Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of ICT and Natural Sciences

Amirabbas Hojjati

A Multi-View Self-Supervised Approach to Learn Representations of EEG Data for Downstream Prediction Tasks

Master's thesis in Simulation and Visualization
Supervisor: Ibrahim A. Hameed
Co-supervisor: Anis Yazidi | Rabindra Khadka
June 2023

NTNU
Norwegian University of Science and Technology

Amirabbas Hojjati

A Multi-View Self-Supervised Approach to Learn Representations of EEG Data for Downstream Prediction Tasks

Master's thesis in Simulation and Visualization
Supervisor: Ibrahim A. Hameed
Co-supervisor: Anis Yazidi | Rabindra Khadka
June 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of ICT and Natural Sciences

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

Department of ICT and Natural Sciences

IE502414 - Master's Thesis in Simulation and Visualization

A Multi-View Self-Supervised Approach to Learn Representations of EEG Data for Downstream Prediction Tasks

Author:
Amirabbas Hojjati

Supervisor:
Prof. Ibrahim A. Hameed

External Supervisors:
Prof. Anis Yazidi
Rabindra Khadka

Acknowledgement

The journey of producing this thesis has been an enlightening and challenging experience, teaching me valuable lessons while signifying the support I have received from a range of remarkable individuals. This acknowledgement serves as a heartfelt expression of my appreciation and an opportunity to celebrate the invaluable support and mentorship that I have received during this journey.

I would like to express my deepest gratitude to my esteemed supervisors, Prof. Ibrahim A. Hameed and Prof. Anis Yazidi, for their support and guidance throughout the course of my thesis research, and providing me with insights to formulate and address an important problem. I am truly grateful for the opportunity to contribute to this field under their mentorship.

Furthermore, I would like to extend my heartfelt appreciation to Rabindra Khadka, a fellow researcher and PhD candidate whose assistance has been indispensable throughout this journey. Our weekly meetings on Sundays created a platform for insightful discussions, inspiring ideas, and invaluable feedback, without which this thesis would not have been the same. I am genuinely thankful for these moments of collaboration and brainstorming.

In addition, I want to express my gratitude to the numerous scholars, experts, and research participants whose contributions have enriched my thesis. To the libraries, archives, and data repositories that granted me access to valuable resources, I extend my heartfelt appreciation. The collaborative spirit of the broader research community has made an indelible impact on my work, and I am proud to be a part of it.

Lastly, I would like to acknowledge the support and encouragement offered by friends, family, classmates and the extended academic community, who have all played a significant role in my personal and academic growth. I am truly grateful for the opportunities, experiences, and connections fostered during this thesis and beyond.

This thesis can be considered in the context of the AI-Mind[1] project which aims to find an AI-based solution "to support healthcare professionals in their diagnosis and offering timely interventions to patients".

[1] https://www.ai-mind.eu/ (As of June 2023).

Abstract

This thesis presents a self-supervised deep learning approach for learning and extracting representations from long-period electroencephalogram (EEG) input data, in order to be used in downstream tasks such as prediction of dementia, Alzheimer's, general abnormality, or any other long-period and instance-level downstream task. The proposed method employs multi-view contrastive learning and Transformer-based architecture to extract useful representations from raw EEG data in both time and frequency domains. The study investigates the use of unlabeled data augmentations in conjunction with Transformers for the goal of feature representation learning and the combination of different views of time and frequency for effective pre-training tasks. The developed model is evaluated and validated using pre-training and downstream prediction tasks, demonstrating promising results in encoding long-period EEG data, as well as using the resulting representations for condition prediction. This research aims to contribute to the advancement of deep learning techniques in the analysis of EEG data and has potential applications in the early detection and diagnosis of neurological disorders, and opens the door for further research and investigation in this area.

Table of Contents

List of Figures

List of Tables

1 Chapter I - Introduction

In recent years, the prevalence of various patient conditions has started to pose a challenge to healthcare systems worldwide. With an aging population, conditions such as dementia and other brain abnormalities stand out as important and pressing concerns. We consider dementia as a potential use case for the results of this study, but we carry out main experiments on general abnormalities.

Dementia is a common term for a condition of decreased cognitive ability, which is severe enough to interfere with normal functioning of an individual in daily life. It is a challenging condition both in terms of diagnosis and treatment, and the prevalence of dementia is on the rise as life expectancy and general population age is increasing. According to World Health Organization, Around 55 million people suffer from dementia, a figure that is expected to grow to 139 million by 2050 [1].

One of the ways to assess a patient for dementia is cognitive assessment tests. These kinds of tests can range from simple questionnaires with basic questions to detailed neuropsychological cognitive tests which aim to assess a patient's cognitive ability in different areas of interest, including memory, attention, etc. It is also important to distinguish dementia from mild cognitive impairment (MCI) which is usually characterized by a noticeable decrease in cognitive ability while maintaining the ability to continue with functions necessary for daily life, and may or may not develop progress to dementia [2]. Several studies showed the relevance of cognitive scores to identifying MCI and dementia, with some suggesting that they can add significant value to the diagnosis and prediction of these conditions even for years into the future [3]–[5].

Electroencephalography (EEG) analysis is a popular method that has been used in dementia research, as well as research for other conditions related to brain activities, for years. EEG data represents the recordings of the brain's electrical activity, which is gathered non-invasively via electrodes positioned on the scalp. This information allows for the examination of brain functionality, cognitive processes, and neurological conditions by assessing the time-based changes and spatial arrangement of brain signals.

It is clear that EEG activity, especially when considered in certain frequency bands, is affected by dementia in different stages [6]. It has also shown a great potential in different tasks including prediction and separation of different types of dementia [7]. In [8], a strong correlation was demonstrated between certain aspects of the brain's EEG data and cognitive scores. In [9], it was suggested that resting-state EEG biomarkers have significant correlation with cognitive score, especially in early stages of cognitive decline. Additionally, the EEG data were found to be in correlation with age and education level. This can suggest that EEG data can be a predictor of cognitive score, and subsequently the cognitive functioning of the brain that dementia can deteriorate.

With recent advances in computing power and deep learning, EEG data can unlock even more potential in areas related to dementia and cognitive functioning. While less complex machine learning models such as support vector machines were tried initially to perform tasks such as emotional state classification [10] and seizure detection [11], there are increasingly more studies which have used deep learning methods to derive more predictive value from this type of data in different domains. Several studies have adopted recurrent neural networks while a significant number of them tried to make use of the popular convolutional neural networks (CNN). For example, In [12], the authors surveyed studies for detecting gaps in responsiveness and alertness using LSTM networks and found it to be effective.

On the other hand, in [12], [13], the authors studied using a convolutional neural network to predict different stages that happen in seizure which resulted in promising results, especially when the data is transformed into images using some transformation method, like time-slicing [14]. Since labeled data can be scarce and hard to obtain in clinical settings, semi-supervised approaches are gaining popularity. In this approach, the model is able to learn the feature representations from EEG data in an unsupervised manner, and then use the learned features for a chosen task, for example classification of emotions [15].

EEG data can contain big amounts of valuable information but it can also contain noise. It is usually analyzed either in its original form which is as a time-series in the time domain, or it can be transformed into the frequency domain and be analyzed and used in different frequency bands. It can also be analyzed in the time-frequency domain which considers frequencies with respect to time, with a method such as short-time fourier transform, wavelet transforms or spectral analysis. Considering each domain without the other domains might result in some loss of full potential of EEG data. Hence, there have been attempts to take several domains or views into account simultaneously. The goal is to find a shared representation from all views and use all the available information, and use the final representation to perform a task like seizure classification [16].

In this thesis, the goal is to develop a method in order to learn useful representations of long-period EEG data in a self-supervised way, in order to predict a patient's condition (dementia, Alzheimer's, abnormalities, etc.) based on the EEG data from their brain. This method should be able to extract the necessary data in different domains of interest (for example, time and frequency) and use information in all domains in order to carry out the downstream prediction task on a long-period EEG input, which can be an instance-level classification task in the case of predicting dementia. Although dementia has been listed as one of the main use cases, the method can potentially be applied in any domain where an overall instance-level prediction is desirable.There are several subgoals that need to be addressed in order to achieve the main goal:

- How can the raw EEG data be transformed into a format that is suitable for a given deep neural network to perform the prediction task?
- Is it possible to leverage unlabeled data in order to find the most interesting feature representations and use them as a starting point for prediction with the labeled data?
- How can data from different views be leveraged in order to perform a single effective prediction task that can make the most use of the available information?
- How to evaluate and validate the results in a way that can objectively show if the model is really effective at performing the given task?

The remainder of this thesis is structured as follows:

- Chapter 2 presents the methods and concepts used in the study.
- Chapter 3 includes literature review and existing work.
- Chapter 4 presents implementation and design of the proposed method.
- Chapter 5 provides evaluation and analysis of the proposed method.
- Chapter 6 concludes the thesis with some notes for further improvements.

2 Chapter II - Theoretical Background and Technical Concepts

Several methods and algorithms have been used in this study in order to reach the final outcome. In this section, we will examine the essential techniques and concepts utilized throughout this thesis, with a particular emphasis on EEG signal processing and machine learning methodologies. This part aims to present a deeper understanding of the relevant theories and technical elements, and thus creating a robust foundation for comprehending the research carried out and the subsequent findings.

2.1 Short-Time Fourier Transform (STFT)

Fourier transform is a good way of representing a signal in terms of its frequencies. This is the case for many time-based signals including EEG signals which usually comprise many frequencies at different amplitudes. When the signal is non-stationary, meaning that its characteristics change over time, a fourier transform is not enough to capture the frequency-related information of the original signal. That is where Short-Time Fourier Transform can be useful.

Short-Time Fourier Transform is a way of analyzing a non-stationary signal that varies over the period of its duration. The method works by applying the Fourier Transform to short and overlapping intervals of the original signal. This results in a spectral representation of the signal that varies over time and can show the changing frequency content of it. In its discrete form, it can be calculated using the following formula or some variation of it:

$$X_m(\omega) = \sum_{n=-\infty}^{\infty} x(n)w(n - mR)e^{-j\omega n} \tag{1}$$

Where x is the input signal at time n, ω is the frequency variable, w is the analysis filter (window function), and R is the hop size. After applying STFT to the input signal, each input signal can be represented in terms of frequency, amplitude and time.

We used STFT in this study in order to obtain a different view of the original time-based EEG signal and further used it as the input data in the pre-training pipeline.

2.2 Deep Learning Architectures

Deep learning models usually consist of multi-layered, hierarchical architectures designed for processing and learning complex data patterns. These models employ artificial neural networks, which draw inspiration from the human brain. There have been various types of deep learning architectures which were introduced in recent developments of artificial intelligence.

Convolutional Neural Networks (CNNs) specialize in image recognition by utilizing convolutional and pooling layers for feature detection and dimensionality reduction. Recurrent Neural Networks (RNNs) manage sequential data through connections with previous states that enable information persistence throughout the network, making them suitable for natural language processing and time series analysis. Long Short-Term Memory (LSTM) networks, a subtype of RNNs, address the vanishing gradient problem using

gating mechanisms that regulate information flow. Gated Recurrent Units (GRUs) are simplified LSTMs with fewer gates and parameters, yet exhibit comparable performance at a faster speed. Autoencoders learn efficient data representations through input data compression and reconstruction. Variational Autoencoders (VAEs) introduce a probabilistic component, learning a continuous latent space for generative purposes. Generative Adversarial Networks (GANs) consist of two adversarial networks, a generator and a discriminator, for producing realistic data samples. Transformer models, such as BERT and GPT, leverage self-attention mechanisms for text processing and generation tasks, achieving state-of-the-art results in numerous natural language processing, in addition to other generation tasks. The next sections briefly discuss the relevant architectures and layers relevant to this study.

2.2.1 Fully Connected Networks

Fully connected neural networks represent a category of artificial neural networks in which every neuron within a layer has connections to all neurons in the neighboring layers. These networks are composed of an input layer, multiple hidden layers, and an output layer. They are extensively employed in various applications, including image identification, natural language processing, and pattern detection. The learning process in these networks involves updating weights and biases using backpropagation and gradient descent techniques.

These networks often do not scale well due to reasons such as the increasing number of parameters as the network gets bigger or inability to generalize beyond their training dataset.

2.2.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are utilized extensively for image and video recognition tasks owing to their ability to extract important features, such as edges and textures, through mathematical filters or convolutions.

In the CNN architecture, input images undergo multiple layers of convolutional filters that maintain significant characteristics while decreasing or maintaining dimensionality. Subsequently, these filtered outputs may pass through activation functions such as ReLU before potentially being passed onto pooling layers that further summarize feature maps. They can be trained via backpropagation based on gradient descent methods. They can be used for a variety of tasks. The output from the last layer can be flattened into one-dimensional vector form corresponding to classifications learned during training by connecting fully connected dense layers at the end. They can also be used for various other tasks such as object segmentation, bounding box detection, feature extraction, super-resolution, etc.

Given that Convolutional Neural Networks have demonstrated state-of-the-art performance in conventional computer vision tasks such as object classification, detection and segmentation across various domains including medical imaging diagnosis and self-driving cars among others makes it a highly useful tool.

2.2.3 Transformers and Attention Mechanism

The Transformers architecture, a state of the art development in natural language processing and machine learning, originally designed for sequence-to-sequence tasks like machine translation, was introduced by Vaswani et al. in their 2017 paper titled "Attention is All You Need" [17]. This innovative architecture has substantially enhanced the performance of numerous tasks, such as machine translation, text summarization, and sentiment analysis, by effectively capturing long-range dependencies and contextual information within the input data. The original architecture is shown in Figure 1.

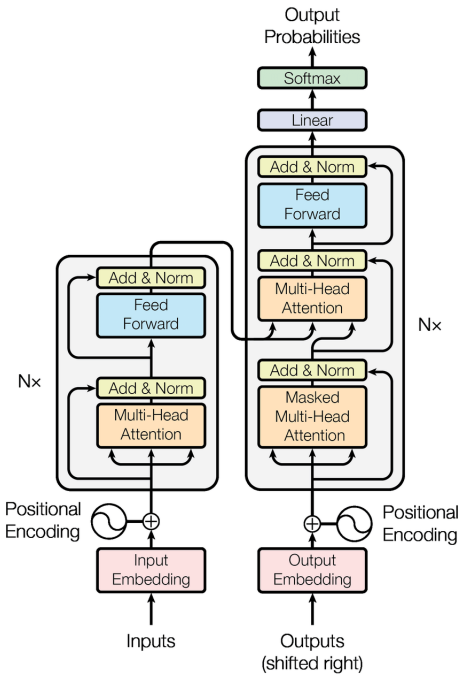


Figure 1: Original Transformer Architecture

A Transformer's architecture comprises two main components: the encoder and the decoder. Both components consist of multiple layers, each containing a multi-head self-attention mechanism, followed by position-wise feed-forward networks. Furthermore, residual connections and layer normalization are utilized throughout the architecture to aid training and enhance model performance.

In the encoder, the input sequence is initially embedded into vectors, which are then combined with positional encodings to integrate information about each element's position within the sequence. This combination of token embeddings and positional encodings is subsequently passed through the multi-head self-attention mechanism, which calculates a weighted sum of the input elements based on their relevance to one another. The attention mechanism's output is then processed by the position-wise feed-forward networks, and the resulting representation is forwarded to the next layer in the encoder. This procedure is repeated for each layer in the encoder, progressively refining the input representation.

Conversely, the decoder is responsible for producing the output sequence. Like the encoder, the decoder also employs multi-head self-attention mechanisms and position-wise feed-forward networks within its layers. However, the decoder incorporates an additional attention mechanism that focuses on the encoder's output, enabling it to integrate information from the input sequence when generating the output. This cross-attention mechanism

allows the decoder to concentrate on the most relevant parts of the input sequence while producing each element of the output sequence.

At the heart of the Transformers architecture lies the attention mechanism, which enables the model to assign importance to different input elements when generating an output. This mechanism allows the model to concentrate on the most pertinent parts of the input sequence, thereby improving its capacity to comprehend and process intricate structures. The attention mechanism is founded on the idea of scaled dot-product attention, which calculates the similarity between input elements using their dot product. This similarity is then scaled by the square root of the input dimension and passed through a softmax function to obtain attention weights. These weights are employed to compute a weighted sum of the input elements, effectively capturing the contextual information and dependencies between them.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2}$$

Where Q, K, V refer to queries, keys and values matrices, d_k is the dimension of the key matrix. Furthermore, a multi-head attention strategy can be applied within the architecture. The multi-head attention approach performs the attention mechanism multiple times, concatenates the output and linearly projects it to the expected dimension, an operation that can be performed in parallel leading to an increased training speed. This is accomplished through the utilization of several attention heads, each concentrating on distinct features of the input, thus permitting the model to develop a more comprehensive understanding of the input data. By employing multiple attention heads, the model has the potential to identify various kinds of relationships between input tokens in the input sequence.

$$\begin{aligned} MultiHead(Q, K, V) &= Concat(head_1, ..., head_h)W^O \\ where \quad head_i &= Attention(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{3}$$

Where each head represents a subspace in the embeddings dimension projected h times, where queries, keys, and values are multiplied by the respectively learnable parameters W^Q, W^K and W^V , and the resulting matrices are concatenated and multiplied again by W^O to yield the final values. The Transformers architecture can utilize the attention mechanism to efficiently process and understand complex input sequences by focusing on the most relevant portions of the input data. The integration of multi-head self-attention, position-wise feed-forward networks, and cross-attention mechanisms in the encoder and decoder components enables the model to capture long-range dependencies and contextual information, resulting in significant advancements in various machine learning tasks.

2.3 Multi-view Learning

Multi-view learning is a method to leverage multiple perspectives, or "views" of the data to improve the overall learning performance. This can be especially useful when dealing with complex or high-dimensional data. The main idea is that the different views of the data can contain complementary information, which, when fused together, can lead to better representations and, consequently, better predictions. The process may be summarized in following steps:

- The first step is to gather data from multiple sources, which can be different inputs, different modalities (e.g., text, images, audio), or even different feature representations extracted from the same source.
- Turning the input data into multiple views according to the available information. For instance, if a dataset contains images and corresponding text descriptions, the image pixels and text descriptions can be treated as separate views. In case of signals like EEG, it can be time-domain representation and its transformed frequency-domain representation.
- Employing techniques to produce an appropriate representation for each view. This process might involve training a convolutional neural network (CNN) for images and a recurrent neural network (RNN) or Transformer for text and other types of data. The objective is to extract pertinent features from each view for subsequent learning tasks.
- Integrating the learned representations from each view to produce a comprehensive representation. Various methods can be used to achieve this, such as concatenation, weighted sum, or more sophisticated approaches. The aim is to utilize the supplementary information from each view to create a more informative and robust representation.
- Employing a deep learning model with the unified representation as input. This step can involve any supervised or unsupervised learning task, including classification, regression, clustering, or contrastive learning approaches.

Figure 2 shows an example of a multi-view learning framework which takes into account and feeds different views of an entity into separate networks in order to learn good representations for each view, and to carry out a learning task. First example shows an image and its depth map as two views, while second example shows a time series and its respective spectrogram as two separate views.

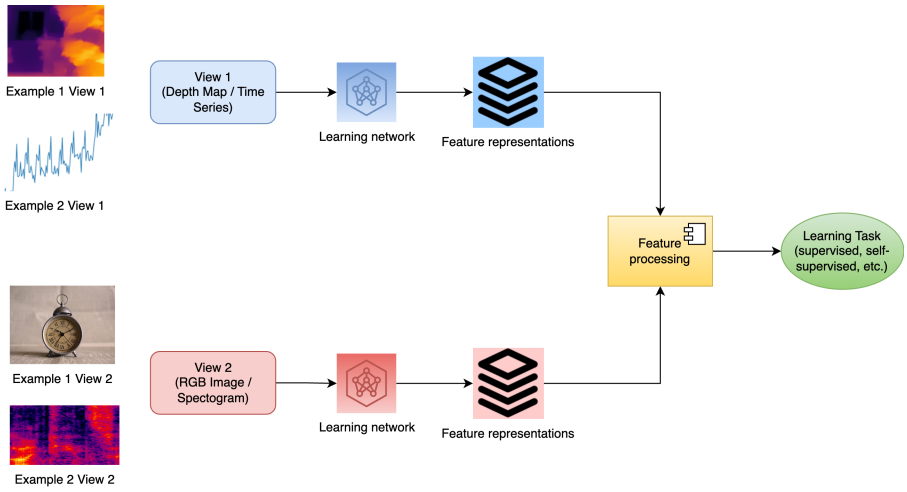


Figure 2: Multi-view Learning

2.4 Contrastive Learning

In the field of artificial intelligence, specifically within the domain of self-supervised learning, contrastive learning is a method that allows machines to derive meaningful represent-

ations from data without depending on explicit labels. Self-supervised learning, a subset of unsupervised learning, is driven by the intrinsic structure of data itself rather than human-generated labels. This technique has managed to gain considerable interest due to its capacity to utilize large quantities of unlabeled data, which is readily available in the real world.

The primary objective of contrastive learning is to develop a representation capable of distinguishing between similar and dissimilar data points. This is accomplished by training a model to identify and differentiate positive and negative pairs of data samples. Positive pairs consist of two instances of the same data point, while negative pairs are made up of two distinct data points. By learning to recognize these pairs, the model can effectively understand the underlying structure and patterns within the data.

Contrastive learning generally involves the following stages:

- Data Augmentation: To generate positive pairs, original data samples undergo various augmentation techniques, such as adding noise, rotation, scaling, or cropping. This results in multiple versions of the same data point, which are treated as positive pairs.
- Encoder Network: An encoder network, typically a deep neural network, is employed to map the augmented data samples into a latent space. The aim is to learn a representation where similar data points are close together, and dissimilar data points are far apart.
- Contrastive Loss Function: A contrastive loss function is used to assess the similarity between the representations of positive and negative pairs. As mentioned, the goal is to minimize the distance between positive pairs while maximizing the distance between negative pairs. Widely used contrastive loss functions include triplet loss [18], N-pair loss [19], and InfoNCE loss [20].
- Optimization: Gradient-based optimization techniques, such as stochastic gradient descent or adaptive optimizers like Adam, are used to train the model to minimize the contrastive loss function. This process updates the encoder network's weights, resulting in better data representations.

Contrastive learning has demonstrated promising results in various AI applications, including computer vision, natural language processing, and reinforcement learning. By leveraging the power of self-supervised learning, contrastive learning allows models to learn from vast amounts of unlabeled data, potentially leading to more robust and generalizable AI systems.

Figure 3 shows an example of a contrastive framework being applied to a dataset consisting of time series and spectrogram data points, in which a contrastive loss function attempts to pull the representations of two variations of the same data point closer, while pushing away the representations of the variations of different data points.

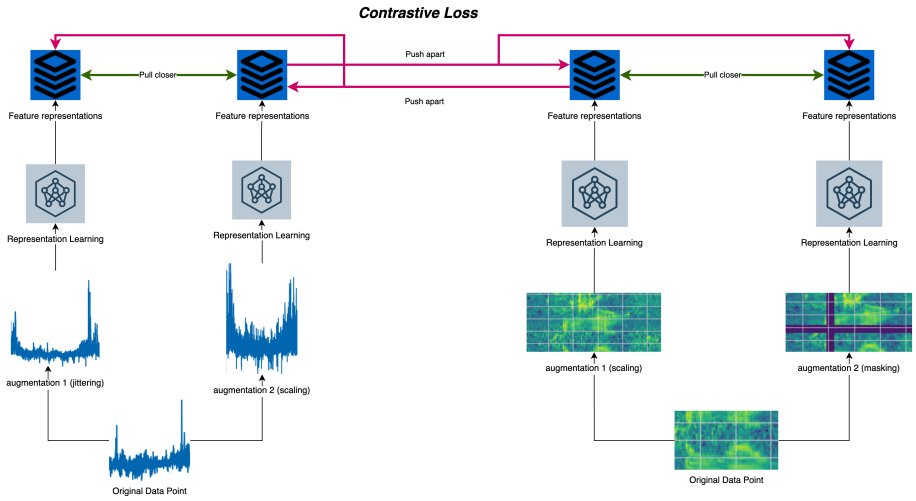


Figure 3: Contrastive Learning

2.5 Data Augmentation

Data augmentation is a commonly used method in machine learning, especially in computer vision, to expand and diversify a dataset by applying various transformations to the original data. This approach aids in enhancing the performance and generalization abilities of AI models by supplying them with a broader range of representative training samples. In the domain of contrastive learning, data augmentation is essential for deriving significant and reliable data representations and is heavily influential in the final outcome [21].

As mentioned, contrastive learning seeks to learn valuable representations by comparing and contrasting data samples. The primary concept is to guide the model to generate similar representations for semantically related samples and dissimilar representations for unrelated samples. This is accomplished by devising an appropriate loss function that encourages the model to minimize the distance between positive pairs (similar samples) and maximize the distance between negative pairs (dissimilar samples).

Data augmentation is particularly vital in contrastive learning for the following reasons:

- Producing positive pairs: Data augmentation methods, such as rotation, scaling, flipping, and jittering, can generate multiple transformed versions of the same data. These augmented data can be regarded as positive pairs since they possess the same semantic content. By comparing these positive pairs, the model learns to identify the underlying structure and invariances in the data.
- Strengthening negative pairs: Besides generating positive pairs, data augmentation can also assist in creating more diverse and challenging negative pairs. By applying various transformations to unrelated images, the model encounters a broader range of variations, making it harder for the model to differentiate between positive and negative pairs. This forces the model to learn more discriminative features and representations.
- Regularization: Data augmentation serves as a form of regularization, preventing the model from overfitting to the training data. By introducing variations in the

input data, the model is forced to learn more general and robust features that can better generalize to unseen data.

- Enhanced performance: Data augmentation is also employed to boost the performance of contrastive learning algorithms. By providing the model with more diverse and representative training samples, the model can learn more meaningful and transferable representations, which can be advantageous for downstream tasks such as classification, detection, etc.

3 Chapter III - Related Work

There have been numerous studies on the processing of EEG data and extracting useful features from them, and different ways they can be used in a predictive model to achieve a task such as classification or regression. In this section, a brief review of the related body of research is presented.

3.1 Extraction of Features from EEG Data

Liu et al. [16] used features from both time and frequency domains and after preprocessing, fed each domain separately into convolution layers and then fed the concatenated output to feed-forward layers and classification layers respectively. In this specific case, the representation of each view is being learned separately for each view, as opposed to jointly based on a shared cost.

In [22], the authors transformed original time-series EEG data into frequency domain, and they also extracted some statistical data from frequency domain as an additional domain. They then built a latent intact representation from all three views, namely time, frequency and frequency statistics, after applying dimensionality reduction on time and frequency views. They finally applied a support vector machine (SVM) classifier to perform the prediction on final feature representations.

Tang et al. [23] proposed a multi-view method which extracts three feature groups from the time-series data after applying wavelet packet transform, namely local fractal spectrum, relative band energy, and synchronization modularity features to be used as a different view of the input data. These features are then fed separately into convolution layers (depth-wise and point-wise), and then concatenated and fed to attention, gated recurrent unit (GRU), fully connected and a softmax classification layer respectively to predict a binary class (seizure vs non-seizure period).

In [24], Chen et al. used a different set of features, namely frequency domain features and brain connectivity features. The brain connectivity features, which replaces the time domain features in some other studies, can introduce a new type of information to the set of features. In this study, they preprocessed data and extracted several types of features, including phase lag index (PLI) and phase lock value (PLV) from connectivity domain, and power spectral density (PSD) and differential entropy (DE) from frequency domain. They used different fusion methods on different combinations of these features and found some of them outperform the others, especially approximate empirical kernel map or AEKM on . Finally, they applied a classical support vector machine classifier to predict different emotion states from those extracted features.

In conclusion, using a combination of parameters and features can in a lot of cases lead to more reliable results. This combination can help capture the most valuable and useful patterns from raw EEG data in a comprehensive way and it can contain frequency domain features, time domain analysis, fractal or entropy values, etc [25].

3.2 Automatic Feature Extraction from Raw EEG Data

The authors in [26] extensively used common spatial pattern for feature extraction and a support matrix machine for classification in stacked layers which could recursively en-

code the representations from previous layers to output the final representations. Their approach showed promising results over the decoding of EEG patterns to perform the task of classification.

In [27], the authors extracted several aggregated features in the form of images and applied multiple pre-trained deep learning models as feature extractors and then fused the extracted features togethers. Finally, with frozen feature extraction layers, they applied fully connected layers for training on top of the model to perform the final classification task.

3.3 Learning Representations from Unlabeled EEG Data

EEG data can contain valuable information that can be extracted with state of the art algorithms, but it is usually dependent on a huge volume of labeled data to work with, a data that is scarce and difficult to gather. Self-supervised learning methods mitigate this problem by developing models that can learn how to transform and represent EEG data in a way that would be much easier for a subsequent supervised model to fine-tune and map those representations to the desired output.

It has been demonstrated that self-supervised learning approaches can lead to better performance in downstream supervised tasks, especially in cases where labeled data is scarce and hard to obtain, but it is also dependent on the self-supervised tasks (e.g., sampling by relative positioning of the original time-series data) as well as the models that are used to learn representations based on those self-supervised tasks [28].

Kumar et al. [29] used various augmentation methods for self-supervised tasks, including jittering (random uniform noise) and random masking of signals as a type of augmentation, along with random horizontal flipping and scaling as another type of augmentation. They used time-domain and spectrogram views as two views of the EEG data. Data is segmented into a 30-second interval (epoch) with each interval belonging to a category. The similarity between augmented epochs are maximized, whereas the similarity between the current epoch and other different epochs (in the pretext group) are minimized in a batch. For each epoch, two types of augmentations are applied, the spectrogram view is created, each view goes through an encoder, and then a projection. Three groups of features are created: time-series features, spectrogram features, and concatenated features. For each pair of feature groups in a pair of samples, three contrastive losses are calculated, one for time-series pair, one for spectrogram pair, and one for concatenated pair. Another loss is also presented which pushes for the complementary information present in each of the views.

In [15], the authors tried an "Attention-based Recurrent AutoEncoder" to learn feature representations in the unsupervised part of the work which resulted in promising results. This way, there is no need to define different self-supervised tasks anymore in order to learn representations and to find good initial weights for the supervised part of the work. The autoencoder is first trained in an unsupervised manner by trying to reconstruct the input data, and then the encoded features are fed to a supervised model to perform a downstream task which is emotion classification in this case.

3.4 Using Multiple Views of EEG Data

Real life events can be captured or represented using different formats and can be viewed through different lenses. Although it is possible to sometimes capture data for an event from different views or transform an existing dataset to represent a different aspect of the data, it is usually a challenge to use them in a way that captures useful data representations for other downstream tasks.

A common way to extract useful representations from different views is to find a way to encode the input from each view and map the inputs into a common representation space in which the mutual information from different views are maximized [30], [31].

There have been various research studies that have leveraged multi-view settings in order to use EEG data to carry out a downstream task such as classification. In [32], the authors considered each channel's data in 30-second windows and calculated the average power of the theta band (4-7.5) Hz, and used it as one of the views. For the second view, they converted these band powers into dB's and used it as the second view of the input data. They build a fuzzy system to use both views to carry out the task of drowsiness estimation.

In [33], the authors used two views to learn the representations: raw EEG data and the spectrogram data obtained from Hilbert-Huang Transform. They used a method for self-supervised learning called Dense Predictive Coding that predicts future feature representations based on the past features, and uses a loss function to maximize similarity with the true representation (and similar positive representations) at a given time period and minimize it compared to all the other representations. Similarly, authors in [29], use time-series and spectrogram views to learn EEG representations, but they use augmented views of those views to compare with ground truth and optimize the loss function. They used true and augmented views of time-series, spectrogram and the combination of time-series and spectrogram features.

In [16], the authors used the time-series EEG data and its frequency domain version using Fast Fourier Transform (FFT) as two views. They used convolutional neural networks to encode and learn representations from each view separately and then used fully connected layers to merge them and learn a shared representation and subsequently used a classification layer for a downstream seizure prediction task. In [22], the authors used three views: time-domain view on which Principal component analysis (PCA) was applied, frequency-domain using FFT and then PCA, statistical features from the time-domain (median, mean, mode, etc.). They then used intact space learning and applied a support vector machine (SVM) on the final feature space to perform a Tinnitus classification task.

The authors in [34] used multi-view learning for motor imagery task recognition. They used three different views of the EEG data, time domain view, frequency domain view, and time-frequency domain view using wavelet packet decomposition (WPD). They applied a feature extraction algorithm called common spatial patterns (CSP) on top of all views to extract spatial domain feature representations. A deep restricted boltzmann machine (RBM) was used in combination with a dimensionality reduction method to learn the features in multiple views while removing the redundant features. A SVM was finally applied to carry out the prediction task.

Multi-view learning can also be applied for feature selection and dimensionality reduction in the case of small datasets, in order to find useful features of those datasets and remove the redundant features [12].

3.5 Using Transformers to encode EEG data

Transformers, initially introduced by Vaswani et al. in 2017 [17], have resulted in significant advancements in the domain of natural language processing (NLP) and have been effectively employed in diverse fields such as computer vision, speech recognition, and bioinformatics. Recently, there have been increasing efforts for investigating the capabilities of Transformers in the context of EEG data, encompassing tasks like seizure identification, cognitive state classification, and brain-computer interface (BCI) implementations. In this context, Transformers offer advantages such as good handling of long-range dependencies and have the potential to significantly improve the performance of EEG data analysis. This section briefly mentions several attempts to use Transformers in EEG-related tasks.

In a 2021 article [35], Kostas et al. attempted to make a pre-trained model with the ability to be applied (via fine-tuning) to various downstream supervised tasks. Their approach involves using a convolution-based features encoder that helps with reducing the dimension of input data and producing embeddings. The results then go through a Transformer encoder block to produce the final output. For the pre-training task, a masking is applied to the input and the model tries to predict the unmasked input. A contrastive loss tries to minimize the distance of the predicted output and the original unmasked input, while maximizing the distance with a batch of other negative samples. Their architecture closely follows that of wav2vec 2.0 [36] by Baevski et al. which in their case was successfully implemented to learn useful features from speech audio and serves as a pre-trained model for encoding speech features.

Wei et al. proposed a model for emotion recognition in [37]. The input is segmented into 1-second intervals, baseline gets removed and a wavelet transform is applied, resulting in an output with frequency, channel, and time dimensions that is then fed to the model. Their model architecture includes three parts: (i) A partitioning section which divides the input into non-overlapping patches through convolution operations, (ii) a Transformer section which is responsible for capturing global relationships, (iii) and an emotion classifier which is used to identify the emotion.

In [38], Song et al. paid attention to both channel and time dimensions. After spatially filtering and preprocessing the initial EEG data, they apply the attention mechanism on the feature channels to find a way to score the features in that dimension, and they subsequently apply attention globally to extract the more suitable features, and finally use the features for a classification task. In another successful attempt in [39], in order to capture better initial features from EEG data, Song et al. use convolutions more extensively in initial layers in both time and channel dimensions to encode the data and extract useful local feature maps, and then fed those features into Transformer layers which used attention mechanism to encode more global features within the data.

4 Chapter IV - Implementation and Architecture

In this section, we will present the step by step implementation details of our proposed method. As mentioned, the primary goal is to find useful representations of EEG data in a self-supervised way, with the final goal of using those representations or the pre-trained weights of the model in a downstream task like predicting dementia. The first step is finding the right set of datasets and preparing them for the task.

4.1 Datasets

We used different datasets for each part of this study. For pre-training, we used two main datasets:

- The Temple University Hospital (TUH) dataset [40]: TUH dataset is originally the result of 14 years of clinical records that have been collected from Temple University Hospital and have been made publicly available after curation and processing, along with textual notes that have been paired with the EEG data. It comprises many different frequencies and channels, with the majority of data having 31 channels and a sampling rate of 250 Hz. We used a version of the dataset that divided samples between normal and abnormal classes, with the abnormal class representing different abnormalities.
- The NMT sculp dataset [41]: Originally consists of 2417 samples which covers around 625 hours. NMT dataset presents a unique dataset representing a south-asian population. For the data collection, they used the standard 10-20 electrode placement system with 19 EEG channels on the scalp and two reference channels near the ear. The sampling rate and average duration for the samples are 250 Hz and 15 minutes, respectively. Similar to the TUH dataset that was used in this study, this dataset also consists of normal and abnormal labels, with the abnormal referring to different types of abnormalities and pathological conditions.

Table 1 shows the size of the datasets used in this research in summary.

Table 1: Summary of the datasets

Dataset	Labels		Total
	Normal	**Abnormal**	
TUH	1521	1472	2993
NMT	1869	398	2267
Total	3390	1870	5260

4.2 Data Preprocessing

We have applied several preprocessing steps to make sure the input data to the model is in the right format. The following transformations were applied to the datasets [42]:

- Adjust recordings to a consistent 20-minute duration.

- Drop samples that were too short for our use case.
- Choose a common subset of 19 channels.
- Limit signals to a range of ±800 V.
- Reduce signal's sampling rate to 100Hz through downsampling.
- Implement 1-40 Hz bandpass filtering to minimize noise and distortions.
- Employ a sliding window approach with 5-second overlapping intervals, creating a (19x500) window dimension.
- Rearrange EEG signals into a tensor with dimensions 100 x 19 x 500. We can consider this tensor as 100 frames of 19 x 500 images.
- Conduct normalization for each channel to achieve a zero-mean and a standard deviation of 1.
- Current source density or Surface Laplacian (SL) was applied to provide estimates of the current flow around the scalp in order to have a more localized and accurate representation of the original EEG data [43].

We also created a pretext dataset from the original TUH and NMT datasets. The pretext dataset consists of all the data points (after preprocessing) in NMT and the majority of data points in TUH, which are exclusively used as pre-training data. The labeled part of data is only used for evaluation purposes. Table 2 shows the summary of the dataset after preprocessing.

Table 2: Summary of the datasets after preprocessing

Dataset	Labels		**Pretext**	Total
	Normal (train/eval)	**Abnormal (train/eval)**		
TUH	276 / 150	270 / 126	2171	2993
NMT			2267	2267
Total	276 / 150	270 / 126	4438	4560

4.3 Data Augmentation

As discussed in the previous section, data augmentation is a vital part of contrastive learning pipelines. In this study, two main groups of augmentations have been independently applied on the input data, each of those consisting of several augmentation steps. These augmentations served as the reference for comparing the final representations in contrastive learning. Each augmentation went through the pipeline and was applied on both time-domain and frequency-domain data, and in the end, the loss function was applied on the final outputs. We refer to the augmentation groups as type 1 and type 2 augmentations influenced by [29], [44].

Type 1 augmentation consists of the following steps:

- Jittering (noise): Similar to adding noise to images, we add an amount of noise to the input data. This noise includes a low-frequency component and a high-frequency component, and is applied to each channel independently.
- Masking: We choose a specific range that represents the number of points to be masked, in our case a random number between 10 and 30. Then, a place is randomly selected each frame in each channel and the points will be masked (replaced with zero).

Type 2 augmentation can consist of the following steps:

- Scaling: In scaling, the input data is multiplied by a factor that is based on samples drawn from a gaussian distribution (gaussian noise). This augmentation affects the signal's amplitude and introduces some amount of noise to it. This augmentation is also applied to each frame in each channel.
- Flipping (optional): With a probability of 50%, the order of frames is reversed. This augmentation did not yield helpful results, possibly due to adverse effects of breaking the temporal continuity, so it was not used in the final model run.

4.4 Model Architecture

In this section, we will discuss the structure of the model and the different constituent parts. Figure 4 shows the overall architecture of the model and the relevant pipelines. We will discuss each step in the following sections.

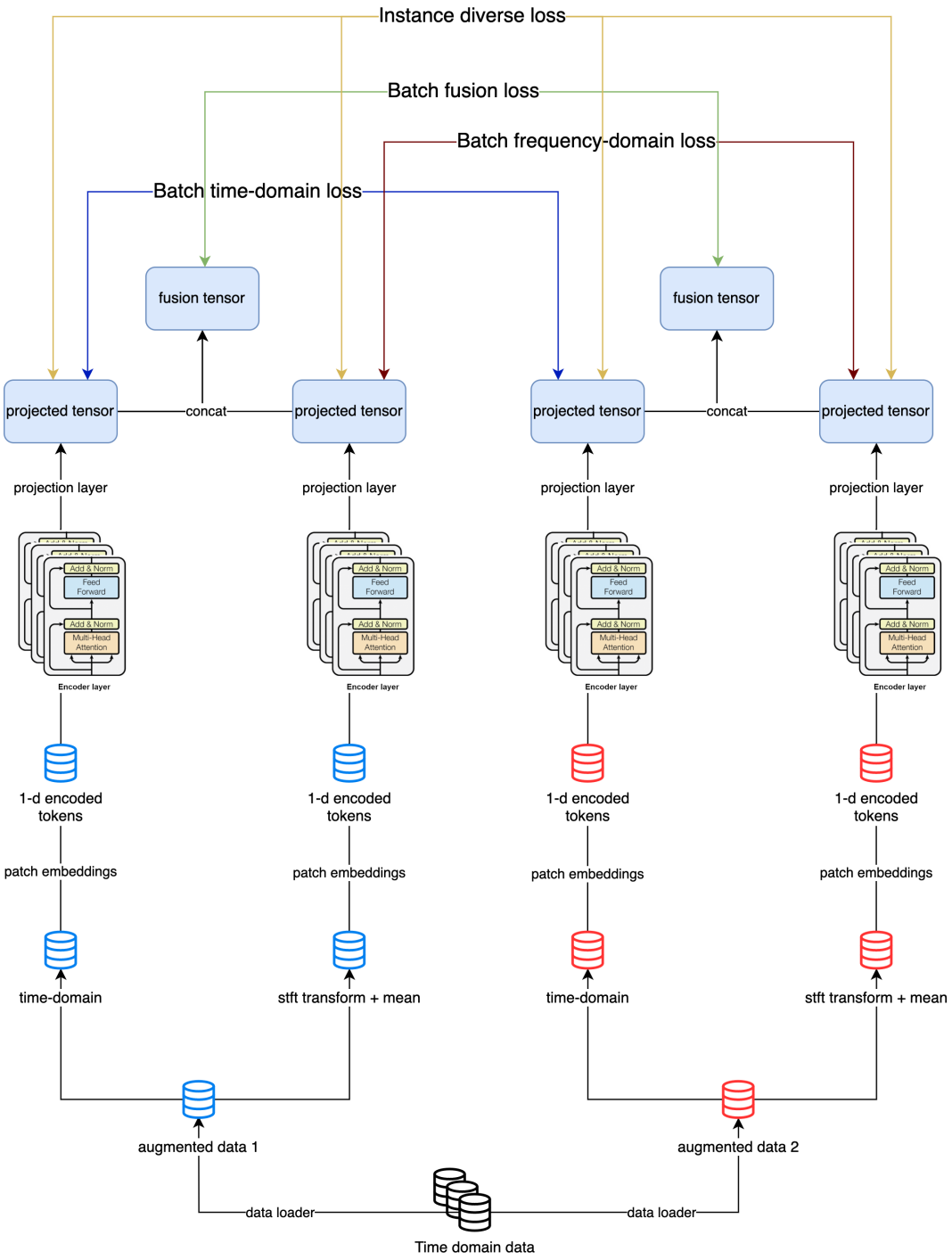


Figure 4: Model Architecture

4.5 Creating Multiple Views of EEG Data

The input data from the datasets are in time-domain. Using a short-time fourier transform, the input time-domain data is transformed into its spectrogram representation in frequency-domain. The resulting time-domain and frequency-domain data is then used as input to the subsequent model. Figure 5 shows the transformation from time to frequency domain.

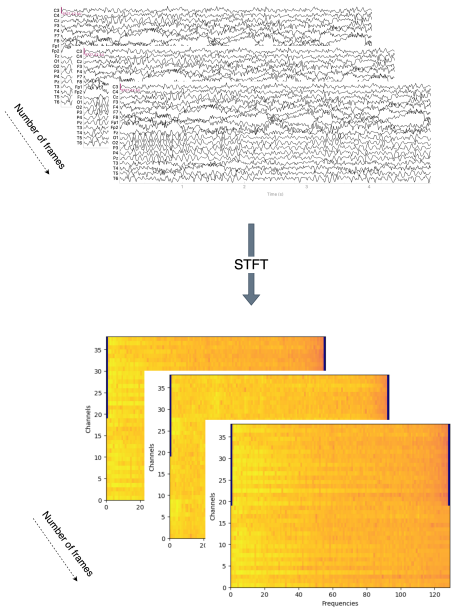


Figure 5: Creating frequency-domain view (averaged over time in each frame) from time-domain view

The transformation is applied separately to each input channel and each frame, and the resulting real and imaginary parts are concatenated, hence we have $2 \times channels$ in the channel dimension. As the result, since a new dimension will be added to the input data which is the frequency dimension, we average the values across time points within each frame. The resulting tensor will have the shape of $(batches \times frames \times 2 * channels \times frequencies)$.

4.6 Patch Embeddings

Inspired by Video Transformers, producing some sort of input embeddings is considered as one of the useful ways to encode input data into a format suitable for Transformers. There are several approaches to produce embeddings from the raw input. Large embedding networks (2D or 3D CNNs) can be leveraged to produce these embeddings, either used as pre-trained or end-to-end. However, end-to-end training with large networks can become very compute-hungry and pre-trained models for EEG data are not generally available as for image data, so a good alternative is to use small convolutional or linear layers, and train them end-to-end with the Transformer module (minimal embeddings) [45].

We can consider the input to our model as 100 frames, with each frame containing 19 channels by 500 points for time-domain, and 19 channels by 129 points for frequency-domain. In the first step of the architecture, we create patches (tokens) of the input data, which serve as a dimensionality reduction component and also a bottleneck to filter useful information from the input data.

A dimension is added to each frame which will later serve as the embedding dimension. We create initial feature maps, 40 in our experiments, and then apply two convolutions which serve as spatiotemporal filters, inspired by [39]:

- One convolution over the time (or frequency) points, that serves an aggregating function, across the whole frame.

- One convolution over the channels, that serves as a channel aggregating function, which summarizes the information across the channels at each time point.

These convolutions are followed by a batch normalization, a non-linear activation function (ELU), and an average pooling component that can serve as a smoothing layer that helps with both overfitting and computational complexity of the model.

The output of these layers (patch embeddings) are flattened and then fed to a projection layer, which is used to increase the size of the final patch embeddings to our desired size. Figure 6 shows the main steps in the pipeline of patch embeddings for the time-domain data as an example.

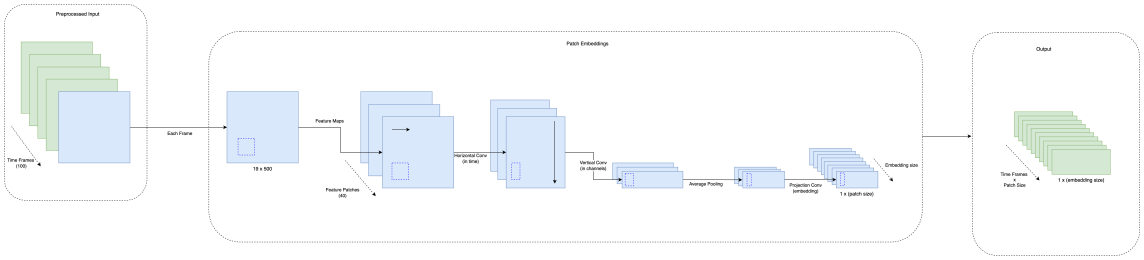


Figure 6: Patch Embeddings

4.7 Positional Embeddings (Time Embeddings)

The Transformer architecture typically employs certain markers to encode and keep track of the position of the tokens during the transformation processes. These markers play a role in maintaining the order and context of the input data when dealing with sequential information. Taking inspiration from some of the video Transformers available in the literature [46], we have opted to apply positional embeddings as well, focusing on the frame dimension in order to effectively keep track of the temporal direction of the frames. As a result, we refer to these embeddings as time embeddings.

Time embeddings, which are initialized as learnable parameters, serve as a component in preserving the sequential information within the data. It is important to note that these time embeddings are added equally to all the samples and patches in a batch and are only added in the dimension of the frames and not in the dimension of the patch embeddings, ensuring that the temporal context is maintained without interfering with the locally encoded information.

To implement this, firstly, the output obtained from the convolutions and projection operations in the previous layer is transposed. This is achieved by switching the patch and frame dimensions, effectively reordering the input data for further processing. The time embeddings are then added to the frame dimension, which allows the model to maintain an understanding of the temporal context throughout the transformations. Finally, the frame and patch dimensions are concatenated, resulting in a tensor that contains three dimensions - batch, token, and embeddings. This output tensor resembles the 1D token embeddings commonly used in language models, as shown in Figure 7 The general assumption here is that the model would be better equipped to handle both smaller local and longer temporal aspects of the data, ultimately leading to improved quality of the resulting embeddings.

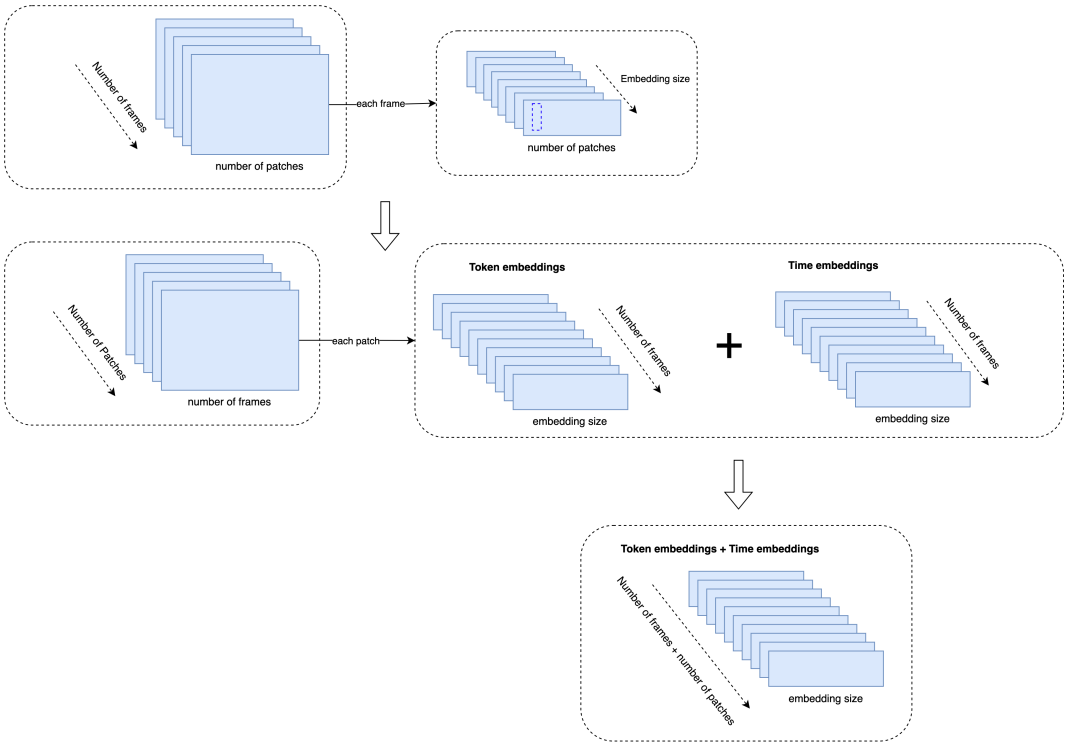


Figure 7: Adding time embeddings to the frames

4.8 Transformer

As discussed, the input EEG data is encoded into token embeddings for both time-domain and frequency-domain. As a result, the initial dimensions of the input which was $100 \times 19 \times 500$ (and $100 \times 19 \times 137$ for frequency-domain) representing number of frames, number of channels, and number of time points in each frame respectively, is turned into a sequence of tokens, with each token having a dimension of the chosen embedding size.

Transformers have shown remarkable performance in encoding sequence data into useful representations. In this research, we also make use of a Transformer to encode the resulting token embeddings of the initial EEG data. Since we are only interested in encoding and extracting representations, we use the encoder part of the Transformer with multi-head attention mechanism as shown in Figure 8.

As it is shown, after applying the positional embeddings (time embeddings) to the pre-processed tokens, a multi-head attention layer, which was described in previous sections, is applied and followed by a fully connected layer. Dropouts and layer norm blocks are also applied between the steps to enhance training stability and regularization. The input of each block is also directly added to the output as residual in order to help preserve the information in the input and maintain the useful information in the layers.

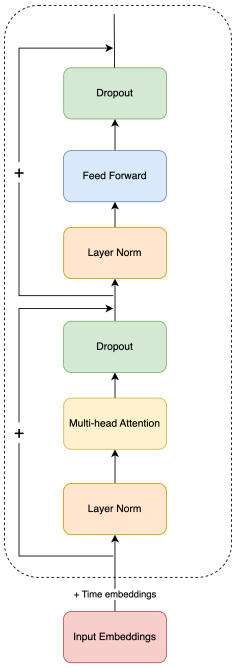


Figure 8: Encoder-Only Transformer

4.9 Projections and Embeddings

After applying patch embeddings and Transformers, the output from the Transformer module is fed into a final fully connected layer to extract representations. A global average pooling is applied on the output of the previous layer which serves as a dimensionality reduction component to reduce computational complexity. Finally, After layer normalization and dropout, a linear projection layer is applied to extract the final representation that will later be used for training and evaluation.

Figure 9 shows the final projections that have been applied to obtain the output representation in each domain.

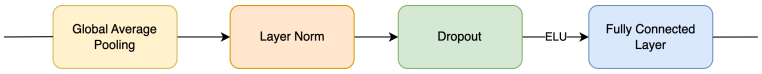


Figure 9: Embedding Layer

4.10 Loss Function

As discussed, contrastive loss is a popular method to learn invariance between multiple views of the same data, or positive samples, as opposed to negative samples. Since it is building global level representations of the input, it can be considered an instance-based or instance-level method that is good for capturing global relationships within the input data [45], [47]. Hence, we also use contrastive learning on top of Transformers to capture global and long-range relationships and representations from the input EEG data.

At this point in the architecture, we have mainly four output representations from the input data, coming from the two augmentation types and two domains (time and frequency). In order to make the most use of both views, we also make a combined representation using

the concatenation of time and frequency domain representations, so in the end we have six different output representations (embeddings):

1. Time-domain representation under type 1 augmentation.
2. Frequency-domain representation under type 1 augmentation.
3. Time-domain representation under type 2 augmentation.
4. Frequency-domain representation under type 2 augmentation.
5. Combined-domain (fusion) representation under type 1 augmentation.
6. Combined-domain representation under type 2 augmentation.

For each representation, we apply a simple projection head on top which turns them into lower dimensional outputs for better results [48], as shown in Figure 10 In order to compute the loss, we apply the contrastive loss over these projections, with each of them representing different augmentations of different views.

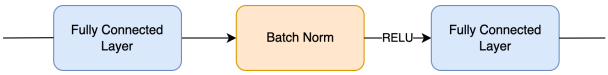


Figure 10: Projections over Representations

Similar to the steps outlined in [29], we use four components in the loss function. Three of the components are based on NT-Xent loss [48] with cosine similarity as the distance metric represented by equations 4 and 5. Each of these three components represent the loss being applied on each of the three domains (time, frequency and combined). The loss tries to maximize the pairwise similarity between two augmented views of each domain (in numerator), while minimizing the similarity between each view of a sample with the other samples in a batch (in denominator). This loss is applied three times, hence creating three loss components, time-domain loss, frequency-domain loss and combined (fusion) loss.

$$\ell(i,j) = -\log \frac{\exp\left(\cos\left(z_i, z_j\right)/\tau\right)}{\sum_{k=1}^{2N} 1_{[k \neq i]} \exp\left(\cos\left(z_i, z_k\right)/\tau\right)} \tag{4}$$

$$L = \frac{1}{2N} \sum_{k=1}^{N} \ell(2k-1, 2k) + \ell(2k, 2k-1) \tag{5}$$

Where i and j are the positive samples, z_i and z_j are subsequently the representations obtained from applying projection on each augmented view of a domain, N is the batch size, and τ is a temperature parameter. This is applied for each positive pair in both variations (i, j) and (j, i), in each mini-batch. Note that each sample has two augmentations, so there are $2N$ augmentations in total.

As the contrastive loss components above, especially the fusion loss, can push to optimize for the shared information between the two views, as recommended in [29], we use an additional loss term called Diverse Loss represented by equations 6 and 7, designed to leverage the integration of complementary details across time-domain and frequency-domain views for a single sample. It achieves this by applying contrastive loss to single

sample features from both time and spectrogram projections, drawing time-domain attributes nearer while distancing spectrogram attributes, and vice versa. Consequently, the resulting representations exhibit a variety of information from one another for each individual sample.

$$\ell_d(a,b) = -\log \frac{\exp\left(\cos\left(z[a], z[b]\right)/\tau_d\right)}{\sum_{i=1}^{4} 1_{[i\neq a]} \exp\left(\cos\left(z[a], z[i]\right)/\tau_d\right)} \tag{6}$$

$$L_D = \frac{1}{4N} \sum_{k=1}^{N} \ell_d(1,2) + \ell_d(2,1) + \ell_d(3,4) + \ell_d(4,3) \tag{7}$$

Where $z[i]$ denotes a feature representation of an augmentation. Numbers 1 through 4 in equation 7 which replace a, b and i in equation 6 denote the first four representations introduced in the beginning of this section for time and frequency domains which will be calculated for each sample separately, N is the batch size, and τ is a temperature parameter. Similar to the approach in NT-Xent loss, it is applied on both variations (a, b) and (b, a) for both domains.

$$L_{total} = \lambda_1(L_{TT}, L_{FF}, L_{SS}) + \lambda_2 L_D \tag{8}$$

Finally, L_{total} in equation 8 is the total loss in which L_D refers to the Diverse loss, and L_{TT}, L_{FF} and L_{SS} refer to the contrastive losses for the time, frequency, and combined domains respectively. Together with a linear combination, they constitute the total loss.

5 Chapter V - Evaluation and Discussion

5.1 Pre-Training Settings

There are a few hyperparameters and settings that should be set during pre-training on pretext data. Each of them might have a significant or small effect on the final quality and effectiveness of representations. Table 3 shows the parameters and their initial values that were set for pre-training. Some of the parameters were varied in several experiments, hence multiple values in some rows.

Table 3: Model Hyperparameters

Parameter	Value	Info
Embedding size	256 / 768 / 1536	Dimension of patch embedding tokens
Representation size	256	Dimension of the final representation for each domain
Projection size	128	Projection dimension for calculating loss function
Temperature	1	Contrastive loss temperature
Diverse Temperature	10	Diverse loss temperature
Optimiser	Adam	Optimisation algorithm
Learning rate	0.00001	Learning rate for Adam optimiser
Beta1	0.9	Parameter for Adam optimiser
Beta2	0.99	Parameter for Adam optimiser
Weight decay	3*(10^-5)	Parameter for Adam optimiser
Transformer layers	2 / 4	Number of Transformer layers
Attention heads	4 / 8	Number of attention heads in the Transformer
Scheduler	Cyclic / ReduceOnPlateau	Learning rate scheduler
Patience	5	Parameter for ReduceOnPlateau scheduler
Factor	0.2	Parameter for ReduceOnPlateau scheduler
Base LR	0.00001	Parameter for Cyclic scheduler
Max LR	0.01	Parameter for Cyclic scheduler

5.2 Evaluation Settings

As discussed, the outcome of pre-training is the weights of the model that can produce representations from a period of EEG input data. To evaluate the quality of representations, we follow the standard linear evaluation protocol which was used in numerous studies [29],

[48], [49]. In order to carry on the initial evaluation, we freeze the pre-trained weights of the model, add a linear classification layer on top, and train the classification layer of the model on labeled data, and finally measure the test accuracy. We use a binary classifier with cross-entropy loss for the classification task between normal and abnormal labels. We only used the time encoder representations to evaluate the results, but spectrogram or fusion representations could also be used. The simplified process is shown in Figure 11.

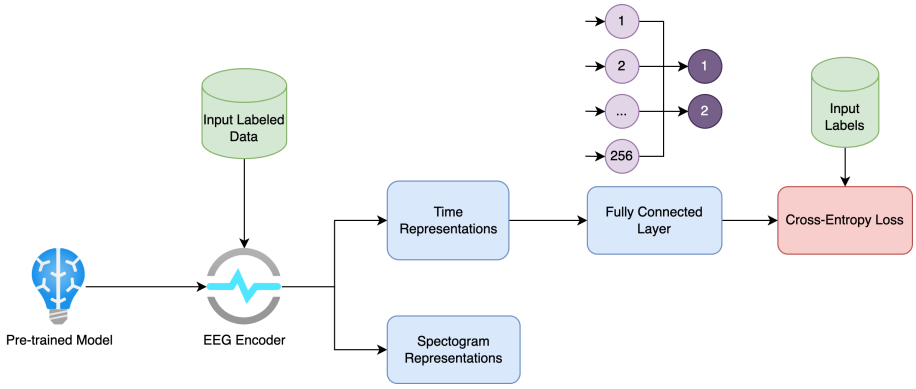


Figure 11: Linear Evaluation on Labeled Data

5.3 Results and Discussion

In the first step, we pre-trained the model on the pretext data. Due to computational requirements of pre-training and lack of sufficient resources, we only tried several sets of hyperparameters and settings that will be presented in this section. More comprehensive analysis and ablation study can be done to find the best set of hyperparameters, as well as most contributing factors.

We tried four different combinations pre-training settings:

- Augmentation includes frame flipping with 50% probability (reversing the order of frames), two Transformer layers, eight attention heads, ReduceOnPlateau learning rate scheduler, 32 batch size, and 256 embedding size
- Frame flipping, two Transformer layers, four attention heads, Cyclic learning rate scheduler, 32 batch size, and 768 embedding size
- No frame flipping, one Transformer layers, four attention heads, Cyclic learning rate scheduler, 16 batch size, and 1536 embedding size
- No frame flipping, two Transformer layers, four attention heads, Cyclic learning rate scheduler, 32 batch size, and 1536 embedding size

The resulting representations of each setting was evaluated using the aforementioned linear evaluation protocol on a labeled portion of data that was set aside for evaluation. The resulting representations along with a linear classification head was trained and evaluated on the labeled data in a supervised manner. The supervised evaluation is being applied every 50 epochs after epoch 100.

The first two settings did not yield promising results in the supervised evaluation. It seems the problem mainly arises from strong augmentation that uses frame flipping, hence

distorting the causal and time-dependent relationship between the frames in time, that the model can not learn from. Figure 12 shows the overall loss per epoch in pre-training and the test F1 score for linear evaluation for the first two variations.

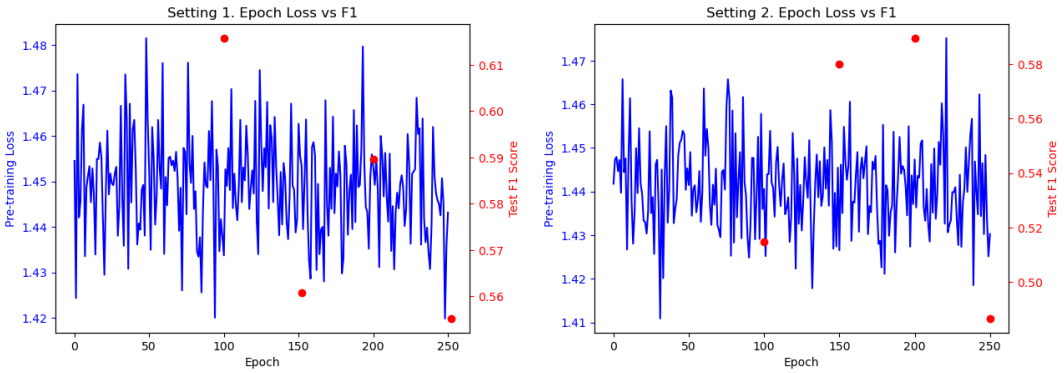


Figure 12: Pre-training epoch loss and test F1 score for Settings 1 and 2 with frame flipping augmentations in the first 250 epochs

On the other hand, the last two settings resulted in decreasing loss and improved evaluation scores. Figure 13 shows the loss per epoch and test F1 score for settings 3 and 4.

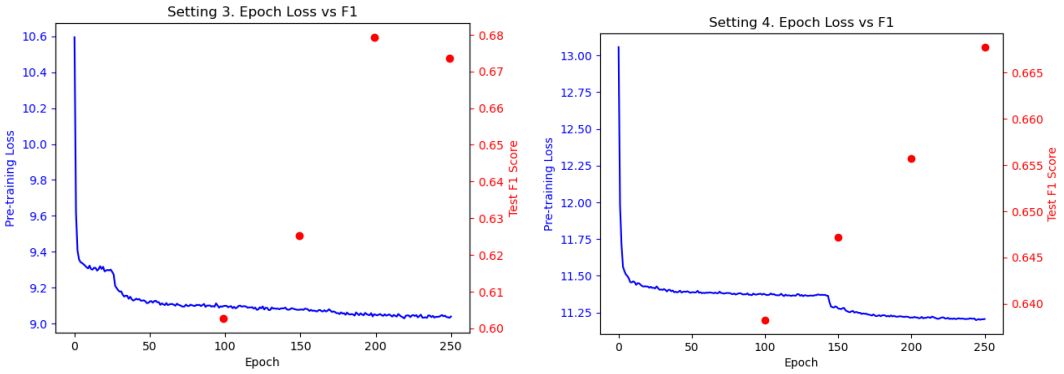


Figure 13: Pre-training epoch loss and test F1 score for Settings 3 and 4 without frame flipping augmentations in the first 250 epochs

We continued pre-training the model with setting 3 parameters for up to 1000 epochs to examine whether or not the metrics would continue improving. Figure 14 shows the full pre-training and F1 results for 1000 epochs. As it can be seen, the evaluations metrics, as well as epoch loss, showed continuous improvement.

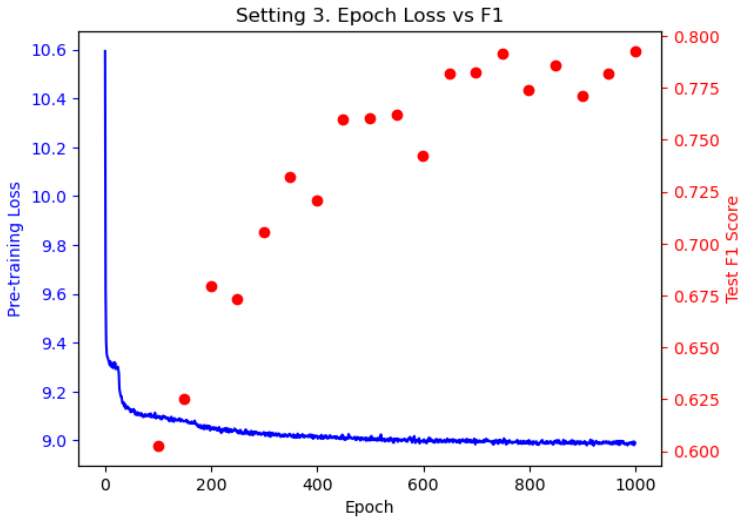


Figure 14: Pre-training epoch loss and test F1 score for Setting 3 for 1000 epochs

Additionally, to compare the performance of the multi-view model against its single-view version, we pre-trained the single-view model of setting 3, and compared the test accuracy of the linear evaluation for both of them. The single-view model only has the time-domain encoder, so we compared its pre-training loss only with the time-domain loss of the multi-view model. As it can be seen in Figure 15, the multi-view model outperforms the single-view version of itself in terms of test accuracy.

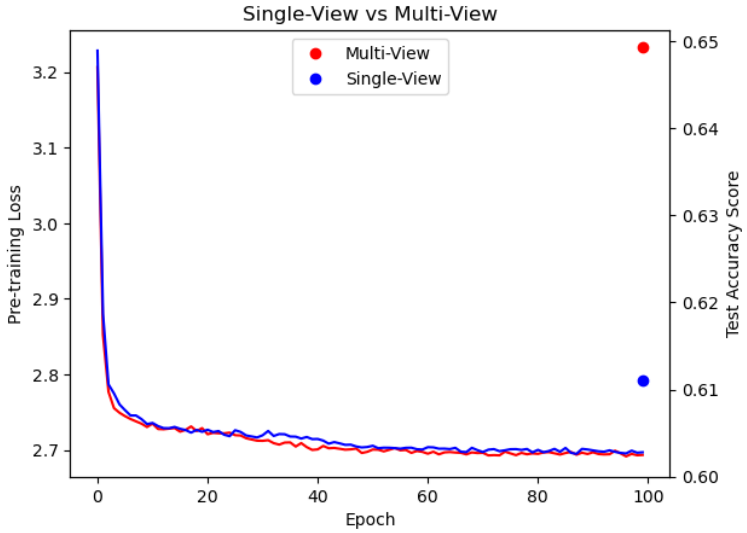


Figure 15: Pre-training time-domain loss and test accuracy score for Setting 3, Single-view vs Multi-view

The size of the final complete multi-view pre-training model with setting 3 is 58.345.920 parameters, and the size of the time encoder with setting 3 which we used as a pre-trained model for evaluations is 28.976.000 parameters.

In the rest of this section, we present more details about the multi-view model with setting 3. As mentioned, there are four different terms in the loss function, each responsible for one aspect of the learning, namely time-domain loss, frequency-domain loss, fusion-domain

loss, and Diverse loss. Epoch loss simply refers to the total loss obtained by adding all the terms. Figure 16 shows the evolution of different parts of the loss function during the pre-training of the model.

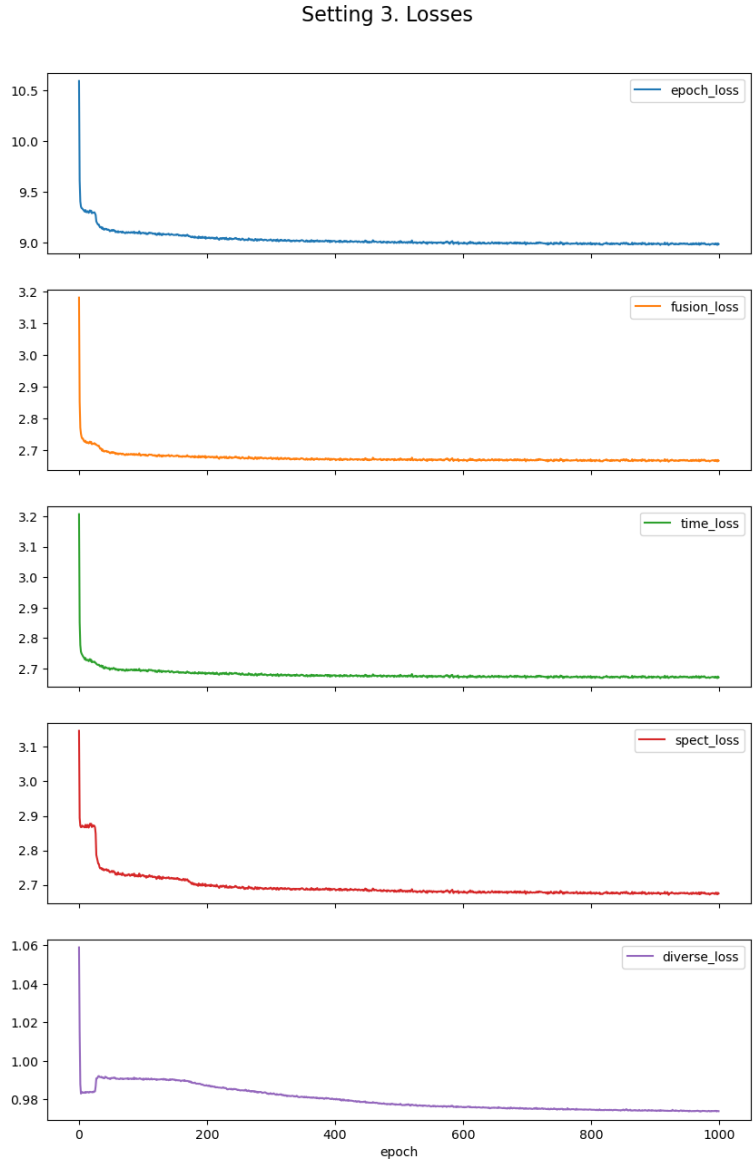


Figure 16: Evolution of the different components of the loss function during pre-training

In addition to losses, we also measure different metrics for the supervised evaluation that happens every 50 epochs. The measured metrics are maximum validation F1, average validation F1, maximum validation accuracy, maximum balanced validation accuracy (samples are weighted according to the inverse of the overall in-class frequency), and maximum Cohen's kappa score, measured across all supervised epochs. Figure 17 shows the supervised validation metrics measured on the labeled data.

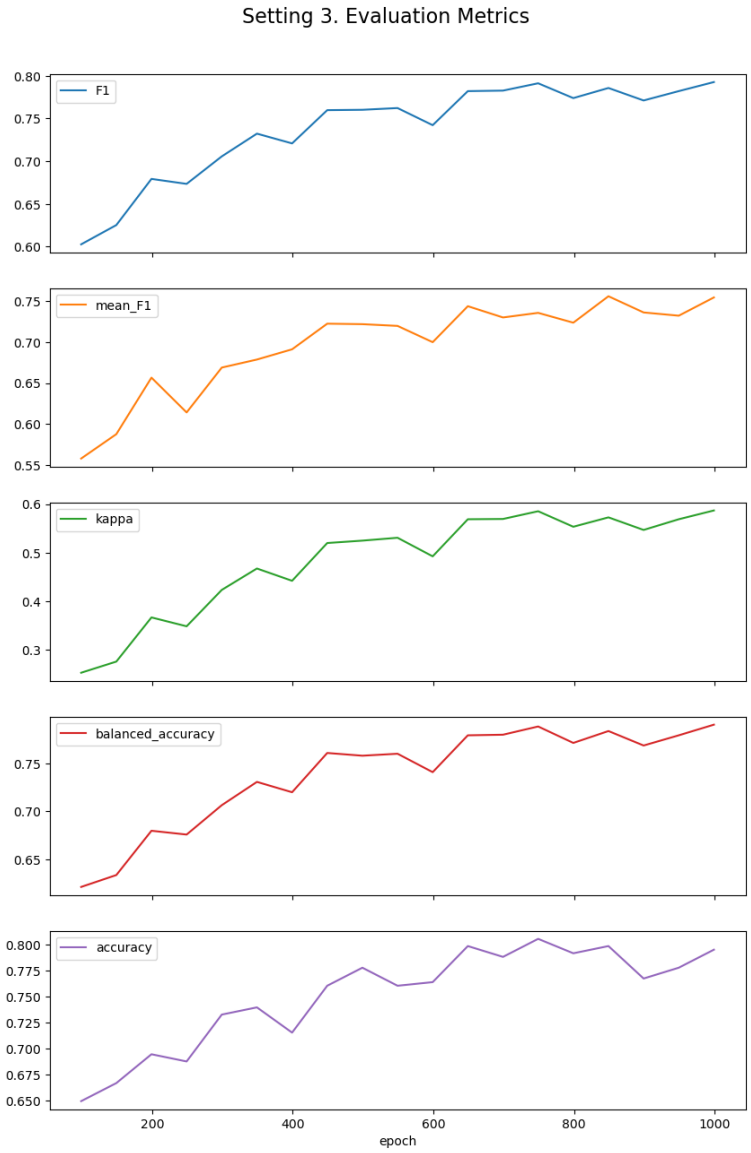


Figure 17: Supervised evaluation metrics measured on the validation portion of the labeled data

To validate the usefulness of the learned representation, we compared the results of four experiments:

- Linear evaluation on top of the frozen pre-trained weights of the model.
- Linear evaluation on top of the frozen randomly initialized weights.
- Fully supervised evaluation with the encoder and linear layer initialized with the pre-trained weights of the model
- Fully supervised evaluation with the encoder and linear layer initialized with randomly initialized weights.

Table 4 shows the comparison between the evaluation metrics.

Table 4: Comparison of supervised evaluation between pre-trained weights and randomly initialized weights

	F1	**mean F1**	**kappa**	**Balanced accuracy**	**accuracy**
Random + Linear Head	43.90	36.87	5.34	52.49	57.98
Pre-trained + Linear Head	79.27	75.58	58.69	79.04	80.55
Random + Fully Supervised	76.15	67.35	53.01	75.95	77.17
Pre-trained + Fully Supervised	81.53	75.38	63.16	81.30	81.88

The results indicate that the learned representations have found useful patterns from the input data. The pre-trained weights perform better than both a supervised classification layer on top of randomly initialized weights, and a fully supervised version of the model with randomly initialized weights, which are good baselines to assess the effectiveness of the representations. They are effective to be used in two common scenarios:

- To be used as a frozen pre-trained network which produces embeddings from the input data, followed by a custom classification head that can be trained to learn to classify the produced embeddings in a specific down-stream task.
- To be used as initialization weights, that can be fully fine-tuned in order to perform a specific down-stream task.

6 Chapter VI - Conclusion and Future Work

6.1 Conclusion

In this study, we presented a new approach based on multi-view contrastive learning that uses Transformers in its architecture in order to obtain useful representations from input EEG data to be used in various downstream tasks that involves finding patterns and encoding longer period EEG data, such as predicting dementia or general abnormality.

In chapter 1, we discussed the importance of the problem, challenges, and the overview of the problems that we addressed. In chapter 2, we address some of the theoretical concepts, algorithms, and methodologies that were used throughout this research. In chapter 3, we review relevant studies and important research in this area of study. In chapter 4, we present the datasets that were used, data processing pipeline, and training pipeline that includes data augmentation. We also discuss the model architecture in detail and explain each part of it in isolation, including the loss functions. In the last chapter, we presented the results of the pre-training, and as well as the method that we used to evaluate the quality of the pre-trained model and the respective results.

We found that in addition to common approaches with Transformers such as Masked Token Modeling where the task is to predict the masked token, a Contrastive Task, which operates according to similarities and dissimilarities between different versions of the same input, can yield promising results and be effective at encoding EEG data, given the appropriate structure. It is however important to note that the input data needs to be properly tokenized (we used patch embeddings) in order to be used within a Transformer. Additionally, introducing multiple views of the same data, such as time-domain view and frequency-domain view, as well as their combination, can contribute to the learning process. This can open the door to more research and investigation in this area.

6.2 Future Work and Suggestions

It is important to note that due to the limitation of computing resources, it was not possible to perform a complete analysis of hyperparameters or an ablation study to identify the factors contributing to the performance of the model. Since the experiments were done only within a limited set of configurations, it is likely that with a more thorough sensitivity analysis and hyperparameter tuning, even a significantly better result might be achieved. There are many parameters that can contribute to the quality of the final representations, including but not limited to:

- Number of the Transformer layers
- Number of the attention heads
- Embedding size of the initial tokens
- Number of feature maps in the patch embedding module
- Size of the pooling layer in the patch embedding module
- A fully connected layer instead of global average pooling in the final embedding layer
- Positional embeddings for patches in addition to time embeddings

- Loss function and optimiser's hyperparameters

Since data augmentation is a vital part of the pipeline, designing and implementing the right type of augmentation can have a non-negligible impact on the pre-training results. We discarded the time-related augmentation (frame flipping) as it adversely affected the learning, but a different type of augmentation such as spatiotemporal patching or temporal cropping might be better at preserving the causal relationship between the frames.

In addition to model configuration and hyperparameters, the addition of a second dataset (NMT) contributed significantly to the learning of the model. We believe that increasing both the number of datasets and their diversity can lead to a more general model and more powerful representations.

Finally, our approach in this study is trying to find a general representation of EEG data, so naturally future experiments can include applying the resulting pre-trained model on different domains via transfer learning and fine-tuning to confirm the usefulness of the representations in the respective domains.

A Appendix

A.1 Plan for future publication

In the future publications, our research will focus on the following aspects to enhance the understanding and applicability of the proposed self-supervised multi-view approach for learning and extracting representations from long-period EEG data:

1. Ablation study and comparative analysis: Carry out a comprehensive ablation study to identify the key factors contributing to the performance of the model, and to understand their impact on the quality of the final representations. This will involve examining various hyperparameters, such as the number of Transformer layers, attention heads, embedding size, feature maps in the patch embedding module, and the size of the pooling layer in the patch embedding module. This will aid in refining the model and improving its performance.
2. Enhanced data augmentation: Investigate different data augmentation techniques, such as spatiotemporal patching or temporal cropping, which can potentially improve the learning of meaningful representations while preserving the causal relationships between frames.
3. Additional datasets: To increase the generalizability of the model, the research will incorporate a greater number of datasets with diverse characteristics. This would help in developing more powerful representations and promote adaptability across different domains.
4. Transfer learning and fine-tuning: Future experiments will involve applying the pretrained model to various domains using transfer learning and fine-tuning approaches. This will provide insights into the effectiveness and applicability of the learned representations in different contexts, thereby confirming their usefulness in multiple domains.

Overall, the future publication will build upon the findings of the current thesis by exploring the factors contributing to the model's performance, refining the proposed approach, and demonstrating its applicability across various domains. This will ultimately contribute to the advancement of deep learning techniques in the analysis of EEG data, with potential applications in early detection and diagnosis of neurological disorders.

Bibliography

[1] W. H. Organization. 'Dementia'. (2023), [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/dementia.

[2] Z. Arvanitakis, R. C. Shah and D. A. Bennett, 'Diagnosis and management of dementia: Review', en, *JAMA*, vol. 322, no. 16, pp. 1589–1599, Oct. 2019.

[3] Y. D. Reijmer, E. van den Berg, S. van Sonsbeek *et al.*, 'Dementia risk score predicts cognitive impairment after a period of 15 years in a nondemented population', *Dementia and Geriatric Cognitive Disorders*, vol. 31, no. 2, pp. 152–157, 2011. DOI: 10.1159/000324437. [Online]. Available: https://doi.org/10.1159/000324437.

[4] B. N. Harding, J. S. Floyd, J. F. Scherrer *et al.*, 'Methods to identify dementia in the electronic health record: Comparing cognitive test scores with dementia algorithms', *Healthcare*, vol. 8, no. 2, p. 100 430, 2020, ISSN: 2213-0764. DOI: https://doi.org/10.1016/j.hjdsi.2020.100430. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2213076420300294.

[5] E. Vuoksimaa, J. O. Rinne, N. Lindgren, K. Heikkilä, M. Koskenvuo and J. Kaprio, 'Middle age self-report risk score predicts cognitive functioning and dementia in 20–40 years', *Alzheimer's &amp Dementia: Diagnosis, Assessment &amp Disease Monitoring*, vol. 4, no. 1, pp. 118–125, Jan. 2016. DOI: 10.1016/j.dadm.2016.08.003. [Online]. Available: https://doi.org/10.1016/j.dadm.2016.08.003.

[6] R. Nardone, L. Sebastianelli, V. Versace *et al.*, 'Usefulness of EEG techniques in distinguishing frontotemporal dementia from alzheimer's disease and other dementias', *Disease Markers*, vol. 2018, pp. 1–9, Sep. 2018. DOI: 10.1155/2018/6581490. [Online]. Available: https://doi.org/10.1155/2018/6581490.

[7] J. Snaedal, G. H. Johannesson, T. E. Gudmundsson *et al.*, 'Diagnostic accuracy of statistical pattern recognition of electroencephalogram registration in evaluation of cognitive impairment and dementia', *Dementia and Geriatric Cognitive Disorders*, vol. 34, no. 1, pp. 51–60, 2012. DOI: 10.1159/000339996. [Online]. Available: https://doi.org/10.1159/000339996.

[8] T. Zorick, J. Landers, A. Leuchter and M. A. Mandelkern, 'EEG multifractal analysis correlates with cognitive testing scores and clinical staging in mild cognitive impairment', *Journal of Clinical Neuroscience*, vol. 76, pp. 195–200, Jun. 2020. DOI: 10.1016/j.jocn.2020.04.003. [Online]. Available: https://doi.org/10.1016/j.jocn.2020.04.003.

[9] J. Choi, B. Ku, Y. G. You *et al.*, 'Resting-state prefrontal EEG biomarkers in correlation with MMSE scores in elderly individuals', *Scientific Reports*, vol. 9, no. 1, Jul. 2019. DOI: 10.1038/s41598-019-46789-2. [Online]. Available: https://doi.org/10.1038/s41598-019-46789-2.

[10] X.-W. Wang, D. Nie and B.-L. Lu, 'Emotional state classification from EEG data using machine learning approach', *Neurocomputing*, vol. 129, pp. 94–106, Apr. 2014. DOI: 10.1016/j.neucom.2013.06.046. [Online]. Available: https://doi.org/10.1016/j.neucom.2013.06.046.

[11] A. Shoeb and J. Guttag, 'Application of machine learning to epileptic seizure detection', in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ser. ICML'10, Haifa, Israel: Omnipress, 2010, pp. 975–982, ISBN: 9781605589077.

[12] G. Li, C. H. Lee, J. J. Jung, Y. C. Youn and D. Camacho, 'Deep learning for EEG data analytics: A survey', *Concurrency and Computation: Practice and Experience*, vol. 32, no. 18, Feb. 2019. DOI: 10.1002/cpe.5199. [Online]. Available: https://doi.org/10.1002/cpe.5199.

[13] A. R. Ozcan and S. Erturk, 'Seizure prediction in scalp EEG using 3d convolutional neural networks with an image-based approach', *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 11, pp. 2284–2293, Nov. 2019. DOI: 10.1109/tnsre.2019.2943707. [Online]. Available: https://doi.org/10.1109/tnsre.2019.2943707.

[14] S. Thundiyil, M. Thungamani and S. Hariprasad, 'Big EEG data images for convolutional neural networks', in *2021 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, IEEE, Dec. 2021. DOI: 10.1109/spmb52430.2021.9672272. [Online]. Available: https://doi.org/10.1109/spmb52430.2021.9672272.

[15] G. Zhang and A. Etemad, 'Deep recurrent semi-supervised eeg representation learning for emotion recognition', Sep. 2021, pp. 1–8. DOI: 10.1109/ACII52823.2021.9597449.

[16] C.-L. Liu, B. Xiao, W.-H. Hsaio and V. S. Tseng, 'Epileptic seizure prediction with multi-view convolutional neural networks', *IEEE Access*, vol. 7, pp. 170 352–170 361, 2019. DOI: 10.1109/access.2019.2955285. [Online]. Available: https://doi.org/10.1109/access.2019.2955285.

[17] A. Vaswani, N. Shazeer, N. Parmar *et al.*, 'Attention is all you need', in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[18] F. Schroff, D. Kalenichenko and J. Philbin, 'Facenet: A unified embedding for face recognition and clustering', in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 815–823. DOI: 10.1109/CVPR.2015.7298682.

[19] K. Sohn, 'Improved deep metric learning with multi-class n-pair loss objective', in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon and R. Garnett, Eds., vol. 29, Curran Associates, Inc., 2016. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2016/file/6b180037abbebea991d8b1232f8a8ca9-Paper.pdf.

[20] A. van den Oord, Y. Li and O. Vinyals, *Representation learning with contrastive predictive coding*, 2019. arXiv: 1807.03748 [cs.LG].

[21] J.-B. Grill, F. Strub, F. Altché *et al.*, 'Bootstrap your own latent a new approach to self-supervised learning', in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS'20, Vancouver, BC, Canada: Curran Associates Inc., 2020, ISBN: 9781713829546.

[22] Z.-R. Sun, Y.-X. Cai, S.-J. Wang *et al.*, 'Multi-view intact space learning for tinnitus classification in resting state EEG', *Neural Processing Letters*, vol. 49, no. 2, pp. 611–624, May 2018. DOI: 10.1007/s11063-018-9845-1. [Online]. Available: https://doi.org/10.1007/s11063-018-9845-1.

[23] L. Tang, N. Xie, M. Zhao and X. Wu, 'Seizure prediction using multi-view features and improved convolutional gated recurrent network', *IEEE Access*, vol. 8, pp. 172 352–172 361, 2020. DOI: 10.1109/access.2020.3024580. [Online]. Available: https://doi.org/10.1109/access.2020.3024580.

[24] C. Chen, Z. Li, F. Wan, L. Xu, A. Bezerianos and H. Wang, 'Fusing frequency-domain features and brain connectivity features for cross-subject emotion recognition', *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–15, 2022. DOI: 10.1109/tim.2022.3168927. [Online]. Available: https://doi.org/10.1109/tim.2022.3168927.

[25] L.-M. Sanchez-Reyes, J. Rodriguez-Resendiz, G. N. Avecilla-Ramirez, M.-L. Garcia-Gomar and J.-B. Robles-Ocampo, 'Impact of EEG parameters detecting dementia diseases: A systematic review', *IEEE Access*, vol. 9, pp. 78 060–78 074, 2021. DOI: 10.1109/access.2021.3083519. [Online]. Available: https://doi.org/10.1109/access.2021.3083519.

[26] S. Liang, W. Hang, M. Yin *et al.*, 'Deep EEG feature learning via stacking common spatial pattern and support matrix machine', *Biomedical Signal Processing and Control*, vol. 74, p. 103 531, Apr. 2022. DOI: 10.1016/j.bspc.2022.103531. [Online]. Available: https://doi.org/10.1016/j.bspc.2022.103531.

[27] D. Hu, J. Cao, X. Lai, Y. Wang, S. Wang and Y. Ding, 'Epileptic state classification by fusing hand-crafted and deep learning EEG features', *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 4, pp. 1542–1546, Apr. 2021. DOI: 10.1109/tcsii.2020.3031399. [Online]. Available: https://doi.org/10.1109/tcsii.2020.3031399.

[28] H. Banville, O. Chehab, A. Hyvärinen, D.-A. Engemann and A. Gramfort, 'Uncovering the structure of clinical EEG signals with self-supervised learning', *Journal of Neural Engineering*, vol. 18, no. 4, p. 046 020, Mar. 2021. DOI: 10.1088/1741-2552/abca18. [Online]. Available: https://doi.org/10.1088/1741-2552/abca18.

[29] V. Kumar, L. Reddy, S. K. Sharma *et al.*, *Muleeg: A multi-view representation learning on eeg signals*, 2022. arXiv: 2204.03272 [cs.LG].

[30] G. Bhatt, P. Jha and B. Raman, 'Representation learning using step-based deep multi-modal autoencoders', *Pattern Recognition*, vol. 95, pp. 12–23, Nov. 2019. DOI: 10.1016/j.patcog.2019.05.032. [Online]. Available: https://doi.org/10.1016/j.patcog.2019.05.032.

[31] H. Li, Z. Deng, H. Yang *et al.*, 'circRNA-binding protein site prediction based on multi-view deep learning, subspace learning and multi-view classifier', *Briefings in Bioinformatics*, vol. 23, no. 1, Sep. 2021. DOI: 10.1093/bib/bbab394. [Online]. Available: https://doi.org/10.1093/bib/bbab394.

[32] Y. Jiang, Y. Zhang, C. Lin, D. Wu and C.-T. Lin, 'EEG-based driver drowsiness estimation using an online multi-view and transfer TSK fuzzy system', *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1752–1764, Mar. 2021. DOI: 10.1109/tits.2020.2973673. [Online]. Available: https://doi.org/10.1109/tits.2020.2973673.

[33] J. Ye, Q. Xiao, J. Wang, H. Zhang, J. Deng and Y. Lin, 'ICoSleep/i: A multi-view representation learning framework for self-supervised learning of sleep stage classification', *IEEE Signal Processing Letters*, vol. 29, pp. 189–193, 2022. DOI: 10.1109/lsp.2021.3130826. [Online]. Available: https://doi.org/10.1109/lsp.2021.3130826.

[34] J. Xu, H. Zheng, J. Wang, D. Li and X. Fang, 'Recognition of EEG signal motor imagery intention based on deep multi-view feature learning', *Sensors*, vol. 20, no. 12, p. 3496, Jun. 2020. DOI: 10.3390/s20123496. [Online]. Available: https://doi.org/10.3390/s20123496.

[35] D. Kostas, S. Aroca-Ouellette and F. Rudzicz, 'BENDR: Using transformers and a contrastive self-supervised learning task to learn from massive amounts of EEG data', *Frontiers in Human Neuroscience*, vol. 15, Jun. 2021. DOI: 10.3389/fnhum.2021.653659. [Online]. Available: https://doi.org/10.3389/fnhum.2021.653659.

[36] A. Baevski, Y. Zhou, A. Mohamed and M. Auli, 'Wav2vec 2.0: A framework for self-supervised learning of speech representations', in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan and H. Lin, Eds., vol. 33, Curran Associates, Inc., 2020, pp. 12 449–12 460. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/92d1e1eb1cd6f9fba3227870bb6d7f07-Paper.pdf.

[37] Y. Wei, Y. Liu, C. Li, J. Cheng, R. Song and X. Chen, 'Tc-net: A transformer capsule network for eeg-based emotion recognition', *Computers in Biology and Medicine*, vol. 152, p. 106 463, 2023, ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2022.106463. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010482522011714.

[38] Y. Song, X. Jia, L. Yang and L. Xie, *Transformer-based spatial-temporal feature learning for eeg decoding*, 2021. arXiv: 2106.11170 [eess.SP].

[39] Y. Song, Q. Zheng, B. Liu and X. Gao, 'Eeg conformer: Convolutional transformer for eeg decoding and visualization', *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 31, pp. 710–719, 2023. DOI: 10.1109/TNSRE.2022.3230250.

[40] I. Obeid and J. Picone, 'The temple university hospital EEG data corpus', *Frontiers in Neuroscience*, vol. 10, May 2016. DOI: 10.3389/fnins.2016.00196. [Online]. Available: https://doi.org/10.3389/fnins.2016.00196.

[41] H. A. Khan, R. U. Ain, A. M. Kamboh *et al.*, 'The NMT scalp EEG dataset: An open-source annotated dataset of healthy and pathological EEG recordings for predictive modeling', *Frontiers in Neuroscience*, vol. 15, Jan. 2022. DOI: 10.3389/fnins.2021.755817. [Online]. Available: https://doi.org/10.3389/fnins.2021.755817.

[42] L. A. Gemein, R. T. Schirrmeister, P. Chrabaszcz *et al.*, 'Machine-learning-based diagnostics of eeg pathology', *NeuroImage*, vol. 220, p. 117 021, 2020, ISSN: 1053-8119. DOI: https://doi.org/10.1016/j.neuroimage.2020.117021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1053811920305073.

[43] J. Kayser and C. E. Tenke, 'On the benefits of using surface laplacian (current source density) methodology in electrophysiology', *International Journal of Psychophysiology*, vol. 97, no. 3, pp. 171–173, Sep. 2015. DOI: 10.1016/j.ijpsycho.2015.06.001. [Online]. Available: https://doi.org/10.1016/j.ijpsycho.2015.06.001.

[44] M. N. Mohsenvand, M. R. Izadi and P. Maes, 'Contrastive representation learning for electroencephalogram classification', in *Proceedings of the Machine Learning for Health NeurIPS Workshop*, E. Alsentzer, M. B. A. McDermott, F. Falck, S. K. Sarkar, S. Roy and S. L. Hyland, Eds., ser. Proceedings of Machine Learning Research, vol. 136, PMLR, Nov. 2020, pp. 238–253. [Online]. Available: https://proceedings.mlr.press/v136/mohsenvand20a.html.

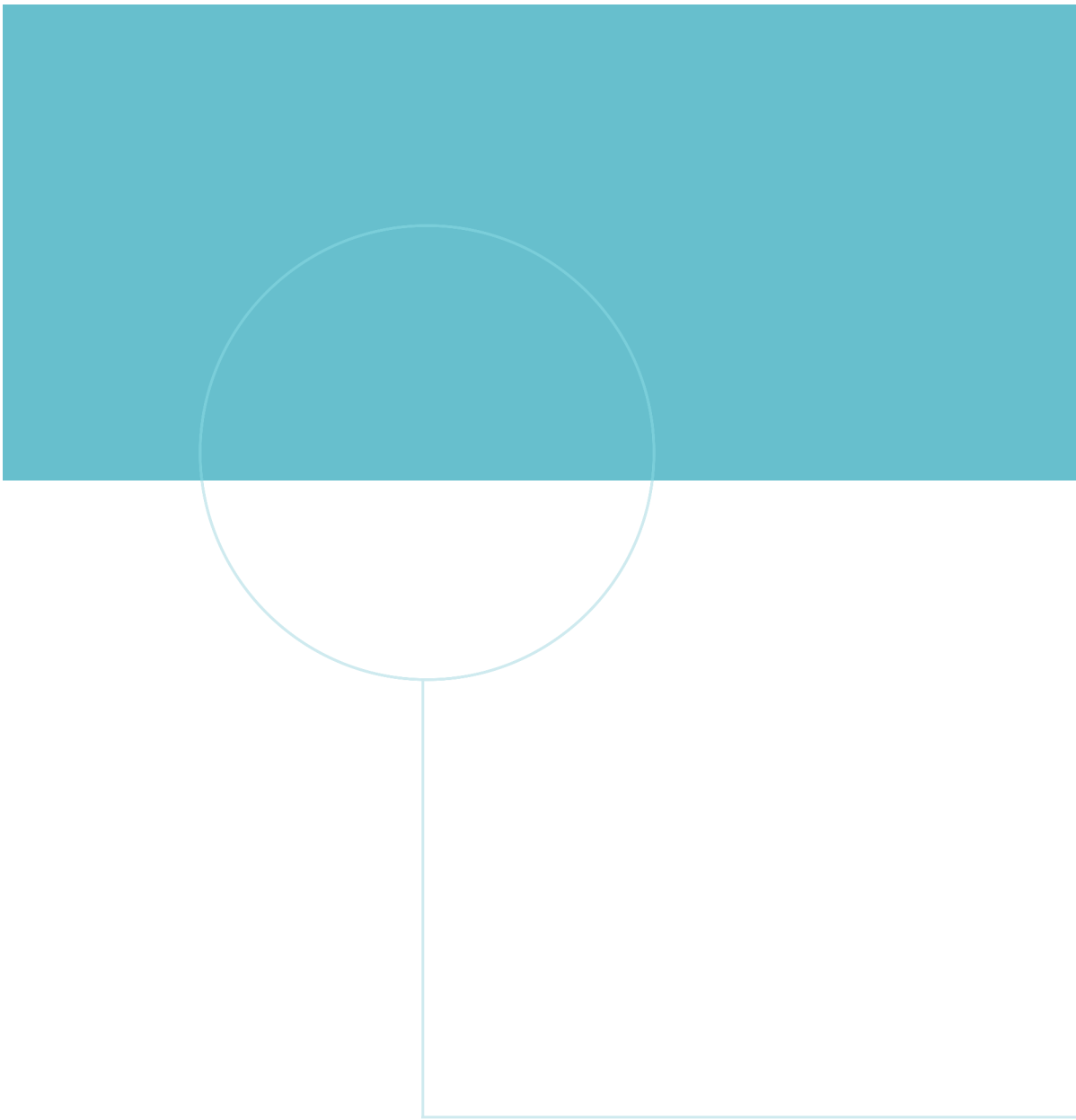
[45] J. Selva, A. S. Johansen, S. Escalera, K. Nasrollahi, T. B. Moeslund and A. Clapes, 'Video transformers: A survey', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–20, 2023. DOI: 10.1109/tpami.2023.3243465. [Online]. Available: https://doi.org/10.1109/tpami.2023.3243465.

[46] G. Bertasius, H. Wang and L. Torresani, 'Is space-time attention all you need for video understanding?', in *Proceedings of the 38th International Conference on Machine Learning*, M. Meila and T. Zhang, Eds., ser. Proceedings of Machine Learning Research, vol. 139, PMLR, 18–24 Jul 2021, pp. 813–824. [Online]. Available: https://proceedings.mlr.press/v139/bertasius21a.html.

[47] N. Park, W. Kim, B. Heo, T. Kim and S. Yun, 'What do self-supervised vision transformers learn?', in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=azCKuYyS74.

[48] T. Chen, S. Kornblith, M. Norouzi and G. Hinton, 'A simple framework for contrastive learning of visual representations', in *Proceedings of the 37th International Conference on Machine Learning*, ser. ICML'20, JMLR.org, 2020.

[49] R. Zhang, P. Isola and A. A. Efros, *Colorful image colorization*, 2016. arXiv: 1603.08511 [cs.CV].



NTNU

Norwegian University of
Science and Technology