UNSUPERVISED REPRESENTATION LEARNING OF BRAIN ACTIVITY VIA BRIDGING VOXEL ACTIVITY AND FUNCTIONAL CONNECTIVITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Effective brain representation learning is a key step toward the understanding of cognitive processes and unlocking detecting and potential therapeutic interventions for neurological diseases/disorders. Existing studies have focused on either (1) voxel-level activity, where only a single weight relating the voxel activity to the task (i.e., aggregation of voxel activity over a time window) is considered, missing their temporal dynamics, or (2) functional connectivity of the brain in the level of region of interests, missing voxel-level activities. In this paper, we bridge this gap and design BRAINMIXER, an unsupervised learning framework that effectively utilizes both functional connectivity and associated time series of voxels to learn voxel-level representation in an unsupervised manner. BRAINMIXER employs two simple yet effective MLP-based encoders to simultaneously learn the dynamics of voxel-level signals and their functional correlations. To encode voxel activity, BRAINMIXERfuses information across both time and voxel dimensions via a dynamic self-attention mechanism. To learn the structure of the functional connectivity graph, BRAINMIXER presents a temporal graph patching and encodes each patch by combining its nodes' features via a new adaptive temporal pooling. Our experiments show that BRAINMIXER attains outstanding performance and outperforms 14 baselines in different downstream tasks and experimental setups.

1 INTRODUCTION

The recent advancement of neuroimaging has provided rich information to analyze the human brain. The provided data, however, is high-dimensional and complex in nature (Poldrack & Gorgolewski, 2014), which makes it hard to take advantage of powerful machine learning models in analyzing them. To overcome this challenge, representation learning serves as the backbone of machine learning methods on neuroimage data and provides a low-dimensional representation of brain components at different levels of granularity, enabling the understanding of behaviors (Schneider et al., 2023), brain functions (Yamins & DiCarlo, 2016) and/or detecting neurological diseases (Uddin et al., 2017).

In the brain imaging literature, studies have mainly focused on two spatial scales—voxel-level and network-level—as well as two analysis approaches—multivariate pattern analysis (MVPA) and functional connectivity (Mahmoudi et al., 2012; Van Den Heuvel & Pol, 2010). The MVPA approach is often employed at the voxel-level scale and in task-based studies to associate neural activities at a very fine-grained and local level with particular cognitive functions, behaviors, or stimuli. This method has found applications in various areas, including the detection of neurological conditions (Sundermann et al., 2014; Bray et al., 2009), neurofeedback interventions (Cortese et al., 2021), decoding neural responses to visual stimuli (Horikawa & Kamitani, 2017), deciphering memory contents (Lee & Baker, 2016; Chadwick et al., 2012), and classifying cognitive states (Mitchell et al., 2003). The functional connectivity analysis, on the other hand, focuses on the temporal correlations or statistical dependencies between the activity of different brain regions at larger scales to assess how these areas communicate and collaborate. This method has been utilized to study various topics such as task-related network dynamics (Gonzalez-Castillo & Bandettini, 2018; Hutchison et al., 2013) and the effects of neurological disorders on brain connectivity (Greicius, 2008; Du et al., 2018).

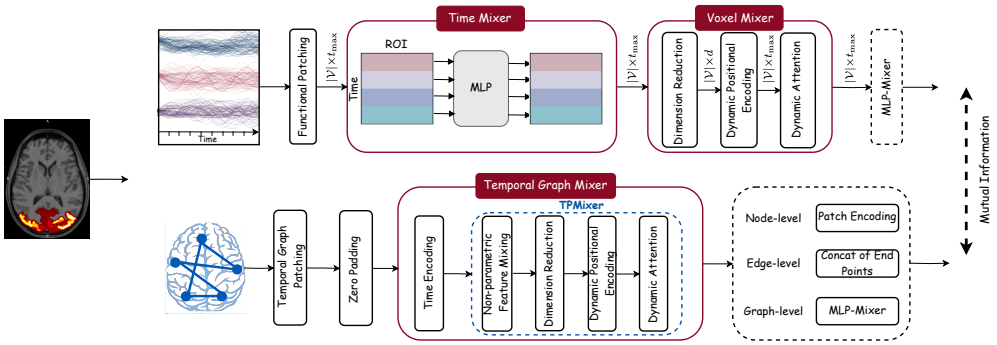


Figure 1: **Schematic of the BRAINMIXER**. BRAINMIXER consists of two main modules: (1) Voxel Activity Encoder (top), and (2) Functional Connectivity Encoder (bottom).

Limitation of Previous Methods. Despite the advances in the representation learning of brain signals, existing studies suffer from a subset of five limitations: ① Study the human brain at a single scale: Most existing studies study the brain at either voxel-level or functional connectivity, while these two scales can provide complementary information to each other; e.g., although voxel-level activity provides detailed and more accurate information about brain activity, it misses the information about how different areas communicate with each other at a high level. Recently, this limitation has motivated researchers to search for new methods of integrating these two levels of analyses (Nieto-Castanon, 2022; McNorgan et al., 2020). ② Supervised setting: Learning brain activity in a supervised setting relies on a large number of clinical labels while obtaining accurate and reliable clinical labels is challenging due to its high cost (Avberšek & Repovš, 2022).③ Missing information by averaging: Most existing studies on voxel activities aggregate measured voxel activity (e.g., its blood-oxygen level dependence) over each time window to obtain a single beta weight (Roth et al., 2022; Vassena et al., 2020; Roth & Merriam, 2023). However, this approach misses the voxel activity dynamic over each task. Moreover, most studies on brain functional connectivity also aggregate closed voxels to obtain brain activity in the Region of Interest (ROI) level, missing individual voxel activities. ④ Missing the dynamics of the interactions: Some existing studies neglect the fact that the functional connectivity of the human brain dynamically changes over time, even in resting-state neuroimaging data (Calhoun et al., 2014). In task-dependent neuroimage data, subjects are asked to perform different tasks in different time windows, and the dynamics of the brain activity play an important role in understanding neurological disease/disorder (Hernandez et al., 2015). ⑤ Designed for a particular task or neuroimaging modality: Due to the different and complex clinical patterns of brain signals (da Silva, 1991), some existing methods are designed for a particular type of brain signal data (Lanciano et al., 2020; Cai et al., 2023), and there is a lack of a unified framework.

Application to Understanding Object Representation in the Brain. Understanding object representation in the brain is a key step toward revealing the basic building blocks of human visual processing (Hebart et al., 2023). Due to the hierarchical nature of human visual processing, it requires analyzing brain activity at different scales, i.e., both functional connectivity and voxel activity. However, there is a small number of studies in this area, possibly due to the lack of proper large-scale datasets. Recently, Hebart et al. (2023) provided a large-scale fMRI and MEG datasets, THINGS, to fill this gap. However, the preprocessed data by Hebart et al. (2023) not only does not provide functional connectivity, but it also has aggregated voxel activity over each time window, and provides a single beta weight for each voxel, missing dynamics of voxel activity. To address this limitation, we present two newly preprocessed versions of this dataset that provide both functional connectivity and voxel activity timeseries of fMRI and MEG modalities. See Appendix B for more details.

Contributions. To overcome the above limitations, we leverage both voxel-level activity and functional connectivity of the brain. We present BRAINMIXER, an unsupervised MLP-based brain representation learning approach that jointly learns representations of the voxel activity and functional connectivity. BRAINMIXER uses a novel multivariate timeseries encoder that binds information across both time and voxel dimensions. It uses a simple MLP with functional patching to fuse information across different timestamps and learns dynamic self-attention weights to fuse information across voxels based on their functionality. On the other hand, BRAINMIXER uses a novel temporal graph learning method to encode the brain functional connectivity. The graph encoder first extracts temporal patches using temporal random walks and then fuses information within each patch using the

designed dynamic self-attention mechanism. We further propose an adaptive permutation invariant pooling to obtain patch encodings. Since voxel activity and functional connectivity encodings are different views of the same context, we propose an unsupervised pre-training approach to jointly learn voxel activity and functional connectivity by maximizing their mutual information. In the experimental evaluations, we provide two new large-scale graph and timeseries datasets based on THINGS (Hebart et al., 2023). Extensive experiments on six datasets show the superior performance of BRAINMIXER and the significance of each of its components in a variety of downstream tasks.

For the sake of consistency, we explain BRAINMIXER for fMRI modality; however, as it is shown in §4, it can simply be used for any other neuroimaging modalities that provide a timeseries for each part of the brain (e.g., MEG and EEG). When dealing with MEG or EEG, we can replace the term "voxel" with "channel". Supplementary materials can be found in this link.

2 RELATED WORK

To situate our BRAINMIXER in a broader context, we briefly review machine learning models for timeseries, graphs, and neuroscience. For additional discussion of related work see Appendix C.

Timeseries Learning. Attention mechanisms are powerful models to capture long-range dependencies and so recently, Transformer-based models have attracted much attention in time series forecasting (Zerveas et al., 2021; Li et al., 2019). Due to the quadratic time complexity of attention mechanisms, several studies aim to reduce the time and memory usage of these methods (Child et al., 2019). Another type of work uses (hyper)graph learning frameworks to learn (higher-order) patterns in timeseries (Park et al., 2009; Sawhney et al., 2021). 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. All these methods are different from BRAINMIXER, as ① they use static attention mechanisms, ② do not take advantage of the functionality of voxels in patching, and ③ are designed for timeseries forecasting and cannot simply be extended to various downstream tasks on the brain.

MLP-based Graphs Learning. Learning on graphs has been an active research area in recent years (Jiang et al., 2021; Veličković et al., 2018; Chamberlain et al., 2023). While most studies use message-passing frameworks to learn the local and global structure of the graph, recently, due to the success of MLP-based methods (Tolstikhin et al., 2021), MLP-based graph learning methods have attracted much attention (Hu et al., 2021; Behrouz et al., 2023). For example, Cong et al. (2023) and He et al. (2023) presented two extensions of MLP-MIXER to graph-structured data. However, all these methods are different from BRAINMIXER and specifically FC Encoder, as either ① use time-consuming graph clustering algorithms for patching, ② are static methods and cannot capture temporal properties, or ③ are attention-free and cannot capture the importance of nodes.

Graph Learning and Timeseries for Neuroscience. In recent years, several studies have analyzed functional connectivity to differentiate human brains with a neurological disease/disorder (Jie et al., 2016; Chen et al., 2011; Wee et al., 2011). With the success of graph neural networks in graph data analysis, deep learning models have been developed to predict brain diseases by studying brain network structures (Behrouz & Seltzer, 2022; Zhu et al., 2022; Cui et al., 2022b). Moreover, several studies focus on brain signals (Craik et al., 2019; Shoeibi et al., 2021) to detect neurological diseases. 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 different from BRAINMIXER as they are designed for a particular task (e.g., brain classification), a particular neuroimaging modality (e.g., fMRI or EEG), and/or supervised settings.

3 METHOD:BRAINMIXER

In this section, we first discuss the notation we use throughout the paper. Detailed discussion about background concepts can be found in Appendix A.

Notation. We represent the neuroimaging of a human brain as $\mathcal{B} = \{\mathcal{B}^{(t)}\}_{t=1}^{T}$ where $\mathcal{B}^{(t)} = (\mathcal{V}, \mathcal{G}_F^{(t)}, \mathcal{X}^{(t)}, \mathbb{F})$ represents the neural data in time window $1 \leq t \leq T$. Here, $\mathcal{V}$ is the set of voxels,

$\mathcal{G}_F^{(t)} = (\mathcal{V}, \mathcal{E}^{(t)}, \mathcal{A}^{(t)})$ is the functional connectivity graph, $\mathcal{E}^{(t)} \subseteq \mathcal{V} \times \mathcal{V}$ is the set connections between voxels, $\mathcal{A}^{(t)}$ is the correlation matrix (weighted adjacency matrix of $\mathcal{G}_F^{(t)}$), $\mathcal{X}^{(t)} \in \mathbb{R}^{|\mathcal{V}| \times \tilde{T}(t)}$ is a multivariate timeseries of voxels activities, $\tilde{T}(t)$ is the length of the timeseries, and $\mathbb{F}$ is the set of functional systems in the brain (Schaefer et al., 2018) 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 let $t_{\max} = \max_{t=1,\ldots,T} \tilde{T}(t)$, representing the maximum length of timeseries. BRAINMIXER consists of two main modules ① *Voxel Activity* (VA) Encoder and ② Functional Connectivity (FC) Encoder:

3.1 VOXEL ACTIVITY ENCODER

The main goal of this module is to learn the time series of the voxel-level activity. However, the activities of voxels are not disjoint; for example, an increase in fusiform face area (FFA) activity might be associated with a rise in V1 activity. Accordingly, effectively learning their dynamics patterns requires both capturing cross-voxel and within-voxel time series information. The vanilla MLP-MIXER (Tolstikhin et al., 2021) can be used to bind information across both of these dimensions, but the human brain has unique traits that make directly applying vanilla MLP-MIXER insufficient/impractical. First, there does not exist in general a canonical grid of the brain to encode voxel activities, which makes patch extraction challenging. Second, contrary to images that can be divided into patches of the same size, the partitioning of voxels might not be all the same size due to the complex brain topology. Third, vanilla MLP-MIXER employs a fixed static mixing matrix for binding patches, while in the brain the functionality of each patch is important and a different set of patchs should be mixed differently based on their connections and functionality. To address these challenges, the *VA Encoder* employs two submodules, *time-mixer* and *voxel-mixer* with dynamic mixing matrix, to fuse information across both time and voxel dimensions, respectively.

The human brain is comprised of functional systems (FS) (Schaefer et al., 2018), which are groups of voxels that perform similar functions (Smith et al., 2013). We take advantage of this hierarchical structure and patch voxels based on their functionality. However, the main challenge is that the sizes of the patches (set of voxels with similar functionality) are different. To this end, inspired by the inference of ViT models (Dosovitskiy et al., 2021), we linearly interpolate patches with smaller sizes.

Functional Patching. Let $|\mathcal{V}|$ be the number of voxels and $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times (T \times t_{\max})}$ represents the time series of voxels activities over all time windows. We split $\mathbf{X}$ to spatio-temporal patches $\mathbf{X}_i$ with size $|f_i| \times t_{\max}$, where $f_i \in \mathbb{F}$ is a functional system (Schaefer et al., 2018). To address the challenge of different patch sizes, we use INTERPOLATE$(.)$ to linearly interpolate patches to the same size N_p: i.e., $\tilde{\mathbf{X}}_i = \text{INTERPOLATE}(\mathbf{X}_i)$, where $\tilde{\mathbf{X}}_i \in \mathbb{R}^{N_p \times t_{\max}}$. We let $\tilde{\mathbf{X}} \in \mathbb{R}^{|\mathcal{V}| \times t_{\max}}$ be the matrix of $\tilde{\mathbf{X}}_i$.

Voxel-Mixer. Since the effect of each task (e.g., in task-based fMRI) on brain activity as well as the time it lasts varies (Yang et al., 2023a), for different tasks, we might need to emphasize more on a subset of voxels. To this end, to bind information across voxels, we use a dynamic attention mechanism that uses a learnable dynamic mixing matrix $\mathbf{P}_i$, learning to mix a set of input voxels based on their functionality. While using different learnable matrices for mixing voxels activity provides a more powerful architecture, its main challenge is a large number of parameters. To mitigate this challenge, we first reduce the dimensions of $\tilde{\mathbf{X}}$, split it into a set of segments, denoted as S, and then combine the transformed matrices. Given a segment $s \in S$ we have:

$$\hat{\mathbf{X}}^{(t)^{(s)}} = \tilde{\mathbf{X}}^{(t)} \, \mathbf{W}_{\text{segment}}^{(s)} \quad \in \mathbb{R}^{|\mathcal{V}| \times d}, \qquad \textit{(Dimension Reduction)}$$

$$\mathbf{P}_i^{(s)} = \text{SOFTMAX}\left(\text{FLAT}\left(\hat{\mathbf{X}}^{(t)^{(s)}}\right) \mathbf{W}_{\text{flat}}^{(s)^{(i)}}\right) \quad \in \mathbb{R}^{1 \times |\mathcal{V}|}, \qquad \textit{(Learning Dynamic Mixer)}$$

$$\mathbf{X}_{\text{PE}}^{(t)} = \left[\Big\|_{s \in S} \mathbf{P}^{(s)} \tilde{\mathbf{X}}^{(t)^{(s)}}\right] \mathbf{W}_{\text{PE}} \quad \in \mathbb{R}^{|\mathcal{V}| \times t_{\max}}, \qquad \textit{(Dynamic Positional Encoding)}$$

$$\mathbf{H}_{\text{Voxel}}^{(t)} = \texttt{Norm}\left(\tilde{\mathbf{X}}^{(t)}\right) + \text{SIGMOID}\left(\frac{\mathbf{X}_{\text{PE}}^{(t)} \, \mathbf{X}_{\text{PE}}^{(t)^\top}}{\sqrt{\tilde{T}}}\right) \mathbf{X}_{\text{PE}}^{(t)}, \qquad \textit{(Dynamic Self-Attention)}$$

where $\mathbf{W}_{\text{segment}}^{(s)} \in \mathbb{R}^{t_{\max} \times d}$, $\mathbf{W}_{\text{flat}}^{(s)^{(i)}} \in \mathbb{R}^{d|\mathcal{V}| \times |\mathcal{V}|}$, $\mathbf{W}_{\text{PE}} \in \mathbb{R}^{t_{\max} \times t_{\max}}$ are learnable parameters, $\|$ is concatenation, and SIGMOID$(.)$ is row-wise sigmoid normalization. Note that for different segments we use different dimensionality reduction matrices to reinforce the power of the Voxel Mixing.

Time Mixer. We then fuse information in the time dimension by using the Time Mixer submodule. To this end, the Time Mixer employs a 2-layer MLP with layer-normalization (Ba et al., 2016):

$$\mathbf{H}_{\textit{Time}}^{(t)} = \mathbf{H}_{\text{Voxel}}^{(t)} + \left(\sigma\left(\texttt{LayerNorm}\left(\mathbf{H}_{\text{Voxel}}^{(t)}\right)\mathbf{W}_{\text{Time}}^{(1)}\right)\mathbf{W}_{\text{Time}}^{(2)}\right) \in \mathbb{R}^{|\mathcal{V}|\times t_{\max}}, \tag{1}$$

where $\mathbf{W}_{\text{Time}}^{(1)}$ and $\mathbf{W}_{\text{Time}}^{(1)}$ are learnable matrices, $\sigma(.)$ is an activation function (we use GeLU (Hendrycks & Gimpel, 2020)), and `LayerNorm` is layer normalization (Ba et al., 2016).

3.2 Functional Connectivity Encoder

To encode the functional connectivity graph, we design an MLP-based architecture that learns both the structural and temporal properties of the graph. Inspired by the recent success of all-MLP architecture in graphs (Cong et al., 2023), we extend MLP-Mixer to temporal graphs. We first define patches in temporal graphs. While patches in images, videos, and multivariate timeseries can simply be non-overlapping regular grids, patches in graphs are overlapping non-grid structures, which makes the patching extraction challenging. He et al. (2023) suggest using graph partitioning algorithms to extract graph patches; however, these partitioning algorithms ① only consider structural properties, missing the temporal dependencies, and ② can be time-consuming, limiting the scalability to dense graphs like brain functional connectome. To this end, we propose a temporal-patch extraction algorithm such that nodes (voxels) in each patch share similar temporal and structural properties.

Temporal Patching. To extract temporal patches from the graph, we use a biased temporal random walk that walks over both nodes (voxels) and timestamps. Given a functional connectivity graph $\mathcal{G}_F = \{\mathcal{G}_F^{(t)}\}_{t=1}^{T}$, we sample M walks with length $m+1$ started from node (voxel) $v_0 \in \mathcal{V}$ like: $\mathcal{W}alk : (v_0, t_0) \rightarrow (v_1, t_1) \rightarrow \cdots \rightarrow (v_m, t_m)$, such that $(v_{i-1}, v_i) \in \mathcal{E}^{(t_i)}$, and $t_0 \geq t_1 \geq t_2 \geq \cdots \geq t_m$. Note that, contrary to some previous temporal random walks (Wang et al., 2021; Behrouz et al., 2023), we allow the walker to walk in the same timestamp at each step. While backtracking over time, we aim to capture temporal information and extract the dynamics of voxels' activity over *related* timestamps. Previous studies show that doing a task can affect brain activity even after 2 minutes (Yang et al., 2023a). To this end, since more recent connections can be more informative, we use a biased sampling procedure. Let v_{pre} be the previously sampled node, we use hyperparameters $\theta, \theta_0 \geq 0$ to sample a node v with probability proportional to $\exp\left(\theta(t - t_{\text{pre}} + \theta_0)\right)$, where t and t_{pre} are the timestamps that $(v_{\text{pre}}, v) \in \mathcal{E}^{(t)}$ and the timestamp of the previous sample, respectively. In this sampling procedure, smaller (resp. larger) θ means less (resp. more) emphasis on recent timestamps. Each walk started from v can be seen as a temporal subgraph, and so we let ρ_v be the union of all these subgraphs (walks started from v). We treat each of ρ_v as a temporal patch.

Temporal Pooling Mixer. Given the temporal graph patches that we extracted above, we need to encode each patch to obtain patch encodings (we later use these patch encodings as their corresponding voxel's encodings). While simple poolings (e.g., Sum(.)) are shown to miss information (Behrouz et al., 2023), more complicated pooling functions consider a static pooling rule. However, as discussed above, the effect of performing a task on the neuroimaging data might last for a period of time and the pooling rule might change over time. To this end, we design a temporal pooling, TPMixer(.), that dynamically pools a set of voxels in a patch based on their timestamps.

Given a patch $\rho_{v_0} = \{v_0, v_1, \ldots, v_k\}$, for each voxel we consider the correlation of its activity with other voxels' as its preliminary feature vector. That is, for each voxel v, we consider its feature vector in the time window t as $\mathcal{A}_v^{(t)}$, the v's corresponding row in $\mathcal{A}^{(t)}$. We abuse the notation and use $\mathcal{A}_{\rho_v}^{(t)}$ to refer to the set of $\mathcal{A}^{(t)}$'s rows corresponding to ρ_v. Since patch sizes are different, we zero pad $\mathcal{A}_{\rho_v}^{(t)}$ matrices to a fixed size. Note that this zero padding is important to capture the size of each voxel neighborhood. The voxel with more zero-padded dimensions in its patch has less correlation with others. To capture both cross-feature and cross-voxel dependencies, we can use the same architecture as the Time Mixer and Voxel-Mixer. However, the main drawback of this approach is that a pooling function is expected to be permutation invariant while the Voxel Mixer phase is permutation variant.

To address this challenge, we fuse information across features in a non-parametric manner as follows:

$$\mathbf{H}_{\text{F}}^{(t)} = \mathcal{A}_{\rho_v}^{(t)} + \sigma\left(\texttt{Softmax}\left(\texttt{LayerNorm}\left(\mathcal{A}_{\rho_v}^{(t)}\right)^\top\right)\right)^\top \in \mathbb{R}^{|\rho_v|\times d'}, \tag{2}$$

where $\sigma(.)$ is an activation function, $\texttt{Softmax}(.)$ is used to normalize across features to bind and fuse feature-wise information in a non-parametric manner, avoiding permutation variant operations, and d' is the feature vector size. To dynamically fuse information across voxels, we use the same idea as dynamic self-attention in §3.1 and learn dynamic matrices $\mathbf{P}_{\text{Pool}_i}$; let d_{patch} be the patch size:

$$\mathbf{P}_{\text{Pool}_i} = \textsc{Softmax}\left(\textsc{Flat}\left(\mathbf{H}_{\text{F}}^{(t)}\right)\mathbf{W}_{\text{Pool}}^{(i)}\right) \in \mathbb{R}^{1\times d'} \tag{3}$$

$$\mathbf{h}_{\rho_v} = \textsc{Mean}\left(\texttt{Norm}(\mathbf{H}_{\text{F}}^{(t)}) + \mathbf{H}_{\text{PE}}^{(t)}\ \textsc{Softmax}\left(\frac{\mathbf{H}_{\text{PE}}^{(t)\top}\mathbf{H}_{\text{PE}}^{(t)}}{\sqrt{d_{\text{patch}}}}\right)\right) \in \mathbb{R}^{1\times d'}, \tag{4}$$

where $\mathbf{H}_{\text{PE}}^{(t)} = \mathbf{H}_{\text{F}}^{(t)}\mathbf{P}_{\text{Pool}}$ is the transformation of $\mathbf{H}_{\text{F}}^{(t)}$ by dynamic matrix $\mathbf{P}_{\text{Pool}}$.

Theorem 1. TPMIXER *is permutation invariant and a universal approximator of multisets.*

Time Encoding. To distinguish different timestamps in the functional connectivity graph, we use a non-learnable time encoding module proposed by Cong et al. (2023). This encoding approach helps reduce the number of parameters, and also it has been shown to be more stable and generalizable (Cong et al., 2023). Given hyperparameters α, β, and d, we use feature vector $\boldsymbol{\omega} = \{\alpha^{-i/\beta}\}_{i=0}^{d-1}$ to encode each timestamp t using $\cos(\boldsymbol{\omega}t)$ function. Therefore, we obtain the time encoding as $\boldsymbol{\eta}_t = \cos(\boldsymbol{\omega}t)$.

Voxel-, Edge-, and Graph-level Encodings. Depending on the downstream task, we might obtain voxel-, edge-, or graph-level encodings. For each voxel $v \in \mathcal{V}$, we let $\mathcal{E}^{(t)}[\rho_v]$ be the set of connections in the patch of v. To obtain the voxel-level encoding of each voxel v, $\boldsymbol{\psi}_v$, we use patch encoding and concatenate it with all the weighted mean of timestamp encodings; i.e., $\boldsymbol{\psi}_v^t = \text{MLP}([\mathbf{h}_{\rho_v} \| \mathcal{T}_v])$, where $\mathcal{T}_v = \frac{\sum_{t_0=1}^{t}\mathcal{E}^{(t_0)}[\rho_v]\boldsymbol{\eta}_{t_0}}{\sum_{t_0=1}^{t}\mathcal{E}^{(t_0)}[\rho_v]}$. For a connection $e = (u, v) \in \mathcal{E}^{(t)}$, we obtain its encoding by concatenating its endpoints and its timestamp encodings; i.e., $\boldsymbol{\zeta}_{(u,v)}^{(t)} = \text{MLP}([\boldsymbol{\psi}_u^t, \boldsymbol{\psi}_v^t, \boldsymbol{\eta}_t])$. Finally, to obtain the graph level encoding, we use vanilla MLP-MIXER (Tolstikhin et al., 2021) on patch encodings; let $\boldsymbol{\Psi}^{(t)}$ be the matrix whose rows are $\boldsymbol{\psi}_v^{(t)}$:

$$\boldsymbol{\Psi}_{\text{patch}}^{(t)} = \boldsymbol{\Psi}^{(t)} + \mathbf{W}_{\text{patch}}^{(2)}\sigma\left(\mathbf{W}_{\text{patch}}^{(1)}\texttt{LayerNorm}\left(\boldsymbol{\Psi}^{(t)}\right)\right), \tag{5}$$

$$\textsc{Enc}(\mathcal{G}_F^{(t)}) = \textsc{Mean}\left(\boldsymbol{\Psi}_{\text{patch}}^{(t)} + \sigma\left(\texttt{LayerNorm}\left(\boldsymbol{\Psi}_{\text{patch}}^{(t)}\right)\mathbf{W}_{\text{channel}}^{(1)}\right)\mathbf{W}_{\text{channel}}^{(2)}\right). \tag{6}$$

Similar to the above, to obtain the brain-level encoding, $\mathbf{Z}_{\mathbf{V}}^{(t)}$, based on voxel acitivity timeseries, we use MLP-MIXER on $\mathbf{H}_{\textit{Time}}^{(t)}$.

3.3 SELF-SUPERVISED PRE-TRAINING

In §3.1 and §3.2 we obtained the encodings of the same contexts, from different perspectives. In this section, inspired by (Hjelm et al., 2019; Bachman et al., 2019), we use the mutual information of these two perspectives from the same context, to learn voxel- and brain-level encodings in a self-supervised manner. To this end, let $\boldsymbol{\Psi}$ be the voxel-level encodings obtained from functional connectome, $\mathbf{Z}_{\mathbf{F}}^{(t)} = \textsc{Enc}(\mathcal{G}_F^{(t)})$ be the global encoding (brain-level) of the functional connectome, $\mathbf{H}_{\text{Voxel}}^{(t)}$ be the voxel activity encodings from the brain activity timeseries, and $\mathbf{Z}_{\mathbf{V}}^{(t)}$ be the global encoding (brain-level) of the voxel activity timeseries, we aim to maximize $I(\mathbf{Z}_{\mathbf{V}}^{(t)}, \boldsymbol{\psi}_{v,i}^{(t)}) + I(\mathbf{Z}_{\mathbf{F}}^{(t)}, (\mathbf{H}_{\text{Time}}^{(t)})_{v,j})$ for all $v \in \mathcal{V}$ and possible i, j. Following previous studies (Bachman et al., 2019), we use Noise-Contrastive Estimation (NCE) (Gutmann & Hyvärinen, 2010) and minimize the following loss function:

$$\mathbb{E}_{(\mathbf{Z}_{\mathbf{F}}^{(t)}, \boldsymbol{\psi}_{v,i}^{(t)})}\left[\mathbb{E}_{\mathcal{N}}\left[\mathcal{L}_\Phi(\mathbf{Z}_{\mathbf{F}}^{(t)}, \boldsymbol{\psi}_{v,i}^{(t)}, \mathcal{N})\right]\right] + \mathbb{E}_{(\mathbf{Z}_{\mathbf{V}}^{(t)}, (\mathbf{H}_{\text{Voxel}}^{(t)})_{v,j})}\left[\mathbb{E}_{\mathcal{N}}\left[\mathcal{L}_\Phi(\mathbf{Z}_{\mathbf{V}}^{(t)}, (\mathbf{H}_{\text{Voxel}}^{(t)})_{v,j}, \mathcal{N})\right]\right], \tag{7}$$

where $\mathcal{N}$ is the set of negative samples, $(\mathbf{Z}_{\mathbf{V}}^{(t)}, \boldsymbol{\psi}_{v,i}^{(t)})$ and $(\mathbf{Z}_{\mathbf{F}}^{(t)}, (\mathbf{H}_{\text{Voxel}}^{(t)})_{v,j})$ are the positive sample pairs, and $\mathcal{L}_\Phi$ is a standard $\texttt{Log-Softmax}$.

Table 1: Performance on multi-class brain classification: Mean ACC (%) $\pm$ standard deviation.

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	$55.03_{\pm 1.35}$	$55.17_{\pm 1.74}$	$91.71_{\pm 1.48}$	$47.94_{\pm 1.15}$
BRAINMIXER	$\mathbf{67.24}_{\pm \mathbf{1.47}}$	$\mathbf{62.58}_{\pm \mathbf{1.12}}$	$\mathbf{96.32}_{\pm \mathbf{0.29}}$	$\mathbf{57.83}_{\pm \mathbf{1.03}}$

Data Augmentation & Negative Samples. MLP-MIXER-based architectures are known to have the potenial of overfitting (Liu et al., 2021). To mitigate this, we perform data augmentation. For $\mathcal{G}_F^{(t)} = (\mathcal{V}, \mathcal{E}^{(t)})$, in patch extraction, we randomly mask p connections and then we sample temporal walks to generate new patches. Note that, at the end, each patch is an induced subgraph and might include masked connections as well. Furthermore, to generate negative samples: ① To corrupt the functional connectivity, we randomly change one endpoint of a subset of connections. ② To corrupt the timeseries, we follow existing studies (Yue et al., 2022; Woo et al., 2022) on timeseries and replace a brain signal in time window t with another signal that is randomly selected from the batch. Given a pre-trained model $\mathcal{M}$, for different downstream tasks in a semi-supervised setting, we fine-tune $\mathcal{M}$ using a small subset of labeled training data. Also, for each voxel, we concatenate its encodings from VA and FC Encoders.

4 EXPERIMENTS

Dataset. We use six real-world datasets: ① We present BVFC, a task-based fMRI large-scale dataset that includes voxel activity timeseries and functional connectivity of 3 subjects when looking at the 8460 images from 720 categories. This data is based on THINGS dataset (Hebart et al., 2023). ②BVFC-MEG is the MEG counterpart of BVFC. ③ ADHD (Milham et al., 2011) contains data for 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) consists of EEG data (31 channels) of 642 subjects. ⑤ ASD (Craddock et al., 2013) contains data for 45 subjects in the ASD group and 45 subjects in the TD control group. ⑥ 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.

Evaluation Tasks. In our experiments we focus on 4 downstream tasks: ① Edge-Anomaly Detection (AD), ② Voxel AD, ③ Brain AD, and ④ Brain Classification. For the edge and voxel AD tasks, we follow previous studies (Behrouz & Seltzer, 2023a; Ma et al., 2021), and inject 1% and 5% anomalous edges into the functional connectivity in the control group. For brain AD all datasets has ground-truth anomalies (see Appendix E.2). The ground truth anomalies in BVFC are the brain responses to not recognizable images, generated by BigGAN (Brock et al., 2019), and for other datasets are brain activity of people living with ADHD, seizure, and ASD. For brain classification, we focus on the prediction of ⓘ categories of images seen by the subjects (in BVFC, and BVFC-MEG), and ⓘⓘ age prediction and mental state decoding (in HCP-Age, and HCP-Mental). In all experiments, we perform statistical comparison with baselines via paired t-tests and shade significance results with blue and others with gray. The details of the setup is in Appendix E.

Baselines. For anomaly detection and graph classification tasks, we compare BRAINMIXER with state-of-the-art time series, graph, and brain anomaly detection and learning models: ① Graph-based methods: GOutlier (Aggarwal et al., 2011), NETWALK (Yu et al., 2018), HYPERSAGCN (Zhang et al., 2020), 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 & Seltzer, 2023b), and BNTRANSFORMER (Kan et al., 2022b), PTGB (Yang et al., 2023b). ③ Time-series-based methods: USAD (Audibert et al., 2020), Time Series Transformer (TST) (Zerveas et al., 2021), and MVTS (Potter et al., 2022). We may exclude some baselines in some tasks as they cannot be applied in that setting. We

Table 2: Performance on anomaly detection: Mean AUC-PR (%) ± standard deviation†.

	Methods	BVFC	BVFC-MEG	HCP		ADHD		TUH-EEG		ASD	
	Anomaly %			1%	5 %	1%	5 %	1%	5 %	1%	5 %
Edge-level AD	GOUTLIER	65.12±2.97	59.45±2.61	62.47±1.15	61.83±1.28	65.37±0.93	64.70±2.09	65.61±1.82	64.12±0.97	60.85±0.97	59.13±1.86
	NETWALK	71.67±1.56	62.75±1.16	73.12±1.25	72.19±1.31	70.29±2.15	69.86±2.58	71.14±1.36	70.27±1.42	69.07±2.20	68.52±2.55
	HYPERSAGCN	80.17±1.59	70.83±1.27	82.94±1.14	81.98±1.58	84.22±1.61	83.96±1.47	73.99±0.83	72.65±0.97	73.26±1.08	73.18±0.92
	GRAPHMIXER	87.13±0.99	75.91±1.59	86.87±1.96	86.19±1.48	85.12±1.46	84.86±1.58	75.93±0.95	75.12±1.08	84.91±2.27	83.52±2.03
	BRAINNETCNN	80.92±1.18	71.54±2.07	80.79±1.23	79.44±1.18	80.58±1.62	79.95±2.01	73.06±1.74	72.87±1.31	72.68±2.12	72.01±1.45
	BRAINGNN	81.96±1.76	72.68±1.13	82.15±1.84	81.38±1.61	79.02±1.85	78.64±1.43	72.96±1.58	71.73±1.14	72.14±1.25	71.82±1.73
	FBNETGEN	81.58±1.92	72.66±1.52	82.05±1.19	81.53±1.82	79.89±1.63	78.97±1.84	73.04±1.53	72.56±1.33	72.51±1.28	71.62±1.82
	ADMIRE	87.12±1.61	75.91±1.43	87.01±1.27	86.38±1.17	86.23±1.74	85.18±2.21	76.68±1.82	75.14±1.67	86.52±1.72	85.44±1.49
	PTGB	86.52±1.64	75.93±1.71	86.83±1.59	86.00±1.28	86.14±1.15	85.22±1.21	75.98±1.16	74.92±1.08	86.18±1.58	85.72±1.05
	BNTRANSFORMER	86.61±1.72	75.82±1.18	86.22±1.77	85.15±1.12	85.83±1.97	85.14±1.67	75.91±1.72	75.24±1.53	74.92±1.18	74.11±1.37
	BRAINMIXER	**91.62±1.36**	**82.58±1.92**	**90.14±1.72**	**90.02±1.49**	**91.74±0.93**	**91.48±1.41**	**80.91±1.19**	**80.85±1.62**	**90.44±1.57**	**90.27±1.39**
Voxel-level AD	USAD	68.27±1.16	62.73±1.27	65.49±1.31	65.01±1.18	72.79±1.48	72.19±0.94	72.81±1.42	71.36±1.03	66.28±1.16	65.17±1.15
	TST	70.62±1.48	68.57±1.81	69.18±1.64	69.11±1.32	74.81±1.14	73.99±1.47	73.71±1.55	73.03±1.47	69.23±1.82	68.94±1.73
	MVTS	N/A	N/A	N/A	N/A	N/A	N/A	77.48±1.81	77.02±1.29	N/A	N/A
	GOUTLIER	64.66±2.38	60.17±1.25	63.59±1.62	63.07±1.52	68.97±1.16	67.12±0.93	65.18±1.09	65.01±1.57	59.67±1.42	58.49±1.35
	NETWALK	68.73±1.16	63.61±1.31	66.98±1.44	66.04±1.63	75.16±1.23	74.73±1.01	72.21±0.91	71.62±1.46	71.28±1.17	71.02±1.49
	HYPERSAGCN	78.84±1.22	71.62±1.96	80.74±1.51	79.18±1.83	83.94±1.13	83.01±0.92	75.62±1.12	74.83±0.78	74.93±1.47	74.15±1.19
	GRAPHMIXER	76.94±1.68	71.44±1.39	81.55±1.82	81.07±1.27	81.37±1.09	80.83±1.16	72.95±1.26	72.01±0.82	72.49±1.28	72.27±1.69
	BRAINNETCNN	80.17±1.49	73.91±1.54	82.75±1.27	82.21±1.73	82.79±1.08	81.12±1.16	73.98±1.24	73.01±1.08	73.18±0.95	72.88±1.04
	BRAINGNN	79.92±1.63	73.25±1.94	82.99±1.65	82.13±1.66	81.14±1.05	80.83±0.87	73.06±1.14	72.74±0.86	72.54±1.38	71.12±1.19
	FBNETGEN	79.17±2.04	72.35±1.84	82.26±1.37	81.62±1.49	80.91±1.12	80.94±1.74	72.53±1.48	72.06±1.29	72.11±1.94	71.28±1.22
	PTGB	85.18±1.83	76.16±1.08	85.72±1.14	84.95±1.33	86.43±1.16	86.36±1.15	77.54±1.37	77.32±1.21	77.92±1.26	77.76±1.25
	BNTRANSFORMER	85.19±1.23	75.67±1.14	85.02±0.96	84.36±1.59	86.13±1.21	86.11±1.82	77.96±1.32	77.08±1.06	76.05±1.52	75.72±1.18
	BRAINMIXER	**90.14±1.57**	**81.52±1.32**	**89.27±1.61**	**88.94±1.24**	**89.97±1.14**	**89.81±1.27**	**79.45±1.19**	**79.23±0.94**	**89.51±1.78**	**89.24±1.59**
Brain-level AD	USAD	71.93±1.15	61.32±1.71	67.79±2.28	67.36±2.61	82.87±2.03	80.52±1.84	72.03±1.17	71.48±1.05	71.62±1.58	70.98±1.41
	TST	72.47±1.23	67.12±2.07	67.94±1.69	67.22±1.17	83.54±1.38	83.04±1.12	72.96±1.39	72.11±1.58	72.76±1.71	72.04±1.56
	MVTS	N/A	N/A	N/A	N/A	N/A	N/A	83.53±1.91	82.41±1.02	N/A	N/A
	NETWALK	72.16±1.44	69.57±1.73	69.14±1.49	68.66±1.52	83.11±1.02	82.81±1.61	71.06±1.05	69.94±1.12	72.85±1.17	72.21±1.34
	HYPERSAGCN	80.25±1.15	76.91±1.18	72.26±1.47	72.01±1.21	86.94±1.63	86.17±1.49	75.31±0.85	74.79±1.09	76.72±1.32	75.81±1.58
	GMM	81.79±1.24	77.84±1.52	74.87±1.58	74.02±1.10	85.89±0.98	85.03±1.18	76.62±1.17	76.11±1.26	76.37±1.83	75.68±1.59
	GRAPHMIXER	82.56±1.19	77.91±1.26	75.03±1.72	74.46±1.53	86.02±1.15	85.64±1.09	77.49±1.09	76.63±1.22	76.82±1.84	76.18±1.80
	BRAINNETCNN	78.47±1.18	73.12±1.27	70.73±1.77	70.12±1.86	85.84±0.96	85.07±1.52	73.92±0.97	73.07±1.51	75.96±1.66	75.03±1.28
	BRAINGNN	79.81±1.57	75.28±1.61	72.98±1.55	72.41±1.16	84.59±1.26	83.72±1.35	72.41±1.38	71.55±1.16	75.12±1.33	74.57±1.52
	FBNETGEN	78.94±1.24	74.49±1.33	71.62±1.53	71.06±1.48	84.67±1.26	84.08±1.37	72.69±1.18	71.87±1.12	75.34±1.21	74.73±1.39
	ADMIRE	83.72±1.18	78.83±1.56	75.52±1.81	74.59±1.12	86.27±1.72	85.18±1.56	78.12±1.47	77.59±1.68	77.18±1.61	76.33±1.45
	PTGB	84.08±1.35	79.68±1.62	76.01±1.07	75.13±1.48	87.59±1.12	86.99±0.96	79.17±1.36	78.64±1.55	80.56±1.29	80.04±1.16
	BNTRANSFORMER	83.86±1.52	79.03±1.78	75.64±1.82	75.09±1.18	87.54±1.04	86.92±1.48	79.36±1.71	78.08±1.16	77.19±2.01	76.58±1.73
	BRAINMIXER	**88.13±1.27**	**84.59±1.70**	**80.67±1.13**	**80.49±1.07**	**91.38±0.94**	**90.98±1.02**	**85.74±1.16**	**85.63±1.23**	**89.14±1.54**	**88.99±1.15**

† We perform statistical comparison with baselines via paired t-tests. Shaded blue indicates significance improvement over the baselines (p-value ≤ 0.05), while gray shaded boxes indicate (p-value > 0.05). The maximum p-value is 0.058.

Table 3: Ablation study on BRAINMIXER. AUC-PR scores on edge AD and ACC on classification.

Methods	BVFC		BVFC-MEG		HCP		ADHD	
	Edge AD	Classification	Edge AD	Classification	Edge AD	Classification	Edge AD	Classification
BRAINMIXER	**91.62±1.36**	**67.24±1.47**	**82.58±1.92**	**62.68±1.12**	**90.02±1.49**	**96.32±0.29**	**91.48±1.41**	**90.98±1.02**
Without Pre-training	88.75±2.16	63.58±2.09	80.21±1.63	61.02±1.37	88.14±1.29	93.81±0.92	90.18±1.13	89.27±1.06
Without VA Encoder	87.99±2.04	59.14±4.51	78.52±2.18	60.53±1.83	86.97±2.05	92.41±1.24	88.29±1.41	88.76±1.19
Replace VA Encoder with TST	89.02±1.18	62.89±1.49	80.46±2.00	61.78±1.24	89.01±0.86	93.53±1.78	90.06±1.55	88.94±1.98
Without FC Encoder	84.27±4.37	65.82±2.18	77.09±3.41	59.73±1.12	85.59±2.47	91.64±1.58	86.97±1.16	87.62±2.16
Replace FC Encoder with BNTRANSFORMER	87.18±2.03	66.12±1.27	78.85±1.36	62.01±0.87	86.76±1.44	94.24±1.25	88.03±1.24	88.81±0.98
Without Functional Patching	86.35±2.97	60.42±3.53	77.21±1.93	60.28±1.72	86.14±3.09	91.97±1.88	87.51±1.86	88.25±2.53
Replace Functional Patching with Partitioning	88.56±1.42	66.50±1.92	79.26±1.51	60.63±1.87	87.55±1.29	96.14±1.04	90.10±1.78	89.69±1.46
Replace TPMIXER by MEAN(.)	88.51±1.03	63.38±1.48	78.94±1.85	60.91±2.01	87.52±1.91	93.31±1.73	89.04±0.95	89.11±1.52
Static Self-Attention	88.39±1.40	63.01±2.10	78.63±1.97	60.78±1.64	87.04±1.53	92.95±1.49	88.96±1.22	88.83±2.07
Remove Time Encoding	89.58±0.81	66.14±1.52	79.91±1.75	61.19±1.36	88.82±2.07	94.12±1.92	90.57±0.91	89.99±1.04
fix $\theta = 0$	83.60±4.52	59.33±2.58	75.96±2.05	59.11±1.46	85.39±1.52	90.51±1.38	86.24±2.01	87.18±1.94

use the same training, hyperparameter tuning, and testing procedure as BRAINMIXER. The details of baselines can be found in Appendix E.1.

Brain Classification. Table 1 reports the performance of BRAINMIXER and baselines on multi-class brain classification tasks. BRAINMIXER achieves the best accuracy on all datasets with 14.3% average improvement (20.3% best improvement) over the best baseline. There are three main reasons for BRAINMIXER's superior performance: ① While the time series-based model only uses voxel activity timeseries, and graph-based methods only use functional connectivity graph, BRAINMIXER takes advantage of both and learns the brain representation at different levels of granularity, which can provide complementary information. ② Static methods (e.g., PTGB, BRAINGNN, etc.), miss the dynamics of brain activity, while BRAINMIXER employs a time encoding module to learn temporal properties. ③ Compared to graph learning methods (e.g., GMM, GRAPHMIXER, etc.), BRAINMIXER is specifically designed for the brain, taking advantage of its special properties.

Anomaly Detection. Table 2 reports the performance of BRAINMIXER and baselines on anomaly detection tasks at different scales: i.e., edge-, voxel-, and brain-level. BRAINMIXER achieves the best AUC-PR on all datasets with 6.2%, 5.7%, 4.81% average improvement over the best baseline in edge AD, voxel AD, and brain AD, respectively. The main reasons for this superior performance are as above. Note that brain-level anomaly detection can also be seen as a brain classification task. However, here, based on the nature of the data, we separate these two tasks.

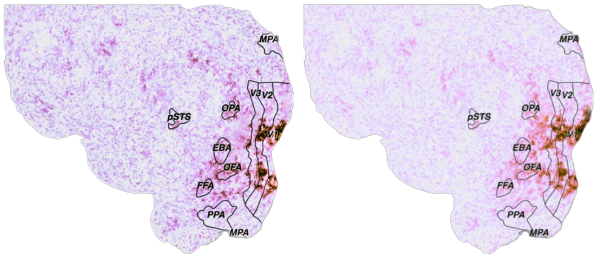


Figure 2: Distribution of abnormal voxel activities detected by BRAINMIXER in the visual cortex when seeing (Left) GAN-generated, (Right) Normal image.

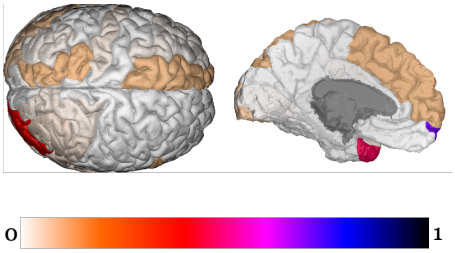


Figure 3: The distribution of detected abnormal voxels by BRAINMIXER in condition ADHD group.

Ablation Study. We next conduct ablation studies on the BVFC, BVFC-MEG, HCP, and ADHD datasets to validate the effectiveness of BRAINMIXER's critical components. Table 3 shows AUC-PR for edge AD and accuracy for classification tasks. The first row reports the performance of the complete BRAINMIXER implementation with pre-training. Each subsequent row shows results for BRAINMIXER with one module modification: row 2 removes the pre-training phase, row 3 removes the VA Encoder module, row 4 replace VA Encoder with TST, row 5 removes FC Encoder module, row 6 replace it with BNTRANSFORMER, row 7 replaces functional patching with random patching, row 8 replace temporal patching with partitioning (Karypis & Kumar, 1998), row 9 replaces TPMIXER with MEAN(.) pooling, row 10 replaces dynamic with static self-attention, row 11 removes time encoder, the last row set $\theta = 0$, removing biased in the sampling. These results show that each component is critical for achieving BRAINMIXER's superior performance. The greatest contribution comes from biased sampling, VA and FC encoders, functional patching, and dynamic self-attention.

Parameter Sensitivity. We discuss the effect of the number of walks, M, the walk length, m, and time decay, θ on the performance in Appendix F. Results show that increasing the number of walks results in better performance as each patch is a better representation of the node's neighborhood. The effect of the walk length on performance peaks at a certain point, but the exact value varies with datasets. In Appendix F, we further discuss how aggregating timeseries to obtain beta weights and aggregating voxels to obtain ROIs can affect performance.

How Does BRAINMIXER Detect GAN Generated Images? The visual cortex, responsible for processing visual information, is hierarchically organized with multiple layers building upon simpler features at lower stages (Van Essen & Maunsell, 1983). Initially, neurons detect edges and colors, but on deeper levels, they specialize in recognizing more complex patterns and objects. Figure 2 (Left) (resp. (Right)) reports the distribution of detected voxel activity by BRAINMIXER when the subject looking at non-recognizable images (resp. natural images). Interestingly, while the distributions share similar patterns in lower levels (e.g., V1 and V2 voxels), higher-level voxels (e.g., V3) are less active when the subject sees non-recognizable images. These results show the potential of BRAINMIXER in learning meaningful representation of voxels activity. Additional details can be found in Appendix F.

Case Study: ADHD In this case study, we train our model on the neuroimages of the typically developed group and test it on the ADHD condition group to detect abnormal voxel activities that might be correlated to ADHD symptoms. Figure 3 reports the distribution of anomalous voxels within the brain of the ADHD group. 78% of all found abnormal voxel activities by BRAINMIXER are located in the Frontal Pole, Left and Right Temporal Poles, and Lingual Gyrus. Surprisingly, these findings are consistent with previous studies on ADHD, which use diffusion tensor imaging (Lei et al., 2014) and Forman–Ricci curvature changes (Chatterjee et al., 2021).

5 CONCLUSION

In this work, we present an unsupervised pre-training framework, BRAINMIXER, that bridges the representation learning of voxel activity and functional connectivity by maximizing their mutual information. The experimental results show the potential of BRAINMIXER in ① detecting abnormal brain activity that might cause a brain disease/disorder, ② disease/disorder detection, and ③ understanding object representation in the brain. We discuss potential limitations and future work in Appendix G.

REFERENCES

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

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*, pp. 3395–3404, 2020.

Lev Kiar Avberšek and Grega Repovš. Deep learning in neuroimaging data analysis: applications, challenges, and solutions. *Frontiers in neuroimaging*, 1:981642, 2022.

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

Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/ddf354219aac374f1d40b7e760ee5bb7-Paper.pdf.

Hubert Banville, Omar Chehab, Aapo Hyvärinen, Denis-Alexander Engemann, and Alexandre Gramfort. Uncovering the structure of clinical eeg signals with self-supervised learning. *Journal of Neural Engineering*, 18(4):046020, 2021.

Ali Behrouz and Margo Seltzer. Anomaly detection in multiplex dynamic networks: from blockchain security to brain disease prediction. In *NeurIPS 2022 Temporal Graph Learning Workshop*, 2022. URL https://openreview.net/forum?id=UDGZDfwmay.

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, Farnoosh Hashemi, Sadaf Sadeghian, and Margo Seltzer. CAT-walk: Inductive hypergraph learning via set walks. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=QG4nJBNEar.

Signe L Bray, Catie Chang, and Fumiko Hoeft. Applications of multivariate pattern classification analyses in developmental neuroimaging of healthy and clinical populations. *Frontiers in human neuroscience*, 3:898, 2009.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=B1xsqj09Fm.

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, pp. 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.

Vince D. Calhoun, Robyn Miller, Godfrey Pearlson, and Tulay Adalı. The chronnectome: Time-varying connectivity networks as the next frontier in fmri data discovery. *Neuron*, 84(2):262–274, 2014. ISSN 0896-6273. doi: https://doi.org/10.1016/j.neuron.2014.10.015. URL https://www.sciencedirect.com/science/article/pii/S0896627314009131.

Martin J Chadwick, Heidi M Bonnici, and Eleanor A Maguire. Decoding information in the human hippocampus: a user's guide. *Neuropsychologia*, 50(13):3107–3121, 2012.

Benjamin Paul Chamberlain, Sergey Shirobokov, Emanuele Rossi, Fabrizio Frasca, Thomas Markovich, Nils Yannick Hammerla, Michael M. Bronstein, and Max Hansmire. Graph neural networks for link prediction with subgraph sketching. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=m1oqEOAozQU.

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.

Gang Chen, B Douglas Ward, Chunming Xie, Wenjun Li, Zhilin Wu, Jennifer L Jones, Malgorzata Franczak, Piero Antuono, and Shi-Jiang Li. Classification of alzheimer disease, mild cognitive impairment, and normal cognitive status with large-scale network analysis based on resting-state functional mr imaging. *Radiology*, 259(1):213, 2011.

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.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

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.

Aurelio Cortese, Saori C Tanaka, Kaoru Amano, Ai Koizumi, Hakwan Lau, Yuka Sasaki, Kazuhisa Shibata, Vincent Taschereau-Dumouchel, Takeo Watanabe, and Mitsuo Kawato. The decnef collection, fmri data from closed-loop decoded neurofeedback experiments. *Scientific data*, 8(1): 65, 2021.

Cameron Craddock, Yassine Benhajali, Carlton Chu, Francois Chouinard, Alan Evans, András Jakab, Budhachandra Singh Khundrakpam, John David Lewis, Qingyang Li, Michael Milham, et al. The neuro bureau preprocessing initiative: open sharing of preprocessed neuroimaging data and derivatives. *Frontiers in Neuroinformatics*, 7:27, 2013.

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)*, 2022a.

Hejie Cui, Wei Dai, Yanqiao Zhu, Xiaoxiao Li, Lifang He, and Carl Yang. Interpretable graph neural networks for connectome-based brain disorder analysis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 375–385. Springer, 2022b.

Fernando Lopes da Silva. Neural mechanisms underlying brain waves: from neural membranes to networks. *Electroencephalography and clinical neurophysiology*, 79(2):81–93, 1991.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Yuhui Du, Zening Fu, and Vince D Calhoun. Classification and prediction of brain disorders using functional connectivity: promising but challenging. *Frontiers in neuroscience*, 12:525, 2018.

James S. Gao, Alexander G. Huth, Mark D. Lescroart, and Jack L. Gallant. Pycortex: an interactive surface visualizer for fmri. *Frontiers in Neuroinformatics*, 9, 2015. ISSN 1662-5196. doi: 10.3389/fninf.2015.00023. URL https://www.frontiersin.org/articles/10.3389/fninf.2015.00023.

Eloy Geenjaar, Amrit Kashyap, Noah Lewis, Robyn Miller, and Vince Calhoun. Spatio-temporally separable non-linear latent factor learning: an application to somatomotor cortex fmri data. *arXiv preprint arXiv:2205.13640*, 2022.

Javier Gonzalez-Castillo and Peter A Bandettini. Task-based dynamic functional connectivity: Recent findings and open questions. *Neuroimage*, 180:526–533, 2018.

Michael Greicius. Resting-state functional connectivity in neuropsychiatric disorders. *Current opinion in neurology*, 21(4):424–430, 2008.

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*, pp. 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*, pp. 12724–12745. PMLR, 2023.

Martin N. Hebart, Adam H. Dickter, Alexis Kidder, Wan Y. Kwok, Anna Corriveau, Caitlin Van Wicklin, and Chris I. Baker. Things: A database of 1,854 object concepts and more than 26,000 naturalistic object images. *PLOS ONE*, 14(10):1–24, 10 2019. doi: 10.1371/journal.pone.0223792. URL https://doi.org/10.1371/journal.pone.0223792.

Martin N Hebart, Oliver Contier, Lina Teichmann, Adam H Rockter, Charles Y Zheng, Alexis Kidder, Anna Corriveau, Maryam Vaziri-Pashkam, and Chris I Baker. Things-data, a multimodal collection of large-scale datasets for investigating object representations in human brain and behavior. *Elife*, 12:e82580, 2023.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus), 2020.

Leanna M Hernandez, Jeffrey D Rudie, Shulamite A Green, Susan Bookheimer, and Mirella Dapretto. Neural signatures of autism spectrum disorders: insights into brain network dynamics. *Neuropsychopharmacology*, 40(1):171–189, 2015.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bklr3j0cKX.

Tomoyasu Horikawa and Yukiyasu Kamitani. Generic decoding of seen and imagined objects using hierarchical visual features. *Nature communications*, 8(1):15037, 2017.

Yang Hu, Haoxuan You, Zhecan Wang, Zhicheng Wang, Erjin Zhou, and Yue Gao. Graph-mlp: Node classification without message passing in graph. *arXiv preprint arXiv:2106.04051*, 2021.

R Matthew Hutchison, Thilo Womelsdorf, Elena A Allen, Peter A Bandettini, Vince D Calhoun, Maurizio Corbetta, Stefania Della Penna, Jeff H Duyn, Gary H Glover, Javier Gonzalez-Castillo, et al. Dynamic functional connectivity: promise, issues, and interpretations. *Neuroimage*, 80: 360–378, 2013.

Yuli Jiang, Yu Rong, Hong Cheng, Xin Huang, Kangfei Zhao, and Junzhou Huang. Query driven-graph neural networks for community search: From non-attributed, attributed, to interactive attributed, 2021. URL https://arxiv.org/abs/2104.03583.

Biao Jie, Mingxia Liu, Xi Jiang, and Daoqiang Zhang. Sub-network based kernels for brain network classification. In *Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pp. 622–629, 2016.

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*, pp. 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 (eds.), *Advances in Neural Information Processing Systems*, 2022b. URL https://openreview.net/forum?id=1cJ1cbA6NLN.

George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, 1998. doi: 10.1137/S1064827595287997. URL https://doi.org/10.1137/S1064827595287997.

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.

Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41, 2022.

Demetres Kostas, Stephane Aroca-Ouellette, and Frank Rudzicz. Bendr: using transformers and a contrastive self-supervised learning task to learn from massive amounts of eeg data. *Frontiers in Human Neuroscience*, 15:653659, 2021.

Tommaso Lanciano, Francesco Bonchi, and Aristides Gionis. Explainable classification of brain networks via contrast subgraphs. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery; Data Mining*, KDD '20, pp. 3308–3318, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450379984. doi: 10.1145/3394486.3403383. URL https://doi.org/10.1145/3394486.3403383.

Sue-Hyun Lee and Chris I Baker. Multi-voxel decoding and the topography of maintained information during visual working memory. *Frontiers in systems neuroscience*, 10:2, 2016.

Du Lei, Jun Ma, Xiaoxia Du, Guohua Shen, Xingming Jin, and Qiyong Gong. Microstructural abnormalities in the combined and inattentive subtypes of attention deficit hyperactivity disorder: a diffusion tensor imaging study. *Scientific reports*, 4(1):6875, 2014.

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.

Hanxiao Liu, Zihang Dai, David So, and Quoc V Le. Pay attention to mlps. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 9204–9215. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/4cc05b35c2f937c5bd9e7d41d3686fff-Paper.pdf.

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*, pp. 1–1, 2021. doi: 10.1109/TKDE.2021.3118815.

Abdelhak Mahmoudi, Sylvain Takerkart, Fakhita Regragui, Driss Boussaoud, Andrea Brovelli, et al. Multivoxel pattern analysis for fmri data: a review. *Computational and mathematical methods in medicine*, 2012, 2012.

Razvan Marinescu, Arman Eshaghi, Daniel Alexander, and Polina Golland. Brainpainter: A software for the visualisation of brain structures, biomarkers and associated pathological processes. *arXiv preprint arXiv:1905.08627*, 2019.

Chris McNorgan, Gregory J Smith, and Erica S Edwards. Integrating functional connectivity and mvpa through a multiple constraint network analysis. *Neuroimage*, 208:116412, 2020.

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.

Tom M Mitchell, Rebecca Hutchinson, Marcel Adam Just, Radu S Niculescu, Francisco Pereira, and Xuerui Wang. Classifying instantaneous cognitive states from fmri data. In *AMIA annual symposium proceedings*, volume 2003, pp. 465. American Medical Informatics Association, 2003.

Mostafa Neo Mohsenvand, Mohammad Rasool Izadi, and Pattie Maes. Contrastive representation learning for electroencephalogram classification. In *Machine Learning for Health*, pp. 238–253. PMLR, 2020.

Alfonso Nieto-Castanon. Brain-wide connectome inferences using functional connectivity multivariate pattern analyses (fc-mvpa). *PLoS Computational Biology*, 18(11):e1010634, 2022.

Youngser Park, C Priebe, D Marchette, and Abdou Youssef. Anomaly detection using scan statistics on time series hypergraphs. In *Link Analysis, Counterterrorism and Security (LACTS) Conference*, pp. 9. SIAM Pennsylvania, 2009.

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)*, pp. 1304–1311. IEEE, 2022.

Raimon HR Pruim, Maarten Mennes, Daan van Rooij, Alberto Llera, Jan K Buitelaar, and Christian F Beckmann. Ica-aroma: A robust ica-based strategy for removing motion artifacts from fmri data. *Neuroimage*, 112:267–277, 2015.

Tiffany D Rogers, Eric McKimm, Price E Dickson, Dan Goldowitz, Charles D Blaha, and Guy Mittleman. Is autism a disease of the cerebellum? an integration of clinical and pre-clinical research. *Front Syst Neurosci*, 7:15, May 2013.

Zvi N. Roth and Elisha P. Merriam. Representations in human primary visual cortex drift over time. *Nature Communications*, 14(1):4422, Jul 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-40144-w. URL https://doi.org/10.1038/s41467-023-40144-w.

Zvi N. Roth, Kendrick Kay, and Elisha P. Merriam. Natural scene sampling reveals reliable coarse-scale orientation tuning in human v1. *Nature Communications*, 13(1):6469, Oct 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-34134-7. URL https://doi.org/10.1038/s41467-022-34134-7.

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.

Ramit Sawhney, Shivam Agarwal, Arnav Wadhwa, Tyler Derr, and Rajiv Ratn Shah. Stock selection via spatiotemporal hypergraph attention network: A learning to rank approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 497–504, 2021.

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*, pp. 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.

B Sundermann, D Herr, W Schwindt, and B Pfleiderer. Multivariate classification of blood oxygen level–dependent fmri data with diagnostic intention: a clinical perspective. *American Journal of neuroradiology*, 35(5):848–855, 2014.

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 (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=EI2KOXKdnP.

Asher Trockman and J Zico Kolter. Patches are all you need? *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=rAnB7JSMXL. Featured Certification.

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.

Martijn P Van Den Heuvel and Hilleke E Hulshoff Pol. Exploring the brain network: a review on resting-state fmri functional connectivity. *European neuropsychopharmacology*, 20(8):519–534, 2010.

David C Van Essen and John HR Maunsell. Hierarchical organization and functional streams in the visual cortex. *Trends in neurosciences*, 6:370–375, 1983.

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.

Eliana Vassena, James Deraeve, and William H. Alexander. Surprise, value and control in anterior cingulate cortex during speeded decision-making. *Nature Human Behaviour*, 4(4):412–422, Apr 2020. ISSN 2397-3374. doi: 10.1038/s41562-019-0801-5. URL https://doi.org/10.1038/s41562-019-0801-5.

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.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rklz9iAcKQ.

Yanbang Wang, Yen-Yu Chang, Yunyu Liu, Jure Leskovec, and Pan Li. Inductive representation learning in temporal networks via causal anonymous walks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=KYPz4YsCPj.

Chong-Yaw Wee, Pew-Thian Yap, Wenbin Li, Kevin Denny, Jeffrey N Browndyke, Guy G Potter, Kathleen A Welsh-Bohmer, Lihong Wang, and Dinggang Shen. Enriched white matter connectivity networks for accurate identification of mci patients. *Neuroimage*, 54(3):1812–1822, 2011.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. CoST: Contrastive learning of disentangled seasonal-trend representations for time series forecasting. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=PilZY3omXV2.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34:22419–22430, 2021.

Xingqian Xu, Zhangyang Wang, Gong Zhang, Kai Wang, and Humphrey Shi. Versatile diffusion: Text, images and variations all in one diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7754–7765, 2023.

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

Huzheng Yang, James Gee, and Jianbo Shi. Memory encoding model, 2023a.

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

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, pp. 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.

Zhizhang Yuan, Daoze Zhang, Yang Yang, Junru Chen, and Yafeng Li. PPi: Pretraining brain signal model for patient-independent seizure detection. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=tEmFyqjaJh.

Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8980–8987, 2022.

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*, pp. 2114–2124, 2021.

Ruochi Zhang, Yuesong Zou, and Jian Ma. Hyper-sagnn: a self-attention based graph neural network for hypergraphs. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ryeHuJBtPH.

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, pp. 11106–11115, 2021.

Tian Zhou, Ziqing Ma, Qingsong Wen, Liang Sun, Tao Yao, Wotao Yin, Rong Jin, et al. Film: Frequency improved legendre memory model for long-term time series forecasting. *Advances in Neural Information Processing Systems*, 35:12677–12690, 2022a.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*, pp. 27268–27286. PMLR, 2022b.

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)*, pp. 272–276. IEEE, 2022.

Appendix

Table of Contents

A Backgrounds **18**
A.1 Graphs and Machine Learning on Graphs . . . 18
A.2 Voxel Time Series Activity . . . 18
A.3 Brain Functional Connectivity . . . 18
A.4 MLP-MIXER . . . 19

B BVFC Dataset **20**
B.1 THINGS Dataset . . . 20
B.2 Preprocessing . . . 21
B.3 Image Classification . . . 21
B.4 Anomaly Detection . . . 21

C Additional Related Work **22**
C.1 Our Contributions . . . 22

D Theoretical Guarantee of TSETMIXER Performance **23**

E Experimental Setup **24**
E.1 Baselines . . . 24
E.2 Datasets . . . 26
E.3 Implementation, Training, and Hyperparameters Tuning . . . 27
E.4 Injecting Synthetic Anomalies . . . 28
E.5 Visualization Tools . . . 28

F Additional Experimental Results **28**
F.1 Parameter Sensitivity . . . 28
F.2 The Effect of Functional and Temporal Patching . . . 31
F.3 Performance Comparison using Different Metrics . . . 33
F.4 Graph Regression . . . 34
F.5 Qualitative Results . . . 34
F.6 How Does BRAINMIXER Detect GAN Generated Images? . . . 35
F.7 Case Study: ASD . . . 35
F.8 The Effect of Objective . . . 36
F.9 Number of Parameters . . . 36
F.10 Effect of the Number of Parameters on Accuracy . . . 36

G Limitations and Future Work **36**

A BACKGROUNDS

We begin by reviewing the preliminaries and background concepts that we refer to in the main paper.

A.1 GRAPHS AND MACHINE LEARNING ON GRAPHS

Temporal Graphs. We first define the concept of temporal graphs, which are graphs such that each connection is associated with a timestamp. We formally define temporal graph as follows:

Definition 1 (Temporal Graphs)**.** *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$ be a temporal graph, where $\mathcal{V}$ is the set of nodes, $\mathcal{T}$ is the set of timestamps, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V} \times \mathcal{T}$ is the set of edges. That is, each connection between two nodes (u, v) is associated with a timestamp like $t \in \mathcal{T}$.*

In this study, we use snapshot based representation of temporal graphs. That is, $\mathcal{G} = \{\mathcal{G}^{(t)}\}$ is the set of graphs, where $\mathcal{G}^{(t)} = (\mathcal{V}, \mathcal{E}^{(t)})$ represents the state of the graph $\mathcal{G}$ at timestamp t.

Node Representation Learning. Node representation learning in graphs is a process that aim to map nodes of a graph into a vector space. This representation seeks to capture the structure of the graph, the features of the nodes, their dynamic over time, and their relationships. The core idea is to represent each node with a vector that encapsulates not just its own attributes but also its position, dynamics, and role within the larger graph structure.

Definition 2 (Node Representation Learning)**.** *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph with nodes $\mathcal{V}$ and edges $\mathcal{E}$. The goal is to learn a function $f : \mathcal{V} \to \mathbb{R}^d$, where d is the dimension of the target vector space (usually $d \ll |V|$, implying a lower-dimensional representation).*

A.2 VOXEL TIME SERIES ACTIVITY

Neuroimaging modalities (e.g., fMRI, MEG) provide (estimated) recordings of neural activity signals. To this end, their estimation is built up in a 3-D image building block, units called voxels, which represent a small cube of brain tissue. For example, fMRI measures the blood-oxygen level dependence (BOLD) of each voxel in order to estimate the neural activity of the whole brain over time. In the literature, for each voxel, most studies aggregate its activity (e.g., its BOLD) over each time window, called beta weight (Roth et al., 2022; Vassena et al., 2020; Roth & Merriam, 2023). However, this approach misses the voxel activity dynamic over each task. In this study, we consider voxels' activity as it is (without aggregation) and model it as timeseries data. We model this data as a multivariate time seris: An fMRI scan involves thousands of voxels, leading to a multivariate time series $\{X_1(t), X_2(t), ..., X_n(t)\}$ where $X_i(t)$ is the time series for the i-th voxel.

A.3 BRAIN FUNCTIONAL CONNECTIVITY

The brain's functional connectivity is a graph, derived from a neuroimaging modality (often fMRI), where each node represents a brain parcel or ROI, and two nodes are connected if there is a statistical association between their functionality. In more details, as discussed above, fMRI measures brain activity by detecting changes in blood flow. The primary data from fMRI is the Blood Oxygen Level Dependent (BOLD) signal, reflecting changes in the oxygenation level of the blood. Deriving a brain network from fMRI data involves ① preprocessing, ② parcellation using atlases, and ③ computing correlations.

1. Preprocessing: Is the sequence of actions to clean the data and make it for process. Common preprocessing techniques are:
    - **Motion Correction:** Aligns all the neuroimages to a reference neuroimage to correct for patient movement.
    - **Band-Pass Filtering:** Isolating the frequency band that corresponds to the fMRI signal (usually 0.01 to 0.1 Hz).
    - **Slice Timing Correction:** Adjusts for the time difference in image acquisition between slices.
    - **Smoothing:** Applies a Gaussian filter to reduce spatial noise and improve signal-to-noise ratio.

2. Parcellation using Atlases: Brain atlases divide the brain into regions of interest (ROIs). Each ROI represents a node in the brain network. Common atlases include:
   - **AAL (Automated Anatomical Labeling):** Divides the brain into areas based on anatomical structures.
   - **Harvard-Oxford Atlas:** Based on probabilistic information from a large population.
   - **Functional Atlases:** Based on functional connectivity patterns, e.g., resting-state networks.
3. Computing Correlations: For each ROI defined by the atlas, the time series of the fMRI signal is extracted. This step involves averaging the fMRI signal over all voxels within each ROI (Note that in this paper, we argue that the best case is to consider each voxel as an ROI). Next, to capture the statistical association of ROIs activity, we compute the Pearson correlation coefficient between the time series of every pair of ROIs:

$$\mathbf{C}_{ij} = \frac{\sum_{k=1}^{T}(X_i(k) - \bar{X}_i)(X_j(k) - \bar{X}_j)}{\sqrt{\left(\sum_{k=1}^{T}(X_i(k) - \bar{X}_i)^2\right)\left(\sum_{k=1}^{T}(X_j(k) - \bar{X}_j)^2\right)}}, \tag{8}$$

   Here, C_{ij} is the correlation between ROI i and ROI j, and X_i, X_j are the time series for ROIs i and j, respectively. To construct a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of ROI and $\mathcal{E}$ is the set of connections, a threshold is applied to the correlation matrix. Only correlations above a certain value are considered to represent connections in $\mathcal{E}$.

In EEG and MEG data the process is the same while each signal corresponds to a channel and so in the constructed brain network, each node is a channel and each connection shows high Pearson's correlation between its endpoints.

A.4 MLP-MIXER

The MLP-MIXER architecture (Tolstikhin et al., 2021) is a novel neural network design that has attracted much attention in the field of computer vision. It presents itself as a distinctive alternative to the CNNs and Transformer models. The structure of MLP-MIXER is composed of two key sub-layers in each layer: the patch mixing layer and the channel mixing layer. The patch mixing layer processes spatial information within each channel independently, whereas the channel mixing layer combines the information across various channels. This dual process of mixing is crucial for the MLP-MIXER's capability to detect both local and global image dependencies.

The mathematical representation of the MLP-MIXER is as follows:

patch Mixer:

$$\mathbf{H}_{\text{patch}} = \mathbf{E} + \mathbf{W}^{(2)}_{\text{patch}}\sigma\left(\mathbf{W}^{(1)}_{\text{patch}}\texttt{LayerNorm}(\mathbf{E})^{\top}\right)^{\top}, \tag{9}$$

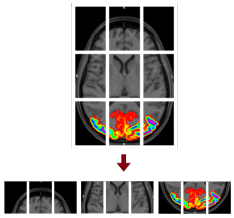
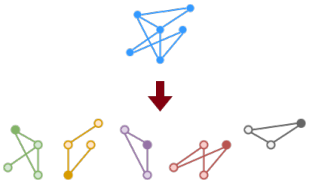
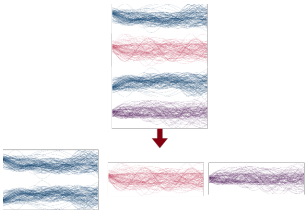
Channel Mixer:

$$\mathbf{H}_{\text{channel}} = \mathbf{H}_{\text{patch}} + \mathbf{W}^{(2)}_{\text{channel}}\sigma\left(\mathbf{W}^{(1)}_{\text{channel}}\texttt{LayerNorm}(\mathbf{H}_{\text{patch}})\right), \tag{10}$$

Challenges of Extending MLP-MIXER to Graphs and Time Series. In graphs, the vanilla MLP-MIXER (Tolstikhin et al., 2021) can be used to bind information across both of feature and node dimensions, but directly applying vanilla MLP-MIXER to graphs is insufficient and impractical. First, there does not exist in general a canonical grid of the graphs (contrary to images) to encode nodes, which makes patch extraction challenging. Second, contrary to images that can be divided into patches of the same size, the partitioning of nodes in graphs might not be all the same size due to the complex graph topology. Moreover, in temporal graphs, dynamics of the graph and its temporal properties should be captured to effectively learn its node encodings. The vanilla MLP-MIXER is not capable of learning temporal dynamics.

Similarly, in multivariate time series data, there is no canonical grid and patches are not necessarily the same size.

Table 4: The differences of VA Encoder and FC Encoder with MLP-MIXER.

	MLP-MIXER (Tolstikhin et al., 2021)	FC Encoder	VA Encoder
Patches			
Input	Images (2-d Regular grid) Same width and height	Graphs (Irregular and non-grid) Different sizes (#Nodes, #Edges)	Time series (Unstructured) Variable Length
Patch Extraction	Based on pixels order Non-overlapping patches Same patches at each epoch	Based on Temporal Random Walks Overlapping patches Different patches at each epoch	Based on Brain Functional Systems Overlapping patches Different patches at each epoch
Patch Encoder	Same patch size Usign 2-d MLP	Variable patch size Using TPMIXER (1-d MLP + Dynamic self-attention)	Variable patch size Using Voxel-Mixer (2-d MLP + Dynamic self-attention)
Positional Information	Implicitly ordered	No universal ordering Permutation invariant	No universal ordering Permutation variant

Challenges of using MLP-MIXER and its variants for Brain Activity. While there are existing studies that aim to address the above limitations and define patches in graphs using graph partitioning algorithms (He et al., 2023), or first-hop neighbourhood (Cong et al., 2023), there are designed for general cases and miss special properties of the brain. ① The human brain is comprised of functional systems (Schaefer et al., 2018), which are groups of voxels that perform similar functions (Smith et al., 2013). Recently, Trockman & Kolter (2023) show that the main power of vision architectures like MLP-MIXER and VIT (Khan et al., 2022) comes from patching, splitting the image into multiple same size parts which might show the same concept. Inspired by this, we suggest using functional patching in analysis of brain activity, i.e. splitting voxels into some groups such that each group has similar functionality. ② This approach results in another challenge which is the size of functional systems are not the same and simply using vanilla MLP-MIXER on functional patches is not feasible. To this end, we present an interpolation method to linearly interpolate functional systems to the same size.

The main differences of our VA Encoder and FC Encoder with MLP-MIXER are summarized in Table 4. In this table, 1-d MLP and 2-d MLP refers to applying MLP in one of the dimensions and both dimensions, respectively.

B BVFC DATASET

In this section, we introduce BVFC dataset and discuss how it is derived from the fMRI and MEG data.

B.1 THINGS DATASET

The Things dataset[1] (Hebart et al., 2023) is a large battery of visual object recognition datasets that use a common set of images as visual stimuli. These datasets cover a broad range of data types, ranging from behavioral aspects, such as similarity judgments, to neural responses to the presented images, such as fMRI and MEG recordings. The shared image database includes more than 26,000 images in total, categorized into 1,854 object concepts. In this work, we used ① the THINGS fMRI1 dataset, consisting of event-related BOLD responses of three human subjects to 8,460 images selected from 720 categories (12 images per each). ② the THINGS MEG1 dataset, consisting of Magnetoencephalography (MEG) of 4 subjects for 22,248 images (1,854 categories, 12 images per

[1] https://things-initiative.org

category), collected over the course of 12 sessions. Both preprocessed and raw versions of the fMRI and MEG datasets are provided by the Things authors[2]. However, in the preprocessed version of fMRI dataset, each voxel is associated with a single beta value, which misses the dynamic of voxel activity over time. For the purpose of this work, we utilized the raw version as we required the time series of fMRI recordings and follow the following preprocess steps:

B.2 PREPROCESSING

The beta values provided in the preprocessed version of the THINGS fMRI dataset are single measures of each voxel's response to a certain stimulus, which are obtained by applying a general linear model (GLM) to the voxels' time series. Since the preprocessed dataset only offers beta values, we utilized and preprocessed the raw data without applying the GLM step at the end. The preprocessing pipeline used by the authors of THINGS also includes a semi-supervised ICA-denoising step, which requires prior experience with fMRI noise-signal classification. We replaced this stage with ICA-AROMA (Pruim et al., 2015), an automatic ICA-denoising tool, to improve the replicability of our results without the need for manual supervision or intervention in the denoising step. For the rest of the preprocessing steps, we followed the same pipeline used by Hebart et al. (2023). For each image, we use 13 seconds of fMRI signals of the human subject, and treat each as a time window. We use the output of the preprocess time series as the voxel level brain activity. To derive the brain functional connectivity from the time series data, we consider each voxel as an ROI and calculate the statistical association of the time series of two voxels v_i and v_j in each time window using Pearson's correlation:

$$\mathbf{C}_{ij} = \frac{\sum_{k=1}^{T}(X_i(k) - \bar{X}_i)(X_j(k) - \bar{X}_j)}{\sqrt{\left(\sum_{k=1}^{T}(X_i(k) - \bar{X}_i)^2\right)\left(\sum_{k=1}^{T}(X_j(k) - \bar{X}_j)^2\right)}}, \tag{11}$$

where $X_i(k)$ and $X_j(k)$ are v_i and v_j activities at time k, and $\bar{X}_i$ and $\bar{X}_j$ are their average activity over the time window, respectively. We next, for each voxel removes negative elements and then keep 90-percentile of its corresponding correlation. We use the same approach on the time series of channels in MEG to obtain brain connectivity networks. For the preprocessing scripts visit this link.

B.3 IMAGE CLASSIFICATION

Understanding object representation in the brain is a key step toward revealing the basic building blocks of human visual processing (Hebart et al., 2023). Toward this direction, in the first downstream task on BVFC we aim to classify seen images during the fMRI and MEG recording. As discussed above, the fMRI dataset consists of responses of three human subjects to 8,460 images selected from 720 categories (12 images per each) from THINGS database (Hebart et al., 2019). Each of the images has a high-level concept as its high-level label, which described the type of the object in the image (e.g., "Food", "Human Body", etc.)[3]. In the first task, we aim to predict the high-level label of the seen image by using the fMRI responses of the human subject. This task is a multi-class classification tasks with 9 classes.

B.4 ANOMALY DETECTION

In the fMRI1 THINGS dataset (Hebart et al., 2023), there are 100 unique catch images that were created using the generative adversarial neural network, BigGAN (Brock et al., 2019). These images were generated by interpolating between two latent vectors, yielding novel objects that were not recognizable. We take advantage of these images and design a downstream task to detect these images.

The visual cortex, responsible for processing visual information, is hierarchically organized with multiple layers building upon simpler features at lower stages (Van Essen & Maunsell, 1983). Initially,

[2] https://plus.figshare.com/collections/THINGS-data_A_multimodal_collection_of_large-scale_datasets_for_investigating_object_representations_in_brain_and_behavior/6161151

[3] Note that the high-level labels of images are different from their original labels, as each high-level class include a set of primary labels. For example, all "Pizza", "Fast Food", and "Bacon" are in a high-level class of "Food".

neurons detect edges and colors, but on deeper levels, they specialize in recognizing more complex patterns and objects. Accordingly, we expect our model to detect GAN generated images by using the subject's brain fMRI response. We model this task as a binary classification task, where the brain fMRI response to each natural image is considered "normal" and the brain fMRI response to GAN generated images is considered "Abnormal". For further information about the generated images by GAN and its architecture see the original paper of THINGS dataset (Hebart et al., 2023) and original paper of BigGAN (Brock et al., 2019).

C Additional Related Work

To further situate our BrainMixer in a broader context, we briefly review self-supervised representation learning of brain activity and time series representation learning.

Self-supervised Representation Learning of Brain Activity. In representation learning of brain activity, such as fMRI, MEG and EEG, obtaining labeled data is challenging and costly. To address this, various self-supervised learning techniques have been introduced. Banville et al. (2021) suggest using relative positioning, temporal shuffling, and contrastive predictive coding specifically to EEG data. Additionally, Mohsenvand et al. (2020) and Kostas et al. (2021) presents an approach to learn EEG signals using negative sampling and contrastive learning, which is only suitable for SEEG and EEG data. Yang et al. (2023b) propose an unsupervised pre-training technique 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. All these methods are either ① are designed for a specific type of neuroimaging data (e.g., EEG) and cannot be generalized to other neuroimage modalities, ② are based on negative sampling generation which bias the performance toward the patterns of generated negative samples, being unable to generalize to complex and unseen patterns, and/or ③ uses either time series data or connectivity network, missing information from different level of granularity.

Time Series Representation Learning. Representation learning of multivariate time series has been getting increasingly popular with the motivation that modeling the complex relationships between covariates should improve the forecasting performance(Chen et al., 2023). to this end, Transformers (Vaswani et al., 2017) attract much attention due to their superior performance in modeling long-term sequential data. (Zhou et al., 2021) present Informer and (Wu et al., 2021) present Autoformer to address the efficiency challenges in long-term forecasting. Zhou et al. (2022b) design FEDformer and later FiLM (Zhou et al., 2022a) that decompose the sequences using Fast Fourier Transformation for better extraction of long-term information. Recently, Chen et al. (2023) design TSMixer, and all MLP architecture for time series forecasting. Not only the purpose of these methods are different from VA Encoder, but also their architectures are different from VA Encoder from a subset of following aspects: ① They bind information across time series, missing the cross-time dependencies of signals. ② These methods use fixed static learnable matrices for binding time series, while in the brain signals, the functionality of each time series is important and a different set of signals should be mixed differently based on their corresponding voxel's (channel's) connections and functionality. ③ They treat each time series the same, while in multivariate time series some signals cab be more important than others for a specific downstream task.

MLP-Mixer for fMRI Data. MLP-Mixer shows promising performance on image data. One approach to learn from fMRI data is to treat fMRI image in each time window as an image and then employ an MLP-Mixer to learn representation for voxels. Geenjaar et al. (2022) designed a fully-differentiable non-linear framework for whole-brain dynamic factor learning and applied MLP-Mixer to fMRI data. However, this study suffers from all the MLP-Mixer limitations that we discussed. For more explanations and illustrative examples see our discussion on the difference of our encoders with MLP-Mixer in Appendix A.4.

C.1 Our Contributions

As we discussed in Sections 1, 2, and Appendix C, existing studies miss a subset of the following: ① the functional connectivity between voxels, ② timeseries activity of voxels, ③ special properties of

the brain like hierarchical structure and its complex dynamics. Here, we summarize our contributions as follows:

1. Voxel Activity Encoder: We present VA Encoder, a novel multivariate time series encoder that employs a dynamic attention mechanism to bind information across both voxel and time dimensions. VA Encoder by learning the representation of each voxel allows us to obtain brain activity encodings at different level of granularity (e.g., voxel- , functional system-, and/or brain-level encodings). Our experiments (row 5 in Table 3) show that VA Encoder alone, i.e. without using functional connectivity, outperforms baselines in different downstream tasks.

2. Simple and Low Cost, but Effective Patching: We propose functional patching for learning brain activity. While existing patching methods are either ⓘ grid-based and inapplicable to graphs and time series, ⓘⓘ requires additional computational cost, and/or ⓘⓘⓘ cannot use specific brain properties (e.g., functional systems), missing the functional similarity of voxels. Our functional patching uses additional knowledge about the brain functional systems and patch the brain into some groups in which voxels have similar functionality. Our experimental results show that removing functional patching and replacing it with either random patching or clustering patching can damage the performance (See Appendix F.2 and Table 3).

3. Functional Connectivity Encoder: To encode the functional connectivity graph, we design an MLP-based architecture that learns both the structural and temporal properties of the graph using temporal random walks. FC Encoder first extracts temporal patches using temporal random walks and then fuses information within each patch using a novel dynamic self-attention mechanism. To obtain the brain activity encoding at different level of granularity, we further propose an adaptive permutation invariant pooling method that theoretically is the universal approximator of any multi-set function. Our experimental results in Table 3 show that FC Encoder alone, i.e. without using time series of voxel activity, outperforms baselines in different downstream tasks.

4. Self-Supervised Pre-training Framework: We present a novel self-supervised pre-training framework without using contrastive learning, which requires generating negative samples. Existing pre-trianing methods for the representation learning of brain activity suffers from two main limitations: ⓘ They require negative samples to learn from data in a contrastive manner (Yang et al., 2023b). However, brain activity is complex by its nature, and simple negative samples cause missing complex patterns, damaging the performance. ⓘⓘ They are based on a meta knowledge about a specific brain disease and so cannot generalize to other neuroimage modalities and different neuroimaging tasks (Yuan et al., 2023). Our framework allows self-supervised pre-training of any neuroimaging data that provides multivariate time series (e.g., fMRI, EEG, MEG, iEEG, etc.) without using any meta knowledge about the disease or downstream tasks, making it generalizable to different neuroimage modalities and different downstream tasks.

D Theoretical Guarantee of TSetMixer Performance

Theorem 1. TPMixer *is permutation invariant and a universal approximator of multisets.*

Proof. Let $\pi(S)$ be an arbitrary permutation of set S, we aim to show that $\Psi(S) = \Psi(\pi(S))$. We first recall the TSetMixer and its two phases: Let $S = \{\mathbf{v}_1, \ldots, \mathbf{v}_d\}$, where $\mathbf{v}_i \in \mathbb{R}^{d_1}$, be the input set and $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_d]^T \in \mathbb{R}^{d \times d_1}$ be its matrix representation: we first fuse information across features in a non-parametric manner as follows:

$$\mathbf{H}_{\text{F}}^{(t)} = \mathbf{V} + \sigma\left(\texttt{Softmax}\left(\texttt{LayerNorm}\left(\mathbf{V}\right)^{\top}\right)\right)^{\top}, \tag{12}$$

Now, for $\pi(S)$, let $\pi(\mathbf{V}) = [\pi(\mathbf{v}_1), \dots, \pi(\mathbf{v}_d)]^T \in \mathbb{R}^{d \times d_1}$ be its matrix representation, for the first phase we have:

$$\pi(\mathbf{V}) + \sigma\left(\texttt{Softmax}\left(\texttt{LayerNorm}\left(\pi(\mathbf{V})\right)^\top\right)\right)^\top \tag{13}$$

$$= \pi(\mathbf{V}) + \sigma\left(\texttt{Softmax}\left(\pi\left(\texttt{LayerNorm}\left(\mathbf{V}\right)^\top\right)\right)\right)^\top \tag{14}$$

$$= \pi(\mathbf{V}) + \pi\left(\sigma\left(\texttt{Softmax}\left(\texttt{LayerNorm}\left(\mathbf{V}\right)^\top\right)\right)^\top\right) \tag{15}$$

$$= \pi\left(\mathbf{V} + \sigma\left(\texttt{Softmax}\left(\texttt{LayerNorm}\left(\mathbf{V}\right)^\top\right)\right)^\top\right) \tag{16}$$

$$= \pi\left(\mathbf{H}_{\text{F}}^{(t)}\right), \tag{17}$$

which means that the first phase of TSETMIXER is equivariant. In the above, we used the fact that `Softmax` is permutation equivariant. In the second part, we first have:

$$\mathbf{P}_{\text{Pool}_i} = \textsc{Softmax}\left(\textsc{Flat}\left(\mathbf{H}_{\text{F}}^{(t)}\right)\mathbf{W}_{\text{Pool}}^{(i)}\right) \tag{18}$$

$$\Rightarrow \quad \textsc{Softmax}\left(\textsc{Flat}\left(\pi(\mathbf{H}_{\text{F}}^{(t)})\right)\mathbf{W}_{\text{Pool}}^{(i)}\right) \tag{19}$$

$$= \textsc{Softmax}\left(\pi\left(\textsc{Flat}\left(\mathbf{H}_{\text{F}}^{(t)})\right)\mathbf{W}_{\text{Pool}}^{(i)}\right)\right) \tag{20}$$

$$= \pi\left(\textsc{Softmax}\left(\textsc{Flat}\left(\mathbf{H}_{\text{F}}^{(t)})\right)\mathbf{W}_{\text{Pool}}^{(i)}\right)\right) \tag{21}$$

$$= \pi\left(\mathbf{P}_{\text{Pool}_i}\right)). \tag{22}$$

Also, note that we defined $\mathbf{H}_{\text{PE}}^{(t)}$ as $\mathbf{H}_{\text{PE}}^{(t)} = \mathbf{H}_{\text{F}}^{(t)}\mathbf{P}_{\text{Pool}}$. Therefore, we have:

$$\textsc{Mean}\left(\texttt{Norm}\left(\pi\left(\mathbf{H}_{\text{F}}^{(t)}\right)\right) + \pi\left(\mathbf{H}_{\text{PE}}^{(t)}\right) \ \textsc{Softmax}\left(\pi\left(\frac{\mathbf{H}_{\text{PE}}^{(t)\top}\mathbf{H}_{\text{PE}}^{(t)}}{\sqrt{d_{\text{patch}}}}\right)\right)\right) \tag{23}$$

$$= \textsc{Mean}\left(\pi\left(\texttt{Norm}\left(\mathbf{H}_{\text{F}}^{(t)}\right) + \mathbf{H}_{\text{PE}}^{(t)} \ \textsc{Softmax}\left(\frac{\mathbf{H}_{\text{PE}}^{(t)\top}\mathbf{H}_{\text{PE}}^{(t)}}{\sqrt{d_{\text{patch}}}}\right)\right)\right) \tag{24}$$

$$= \mathbf{h}_{\rho_v} \tag{25}$$

In the last step, we use the fact that $\textsc{Mean}(.)$ is permutation invariant, which results TSETMIXER to be permutation invariant.

Since the patch mixer is just normalization it is inevitable and cannot affect the expressive power of TSETMIXER. Also, channel mixer is a 2-layer MLP with attention, which are the universal approximator of any function. Therefore, due to the fact that TSETMIXER is permutation invariant, we can conclude that it is a universal approximator of multi-set functions. □

E EXPERIMENTAL SETUP

E.1 BASELINES

Since BRAINMIXER combines functional connectivity and voxel timeseries activity, we compare our method to fourteen previous state-of-the-art methods and baselines on the timeseries, functional connectivity, and graph encoding:

1. GOutlier (Aggarwal et al., 2011): GOutlier uses a generative model for edges in a node cluster and labels outliers as anomaly.
2. NETWALK (Yu et al., 2018): Yu et al. (2018) proposed a random-walk based dynamic graph embedding approach, NETWALK. NETWALK first uses simple random walks and jointly minimizes the pairwise distance of vertex representations of each sampled walk. Next, it uses a clustering-based technique to dynamically detect network anomalies.

3. HYPERSAGCN (Zhang et al., 2020): HyperSAGCN (Self-attention-based graph convolutional network for hypergraphs) utilizes a spectral aggregated graph convolutional network to refine the embeddings of nodes within each hyperedge. HyperSAGCN generates initial node embeddings by hypergraph random walks and combines node embeddings by MEAN(.) pooling to compute the embedding of hyperedge. The model with code can be found in here.
4. 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.
5. 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 MLP-MIXER to encode each representative matrix to obtain node encodings. The model with code can be found in here.
6. 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.
7. 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.
8. 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.
9. ADMIRE (Behrouz & 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.
10. 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.
11. PTGB (Yang et al., 2023b): 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.
12. 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.
13. 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.
14. 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 BRAINMIXER. 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., 2023b), which also is capable of pre-training, we use the same datasets and settings as we use for BRAINMIXER.

E.2 DATASETS

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

- BVFC (This Paper): The main characteristics and pre-processing steps are mentioned in Appendix B. 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 edge- and 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 the experiments, for the sake of efficiency, we remove irrelevant voxels.
- BVFC-MEG (This Paper): The main characteristics and pre-processing steps are mentioned in Appendix B. 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 edge- and 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".
- 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., 2022a) 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.
- ASD (Craddock et al., 2013): This dataset includes the resting fMRI data taken from the Autism Brain Imaging Data Exchange (ABIDE) (Craddock et al., 2013); it contains data for 45 subjects (22 female, age $= 25.4 \pm 8.9$ yrs) in the typically developed control group and 45 subjects (23 female, age $= 23.1 \pm 8.1$ yrs) in the ASD group. We have followed the five pre-processing strategies denoted as DPARSF, followed by Band-Pass Filtering. 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 ASD group as "Anomalous".
- 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. 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.

We model the pre-processed fMRI, MEG, and EEG signals as multivariate time series and use them as our time series activity. Next, we discuss how we derive the brain connectivity network.

Constructing Brain Connectivity Network. To construct the brain connectivity networks for each dataset, we use the same process as we do for BVFC. We use the output of the preprocess time series

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
ASD	90	400	5903	-	Yes
HCP	7440 1067	1000	7635 8041	7 (Mental states) 5 (Age)	Yes

Table 6: Hyperparameters used in the grid search.

Datasets	Sampling Number M	Sampling Time Bias θ	Temporal Walk Length m	Hidden dimensions
BVFC	4, 8, 16, 32, 64, 128	$\{0.5, 2.0, 20, 200, 2000, 20000\} \times 10^{-7}$	1, 2, 3, 4	32, 64, 128
BVFC-MEG	4, 8, 16, 32, 64	$\{0.5, 2.0, 20, 200, 2000, 20000\} \times 10^{-7}$	1, 2, 3, 4, 5, 6	32, 64, 128
ADHD	4, 8, 16, 32, 64, 128	$\{0.5, 2.0, 20, 200, 2000, 20000\} \times 10^{-7}$	1, 2, 3, 4	32, 64, 128
TUH-EEG	4, 8, 16, 32, 64	$\{0.5, 2.0, 20, 200, 2000, 20000\} \times 10^{-7}$	1, 2, 3, 4, 5, 6	32, 64, 128
ASD	4, 8, 16, 32, 64	$\{0.5, 2.0, 20, 200, 2000, 20000\} \times 10^{-7}$	1, 2, 3, 4	32, 64, 128
HCP	4, 8, 16, 32, 64	$\{0.5, 2.0, 20, 200, 2000, 20000\} \times 10^{-7}$	1, 2, 3, 4	32, 64, 128

as the voxel level (channel level for EEG and MEG) brain activity. To derive the brain connectivity from the time series data, we consider each voxel (or channel) as an ROI and calculate the statistical association of the time series of two voxels (channels) v_i and v_j in each time window using Pearson's correlation:

$$\mathbf{C}_{ij} = \frac{\sum_{k=1}^{T}(X_i(k) - \bar{X}_i)(X_j(k) - \bar{X}_j)}{\sqrt{\left(\sum_{k=1}^{T}(X_i(k) - \bar{X}_i)^2\right)\left(\sum_{k=1}^{T}(X_j(k) - \bar{X}_j)^2\right)}}, \tag{26}$$

where $X_i(k)$ and $X_j(k)$ are v_i and v_j activities at time k, and $\bar{X}_i$ and $\bar{X}_j$ are their average activity over the time window, respectively. We next, for each voxel (channel) removes negative elements and then keep 90-percentile of its corresponding correlation.

E.3 Implementation, Training, and Hyperparameters Tuning

In each task, we split the data into training set (70% of the data), validation set (10% of the data), and test set (20% of the data). In the pre-training phase, we use both training set and validation set to train the model and then we fine tune the pre-trained model for downstream tasks using only training set. For downstream tasks, we use validation set to tune hyperparameters as discussed bellow. During the training of both pre-trained model and fine tuning for downstream tasks, the test set is untouched and it is used only for the final evaluation of the method.

In addition to hyperparameters and modules (activation functions) mentioned in the main paper, here, we report the training hyperparameters of BrainMixer: On all datasets, we use a batch size of 32 and use a learning rate of 10^{-3}. We use the maximum training epoch number of 100 with an early stopping strategy to stop training if the validation performance does not increase for more than 7 epochs. Furthermore, a dropout layers with rate = 0.1 is employed in all neural networks. To tune the model's hyperparameters, we systematically perform grid search. The search domains of each hyperparameter are reported in Table 6.

BrainMixer is implemented by `PyTorch` in `Python` and a Linux machine with GPU and 16GB of RAM is used to run evaluations.

Note that we use the same training pipeline as BrainMixer for all the baselines. 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., 2023b), which also is capable of pre-training, we use the same datasets and settings as we use for pre-training of BrainMixer.

E.4 INJECTING SYNTHETIC ANOMALIES

Due to the nature of anomaly detection tasks, specifically in neuroimaging data, there is a lack of unified definition for abnormal brain activity and so the ground truth labels are not available. To mitigate this challenge in evaluation of our approach, we first evaluate our approach using synthetic anomalies and then we perform case studies on real-world data and show that the found anomalies are compatible with previous findings.

① Injecting Abnormal Connections to Brain Connectivity Network: We randomly choose 5% (or 1%, depends on the setting) normal connections and corrupt them. Given a connection (u, v), we randomly change one of its endpoint, assume that u, to another voxel like w such that v and w have not been connected previously. Therefore, from a normal connection (u, v), we generate an abnormal connection (w, v). This method is used to evaluate the performance of anomaly detection methods in different domains (Ma et al., 2021), including neuroimaging (Behrouz & Seltzer, 2023a).

② Injecting Abnormal Activity to Brain Activity Time Series: We randomly choose 5% (or 1%, depends on the setting) of time series and corrupt them as follows: Given a voxel activity time series X_i and X_j, during time window t, we swap this part of these two randomly selected signals and construct $\tilde{X}_i$ and $\tilde{X}_j$ which are corrupted of X_i and X_j, respectively.

Note that only edge-AD and node-AD tasks are evaluated using synthetic data and graph-level anomaly detection methods are evaluated using ground-truth anomalies.

E.5 VISUALIZATION TOOLS

To visualize the average distribution of anomalous connections, we use BrainPainter (Marinescu et al., 2019) with the Desikan-Killiany atlas. Also, to visualize the average distribution of brain activities in the visual cortex we use Pycortex, which is an interactive surface visualizer for fMRI (Gao et al., 2015).

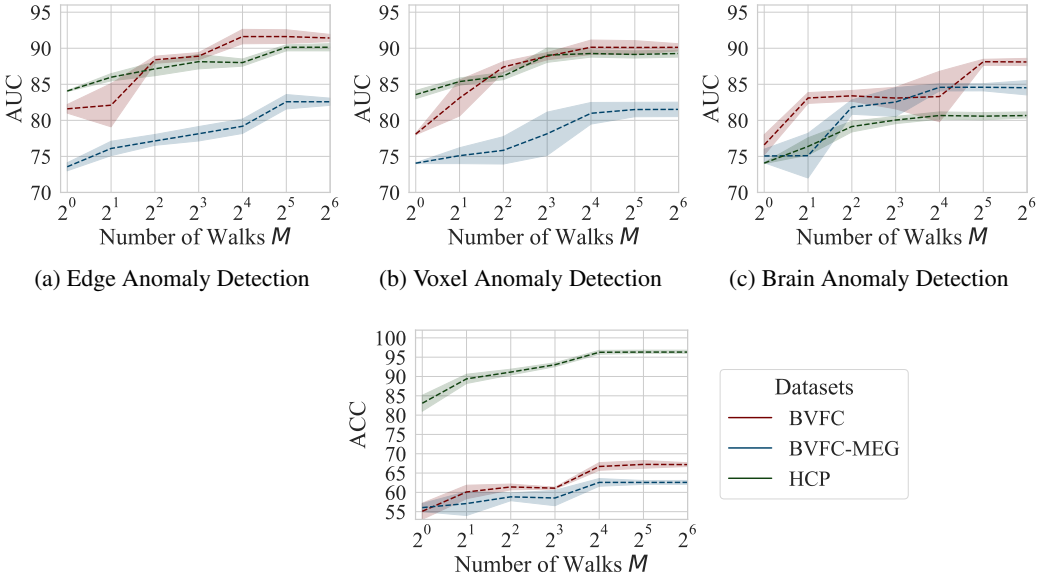
F ADDITIONAL EXPERIMENTAL RESULTS

F.1 PARAMETER SENSITIVITY

In this section, we evaluate the effect of hyperparameters on the performance of BRAINMIXER in different downstream tasks.

The Effect of the Number of Walks. In the first evaluation, we evaluate the effect of the number of walks on the performance of BRAINMIXER. Results are reported in Figure 4. These results show that increasing the number of walks results in better performance. The main reason is that we use the union of walks to capture the neighbourhood of each node over time. The more number of walks the better representation of the temporal neighborhood we can obtain. That is, sampling more walks lets the model to extract more information about the dynamic of nodes neighborhood as well as its structure. Also, notably, we observe that only a small number of sampled walks are needed to achieve competitive performance: in the best case 4 and in the worst case 16 sampled walks are needed to achieve better performance than baselines.

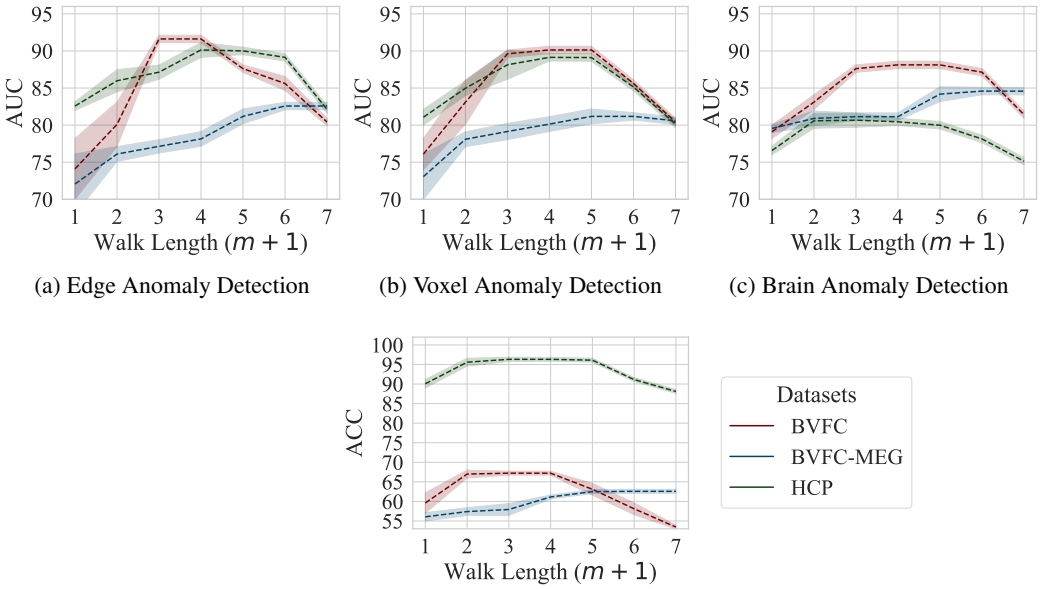
The Effect of the Walk Length. In this experiment, we evaluate the effect of the walk length on the performance of BRAINMIXER. Results are reported in Figure 5. The results suggest that the effect of the walk length on performance peaks at a certain point, but the exact value varies with datasets. The main reason for this is that we use walks to capture the structural and temporal properties of each node. Therefore, for dense brain connectivity networks as well as datasets with a large number of *relevant* time windows we need longer walks so the model can extract enough information about both *relevant* time windows and dense neighborhoods. Accordingly, we see increasing trend in BVFC-MEG's performance when we increase the length of the walk. Also, note that increasing the walk length for more sparser brain connectivity networks or for datasets with a smaller number of *relevant* time windows can damage the performance. The reason is we might consider irrelevant time windows by backtracking over time or consider far nodes (voxels in brain connectivity graph), which



(a) Edge Anomaly Detection (b) Voxel Anomaly Detection (c) Brain Anomaly Detection

(d) Brain Multi-class Classification

Figure 4: The effect of the number of walks on the performance of BRAINMIXER in different downstream tasks.



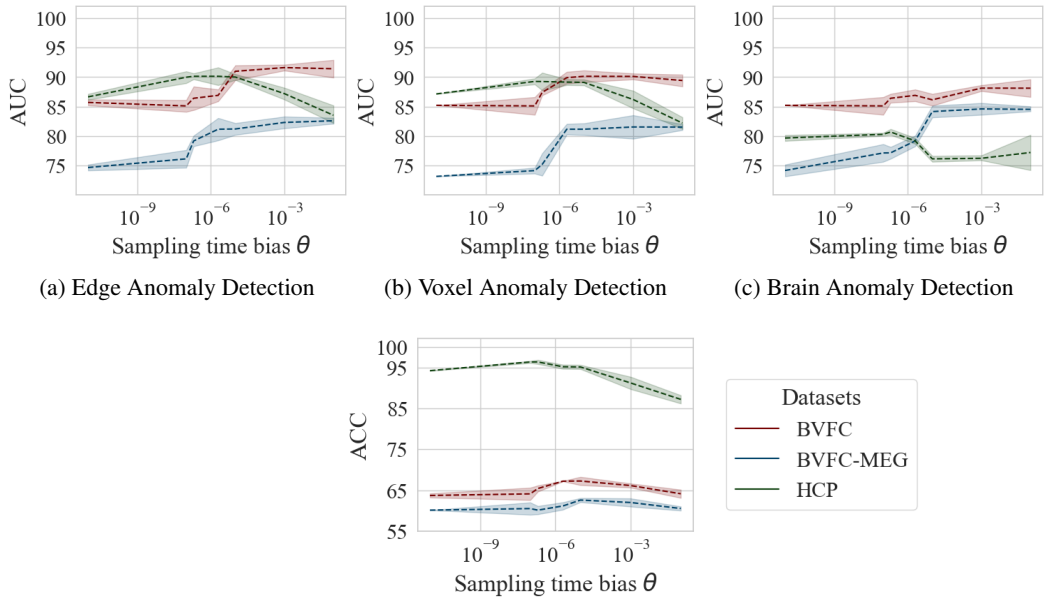
(a) Edge Anomaly Detection (b) Voxel Anomaly Detection (c) Brain Anomaly Detection

(d) Brain Multi-class Classification

Figure 5: The effect of the walk length on the performance of BRAINMIXER in different downstream tasks.

are irrelevant. Accordingly, depends on the structure of the brain connectivity graph and temporal properties of time series signals, we might need longer or shorter walks.

The Effect of the Time Decay θ. As we discussed in section 3, previous studies show that doing a task can affect brain activity even after 2 minutes (Yang et al., 2023a). To this end, since more recent



(a) Edge Anomaly Detection (b) Voxel Anomaly Detection (c) Brain Anomaly Detection

(d) Brain Multi-class Classification

Figure 6: The effect of the time decay θ on the performance of BRAINMIXER in different downstream tasks.

connections can be more informative, we use a biased sampling procedure and control the bias using a variable θ. That is, in the proposed sampling procedure, smaller (resp. larger) θ means less (resp. more) emphasis on recent timestamps.

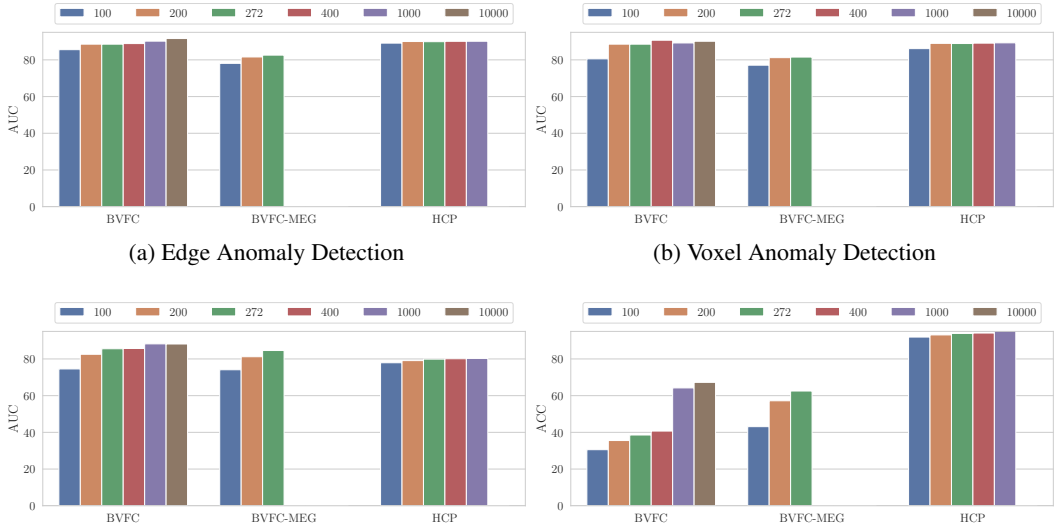
In this experiment, we evaluate the effect of time decay θ on the performance of BRAINMIXER. Results are presented in Figure 6. These results suggest that θ has a dataset-dependent optimal interval. That is, a small θ means an almost uniform sampling of brain activity history, which results in poor performance when the brain activities in recent time-windows are more important in the dataset. Also, very large θ might damage the performance as it makes the model focus on the most recent brain activity or only its own time window, missing long-term and lasting brain activities.

Please note that while the value of θ needs to be tuned to achieve the best performance, choosing arbitrary θ in a wide proper interval can still results in state-of-the-art performance over baselines.

The Effect of the Number of ROIs. The human brain is hierarchically organized and comprised of hierarchical groups of voxels that have similar functionality (Smith et al., 2013). Accordingly, different downstream tasks requires studying the brain at different level of granularity. In this experiment, we evaluate the the effect of the number of ROIs[4] on the performance of BRAINMIXER. We vary the number of ROIs from 10000 (voxel-level activity) to 100 (functional system-level activity) and report the results in Figure 7. The results suggest that using more ROIs and so studying the brain at lower-levels like voxel-level can result in a better performance. While there is a little improvement for downstream tasks that are correlated with human brain functional systems (e.g., HCP dataset and classification mental states), the significant improvements are for tasks that are highly correlated to a specific brain region (e.g., BVFC dataset and classifying seen images, which is closely related to human brain visual cortex). As an example, there is a $\approx 50\%$ performance loss in the accuracy of BRAINMIXER on BVFC in multi-class classification task as it requires learning voxel activity (e.g., V1 and V2) not learning the higher-level aggregated visual cortex activity.

The Effect of the Aggregation of Time Series. Most existing studies on voxel-level brain activity aggregate the voxel activity in each time window and use a single weight relating the voxel activity to

[4]Note that here ROI means any region of interest in the brain not necessarily pre-defined brain regions based on the atlases.



(a) Edge Anomaly Detection

(b) Voxel Anomaly Detection

(c) Brain Anomaly Detection

(d) Brain Multi-class Classification

Figure 7: The effect of the number of ROIs on the performance of BRAINMIXER in different downstream tasks.

the task, called beta weight. However, aggregation of the voxel activity misses the temporal property and dynamics of voxel activity over time. To this end, we suggest using a time series encoder that learn the dynamic of voxel activity over time, instead of simply aggregating them. In this experiment, we evaluate how much the aggregation of voxel activity time series can affect the performance. To this end, we take the mean of voxel activity time series and shorten it to 10%, 20%, 40%, and 60% of its original size. Figure 8 reports the results of this experiment on different downstream tasks. For datasets with low frequency sampling rate (e.g., HCP) aggregation does not significantly damage the performance. For the datasets with high frequency sampling rate (e.g., BVFC-MEG), however, aggregation of the voxel activity can significantly damage the performance ($\approx 12\%$ performance lost in the worst case and $\approx 5\%$ in the best case). These results show the importance of considering voxel activity as a time series instead of aggregating its activity and considering it as a single weight.

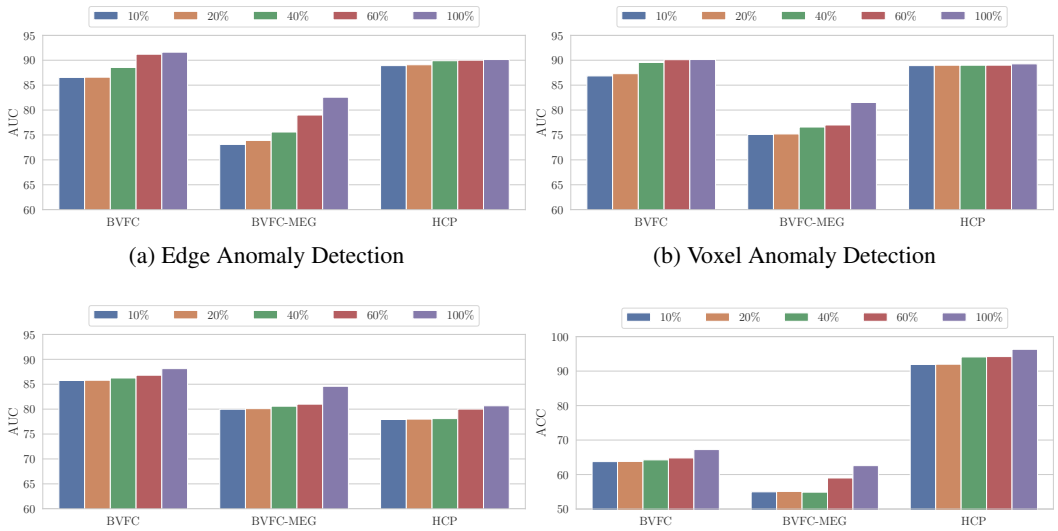
F.2 THE EFFECT OF FUNCTIONAL AND TEMPORAL PATCHING

As discussed in section 3, in both VA Encoder and FC Encoder we first split the data (either time series or graph) into patches. In this section, we replace the proposed functional and temporal patching methods with some existing patching strategies as well as some baselines to evaluate their contribution in BRAINMIXER's superior performance. For patching time series data we evaluate the following methods:

1. Random Patching: We randomly group time series in multivariate time series data and treat each group as a patch.
2. Ordered Patching: We use the actual order of time series in the multivariate time series and group consecutive time series as a patch.
3. Correlation Patching: We calculate the Pearson's correlation of multivariate time series (see Equation 26) and split the data into groups base on their pairwise correlation.
4. Functional Patching: This is our designed patching method, in which we group the time series of voxels in each brain functional system as a patch.

Also, for the graph patching we evaluate the following methods:

1. 1-hop Patching: We use the 1-hop neighborhood of each node as its corresponding patch.



(a) Edge Anomaly Detection

(b) Voxel Anomaly Detection

(c) Brain Anomaly Detection

(d) Brain Multi-class Classification

Figure 8: The effect of the aggregation of the time series on the performance of BRAINMIXER in different downstream tasks.

2. Partitioning Patching: Following He et al. (2023) for graph patching, we use METIS (Karypis & Kumar, 1998), a graph clustering algorithm that partitions a graph into a pre-defined number of clusters.

3. Spectral Clustering Patching: Following Geenjaar et al. (2022), we use spectral clustering patching used in this study.

4. Static Random Walk Patching: We replace temporal random walk in our temporal patching strategy with a static random walk. This random walk still should be able to capture structural properties but missing the dynamic of the graph.

5. Functional Patching in Graph: We use the actual brain functional systems as our patches.

6. Temporal Patching: This is our designed patching method for brain connectivity graph, in which we use temporal random walks that randomly sample temporal and structural neighborhood of each node.

In this experiments, we replace the baselines with our proposed patching method and keep the rest of the model unchanged. Results are reported in Table 7. Results show the superior performance of functional and temporal patching in time series and graph data, respectively. In time series patching, random and ordered patching perform poorly as they might group unrelated time series. Correlation patching performs better but still weaker than functional patching. The main reason for this superior performance is that we expect time series in each patch to have similar functionality and functional patching using the actual brain functional systems provides the best grouping since we know voxels in each functional system has similar functionality.

In graph patching, again our proposed temporal patching outperforms the other patching methods. Surprisingly, here functional patching performs poorly. The main reason is that in the functional connectivity graph, we connect highly correlated voxels (with respect to their activity). However, in each brain functional systems some voxels have complementary activity to others, which means while they have the same functionality, they might not have high correlation and so they are not connected. This fact results in considering disconnected subgraphs as patches, which is undesirable and so damages the performance.

Table 7: The effect of functional and temporal patching on the performance of BRAINMIXER: Mean ACC (%) $\pm$ standard deviation.

Patching Methods		BVFC	BVFC-MEG	HCP-Mental	HCP-Age
Time Series Patching	Random Patching	$60.42_{\pm 3.53}$	$60.28_{\pm 1.72}$	$91.97_{\pm 1.88}$	$45.71_{\pm 3.69}$
	Ordered Patching	$60.58_{\pm 0.60}$	$60.55_{\pm 1.01}$	$91.63_{\pm 1.57}$	$47.21_{\pm 0.91}$
	Correlation Patching	$66.97_{\pm 0.63}$	$60.91_{\pm 1.46}$	$95.08_{\pm 1.21}$	$55.30_{\pm 0.98}$
	Functional Patching	$\mathbf{67.24_{\pm 1.47}}$	$\mathbf{62.58_{\pm 1.12}}$	$\mathbf{96.32_{\pm 0.29}}$	$\mathbf{57.83_{\pm 1.03}}$
Graph Patching	1-hop Patching	$63.59_{\pm 0.09}$	$60.01_{\pm 0.18}$	$89.97_{\pm 0.16}$	$54.91_{\pm 0.71}$
	Partitioning Patching	$66.50_{\pm 1.92}$	$60.63_{\pm 1.87}$	$96.14_{\pm 1.04}$	$56.82_{\pm 1.75}$
	Spectral Clustering Patching	$63.77_{\pm 0.23}$	$59.16_{\pm 1.33}$	$90.24_{\pm 0.95}$	$48.34_{\pm 1.28}$
	Static Random Walk Patching	$66.28_{\pm 1.52}$	$59.94_{\pm 1.20}$	$95.86_{\pm 0.79}$	$57.81_{\pm 0.92}$
	Functional Patching in Graph	$60.03_{\pm 0.68}$	$54.99_{\pm 0.74}$	$91.45_{\pm 0.80}$	$50.11_{\pm 0.96}$
	Temporal Patching	$\mathbf{67.24_{\pm 1.47}}$	$\mathbf{62.58_{\pm 1.12}}$	$\mathbf{96.32_{\pm 0.29}}$	$\mathbf{57.83_{\pm 1.03}}$

Table 8: Performance on anomaly detection: Accuracy (%) $\pm$ standard deviation.

	Methods	BVFC	BVFC-MEG	HCP		ADHD		TUH-EEG		ASD	
	Anomaly %			1%	5 %	1%	5 %	1%	5 %	1%	5 %
Edge-level AD	GOUTLIER	$62.78_{\pm 2.41}$	$55.24_{\pm 2.03}$	$60.83_{\pm 1.73}$	$58.29_{\pm 1.49}$	$63.91_{\pm 1.84}$	$61.05_{\pm 2.32}$	$64.12_{\pm 1.46}$	$61.87_{\pm 1.61}$	$59.59_{\pm 0.83}$	$56.79_{\pm 1.51}$
	NETWALK	$69.21_{\pm 1.92}$	$58.47_{\pm 1.77}$	$71.99_{\pm 1.12}$	$69.64_{\pm 0.89}$	$68.85_{\pm 2.34}$	$66.18_{\pm 1.98}$	$70.71_{\pm 0.97}$	$67.92_{\pm 1.01}$	$68.39_{\pm 1.79}$	$65.87_{\pm 2.06}$
	HYPERSAGCN	$78.84_{\pm 1.73}$	$67.47_{\pm 0.92}$	$80.33_{\pm 1.61}$	$78.61_{\pm 1.24}$	$82.05_{\pm 1.75}$	$80.57_{\pm 1.32}$	$72.45_{\pm 1.67}$	$69.15_{\pm 0.89}$	$72.73_{\pm 0.92}$	$70.68_{\pm 1.77}$
	GRAPHMIXER	$84.82_{\pm 2.01}$	$71.34_{\pm 0.98}$	$85.36_{\pm 1.58}$	$83.99_{\pm 1.19}$	$83.89_{\pm 1.78}$	$81.58_{\pm 0.99}$	$74.29_{\pm 1.78}$	$72.48_{\pm 1.65}$	$83.33_{\pm 1.36}$	$80.94_{\pm 1.62}$
	BRAINNETCNN	$77.26_{\pm 0.89}$	$68.78_{\pm 1.49}$	$78.43_{\pm 0.94}$	$75.08_{\pm 1.33}$	$79.22_{\pm 1.97}$	$76.54_{\pm 1.17}$	$72.11_{\pm 1.49}$	$68.81_{\pm 1.06}$	$70.52_{\pm 1.47}$	$68.85_{\pm 1.80}$
	BRAINGNN	$78.09_{\pm 0.73}$	$69.71_{\pm 1.85}$	$80.85_{\pm 1.52}$	$78.54_{\pm 1.12}$	$77.02_{\pm 1.34}$	$75.68_{\pm 1.23}$	$71.36_{\pm 1.38}$	$68.83_{\pm 1.83}$	$70.24_{\pm 1.77}$	$68.92_{\pm 1.25}$
	FBNETGEN	$78.35_{\pm 1.98}$	$70.44_{\pm 1.58}$	$80.82_{\pm 1.25}$	$78.30_{\pm 1.56}$	$77.66_{\pm 0.89}$	$75.74_{\pm 1.78}$	$72.11_{\pm 1.57}$	$69.63_{\pm 1.47}$	$70.97_{\pm 1.42}$	$69.69_{\pm 1.18}$
	ADMIRE	$84.91_{\pm 1.71}$	$72.60_{\pm 1.91}$	$85.23_{\pm 1.07}$	$83.58_{\pm 1.44}$	$84.43_{\pm 1.68}$	$82.34_{\pm 1.19}$	$74.88_{\pm 1.07}$	$72.34_{\pm 1.72}$	$85.84_{\pm 1.55}$	$82.76_{\pm 1.48}$
	PTGB	$84.13_{\pm 1.78}$	$71.24_{\pm 1.43}$	$84.44_{\pm 1.86}$	$83.61_{\pm 1.35}$	$84.75_{\pm 1.49}$	$82.83_{\pm 1.54}$	$73.94_{\pm 1.56}$	$71.18_{\pm 1.48}$	$85.78_{\pm 1.22}$	$82.89_{\pm 1.61}$
	BNTRANSFORMER	$82.52_{\pm 1.64}$	$73.21_{\pm 1.78}$	$84.92_{\pm 1.29}$	$82.85_{\pm 1.61}$	$83.52_{\pm 1.81}$	$82.72_{\pm 1.31}$	$74.07_{\pm 1.36}$	$72.54_{\pm 1.15}$	$73.76_{\pm 1.80}$	$71.18_{\pm 1.59}$
	BRAINMIXER	$\mathbf{87.75_{\pm 1.58}}$	$\mathbf{79.12_{\pm 1.53}}$	$\mathbf{88.19_{\pm 1.97}}$	$\mathbf{87.83_{\pm 1.39}}$	$\mathbf{89.77_{\pm 1.12}}$	$\mathbf{88.12_{\pm 1.57}}$	$\mathbf{79.88_{\pm 1.24}}$	$\mathbf{77.47_{\pm 1.14}}$	$\mathbf{89.92_{\pm 1.57}}$	$\mathbf{88.39_{\pm 1.62}}$
Voxel-level AD	USAD	$65.13_{\pm 2.23}$	$61.28_{\pm 1.91}$	$63.38_{\pm 2.11}$	$62.18_{\pm 1.42}$	$70.41_{\pm 1.71}$	$69.06_{\pm 1.39}$	$69.46_{\pm 2.32}$	$68.07_{\pm 1.88}$	$64.29_{\pm 2.12}$	$63.41_{\pm 1.87}$
	TST	$67.12_{\pm 2.06}$	$67.10_{\pm 2.16}$	$67.16_{\pm 1.00}$	$66.03_{\pm 2.10}$	$72.11_{\pm 1.82}$	$70.50_{\pm 2.04}$	$70.31_{\pm 1.89}$	$69.32_{\pm 2.15}$	$67.14_{\pm 2.08}$	$66.41_{\pm 1.81}$
	MVTS	N/A	N/A	N/A	N/A	N/A	N/A	$73.15_{\pm 1.79}$	$73.01_{\pm 2.08}$	N/A	N/A
	GOUTLIER	$61.28_{\pm 1.78}$	$59.33_{\pm 1.16}$	$61.47_{\pm 1.70}$	$61.12_{\pm 1.93}$	$66.83_{\pm 1.82}$	$64.79_{\pm 2.16}$	$62.26_{\pm 2.02}$	$61.33_{\pm 1.51}$	$57.75_{\pm 1.95}$	$56.87_{\pm 1.95}$
	NETWALK	$65.31_{\pm 1.90}$	$62.38_{\pm 2.13}$	$64.65_{\pm 1.76}$	$63.39_{\pm 2.15}$	$73.71_{\pm 1.88}$	$71.12_{\pm 1.41}$	$69.40_{\pm 1.87}$	$68.89_{\pm 1.76}$	$69.71_{\pm 1.63}$	$68.49_{\pm 2.11}$
	HYPERSAGCN	$75.01_{\pm 1.50}$	$70.01_{\pm 1.72}$	$78.61_{\pm 1.98}$	$76.24_{\pm 1.46}$	$81.22_{\pm 1.54}$	$80.34_{\pm 1.26}$	$72.28_{\pm 1.45}$	$70.27_{\pm 2.20}$	$72.59_{\pm 1.56}$	$72.42_{\pm 1.92}$
	GRAPHMIXER	$75.06_{\pm 1.45}$	$70.14_{\pm 2.14}$	$79.32_{\pm 1.56}$	$78.14_{\pm 1.98}$	$79.45_{\pm 1.87}$	$77.56_{\pm 1.79}$	$69.70_{\pm 1.43}$	$68.18_{\pm 2.03}$	$70.42_{\pm 1.71}$	$70.39_{\pm 1.94}$
	BRAINNETCNN	$77.10_{\pm 1.29}$	$72.43_{\pm 2.09}$	$80.19_{\pm 1.64}$	$79.36_{\pm 1.43}$	$80.48_{\pm 1.33}$	$78.88_{\pm 1.65}$	$70.72_{\pm 1.62}$	$69.57_{\pm 1.10}$	$71.22_{\pm 1.75}$	$70.79_{\pm 2.04}$
	BRAINGNN	$76.48_{\pm 1.55}$	$72.06_{\pm 1.35}$	$80.26_{\pm 1.80}$	$79.09_{\pm 1.93}$	$80.17_{\pm 1.24}$	$77.97_{\pm 2.18}$	$70.32_{\pm 2.20}$	$68.88_{\pm 2.15}$	$70.39_{\pm 1.63}$	$69.33_{\pm 2.16}$
	FBNETGEN	$76.29_{\pm 2.02}$	$71.50_{\pm 1.96}$	$81.41_{\pm 0.91}$	$78.18_{\pm 1.86}$	$78.66_{\pm 1.47}$	$78.67_{\pm 1.48}$	$69.85_{\pm 2.06}$	$68.38_{\pm 1.97}$	$70.50_{\pm 1.59}$	$69.44_{\pm 2.08}$
	PTGB	$82.26_{\pm 1.37}$	$75.59_{\pm 1.42}$	$83.25_{\pm 1.51}$	$81.80_{\pm 2.02}$	$84.51_{\pm 1.32}$	$83.29_{\pm 1.39}$	$74.73_{\pm 1.02}$	$74.07_{\pm 2.13}$	$75.59_{\pm 2.04}$	$75.12_{\pm 2.12}$
	BNTRANSFORMER	$82.25_{\pm 1.56}$	$74.71_{\pm 2.10}$	$84.16_{\pm 1.03}$	$81.12_{\pm 1.11}$	$84.64_{\pm 1.62}$	$83.46_{\pm 2.13}$	$74.07_{\pm 1.68}$	$74.40_{\pm 1.97}$	$74.47_{\pm 1.18}$	$73.04_{\pm 1.74}$
	BRAINMIXER	$\mathbf{87.11_{\pm 1.03}}$	$\mathbf{80.38_{\pm 1.59}}$	$\mathbf{88.01_{\pm 1.22}}$	$\mathbf{85.84_{\pm 1.32}}$	$\mathbf{88.84_{\pm 1.37}}$	$\mathbf{86.55_{\pm 1.19}}$	$\mathbf{76.98_{\pm 1.42}}$	$\mathbf{76.11_{\pm 1.52}}$	$\mathbf{87.19_{\pm 1.93}}$	$\mathbf{87.11_{\pm 1.84}}$
Brain-level AD	USAD	$69.12_{\pm 1.92}$	$60.27_{\pm 1.41}$	$64.23_{\pm 1.55}$	$64.14_{\pm 2.27}$	$79.71_{\pm 2.09}$	$78.47_{\pm 2.11}$	$68.01_{\pm 2.06}$	$67.12_{\pm 1.18}$	$70.28_{\pm 1.87}$	$67.09_{\pm 1.68}$
	TST	$70.94_{\pm 2.26}$	$66.03_{\pm 1.94}$	$64.61_{\pm 1.78}$	$63.18_{\pm 1.69}$	$80.43_{\pm 2.01}$	$81.51_{\pm 1.21}$	$68.82_{\pm 2.03}$	$68.09_{\pm 1.79}$	$71.47_{\pm 1.64}$	$69.42_{\pm 2.08}$
	MVTS	N/A	N/A	N/A	N/A	N/A	N/A	$79.24_{\pm 1.56}$	$78.39_{\pm 1.80}$	N/A	N/A
	NETWALK	$70.45_{\pm 1.29}$	$68.23_{\pm 1.67}$	$66.45_{\pm 1.37}$	$65.23_{\pm 1.44}$	$80.89_{\pm 1.11}$	$80.28_{\pm 1.83}$	$67.01_{\pm 1.87}$	$65.37_{\pm 1.45}$	$71.29_{\pm 1.63}$	$69.26_{\pm 1.49}$
	HYPERSAGCN	$78.10_{\pm 1.18}$	$75.76_{\pm 1.22}$	$69.08_{\pm 1.42}$	$69.16_{\pm 1.37}$	$83.18_{\pm 1.39}$	$84.32_{\pm 1.65}$	$71.14_{\pm 1.11}$	$71.59_{\pm 1.32}$	$75.46_{\pm 1.60}$	$72.09_{\pm 1.37}$
	GMM	$79.55_{\pm 1.07}$	$76.09_{\pm 1.21}$	$71.44_{\pm 1.26}$	$71.55_{\pm 1.42}$	$82.14_{\pm 1.24}$	$83.20_{\pm 1.14}$	$72.00_{\pm 1.24}$	$72.82_{\pm 1.08}$	$75.76_{\pm 1.55}$	$72.08_{\pm 1.46}$
	GRAPHMIXER	$80.49_{\pm 1.10}$	$76.68_{\pm 1.38}$	$72.28_{\pm 1.24}$	$72.04_{\pm 1.17}$	$83.26_{\pm 1.05}$	$83.13_{\pm 1.21}$	$73.68_{\pm 1.37}$	$72.34_{\pm 1.33}$	$75.42_{\pm 1.64}$	$73.59_{\pm 1.62}$
	BRAINNETCNN	$76.29_{\pm 1.32}$	$72.70_{\pm 1.13}$	$67.59_{\pm 1.19}$	$67.89_{\pm 1.78}$	$82.77_{\pm 1.46}$	$82.04_{\pm 1.17}$	$69.83_{\pm 1.29}$	$69.29_{\pm 1.12}$	$74.86_{\pm 1.38}$	$72.15_{\pm 1.38}$
	BRAINGNN	$77.26_{\pm 1.42}$	$74.14_{\pm 1.28}$	$69.43_{\pm 1.35}$	$69.68_{\pm 1.59}$	$81.09_{\pm 1.42}$	$81.34_{\pm 1.44}$	$68.91_{\pm 1.53}$	$67.48_{\pm 1.28}$	$74.02_{\pm 1.15}$	$71.47_{\pm 1.47}$
	FBNETGEN	$76.28_{\pm 1.10}$	$73.29_{\pm 1.69}$	$68.22_{\pm 1.28}$	$68.53_{\pm 1.32}$	$81.08_{\pm 1.33}$	$82.57_{\pm 1.22}$	$68.51_{\pm 1.37}$	$67.95_{\pm 1.39}$	$74.12_{\pm 1.48}$	$71.61_{\pm 1.26}$
	ADMIRE	$81.54_{\pm 1.22}$	$77.27_{\pm 1.16}$	$72.92_{\pm 2.06}$	$71.34_{\pm 1.59}$	$83.35_{\pm 1.16}$	$83.19_{\pm 1.43}$	$74.32_{\pm 1.31}$	$73.11_{\pm 1.45}$	$76.62_{\pm 1.76}$	$73.88_{\pm 1.53}$
	PTGB	$82.01_{\pm 1.51}$	$78.12_{\pm 1.25}$	$73.53_{\pm 1.64}$	$72.62_{\pm 1.26}$	$84.42_{\pm 1.47}$	$84.62_{\pm 1.17}$	$75.81_{\pm 1.24}$	$74.19_{\pm 1.42}$	$79.98_{\pm 1.58}$	$77.29_{\pm 1.46}$
	BNTRANSFORMER	$81.60_{\pm 1.18}$	$78.14_{\pm 1.22}$	$72.13_{\pm 1.55}$	$72.84_{\pm 1.67}$	$84.71_{\pm 1.39}$	$84.09_{\pm 1.28}$	$75.14_{\pm 1.94}$	$74.44_{\pm 1.23}$	$78.18_{\pm 1.46}$	$73.50_{\pm 1.38}$
	BRAINMIXER	$\mathbf{86.62_{\pm 1.81}}$	$\mathbf{83.22_{\pm 2.01}}$	$\mathbf{77.92_{\pm 1.86}}$	$\mathbf{77.13_{\pm 1.35}}$	$\mathbf{88.51_{\pm 1.27}}$	$\mathbf{88.19_{\pm 1.61}}$	$\mathbf{81.22_{\pm 1.59}}$	$\mathbf{81.74_{\pm 1.27}}$	$\mathbf{88.14_{\pm 1.41}}$	$\mathbf{85.27_{\pm 1.35}}$

F.3 PERFORMANCE COMPARISON USING DIFFERENT METRICS

We compared the performance of BRAINMIXER with baselines in section 4. In multi-class classification tasks, we used accuracy as our metrics. Also, for the anomaly detection tasks, since we have binary labels (either anomaly or normal), we used AUC-PR as the metrics. In this section, we additionally evaluate the BRAINMIXER and baselines using accuracy for anomaly detection tasks and top-1 accuracy for multi-class brain classification tasks.

Accuracy. In this part, we compare the performance of BRAINMIXER and baselines in anomaly detection tasks, using accuracy metric. Table 8 reports the results. Similar to Table 2, these results show the superior performance of BRAINMIXER in all edge-level, voxel-level, and brain-level anomaly detection tasks.

Comparison of Best results. In Table 1, we reported the average of accuracy. In this experiment, we report the best performance of each model among 20 times of running. Table 9 reports the result and show that BRAINMIXER significantly outperforms baselines.

Table 9: Performance on multi-class brain classification: Top-1 ACC (%) (Best results among 20 times run).

Methods	BVFC	BVFC-MEG	HCP-Mental	HCP-Age
USAD	50.48	51.16	75.05	40.84
HYPERSAGCN	53.41	53.09	91.94	49.31
GMM	54.54	54.72	92.73	48.99
GRAPHMIXER	54.37	54.27	92.56	49.43
BRAINNETCNN	50.91	51.68	85.25	44.30
BRAINGNN	52.29	52.05	87.40	44.61
FBNETGEN	51.13	52.32	86.34	44.61
ADMIRE	55.73	56.78	91.65	49.48
PTGB	57.69	56.73	93.88	49.90
BNTRANSFORMER	56.37	56.80	93.17	49.08
BRAINMIXER	**68.67**	**63.68**	**96.63**	**58.88**

Table 10: Performance on brain network regression task: MAE ↓ (the lower value is better).

Dataset	BRAINMIXER	BNTRANSFORMER	BRAINGNN	BRAINNETCNN	GMM
HCP-AGGRESSIVE	**0.81**	0.96	1.72	1.59	1.05
HCP-INTRUSIVE	**0.95**	1.09	1.19	1.27	1.01
HCP-RULE-BREAK	**1.06**	1.14	2.01	1.44	1.38

F.4 GRAPH REGRESSION

In this section we evaluate the performance of BRAINMIXER in a regression task and compare it with baselines. In this task, we aim to predict Achenbach adult self-report (ASR) scores in HCP dataset, which are "Aggressive", "Intrusive", and "Rule-Break" scores. In this experimental setup, we use L1 loss to fine-tune the model for the downstream regression task. We also use the commonly used metric of Mean Absolute Errors (MAEs) on the prediction of these three scores.

Table 10 reports the results. In all three regression tasks, BRAINMIXER achieves the lower MAE and outperforms all the baselines.

F.5 QUALITATIVE RESULTS

In this section, we report some success and failure cases of BRAINMIXER in image classification task and detecting synthetic images based on fMRI. Figure 9 (resp. Figure 10) shows four examples of BRAINMIXER success (resp. failures) in prediction of the image label based on the brain activity. Finally, Figure 11 shows four example of BRAINMIXER failure in detecting synthetic (abnormal) images based on the fMRI.

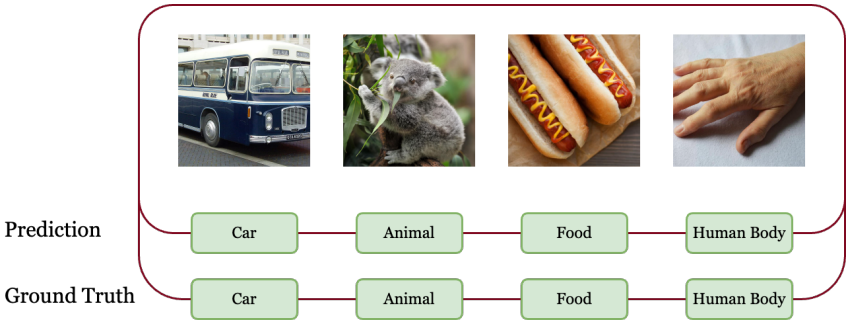


Figure 9: Examples of success cases in prediction of image labels based on fMRI.

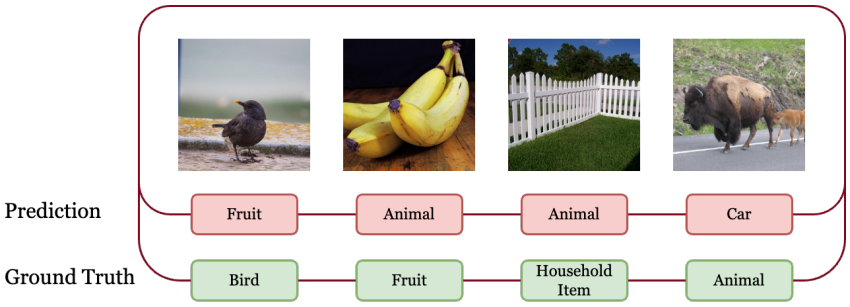


Figure 10: Examples of failure cases in prediction of image labels based on fMRI.

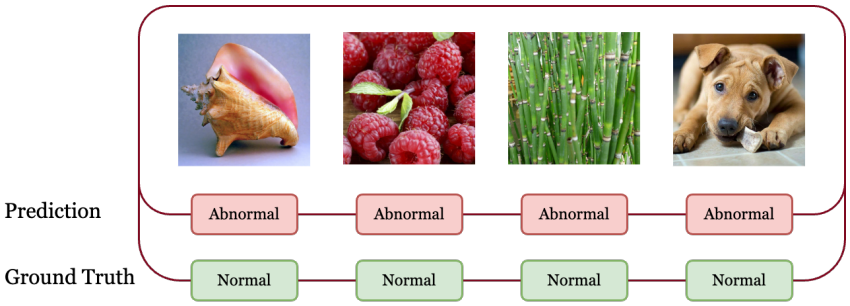


Figure 11: Examples of failure cases in detecting synthetic images based on fMRI.

F.6 How Does BrainMixer Detect GAN Generated Images?

The visual cortex, responsible for processing visual information, is hierarchically organized with multiple layers building upon simpler features at lower stages (Van Essen & Maunsell, 1983). Initially, neurons detect edges and colors, but on deeper levels, they specialize in recognizing more complex patterns and objects.

As we discussed in Appendix B, BVFC consists of the fMRI response when the subject sees the GAN-generated image, and we define an anomaly detection task, in which we aim to detect brain responses to GAN-generated images. While we report the performance of BrainMixer in detecting these anomalies, in this experiment, we examine how BrainMixer can detect fMRI responses to the GAN generated images. To this end, we split the test set into two groups based on BrainMixer's prediction: ① data samples that BrainMixer has detected as normal, and ② data samples that BrainMixer has detected as abnormal. Figure 2 (Left) reports the distribution of fMRI responses that BrainMixer found abnormal (i.e., corresponds to synthetic images) and Figure 2 (Right) reports the the distribution of fMRI responses that BrainMixer found normal (corresponds to natural images). Interestingly, while the distributions share similar patterns in lower levels (e.g., V1 and V2 voxels), higher-level voxels (e.g., V3) are less active when the subject sees non-recognizable images. This voxel activity drop in the V3 is $\approx 57\%$. These results are compatible with our expectation about the hierarchical structure of the visual cortex and so support that BrainMixer can learn a powerful representation for voxel activity.

F.7 Case Study: ASD

In this experimental design we train our model on a healthy control group, which lets the model learn normal brain patterns. After the training, we test our model on the ASD group and report the abnormal brain regions in the ASD group. The most repeatedly abnromal regions in ASD group are ① Right Cerebellum Cortex, ② Right-precuneus, and ③ Left-lingual. Our findings about the abnormal activity in the cerebellum cortex is consistent with previous studies (Rogers et al., 2013).

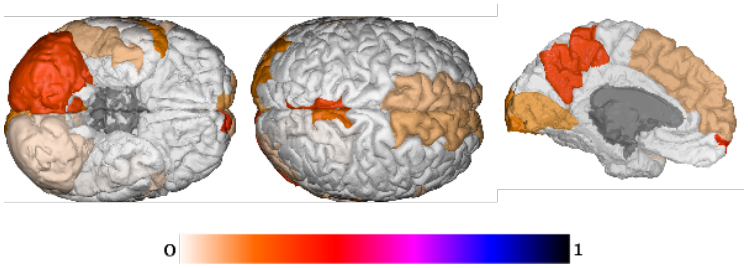


Figure 12: The distribution of detected abnormal voxels by BRAINMIXER in condition ASD group

Table 11: Performance on multi-class brain classification using different objectives (ACC).

Methods	BVFC	BVFC-MEG	HCP-Mental	HCP-Age
BRAINMIXER with contrastive learning and margin-based pairwise loss	47.30	49.77	53.78	49.41
BRAINMIXER with DGI	63.49	60.82	94.12	58.36
BRAINMIXER	**68.67**	**63.68**	**96.63**	**58.88**

Table 12: Number of parameters in different models designed for neuroimage data.

Methods	BVFC	BVFC-MEG	HCP-Mental	HCP-Age
BRAINNETCNN	4.1M	0.97M	1.1M	1.1M
BRAINGNN	6.7M	0.8M	0.8M	0.8M
FBNETGEN	7.2M	0.56M	1.1M	0.9M
ADMIRE	89.3M	4.9M	4.2M	4.2M
PTGB	146.1M	10.1M	9.6M	9.6M
BNTRANSFORMER	187.8M	8.7M	12.4M	12.4M
BRAINMIXER	117.4M	9.4M	8.3M	8.3M

F.8 THE EFFECT OF OBJECTIVE

In this experiment, to evaluate the significance of our loss function, we train the model with two other well known loss functions. ① Contrastive learning: we replace our loss function with margin-based pairwise. In this loss function, we aim to maximize the distances of positive and negative samples. ② Deep Graph InfoMax (Veličković et al., 2019): We use the encoding of each node as its local feature. Furthermore, we use the suumary of the all encoding as the global encoding of the graph. Results are reported in Table 11.

F.9 NUMBER OF PARAMETERS

To compare the capacity of the models, we report the number of parameters in Table 12.

F.10 EFFECT OF THE NUMBER OF PARAMETERS ON ACCURACY

To evaluate the effect of the number of parameters on the performance of the model, we use BVFC and employ BRAINMIXER with different capacity. We restrict BRAINMIXER's and its encoders' capacity to 80%, 60%, 50%, and 30% of their original capacity, which we have reported in Table 12. The results are reported in Figure 13.

G LIMITATIONS AND FUTURE WORK

In this work, we present an unsupervised pre-training framework, BRAINMIXER, that bridges the representation learning of voxel activity and functional connectivity by maximizing their mutual

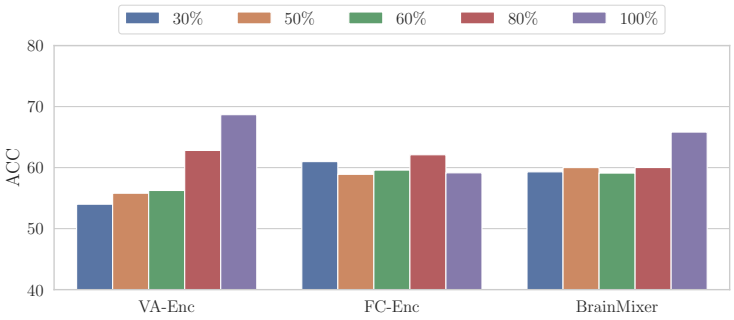


Figure 13: The performance of the BRAINMIXERwith different capacity on BVFC.

information. The promising performance of BRAINMIXER in several downstream tasks raises many interesting directions for future studies: While BRAINMIXER with a simple MLP can successfully classify observed images based on fMRI, one future direction is to pair BRAINMIXER with diffusion models (Xu et al., 2023) to directly decode brain visual system in an end-to-end manner. There are, however, a few limitations for BRAINMIXER: ⓘ In this study, we focus on designing a powerful unsupervised framework that could provide us with a robust and effective brain activity representation. However, reliability of machine learning methods for downstream tasks in sensitive domains (like healthcare) is critical. Evaluation of BRAINMIXER's prediction reliability and modifying BRAINMIXER so that it can provide us with the uncertainty of its prediction is left for future studies. ⓘⓘ The current approach is capable of using one neuroimage modalities, while different neuroimage modalities can provide complementary information, which can help understanding and detecting neurological disease or disorders. One potential future work is to design multimodal BRAINMIXER, where it can learn from different neuroimage modalities, taking advantage of their complementary information.