Brain-Mamba: Encoding Brain Activity via Selective State Space Models

Ali Behrouz AB2947@CORNELL.EDU
Cornell University, USA

Farnoosh Hashemi SH2574@CORNELL.EDU
Cornell University, USA

Abstract

Representation learning of brain activity is a key step toward unleashing machine learning models for use in the diagnosis of neurological diseases/disorders. Diagnosis of different neurological diseases/disorders, however, might require paying more attention to either spatial or temporal resolutions of brain activity. Accordingly, a generalized brain activity learner requires the ability of learning from both resolutions. Most existing studies, however, use domain knowledge to design brain encoders, and so are limited to a single neuroimage modality (e.g., EEG or fMRI) and its single resolution. Furthermore, their architecture design either: (1) uses self-attention mechanism with quadratic time with respect to input size, making its scalability limited, (2) is *purely* based on message-passing graph neural networks, missing long-range dependencies and temporal resolution, and/or (3) encode brain activity in each unit of brain (e.g., voxel) separately, missing the dependencies of brain regions. In this study, we present BRAINMAMBA, an attention free, scalable, and powerful framework to learn brain activity multivariate timeseries. BRAINMAMBA uses two modules: (i) A novel multivariate timeseries encoder that leverage an MLP to fuse information across variates and an Selective Structured State Space (S4) architecture to encode each timeseries. (ii) A novel graph learning framework that leverage message-passing neural networks along with S4 architecture to selectively choose important brain regions. Our experiments on 7 real-world datasets with 3 modalities show that BRAINMAMBA attains outstanding performance and outperforms all baselines in different downstream tasks.

Data and Code Availability All the datasets used in this study are publicly available. The fMRI and MEG THINGS datasets are publicly available at this link. The HCP-Age and HCP-Mental datasets can be found in this link. The MPI-EEG dataset is publicly available at this link. The Temple University Hospital EEG Seizure Corpus (TUSZ) is publicly available at this link. The implementation of this work is available at this link.

Institutional Review Board (IRB) this study does not require IRB approval as all the used datasets were previously published and publicly available.

1. Introduction

Recent advancements in neuroimaging have significantly enriched our understanding of the human brain, offering detailed insights into its functioning and structure (Poldrack and Gorgolewski, 2014). Representation learning of brain activity based on the neuroimaging data is a key step toward analyzing the provided information, and unleashing deep learning techniques for use in understanding of cognitive process and the diagnosis of neurological diseases/disorders. The recent progress in deep learning techniques has led to powerful models for studying neuroimaging data, enabling the understanding of behaviors (Schneider et al., 2023), brain functions (Yamins and DiCarlo, 2016) and/or detecting neurological diseases (Uddin et al., 2017). These methods, however, use domain knowledge to design brain encoders that are suitable for a specific tasks, and so there is a lack of a universal model with the ability of being employed for different neuroimage modalities.

The main challenge towards such a universal architecture is the different data-modeling approaches that are required to capture temporal and spatial resolutions. For example, Electroencephalogram (EEG) is a non-invasive technique to measure electrical activity in the brain with a high temporal resolu-

tion (Subha et al., 2010). That is, EEGs have high sampling rate and can measure brain activity close to the timing of the actual activity. EEGs, however, have a poor spatial resolution, meaning they are not able to capture the exact location of the electrical activity (Subha et al., 2010). On the other hand, functional Magnetic Resonance Imaging (fMRI) has a significantly better spatial resolution while it has poor temporal resolution and responds to changes in the brain activity relatively slowly (Greve et al., 2013).

In the literature, neuroimage modalities with: ① high temporal resolution (e.g., EEG and MEG[1]) often are modeled as multivariate timeseries (Potter et al., 2022; Tang et al., 2023; Behrouz et al., 2023), focusing on high sampling rate and temporal aspect, and ② high spatial resolution (e.g., fMRI and structural MRI) often are modeled as graphs (Kan et al., 2022b; Li et al., 2021), focusing on spatiotemporal dependencies. A natural way to overcome this challenge is to use two encoders, each focuses on one aspect of data (Behrouz et al., 2023; Tang et al., 2023). However, the existing methods use Transformers-like architectures (Vaswani et al., 2017), which are based on self-attention mechanism (Bahdanau et al., 2015), and so require quadratic time and memory with respect to the input data. This complexity is a significant obstacle for high-dimensional neuroimaging data as different modalities either have ⓘ long-range timeseries (e.g., in EEG and MEG), or ⓘⓘ a large number of spatial units (e.g., voxels in fMRI). To overcome this, studies often use aggregation of local brain response to obtain higher-level Region of Interest (ROI) activity (Kan et al., 2022b; Yang et al., 2023; Li et al., 2021), choose highly active local units (Behrouz et al., 2023), or reduce the dimension of temporal data (Pan et al., 2022); all results in sub-optimal performance and missing information.

To overcome the above challenges, motivated by the recent success of state space models in language modeling and timeseries data (Gu and Dao, 2023; Zhang et al., 2023; Behrouz et al., 2024), we present BrainMamba. To learn the dynamics of brain activity and its temporal properties, BrainMamba uses Brain Timeseries Mamba (BTMamba). Recently, state space models show promising performance in challenging long sequence modeling tasks (Zhang et al., 2023) and classification of biosignals (Tang et al., 2023). However, they suffer from two main limitations when applying on neuroimaging data: ① They treat each variate of the multivariate timeseries separately while in neuroimaging data, the dependencies across different brain regions are important to understand brain activity patterns that might cause a brain disease/disorder (Behrouz et al., 2023) (See §5). ② Their recurrent process is input independent and so the learning process is time-invariant, meaning that they use the same parameters for all input during the recurrent scan. In task-dependent neuroimaging data, however, it is important to adjust the process based on the context of the input data. Accordingly, time-invariant process results in missing the context and so suboptimal performance (Gu and Dao, 2023). To address these challenges, BTMamba first uses a simple Multilayer Perceptron (MLP) to bind the information across variates, capturing the dependencies between the timeseries corresponds to different brain units. Then, it employs an input-dependent selective state space model to select informative timestamps during recurrent scans.

To learn from the underlying graph structure of brain activity, which help to better capture spatiotemporal dependencies, BrainMamba uses Brain Network Mamba (BNMamba), a novel graph learning framework that uses message-passing neural network (MPNN) to first learn the local dependencies of brain units (e.g., voxels or channels), then treats nodes as graph tokens, and finally uses an input-dependent selective state space model to select and aggregate informative brain regions in linear time. This input-dependent selection mechanism is specially important for brain networks as for different disease/disorder or different tasks (in task-based neuroimaging), diagnosis requires more attention to specific parts of the brain (Franzmeier et al., 2020; Chatterjee et al., 2021).

Finally, BrainMamba uses a learnable gate to combine the information obtained from the encoders. In pre-training setups, we adapt the training procedure proposed by Behrouz et al. (2023) and maximize the mutual information of the output of the encoders. The overview of BrainMamba is illustrated in Figure 1. Our extensive experiments on seven datasets, three different modalities, and on four downstream tasks show that not only BrainMamba achieves superior performance with respect to the state-of-the-art methods, but each of its encoders *alone* also outperforms their corresponding baselines. Notably, the superior performance is achieved while BrainMamba requires less memory and time compared to baselines.

1. Magnetoencephalography

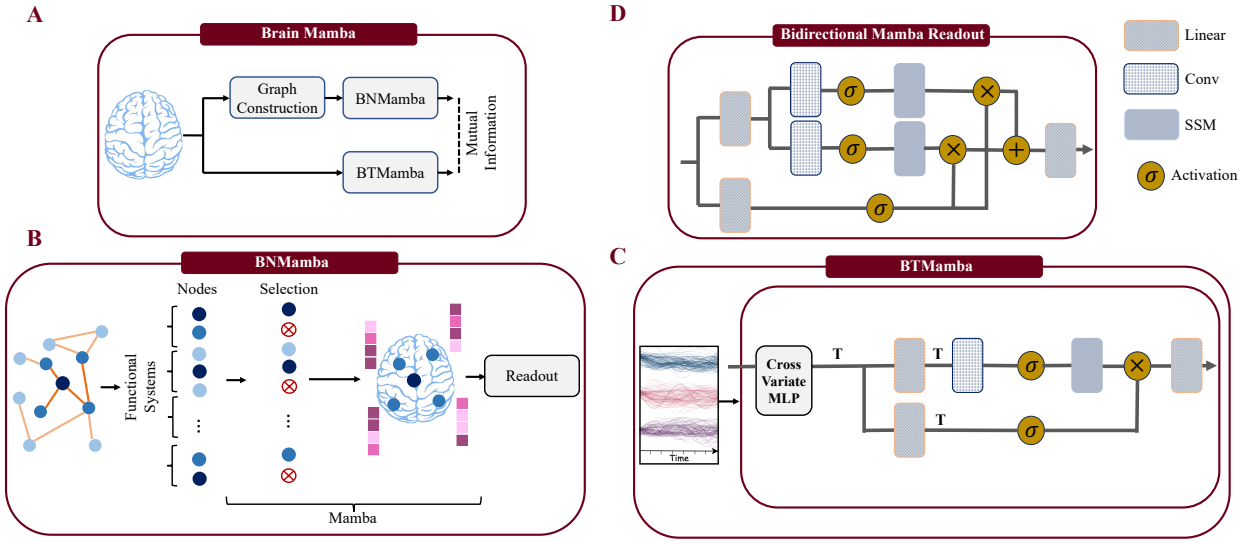

Figure 1: **Schematic of the BrainMamba.** (A) The overview of the BRAINMAMBA framework. (B) The architecture of BNMAMBA: It first ordered brain units based on their corresponding functional system, and then use Mamba to select informative brain units. (C) The architecture of BTMAMBA, where it first bind information cross variates by an MLP and then use Mamba to encode each variate of the time series. (D) The architecture of the bidirectional Mamba readout.

Contributions. Our main contributions are:

- We present BNMAMBA, a graph learning method that use special traits of brain networks to efficiently and effectively encode spatiotemporal brain networks. It leverages: ① MPNNs to encode local dependencies of brain units, ② a tokenizer and ordering mechanism to order nodes with respect to their functionality in the brain, ② a selective structured state space model to efficiently select informative and relevant brain regions, and ④ a simple adaptive readout block to learn the brain-level encoding.
- We propose BTMAMBA, a multivariate timeseries encoders for brain signals that leverages: ① an MLP block to bind temporal information across brain units, ② a time-dependent S4 block to selectively encode each variate, and ③ the same readout function as BNMAMBA to obtain brain-level encoding.
- We extensively evaluate the BRAINMAMBA and each of its encoders, BTMAMBA and BNMAMBA, on seven datasets with three different modalities, and on three different downstream tasks. The results show that not only BRAINMAMBA achieve superior performance compared to state-of-the-art models, but each of its encoders *alone* also outperform their corresponding baselines.

2. Related Work and Background

2.1. Multivariate Timeseries Learning

Attention mechanisms (Bahdanau et al., 2015) are powerful models to learn long-range dependencies in data. Accordingly, Transformer-based models have attracted much attention in time series forecasting (Zerveas et al., 2021; Li et al., 2019). Despite their power, their quadratic time complexity is a critical challenge when applying on large-scale datasets. Accordingly, several studies aim to reduce the time and memory usage of these methods by using sparse attentions (Wu et al., 2020; Zhou et al., 2021). Concurrently, to improve the efficiency of timeseries forecasting, inspired by the recent success of MLP-MIXER (Tolstikhin et al., 2021), Li et al. (2023) and Chen et al. (2023) presented two variants of MLP-MIXER for timeseries forecasting.

State Space Models. State Space Models (SSMs), a type of sequence models, are usually known as linear time-invariant systems that map input sequence $x(t) \in \mathbb{R}^L$ to response sequence $y(t) \in \mathbb{R}^L$ (Aoki, 2013). Specifically, SSMs use a latent state $h(t) \in \mathbb{R}^{N \times L}$, evolution parameter $\mathbf{A} \in \mathbb{R}^{N \times N}$, and projection parameters $\mathbf{B} \in \mathbb{R}^{N \times 1}, \mathbf{C} \in \mathbb{R}^{1 \times N}$ such that:

$$\begin{aligned} h'(t) &= \mathbf{A}\, h(t) + \mathbf{B}\, x(t), \\ y(t) &= \mathbf{C}\, h(t). \end{aligned} \tag{1}$$

Due to the hardness of solving the above differential equation in deep learning settings, discrete space state models (Gu et al., 2020; Zhang et al., 2023) discretize the above system using a parameter $\mathbf{\Delta}$:

$$h_t = \bar{\mathbf{A}}\, h_{t-1} + \bar{\mathbf{B}}\, x_t,$$
$$y_t = \bar{\mathbf{C}}\, h_t, \tag{2}$$

where

$$\bar{\mathbf{A}} = \exp\left(\mathbf{\Delta A}\right),$$
$$\bar{\mathbf{B}} = \left(\mathbf{\Delta A}\right)^{-1}\left(\exp\left(\mathbf{\Delta A} - I\right)\right) . \mathbf{\Delta B}. \tag{3}$$

Gu et al. (2020) shows that discrete-time SSMs are equivalent to the following convolution:

$$\bar{\mathbf{K}} = \left(\bar{\mathbf{C}}\bar{\mathbf{B}}, \bar{\mathbf{C}}\bar{\mathbf{A}}\bar{\mathbf{B}}, \ldots, \bar{\mathbf{C}}\bar{\mathbf{A}}^{L-1}\bar{\mathbf{B}}\right),$$
$$y = x * \bar{\mathbf{K}}, \tag{4}$$

and accordingly can be computed very efficiently. Discrete space state models show promising performance on timeseries data (Zhang et al., 2023; Tang et al., 2023), but they suffer from two main limitations: ① They lack selection mechanism, causing missing the context as discussed by Gu and Dao (2023) ② They treat each variate of the multivariate timeseries seperately, missing the inter-variate dependencies. Recently, Gu and Dao (2023) introduce an efficient and powerful architecture, called Mamba, with a novel selection mechanism for 1-d sequences (e.g., language and DNA) that potentially can address the first limitation. BTMamba can be seen as an adapted selective state space model proposed by Gu and Dao (2023) to multivariate timeseries data.

2.2. Graph and/or Timeseries Learning for Brain

With the success of graph neural networks in graph data analysis, they have become powerful and popular methods to analyze the brain functional connectivity (Behrouz and Seltzer, 2023a; Behrouz and Hashemi, 2023) and to diagnosis of neurological disease/disorder (Kan et al., 2021; Tang et al., 2023; Kan et al., 2022b). The first group of studies use the statistical correlation of signals to construct the underlying graph and then employ pure MPNN to learn the node (brain regions) encodings (Kan et al., 2021; Zhu et al., 2022; Li et al., 2021). The second group of studies try to learn the structure of the underlying graph from the brain activity (Behrouz and Hashemi, 2023; Tang et al., 2023). While this approach potentially provide a more flexible paradigm, it has more parameters and requires more data, which is costly in neuroimaging (Bassett and Sporns, 2017). Recently, attention-based and transformer-based models attracts attention (Behrouz et al., 2023; Kan et al., 2022b; Hu et al., 2023). While these methods show promising performance, their attention mechanism has quadratic time complexity, limiting their applicability to voxel-level activity of the whole brain.

On the other hand, several studies focus on purely brain signals (multivariate timeseries data) to detect neurological diseases (Pan et al., 2022; Tang et al., 2022; Shoeibi et al., 2021; Craik et al., 2019). For example, Cai et al. (2023) designed a self-supervised learning framework to detect seizures from EEG and SEEG data. However, all these methods are use domain knowledge and also suffer from a subset of the following limitations: they are designed for ① a particular task (e.g., brain classification), ② a particular neuroimaging modality (e.g., fMRI or EEG) or ③ supervised settings, and cannot capture ④ long-range dependencies in the timeseries data, and/or ⑤ inter-variate dependencies of timeseries. Our BTMamba can be used on any neuroimage modalities that provide multivariate timeseries recorded from multiple units across the brain. Further, its efficient linear-time selection mechanism helps it to select informative timestamps and so capture the long-range dependencies across time.

Table 1 summarizes the differences of BrainMamba with recent brain activity encoders. While there are several studies, for the sake of presentation, we discuss a sample of recent and state-of-the-art methods that are a good representative of all studies. One of the important properties of BrainMamba is its linear time complexity, making it a powerful backbone for the architecture of foundation models for use in encoding brain activity.

3. Problem Setup

We represent the multivariate timeseries of brain activity as $\mathcal{X} = \{\mathcal{X}^{(t)}\}_{t=1}^{T}$, where $\mathcal{X}^{(t)} \in \mathbb{R}^{|\mathcal{V}| \times \tilde{T}(t)}$ represents the neural data in time window $1 \leq t \leq T$, $\mathcal{V}$ is the set of brain units, and $\tilde{T}(t)$ is the length of the timeseries in time window t. In task-dependent data, each time window t corresponds to a task, and in resting state data, we have $T = 1$. We further let $t_{\max} = \max_{t=1,\ldots,T} \tilde{T}(t)$, representing the maximum length of timeseries. Using the multivariate timeseries data, in each time window t, we follow the literature (e.g., (Kan et al., 2022b; Behrouz

Table 1: The comparison of recent brain activity encoders.

	MVTS (Potter et al., 2022)	BNTransformer (Kan et al., 2022b)	GraphS4mer (Tang et al., 2023)	PTGB (Yang et al., 2023)	BrainMixer (Behrouz et al., 2023)	BrainMamba (This work)
Data	EEG (Timeseries)	fMRI (Graph)	EEG† (Timeseries + Graph)	fMRI (Graph)	All‡ (Timeseries + Graph)	All‡ (Timeseries + Graph)
Brain Unit	Channels	ROIs	Channels	ROIs	Highly active voxels (Channels in EEG)	All voxels (Channels in EEG)
Spatial Encoding	-	✓	✓	✓	✓	✓
Temporal Encoding	✓	-	✓	-	✓	✓
Inter-variate Information Fusing	-	-	-	-	✓	✓
Pre-training	✓	-	-	✓	✓	✓
Long-range Sequence	-	-	✓	-	-	✓
Time Complexity	Quadratic	Quadratic	Quadratic	Quadratic	Quadratic	**Linear**

† Although GraphS4mer can potentially be employed on neuroimage modalities with low sampling frequency (e.g. fMRI), it is best suited for timeseries with long-range temporal dependencies (e.g., EEG) as discussed by Tang et al. (2023).
‡ All fMRI, EEG, MEG, and generally any neuroimaging modalities that provide multivariate timeseries recorded from multiple units across the brain.

and Seltzer, 2023a)) and construct the brain network (functional connectivity graph) $\mathcal{G}_F^{(t)} = (\mathcal{V}, \mathcal{E}^{(t)})$ such that $\mathcal{E}^{(t)} \subseteq \mathcal{V} \times \mathcal{V}$ is the set connections between brain units. The connections are constructed based on the Pearson correlation of corresponding timeseries. That is, $(u, v) \in \mathcal{E}^{(t)}$ iff $\mathcal{X}_u^{(t)}$ and $\mathcal{X}_v^{(t)}$ have high correlation.

In this work, we aim to learn a low dimensional vector representation for (1) each brain unit and (2) the brain, which we later will use for binary (i.e., anomaly detection) and multi-class classification tasks.

4. Model: BrainMamba

BrainMamba consists of two main modules: ① BNMamba and ② BTMamba.

4.1. Brain Network Encoder

The main goal of this encoder is to encode spatio-temporal properties of the brain networks, obtained from the neuroimaging data. A natural choice to learn the node encodings of a graph is to use MPNNs. However, as discussed earlier, the relevance of the activity of each brain unit for each neurological disease or disorder is different and a simple MPNN will treat all the nodes the same. They further suffer from over-squashing (Di Giovanni et al., 2023), making them limited in capturing long-range dependencies. Alternatively, Graph Transformers (Yun et al., 2019) are powerful architectures that can learn both long-range dependencies as well as the importance of brain units via its attention mechanism. They, however, have quadratic time complexity, limiting their applicability to large-scale neuroimage datasets.

Inspired by the success of Mamba architecture in 1-d sequential data (Gu and Dao, 2023), Brain Network Mamba uses discrete state space models with a selection mechanism. To adapt Mamba in this context, there are two critical challenges: ① Mamba is a sequential encoder and cannot simply be applied on complex structures like graphs. ② Simply treating a graph as a sequence of nodes requires a specific ordering of nodes and also will miss the local dependencies. To address the above challenges, BNMamba uses MPNN, learning local dependencies, simultaneously with a new selective graph SSM architecture, learning long-range dependencies.

Tokenization. Inspired by Graph Transformer (Yun et al., 2019; Behrouz and Hashemi, 2024), we tokenize the brain network as a sequence of nodes each of which is associated with a positional encoding and an initial feature vector. To encode the position of each node, we use a special traits of the brain. The human brain is comprised of functional systems (Schaefer et al., 2018), which are groups of brain units (e.g., voxels) that perform similar functions (Smith et al., 2013). To capture the position of a brain unit, we consider its distance to all other brain units within its corresponding functional systems. The main challenge in graph-structured data is the lack of orders for nodes. In BNMamba, we suggest functional ordering, which is sorting nodes with respect to the size of boundary edges[2] in their corresponding functional system . The intuition of this approach comes from the recurrent architecture of the SSMs. In SSMs, including Mamba (Gu and Dao, 2023), the hidden states are updated based on prior elements in the se-

2. i.e., edges between inside and outside.

quence. Accordingly, earlier tokens has less information about the other tokens in the sequence. In the proposed functional ordering earlier tokens has less connections to other tokens, meaning that they have less functional correlation with other brain units and so are less dependent to others. On the other hand, later tokens are brain units with large number of connections, meaning that they are more dependent to the encodings of others.

While the functional ordering uses a special trait of the brain to group and order brain units, the order of brain units in a functional system is still unknown. To this end, during the training phase, we randomly shuffle the order of brain units within a functional system to make the model robust to permutation.

Selective SSM Module. In the previous part, we tokenize the brain network into a sequence of nodes. Next, we discuss the selective SSM module. Given $u \in \mathcal{V}$ and $1 \leq t \leq T$, let $\boldsymbol{\phi}_u^{(t)}$ represents the initial feature vector of u, obtained from the concatenation of the node positional encoding and its corresponding brain activity time series $\mathcal{X}_u^{(t)}$. We define $\boldsymbol{\Phi}^{(t)}$ as the matrix whose rows are $\boldsymbol{\phi}_u^{(t)}$, ordered by the functional ordering discussed above. Given $1 \leq t \leq T$, we define our selective SSM as follows:

$$
\begin{aligned}
&\boldsymbol{\Phi}_{\text{input}}^{(t)} = \sigma\left(\texttt{Conv}\left(\mathbf{W}_{\text{input}}\, \texttt{LayerNorm}\left(\boldsymbol{\Phi}^{(t)}\right)\right)\right),\\
&\mathbf{B}^{(t)} = \mathbf{W}_{\mathbf{B}}\, \boldsymbol{\Phi}_{\text{input}}^{(t)}, \quad \mathbf{C}^{(t)} = \mathbf{W}_{\mathbf{C}}\, \boldsymbol{\Phi}_{\text{input}}^{(t)},\\
&\boldsymbol{\Delta}^{(t)} = \mathbf{W}_{\Delta}\, \boldsymbol{\Phi}_{\text{input}}^{(t)},\\
&\bar{\mathbf{A}}^{(t)} = \texttt{Discrete}_{\mathbf{A}}\left(\mathbf{A}^{(t)}, \boldsymbol{\Delta}\right),\\
&\bar{\mathbf{B}}^{(t)} = \texttt{Discrete}_{\mathbf{B}}\left(\mathbf{B}^{(t)}, \boldsymbol{\Delta}\right),\\
&\boldsymbol{y}^{(t)} = \texttt{SSM}_{\bar{\mathbf{A}},\bar{\mathbf{B}},\mathbf{C}}\left(\boldsymbol{\Phi}_{\text{input}}^{(t)}\right),\\
&\boldsymbol{y}_{\text{out}}^{(t)} = \mathbf{W}_{\text{out}}\left(\boldsymbol{y}^{(t)} \odot \sigma\left(\mathbf{W}\texttt{LayerNorm}\left(\boldsymbol{\Phi}^{(t)}\right)\right)\right), \quad (5)
\end{aligned}
$$

where $\mathbf{W}, \mathbf{W}_{\mathbf{B}}, \mathbf{W}_{\mathbf{C}}, \mathbf{W}_{\boldsymbol{\Delta}}$, and $\mathbf{W}_{\text{out}}$ are learnable parameters, $\sigma(.)$ is nonlinear function, $\texttt{LayerNorm}(.)$ is layer normalization (Ba et al., 2016), $\texttt{SSM}(.)$ is the state space model discussed in Equations 2 and 4, and $\texttt{Discrete}(.)$ is discretization process discussed in Equation 3. In the above formulation, all parameters of $\texttt{SSM}(.)$, i.e., $\mathbf{B}, \mathbf{C}$, and $\boldsymbol{\Delta}$ are functions of the input $\boldsymbol{\Phi}_{\text{input}}^{(t)}$ and so the recurrent process is time-variant and can select the relevant information based on the input as discussed by (Gu and Dao, 2023). As discussed earlier, this selective SSM is particularly important for capturing *long-range* dependencies in the neuroimaging data.

Message-Passing. To capture the local dependencies in the brain network, we use a message-passing graph neural network model. Given $u \in \mathcal{V}$ and $1 \leq t \leq T$, let $\boldsymbol{\psi}_u^{(t)^{(0)}} = \boldsymbol{\phi}_u^{(t)}$ be the initial feature vector of u, as defined above. Then ℓ-th layer of message-passing neural network can be written as:

$$
\begin{aligned}
&m_{v\to u}^{(\ell)} = \mathbf{W}_{\text{local}}^{(\ell)} \textsc{Concat}\left(\psi_u^{(t)^{(\ell-1)}}, \psi_v^{(t)^{(\ell-1)}}\right)\\
&\boldsymbol{\psi}_u^{(t)^{(\ell)}} = \textsc{Sum}\left(\left\{m_{v\to u}^{(\ell)} | v \in \mathcal{N}^{(t)}(u)\right\}\right) + \boldsymbol{\psi}_u^{(t)^{(\ell-1)}} \quad (6)
\end{aligned}
$$

where $\mathbf{W}_{\text{local}}^{(\ell)}$ is a learnable matrix and $\mathcal{N}^{(t)}(u)$ is the set of all neighbors of u in timestamp t. i.e., $\mathcal{N}^{(t)}(u) = \{v | (u, v) \in \mathcal{E}^{(t)}\}$. We let $\boldsymbol{\Psi}^{(t)}$ be the output of the above message-passing procedure after the last layer whose rows are $\boldsymbol{\psi}_u^{(t)}$ for $u \in \mathcal{V}$. Next, to take the advantage of both local and long-range dependencies, we aggregate the output of the message-passing process with the selective SSM:

$$
\mathbf{H}^{(t)} = \texttt{Agg}\left(\boldsymbol{\Psi}^{(t)}, \boldsymbol{y}_{\text{out}}^{(t)}\right), \quad (7)
$$

where $\texttt{Agg}(.)$ is aggregation function (we use summation in our experiments). We use $\mathbf{h}_u^{(t)}$ to refer to the corresponding row of the $\mathbf{H}^{(t)}$ to node a $u \in \mathcal{V}$.

Readout Function (Mamba Readout). Given the encoding of brain units obtained from the above procedure, the main goal of the readout function is to learn the global encoding, i.e., low-dimensional representation of the brain. To this end, we aim to use a selective SSM on the sequence of nodes and treat the last output as the graph-level representation. Accordingly, the selection mechanism filters irrelevant brain units to the brain-level downstream task, providing a high quality encoding. Contrary to Equation 7, in which we aim to learn node encodings, we do not need the output for middle states and we only look at the final output of the recurrent scan. Accordingly, the procedure is more sensitive to the choice of ordering and the proposed functional ordering is not useful. To this end, inspired by Wang et al. (2023), we use a bidirectional procedure. That is, given an ordered sequence of nodes obtained from the functional ordering, we use two selective state space models, each with the same architecture as Equation 7, and feed the ordered sequence forward and backward. This procedure helps decreasing the sensitivity to the ordering of nodes. We use $\boldsymbol{\eta}_{\text{BN}}^{(t)}$ to refer to the brain network level encoding.

4.2. Multivariate Brain Signal Encoder

The main goal of this encoder is to learn the temporal properties and long-range dependencies in the multivariate timeseries of brain activity. As discussed earlier, while state space models are powerful methods for efficiently modeling timeseries data, they suffer from two limitations: ① they cannot bind information across variates in multivariate timeseries data and ② they are time-invariant and use the same mechanism for all input tokens. In this section, we present a new architecture that can address the above limitations.

Inter-variate Information Fusing. In multivariate timeseries data, there are dependencies across both time and variate dimensions. In the first module, given $1 \leq t \leq T$, to bind and fuse information across variates, we suggest using a simple MLP on the transpose of timeseries matrix $\mathcal{X}^{(t)}$:

$$\boldsymbol{Z}^{(t)} = \mathbf{W}_{\text{Time}_1} \sigma \left(\mathbf{W}_{\text{Time}_2} \texttt{LayerNorm} \left(\mathcal{X}^{(t)^\top} \right) \right)^\top, \quad (8)$$

where $\mathbf{W}_{\text{Time}_1}$ and $\mathbf{W}_{\text{Time}_2}$ are learnable parameters. This module mixes the information across rows of matrix $\mathcal{X}^{(t)}$, capturing the variates wise dependencies.

Variate Encoder. Next, given a brain unit $u \in \mathcal{V}$, to encode its corresponding timeseries data, we use the same architecture as Equation 7 with input $\boldsymbol{Z}_u^{(t)}$. We use $\boldsymbol{\Upsilon}^{(t)}$ to denote the output of this step, where $\boldsymbol{\Upsilon}_u^{(t)}$ is the encoding of the timeseries associated with a brain unit $u \in \mathcal{V}$.

Readout Function. In graph-level downstream tasks, we use the same readout function as BN-Mamba, which is discussed earlier, to obtain the global encoding. We use $\boldsymbol{\eta}_{\text{BT}}$ to refer to this global encoding.

4.3. Dynamics Across Time Windows

In resting state neuroimage data, we have $T = 1$, i.e., there is only one time window. In task-based neuroimage data, however, we have different time windows and the dynamics of brain activity across all time windows can be a key to diagnosis a disease/disorder (Fiorenzato et al., 2019). In the above procedures, we calculate the encodings based on only the current time window, missing the dynamics across all time windows. We use a similar approach as GRU mechanism (Chung et al., 2014) but with state space models. Given $u \in \mathcal{V}$, to update the brain unit encodings in brain network over time windows, we treat $\mathbf{h}_u^{(1)}, \ldots, \mathbf{h}_u^{(T)}$ as corresponding tokens of node u and use a selective SSM, the same architecture as Equation 7, to update the node encodings over time windows in a recurrent manner. We use the same procedure on $\boldsymbol{\Upsilon}_u^{(1)}, \ldots, \boldsymbol{\Upsilon}_u^{(T)}$ to update timeseries encodings over time. We use $\hat{\mathbf{H}}$, a matrix whose rows corresponds to nodes' encodings, to denote the final output of the brain network encoding process and $\hat{\boldsymbol{\Upsilon}}$ to refer to the final output of the timeseries encoding process.

To take advantage of both temporal and spatio-temporal encodings, respectively obtained from $\hat{\boldsymbol{\Upsilon}}$ and $\hat{\mathbf{H}}$, we concatenate them and use an MLP to calculate the final encodings:

$$\boldsymbol{\Omega} = \texttt{MLP} \left(\hat{\mathbf{H}} \| \hat{\boldsymbol{\Upsilon}} \right), \quad (9)$$

where $\|$ is row-wise concatenation.

4.4. Training

Supervised and Semi-Supervised Settings. In semi-supervised settings, given a pre-trained or randomly initialize model, we end-to-end train our model using provided labels or contrastive learning, discussed by Behrouz et al. (2023). When the initial model is pre-trained, we freeze all modules except Equation 9, letting the MLP(.) learn how to incorporate the temporal and spatia-temporal information.

Pre-training Setup. In the pre-training setup, since our model is capable of obtaining brain activity encodings from two different views (i.e., temporal and spatio-temporal views), we use the framework proposed by Behrouz et al. (2023), and maximize the mutual information between $\hat{\mathbf{H}}$ and $\hat{\boldsymbol{\Upsilon}}$ using Noise-Contrastive Estimation (NCE) (Gutmann and Hyvärinen, 2010) and minimize the following loss function:

$$\begin{aligned} \texttt{Loss} = & \mathbb{E}_{(\hat{\boldsymbol{\Upsilon}}, \hat{\mathbf{H}}_{v,i})} \left[\mathbb{E}_{\mathcal{N}} \left[\mathcal{L}(\hat{\boldsymbol{\Upsilon}}, \hat{\mathbf{H}}_{v,i}, \mathcal{N}) \right] \right] \\ & + \mathbb{E}_{(\hat{\mathbf{H}}, \hat{\boldsymbol{\Upsilon}}_{v,j})} \left[\mathbb{E}_{\mathcal{N}} \left[\mathcal{L}(\hat{\mathbf{H}}, \hat{\boldsymbol{\Upsilon}}_{v,j}, \mathcal{N}) \right] \right], \end{aligned} \quad (10)$$

where $\mathcal{N}$ is the set of negative samples, $(\hat{\boldsymbol{\Upsilon}}, \hat{\mathbf{H}}_{v,i}^{(t)})$ and $\hat{\mathbf{H}}, (\hat{\boldsymbol{\Upsilon}})_{v,j})$ are the positive sample pairs, and $\mathcal{L}$ is a standard Log-Softmax function.

5. Experiments and Results

Datasets. We use seven real datasets with three different modalities (i.e., fMRI, MEG, EEG): ①

Bvfc (Behrouz et al., 2023) is a task-based fMRI dataset that includes voxel activity timeseries and functional connectivity of 3 subjects when looking at the 8460 images from 720 categories. ② Bvfc-MEG is the MEG counterpart of Bvfc. ③ ADHD (Milham et al., 2011) consists of fMRI of 250 subjects in the ADHD group and 450 subjects in the typically developed (TD) control group. ④ The Seizure detection TUH-EEG dataset (Shah et al., 2018) contains EEG data (with 31 channels) of 642 subjects. ⑤ HCP-mental (Van Essen et al., 2013) contains data from 7440 neuroimaging samples each of which is associated with one of the seven ground-truth mental states. ⑤ HCP-age (Van Essen et al., 2013) contains the same data but we aim to predict the age of human subjects using their fMRI. ⑦ MPI-EEG (Babayan et al., 2019) consists of 204 EEG data (two healthy groups (1) young with age $25.1_{\pm 3.1}$ and (2) elderly with age 67.6 ± 4.7) with 62 chennels. The details of the dataset can be found in Appendix A

Evaluation Tasks. We focus on three tasks: ① Brain Classification (multi-class classification) ② Brain Unit Anomaly Detection (AD), and ③ Brain AD (binary classification). For the brain unit AD tasks, we follow previous studies (Ma et al., 2021; Behrouz and Seltzer, 2023a; Behrouz and Hashemi, 2023), and corrupt 1% and 5% of the data. For brain AD, the ground truth anomalies in Bvfc are the brain responses to not recognizable images and for ADHD and TUH-EEG datasets are brain activity of people living with ADHD and seizure, respectively. For brain classification, we focus on the prediction of ⓘⓘ age prediction and mental state decoding (in HCP-Age, and HCP-Mental), and ⓘ categories of images seen by the subjects (in Bvfc, and Bvfc-MEG). We perform statistical comparison with baselines via paired t-tests and shade significance results (p-value ≤ 0.05) with blue and others with gray.

In binary classification tasks, due to the potential of imbalanced data, we follow the literature (Ma et al., 2021; Tang et al., 2023; Behrouz et al., 2023) and report Area Under Precision-Recall Curve (AUC-PR). For multi-class classification, we report the accuracy of the methods.

Baselines. We compare BrainMamba with state-of-the-art time series, graph, and brain activity encoder models: ① Graph-based methods: GOutlier (Aggarwal et al., 2011), NetWalk (Yu et al., 2018), Graph MLP-Mixer (GMM) (He et al., 2023), GraphMixer (Cong et al., 2023). ② brain-network-based methods: BrainGnn (Li et al., 2021), FbNetGen (Kan et al., 2022a), BrainNetCnn (Kawahara et al., 2017), ADMire (Behrouz and Seltzer, 2023b), BNTransformer (Kan et al., 2022b), PTGB (Yang et al., 2023), and BrainMixer (Behrouz et al., 2023). ③ Time-series-based methods: Usad (Audibert et al., 2020), Time Series Transformer (TST) (Zerveas et al., 2021), Mvts (Potter et al., 2022), and GraphS4mer (Tang et al., 2023). For the sake of fair comparison, we use the same training, hyperparameter tuning, and testing procedure as BrainMamba.

Table 2: Performance on multi-class brain classification: Mean ACC (%) ± std. We shade significance results with blue and others with gray.

Methods	Bvfc	Bvfc-MEG	HCP-Mental	HCP-Age
Usad	$48.52_{\pm 1.94}$	$50.02_{\pm 1.13}$	$73.49_{\pm 1.56}$	$39.17_{\pm 1.68}$
HyperSAGCN	$51.92_{\pm 1.47}$	$51.19_{\pm 1.88}$	$90.37_{\pm 1.61}$	$47.38_{\pm 1.96}$
GMM	$53.11_{\pm 1.44}$	$53.04_{\pm 1.73}$	$90.92_{\pm 1.83}$	$47.75_{\pm 1.26}$
GraphMixer	$53.17_{\pm 1.21}$	$53.12_{\pm 1.18}$	$91.13_{\pm 1.44}$	$48.32_{\pm 1.11}$
BrainNetCnn	$49.10_{\pm 1.83}$	$50.12_{\pm 1.57}$	$83.58_{\pm 1.68}$	$42.26_{\pm 2.03}$
BrainGNN	$50.63_{\pm 1.67}$	$51.08_{\pm 0.96}$	$85.25_{\pm 2.17}$	$43.08_{\pm 1.54}$
FbNetGen	$50.18_{\pm 0.98}$	$50.94_{\pm 1.39}$	$84.47_{\pm 1.88}$	$42.83_{\pm 1.78}$
Admire	$54.36_{\pm 1.39}$	$54.87_{\pm 1.92}$	$89.74_{\pm 1.93}$	$47.82_{\pm 1.72}$
PTGB	$55.89_{\pm 1.78}$	$55.11_{\pm 1.62}$	$92.58_{\pm 1.31}$	$48.41_{\pm 1.47}$
BNTransformer	OOM†	$55.17_{\pm 1.74}$	$91.71_{\pm 1.48}$	$47.94_{\pm 1.15}$
GraphS4mer	OOM†	$74.39_{\pm 0.92}$	$82.30_{\pm 1.83}$	$46.32_{\pm 1.09}$
BrainMixer	OOM†	$62.58_{\pm 1.12}$	$96.32_{\pm 0.29}$	$57.83_{\pm 1.03}$
BrainMamba	**$75.19_{\pm 1.98}$**	**$78.03_{\pm 1.69}$**	**$96.57_{\pm 1.05}$**	**$59.62_{\pm 1.71}$**

† OOM: Out of Memory.

Brain Classification. Table 2 reports the performance of BrainMamba and baselines on multi-class brain classification task. BrainMamba outperforms all the baselines with three significant improvement our of the four datasets. By average BrainMamba achieve 9.94% performance improvement over the best baselines. The best improvement (24.68%) comes from the Bvfc-MEG data with high sampling rate, resulting in long-range brain signals. The reason for this superior performance is five fold: ① Compared to three state-of-the-art methods, BrainMamba is the most efficient methods and can scale to large-scale datasets (e.g., Bvfc). ② Compared to GraphS4mer, BrainMamba has the capability of learning from data with both high (e.g., MEG) and low (e.g., fMRI) temporal resolution. ③ Compared to PTGB and BrainMixer, which are capable of pre-training, BrainMamba shows superior performance, supporting the significance of the architecture design. For example, PTGB only uses MPNN networks which are important in capturing local dependencies, but miss the long-range dependencies. On the other hand, BrainMixer misses local dependencies due to the lack of message-passing mech-

Table 3: Performance on anomaly detection: Mean AUC-PR (%) ± standard deviation. OOM: Out of Memory. We shade significance results (corrected p-value ≤ 0.05) with blue and others with gray.

	Methods	Bvfc	Bvfc-MEG	HCP		ADHD		TUH-EEG		MPI-EEG	
	Anomaly %			1%	5 %	1%	5 %	1%	5 %	1%	5 %
Brain Unit-level AD	Usad	68.27$_{\pm 1.16}$	62.73$_{\pm 1.27}$	65.49$_{\pm 1.31}$	65.01$_{\pm 1.18}$	72.79$_{\pm 1.48}$	72.19$_{\pm 0.94}$	72.81$_{\pm 1.42}$	71.36$_{\pm 1.03}$	67.75$_{\pm 1.02}$	67.59$_{\pm 1.53}$
	TST	70.62$_{\pm 1.48}$	68.57$_{\pm 1.81}$	69.18$_{\pm 1.64}$	69.11$_{\pm 1.32}$	74.81$_{\pm 1.14}$	73.99$_{\pm 1.47}$	73.71$_{\pm 1.55}$	73.03$_{\pm 1.47}$	68.37$_{\pm 1.59}$	67.81$_{\pm 1.94}$
	Mvts	N/A	N/A	N/A	N/A	N/A	N/A	77.48$_{\pm 1.81}$	77.02$_{\pm 1.29}$	74.16$_{\pm 1.20}$	73.59$_{\pm 1.68}$
	GOutlier	64.66$_{\pm 2.38}$	60.17$_{\pm 1.25}$	63.59$_{\pm 1.62}$	63.07$_{\pm 1.52}$	68.97$_{\pm 1.16}$	67.12$_{\pm 0.93}$	65.18$_{\pm 1.09}$	65.01$_{\pm 1.57}$	61.89$_{\pm 2.01}$	60.88$_{\pm 2.23}$
	NetWalk	68.73$_{\pm 1.16}$	63.61$_{\pm 1.31}$	66.98$_{\pm 1.44}$	66.04$_{\pm 1.63}$	75.16$_{\pm 1.23}$	74.73$_{\pm 1.01}$	72.21$_{\pm 0.91}$	71.62$_{\pm 1.46}$	72.18$_{\pm 1.15}$	71.77$_{\pm 1.49}$
	GraphMixer	76.94$_{\pm 1.68}$	71.44$_{\pm 1.39}$	81.55$_{\pm 1.82}$	81.07$_{\pm 1.27}$	81.37$_{\pm 1.09}$	80.83$_{\pm 1.16}$	72.95$_{\pm 1.26}$	72.01$_{\pm 0.82}$	78.95$_{\pm 1.33}$	78.51$_{\pm 1.84}$
	BrainNetCnn	80.17$_{\pm 1.49}$	73.91$_{\pm 1.54}$	82.75$_{\pm 1.27}$	82.21$_{\pm 1.73}$	82.79$_{\pm 1.08}$	81.12$_{\pm 1.16}$	73.98$_{\pm 1.24}$	73.01$_{\pm 1.08}$	N/A	N/A
	BrainGnn	79.92$_{\pm 1.63}$	73.25$_{\pm 1.94}$	82.99$_{\pm 1.65}$	82.13$_{\pm 1.66}$	81.14$_{\pm 1.05}$	80.83$_{\pm 0.87}$	73.06$_{\pm 1.14}$	72.74$_{\pm 0.86}$	N/A	N/A
	FbNetGen	79.17$_{\pm 2.04}$	72.35$_{\pm 1.84}$	82.26$_{\pm 1.37}$	81.62$_{\pm 1.49}$	80.91$_{\pm 1.12}$	80.94$_{\pm 1.74}$	72.53$_{\pm 1.48}$	72.06$_{\pm 1.29}$	N/A	N/A
	PTGB	85.18$_{\pm 1.83}$	76.16$_{\pm 1.08}$	85.72$_{\pm 1.14}$	84.95$_{\pm 1.33}$	86.43$_{\pm 1.16}$	86.36$_{\pm 1.15}$	77.54$_{\pm 1.37}$	77.32$_{\pm 1.21}$	N/A	N/A
	BNTransformer	OOM	75.67$_{\pm 1.14}$	85.02$_{\pm 0.96}$	84.36$_{\pm 1.59}$	86.13$_{\pm 1.21}$	86.11$_{\pm 1.82}$	77.96$_{\pm 1.32}$	77.08$_{\pm 1.06}$	N/A	N/A
	GraphS4mer	OOM	81.09$_{\pm 0.57}$	84.19$_{\pm 0.85}$	83.99$_{\pm 1.41}$	82.95$_{\pm 1.13}$	83.06$_{\pm 1.49}$	78.33$_{\pm 1.26}$	79.01$_{\pm 1.08}$	84.19$_{\pm 1.00}$	83.98$_{\pm 1.72}$
	BrainMixer	OOM	81.52$_{\pm 1.32}$	89.27$_{\pm 1.61}$	88.94$_{\pm 1.24}$	89.97$_{\pm 1.14}$	89.81$_{\pm 1.27}$	79.45$_{\pm 1.19}$	79.23$_{\pm 0.94}$	84.07$_{\pm 1.13}$	84.14$_{\pm 1.26}$
	BrainMamba	**91.58**$_{\pm 1.24}$	**82.07**$_{\pm 1.10}$	**91.04**$_{\pm 0.89}$	**90.97**$_{\pm 1.33}$	**91.26**$_{\pm 1.00}$	**91.03**$_{\pm 1.64}$	**80.22**$_{\pm 1.02}$	**80.18**$_{\pm 1.57}$	**85.99**$_{\pm 0.63}$	**85.46**$_{\pm 1.05}$
Brain-level AD	Usad	71.93$_{\pm 1.15}$	61.32$_{\pm 1.71}$	67.79$_{\pm 2.28}$	67.36$_{\pm 2.61}$	82.87$_{\pm 2.03}$	80.52$_{\pm 1.84}$	72.03$_{\pm 1.17}$	71.48$_{\pm 1.05}$	75.19$_{\pm 1.52}$	73.60$_{\pm 0.91}$
	TST	72.47$_{\pm 1.23}$	67.12$_{\pm 2.07}$	67.94$_{\pm 1.69}$	67.22$_{\pm 1.17}$	83.54$_{\pm 1.38}$	83.04$_{\pm 1.12}$	72.96$_{\pm 1.39}$	72.11$_{\pm 1.58}$	74.26$_{\pm 1.33}$	73.95$_{\pm 1.98}$
	Mvts	N/A	N/A	N/A	N/A	N/A	N/A	83.53$_{\pm 1.91}$	82.41$_{\pm 1.02}$	86.44$_{\pm 1.78}$	83.92$_{\pm 1.63}$
	NetWalk	72.16$_{\pm 1.44}$	69.57$_{\pm 1.73}$	69.14$_{\pm 1.49}$	68.66$_{\pm 1.52}$	83.11$_{\pm 1.02}$	82.81$_{\pm 1.61}$	71.06$_{\pm 1.05}$	69.94$_{\pm 1.12}$	75.19$_{\pm 1.41}$	74.60$_{\pm 1.26}$
	GMM	81.79$_{\pm 1.24}$	77.84$_{\pm 1.52}$	74.87$_{\pm 1.58}$	74.02$_{\pm 1.10}$	85.89$_{\pm 0.98}$	85.03$_{\pm 1.18}$	76.62$_{\pm 1.17}$	76.11$_{\pm 1.26}$	77.73$_{\pm 1.82}$	77.35$_{\pm 0.99}$
	GraphMixer	82.56$_{\pm 1.19}$	77.91$_{\pm 1.26}$	75.03$_{\pm 1.72}$	74.46$_{\pm 1.53}$	86.02$_{\pm 1.15}$	85.64$_{\pm 1.09}$	77.49$_{\pm 1.09}$	76.63$_{\pm 1.22}$	76.52$_{\pm 1.07}$	75.93$_{\pm 1.06}$
	BrainNetCnn	78.47$_{\pm 1.18}$	73.12$_{\pm 1.27}$	70.73$_{\pm 1.77}$	70.12$_{\pm 1.86}$	85.84$_{\pm 0.96}$	85.07$_{\pm 1.52}$	73.92$_{\pm 0.97}$	73.07$_{\pm 1.51}$	N/A	N/A
	BrainGnn	79.81$_{\pm 1.57}$	75.28$_{\pm 1.61}$	72.98$_{\pm 1.55}$	72.41$_{\pm 1.16}$	84.59$_{\pm 1.26}$	83.72$_{\pm 1.35}$	72.41$_{\pm 1.38}$	71.55$_{\pm 1.16}$	N/A	N/A
	FbNetGen	78.94$_{\pm 1.24}$	74.49$_{\pm 1.33}$	71.62$_{\pm 1.53}$	71.06$_{\pm 1.48}$	84.67$_{\pm 1.26}$	84.08$_{\pm 1.37}$	72.69$_{\pm 1.18}$	71.87$_{\pm 1.12}$	N/A	N/A
	ADMire	83.72$_{\pm 1.18}$	78.83$_{\pm 1.56}$	75.52$_{\pm 1.81}$	74.59$_{\pm 1.12}$	86.27$_{\pm 1.72}$	85.18$_{\pm 1.56}$	78.12$_{\pm 1.47}$	77.59$_{\pm 1.68}$	N/A	N/A
	PTGB	84.08$_{\pm 1.35}$	79.68$_{\pm 1.62}$	76.01$_{\pm 1.07}$	75.13$_{\pm 1.48}$	87.59$_{\pm 1.12}$	86.99$_{\pm 0.96}$	79.17$_{\pm 1.36}$	78.64$_{\pm 1.55}$	N/A	N/A
	BNTransformer	OOM	79.03$_{\pm 1.78}$	75.64$_{\pm 1.82}$	75.09$_{\pm 1.18}$	87.54$_{\pm 1.04}$	86.92$_{\pm 1.48}$	79.36$_{\pm 1.71}$	78.08$_{\pm 1.16}$	N/A	N/A
	GraphS4mer	OOM	86.15$_{\pm 0.42}$	72.88$_{\pm 0.94}$	71.77$_{\pm 1.05}$	84.33$_{\pm 0.79}$	83.95$_{\pm 0.88}$	89.95$_{\pm 0.34}$	88.69$_{\pm 0.73}$	89.44$_{\pm 1.17}$	89.25$_{\pm 1.68}$
	BrainMixer	OOM	84.59$_{\pm 1.70}$	80.67$_{\pm 1.13}$	80.49$_{\pm 1.07}$	91.38$_{\pm 0.94}$	90.98$_{\pm 1.02}$	85.74$_{\pm 1.16}$	85.63$_{\pm 1.23}$	90.82$_{\pm 1.51}$	89.74$_{\pm 1.89}$
	BrainMamba	**88.49**$_{\pm 1.01}$	**87.23**$_{\pm 0.92}$	**81.26**$_{\pm 1.05}$	**81.57**$_{\pm 1.80}$	**91.41**$_{\pm 0.46}$	**91.95**$_{\pm 1.02}$	**92.17**$_{\pm 1.31}$	**91.76**$_{\pm 0.99}$	**92.28**$_{\pm 0.76}$	**91.07**$_{\pm 1.24}$

anism. The architecture of BrainMamba, however, is capable of learning both local and long-range dependencies. ④ Compared to static methods (e.g., BrainGNN, BrainNetCNN, etc.), BrainMamba can take advantage of dynamics of the brain activity over time. ⑤ Finally, compared to timeseries encoder, BrainMamba takes advantage of underlying graph structured data of brain signals, learning spatio-temporal properties.

Anomaly Detection. We further evaluate the performance of BrainMamba in anomaly detection tasks. Table 3 reports the performance of BrainMamba and baselines on anomaly detection tasks at different scales: i.e., brain unit-level and brain-level. BrainMamba achieves the best AUC-PR on all datasets with 1.43% and 1.75% average improvement over the best baseline in brain unit-level AD, and brain-level AD, respectively. The main reasons for this superior performance are the same as the reasons we discuss above.

Ablation Study. We conduct ablation studies on our model using the Bvfc, Bvfc-MEG, HCP, and ADHD datasets to validate the effectiveness of BrainMamba's critical components and their contributions in its performance. Table 4 shows AUC-PR for Brain-Unit AD and accuracy (ACC) for brain multi-class classification tasks. The first row reports the performance of the complete BrainMamba implementation with pre-training. Each subsequent row shows results for BrainMamba with one module modification: row 2 removes the pre-training phase, row 3 is the brain network encoder alone (BrainMamba without timeseries encoder), row 4 removes message-passing from the BNMamba, row 5 replaces functional ordering with random ordering in the training phase, row 6 is the brain timeseries encoder alone (BrainMamba without brain network encoder), row 7 removes cross-variate MLP, and row 8 replace the bidirectional readout function with a simple unidirectional readout (using state space model discussed in Eq.7). The results show that each component and choice of architecture design is critical for achieving BrainMamba's superior performance. The greatest contribution comes from BNMamba, message-passing mechanism, BTMamba, cross-variate MLP, functional ordering, bidirectional readout, respectively. Note that the readout function is only used in brain classification tasks and its modification cannot affect the performance in unit AD.

Also, comparing the performance of the BNMamba with graph learning methods and BTMamba with time series encoders in Tables 2 and 3 shows that

Table 4: Ablation study on BrainMamba. AUC-PR scores for brain AD and ACC for classification.

Methods	Bvfc		Bvfc-MEG		HCP		ADHD	
	Unit-level AD	Classification	Unit-level AD	Classification	Unit-level AD	Classification	Unit-level AD	Classification
BrainMamba	**91.58**$_{\pm 1.24}$	**75.19**$_{\pm 1.98}$	**82.07**$_{\pm 1.10}$	**78.03**$_{\pm 1.69}$	**91.04**$_{\pm 0.89}$	**96.57**$_{\pm 1.05}$	**91.26**$_{\pm 1.09}$	-
w/o Pre-training	86.15$_{\pm 0.92}$	68.55$_{\pm 1.93}$	79.84$_{\pm 1.22}$	77.00$_{\pm 0.81}$	88.59$_{\pm 1.08}$	93.19$_{\pm 1.35}$	89.37$_{\pm 1.44}$	-
BNMamba	**86.19**$_{\pm 1.03}$	**71.02**$_{\pm 0.98}$	**77.14**$_{\pm 1.19}$	**67.41**$_{\pm 1.33}$	**88.65**$_{\pm 1.28}$	**94.19**$_{\pm 0.94}$	**86.98**$_{\pm 1.53}$	-
w/o MPNN	84.09$_{\pm 1.25}$	70.83$_{\pm 0.24}$	74.11$_{\pm 1.60}$	66.02$_{\pm 1.39}$	85.99$_{\pm 1.19}$	92.68$_{\pm 0.88}$	85.26$_{\pm 1.20}$	-
w/o Functional Ordering	85.97$_{\pm 0.98}$	70.88$_{\pm 1.06}$	76.85$_{\pm 1.00}$	65.49$_{\pm 1.23}$	87.12$_{\pm 1.67}$	93.74$_{\pm 0.80}$	86.19$_{\pm 0.61}$	-
BTMamba	**89.44**$_{\pm 1.09}$	**73.29**$_{\pm 1.15}$	**80.37**$_{\pm 0.92}$	**71.38**$_{\pm 0.89}$	**90.57**$_{\pm 1.42}$	**94.10**$_{\pm 1.46}$	**89.74**$_{\pm 0.71}$	-
w/o Cross-variate MLP	85.98$_{\pm 1.12}$	71.16$_{\pm 1.39}$	79.22$_{\pm 1.66}$	67.82$_{\pm 1.73}$	89.52$_{\pm 1.68}$	92.85$_{\pm 0.32}$	88.54$_{\pm 1.16}$	-
Unidirectional Readout	91.58$_{\pm 1.24}$	74.88$_{\pm 1.00}$	82.07$_{\pm 1.10}$	77.14$_{\pm 1.14}$	91.04$_{\pm 0.89}$	94.91$_{\pm 0.97}$	91.23$_{\pm 1.09}$	-

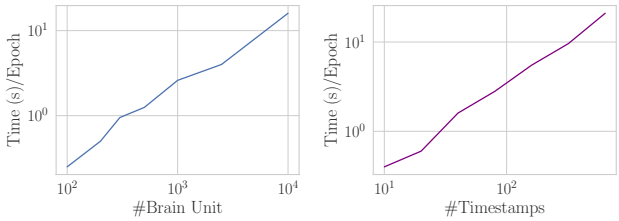

Figure 2: The effect of the number of brain units and the number of timestamps on the training time.

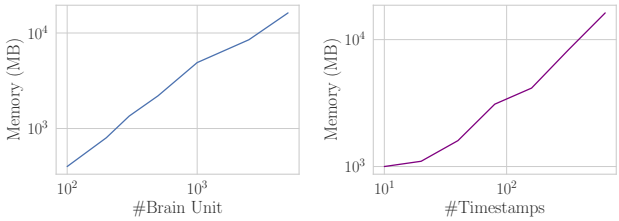

Figure 3: The effect of the number of brain units and the number of timestamps on the memory usage.

each of these two components alone can outperform their corresponding baselines.

Efficiency. We further evaluate the scalability and memory usage of BrainMamba with respect to the input size (either high spatial or temporal resolution). To this end, we change the number of voxels in BVFC and the number of samples from the timeseries in BVFC-MEG. Figures 2 and 3 report the results. BrainMamba time and memory scales linearly with respect to both the number of brain units and the number of samples in the time series. This scalability allow us to use BrainMamba on large datasets, making BrainMamba a potentially powerful backbone for the foundation models on neurosignals.

The Effect of Selection. In this part, we use corrupted data with different percent of corruption and report the performance of BrainMamba with and without selection (i.e., we use time-invariant state space model). Results are reported in Figure 4. BrainMamba with selective state space module is more robust to the corrupted data than BrainMamba without selection (i.e., time-invariant mechanism). These results show the importance of the architecture design of BrainMamba, specifically for noisy neuroimage data.

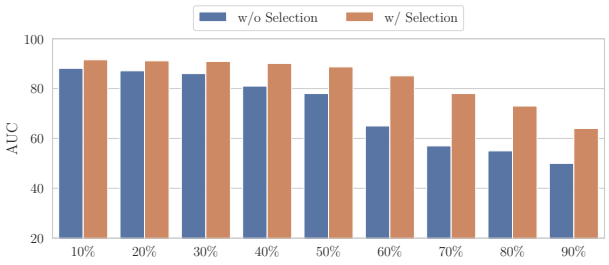

Figure 4: The effect of selection mechanism on noisy data.

6. Conclusion

In conclusion, in this paper, we developed BrainMamba, a general and efficient encoder for modeling spatio-temporal long-range dependencies in multivariate brain signals. Its design allows for encoding the actual brain signals, using an timeseries encoder, and their spatial dependencies, using a graph encoder, making it a powerful model for variety of neuroimaging data. While using two encoders, its efficient design based on selective state space models make its time complexity linear with respect to the input data, enabling training on large-scale data. Our experimental evaluations on various tasks, including seizure, ADHD, mental state detection, and with three different modalities show that BrainMamba outperforms all the baselines, while using less time and memory.

References

Charu C. Aggarwal, Yuchen Zhao, and Philip S. Yu. Outlier detection in graph streams. In *2011 IEEE 27th International Conference on Data Engineering*, pages 399–409, 2011. doi: 10.1109/ICDE.2011.5767885.

Masanao Aoki. *State space modeling of time series*. Springer Science & Business Media, 2013.

Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A Zuluaga. Usad: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3395–3404, 2020.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016.

Anahit Babayan, Miray Erbey, Deniz Kumral, Janis D Reinelt, Andrea MF Reiter, Josefin Röbbig, H Lina Schaare, Marie Uhlig, Alfred Anwander, Pierre-Louis Bazin, et al. A mind-brain-body dataset of mri, eeg, cognition, emotion, and peripheral physiology in young and old adults. *Scientific data*, 6(1):1–21, 2019.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.0473.

Danielle S. Bassett and Olaf Sporns. Network neuroscience. *Nature Neuroscience*, 20(3):353–364, Mar 2017. ISSN 1546-1726. doi: 10.1038/nn.4502. URL https://doi.org/10.1038/nn.4502.

Ali Behrouz and Farnoosh Hashemi. Learning temporal higher-order patterns to detect anomalous brain activity. In Stefan Hegselmann, Antonio Parziale, Divya Shanmugam, Shengpu Tang, Mercy Nyamewaa Asiedu, Serina Chang, Tom Hartvigsen, and Harvineet Singh, editors, *Proceedings of the 3rd Machine Learning for Health Symposium*, volume 225 of *Proceedings of Machine Learning Research*, pages 39–51. PMLR, 10 Dec 2023. URL https://proceedings.mlr.press/v225/behrouz23a.html.

Ali Behrouz and Farnoosh Hashemi. Graph mamba: Towards learning on graphs with state space models. *arXiv preprint arXiv:2402.08678*, 2024.

Ali Behrouz and Margo Seltzer. Anomaly detection in human brain via inductive learning on temporal multiplex networks. In *Machine Learning for Healthcare Conference*, volume 219. PMLR, 2023a.

Ali Behrouz and Margo Seltzer. ADMIRE++: Explainable anomaly detection in the human brain via inductive learning on temporal multiplex networks. In *ICML 3rd Workshop on Interpretable Machine Learning in Healthcare (IMLH)*, 2023b. URL https://openreview.net/forum?id=t4H8acYudJ.

Ali Behrouz, Parsa Delavari, and Farnoosh Hashemi. Unsupervised representation learning of brain activity via bridging voxel activity and functional connectivity. In *NeurIPS 2023 AI for Science Workshop*, 2023. URL https://openreview.net/forum?id=HSvg7qFFd2.

Ali Behrouz, Michele Santacatterina, and Ramin Zabih. Mambamixer: Efficient selective state space models with dual token and channel selection. *arXiv preprint arXiv:2403.19888*, 2024.

Donghong Cai, Junru Chen, Yang Yang, Teng Liu, and Yafeng Li. Mbrain: A multi-channel self-supervised learning framework for brain signals. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 130–141, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701030. doi: 10.1145/3580305.3599426. URL https://doi.org/10.1145/3580305.3599426.

Tanima Chatterjee, Réka Albert, Stuti Thapliyal, Nazanin Azarhooshang, and Bhaskar DasGupta. Detecting network anomalies using forman–ricci curvature and a case study for human brain networks. *Scientific Reports*, 11(1):8121, Apr 2021. ISSN 2045-2322. doi: 10.1038/s41598-021-87587-z. URL https://doi.org/10.1038/s41598-021-87587-z.

Si-An Chen, Chun-Liang Li, Nate Yoder, Sercan O Arik, and Tomas Pfister. Tsmixer: An all-mlp architecture for time series forecasting. *arXiv preprint arXiv:2303.06053*, 2023.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Weilin Cong, Si Zhang, Jian Kang, Baichuan Yuan, Hao Wu, Xin Zhou, Hanghang Tong, and Mehrdad Mahdavi. Do we really need complicated model architectures for temporal networks? In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ayPPc0SyLv1.

Alexander Craik, Yongtian He, and Jose L Contreras-Vidal. Deep learning for electroencephalogram (eeg) classification tasks: a review. *Journal of neural engineering*, 16(3):031001, 2019.

Hejie Cui, Wei Dai, Yanqiao Zhu, Xuan Kan, Antonio Aodong Chen Gu, Joshua Lukemire, Liang Zhan, Lifang He, Ying Guo, and Carl Yang. BrainGB: A Benchmark for Brain Network Analysis with Graph Neural Networks. *IEEE Transactions on Medical Imaging (TMI)*, 2022.

Francesco Di Giovanni, Lorenzo Giusti, Federico Barbero, Giulia Luise, Pietro Lio, and Michael M Bronstein. On over-squashing in message passing neural networks: The impact of width, depth, and topology. In *International Conference on Machine Learning*, pages 7865–7885. PMLR, 2023.

Eleonora Fiorenzato, Antonio P Strafella, Jinhee Kim, Roberta Schifano, Luca Weis, Angelo Antonini, and Roberta Biundo. Dynamic functional connectivity changes associated with dementia in parkinson's disease. *Brain*, 142(9):2860–2872, 2019.

Nicolai Franzmeier, Julia Neitzel, Anna Rubinski, Ruben Smith, Olof Strandberg, Rik Ossenkoppele, Oskar Hansson, and Michael Ewers. Functional brain architecture is associated with the rate of tau accumulation in alzheimer's disease. *Nature communications*, 11(1):347, 2020.

Douglas N Greve, Gregory G Brown, Bryon A Mueller, Gary Glover, Thomas T Liu, and Function Biomedical Research Network. A survey of the sources of noise in fmri. *Psychometrika*, 78: 396–416, 2013.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. Hippo: Recurrent memory with optimal polynomial projections. *Advances in neural information processing systems*, 33:1474–1487, 2020.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010.

Xiaoxin He, Bryan Hooi, Thomas Laurent, Adam Perold, Yann LeCun, and Xavier Bresson. A generalization of vit/mlp-mixer to graphs. In *International Conference on Machine Learning*, pages 12724–12745. PMLR, 2023.

Jinlong Hu, Yangmin Huang, Nan Wang, and Shoubin Dong. Brainnpt: Pre-training of transformer networks for brain network classification. *arXiv preprint arXiv:2305.01666*, 2023.

Xuan Kan, Hejie Cui, Ying Guo, and Carl Yang. Effective and interpretable fmri analysis via functional brain network generation. *arXiv preprint arXiv:2107.11247*, 2021.

Xuan Kan, Hejie Cui, Joshua Lukemire, Ying Guo, and Carl Yang. Fbnetgen: Task-aware gnn-based fmri analysis via functional brain network generation. In *International Conference on Medical Imaging with Deep Learning*, pages 618–637. PMLR, 2022a.

Xuan Kan, Wei Dai, Hejie Cui, Zilong Zhang, Ying Guo, and Carl Yang. Brain network transformer. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022b. URL https://openreview.net/forum?id=1cJ1cbA6NLN.

Jeremy Kawahara, Colin J Brown, Steven P Miller, Brian G Booth, Vann Chau, Ruth E Grunau, Jill G Zwicker, and Ghassan Hamarneh. Brainnetcnn: Convolutional neural networks for brain networks; towards predicting neurodevelopment. *NeuroImage*, 146:1038–1049, 2017.

Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan.

Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in neural information processing systems*, 32, 2019.

Xiaoxiao Li, Yuan Zhou, Nicha Dvornek, Muhan Zhang, Siyuan Gao, Juntang Zhuang, Dustin Scheinost, Lawrence H Staib, Pamela Ventola, and James S Duncan. Braingnn: Interpretable brain graph neural network for fmri analysis. *Medical Image Analysis*, 74:102233, 2021.

Zhe Li, Zhongwen Rao, Lujia Pan, and Zenglin Xu. Mts-mixers: Multivariate time series forecasting via factorized temporal and channel mixing. *arXiv preprint arXiv:2302.04501*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Xiaoxiao Ma, Jia Wu, Shan Xue, Jian Yang, Chuan Zhou, Quan Z. Sheng, Hui Xiong, and Leman Akoglu. A comprehensive survey on graph anomaly detection with deep learning. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2021. doi: 10.1109/TKDE.2021.3118815.

Michael P. Milham, Jan Buitelaar, F. Xavier Castellanos, Daniel Dickstein, Damien Fair, David Kennedy, Beatric Luna, Michael P. Milham, Stewart Mostofsky, Joel Nigg, Julie B. Schweitzer, Katerina Velanova, Yu-Feng Wang, and Yu-Feng Zang. 1000 functional connectome project. *1000 Functional Connectome Project*, 1, July 2011.

Yue-Ting Pan, Jing-Lun Chou, and Chun-Shu Wei. MAtt: A manifold attention network for EEG decoding. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

Russell A Poldrack and Krzysztof J Gorgolewski. Making big data open: data sharing in neuroimaging. *Nature neuroscience*, 17(11):1510–1517, 2014.

İlkay Yıldız Potter, George Zerveas, Carsten Eickhoff, and Dominique Duncan. Unsupervised multivariate time-series transformers for seizure identification on eeg. In *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1304–1311. IEEE, 2022.

Anwar Said, Roza G Bayrak, Tyler Derr, Mudassir Shabbir, Daniel Moyer, Catie Chang, and Xenofon D. Koutsoukos. Neurograph: Benchmarks for graph machine learning in brain connectomics. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=MEa0cQeURw.

Alexander Schaefer, Ru Kong, Evan M Gordon, Timothy O Laumann, Xi-Nian Zuo, Avram J Holmes, Simon B Eickhoff, and B T Thomas Yeo. Local-Global parcellation of the human cerebral cortex from intrinsic functional connectivity MRI. *Cereb Cortex*, 28(9):3095–3114, September 2018.

Steffen Schneider, Jin Hwa Lee, and Mackenzie Weygandt Mathis. Learnable latent embeddings for joint behavioural and neural analysis. *Nature*, pages 1–9, 2023.

Vinit Shah, Eva Von Weltin, Silvia Lopez, James Riley McHugh, Lillian Veloso, Meysam Golmohammadi, Iyad Obeid, and Joseph Picone. The temple university hospital seizure detection corpus. *Frontiers in neuroinformatics*, 12:83, 2018.

Afshin Shoeibi, Marjane Khodatars, Navid Ghassemi, Mahboobeh Jafari, Parisa Moridian, Roohallah Alizadehsani, Maryam Panahiazar, Fahime Khozeimeh, Assef Zare, Hossein Hosseini-Nejad, et al. Epileptic seizures detection using deep learning techniques: A review. *International Journal of Environmental Research and Public Health*, 18(11): 5780, 2021.

Stephen M Smith, Diego Vidaurre, Christian F Beckmann, Matthew F Glasser, Mark Jenkinson, Karla L Miller, Thomas E Nichols, Emma C Robinson, Gholamreza Salimi-Khorshidi, Mark W Woolrich, Deanna M Barch, Kamil Uğurbil, and David C Van Essen. Functional connectomics from resting-state fMRI. *Trends Cogn Sci*, 17(12):666–682, November 2013.

D Puthankattil Subha, Paul K Joseph, Rajendra Acharya U, and Choo Min Lim. Eeg signal analysis: a survey. *Journal of medical systems*, 34:195–212, 2010.

Siyi Tang, Jared Dunnmon, Khaled Kamal Saab, Xuan Zhang, Qianying Huang, Florian Dubost, Daniel Rubin, and Christopher Lee-Messer. Self-supervised graph neural networks for improved

electroencephalographic seizure analysis. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=k9bx1EfHI_-.

Siyi Tang, Jared A Dunnmon, Qu Liangqiong, Khaled K Saab, Tina Baykaner, Christopher Lee-Messer, and Daniel L Rubin. Modeling multivariate biosignals with graph neural networks and structured state space models. In Bobak J. Mortazavi, Tasmie Sarker, Andrew Beam, and Joyce C. Ho, editors, *Proceedings of the Conference on Health, Inference, and Learning*, volume 209 of *Proceedings of Machine Learning Research*, pages 50–71. PMLR, 22 Jun–24 Jun 2023. URL https://proceedings.mlr.press/v209/tang23a.html.

Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Peter Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. MLP-mixer: An all-MLP architecture for vision. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=EI2KOXKdnP.

Lucina Q Uddin, DR Dajani, W Voorhies, H Bednarz, and RK Kana. Progress and roadblocks in the search for brain-based biomarkers of autism and attention-deficit/hyperactivity disorder. *Translational psychiatry*, 7(8):e1218–e1218, 2017.

David C Van Essen, Stephen M Smith, Deanna M Barch, Timothy EJ Behrens, Essa Yacoub, Kamil Ugurbil, Wu-Minn HCP Consortium, et al. The wu-minn human connectome project: an overview. *Neuroimage*, 80:62–79, 2013.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Junxiong Wang, Jing Yan, Albert Gu, and Alexander Rush. Pretraining without attention. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 58–69, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.5. URL https://aclanthology.org/2023.findings-emnlp.5.

Sifan Wu, Xi Xiao, Qianggang Ding, Peilin Zhao, Ying Wei, and Junzhou Huang. Adversarial sparse transformer for time series forecasting. *Advances in neural information processing systems*, 33:17105–17115, 2020.

Daniel LK Yamins and James J DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nature neuroscience*, 19(3):356–365, 2016.

Yi Yang, Hejie Cui, and Carl Yang. Ptgb: Pre-train graph neural networks for brain network analysis. In *Conference on Health, Inference, and Learning*, pages 526–544. PMLR, 2023.

Wenchao Yu, Wei Cheng, Charu C. Aggarwal, Kai Zhang, Haifeng Chen, and Wei Wang. Netwalk: A flexible deep embedding approach for anomaly detection in dynamic networks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, page 2672–2681, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355520. doi: 10.1145/3219819.3220024. URL https://doi.org/10.1145/3219819.3220024.

Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. *Advances in neural information processing systems*, 32, 2019.

George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 2114–2124, 2021.

Michael Zhang, Khaled Kamal Saab, Michael Poli, Tri Dao, Karan Goel, and Christopher Re. Effectively modeling time series with simple discrete state spaces. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=2EpjkjzdCAa.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of*

the AAAI conference on artificial intelligence, volume 35, pages 11106–11115, 2021.

Yanqiao Zhu, Hejie Cui, Lifang He, Lichao Sun, and Carl Yang. Joint embedding of structural and functional brain networks with graph neural networks for mental illness diagnosis. In *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 272–276. IEEE, 2022.

Appendix A. Details of Datasets

A.1. Datasets

We use seven real-world datasets with different neuroimage modalities and downstream tasks, whose descriptions are as follows:

- BVFC (Behrouz et al., 2023): BVFC is a task-based fMRI dataset that includes voxel activity timeseries and functional connectivity of 3 subjects when looking at the 8460 images from 720 categories. For the multi-class classification task, we aim to predict the label of the seen image (9 labels) using the fMRI response of a human subject (3 subjects). For the node-level anomaly detection tasks, we use synthetic injected anomalies and for the graph anomaly detection, we aim to detect GAN generated images using the fMRI response. We label brain activities that correspond to seeing a GAN generated image (resp. natural image) as "Anomalous" (resp. "Normal"). In multi-class classification tasks, the labels are "Food", "Human Body", "Car", "Fruit", "Animals", "Verdure", "Accessories", "Fish", and "Misc.".

- BVFC-MEG (Behrouz et al., 2023): BVFC-MEG is the MEG counterpart of the BVFC. For the multi-class classification task, we aim to predict the label of the seen image (9 labels) using the MEG response of a human subject (4 subjects). For the node-level anomaly detection tasks, we use synthetic injected anomalies and for the graph anomaly detection, we aim to detect natural scenes using the MEG response. We label MEG response that correspond to seeing natural scenes as "Anomalous" and seeing other objects as "Normal". The labels in multiclass classification tasks are the same as BVFC.

- ADHD (Milham et al., 2011): ADHD (Milham et al., 2011) contains resting-state fMRI of 250 subjects in the ADHD group and 450 subjects in the typically developed (TD) control group. We follow the standard pre-processing steps (Cui et al., 2022) to obtain brain networks. For the edge and node anomaly detection tasks, we use synthetic anomalies, while for the graph anomaly detection task we label brain networks of the typically developed control group as "Normal" and brain networks of the ADHD group as "Anomalous".

- TUH-EEG (Shah et al., 2018): The seizure detection TUH-EEG dataset (Shah et al., 2018) consists of EEG data with 31 channels of 642 subjects. For the edge and node anomaly detection tasks, we use synthetic anomalies, while for the graph anomaly detection task we label brain networks of people with and without seizure as "Anomalous" and "Normal", respectively.

- MPI-EEG: MPI-EEG (Babayan et al., 2019) consists of 204 EEG data (two healthy groups (1) young with age $25.1_{\pm 3.1}$ and (2) elderly with age 67.6 ± 4.7) with 62 chennels.

- HCP (Van Essen et al., 2013): HCP (Van Essen et al., 2013) contains data from 7440 neuroimaging samples each of which is associated with one of the seven ground-truth mental states. Following previous studies (Said et al., 2023), we define two downstream multi-class classification tasks: ① Mental states prediction, in which we aim to predict the mental state using the fMRI. In these tasks, the labels are "Emotion Processing", "Gambling", "Language", "Motor", "Relational Processing", "Social Cognition", and "Working Memory". ② We aim to predict the age of human subjects using their fMRI. In this tasks, we split the age into 5 groups, balancing the number of samples in each class. Similar to other datasets, we use synthetic anomalies for the edge and node anomaly detection tasks.

Appendix B. Details of Baselines

Since BrainMamba combines functional connectivity and voxel timeseries activity, we compare our method to fourteen previous state-of-the-art methods

Table 5: Datasets statistics.

Datasets	Number of Graphs	Average Number of Nodes	Average Number of Edges	Number of Classes (Multi-class Classification)	Ground-Truth Anomaly (Binary Classification)
BVFC	25380	11776	352479	9	Yes
BVFC-MEG	88992	272	9841	9	Yes
ADHD	700	400	6194	-	Yes
TUH-EEG	642	31	252	-	Yes
MPI-EEG	204	62	2419	-	Yes
HCP	7440 1067	1000	7635 8041	7 (Mental states) 5 (Age)	Yes

and baselines on the timeseries, functional connectivity, and graph encoding:

1. BrainMixer (Behrouz et al., 2023): BrainMixer uses two encoders, one for timeseries encoding and one for graph encoding based on the MLP-Mixer architecture. Then maximizes the mutual information of these two encoders to learn the timeseries encodings.

2. Graph MLP-Mixer (GMM) (He et al., 2023): Graph MLP-Mixer uses graph partitioning algorithms to split the input graph into overlapping graph patches (subgraphs) and then employs a graph neural network to encode each patch. It then uses an MLP to fuse information across patch encodings. The model with code can be found in here. Note that Graph MLP-Mixer cannot take advantage of temporal properties of the graph as it is designed for static networks.

3. GraphMixer (Cong et al., 2023): GraphMixer is a simple method that concatenates the 1-hop temporal connections and their time encoding of each node as its representative matrix. It then uses an MLPMixer to encode each representative matrix to obtain node encodings. The model with code can be found in here.

4. FbNetGen (Kan et al., 2022a): FbNetGen is a graph neural network based generative model for functional brain networks from fMRI data that includes three components: a dimension reduction phase to denoise the raw time-series data, a graph generator for brain networks generation from the encoded features, and a GNN predictor for predictions based on the generated brain networks. The model with code can be found in here.

5. BrainGnn (Li et al., 2021): BrainGnn is a graph neural network-based framework that maps regional and cross-regional functional connectivity patterns. Li et al. (2021) propose a novel clustering-based embedding method in the graph convolutional layer as well as a graph node pooling to learn ROI encodings in the brain. The model with code can be found in here.

6. BrainNetCnn (Kawahara et al., 2017): BrainNetCnn is a CNN-based approach that uses novel edge-to-edge, edge-to-node and node-to-graph convolutional filters that leverage the topological locality of structural brain networks.

7. ADMire (Behrouz and Seltzer, 2023b): ADMire is a random walk-based approach that models brain connectivity networks as multiplex graphs. It uses inter-view and intra-view walks to capture the causality between different neuroimage modalities or different frequency band filters.

8. BNTransformer (Kan et al., 2022b): BNTransformer adapts Transformers (Vaswani et al., 2017) to brain networks, so it can use unique properties of brain networks. BNTransformer use connection profiles as node features to provide low-cost positional information and then learns pair-wise connection strengths among ROIs with efficient attention weights. It further uses a novel READOUT operation based on self-supervised soft clustering and orthonormal projection. The model with code can be found in here.

9. PTGB (Yang et al., 2023): PTGB is an unsupervised pre-training method designed specifically for brain networks using contrastive learning and maximizing the mutual information between an anchor point of investigation X from a data distribution H and its positive samples, while minimizing its mutual information with its negative

samples. The model with code can be found in here.

10. Usad (Audibert et al., 2020): Usad is an unsupervised representation learning method in time series, which utilizes an encoder-decoder architecture within an adversarial training framework that allows it to take advantage of both.

11. Time Series Transformer (TST) (Zerveas et al., 2021): TST is a transformer-based framework for unsupervised representation learning of multivariate time series, which is capable of pre-training and can be employed on varius downstream tasks.

12. Mvts (Potter et al., 2022): Mvts is an unsupervised transformer-based model for time series learning, which utilizes special properties of EEGs for seizure identification. It uses an autoencoder mechanism involving a transformer encoder and an unsupervised loss function for training.

We use the same hyperparameter selection process as BrainMamba. Also, we fine tune their training parameters as their original papers using grid search. For the sake of fair comparison, we use the same training, testing and validation data for all the baselines (including same data augmentation and negative sampling). Also, for PTGB (Yang et al., 2023) and BrainMixer (Behrouz et al., 2023), which also are capable of pre-training, we use the same datasets and settings as we use for BrainMamba.

Appendix C. Model Size

To show the efficiency of BrainMamba and to compare it with competitors with respect to the model size, we report the number of parameters of BrainMamba, BrainMixer, and BNTransformer in Figure 5. Mamba backbone of BrainMamba avoids attention module and with less number of parameters provides better performance. These results futher support that the superior performance of BrainMamba over baselines is not because of the large number of parameters, but it is because of its architectural design.

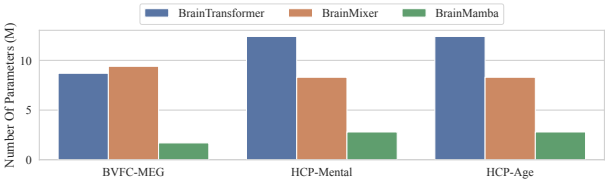

Figure 5: The number of parameters in BrainMamba and its main competitors, BrainMixer and BNTransformer.

Appendix D. Details of Training

In the training of BrainMamba, We randomly split 70% of the datasets for training, 10% for validation, and the remaining 20% are utilized as the test set (making sure that there is no overlap between training and test sets.). We employ AdamW (Loshchilov and Hutter, 2019) with a momentum of 0.9, and a weight decay of 0.05 to optimize BrainMamba. The batch size is set as 64. While we use the same batch size for all baselines, particularly, BNTransformer, GraphS4mer, and BrainMixer face out of memory issue on BVFC dataset. The main reason is the number of voxels in this dataset (> 10000), which due to the quadratic space complexity of Transformers used in these models results in OOM issue. For this method, we also tried batch size of 8, which did not solve this OOM issue.

Appendix E. Limitations & Future Work

The success of BrainMamba in binary and multi-class classification tasks raises many interesting directions for future studies: ① Wider variety of neurological conditions: For the sake of space and time, the current experiments are limited to benchmark studies on ADHD, Seizure, and mental state detection, as well as visual cortex decoding. To further show the usefulness of BrainMamba in wider variety of neurological conditions, we plan to apply BrainMamba for Alzheimer's disease, Autism Spectrum Disorder (ASD), and Schizophrenia, which all are known to be correlated with functional connectivity. ② The current method is based on pre-defined brain network connectivity using Pearson's correlation. In our future study, we plan to learn the functional connectivity network in a data-driven and end-to-end manner. This end-to-end design results in more flexibility

and generalizibility of the framework. ③ The current framework treats each subject the same in the training process, while the neuroimaging of a subject might be noisy, due to special conditions in the time of data collection. To address it, one future direction is to investigate the possibility of robust training across different populations using attention mechanism. ④ BrainMamba is based on sequence encoding and translates both voxel activity and functional connectivity to sequences. Potentially, this idea can be extended to other modalities like historical medical record (text), and/or multimodal neuroimages (fMRI + structural MRI, fMRI + EEG, etc.). One future direction is to extend BrainMamba framework to support multiple modalities, which can help to improve the generalizibility and performance. ⑤ The evaluation of the robustness of BrainMamba with respect to distribution shifts is left for future studies. We hypothesize that due to the nature of architecture design of existing methods as well as BrainMamba, they all might perform poorly with respect to distribution shifts; We expect, however, BrainMamba, due to its selection mechanism, generalize better than existing methods (Behrouz et al., 2023; Kan et al., 2022b) and further our suggestion in ③ can make the model more robust with respect to distribution shifts. For the sake of space, we leave this evaluation for future work.