and scalable algorithm for prediction of chemical toxicity-application to the Tox21 and mutagenicity data sets. *J Chem Inf Model.* (2019) 59:4150–8. doi: 10.1021/acs.jcim.9b00633
19. Zhou CM, Wang Y, Ye HT, Yan S, Ji M, Liu P, et al. Machine learning predicts lymph node metastasis of poorly differentiated-type intramucosal gastric cancer. *Sci Rep.* (2021) 11:1300. doi: 10.1038/s41598-020-80582-w
20. Zhou C, Hu J, Wang Y, Ji MH, Tong J, Yang JJ, et al. A machine learning-based predictor for the identification of the recurrence of patients with gastric cancer after operation. *Sci Rep.* (2021) 11:1571. doi: 10.1038/s41598-021-81188-6
21. LAS VEGAS Investigators. Epidemiology, practice of ventilation and outcome for patients at increased risk of postoperative pulmonary complications: LAS VEGAS - an observational study in 29 countries. *Eur J Anaesthesiol.* (2017) 34:492–507. doi: 10.1097/EJA.0000000000000646
22. Serpa NA, Hemmes SN, Barbas CS, Beiderlinden M, Fernandez-Bustamante A, Futier E, et al. Incidence of mortality and morbidity related to postoperative lung injury in patients who have undergone abdominal or thoracic surgery: a systematic review and meta-analysis. *Lancet Respir Med.* (2014) 2:1007–15. doi: 10.1016/S2213-2600(14)70228-0
23. Rock P, Rich PB. Postoperative pulmonary complications. *Curr Opin Anaesthesiol.* (2003) 16:123–31. doi: 10.1097/00001503-200304000-00004
24. Li P, Li J, Lai Y, Wang Y, Wang X, Su J, et al. Perioperative changes of serum albumin are a predictor of postoperative pulmonary complications in lung cancer patients: a retrospective cohort study. *J Thorac Dis.* (2018) 10:5755–63. doi: 10.21037/jtd.2018.09.113
25. Chen Y, Wu G, Wang R, Chen J. Preoperative albumin level serves as a predictor for postoperative pulmonary complications following elective laparoscopic gastrectomy. *Curr Pharm Des.* (2018) 24:3250–5. doi: 10.2174/1381612824666180713104307
26. Im Y, Park HY, Shin S, Shin SH, Lee H, Ahn JH, et al. Prevalence of and risk factors for pulmonary complications after curative resection in otherwise healthy elderly patients with early stage lung cancer. *Respir Res.* (2019) 20:136. doi: 10.1186/s12931-019-1087-x
27. Park JH, Mun S, Choi DP, Lee JY, Kim HC. Association between high-density lipoprotein cholesterol level and pulmonary function in healthy Korean adolescents: the JS high school study. *BMC Pulm Med.* (2017) 17:190. doi: 10.1186/s12890-017-0548-6
28. Chien YF, Chen CY, Hsu CL, Chen KY, Yu CJ. Decreased serum level of lipoprotein cholesterol is a poor prognostic factor for patients with severe community-acquired pneumonia that required intensive care unit admission. *J Crit Care*. (2015) 30:506–10. doi: 10.1016/j.jcrc.2015.01.001
29. Fang Y, Wang S, Zhu T, Zhang Y, Lian X. Atherogenic high cholesterol/high fat diet induces TLRs-associated pulmonary inflammation in C57BL/6J mice. *Inflamm Res.* (2017) 66:39–47. doi: 10.1007/s00011-016-0990-6
30. Bellan M, Giubertoni A, Piccinino C, Dimagli A, Grimoldi F, Sguazzotti M, et al. Red cell distribution width and platelet count as biomarkers of pulmonary arterial hypertension in patients with connective tissue disorders. *Dis Markers.* (2019) 2019:4981982. doi: 10.1155/2019/4981982
31. Liu G, Mateer SW, Hsu A, Goggins BJ, Tay H, Mathe A, et al. Platelet activating factor receptor regulates colitis-induced pulmonary inflammation through the NLRP3 inflammasome. *Mucosal Immunol.* (2019) 12:862–73. doi: 10.1038/s41385-019-0163-3
32. Tsukiji N, Inoue O, Morimoto M, Tatsumi N, Nagatomo H, Ueta K, et al. Platelets play an essential role in murine lung development through Clec-2/podoplanin interaction. *Blood.* (2018) 132:1167–79. doi: 10.1182/blood-2017-12-823369
33. Jiang GQ, Bai DS, Chen P, Fan J, Tan JW, Peng MH. Starting hemoglobin value predicts early phase prognosis after liver transplantation. *Transplant Proc.* (2011) 43:1669–73. doi: 10.1016/j.transproceed.2010.12.067

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

ORIGINAL RESEARCH
published: 10 August 2021
doi: 10.3389/feart.2021.724599

Investigating Predictability of the TRHR Seasonal Precipitation at Long Lead Times Using a Generalized Regression Model with Regularization

Xiao Peng[1], *Tiejian Li*[2] *and John D. Albertson*[1]*

[1]School of Civil and Environmental Engineering, Cornell University, Ithaca, NY, United States, [2]Department of Hydraulic Engineering, Tsinghua University, Beijing, China

Skillful long-lead climate forecast is of great importance in managing large water systems and can be made possible using teleconnections between regional climate and large-scale circulations. Recent innovations in machine learning provide powerful tools in exploring linear/nonlinear associations between climate variables. However, while it is hard to give physical interpretation of the more complex models, the simple models can be vulnerable to over-fitting, especially when dealing with the highly "non-square" climate data. Here, as a compromise of interpretability and complexity, we proposed a regression model by coupling pooling and a generalized regression with regularization. Performance of the model is tested in estimating the Three-Rivers Headwater Region wet-season precipitation using the sea surface temperatures at lead times of 0–24 months. The model shows better predictive skill for certain long lead times when compared with some commonly used regression methods including the Ordinary Least Squares (OLS), Empirical Orthogonal Function (EOF), and Canonical Correlation Analysis (CCA) regressions. The high skill is found to relate to the persistent regional correlation patterns between the predictand precipitation and predictor SSTs as also confirmed by a correlation analysis. Furthermore, flexibility of the model is demonstrated using a multinomial regression model which shows good skill around the long lead time of 22 months. Consistent clusters of SSTs are found to contribute to both models. Two SST indices are defined based on the major clusters of predictors and are found to be significantly correlated with the predictand precipitation at corresponding lead times. In conclusion, the proposed regression model demonstrates great flexibility and advantages in dealing with collinearity while preserving simplicity and interpretability, and shows potential as a cheap preliminary analysis tool to guide further study using more complex models.

Keywords: the three-rivers headwater region, seasonal precipitation prediction, teleconnection, pooling, elastic net regression, logistic regression, correlation analysis

Edited by:
Axel Hutt,
Inria Nancy - Grand-Est research centre, France

Reviewed by:
Kelsey Barton-henry,
Potsdam Institute for Climate Impact Research (PIK), Germany
Rosmeri Porfirio Da Rocha,
University of São Paulo, Brazil

****Correspondence:***
John D. Albertson
albertson@cornell.edu

Specialty section:
This article was submitted to Atmospheric Science, a section of the journal Frontiers in Earth Science

Received: *13 June 2021*
Accepted: *30 July 2021*
Published: *10 August 2021*

Citation:
Peng X, Li T and Albertson JD (2021) Investigating Predictability of the TRHR Seasonal Precipitation at Long Lead Times Using a Generalized Regression Model with Regularization.
Front. Earth Sci. 9:724599.
doi: 10.3389/feart.2021.724599

1 INTRODUCTION

Skillful long-lead (seasonal to annual) climate forecast is of great importance in managing large water systems. Examples include but are not limited to making water transferring plans for multi-reservoir systems running at annual to inter-annual time scales (Carpenter and Georgakakos, 2001; Block, 2011), informing long-term agricultural decision making (Lemos et al., 2002; Hansen et al., 2011), and developing early warning system for disaster mitigation (Wilhite and Svoboda, 2000; Verdin et al., 2005). While local climate variability always fails to persist through such long lead times, the prediction can be made possible using long-lead teleconnections between regional climate and large-scale circulations. Anomalies of large-scale atmospheric circulations can be anchored by ocean memory due to massive heat capacity of ocean water and be released to perturb other circulations at a much later time (Xie et al., 2009; Xie et al., 2016). These perturbations can therefore be indicated by SST anomalies. There are already well-established SST-based climate indices that have seen good use in long-lead climate forecasts such as the Niño SST indices (Rasmusson and Carpenter, 1982; Trenberth, 1997; Trenberth and Stepaniak, 2001), the Pacific Decadal Oscillation (PDO) (Mantua et al., 1997; Zhang et al., 1997), the Tropical Northern Atlantic (TNA) and the Tropical Southern Atlantic (TSA) indices (Enfield et al., 1999) etc.

Approaches commonly used in developing long-term prediction models based on large-scale teleconnections can be roughly categorized into two classes: 1) physically-based simulation and 2) statistical models. While the physically-based simulation is widely used in investigating causality chains of climate processes, it is usually computationally intensive and requires expertise for parameter calibration (Menemenlis et al., 2005; Sahastrabuddhe and Ghosh, 2021). Its statistical counterpart, in the meantime, provides an easy access to examining statistical associations between climate variables which could be further used to develop prediction models. The statistical models are becoming increasingly popular thanks to the advances of sensing technology and internet which makes tremendously more data available at exceptionally high temporal and spatial resolutions (Liu, 2015). Early efforts of statistical modelling are featured by qualitative analysis comparing time series of different climate variables (Thornthwaite, 1948; Von Storch and Zwiers, 2001). Most early work relied on insights of the expert researchers and were done with data of rather limited size. Recent innovations in machine learning have developed powerful tools for examining linear/non-linear associations between climate variables in massive volumes in a more automated way. Just to name a few examples here: Kernelization is used to extend study domain from linear associations to nonlinear associations (Ali et al., 2019; Bueso et al., 2020). Data processing tips like pooling and convolution are used to enhance model robustness by discarding/smoothing noises (Devineni and Sankarasubramanian, 2010; Schepen et al., 2018). Of the many machine learning approaches, neural network has become extremely popular across a wide range of spatial scales (local-global) (Goddard et al., 2001; Mekanik et al., 2013; Fan et al., 2015; Ham et al., 2019; Reichstein et al., 2019). Ham et al. (2019) even successfully extended lead time of skillful ENSO forecast to one and a half years using a convolution neural network trained on historical simulations, which beat many state-of-the-art dynamical systems in terms of correlation skill for the Niño 3.4 Index. However, even though efforts are being made to improve model interpretability (Gilpin et al., 2018; Carvalho et al., 2019; Worland et al., 2019), tools for explaining the machine learning models are still insufficient (Gilpin et al., 2018) and to find physical interpretation of these models is usually hard or even impossible due to high model complexity. In this study, we looked at a generalized regularized regression method (i.e., elastic net) coupled with pooling as a compromise between model interpretability and complexity, and examined its performance in predicting regional seasonal precipitation based on large-scale SST anomalies.

Regression has been broadly used in climate research and related fields. Typical applications include 1) change point detection (Solow, 1987; Mudelsee, 2000), 2) developing forecast models (Krishnamurti et al., 1999; Mekanik et al., 2013; Kharin and Zwiers, 2002), and 3) identification of covariates with high predictive skill (Wakabayashi and Kawamura, 2004; Matsui and Konishi, 2011). Not only can the regression model identify the linear relationships between the climate variables at a given temporal basis (e.g., monthly or annual), but it also has good interpretability for guiding further research using more complex, nonlinear statistical methods or physically-based modeling experiments. These two features make regression especially popular in exploring teleconnections between regional climate and large-scale circulations. Hurrell (1996) used a multivariate linear regression model to link changes in northern hemisphere temperature to extratropical climate indices. Krishnamurti et al. (1999) developed a superensemble method for improving weather and climate forecast skills by using coefficients from multiple regressions. Wakabayashi and Kawamura (2004) extracted four major teleconnection patterns in predicting Japan summer climate anomalies by combining the empirical orthogonal function (EOF) and regression. Van Oldenborgh and Burgers (2005) developed a synthetic precipitation generator with regression models using the Niño 3.4 Index as the sole regressor to examine decadal variation in global ENSO-precipitation teleconnections. Yang and DelSole (2012) used the regression coefficient maps to explore teleconnections between ENSO and different climate fields. More recently, Zhang et al. (2020) examined teleconnections between the Arctic sea ice decline and major climate indices using the quantile regression analysis. Only a few examples are listed here for context, as our intent is to test a generalized regression model with regularization in long-term seasonal precipitation forecast instead of doing a thorough review applications of regression in climate research.

However, over-fitting and over-parameterization are important issues for most regression analysis. These issues are particularly pertinent in climate research, since the remote sensing data are usually highly “non-square” (i.e., the total number of time series largely exceeds the length of the time

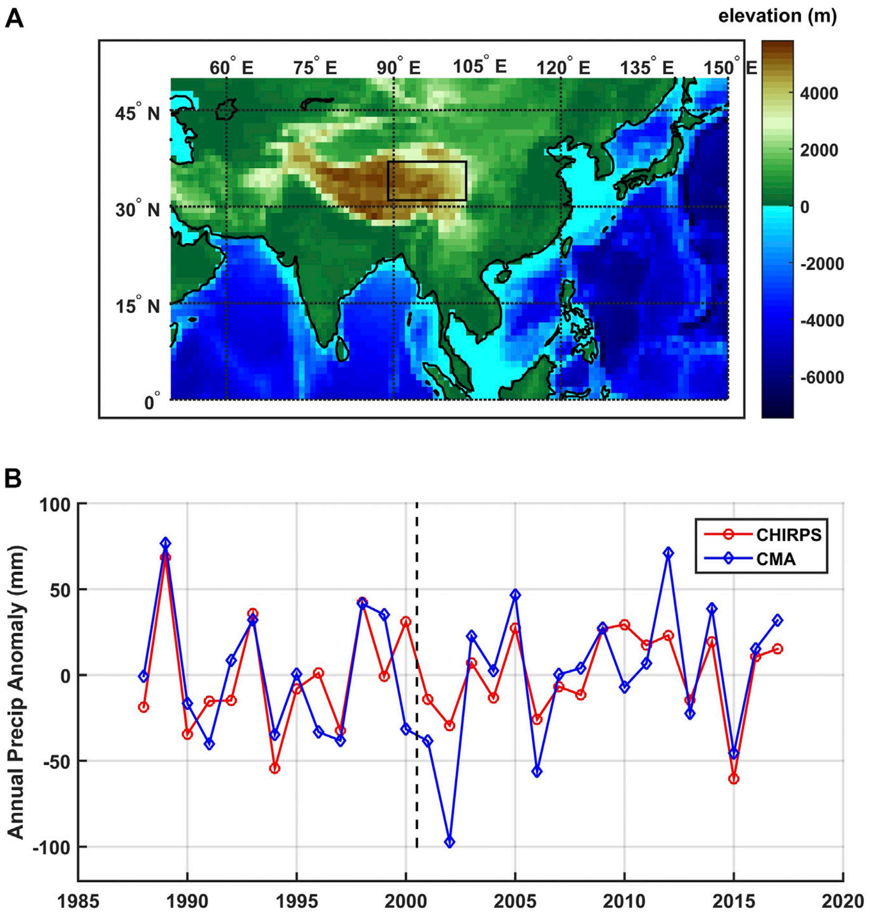


FIGURE 1 | (A) The study region of TRHR (black box) as plotted in an elevation map based on ETOPO-5 (Center, 1988) and **(B)** time series of the spatially averaged precipitation from CHIRPS (red) and CMA (blue) after standardization.

series). Therefore, it is necessary to reduce the effective dimensionality of the problems. Two commonly used approaches have been: 1) to select only a few predominant features/patterns (e.g., the Principal Component Analysis, PCA (Schoof and Pryor, 2001; Wakabayashi and Kawamura, 2004; Li et al., 2020) or the Canonical Correlation Analysis, CCA (Mo, 2003; Yang and DelSole, 2012)); 2) to use only a few well-established climate indices (Van Oldenborgh and Burgers, 2005; Rust et al., 2015; Tan and Shao, 2017; Zhang et al., 2020). However, both methods have intrinsic disadvantages: Traditional dimensionality reduction methods like PCA and CCA try to decompose the global covariance structure of the predictors (PCA) or between the predictand and the predictors (CCA) and can miss important regional patterns while the climate indices are only defined by prior knowledge and thus could limit the domain where we want to explore the potential teleconnections.

In the past few decades, regularization has become increasingly popular in dealing with multicollinearity in regression. The two regularization approaches commonly used with regression are the L-1 norm (the least absolute shrinkage and selection operator, LASSO (Tibshirani, 1996)) and the L-2 norm (the ridge regression (Hoerl and Kennard, 1970)) of regression coefficients. Other popular regularization approaches include the Akaike's Information Criterion (AIC) (Akaike, 1998) and the Bayesian Information Criterion (BIC) (Schwarz, 1978). Both L-1 and L-2 norm regularizations have shown good performance in alleviating or avoiding over-fitting in regression models in climate research (Matsui and Konishi, 2011; Soleh et al., 2015; DelSole and Banerjee, 2017; Kim et al., 2017; Li et al., 2020). Yet, it should be noted that the ridge regression does not directly provoke sparsity of the regression model while the LASSO regression tends to assign non-zero value to only one of many correlated predictors which can make the model difficult to interpret. The lack of interpretability of the lasso model is also pointed out in a recent paper from Stevens et al. (2021) where a graph-guided variation is used as an extra regularization to improve robustness of the regression model in predicting Southwestern United States winter precipitation. Here, we propose to use the elastic net regularization (Zou and Hastie, 2005) which linearly combines the LASSO and ridge regression regularizations. While the LASSO regularization guarantees sparsity of the model, the ridge regression regularization helps improve visualization of the regression coefficient map and therefore, interpretability of the model (Peng et al., 2020). On top of that, a pooling layer is

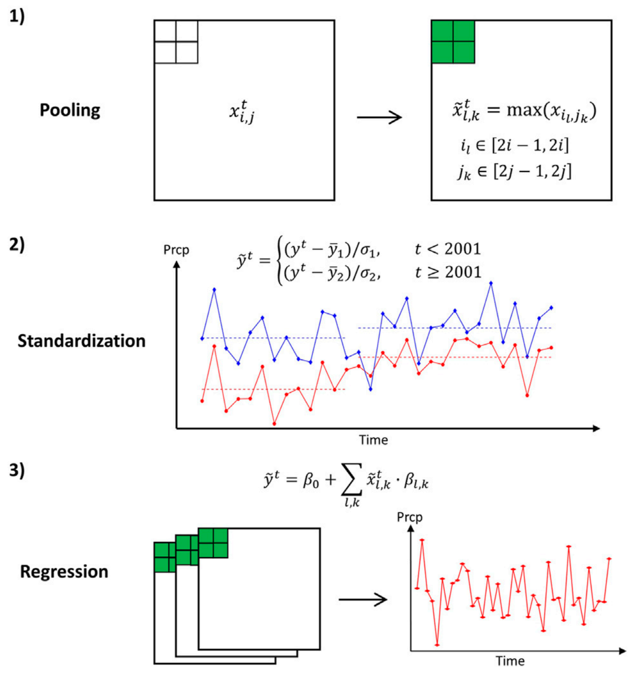

1)
Pooling
$x^t_{i,j}$
$\tilde{x}^t_{l,k} = \max(x_{i_l,j_k})$
$i_l \in [2i-1, 2i]$
$j_k \in [2j-1, 2j]$
2)
Standardization
Prcp
$\tilde{y}^t = \begin{cases} (y^t - \bar{y}_1)/\sigma_1, & t < 2001 \\ (y^t - \bar{y}_2)/\sigma_2, & t \geq 2001 \end{cases}$
Time
3)
Regression
$\tilde{y}^t = \beta_0 + \sum_{l,k} \tilde{x}^t_{l,k} \cdot \beta_{l,k}$
Prcp
Time

FIGURE 2 | A schematic of the proposed the regression model coupling max pooling and elastic net.

added before developing the regression model. The pooling layer is commonly used in machine learning for reducing spatial dimensions (Zeiler and Fergus, 2013; Yu et al., 2014; Kalchbrenner et al., 2014). And this extra pooling layer should help improve robustness of the model by avoiding the realistic problem that major "hot" regions defining large-scale circulations are not fixed to certain spatial grids naturally.

The proposed model is tested to predict the Three-Rivers Headwater Region (TRHR) wet-season precipitation using the Pacific Ocean and Indian Ocean SSTs. The TRHR, located in the eastern Tibetan Plateau (TP), is often called China's Water Tower as from it flow the three major rivers of China: the Yellow River, the Yangtze River, and the Lancang (Mekong) River. Consequently, the TRHR plays a critical role in providing invaluable ecological goods and serviced as well as other resources like energy and food. While great efforts have been devoted to studying teleconnections between the broader TP precipitation and large-scale climates (Benn and Owen, 1998; Shaman and Tziperman, 2005; Feng and Zhou, 2012; Dong et al., 2020), the quantitative studies focusing solely on the TRHR are rather limited and only looks at short lead times (Zhang et al., 2019; Zhao et al., 2019). In this study, we extend the forecast lead time up to 24 months and the performance are compared against some widely-used regression methods including OLS multi-linear regression, the EOF regression, and the CCA regression. The precipitation is predicted in true amplitudes and binary states (wetter or drier than normal) to demonstrate flexibility of the model. In this study, we seek a model that is computationally tractable for fast decision making support for stakeholders while retaining a relatively direct physical interpretation to aid further investigation of the underlying physical processes.

2 DATA

We base our analysis on monthly precipitation data from Jan 1981 through Dec 2019 as collected from the Climate Hazards Group InfraRed Precipitation with Station data (CHIRPS). The original gridded precipitation data incorporates satellite data with *in-situ* station data, with a resolution of 0.05° by 0.05° (Funk et al., 2015). In this study, we spatially averaged the TRHR precipitation over a rectangular area masking 89°E to 103°E and 31°N to 37°N as shown in **Figure 1A**. The monthly climatology for the spatially averaged precipitation is monomodal showing that over 80% of the annual precipitation falls during the 5-month period of May-

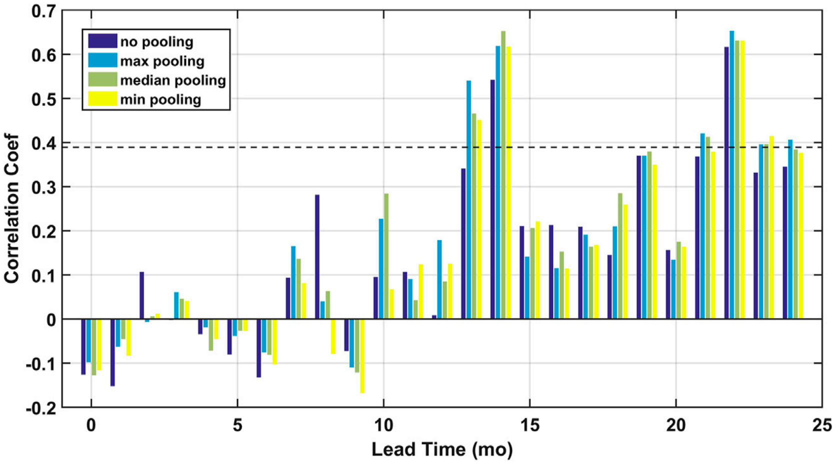


FIGURE 3 | Comparison of the predictive skill in the testing period from the elastic net regression models with no pooling (dark blue), maximum pooling (light blue), median pooling (green) and minimum pooling (yellow). The *p*-value = 0.1 significance level is plotted in the black dashed line.

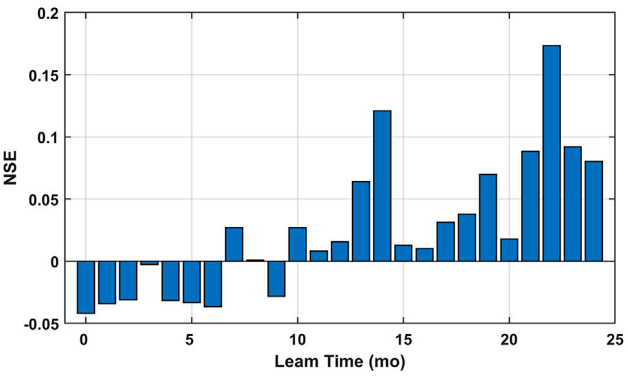


FIGURE 4 | Predictive skill for the testing set as a function of lead time reported in the NSE scores.

Sept, which we define as the wet season in this study (also known as the growing season for the TRHR (Chen et al., 2020)).

The CHIRPS precipitation is double checked against the station-based precipitation (1988–2017) collected from the China Meteorological Administration (CMA). 55 stations with missing data ratio lower than 20% within the study region are selected and the arithmetically averaged precipitation time series is compared against that from the CHIRPS precipitation. A systematic shift is observed around 2000 for both precipitation though the shift is less significant for the station-based precipitation. While difference in the shifts can be due to non-uniform distribution of the CMA stations, we do not want to diverge into this topic. Instead, the shift is removed by separately standardizing the precipitation over 1981–2000 and 2001–2019. The standardization is done by subtracting the mean and dividing by the standard deviation. The standardized CHIRPS precipitation shows good consistency with the station-based precipitation as shown in **Figure 1B**, and correlation is 0.67 for 39 samples (p-value < 0.01). The binary TRHR precipitation is used in the multinomial regression model and the two states are defined as: 0 or dry for standardized precipitations smaller than 0 and 1 or wet for standardized precipitations greater than 0.

SST is selected as the primary predictor since it can indicate perturbations in large-scale atmospheric circulations “anchored” in ocean memory (Xie et al., 2009, Xie et al., 2016). Also, the SST field is less spatially heterogeneous compared to that of other common climate variables including geopotential height, vertical velocity of atmosphere (OMEGA) and wind velocities (Peng et al., 2020), which can help improve robustness of the regression models. Monthly SST data is collected from the Hadley Centre Sea Ice and Sea Surface Temperature (HadISST) data set with a spatial resolution of 1° by 1° (Rayner et al., 2003) over Jan 1979–Dec 2019. Only SSTs from the Pacific Ocean and Indian Ocean basins are used to limit our study to regional processes and the basin range is based on the definitions from the National Oceanic and Atmospheric Administration (NOAA) via https://www.nodc.noaa.gov/woce/woce_v3/wocedata_1/woce-uot/summary/bound.htm. Similar positive shifts are observed for most parts of the Pacific Ocean and Indian Ocean as seen in **Supplementary Figure S1**. To be consistent with the standardization of precipitation, the SSTs are too standardized separately for 1979–2000 and 2001–2019 to remove effects of the trends. This step is to ensure that model skill as measured by the correlation coefficient in the later sections will not be biased by the trends. The only difference here is that standardization of the SSTs is done locally for each grid and uses the monthly climatology means and standard deviations to remove the seasonal cycle.

3 METHODS

3.1 The Regularized Regression

Here, we propose a two-step generalized regression model with regularization for dealing with collinearity when developing linear prediction models. The model first reduces dimensionality of the predictors using pooling which is a commonly used method for down-sampling input

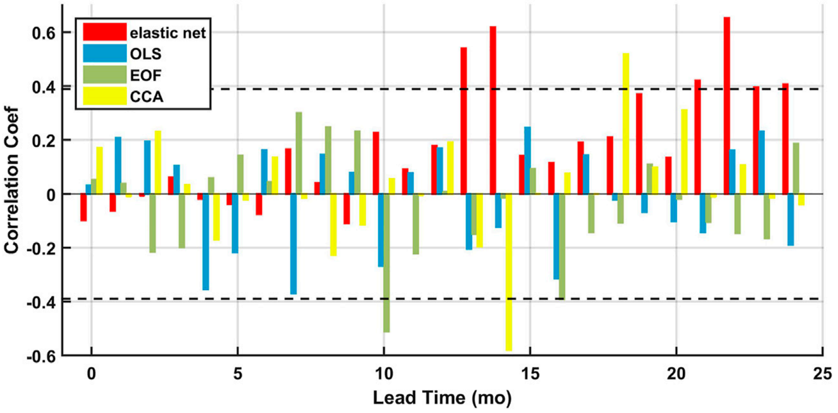


FIGURE 5 | Prediction skill for the testing set of 2001–2019 as measured by the correlation coefficients as function of lead times for elastic net (red), OLS (blue), EOF (green), and CCA (yellow). The p-value = 0.1 significance level is plotted in the black dashed lines.

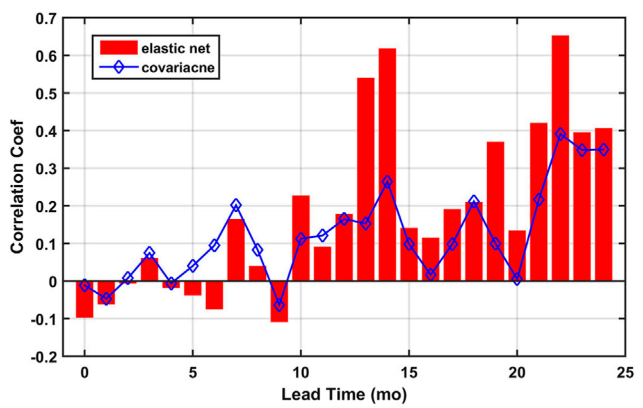


FIGURE 6 | Comparison between the predictive skill from the elastic net model (bar) and *L*s (blue diamond) at varying lead times. The statistical significance levels are not shown here as we are comparing correlation coefficients calculated using samples of different lengths (19 for the predictive skill and 39 for the *L*s).

TABLE 1 | Correlations between *L* and prediction skill at varying lead times for different regression models.

Model	*CC*
Elastic Net	0.82
OLS	0.01
EOF	–0.07
CCA	–0.11

representations. Then a regularized regression model is fitted using the pooled predictors to estimate real-valued or categorical predictand. A schematic is shown in **Figure 2**. In the pooling step, a new grid (in green) is defined by some characteristic values (e.g., maximum or median) of the four small grids. The extra step of standardization is to remove the systematic shift in precipitation around 2000 and to rescale precipitation and SSTs (as they have different amplitudes). In this study, we compared model performance using different pooling approaches (i.e., maximum/median/minimum pooling) with a squared window of four grids by four grids.

The regularized regression is given by **Eq. 1** with regression coefficient (β, β_0) (β_0 is the intercept), and it allows flexibility by using different deviance function (*Dev*) for predictands of different types. For example, the mean squared error (MSE) function is used for estimating the real-valued predictand and the log-likelihood function is used for categorical predictand (Hastie et al., 2009). The deviance function is rescaled by one over the total sample length *N*. The elastic net regression is adopted here and the regularization term uses a linear combination of the L-1 and L-2 norms of the regression coefficients as shown in **Eq. 2**.

$$(\beta, \beta_0) = \min_{\beta, \beta_0} \left(\frac{1}{N} Dev(\beta, \beta_0) + \lambda P_\alpha(\beta) \right) \tag{1}$$

$$P_\alpha(\beta) = \frac{1-\alpha}{2} \|\beta\|_2^2 + \alpha \|\beta\|_1 \tag{2}$$

There are two hyperparameters in the model: α and λ. α balances the regularization between the L-1 and L-2 norms of the regression coefficients β and is set to 0.01 for better visualization (Peng et al., 2020). λ is usually decided using a k-fold (e.g., 5-fold) cross validation (CV) (Tibshirani, 1996) and the λ value associated with minimum cross-validated mean squared errors is used (often referred to as the MinMSE λ). However, this procedure can be computationally burdensome since we have to repeat the CV for all lead times. Therefore, a constant λ is firstly determined using the training set data at 0-months lead and is used for all lead times. A preliminary study demonstrates that the significant predictive skill spikes in the testing period are not sensitive over a rather broad range of λs as shown in **Supplementary Figure S2**. For the true-amplitude predictand, the Pearson's correlation coefficient (CC) and the Nash–Sutcliffe efficiency (NSE) score are used for model performance evaluation. For the binary predictand, an accuracy score *S* is defined as given by **Eq. 3** where **1** is an indicator function and $\hat{y}$ is the predicted probability of y being 1 (i.e., wet).

$$S = \frac{1}{N} \sum_{i=1}^{N} \{ y_i \cdot \mathbf{1}(\hat{y}_i \geq 0.5) + (1 - y_i) \cdot \mathbf{1}(\hat{y}_i < 0.5) \} \tag{3}$$

3.2 Other Regression Models

Performance of the elastic net is compared against some commonly used regression methods in the two-step scheme including the OLS mutil-linear regression (see Hurrell (1996) for details), the EOF regression (see Wakabayashi and Kawamura (2004) for details), and the CCA regression (see Sun and Kim (2016) for details). All regression methods use the same pooled SSTs as the predictors. Though pooling can alleviate the issue of over-fitting, dimensionality of the pooled predictors is still highly non-square (39 years by 1,343 grids). The OLS and CCA regression seek for a linear combination of predictors that maximizes its correlation with the predictand and does not regularize the model complexity. The EOF regression first projects the original predictors onto some "dominant" basis vectors (often referred as EOFs) by decomposing the covariance matrix of the predictors, and then uses the EOFs as the new predictors. It can implicitly regularize the model complexity by using only a few EOFs explaining most variance of the original predictors. The EOF is implemented such that the original (pooled) SSTs are projected onto a set of orthonormal time series which constitute the predictors. It should be noted that it impossible to develop a prediction model this way since we are using data from testing set to construct the basis vectors. Here, the most dominant 50 EOFs accounting for over 88% variance of the original SST data are used in the EOF regression model.

3.3 The Correlation Analysis

A correlation analysis is designed to measure if any correlation patterns between the predictand precipitation and the predictor SSTs persist through time. We propose a new correlation metric *L* to quantitatively measure persistence of any correlation patterns: for a certain lead time, we first compute lagged correlations

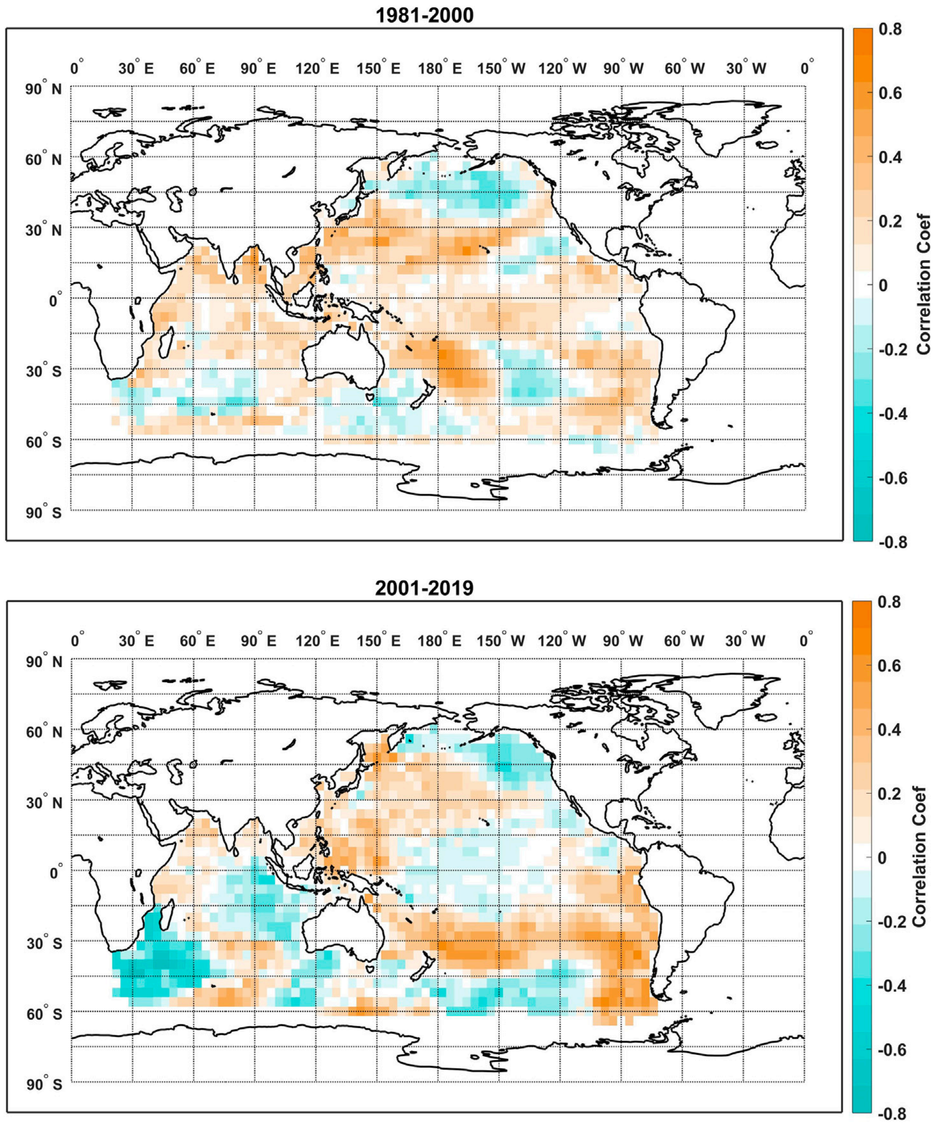


FIGURE 7 | Correlation maps between the TRHR precipitation and the SSTs over the training period of 1981–2000 **(top)** and the testing period of 2001–2019 **(bottom)** at the lead time of 14 months.

between the TRHR precipitation and SSTs at every grid for 1981–2000 and vectorize the correlation map into a column vector denoted by M_1; then this step is repeated for 2001–2019 to compute the column vector M_2; at last, L is defined by computing the correlation coefficients between the vectorized correlation maps M_1 and M_2. L is bounded by a upper limit of 1, which represents the extreme scenario where the correlations between the TRHR precipitation and the SSTs are perfectly consistent before and after 2001 and therefore, a good regression model trained on 1981–2000 should produce significantly high predictive skill on the testing period of 2001–2019. However, L being close to zero does not necessarily mean no predictive skill for regression models since L measures persistence of the global correlations between the TRHR precipitation and the SSTs while the regression model could pick some regional clusters of SSTs that have a persistent correlation with the predictand precipitation. The metric L is used here to estimate how much degradation of performance is resulted from over-fitting by comparing against the testing period predictive skill from the regression models.

4 RESULTS AND DISCUSSIONS

4.1 Comparison of Regression Models

Performance of the regularized regression models in predicting the TRHR precipitation in true amplitudes is examined in this section. The period of 1981–2000 is set as the training period while 2001–2019 is set as the testing period to be consistent with the standardization procedure. Prelim analysis with randomly

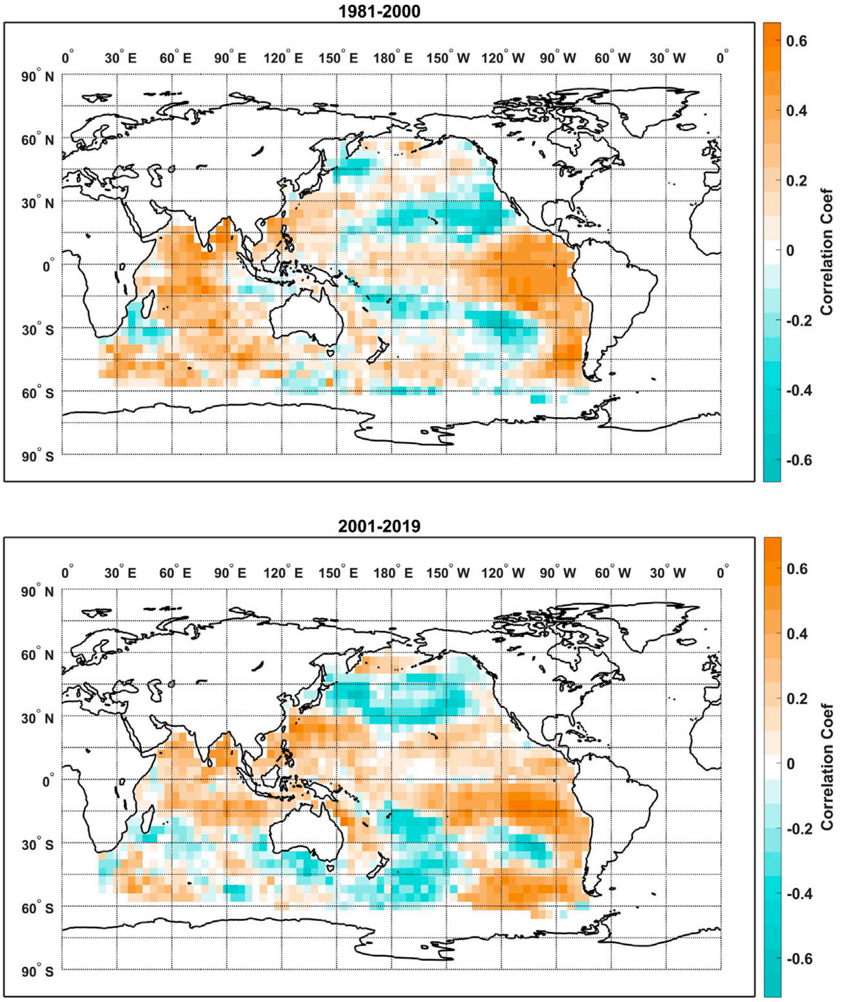


FIGURE 8 | Correlation maps between the TRHR precipitation and the SSTs over the training period of 1981–2000 **(top)** and the testing period of 2001–2019 **(bottom)** at the lead time of 22 months.

split data sets demonstrated consistent model skill patterns as shown in **Supplementary Figure S3**. Since both pooling and regularization are designed for effective dimensionality reduction and thus to avoid over-fitting, we first justify using the extra step of pooling by comparing predictive skill of the regularized regression models with and without pooling. The comparison of testing period correlation coefficients are shown for regression models without pooling and with maximum, median, and minimum pooling in **Figure 3**. Two spikes are observed at lead times of 13–14 months and 21–24 months. Significant improvement in model skill is shown for lead times of 13–14 months when pooling is used and at the lead time of 13 months, the predictive skill drops to below p-value = 0.1 significance level using non-pooled SSTs. For lead times of 21–24 months, consistent improvement, though less significant, is observed. The improvement could be due to the fact that while there exist some consistent large-scale circulation patterns, the signals may not be fixed to certain grids depending on the spatial resolution and projection coordinate system. Therefore, the model robustness can be improved by including signals of the neighboring grids with pooling. However, though not examined here, one must be careful with choosing the pooling window size since displacement of some circulation patterns can be important indicators of climate anomalies (McGregor et al., 2014; Manatsa et al., 2014) and this information may not be resolved when the pooling window is too large. It should be noted that the λ is re-calibrated for the regression models using non-pooled SSTs. And the statistical significance for the regularized regression is not straightforward to calculate and thus is not reported (Javanmard and Montanari, 2014). The maximum pooling is used in the following analyses.

Model skill for the testing period data as measured by the NSE scores is reported in **Figure 4**. Consistent patterns are observed as two spikes of NSE scores are found around lead times of 14 and 22 months. However, even at those two lead times, the predictive skill is barely satisfactory. By further looking at comparison between the observed and estimated precipitation, we figured that the elastic net model markedly underestimated the

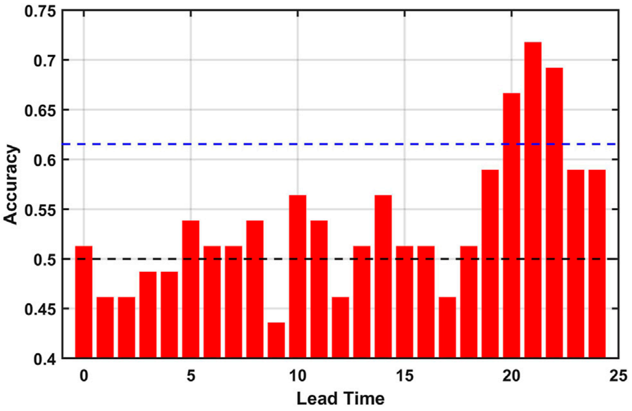


FIGURE 9 | Accuracy of predicting the wet-dry state of TRHR precipitation at varying lead times. The base skill (the null model of totally random guess) is plotted in the black dashed line. The one-tailed *p*-value = 0.1 significance level is estimated using bootstrapping and is plotted in the blue dashed line. The mean accuracy using a totally random guess strategy for 39 samples is collected from 10,000 repeated experiments and the 90th quantile value is used as estimation of the *p*-value = 0.1 significance level.

predictand amplitude. However, the extent of shrinking is consistent across the training and testing periods (as shown in **Supplementary Figure S4A**) and thus, in practice, one could 'learn' how much the amplitude is shrunk by looking at the training data and can then rescale testing period estimations. The rescaled estimations show significantly improved NSE scores (NSE = 0.36 for lead time of 14 months and 0.38 for lead time of 22 months (as shown in **Supplementary Figure S4B**)). A plausible explanation is that the elastic net regression sacrifices accuracy in amplitude estimation for model robustness by selecting only a few predictors and shrinking amplitudes of regression coefficients. This effect is more significant with highly non-square data as in our case since greater regularization must be applied. Therefore, amplitude-based measures such as NSE and the root mean square error (RMSE) may not be applicable for model evaluation. The rescaling method discussed above is not recommended since the model is designed to be biased for more robustness, and in following analyses, the Pearson's correlation coefficient is used as the primary measure of model skill.

We then justify using the regularization by comparing predictive skill of the regression with and without regularization. Comparison of the model skill from the elastic net, OLS, EOF, and CCA regression models are shown in **Figure 5**. Statistically significant positive predictive skill is only observed for the elastic and CCA regressions models. The elastic net models show two spikes of good predictive skill at lead times of 13–14 months and 21–24 months. Statistically significant positive skill is observed for the CCA regression models only at the lead time of 18 months while that of the elastic net regression almost hit the *p*-value = 0.1 significance level at the lead time of 19 months. Overall, the elastic net regression shows more potential in finding the linear associations between the TRHR precipitation and the SSTs. While statistically significant positive skill are only found at rather long lead times, this does not necessarily mean that there is no connection between the TRHR precipitation and large-scale climate fields at shorter lead times. We are limiting our analysis to only using SSTs from the Pacific and Indian Oceans which is only one sector of the complex large-scale circulations including a wider range of variables like geopotential heights, humidity, vertical velocity of atmosphere (OMEGA) and horizontal winds etc. Furthermore, we are limiting our analysis in the frame of linear models as we compare different types of regression models. Instead of developing accurate forecast models, our intent is to examine how pooling and regularization would improve performance of the linear models at rather low costs. The better performance of elastic net is understandable here since it explicitly regularizes model complexity and provokes sparsity in regression coefficients by using the L-1 norm regularization.

4.2 Source of High Model Skill

A correlation analysis is conducted to measure at a certain lead time, how well a linear model based on the global correlation between the TRHR precipitation and the SSTs can perform. The potential predictive skill is estimated by the new correlation metric L defined earlier as L measures how the time-shifted global correlation patterns persist from 1981 to 2000 (the training period) to 2001–2019 (the testing period). A comparison between L and the predictive skill of the elastic net model at varying lead times is shown in **Figure 6**. Spikes in L are observed at lead times of 7, 14, 18, and 22 months. Three of the spikes coincide with good predictive skill from the elastic net model (i.e., lead times of 14, 19, and 22 months) while only one of the spike coincide with good skill from other regression models (i.e., lead time of 18 months for the CCA regression model). To estimate how much potential are realized for each model, correlation coefficients between the series of L and model skill from regression models over the lead times of 0–24 months are computed and reported in **Table 1**. The only statistically significant correlation is found for the elastic net model (0.82 for 25 samples, p-value < 0.01) while rather low correlations are found for other regression models. The results suggest that the OLS, EOF, and CCA regression models do not perform well even when there exist persistent correlations between the predictand precipitation and the SSTs.

Possible explanations are proposed here based on algorithms of the regression methods. For the OLS and CCA regression, the models could be over-fitted to the noisy SST signals for high training skill as both methods decompose the covariance between the predictand precipitation and SSTs to seek a linear combination that either minimizes the MSE (for OLS) or maximizes the correlation (for CCA). Thus, the models are less robust and can perform poorly when evaluated using the testing period samples. As for the EOF regression, the EOF re-constructs the predictors by projecting the global covariance of the SSTs onto some dominant orthonormal basis vectors (EOFs). There are two limiting factors: 1) the assumption of orthogonality may not be appropriate as the new predictors are constructed from the physical variable of SST; 2) any regional persistent correlation patterns between the TRHR precipitation and the

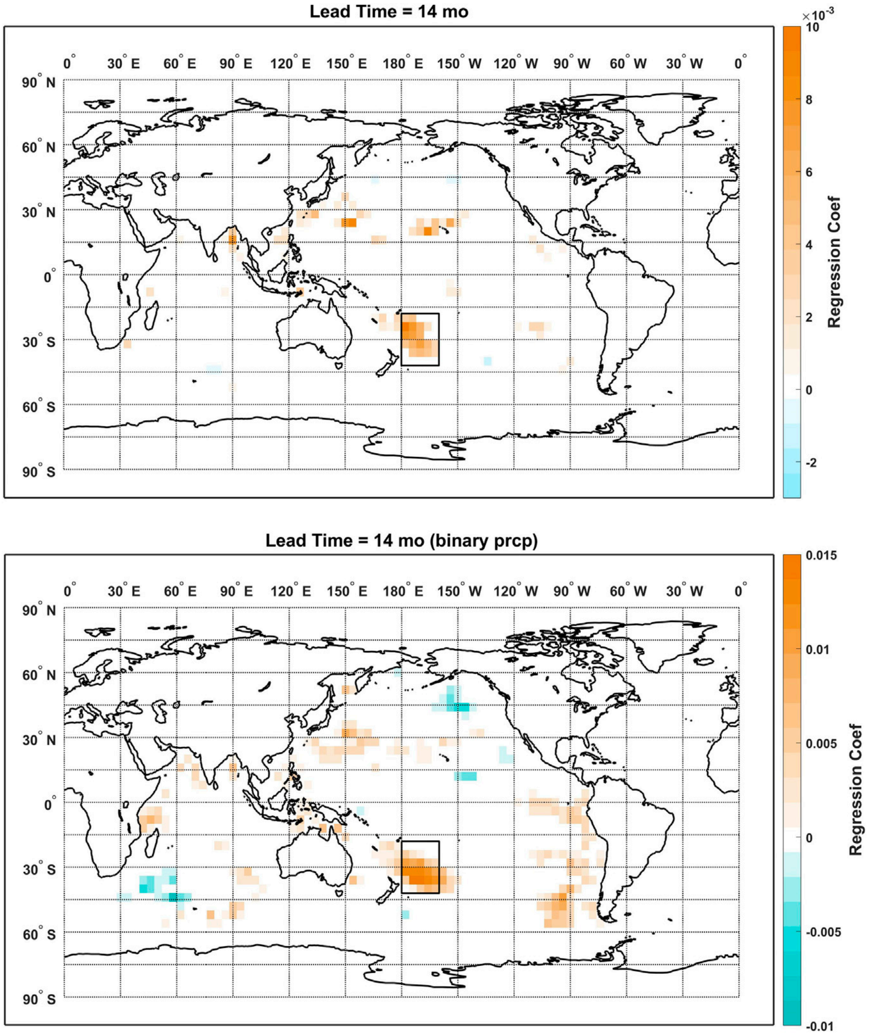

Lead Time = 14 mo
×10^-3
10
8
6
4
2
0
-2
Regression Coef
0° 30° E 60° E 90° E 120° E 150° E 180° E 150° W 120° W 90° W 60° W 30° W 0°
90° N 60° N 30° N 0° 30° S 60° S 90° S
Lead Time = 14 mo (binary prcp)
0.015
0.01
0.005
0
-0.005
-0.01
Regression Coef

FIGURE 10 | Maps of regression coefficients from the elastic net models using true-amplitude **(top)** and binary **(bottom)** predictand precipitation at the lead time of 14 months.

SSTs could be lost if they do not make significantly large contribution to the global covariance. Technically, we are not predicting the TRHR precipitation with the EOF regression models since data of the full study period (1981–2019) is used for constructing the new set of predictors (i.e., EOFs).

Interestingly, comparably high predictive skill are observed for the elastic net models at lead times of 14 and 22 months while persistence of the global correlation is much lower at the lead time of 14 months as indicated by *L*. We specifically looked at the correlation maps between the TRHR precipitation and the SSTs before and after 2000 for the lead times of 14 and 22 months as shown in **Figures 7**, **8**, respectively. For the lead time of 14 months: before 2000, the correlation map features a cluster of positive correlations [180E-210E, 45S-15S] to the east of Australia and an extended band of positive correlations over the mid-north Pacific Ocean [120E-210E, 15N-30N]. Scattered and less significant positive correlations are observed over the northern Indian Ocean and to the west of South America; After 2000, the correlation map is dominated by two major clusters of positive correlations to the east of Australia [180E-210E, 45S-15S] and to the west of South America [260E-280E, 60S-15S] and one major cluster of negative correlations over the southwestern Indian Ocean [30E-60E, 60S-30S]. Less significantly positive correlations are observed over the north-western Pacific which overlaps with the extended band before 2000. For the lead time of 22 months, both correlation maps before and after 2000 are dominated by large clusters of positive correlations over the northern Indian Ocean [60E-90E, 15S-15N] and eastern tropical Pacific Ocean [210E-270E, 15S-0]. The major difference is that clusters of positive correlations over the southern-eastern Pacific and the mid-western Pacific [120E-150E, 15N-30N] get enhanced in correlation amplitude and extended in spatial coverage. A comparison between **Figures 7**, **8** suggests a higher level of persistence in the global correlation between the predictand precipitation and the SSTs at the lead time of 22 months, which is consistent with the higher value of *L*

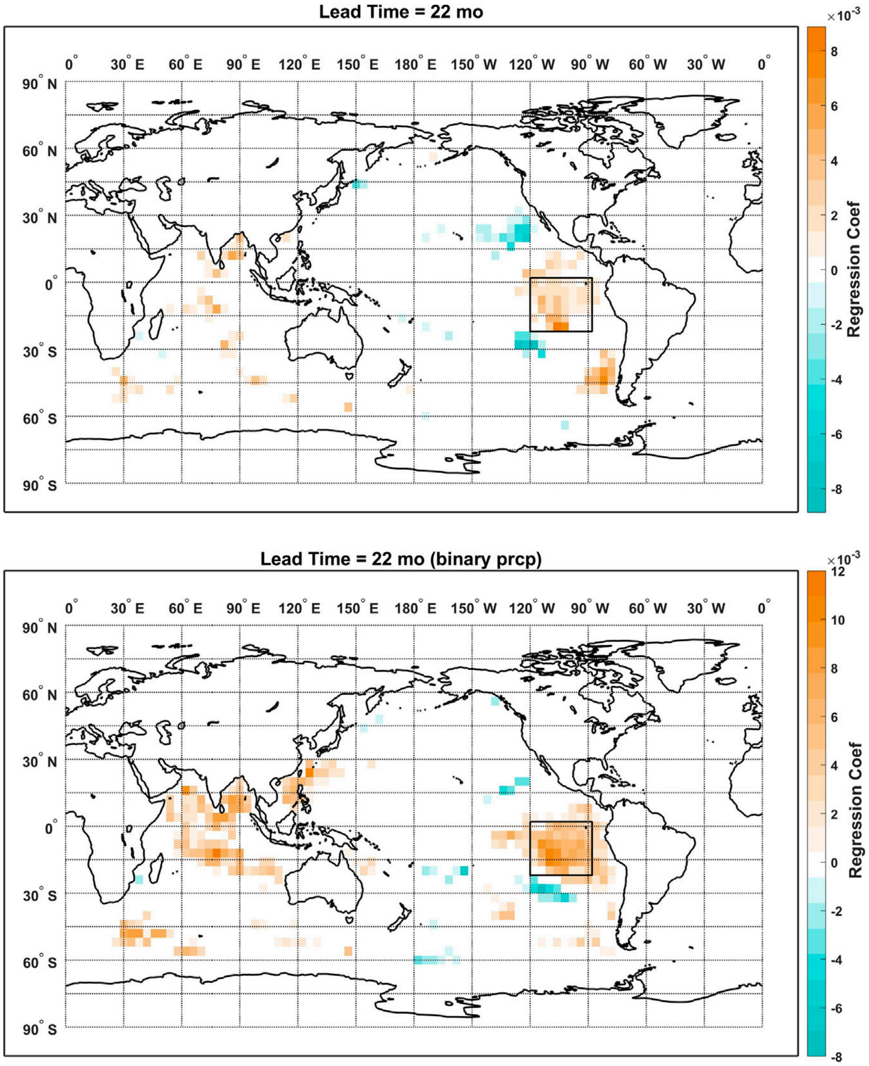


FIGURE 11 | Maps of regression coefficients from the elastic net models using true-amplitude **(top)** and binary **(bottom)** predictand precipitation at the lead time of 22 months.

in **Figure 6**. However, a lower value of *L* does not necessarily mean low predictive skill potential for a linear model as some regional persistent correlation patterns are observed (i.e., the positive correlation cluster to the east of Australia) for the lead time of 14 months. The correlation maps are further compared with the regression coefficient maps in **Section 4.3** as we attempt to interpret the high model skill of the elastic net regression.

4.3 An Alternative Multinomial Regression Model

In this section, the binary precipitation is predicted using the multinomial regression version (i.e., the elastic net logistic regression) of our model. While new machine learning techniques like the classification and regression tree (CART) can have better model skill in multi-class prediction of climate variables (Choubin et al., 2018; Huang et al., 2021), the elastic net logistic regression is tested to demonstrate flexibility and consistency with altered deviance functions. The multinomial regression may have more use in practical application since amplitudes of the predictand tend to be underestimated because of the regularization (Peng et al., 2020). The logistic regression is implemented by simply replacing the MSE function with the log-likelihood function for *Dev* (Hastie et al., 2009). Though the logistic regression is a special case of the multinomial regression, the model could be easily generalized for predictand of more than two categories by separately fitting a regularized Poisson regression for each category of which the coefficients are used to estimate the coefficients of the multinomial regression model (Baker, 1994).

To extend the sample size, the leave-one-out cross validation is used (the testing sample size is thus increased from 19 to 39 here).

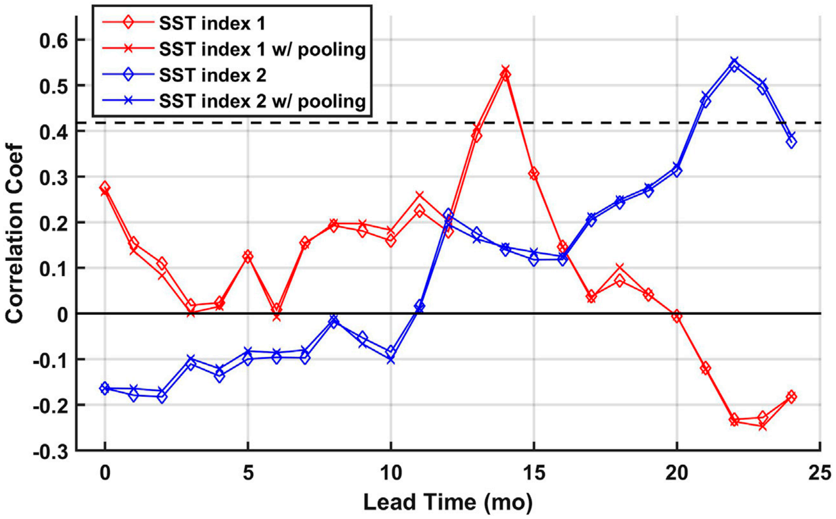


FIGURE 12 | Lagged correlations between the TRHR precipitation and the SST indices. The *p*-value = 0.01 significance level is plotted in the black dashed line.

The leave-one-out CV is not used in the previous analysis for predicting real-valued predictand since the Pearson's correlation coefficient is used for performance evaluation and the leave-one-out CV could result in bias for violating continuity in data. The accuracy as measured by the *S* score as a function of lead time is shown in **Figure 9**. A consistent spike of high accuracy (over 70% accuracy) is observed at around lead times of 22 months. While a local maximum of accuracy is observed at the lead time of 14 months, it is not statistically significant. A major reason could be that the *S* score is not the equivalent measure of the correlation coefficient as we used in **Figures 3**, **5**. A example of high correlation but low *S* is when the model could well predict the extreme events but performs poorly for the less extreme events.

Consistency between the regression models using predictand and deviance function of different types is further examined by comparing the maps of the regressions coefficients for lead times of 14 and 22 months as shown in **Figures 10**, **11**, respectively. For the lead time of 14 months, both models feature a major cluster of positive coefficients to the east of Australia while more non-zero coefficients are observed for the logistic regression model over the southeastern Pacific Ocean and the Indian Ocean. As for the lead time of 22 months, both maps are dominated by the large cluster of positive coefficients over the eastern tropical Pacific Ocean. While positive coefficients are observed over the northern Indian Ocean for both models, the coefficients are more sparsely distributed for the model using the true-amplitude predictand. More non-zero coefficients are found for the logistic regression model at both lead times, which could be due to a less optimal regularization as we only did the cross validation on a relatively sparse sequence of λs with a lead time of 0 months. But overall, the major clusters of non-zero regression coefficients are consistent across the models. And this is also confirmed by the results that statistically significant correlations are found between the vectorized coefficient maps of the two models. The correlation coefficients are 0.45 for the lead time of 14 months and 0.44 for the lead time of 22 months (1,343 samples) and are 0.35 for the lead time of 14 months (327 samples) and 0.40 for the lead time of 22 months (359 samples) when only the non-zero coefficients are considered.

The comparably high predictive skill at the lead time of 14 months is easy to interpret if we compare the coefficient maps from **Figures 10**, **11** to the correlation maps from **Figures 7**, **8**. While less persistence is observed for the global correlation at the lead time of 14 months, the elastic net model managed to select the regional persistent cluster of positive correlations to the east of Australia. At last, two SST indices are defined based on the consistent patterns from the two regression models: the SST index 1 is defined by the mean SST over the domain of [180°W-160°W, 42°S-18°S]; the SST index 2 is defined by the mean SST over the domain of [120°W-88°W, 22°S-2°N]. Lagged correlations between the TRHR precipitation and the SST indices calculated using the non-pooled and pooled SSTs are plotted in **Figure 12**. Statistically significant positive correlations are observed at the corresponding lead times and consistent results are shown for non-pooled and pooled SSTs. The results suggest that the proposed framework managed to select certain regional SSTs that are consistently correlated with the predictand precipitation and demonstrated good interpretability. While correlation does not necessarily imply causation, the elastic net regression models show good potential here in guiding further research with its highly interpretable and flexible models.

5 CONCLUSION

In this paper, we tested a generalized regression model with regularization coupled with pooling in predicting the TRHR wet-season precipitation at lead times of 0–24 months using the

Pacific and Indian Ocean SSTs. The regression is first tested using the true-amplitude predictand and is compared against some widely-used regression models including the OLS multi-linear regression, the EOF regression and the CCA regression. Significantly good predictive skill are observed using the elastic net regression models for certain long lead times which are further examined using a correlation analysis. The results suggest that the elastic net regression achieves good performance in identifying and using the persistent correlation patterns while the other three regression models show relatively poor performance. Low model skill at shorter lead times can be due to that only SST is used as the predictor while teleconnection signals can propagate through other climate fields chronologically. A multinomial elastic net regression model is then used to demonstrate flexibility and consistency of the proposed framework. Consistent model skill and regression coefficient maps are observed even when predictand and deviance functions of different types are used. By comparing the correlation analysis and the regression coefficient maps, we found that the elastic net model managed to select regional persistent correlation patterns as the contributing predictors while the other widely-used regression models are based on the global covariance either between the predictand and the predictors or within the predictors (and thus are vulnerable to over-fitting). At last, two SST indices are defined based on the major clusters of non-zeros coefficients from the elastic net models and are found to be significantly correlated to the TRHR precipitation at the corresponding lead times. Overall, the proposed framework demonstrates good interpretability in identifying covariates with high predictive skill and the potential in guiding further investigation using more complex, nonlinear statistical models or physically based modeling experiments.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/**Supplementary Material**. The codes are processed data can be found via https://github.com/cruiseryy/TRHR_elasticnet_pooling and further inquiries can be directed to the corresponding author.

AUTHOR CONTRIBUTIONS

The idea originated from a discussion among the three authors. XP and JA designed the study and XP carried out the experiment, and wrote the article. JA oversaw the entire project. JA and TL offered technical advice during the experiment and provided inputs to the article writing. All authors contributed to the article and approved the submitted version.

FUNDING

The Cornell authors were supported by the Cornell China Center under the project of “Food-Energy-Water Security in the Three-Rivers Headwater Region: Integrating Science, Data and Decision Tools to Manage Under a Changing Climate and Development Pressure”. The Tsinghua author was supported by the National Key Research and Development Program of China (No. 2016YFE0201900), and the Research Fund of the State Key Laboratory of Hydroscience and Engineering, Tsinghua University (No. 2017-KY-04).

ACKNOWLEDGMENTS

The rainfall data were collected from the CHIRPS (http://chg.geog.ucsb.edu/data/chirps/) and the China Meteorological Administration (CMA). The SST data were collected from the HadISST dataset (https://www.metoffice.gov.uk/hadobs/hadisst/). We thank the two reviewers for their insightful comments.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/feart.2021.724599/full#supplementary-material

REFERENCES

Akaike, H. (1998). “Information Theory and an Extension of the Maximum Likelihood Principle,” in *Selected Papers of Hirotugu Akaike* (New York: Springer), 199–213. doi:10.1007/978-1-4612-1694-0_15

Ali, M., Deo, R. C., Maraseni, T., and Downs, N. J. (2019). Improving Spi-Derived Drought Forecasts Incorporating Synoptic-Scale Climate Indices in Multi-phase Multivariate Empirical Mode Decomposition Model Hybridized with Simulated Annealing and Kernel ridge Regression Algorithms. *J. Hydrol.* 576, 164–184. doi:10.1016/j.jhydrol.2019.06.032

Baker, S. G. (1994). The Multinomial-Poisson Transformation. *The Statistician.* 43, 495–504. doi:10.2307/2348134

Benn, D. I., and Owen, L. A. (1998). The Role of the Indian Summer Monsoon and the Mid-latitude Westerlies in Himalayan Glaciation: Review and Speculative Discussion. *J. Geol. Soc.* 155, 353–363. doi:10.1144/gsjgs.155.2.0353

Block, P. (2011). Tailoring Seasonal Climate Forecasts for Hydropower Operations. *Hydrol. Earth Syst. Sci.* 15, 1355–1368. doi:10.5194/hess-15-1355-2011

Bueso, D., Piles, M., and Camps-Valls, G. (2020). Nonlinear Pca for Spatio-Temporal Analysis of Earth Observation Data. *IEEE Trans. Geosci. Remote. Sens.* 58, 5752–5763. doi:10.1109/tgrs.2020.2969813

Carpenter, T. M., and Georgakakos, K. P. (2001). Assessment of folsom lake Response to Historical and Potential Future Climate Scenarios: 1. Forecasting. *J. Hydrol.* 249, 148–175. doi:10.1016/s0022-1694(01)00417-6

Carvalho, D. V., Pereira, E. M., and Cardoso, J. S. (2019). Machine Learning Interpretability: A Survey on Methods and Metrics. *Electronics* 8, 832. doi:10.3390/electronics8080832

Center, N. G. D. (1988). Data From: Etopo-5 Bathymetry/topography Data

Chen, C., Li, T., Sivakumar, B., Li, J., and Wang, G. (2020). Attribution of Growing Season Vegetation Activity to Climate Change and Human Activities in the Three-River Headwaters Region, china. *J. Hydroinformatics* 22, 186–204. doi:10.2166/hydro.2019.003

Choubin, B., Zehtabian, G., Azareh, A., Rafiei-Sardooi, E., Sajedi-Hosseini, F., and Kişi, Ö. (2018). Precipitation Forecasting Using Classification and Regression Trees (Cart) Model: a Comparative Study of Different Approaches. *Environ. earth Sci.* 77, 1–13. doi:10.1007/s12665-018-7498-z

DelSole, T., and Banerjee, A. (2017). Statistical Seasonal Prediction Based on Regularized Regression. *J. Clim.* 30, 1345–1361. doi:10.1175/jcli-d-16-0249.1

Devineni, N., and Sankarasubramanian, A. (2010). Improving the Prediction of winter Precipitation and Temperature over the continental united states: Role of the Enso State in Developing Multimodel Combinations. *Monthly Weather Rev.* 138, 2447–2468. doi:10.1175/2009mwr3112.1

Dong, Y., Zhai, J., Zhao, Y., Li, H., Wang, Q., Jiang, S., et al. (2020). Teleconnection Patterns of Precipitation in the Three-River Headwaters Region, china. *Environ. Res. Lett.* 15, 104050. doi:10.1088/1748-9326/aba8c0

Enfield, D. B., Mestas-Nuñez, A. M., Mayer, D. A., and Cid-Serrano, L. (1999). How Ubiquitous Is the Dipole Relationship in Tropical atlantic Sea Surface Temperatures? *J. Geophys. Res.* 104, 7841–7848. doi:10.1029/1998jc900109

Fan, Y. R., Huang, W., Huang, G. H., Li, Z., Li, Y. P., Wang, X. Q., et al. (2015). A Stepwise-Cluster Forecasting Approach for Monthly Streamflows Based on Climate Teleconnections. *Stoch Environ. Res. Risk Assess.* 29, 1557–1569. doi:10.1007/s00477-015-1048-y

Feng, L., and Zhou, T. (2012). Water Vapor Transport for Summer Precipitation over the Tibetan Plateau: Multidata Set Analysis. *J. Geophys. Res. Atmospheres.* 117, 1–21. doi:10.1029/2011jd017012

Funk, C., Peterson, P., Landsfeld, M., Pedreros, D., Verdin, J., Shukla, S., et al. (2015). The Climate Hazards Infrared Precipitation with Stations-a New Environmental Record for Monitoring Extremes. *Sci. Data.* 2. doi:10.1038/sdata.2015.66

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., and Kagal, L. (2018). "Explaining Explanations: An Overview of Interpretability of Machine Learning," in 2018 IEEE 5th International Conference on data science and advanced analytics (DSAA), Turin, Italy, October 1–4, 2018 ((IEEE)), 80–89. doi:10.1109/dsaa.2018.00018

Goddard, L., Mason, S. J., Zebiak, S. E., Ropelewski, C. F., Basher, R., and Cane, M. A. (2001). Current Approaches to Seasonal to Interannual Climate Predictions. *Int. J. Climatol.* 21, 1111–1152. doi:10.1002/joc.636

Ham, Y.-G., Kim, J.-H., and Luo, J.-J. (2019). Deep Learning for Multi-Year Enso Forecasts. *Nature* 573, 568–572. doi:10.1038/s41586-019-1559-7

Hansen, J. W., Mason, S. J., Sun, L., and Tall, A. (2011). Review of Seasonal Climate Forecasting for Agriculture in Sub-saharan Africa. *Ex. Agric.* 47, 205–240. doi:10.1017/s0014479710000876

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer Science & Business Media.

Hoerl, A. E., and Kennard, R. W. (1970). Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics* 12, 55–67. doi:10.1080/00401706.1970.10488634

Huang, L., Kang, J., Wan, M., Fang, L., Zhang, C., and Zeng, Z. (2021). Solar Radiation Prediction Using Different Machine Learning Algorithms and Implications for Extreme Climate Events. *Front. Earth Sci.* 9, 202. doi:10.3389/feart.2021.596860

Hurrell, J. W. (1996). Influence of Variations in Extratropical Wintertime Teleconnections on Northern Hemisphere Temperature. *Geophys. Res. Lett.* 23, 665–668. doi:10.1029/96gl00459

Javanmard, A., and Montanari, A. (2014). Confidence Intervals and Hypothesis Testing for High-Dimensional Regression. *J. Machine Learn. Res.* 15, 2869–2909. doi:10.5555/2627435.2697057

Kalchbrenner, N., Grefenstette, E., and Blunsom, P. (2014). *A Convolutional Neural Network for Modelling Sentences*. arXiv. preprint arXiv:1404.2188.

Kharin, V. V., and Zwiers, F. W. (2002). Climate Predictions with Multimodel Ensembles. *J. Clim.* 15, 793–799. doi:10.1175/1520-0442(2002)015<0793:cpwme>2.0.co;2

Kim, J., Oh, H.-S., Lim, Y., and Kang, H.-S. (2017). Seasonal Precipitation Prediction via Data-Adaptive Principal Component Regression. *Int. J. Climatol.* 37, 75–86. doi:10.1002/joc.4979

Krishnamurti, T. N., Kishtawal, C. M., LaRow, T. E., Bachiochi, D. R., Zhang, Z., Williford, C. E., et al. (1999). Improved Weather and Seasonal Climate Forecasts from Multimodel Superensemble. *Science* 285, 1548–1550. doi:10.1126/science.285.5433.1548

Lemos, M. C., Finan, T. J., Fox, R. W., Nelson, D. R., and Tucker, J. (2002). The Use of Seasonal Climate Forecasting in Policymaking: Lessons from Northeast brazil. *Clim. Change.* 55, 479–507. doi:10.1023/a:1020785826029

Li, J., Pollinger, F., and Paeth, H. (2020). Comparing the Lasso Predictor-Selection and Regression Method with Classical Approaches of Precipitation Bias Adjustment in Decadal Climate Predictions. *Monthly Weather Rev.* 148, 4339–4351. doi:10.1175/mwr-d-19-0302.1

Liu, P. (2015). A Survey of Remote-Sensing Big Data. *Front. Environ. Sci.* 3, 45. doi:10.3389/fenvs.2015.00045

Manatsa, D., Morioka, Y., Behera, S. K., Matarira, C. H., and Yamagata, T. (2014). Impact of Mascarene High Variability on the East African 'short rains'. *Clim. Dyn.* 42, 1259–1274. doi:10.1007/s00382-013-1848-z

Mantua, N. J., Hare, S. R., Zhang, Y., Wallace, J. M., and Francis, R. C. (1997). A pacific Interdecadal Climate Oscillation with Impacts on salmon Production. *Bull. Amer. Meteorol. Soc.* 78, 1069–1079. doi:10.1175/1520-0477(1997)078<1069:apicow>2.0.co;2

Matsui, H., and Konishi, S. (2011). Variable Selection for Functional Regression Models via the Regularization. *Comput. Stat. Data Anal.* 55, 3304–3310. doi:10.1016/j.csda.2011.06.016

McGregor, S., Timmermann, A., Stuecker, M. F., England, M. H., Merrifield, M., Jin, F.-F., et al. (2014). Recent walker Circulation Strengthening and pacific Cooling Amplified by atlantic Warming. *Nat. Clim Change.* 4, 888–892. doi:10.1038/nclimate2330

Mekanik, F., Imteaz, M. A., Gato-Trinidad, S., and Elmahdi, A. (2013). Multiple Regression and Artificial Neural Network for Long-Term Rainfall Forecasting Using Large Scale Climate Modes. *J. Hydrol.* 503, 11–21. doi:10.1016/j.jhydrol.2013.08.035

Menemenlis, D., Fukumori, I., and Lee, T. (2005). Using Green's Functions to Calibrate an Ocean General Circulation Model. *Monthly weather Rev.* 133, 1224–1240. doi:10.1175/mwr2912.1

Mo, K. C. (2003). Ensemble Canonical Correlation Prediction of Surface Temperature over the united states. *J. Clim.* 16, 1665–1683. doi:10.1175/1520-0442(2003)016<1665:eccpos>2.0.co;2

Mudelsee, M. (2000). Ramp Function Regression: a Tool for Quantifying Climate Transitions. *Comput. Geosciences.* 26, 293–307. doi:10.1016/s0098-3004(99)00141-7

Peng, X., Steinschneider, S., and Albertson, J. (2020). Investigating Long-Range Seasonal Predictability of East African Short rains: Influence of the Mascarene High on the Indian Ocean walker Cell. *J. Appl. Meteorology Climatology.* 59, 1077–1090. doi:10.1175/jamc-d-19-0109.1

Rasmusson, E. M., and Carpenter, T. H. (1982). Variations in Tropical Sea Surface Temperature and Surface Wind Fields Associated with the Southern Oscillation/El Niño. *Mon. Wea. Rev.* 110, 354–384. doi:10.1175/1520-0493(1982)110<0354:vitsst>2.0.co;2

Rayner, N., Parker, D. E., Horton, E., Folland, C. K., Alexander, L. V., Rowell, D., et al. (2003). Global Analyses of Sea Surface Temperature, Sea Ice, and Night marine Air Temperature since the Late Nineteenth century. *J. Geophys. Res. Atmospheres.* 108. doi:10.1029/2002jd002670

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., et al. (2019). Deep Learning and Process Understanding for Data-Driven Earth System Science. *Nature* 566, 195–204. doi:10.1038/s41586-019-0912-1

Rust, H. W., Richling, A., Bissolli, P., and Ulbrich, U. (2015). Linking Teleconnection Patterns to European Temperature: a Multiple Linear Regression Model. *Meteorol. Z.* 24, 411–423. doi:10.1127/metz/2015/0642

Sahastrabuddhe, R., and Ghosh, S. (2021). Does Statistical Model Perform at Par with Computationally Expensive General Circulation Model for Decadal Prediction? *Environ. Res. Lett.* 16, 064028. doi:10.1088/1748-9326/abfeed

Schepen, A., Zhao, T., Wang, Q. J., and Robertson, D. E. (2018). A Bayesian Modelling Method for post-processing Daily Sub-seasonal to Seasonal Rainfall Forecasts from Global Climate Models and Evaluation for 12 Australian Catchments. *Hydrol. Earth Syst. Sci.* 22, 1615–1628. doi:10.5194/hess-22-1615-2018

Schoof, J. T., and Pryor, S. C. (2001). Downscaling Temperature and Precipitation: A Comparison of Regression-Based Methods and Artificial Neural Networks. *Int. J. Climatol.* 21, 773–790. doi:10.1002/joc.655

Schwarz, G. (1978). Index of Authors. *J. Hyg.* 80, 461–464. doi:10.1017/s0022172400024931

Shaman, J., and Tziperman, E. (2005). The Effect of Enso on Tibetan Plateau Snow Depth: A Stationary Wave Teleconnection Mechanism and Implications for the South Asian Monsoons. *J. Clim.* 18, 2067–2079. doi:10.1175/jcli3391.1

Soleh, A. M., Wigena, A. H., Djuraidah, A., and Saefuddin, A. (2015). Statistical Downscaling to Predict Monthly Rainfall Using Linear Regression with L_1 Regularization (LASSO). *ams* 9, 5361–5369. doi:10.12988/ams.2015.56434

Solow, A. R. (1987). Testing for Climate Change: An Application of the Two-phase Regression Model. *J. Clim. Appl. Meteorol.* 26, 1401–1405. doi:10.1175/1520-0450(1987)026<1401:tfccaa>2.0.co;2

Stevens, A., Willett, R., Mamalakis, A., Foufoula-Georgiou, E., Tejedor, A., Randerson, J. T., et al. (2021). Graph-Guided Regularized Regression of Pacific Ocean Climate Variables to Increase Predictive Skill of Southwestern U.S. Winter Precipitation. *J. Clim.* 34, 737–754. doi:10.1175/jcli-d-20-0079.1

Sun, M., and Kim, G. (2016). Quantitative Monthly Precipitation Forecasting Using Cyclostationary Empirical Orthogonal Function and Canonical Correlation Analysis. *J. Hydrol. Eng.* 21, 04015045. doi:10.1061/(asce)he.1943-5584.0001244

Tan, X., and Shao, D. (2017). Precipitation Trends and Teleconnections Identified Using Quantile Regressions over Xinjiang, china. *Int. J. Climatol.* 37, 1510–1525. doi:10.1002/joc.4794

Thornthwaite, C. W. (1948). An Approach toward a Rational Classification of Climate. *Geographical Rev.* 38, 55–94. doi:10.2307/210739

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *J. R. Stat. Soc. Ser. B (Methodological)* 58, 267–288. doi:10.1111/j.2517-6161.1996.tb02080.x

Trenberth, K. E., and Stepaniak, D. P. (2001). Indices of El Niño Evolution. *J. Clim.* 14, 1697–1701. doi:10.1175/1520-0442(2001)014<1697:lioeno>2.0.co;2

Trenberth, K. E. (1997). The Definition of El Niño. *Bull. Amer. Meteorol. Soc.* 78, 2771–2777. doi:10.1175/1520-0477(1997)078<2771:tdoeno>2.0.co;2

Van Oldenborgh, G. J., and Burgers, G. (2005). Searching for Decadal Variations in Enso Precipitation Teleconnections. *Geophys. Res. Lett.* 32. doi:10.1029/2005gl023110

Verdin, J., Funk, C., Senay, G., and Choularton, R. (2005). Climate Science and Famine Early Warning. *Phil. Trans. R. Soc. B.* 360, 2155–2168. doi:10.1098/rstb.2005.1754

Von Storch, H., and Zwiers, F. W. (2001). *Statistical Analysis in Climate Research*. Cambridge University Press.

Wakabayashi, S., and Kawamura, R. (2004). NOTES and CORRESPONDENCE; Extraction of Major Teleconnection Patterns Possibly Associated with the Anomalous Summer Climate in Japan. *J. Meteorol. Soc. Jpn.* 82, 1577–1588. doi:10.2151/jmsj.82.1577

Wilhite, D. A., and Svoboda, M. D. (2000). Drought Early Warning Systems in the Context of Drought Preparedness and Mitigation, in *Early Warning Systems for Drought Preparedness and Drought Management*. (Geneva: World Meteorological Organization), 1–21.

Worland, S. C., Steinschneider, S., Asquith, W., Knight, R., and Wieczorek, M. (2019). Prediction and Inference of Flow Duration Curves Using Multioutput Neural Networks. *Water Resour. Res.* 55, 6850–6868. doi:10.1029/2018wr024463

Xie, S.-P., Hu, K., Hafner, J., Tokinaga, H., Du, Y., Huang, G., et al. (2009). Indian Ocean Capacitor Effect on Indo-Western Pacific Climate during the Summer Following El Niño. *J. Clim.* 22, 730–747. doi:10.1175/2008jcli2544.1

Xie, S.-P., Kosaka, Y., Du, Y., Hu, K., Chowdary, J. S., and Huang, G. (2016). Indo-western pacific Ocean Capacitor and Coherent Climate Anomalies in post-enso Summer: A Review. *Adv. Atmos. Sci.* 33, 411–432. doi:10.1007/s00376-015-5192-6

Yang, X., and DelSole, T. (2012). Systematic Comparison of Enso Teleconnection Patterns between Models and Observations. *J. Clim.* 25, 425–446. doi:10.1175/jcli-d-11-00175.1

Yu, D., Wang, H., Chen, P., and Wei, Z. (2014). "Mixed Pooling for Convolutional Neural Networks," in International conference on rough sets and knowledge technology. Springer, 364–375. doi:10.1007/978-3-319-11740-9_34

Zeiler, M. D., and Fergus, R. (2013). *Stochastic Pooling for Regularization of Deep Convolutional Neural Networks*. arXi. preprint arXiv:1301.3557.

Zhang, S., Gan, T. Y., and Bush, A. B. G. (2020). Variability of Arctic Sea Ice Based on Quantile Regression and the Teleconnection with Large-Scale Climate Patterns. *J. Clim.* 33, 4009–4025. doi:10.1175/jcli-d-19-0375.1

Zhang, Y., Huang, W., and Zhong, D. (2019). Major Moisture Pathways and Their Importance to Rainy Season Precipitation over the Sanjiangyuan Region of the Tibetan Plateau. *J. Clim.* 32, 6837–6857. doi:10.1175/jcli-d-19-0196.1

Zhang, Y., Wallace, J. M., and Battisti, D. S. (1997). ENSO-like Interdecadal Variability: 1900-93. *J. Clim.* 10, 1004–1020. doi:10.1175/1520-0442(1997)010<1004:eliv>2.0.co;2

Zhao, Y., Xu, X., Liao, L., Wang, Y., Gu, X., Qin, R., et al. (2019). The Severity of Drought and Precipitation Prediction in the Eastern Fringe of the Tibetan Plateau. *Theor. Appl. Climatol.* 137, 141–152. doi:10.1007/s00704-018-2564-8

Zou, H., and Hastie, T. (2005). Regularization and Variable Selection via the Elastic Net. *J. R. Stat. Soc B.* 67, 301–320. doi:10.1111/j.1467-9868.2005.00503.x

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

frontiers
in Artificial Intelligence

ORIGINAL RESEARCH
published: 10 September 2021
doi: 10.3389/frai.2021.670208

Application of Video-to-Video Translation Networks to Computational Fluid Dynamics

*Hiromitsu Kigure**

Independent Researcher, Kanagawa, Japan

In recent years, the evolution of artificial intelligence, especially deep learning, has been remarkable, and its application to various fields has been growing rapidly. In this paper, I report the results of the application of generative adversarial networks (GANs), specifically video-to-video translation networks, to computational fluid dynamics (CFD) simulations. The purpose of this research is to reduce the computational cost of CFD simulations with GANs. The architecture of GANs in this research is a combination of the image-to-image translation networks (the so-called "pix2pix") and Long Short-Term Memory (LSTM). It is shown that the results of high-cost and high-accuracy simulations (with high-resolution computational grids) can be estimated from those of low-cost and low-accuracy simulations (with low-resolution grids). In particular, the time evolution of density distributions in the cases of a high-resolution grid is reproduced from that in the cases of a low-resolution grid through GANs, and the density inhomogeneity estimated from the image generated by GANs recovers the ground truth with good accuracy. Qualitative and quantitative comparisons of the results of the proposed method with those of several super-resolution algorithms are also presented.

Keywords: deep learning, generative adversarial networks (GANs), image-to-image translation networks (pix2pix), long short-term memory (LSTM), computational fluid dynamics (CFD)

OPEN ACCESS

Edited by:
Alex Hansen,
Norwegian University of Science and Technology, Norway

Reviewed by:
Bernhard C. Geiger,
Know Center, Austria
Ajey Kumar,
Symbiosis International (Deemed University), India

Correspondence:
Hiromitsu Kigure
hiro.kig@gmail.com

Specialty section:
This article was submitted to Machine Learning and Artificial Intelligence,
a section of the journal Frontiers in Artificial Intelligence

Received: *20 February 2021*
Accepted: *17 August 2021*
Published: *10 September 2021*

Citation:
Kigure H (2021) Application of Video-to-Video Translation Networks to Computational Fluid Dynamics. Front. Artif. Intell. 4:670208. doi: 10.3389/frai.2021.670208

1 INTRODUCTION

Artificial intelligence is advancing rapidly and has come to be comparable to or outperform humans in several tasks. In generic object recognition, deep convolutional neural networks have surpassed human-level performance (e.g, He et al., 2015; He et al., 2016; Ioffe and Szegedy, 2015). The agent trained by reinforcement learning is capable of reaching a level comparable to professional human game testers (Mnih et al., 2015). In the case of machine translation, Google's neural machine translation system, using Long Short-Term Memory (LSTM) recurrent neural networks [Hochreiter and Schmidhuber (1997), Gers et al. (2000)], is a typical and famous example and its translation quality is becoming comparable to that of humans (Wu et al., 2016).

One of the hottest research topics in artificial intelligence is generative models and one approach to implementing a generative model is generative adversarial networks (GANs) proposed by Goodfellow et al. (2014). GANs consist of two models trained with conflicting objectives. Radford et al. (2016) applied deep convolutional neural networks to those two models, whose architecture is called deep convolutional GANs (DCGAN). DCGAN can generate realistic synthesis images from vectors in the latent space. Isola et al. (2017) proposed the network learning the mapping from an input image to an output image to enable the translation between two images. This network, the so-called pix2pix, can convert black-and-white images into color images, line drawings into photo-realistic images, and so on.

The combination of deep learning and simulation has been recently researched. One of such applications is to use simulation results for improving the prediction performance of deep learning. Since deep learning requires a lot of data for training, numerical simulations that can generate various data by changing physical parameters could help compensate for the lack of training data. Another application is to speed up the solver of computational fluid dynamics (CFD). Guo et al. (2016) used a convolutional neural network (CNN) to predict velocity fields approximately but fast from the geometric representation of the object. Another example is that velocity fields are predicted from parameters such as source position, inflow speed, and time by CNN (Kim et al. (2019)). Their method is feasible to generate velocity fields up to 700 times faster than simulations. As a more general method, not limited to CFD problems, Raissi et al. (2019) proposed the physics-informed neural network (PINN), which utilizes a relatively simple deep neural network to find solutions to various types of nonlinear partial differential equations.

GANs also have been combined with numerical simulations to enable a new type of solution method. Farimani et al. (2017) used the conditional GAN (cGAN) to generate the solution of steady-state heat conduction and incompressible flow from boundary conditions and calculation domain shape/size. Xie et al. (2018) proposed a method for super-resolution fluid flow by a temporally coherent generative model (tempoGAN). They showed that tempoGAN can infer high-resolution, temporal, and volumetric physical quantities from those of low-resolution data.

The above-mentioned studies about the combination of GANs and simulations show that GANs can generate the three-dimensional data of the solution of physical equations. The main topic in this research is the translation of images (distributions of the physical quantity) by GANs. In the case that the accuracy of the simulation is particularly important, a large number of computational grids are needed. Additionally, the number of simulation cases for design optimization is typically numerous. It means that the computational cost (machine power and time) becomes large. In such a case, it is important to reduce the computational cost, and one way to do so is to make effective use of low-cost simulations. Based on such an idea, I investigated the feasibility of time-series image-to-image translation: translation from time-series distribution plots in the case of low-resolution computational grids to those in the case of high-resolution grids. A quantitative evaluation of the quality of generated images was also performed.

The method proposed in this paper is the video (sequential images)-to-video translation in which the difference of solutions between the high- and low-resolution grid simulations is learned. Meanwhile, the PINN constructs universal function approximators of physical laws by minimizing the loss function composed of a mismatch of state variables including the initial and boundary conditions and the residual for the partial differential equations (Meng et al., 2020). In other words, the PINN is an alternative to CFD, while the proposed method is a complement to CFD.

The paper is organized as follows. In **section 2**, I describe the outline of the simulations whose results are input to GANs and the details of the network architecture. In **section 3**, I give the results of time-series image-to-image translation (in other words, video-to-video translation) of physical quantity distribution and a discussion mainly about the quality of generated images. Conclusions are presented in **section 4**.

TABLE 1 | The initial conditions of simulations.

Physical quantity	Description	Value
ρ	Density	$25\pi/36$
v_x	x-component of velocity	$-\sin(2\pi y)$
v_y	y-component of velocity	$\sin(2\pi x)$
v_z	z-component of velocity	0
B_x	x-component of magnetic field	$-B_0\sin(2\pi y)$
B_y	y-component of magnetic field	$B_0\sin(4\pi x)$
B_z	z-component of magnetic field	0
p	Pressure	$5\pi/12$

2 METHODS

2.1 Numerical Simulations

I solved the following ideal magnetohydrodynamic (MHD) equations numerically in two dimensions to prepare input images to GANs:

$$\frac{\partial\rho}{\partial t}+\nabla\cdot(\rho\boldsymbol{v})=0 \tag{1}$$

$$\frac{\partial}{\partial t}(\rho\boldsymbol{v})+\nabla\cdot\left(\rho\boldsymbol{v}\boldsymbol{v}+p_T\boldsymbol{I}-\boldsymbol{B}\boldsymbol{B}\right)=0 \tag{2}$$

$$\frac{\partial\boldsymbol{B}}{\partial t}+\nabla\cdot(\boldsymbol{v}\boldsymbol{B}-\boldsymbol{B}\boldsymbol{v})=0 \tag{3}$$

$$\frac{\partial e}{\partial t}+\nabla\cdot\left((e+p_T)\boldsymbol{v}-\boldsymbol{B}(\boldsymbol{v}\cdot\boldsymbol{B})\right)=0 \tag{4}$$

$$p_T=p+\frac{|\boldsymbol{B}|^2}{2} \tag{5}$$

$$e=\frac{p}{\gamma-1}+\frac{\rho|\boldsymbol{v}|^2}{2}+\frac{|\boldsymbol{B}|^2}{2} \tag{6}$$

where ρ, p, and $\boldsymbol{v}$ are the density, pressure, and velocity of the gas; $\boldsymbol{B}$ is the magnetic field; γ represents the heat capacity ratio and is equal to 5/3 in this paper; p_T and e represent the total pressure and the internal energy density; $\boldsymbol{I}$ is the unit matrix.

One of typical test problems for MHD, the so-called Orszag-Tang vortex problem (Orszag and Tang, 1979), was solved by the Roe scheme (Roe 1981) with MUSCL [monotonic upstream-centered scheme for conservation laws; (van Leer 1979)]. The initial conditions are summarized in **Table 1**. B_0 is a parameter for controlling the magnetic field strength. The compuational domain is $0 \le x \le 1$ and $0 \le y \le 1$. The periodic boundary condition is applied in both x- and y-directions. Simulations for each condition were performed twice on computational grids with different resolutions. The number of grid points is $(N_x \times N_y) = (51 \times 51)$ or (251×251). In the case of $(N_x \times N_y) = (251 \times 251)$, the calculation time is more than 70 times longer than in the other case though the obtained solution is expected to be close to the true solution.

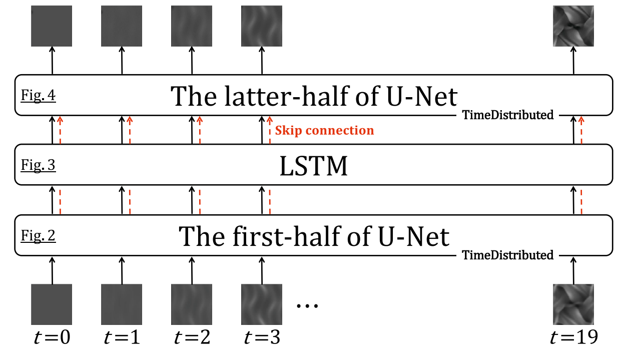


FIGURE 1 | Schematic picture of the architecture of the generator in this research. The generator in the original pix2pix network is a U-shaped network (U-Net). In this research, the LSTM layer is inserted into the middle of U-Net. The skip connections from the first-half of U-Net to the latter-half over the LSTM layer are implemented.

2.2 Generative Adversarial Network Architecture

After the original concept of GANs was proposed by Goodfellow et al. (2014), various GANs have been researched. Among such networks, I focused on pix2pix, which is a type of conditional GAN and a network for learning the relationship between the input and output images. The feasibility of translating from the results of low-resolution grid simulations to those of high-resolution grid simulations has been investigated in this research. Furthermore, in order to enable the translation across two time-series, the architecture combined pix2pix and LSTM has been constructed.

Figure 1 shows the schematic picture of the architecture of the generator in this research. The role of the LSTM layer is to adjust the image translation dependent on the physical time of the simulation; for the initial state of the simulation ($T = 0$), no translation is needed at all, but as physical time passes, progressively larger translations are needed. Note that the weights of the encoder (decoder) before (after) the LSTM layer are the same in the time direction.

Plots of the time evolution of the density in the low-resolution simulations are input to the generator (plots are read as single-channel images). The input images are converted to vectors by the first-half of a U-shaped network (U-Net). In **Figure 2**, I denote the architecture of the first-half of U-Net in detail. It consists of eight convolutional blocks with a kernel size of (4×4) or (2×2). The instance normalization (Ulyanov et al. (2017)) is applied except for the first and last blocks. The activation function is a leaky rectified linear unit [leaky ReLU; Maas et al. (2013)] with a slope of 0.2 for all blocks. A 512-dimensional vector is generated at the end of this architecture.

A series of 512-dimensional vectors converted from the time-series plots is input to the LSTM layer. An input vector $\boldsymbol{x}_t$ originated from the plot at time $= t$ is calculated with the hidden state $\boldsymbol{h}_{t-1}$ and memory cell $\boldsymbol{c}_{t-1}$. A forget gate $(\boldsymbol{f})$, an input gate $(\boldsymbol{i})$, an output gate $(\boldsymbol{o})$, and part of the term to be added to the memory cell $(\boldsymbol{z})$ in **Figure 3** are calculated as follows:

$$\boldsymbol{f} = \sigma\left(\boldsymbol{W}_f \boldsymbol{x}_t + \boldsymbol{R}_f \boldsymbol{h}_{t-1} + \boldsymbol{b}_f\right) \tag{7}$$

$$\boldsymbol{i} = \sigma\left(\boldsymbol{W}_i \boldsymbol{x}_t + \boldsymbol{R}_i \boldsymbol{h}_{t-1} + \boldsymbol{b}_i\right) \tag{8}$$

$$\boldsymbol{o} = \sigma\left(\boldsymbol{W}_o \boldsymbol{x}_t + \boldsymbol{R}_o \boldsymbol{h}_{t-1} + \boldsymbol{b}_o\right) \tag{9}$$

$$\boldsymbol{z} = \tanh\left(\boldsymbol{W}_z \boldsymbol{x}_t + \boldsymbol{R}_z \boldsymbol{h}_{t-1} + \boldsymbol{b}_z\right) \tag{10}$$

where σ is the sigmoid function and tanh is the hyperbolic tangent function; $\boldsymbol{W}_{\cdot}$ and $\boldsymbol{R}_{\cdot}$ are the input-to-hidden weight matrices and the recurrent weight matrices; $\boldsymbol{b}_{\cdot}$ are bias vectors. The hidden state and memory cell are updated by:

$$\boldsymbol{c}_t = \boldsymbol{f} \odot \boldsymbol{c}_{t-1} + \boldsymbol{i} \odot \boldsymbol{z} \tag{11}$$

$$\boldsymbol{h}_t = \boldsymbol{o} \odot \tanh\left(\boldsymbol{c}_t\right) \tag{12}$$

The hidden state $\boldsymbol{h}_t$ is reshaped as $(1, 1, 512)$. The reshaped hidden state $\boldsymbol{h}_t$' is passed to the latter-half of U-Net and is decoded to the image data (**Figure 4**). This part consists of eight deconvolutional blocks with an upsampling of the feature map, convolution with a kernel size of (2×2)or (4×4) (the size of the feature map does not change because the stride

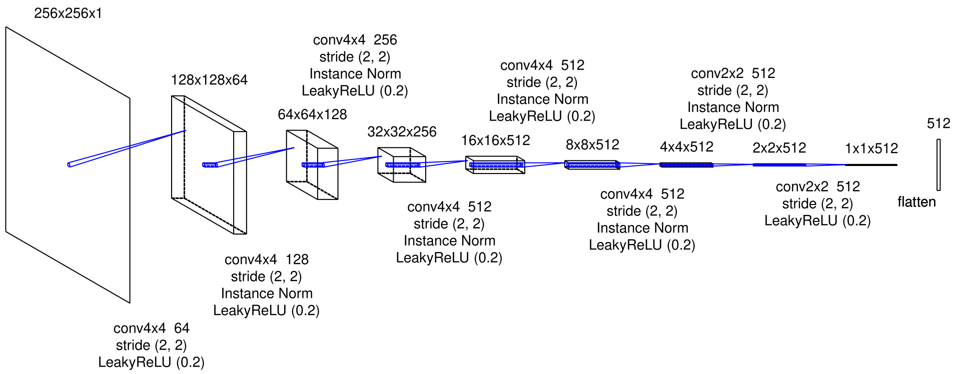


FIGURE 2 | The details of the first-half of U-Net. The expression "conv4x4 64" refers to a convolutional layer with a kernel size of (4×4) and 64 channels. Each feature map is copied and is concatenated to the feature map of the corresponding block in the latter-half of U-Net denoted in **Figure 4**.

of convolution is 1), the instance normalization and activation by ReLU function except the last block. As seen in **Figure 1**, the generator outputs synthetic time-series plots of density distribution.

The authenticity of the images is judged by the discriminator. **Figure 5** shows the details of the architecture of the discriminator in this research. A real image (plot of the density distribution in a high-resolution simulation) or a synthesis image is input to the discriminator. It consists of five convolutional blocks with a kernel size of (4×4). The instance normalization is applied except for the first and last blocks. Except for the last block, the leaky ReLU function with a slope of 0.2 is applied as the activation function. The 16 × 16 patch is eventually output. The discriminator classifies each patch into real or synthetic. We call its architecture the patchGAN (Isola et al., 2017).

The objective of the network is the same as the regular pix2pix as follows:

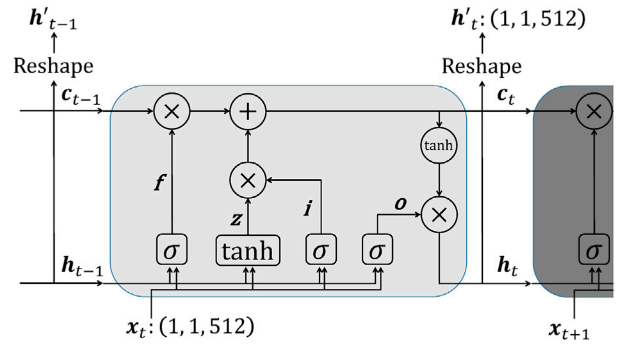


FIGURE 3 | The architecture of LSTM. The input to LSTM ($\boldsymbol{x}_t$) is the vector transformed from an image of density distribution, and the output is the reshaped hidden state vector ($\boldsymbol{h}'_t$) resulting from several operations. The vector $\boldsymbol{c}$ is the memory cell, and $\boldsymbol{f}$, $\boldsymbol{i}$, $\boldsymbol{o}$, and $\boldsymbol{z}$ are a forget gate, an input gate, an output gate, and part of the term to be added to the memory cell (see equations (7) to (10) for details).

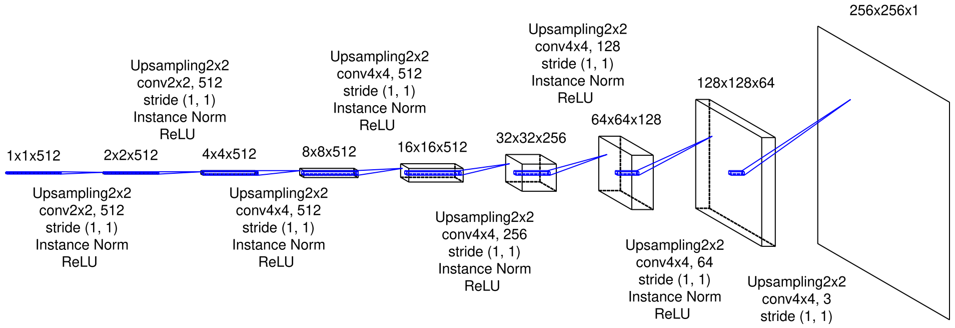


FIGURE 4 | The details of the latter-half of U-Net. The expression "Upsampling2x2" refers to an upsampling layer that doubles the size of input by copying one value twice horizontally and vertically, respectively. From the first-half of U-Net displayed in **Figure 2**, feature maps are passed to corresponding blocks and are concatenated to the feature maps output from the previous blocks.

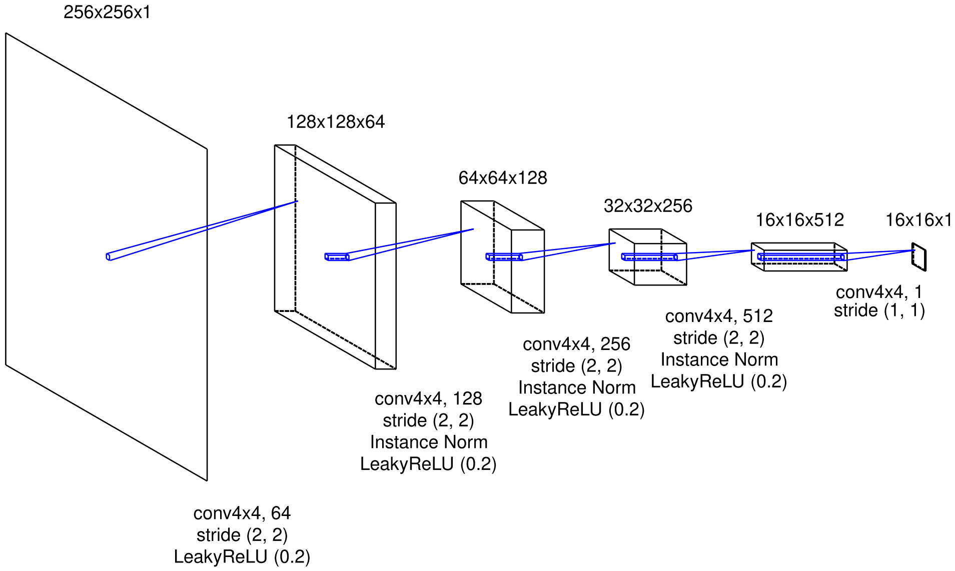


FIGURE 5 | The details of the architecture of the discriminator in this research.

TABLE 2 | The details of the training and testing datasets.

Training/Testing	The number of cases	The total number of images	The pixel size of images	The value of B_0
Training	16	320	256 × 256	0.1–1.5 with the interval of 0.1, and 2.0
Testing	19	380	256 × 256	0.15–1.55 and 1.6–1.9 with the interval of 0.1

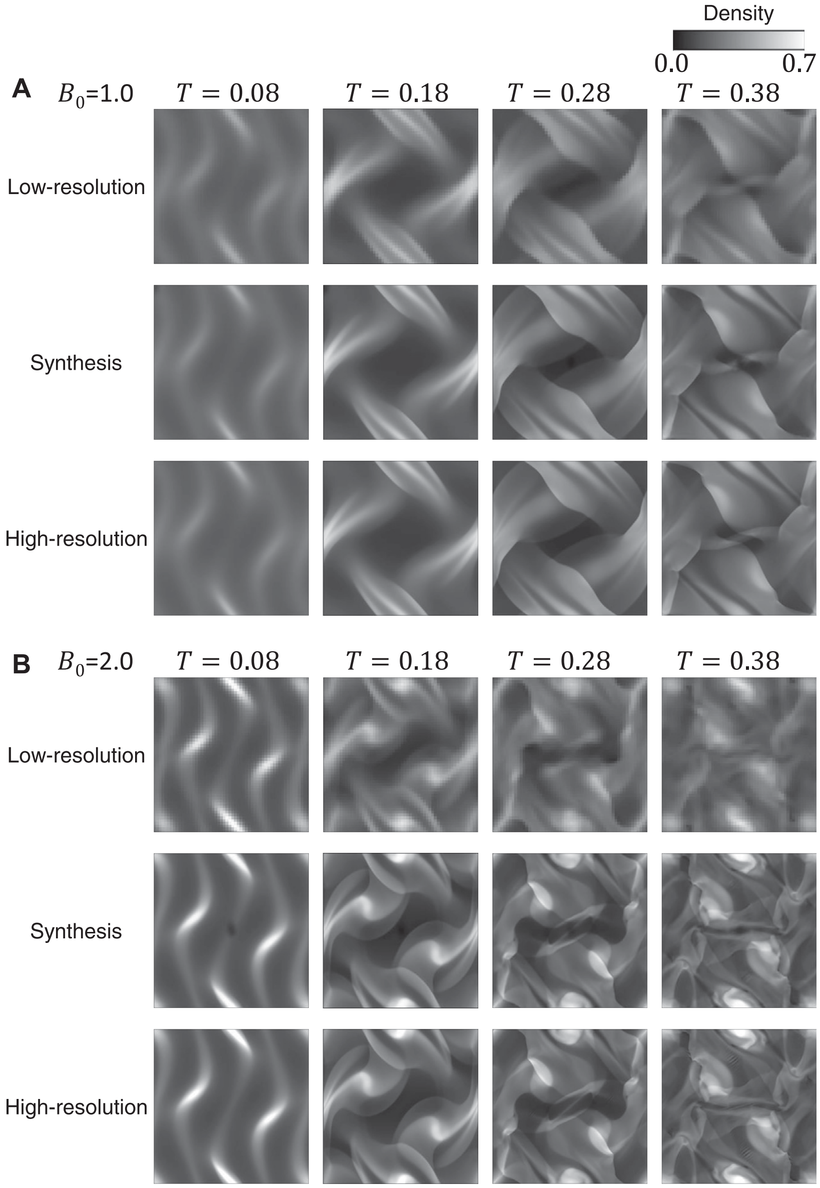


FIGURE 6 | Two examples of the time-evolution of density distribution for the training datasets.

$$G^{*} = \arg\min_{G}\max_{D} \mathcal{L}_{cGAN}(G,D) + \lambda\mathcal{L}_{L1}(G) \tag{13}$$

$$\mathcal{L}_{cGAN}(G,D) = \mathbb{E}[\log, D(x,y)] + \mathbb{E}[\log(1 - D(x,G(x)))] \tag{14}$$

$$\mathcal{L}_{L1}(G) = \mathbb{E}[\|y - G(x)\|] \tag{15}$$

where G and D denote the generator and discriminator, λ is the weighted sum parameter and equal to 100 in this research, and x and y mean the source and target images. $G(x)$ returns a synthesis image and $D(x,y)$ or $D(x,G(x))$ returns the probability that y or $G(x)$ is a real target image. $\mathcal{L}_{L1}(G)$ is the mean absolute error (L1 loss) calculated from the pixel-wise comparison between the real image and the synthetic image. The optimizer is Adam with a learning rate of 0.0002.

The architecture is implemented using Keras 2.5.0 and TensorFlow as a backend. The model was trained on Google

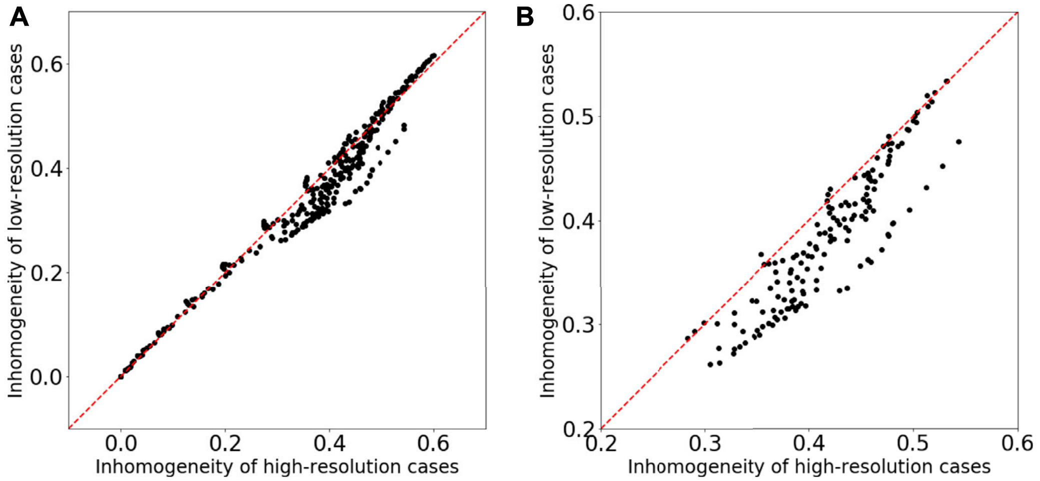


FIGURE 7 | Comparison of the inhomogeneity of the density between the high-resolution grid cases and the low-resolution grid cases for the training datasets. **(A)** The inhomogeneities for all time-series and all magnetic field strength cases are plotted. **(B)** The inhomogeneities for $T \geq 0.12$ and $B_0 \geq 0.6$ are plotted.

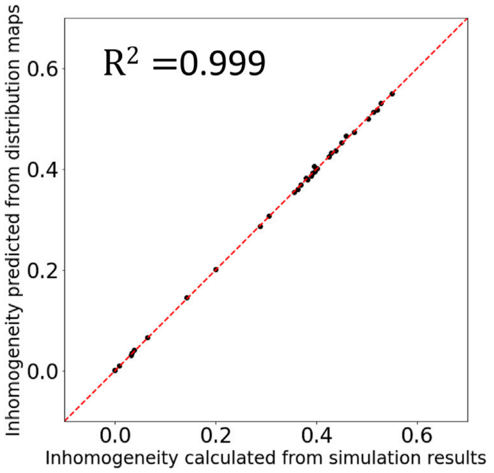


FIGURE 8 | Comparison of the inhomogeneity calculated from the density values on the grids and the inhomogeneity predicted from the distribution maps. The coefficient of determination (R^2) is equal to 0.999.

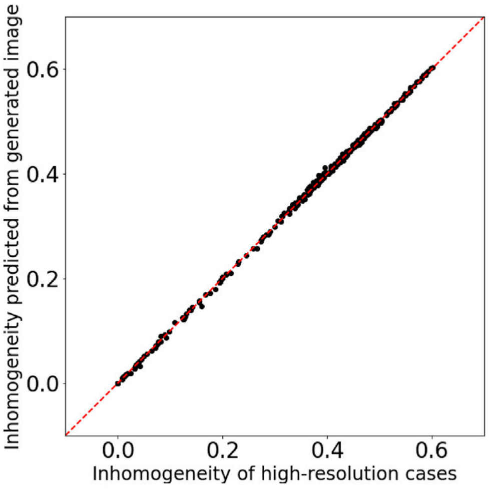


FIGURE 9 | Comparison of the inhomogeneity of the high-resolution grid simulation results and the inhomogeneity predicted from the synthesis images for the training datasets.

Colaboratory with Tesla P100-PCIe GPU. For applying the convolution and deconvolution to the sequential data, sets of operations as shown in **Figures 2,4,5** are passed to the TimeDistributed layer. The skip connections are implemented by feeding the outputs of the previous upsampling block in the latter-half of U-Net and the same-level (it means that the size of the feature map is the same) convolutional block in the first-half of U-Net to the Concatenate layer. The "return_sequences" and "stateful" parameters in the LSTM layer are set to True and False, respectively.

3 RESULTS AND DISCUSSION

In this chapter, I show the results of time-series image-to-image translation for the training datasets first and then explain the way to evaluate the quality of the synthesis images quantitatively. The evaluation result of the synthesis images for the training datasets is presented next. Then, I show the results for the testing datasets. Finally, the quality of the synthesis images is compared with those of images upsampled by conventional super-resolution algorithms. The conditions (the magnetic field strength) of the simulations are shown in **Table 2** that summarizes the details of the training and testing datasets. The sixteen cases were performed to prepare the training datasets, and the nineteen cases were performed to prepare the testing datasets. For each case, two simulations were run with the high-resolution and the low-resolution grids.

3.1 Results for the Training Datasets

Figure 6 shows two examples of the time-evolution of density distribution for the training datasets. The top and bottom images

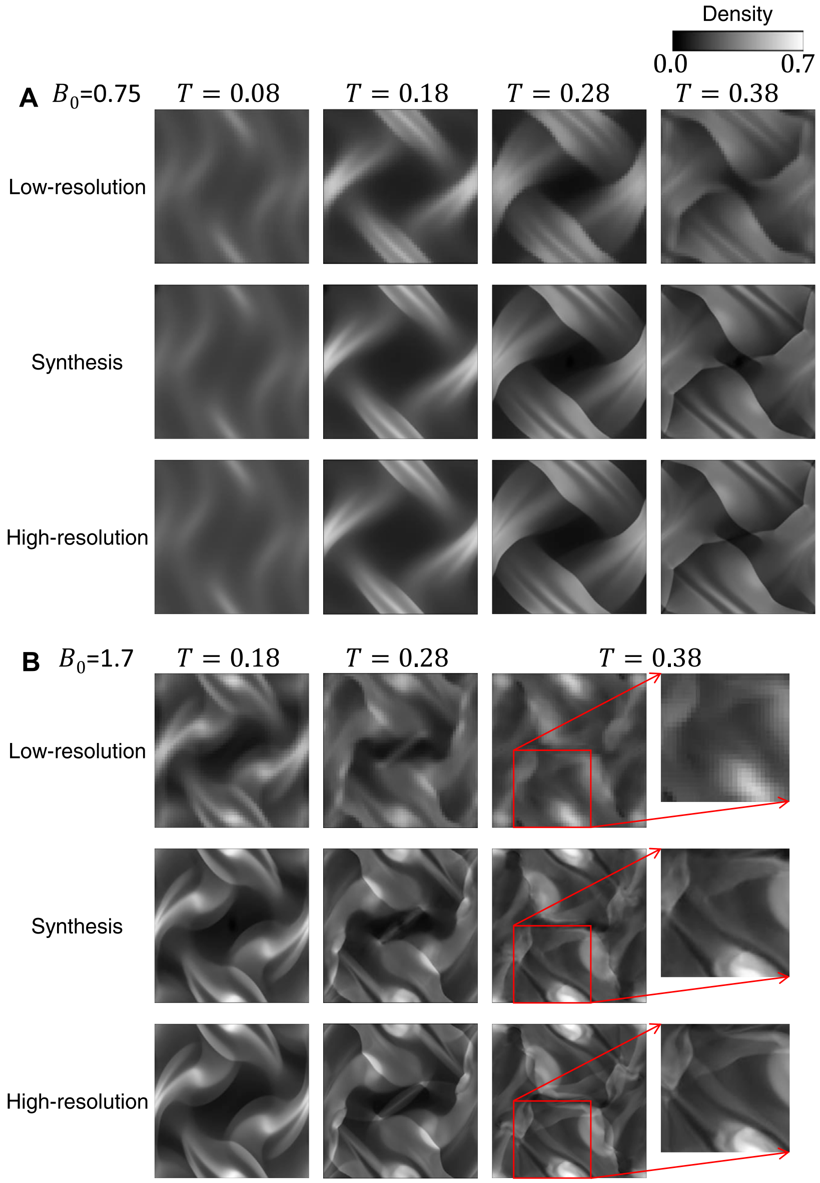


FIGURE 10 | Examples of the time-evolution of density distribution for the testing datasets.

of **Figures 6A,B** show the simulation results, and the middle images are synthesis ones generated from the top ones (the results of low-resolution grid simulations) through the generator. Compared to the high-resolution grid cases, the density distributions in the low-resolution grid cases show less fine structure and become closer to the uniform. **Figure 7** displays the comparison of the inhomogeneity of the density between the high-resolution grid cases and the low-resolution grid cases. The inhomogeneity is defined by $\alpha = \sigma_\rho/\bar{\rho}$, where σ_ρ and $\bar{\rho}$ are the standard deviation and the average of the density. In the low-resolution grid, the numerical diffusion is larger than in the high-resolution grid, and therefore the inhomogeneity of the density tends to be smaller especially from the middle stage of the vortex development and in the relatively strong magnetic field (**Figure 7B**). The synthesis images reproduce the fine structures of the density distributions and appear to be well consistent with the high-resolution grid results.

To quantitatively evaluate the quality of the synthesis images, I estimated the density inhomogeneity from the distribution map. When calculating the density inhomogeneity from the simulation result, we can use the value of the density on each grid; however, the density distribution maps (including synthesis images in this research) have only the information of the RGB values. Therefore, to estimate the density inhomogeneity from the distribution map, I trained a three-layer fully connected neural network with 196,608 (256pixel × 256pixel × 3) inputs, two hidden layers of 1,024 and 128 neurons and one output layer. **Figure 8** shows the result of the inhomogeneity prediction from the density

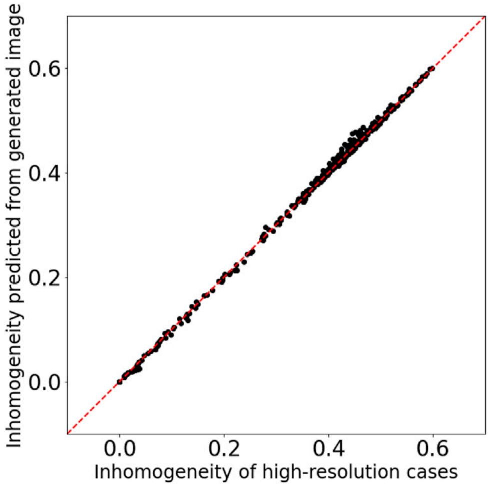


FIGURE 11 | Comparison of the inhomogeneity of the high-resolution grid simulation results and the inhomogeneity predicted from the synthesis images for the testing datasets.

distribution maps. The horizontal axis is the inhomogeneity calculated from the density values on the grids, and the vertical axis is the inhomogeneity predicted from the distribution maps by the trained neural network. The coefficient of determination (R^2) is equal to 0.999. Thus we conclude that the trained neural network provides an accurate estimation of the density inhomogeneity from the distribution maps and the synthesis images.

We can quantitatively evaluate the quality of the synthesis images by inputting those into the neural network and comparing the outputted inhomogeneity with the inhomogeneity calculated from the high-resolution grid simulation results. **Figure 9** shows that the inhomogeneity predicted from the synthesis images matches that calculated from the high-resolution grid simulation results with good accuracy; therefore the quality of the synthesis images is definitely good for the training datasets.

3.2 Results for the Testing Datasets

In the previous subsection, I have shown that the results for the training datasets are pretty good. However, the generalization ability needs to be investigated for practical use. The testing datasets (the magnetic field strength is different from the training datasets as shown in **Table 2**) that were not used for training are input to the trained model, and the synthesis images are output from the generator. **Figures 10A,B** show the comparison of the simulation results and the synthesis images for two example cases. From the 19 cases in the testing datasets, the results for the cases with $B_0 = 0.75$ and 1.7 were selected for presentation. The $B_0 = 1.7$ case is especially suitable for verifying the generalization ability because there is no training data between $B_0 = 1.5$ and 2.0. The top images show the time evolution of the density distribution of low-resolution grid simulations, which are input for the generator; the bottom images show that of high-resolution grid simulations, which are compared with the synthesis images; the middle images are synthesis ones generated through the generator. As with the cases for the training datasets, the synthesis images qualitatively reproduce the density distributions of the high-resolution grid simulations. Even in the $B_0 = 1.7$ case, the synthesis images show the fine structure of the density distribution very similar to that in the ground truth images, as shown in the zoomed-in image in **Figure 10B**.

Figure 11 is almost the same as **Figure 9** but for the testing datasets. The density inhomogeneity predicted from the synthesis images through the fully connected neural network (explained in the previous subsection) is in good agreement with the inhomogeneity calculated from the results

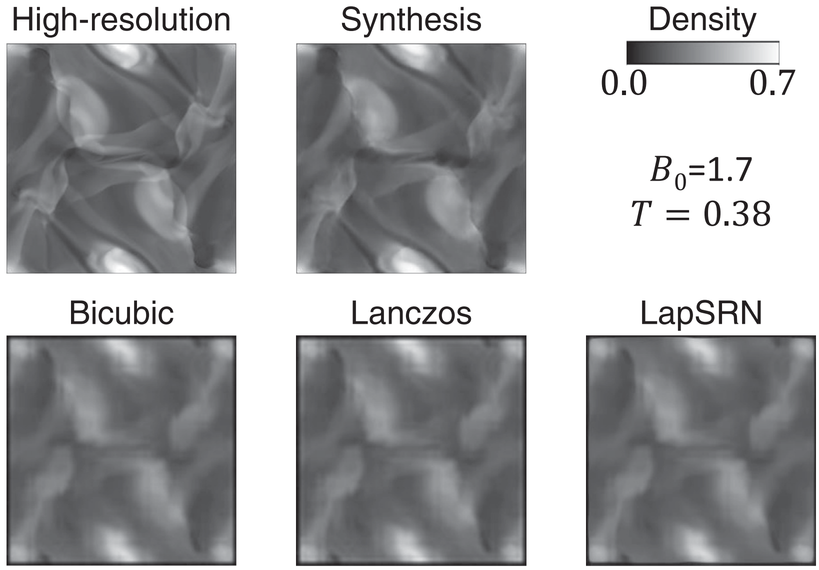


FIGURE 12 | Comparison of the results of the conventional super-resolution algorithms with that of the proposed method and ground truth.

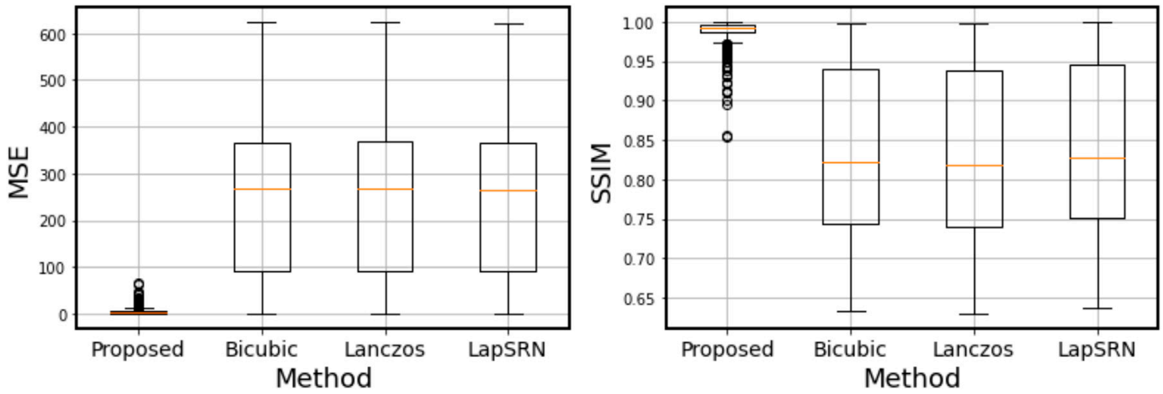


FIGURE 13 | Box plots of the pixel-wise mean squared error (MSE) and the structural similarity index measure (SSIM) calculated in the testing datasets (380 images).

of high-resolution grid simulations. This result indicates that the method in this research is capable of obtaining high generalization ability.

3.3 Comparison With Conventional Super-resolution Algorithms

To demonstrate the effectiveness of the proposed method and the quality of the generated images, I compare the results with those obtained by conventional super-resolution algorithms. The algorithms investigated here are a bicubic interpolation, a Lanczos interpolation, and Laplacian Pyramid Super-Resolution Network [LapSRN; Lai et al. (2017)]. The pixel size of the image to be used as the basis of the super-resolution is 64 × 64, and each algorithm quadruples the pixel size. These results were compared qualitatively and quantitatively with the result of high-resolution grid simulation and the image generated by the proposed method. Plots of the density distribution in high-resolution simulations in the training datasets were used to train LapSRN.

I performed the super-resolution algorithms to the testing datasets (380 images). As an example, the results for the $B_0 = 1.7$ and $T = 0.38$ case are compared in **Figure 12**. In this case, none of the three conventional super-resolution algorithms can work with a quality comparable to the method proposed in this research. To compare the proposed method with the others quantitatively, the pixel-wise mean squared error (MSE) and the structural similarity index measure [SSIM; Wang et al. (2004)] are calculated between the ground truth image and the synthesis image or the result of super-resolution. **Figure 13** shows that the quality of the synthesis images by the proposed method is significantly high compared to that of the results by the conventional super-resolution algorithms.

3.4 Application of This Research

In this subsection, I discuss an application of this research. As mentioned above, results of high computational cost simulations can be estimated from those of low-cost simulations by the method in this paper. However, it is important to note that simulation results of quite a few cases are needed to train the network[1]. Therefore, it is not beneficial for a small number of simulations. The more simulations are required, the greater the benefits arise. One such case is optimization based on CFD simulations. As the number of objective variables to be optimized increases, the number of calculations required to obtain the desired performance is expected to increase; in some cases, it takes several thousand cases to evaluate. In such multi-objective optimization simulations, for example, the first dozens to several hundred cases are simulated on both high- and low-resolution grids, and the results are used to train the GANs. After the GANs are trained, low-resolution grid simulations are run, the results are input to the GANs to reproduce the results of high-resolution grid simulations, and objective variables are estimated from synthesis images by, for example, a neural network.

I demonstrate the estimation of computational cost reduction. If the number of simulations required originally and that to train the GANs are N (several thousands in some cases) and N_t ($N > N_t$), the calculation times of the high- and low-resolution grid simulations are T_h and T_l ($T_h > T_l$), and the computational cost to train the GANs is T_t, the computational cost reduction is roughly equal to

$$N \times T_h - (N_t \times T_h + T_t + N \times T_l) \tag{16}$$

where the first term corresponds to the computational cost in the case that all simulations are run on the high-resolution grid, and the second term corresponds to that in the case that the method in this research is applied (the cost to reproduce the results of high-resolution grid simulations by the GANs is negligible compare to performing the simulations). In this way, by substituting low-resolution grid simulations and the result conversion by the GANs for quite a part of high-resolution grid simulations, a great reduction of the computational cost should be achieved.

[1] In this research, simulation results of 16 cases were used as the training datasets; the training was successful with a relatively small number of data, probably due to the simple situation. If the target is a simulation of a realistic engineering situation, it is expected that much more data will be needed for the training.

4 CONCLUSION

In this paper, I validated an idea to use GANs for reducing the computational cost of CFD simulations. I studied the idea of reproducing the results of high-resolution grid simulations with a high computational cost from those of low-resolution grid simulations with a low computational cost. More specifically speaking, distribution maps of a physical quantity in time series were reproduced using pix2pix and LSTM. The quality of the reproduced synthesis images was good for both the training and testing datasets. The conditions treated in this paper are simple; the computational region is a square with a constant grid interval, the boundary conditions are cyclic, and the governing equations are the ideal MHD equations. In the next step, I need to examine the idea in more realistic conditions.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

HK: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Visualization, Writing.

REFERENCES

Farimani, A. B., Gomes, J., and Pande, V. S. (2017). *Deep Learning the Physics of Transport Phenomena* arXiv.

Gers, F. A., Schmidhuber, J., and Cummins, F. (2000). Learning to Forget: Continual Prediction with Lstm. *Neural Comput.* 12, 2451–2471. doi:10.1162/089976600300015015

Goodfellow, I., Pouget-Abadie, J., and Mirza, M., (2014). “Generative Adversarial Nets,” in *Advances in Neural Information Processing Systems*, 27, 2672–2680.

Guo, X., Li, W., and Iorio, F. (2016). “Convolutional Neural Networks for Steady Flow Approximation,” in KDD ’16: The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, California (New York: Association for Computing Machinery), 481–490. doi:10.1145/2939672.2939738

He, K., Zhang, X., Ren, S., and Sun, J. (2016). “Deep Residual Learning for Image Recognition,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, Nevada (Manhattan, New York: IEEE). doi:10.1109/cvpr.2016.90

He, K., Zhang, X., Ren, S., and Sun, J. (2015). “Delving Deep into Rectifiers: Surpassing Human-Level Performance on Imagenet Classification,” in Proceedings of the IEEE International Conference on Computer Vision (ICCV). doi:10.1109/iccv.2015.123

Hochreiter, S., and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Comput.* 9, 1735–1780. doi:10.1162/neco.1997.9.8.1735

Ioffe, S., and Szegedy, C. (2015). “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” in *Proceedings of Machine Learning Research* PMLR (Proceedings of Machine Learning Research).

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017). “Image-to-image Translation with Conditional Adversarial Networks,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, Hawaii (Manhattan, New York: IEEE). doi:10.1109/cvpr.2017.632

Kim, B., Azevedo, V. C., Thuerey, N., Kim, T., Gross, M., and Solenthaler, B. (2019). Deep Fluids: A Generative Network for Parameterized Fluid Simulations. *Comp. Graphics Forum* 38, 59–70. doi:10.1111/cgf.13619

Lai, W.-S., Huang, J.-B., Ahuja, N., and Yang, M.-H. (2017). “Deep Laplacian Pyramid Networks for Fast and Accurate Super-resolution,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, Hawaii (Manhattan, New York: IEEE). doi:10.1109/cvpr.2017.618

Maas, A. L., Hannun, A. Y., and Ng, A. Y. (2013). “Rectifier Nonlinearities Improve Neural Network Acoustic Models,” in *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*. Editors S. Dasgupta and D. McAllester JMLR.

Meng, X., Li, Z., Zhang, D., and Karniadakis, G. E. (2020). Ppinn: Parareal Physics-Informed Neural Network for Time-dependent Pdes. *Comp. Methods Appl. Mech. Eng.* 370, 113250. doi:10.1016/j.cma.2020.113250

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., et al. (2015). Human-level Control through Deep Reinforcement Learning. *Nature* 518, 529–533. doi:10.1038/nature14236

Orszag, S. A., and Tang, C.-M. (1979). Small-scale Structure of Two-Dimensional Magnetohydrodynamic Turbulence. *J. Fluid Mech.* 90, 129–143. doi:10.1017/s002211207900210x

Radford, A., Metz, L., and Chintala, S. (2016). “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks,” in 4th International Conference on Learning Representations, San Juan, Puerto Rico (arXiv).

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019). Physics-informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations. *J. Comput. Phys.* 378, 686–707. doi:10.1016/j.jcp.2018.10.045

Roe, P. L. (1981). Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes. *J. Comput. Phys.* 43, 357–372. doi:10.1016/0021-9991(81)90128-5

Ulyanov, D., Vedaldi, A., and Lempitsky, V. (2017). “Improved Texture Networks: Maximizing Quality and Diversity in Feed-Forward Stylization and Texture Synthesis,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, Hawaii (Manhattan, New York: IEEE). doi:10.1109/cvpr.2017.437

van Leer, B. (1979). Towards the Ultimate Conservative Difference Scheme. V. A Second-Order Sequel to Godunov’s Method. *J. Comput. Phys.* 32, 101–136. doi:10.1016/0021-9991(79)90145-1

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image Quality Assessment: from Error Visibility to Structural Similarity. *IEEE Trans. Image Process.* 13, 600–612. doi:10.1109/TIP.2003.819861

Wu, Y., Schuster, M., and Chen, Z., (2016). *Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation* arXiv. e-prints , arXiv:1609.08144.

Xie, Y., Franz, E., Chu, M., and Thuerey, N. (2018). Tempogan: A Temporally Coherent, Volumetric gan for Super-resolution Fluid Flow. *ACM Trans. Graph.* 37, 1–15. doi:10.1145/3197517.3201304

Conflict of Interest: The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher’s Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Advantages of publishing in Frontiers

OPEN ACCESS
Articles are free to read
for greatest visibility
and readership

FAST PUBLICATION
Around 90 days
from submission
to decision

HIGH QUALITY PEER-REVIEW
Rigorous, collaborative,
and constructive
peer-review

TRANSPARENT PEER-REVIEW
Editors and reviewers
acknowledged by name
on published articles

Frontiers
Avenue du Tribunal-Fédéral 34
1005 Lausanne | Switzerland

Visit us: www.frontiersin.org
Contact us: frontiersin.org/about/contact

REPRODUCIBILITY OF RESEARCH
Support open data
and methods to enhance
research reproducibility

DIGITAL PUBLISHING
Articles designed
for optimal readership
across devices

FOLLOW US
@frontiersin

IMPACT METRICS
Advanced article metrics
track visibility across
digital media

EXTENSIVE PROMOTION
Marketing
and promotion
of impactful research

LOOP RESEARCH NETWORK
Our network
increases your
article's readership