JOHANNES KEPLER
UNIVERSITY LINZ

Author
Noel Kamm
12211207

Submission
**Institute for
Machine Learning**

First Supervisor
Dr. **Philipp Seidl**, MSc

Second Supervisor
Prof. Dr. **Thomas Knösche**

Assistant Thesis Supervisor
Sasha Brenner, MSc

November 2025

Learning Latent Representations of EEG Signals Using Autoencoders

Bachelor's Thesis

to confer the academic degree of

Bachelor of Science

in the Bachelor's Program

Artificial Intelligence

**JOHANNES KEPLER
UNIVERSITY LINZ**
Altenberger Straße 69
4040 Linz, Austria
jku.at

Abstract

This thesis explores the use of autoencoder-based neural networks for analyzing electroencephalography (EEG) signals, focusing on their ability to learn representations that may help distinguish seizure from non-seizure activity in an unsupervised manner. Automatic seizure detection is challenging due to the high dimensionality, noise and non-stationarity of EEG data. To address these issues, two autoencoder frameworks — a deterministic autoencoder and a variational autoencoder — were trained to reconstruct single-channel EEG recordings from the TUH EEG Seizure Corpus and extract their low-dimensional latent representations. The performance of the models was compared to that of a linear, unsupervised dimensionality reduction technique. Both autoencoders achieved high reconstruction accuracy and produced latent spaces that captured seizure-related structure more effectively than the linear method. Logistic regression trained on latent representations derived from autoencoders achieved significantly better discrimination between seizure and non-seizure activity than logistic regression trained on encodings obtained from the linear baseline. However, no statistically significant performance difference was observed between the variational and deterministic autoencoders. The results demonstrate that autoencoders can both compress EEG data efficiently and reveal meaningful patterns related to seizure activity, providing a robust foundation for future work regarding automated detection of seizures.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor, Sasha Brenner from the Max Planck Institute for Human Cognitive and Brain Sciences Leipzig. This thesis would not have been possible without his enormous investment of time, expertise, and patience. His guidance and scientific insight have continuously inspired me, and I have learned a great deal from him about machine learning, neuroscience, and the process of scientific thinking. Over the course of this work, he has also become a dear friend.

I would also like to thank Prof. Dr. Thomas Knösche and Dr. Nico Scherf from the Max Planck Institute for Human Cognitive and Brain Sciences Leipzig for their supervision, for welcoming me into the institute, and for their valuable feedback throughout the project. My sincere thanks extend to the entire institute for granting me access to its computing facilities, which made it possible to train my models efficiently.

Furthermore, I would like to thank Dr. Philipp Seidl from Johannes Kepler University Linz for kindly agreeing to serve as the examiner of this thesis.

Finally, I would like to thank my parents. They supported me as best as they could throughout this challenging time, which allowed me to focus on my thesis.

Notation

In this thesis, the notation has been chosen to remain clear and consistent in order to support readability and to distinguish between different mathematical objects. Matrices and vectors are written in boldface, with matrices denoted by uppercase letters and vectors by lowercase letters. Superscripts are used for enumeration, for example to index elements of a set or to refer to column and row vectors of a matrix. Subscripts serve either to specify a variable more precisely or to denote individual components of a vector.

Contents

Chapter 1

Introduction

Electroencephalography (EEG) is a medical technique that uses electrodes placed on the scalp to record the brains electrical activity [29]. Neurologists routinely use EEG to detect epileptic seizures, but the sheer volume of data makes manual assessment time-consuming and cumbersome. Automatic seizure detection, however, is challenging and computationally costly due to the high dimensionality, non-stationary nature, and susceptibility to noise of EEG data [25].

To mitigate these issues, various approaches have been explored, ranging from manual feature engineering to established techniques such as Principal Component Analysis (PCA) and Independent Component Analysis [8][5]. More recently, neural-network-based reconstruction models, known as autoencoders, have shown promise in compressing high-dimensional and noisy data into compact representations while retaining the features of interest [4].

This motivates their application to EEG signals, particularly in assessing seizure activity. The aim of this thesis is to investigate whether autoencoders can be effectively employed to compress and reconstruct EEG recordings, and whether the resulting latent representations can support the distinction between seizure and non-seizure activity. To this end, we trained and compared two types of autoencoder frameworks — a Variational Autoencoder (VAE) and a standard, Deterministic Autoencoder (DAE) — to reconstruct EEG signals and analyzed the structure of the spaces occupied by latent representations. Our primary interest was to determine whether seizure and non-seizure encodings exhibit systematic differences in these spaces. To address this question, we employed several complementary analyses, including statistical characterization of the latent representations, dimensionality reduction for visualization and exploration of potential separability, and the training of lightweight classifiers to assess the discriminative capacity of the latent features. As a common baseline, PCA was used to evaluate whether the autoencoder-based representations capture seizure-related structure more effectively.

All code used for this project, including the data preprocessing pipeline, model training, and evaluation was implemented in Python using PyTorch. The full implementation is publicly available at https://github.com/Kammnoel1/VAEEG.

The remainder of this thesis is organized as follows:

- Chapter 2: Background and Related Work — gives an overview of characteristics of seizure data, neural networks and autoencoders, as well as a brief introduction to principal component analysis.
- Chapter 3: Material and Methods — describes the dataset, preprocessing pipeline, model architecture, and the training and evaluation procedures used.
- Chapter 4: Results and Discussion, focusing on descriptive statistics of the latent space as well as on the visualization of that space and the outcomes of the classifiers.
- Chapter 5: Conclusion and Future Work, summarizes the main findings and contributions, and outlines potential directions for future research.

Chapter 2

Background and Related Work

2.1 Seizure Detection in Electroencephalography

The diagnosis of many neurological conditions such as epilepsy crucially depends on EEG, which involves recording the brains electrical activity through electrodes placed on the scalp [30]. Even specialized neurologists often take a long time to analyze and interpret EEG data, and the large amount of data that needs to be manually reviewed overwhelms them and makes real-time seizure detection infeasible. This highlights the need for a partially automated technique to analyze EEG data.

Unfortunately, automated seizure detection in EEG remains a difficult task due to multiple sources of variability and noise. One major challenge is the low signal-to-noise ratio (SNR), meaning that the relevant brain signals are often obscured by unwanted background activity. This reduction in SNR is caused by artifacts commonly present in EEG recordings, such as muscle movements, eye blinks, and imprecise electrode placement, which can overshadow the brain activity that is critical for seizure analysis. Another challenge is the non-stationary nature of EEG signals. Their characteristics can change over time within the same subject, as well as across different subjects [11]. In addition, datasets are often heterogeneous due to the use of different montages during recording [24]. The choice of a montage may vary depending on the use case, the research objective, and the national standards. A montage refers to the specific arrangement of channels across the scalp, as illustrated in Figure 2.1.Each channel measures the voltage difference between two electrodes or between an electrode and a reference point.

Indeed, however promising some results on individual, homogeneous research datasets may be, their poor generalizability and the critical nature of seizure diagnosis make the assessment of trained human neurologists indispensable in a clinical setting. Experts identify seizures by detecting synchronized abnormal brain activity, which typically appears in the EEG as sudden spikes or sharp wave patterns within a characteristic frequency range of approximately 0.01–100 Hz [35], as exemplified in Figure 2.2. Furthermore, seizures can vary in onset and duration, typically lasting from a few seconds to several minutes. Determining their precise onset and offset is difficult, even for trained neurologists, due to the gradual transition between background and seizure activity.

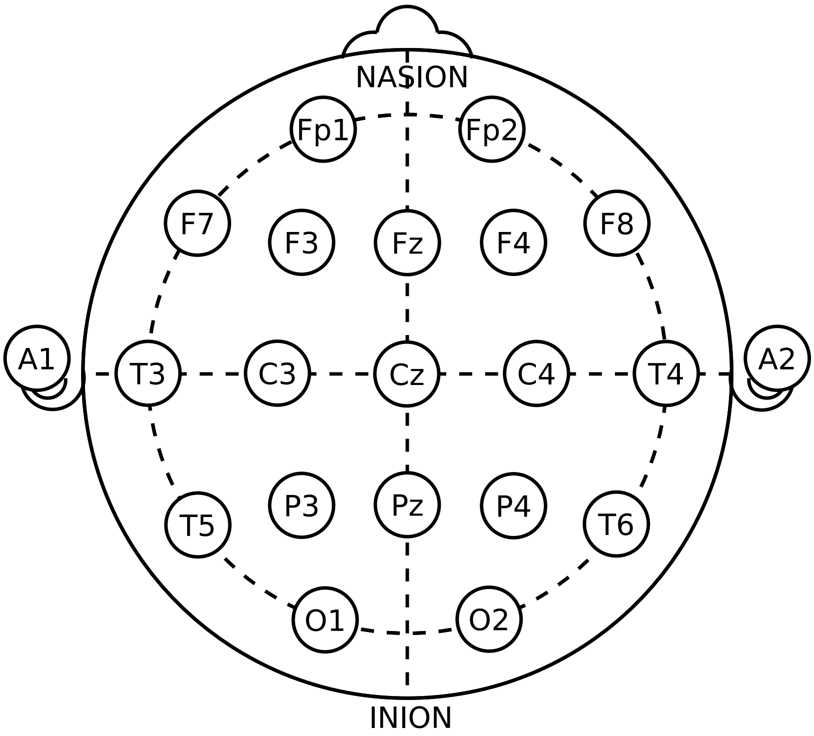


Figure 2.1: Electrode placement according to the International 10–20 system for EEG recording. The positions correspond to standard scalp locations used for measuring electrical brain activity [2].

Seizure variability manifests in both the temporal and spatial domains. In fact, seizures are often separated into types according to the location and pattern of the abnormal brain activity. The International League Against Epilepsy recognizes four main categories of seizure, which are divided into 21 specific types [6]. For the purposes of this thesis, however, we focus on the binary distinction between seizure and background activity.

Finally, the high dimensionality and variability of EEG data present challenges for manual interpretation and machine learning algorithms alike. Dimensionality reduction methods are often employed to address this, capturing the most informative aspects of the data while discarding redundancy and noise.

2.2 Principal Component Analysis

Principal Component Analysis (PCA) is one of the earliest and most widely used linear techniques for reducing high-dimensional data, first introduced by Pearson [32] and later independently developed and named by Hotelling [18]. Its primary objective is to linearly project the data onto a new coordinate system defined by principal components (PCs), which capture the largest variations present in the data [10]. Importantly, PCA is an unsupervised technique and does not rely on labeled information.

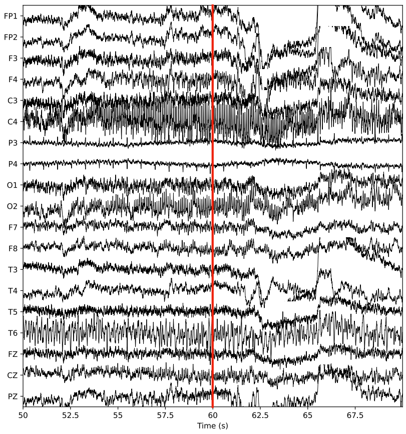


Figure 2.2: EEG recording illustrating seizure activity. The red bar indicates the onset of a seizure. EEG signals preceding the red bar correspond to non-seizure activity, while those following it represent seizure activity.

Let the data matrix be defined as $\mathbf{X} \in \mathbb{R}^{n \times d}$, where n is the number of observations and d is the dimensionality of the data. We denote the data matrix as:

$$\mathbf{X} = \begin{bmatrix} (\mathbf{x}^{(1)})^\top \\ (\mathbf{x}^{(2)})^\top \\ \vdots \\ (\mathbf{x}^{(n)})^\top \end{bmatrix}, \quad \mathbf{x}^{(i)} \in \mathbb{R}^d.$$

where each row vector $(\mathbf{x}^{(i)})^T$ represents one observation and $i = 1, 2, \cdots, n$. In our use case, the dimensions represent the different channels of the recording and each observation corresponds to the signal values of the channels recorded at a specific time point. From the data matrix, we can compute the covariance matrix, where each entry quantifies the covariance between two dimensions, i.e., how channel signals co-vary over time. The covariance matrix captures relationships between brain regions that are relevant for identifying local patterns in seizures. To eliminate biases caused by differences in scale between channels, the covariance matrix is constructed from the mean-centered data matrix.

Formally, the empirical mean vector is obtained by averaging each feature dimension across all n observations:

$$\hat{\boldsymbol{\mu}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}^{(i)} \in \mathbb{R}^d.$$

The corresponding empirical mean matrix is $\hat{\mathbf{M}} = \mathbf{1}\hat{\boldsymbol{\mu}}^\top,\ \mathbf{1} \in \mathbb{R}^n$, so that the mean-centered data matrix is given by $\tilde{\mathbf{X}} = \mathbf{X} - \hat{\mathbf{M}}$. The covariance matrix of the centered data matrix is $\mathbf{C} = \tilde{\mathbf{X}}^\top \tilde{\mathbf{X}} \in \mathbb{R}^{d \times d}$.

PCA proceeds by finding the d linearly-independent, mutually uncorrelated patterns that capture the largest possible variance in the data. This corresponds to solving the eigenvalue problem of the covariance matrix, given as:

$$\mathbf{C}\mathbf{V} = \boldsymbol{\Lambda}\mathbf{V}, \quad \mathbf{V}, \boldsymbol{\Lambda} \in \mathbb{R}^{d\times d}.$$

The columns of matrix $\mathbf{V}$ are the eigenvectors, also called loadings, of $\mathbf{C}$ and Λ is the diagonal matrix of corresponding eigenvalues. The loadings can be written as:

$$\mathbf{V} = \begin{pmatrix} \mathbf{v}^{(1)} & \mathbf{v}^{(2)} & \cdots & \mathbf{v}^{(d)} \end{pmatrix}.$$

The loadings form an orthonormal set. The normalization, i.e. unit length of each loading is imposed by definition, while orthogonality follows as a mathematical consequence of the covariance matrix being symmetric, as follows from the spectral theorem. The corresponding eigenvalues are:

$$\boldsymbol{\lambda} = \operatorname{diag}(\boldsymbol{\Lambda}), \quad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d.$$

By column concatenation of the first k loadings (with $k < d$), we obtain the projection matrix:

$$\mathbf{P} = \begin{pmatrix} \mathbf{v}^{(1)} & \mathbf{v}^{(2)} & \cdots & \mathbf{v}^{(k)} \end{pmatrix} \in \mathbb{R}^{d\times k}.$$

The dimensionality-reduced representation of the data is then:

$$\mathbf{Z} = \begin{pmatrix} \mathbf{z}^{(1)} & \mathbf{z}^{(2)} & \cdots & \mathbf{z}^{(k)} \end{pmatrix} = \tilde{\mathbf{X}}\mathbf{P} \in \mathbb{R}^{n\times k}.$$

Where each column $\mathbf{z}^{(j)} \in \mathbb{R}^n$ represents the j-th PC, i.e. the projection of all samples onto the loading vector $\mathbf{v}^{(j)}$ for $j = 1, 2, \cdots, k$. Since the principal components are constructed as weighted combinations of the original features, PCA is considered a linear dimensionality reduction method.

Additionally, each eigenvalue corresponds to the variance explained by its associated PC. The explained variance ratio of the k-th component is defined as:

$$r_k = \frac{\lambda_k}{\sum_{j=1}^{d} \lambda_j}, \quad 0 \leq r_k \leq 1,$$

and the cumulative explained variance of all d PCs is:

$$R_d = \sum_{i=1}^{d} r_i = 1.$$

PCA transforms the dataset into a lower-dimensional representation while preserving a chosen proportion of the total variance. In practice, the first few PCs usually capture the dominant structure of the data, whereas later PCs tend to represent minor variations. As a result, selecting the number of PCs involves balancing interpretability against the loss of information. Thus, PCA provides a simple and mathematically elegant framework for dimensionality reduction, making it an obvious starting point for evaluating more advanced techniques.

However, EEG signals are highly complex and often exhibit nonlinear dynamics. This motivates the exploration of nonlinear dimensionality reduction methods.

In particular, neural-network-based autoencoders provide a powerful alternative, as they can capture complex, nonlinear relationships in the data that are beyond the reach of variance-based linear projections such as PCA. Before introducing autoencoders, however, it is essential to first outline the basic principles of neural networks themselves.

2.3 Neural Networks

Over the past twenty years, artificial neural networks have advanced within the field of deep learning, achieving remarkable success in areas such as image classification, natural language generation and audio processing — tasks that were considered out of reach for traditional machine learning methods [27] [41] [28]. These achievements have led to the expectation that neural networks may also drive progress in the analysis of EEG data. Indeed, network-based approaches have surpassed the benchmarks in seizure detection established by earlier techniques [12] [38].

The aim of the following section is to introduce the basic concepts and notation of feedforward neural networks, which serve as the foundation for the more advanced architectures used in this thesis. Unless otherwise stated, the presentation and terminology are based on Prince [33]. While many of the principles discussed below — such as Section 2.3.2 Activation Function, Section 2.3.5 Batch Normalization, and Section 2.3.3 Loss Function and Gradient Descent — are generally applicable to other types of neural networks as well, the focus of this thesis is restricted to feedforward networks.

2.3.1 Feedforward neural networks

Feedforward neural networks (FNN) are a class of machine learning models inspired by the structure of the human brain [7]. They are among the most basic and straightforward types of neural networks. An FNN is built from neurons, which are simple computational units that take weighted inputs, add a bias term, and apply a nonlinear transformation. The neurons are arranged in layers, and in a fully-connected design every neuron in one layer is connected to every neuron in the next layer. Each layer processes the outputs of the previous one, and by stacking many such layers an FNN can approximate highly complex functions.

Formally, an FNN can be regarded as a parameterized function:

$$f_\theta : \mathbb{R}^{d_{\text{in}}} \to \mathbb{R}^{d_{\text{out}}},$$

where $d_{\text{in}}, d_{\text{out}} \in \mathbb{N}$ denote the input and output dimensions of the FNN, respectively, and the parameters $\theta = \left(\mathbf{W}^{(1)}, \mathbf{b}^{(1)}, \mathbf{W}^{(2)}, \mathbf{b}^{(2)}, \dots, \mathbf{W}^{(L)}, \mathbf{b}^{(L)}\right)$ consist of the trainable weights and biases. For each fully-connected layer $l = 1, \dots, L$, the weight matrix is denoted by $\mathbf{W}^{(l)} \in \mathbb{R}^{d_l \times d_{l-1}}$, and the corresponding bias vector by $\mathbf{b}^{(l)} \in \mathbb{R}^{d_l}$, where d_{l-1} is the number of neurons in the previous layer and d_l the number of neurons in the current layer.

2.3.2 Activation Function

We define layer functions as mappings that transform the output of the previous layer into the current one. Formally, for each layer $l = 1, \dots, L$, a layer function is given by:

$$f^{(l)} : \mathbb{R}^{d_{l-1}} \to \mathbb{R}^{d_l}, \quad f^{(l)}(\mathbf{x}) = \mathbf{h}^{(l)} = \sigma\Big(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}\Big),$$

where $\sigma : \mathbb{R} \to \mathbb{R}$ is a nonlinear function applied elementwise. The non-linear function σ is commonly referred to as the activation function. The input layer is defined as the identity mapping:

$$f^{(0)}(\mathbf{x}) = \mathbf{h}^{(0)} = \mathbf{x}, \quad \mathbf{x} \in \mathbb{R}^{d_{\text{in}}}.$$

This definition, applied gradually through all L layers, yields the output of the FNN:

$$\hat{\mathbf{y}} = f_\theta(\mathbf{x}) = \Big(f^{(L)} \circ f^{(L-1)} \circ \dots \circ f^{(0)}\Big)(\mathbf{x}) = \mathbf{h}^{(L)}, \quad \hat{\mathbf{y}} \in \mathbb{R}^{d_{\text{out}}}.$$

The central question now is how to optimize the parameters θ such that the network output $\hat{\mathbf{y}}$ approximates a given or implicitly derived target $\mathbf{y} \in \mathbb{R}^{d_{\text{out}}}$.

2.3.3 Loss Function and Gradient Descent

In machine learning, we typically consider a dataset rather than a data matrix. Let the dataset be represented as a multiset of inputs $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^N,\ \mathbf{x}^{(i)} \in \mathbb{R}^{d_{\text{in}}}$. If explicit targets are available, as in supervised learning, or if they can be derived implicitly from the data, as in unsupervised learning, then we write the dataset in labeled form as $\mathcal{D} = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^N,\ \mathbf{y}^{(i)} \in \mathbb{R}^{d_{\text{out}}}$.

To measure the discrepancy between the FNN output and its corresponding target, we introduce a loss function:

$$\mathcal{L} : \mathbb{R}^{d_{\text{out}}} \times \mathbb{R}^{d_{\text{out}}} \to \mathbb{R}.$$

To define the total loss over the entire dataset, individual losses — each depending on the parameters θ — are averaged:

$$L(\theta) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}\Big(f_\theta(\mathbf{x}^{(i)}), \mathbf{y}^{(i)}\Big).$$

This prevents the objective from scaling with the number of samples.

Training an FNN corresponds to finding parameters θ that minimize the total loss. This can be formulated as an optimization problem:

$$\min_\theta \ L(\theta).$$

Since the total loss is generally a non-convex and high-dimensional function, exact minimization is infeasible. Instead, iterative optimization algorithms are employed that update the parameters in the direction of the negative loss gradient. The most basic approach is gradient descent, which updates the parameters according to:

$$\theta^{(t+1)} = \theta^{(t)} - \eta \, \nabla_\theta L(\theta^{(t)}),$$

where ∇ denotes the gradient, $\eta > 0$ the learning rate, and $t \in \mathbb{N}$ the iteration index.

In practice, computing the gradient over the entire dataset is computationally expensive and can lead to poor convergence. Advanced optimizers address this by adapting the learning rate during training, which improves efficiency and stability. However, optimization alone is often insufficient in determining convergence. The architecture of the FNN also plays a crucial role in this regard.

2.3.4 Residual Connection

Training of deep FNNs, formed by stacking many layers, can be impeded by the problem of vanishing or exploding gradients [17]. In such cases, gradients become either extremely small as they are propagated through the network, causing early layers to learn very slowly or stop updating entirely, or grow excessively large, making training unstable. Residual connections, first introduced by He et al. [16], provide an effective mechanism to alleviate this issue by bypassing one or more layers and adding the input directly to the corresponding output.

Formally, a residual connection can be written as:

$$\mathbf{h}^{(l+i)} = f^{(l+i)}(\mathbf{h}^{(l+i-1)}) = \left(f^{(l+i-1)} \circ f^{(l+i-2)} \circ \cdots \circ f^{(l)}\right)(\mathbf{h}^{(l-1)}) + \mathbf{h}^{(l-1)}.$$

Note that $d_{l-1} = d_{l+i}$ for the addition to be well-defined.

To further enhance gradient flow, many architectures apply normalization techniques after certain layers.

2.3.5 Batch Normalization

Training efficiency is increased through computational parallelization; therefore, neural networks are typically trained on mini-batches of size B rather than on individual samples.

Given a dataset of size N, the number of mini-batches is $N_{\text{batch}} = \left\lfloor \frac{N}{B} \right\rfloor$. A mini-batch can be written as:

$$\mathcal{B}^{(k)} = \{\mathbf{x}^{(i)}\}_{i=1}^{B}, \quad k = 1, \ldots, N_{\text{batch}},$$

where each mini-batch element is represented as a vector $\mathbf{x}^{(i)} \in \mathbb{R}^d$.

Batch Normalization (BN), first introduced by Ioffe and Szegedy [20], stabilizes and accelerates training by normalizing the activations within each mini-batch.

For a given mini-batch $\mathcal{B}^{(k)}$, BN computes the empirical mean and variance for each dimension $j = 1, \ldots, d$, and applies the transformation:

$$\tilde{x}_j^{(i)} = \frac{x_j^{(i)} - \hat{\mu}_j^{(k)}}{\sqrt{(\hat{\sigma}_j^{(k)})^2 + \epsilon}}, \quad i = 1, \ldots, B,$$

where $\hat{\mu}_j^{(k)}$ and $(\hat{\sigma}_j^{(k)})^2$ denote the mean and variance of the j-th feature in the k-th batch, and $\epsilon > 0$ is a small constant added for numerical stability.

Even though residual connections and batch normalization have been shown to increase gradient flow and computational stability, the problem of complexity in FNNs with fully-connected layers remains. The following section describes how convolutional layers can be used to reduce the number of trainable parameters and improve generalization for datasets that feature some degree of translational equivariance.

2.3.6 Convolutional Neural Networks

The most defining building block of a Convolutional Neural Network (CNNs) is the convolutional layer. Unlike a fully connected layer, it applies a sliding dot product between a kernel — a tensor whose entries are parameters optimized during training — and local regions of its input. The result of this operation is referred to as a feature map, where each kernel produces its own feature map.

An important property of a convolutional layer is translational equivariance. This means that if its input is shifted, the corresponding feature map shifts in the same way. In other words, the layer preserves the spatial and temporal structure of patterns rather than depending on their absolute position. This property is especially valuable for EEG data analysis. Seizure-related patterns in EEG do not necessarily occur at fixed positions in time or space; instead, their diagnostic value lies in their local structure and dynamics. Translational equivariance allows CNNs to recognize such patterns regardless of their position, making them particularly well-suited for automated seizure detection.

In the following, we focus on the mathematical formulation of one-dimensional convolutional layers. Other variants, such as two-dimensional convolutions, are particularly useful for image data, but are beyond the scope of this thesis and will not be discussed further.

At the core of a one-dimensional convolutional layer lies the valid cross-correlation operation, a concept originating from the field of digital signal processing, here applied to discrete inputs. We denote the input vector $\mathbf{x} \in \mathbb{R}^d$, and the so-called kernel as $\mathbf{k} \in \mathbb{R}^m$, where $d, m \in \mathbb{N}$ with $d \geq m$. The valid cross-correlation of $\mathbf{x}$ and $\mathbf{k}$ is a new vector$\mathbf{r} \in \mathbb{R}^{d-m+1}$, defined componentwise by:

$$r_j = (\mathbf{k} \star \mathbf{x})_j = \sum_{i=1}^{m} k_i \, x_{j+i-1}, \quad j = 1, 2, \ldots, d - m + 1,$$

where $\star$ is a shorthand notation for the valid cross-correlation.

In a convolutional layer, the valid cross-correlation operation is applied systematically across the input, often with multiple kernels in parallel, to extract local features that are then propagated through the network. A convolutional layer can be regarded as a matrix-valued function:

$$f^{(l)} : \mathbb{R}^{F_{l-1} \times T_{l-1}} \longrightarrow \mathbb{R}^{F_l \times T_l},$$

where F_{l-1} and T_{l-1} denote the matrix input dimensions of layer l, commonly referred to as the number of input feature maps and the input length, respectively. Given an input to the convolutional layer l:

$$\mathbf{H}^{(l-1)} = \begin{bmatrix} (\mathbf{u}^{(1)})^\top \\ (\mathbf{u}^{(2)})^\top \\ \vdots \\ (\mathbf{u}^{(F_{l-1})})^\top \end{bmatrix}, \quad \mathbf{u}^{(p)} \in \mathbb{R}^{T_{l-1}}, \quad p = 1, 2, \cdots, F_{l-1}$$

the output of the layer is:

$$f^{(l)}(\mathbf{H}^{(l-1)}) = \mathbf{H}^{(l)} = \begin{bmatrix} (\mathbf{o}^{(1)})^\top \\ (\mathbf{o}^{(2)})^\top \\ \vdots \\ (\mathbf{o}^{(F_l)})^\top \end{bmatrix}, \quad \mathbf{o}^{(q)} \in \mathbb{R}^{T_l}.$$

For each output feature map $q = 1, 2 \cdots, F_l$, we compute:

$$\mathbf{o}^{(q)} = \sigma\left(\sum_{p=1}^{F_{l-1}} \mathbf{u}^{(p)} \star \mathbf{k}^{(p,q)} + b^{(q)} \right).$$

Here, $\mathbf{k}^{(p,q)} \in \mathbb{R}^m$ is the convolution kernel of length m connecting input feature map p to output feature map q, and $b^{(q)} \in \mathbb{R}$ is the bias associated with output feature map q. The output length is given by $T_l = T_{l-1} - m + 1$. For simplicity, additional parameters such as stride, padding, and dilation are omitted here, although they are frequently used in practice.

CNNs offer a powerful foundation for EEG analysis, but they are typically employed in sole classification tasks where the ability to reconstruct the input is not a primary concern. In our case, however, the goal remains to explore how a lower-dimensional encoding of EEG data can facilitate seizure assessment. To this end, we propose the use of reconstruction-based models, which oftentimes incorporate CNNs as part of their architecture.

2.4 Autoencoders

Autoencoders are a specialized type of feedforward neural network designed to learn efficient data representations through reconstruction. Given their central role in this thesis, they are hereby discussed in a dedicated chapter. Unless otherwise stated, the presentation and terminology are based on Prince [33].

2.4.1 Deterministic Autoencoders

DAEs were first introduced by Rumelhart et al. [34] as neural networks trained to reconstruct their inputs as accurate as possible. Their main objective is to encode the input into a compressed and informative representation and then decode it back so that the reconstruction closely matches the original input. As illustrated in Figure 2.3, the compression is performed by the encoder, which maps the input into a lower dimensional representation, which will be referred to as the latent encoding or latent representation. The reconstruction step is performed by the decoder, which maps the latent encoding back into the original input space.

Their flexible network-based architecture imposes few restrictions on the data and thus allows effective reduction of our high-dimensional and noisy EEG signals while preserving information of interest [1]. Recent research has demonstrated the potential of autoencoders in EEG signal reconstruction, providing a promising foundation for further exploration in seizure detection.

Similar to PCA, DAEs are used for unsupervised dimensionality reduction. In contrast to PCAs analytical solution, however, DAEs rely on neural network architectures and iterative optimization during training.

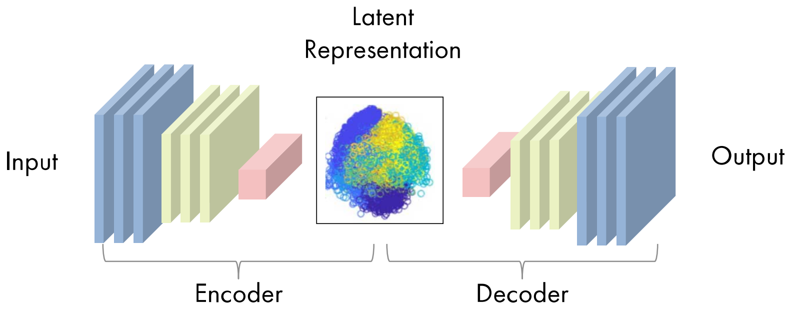


Figure 2.3: Schematic illustration of an autoencoder architecture [26]. The encoder maps the input to latent representations, and the decoder aims to reconstruct the original input from these latent encodings.

We now formalize these concepts with more mathematical rigor. An DAE consists of two main components:

1. Encoder: a parametric function

$$f_\theta : \mathbb{R}^d \to \mathbb{R}^k, \quad d, k \in \mathbb{N},$$

   typically represented by a neural network with parameters θ.

2. Decoder: a parametric function

$$g_\phi : \mathbb{R}^k \to \mathbb{R}^d,$$

   commonly depicted by a neural network with parameters ϕ.

Applying the encoder to an input $\mathbf{x} \in \mathbb{R}^d$ yields the latent representation:

$$z = f_\theta(\mathbf{x}), \quad z \in \mathbb{R}^k.$$

The decoder maps the latent representation back to the input space:

$$\hat{\mathbf{x}} = g_\phi(z) = g_\phi(f_\theta(\mathbf{x}))\,.$$

This can be seen as a reconstruction of the original input.

To evaluate the reconstruction quality, a loss function is defined:

$$\mathcal{L} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}.$$

It measures the discrepancy between input $\mathbf{x}$ and reconstruction $\hat{\mathbf{x}}$. Given a dataset $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^N$, where each observation is given by $\mathbf{x}^{(i)} \in \mathbb{R}^d$, the training objective of the DAE is to find parameters θ and ϕ that minimize the empirical mean of reconstruction errors across the dataset:

$$\min_{\theta,\phi} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}\left(\mathbf{x}^{(i)}, \hat{\mathbf{x}}^{(i)}\right).$$

A key limitation of DAEs is that their training objective places emphasis exclusively on reconstruction. Because the latent space is not explicitly constrained, the resulting encodings may capture only the information needed to reconstruct the input, without organizing it in a way that reflects meaningful structure in the data. As a consequence, the learned representations can be of limited value for downstream tasks such as classification. More advanced forms of autoencoders address this shortcoming by imposing a regularization on the latent space, encouraging the model to learn representations that are both more generalizable and more interpretable.

2.4.2 Variational Autoencoders

VAEs extend the DAE framework by introducing a probabilistic formulation of the latent space and were first proposed by Kingma and Welling [23]. Instead of mapping each input to a single deterministic latent encoding, the encoder outputs the parameters of a probability distribution over latent variables. A latent representation is sampled from this distribution and passed to the decoder, which outputs the parameters of a probability distribution over the input space, from which reconstructions of the data can be generated.

This probabilistic formulation enables the model to approximate the joint distribution between EEG signals and their latent representations, which is particularly helpful for distinguishing different types of brain activity [40]. For example, seizure patterns may cluster in distinct regions of the latent space compared to background activity. Moreover, the continuity of the latent distribution allows for meaningful interpolation between encodings, providing insight into smooth transitions between different brain states.

Let $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N)} \overset{\text{i.i.d.}}{\sim} \Pr(x)$, $\mathbf{x}^{(i)} \in \mathbb{R}^d$, $i = 1, \dots, N$, denote independent and identically distributed (i.i.d.) EEG observations. Similarly, let $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \dots, \mathbf{z}^{(N)} \overset{\text{i.i.d.}}{\sim} \Pr(z)$, $\mathbf{z}^{(i)} \in \mathbb{R}^k$, $i = 1, \dots, N$, denote i.i.d. samples of the corresponding latent variables. Assume there exists an unknown joint distribution $\Pr(\mathbf{x}, \mathbf{z})$ that models both the data and its latent representation. The data distribution can then be written as the marginalization of this joint distribution:

$$\Pr(\mathbf{x}) = \int \Pr(\mathbf{x}, \mathbf{z})\, d\mathbf{z}.$$

By the general product rule, this marginal can be expressed as:

$$\Pr(\mathbf{x}) = \int \Pr(\mathbf{x} \mid \mathbf{z})\ \Pr(\mathbf{z})\, d\mathbf{z}.$$

Although this formulation is indirect, it is powerful: even when $\Pr(\mathbf{x})$ is complex, it may be tractable to specify simple forms for the likelihood $\Pr(\mathbf{x} \mid \mathbf{z})$ and the prior $\Pr(\mathbf{z})$. A common assumption in the literature is to use a standard multivariate normal distribution as the prior:

$$\Pr(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

and a Gaussian likelihood whose mean is given by a nonlinear function $f_\phi(\mathbf{z})$ with spherical covariance:

$$\Pr(\mathbf{x} \mid \mathbf{z}, \phi) = \mathcal{N}(f_\phi(\mathbf{z}),\, \sigma^2 \mathbf{I}).$$

Here, f_ϕ is typically described by a deep neural network with parameters ϕ. It should be able to capture the essential structure of the data, while the covariance accounts for residual variability.

To train the model, we maximize the log-likelihood over the dataset $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^N$:

$$\hat{\phi} = \arg\max_\phi \sum_{i=1}^{N} \log \Pr\left(\mathbf{x}^{(i)} \mid \phi\right),$$

where

$$\Pr\left(\mathbf{x}^{(i)} \mid \phi\right) = \int \mathcal{N}(f_\phi(\mathbf{z}^{(i)}),\, \sigma^2 \mathbf{I})\, \mathcal{N}(\mathbf{0}, \mathbf{I})\, d\mathbf{z}.$$

Unfortunately, this integral is intractable in closed form. Referring to the literature, we can formulate a lower bound on the log-likelihood, which depends on the parameters ϕ as well as on parameters ψ of another distribution. Eventually, we will build a neural network to compute this lower bound and optimize it. This lower bound is known as the evidence lower bound (ELBO) and can be written as:

$$\mathrm{ELBO}(\phi, \psi) = \int q(\mathbf{z} \mid \mathbf{x}, \psi) \log \Pr(\mathbf{x} \mid \mathbf{z}, \phi)\ d\mathbf{z} - D_{\mathrm{KL}}(q(\mathbf{z} \mid \mathbf{x}, \psi) \,\|\, \Pr(\mathbf{z}))\,,$$

where D_{KL} denotes the Kullback–Leibler divergence, and $q(\mathbf{z} \mid \mathbf{x}, \psi)$ is an auxiliary probability distribution depending on parameters ψ. This distribution should approximate the true posterior:

$$\Pr(\mathbf{z} \mid \mathbf{x}, \phi) = \frac{\Pr(\mathbf{x} \mid \mathbf{z}, \phi)\Pr(\mathbf{z})}{\Pr(\mathbf{x} \mid \phi)},$$

which is again intractable due of the normalizing denominator as stated above.

We choose a simple parametric distribution family to approximate the posterior, e.g. a multivariate Gaussian:

$$q(\mathbf{z} \mid \mathbf{x}, \psi) = \mathcal{N}(g_{\psi}^{\mu}(\mathbf{x}),\, g_{\psi}^{\sigma}(\mathbf{x})),$$

where g_ψ is another neural network with parameters ψ predicting the mean μ and variance σ of the approximation q. Given the approximate posterior, we can draw samples $\mathbf{z}^{(i)}$ by first computing its mean and variance using our second neural network g_ψ. The expectation in the ELBO can then be approximated with a Monte Carlo estimate using L samples $\{\mathbf{z}^{(i)}\}_{i=1}^{L}$ drawn from $q(\mathbf{z} \mid \mathbf{x}, \psi)$:

$$\int q(\mathbf{z} \mid \mathbf{x}, \psi) \log \Pr(\mathbf{x} \mid \mathbf{z}, \phi) \approx \frac{1}{L} \sum_{i=1}^{L} \log \Pr\left(\mathbf{x} \mid \mathbf{z}^{(i)}, \phi\right).$$

In the common case $L = 1$, this simplifies to:

$$\mathrm{ELBO}(\phi, \psi) \approx \log \Pr\left(\mathbf{x} \mid \mathbf{z}^{(1)}, \phi\right) - D_{\mathrm{KL}}\left(q(\mathbf{z} \mid \mathbf{x}, \psi) \parallel \Pr(\mathbf{z})\right),$$

where $\mathbf{z}^{(1)}$ is a single sample from $q(\mathbf{z} \mid \mathbf{x}, \psi)$. The above makes explicit that the ELBO consists of two components: the first term corresponds to the expected log-likelihood and reflects the reconstruction quality of the model, while the second term measures the divergence between the approximate posterior and the prior distribution, thereby acting as a regularizer on the latent space. When a Gaussian likelihood with fixed variance is assumed, maximizing the expected log-likelihood is equivalent to minimizing the mean squared error (MSE) between the original data and its reconstruction.

With the complete formulation of the VAE, we refer to the function g_ψ as the encoder and the function f_ϕ as the decoder. The ELBO is computed with respect to the parameters ϕ and ψ, which are then optimized accordingly.

Since the architecture involves a sampling step, direct differentiation is not possible. This difficulty is resolved using the reparameterization trick:

$$\mathbf{z}^{(i)} = \boldsymbol{\mu} + \boldsymbol{\Sigma}^{\frac{1}{2}} \boldsymbol{\epsilon}^{(i)},$$

where $\boldsymbol{\epsilon}^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

By now, we already introduced two distinct approaches for unsupervised dimensionality reduction: the linear method of PCA and the more flexible, neural-network-based framework of autoencoders. Both methods yield latent encodings of the EEG data, and importantly, raise the question whether these representations entail information relevant for distinguishing seizure from non-seizure activity. This question can naturally be cast as a classification task. Since our aim is not to achieve state-of-the-art classification performance, but rather to explore whether the latent spaces learned to capture seizure-related structure in an unsupervised manner, we introduce two simple classifiers as indicators of the information contained in the encodings.

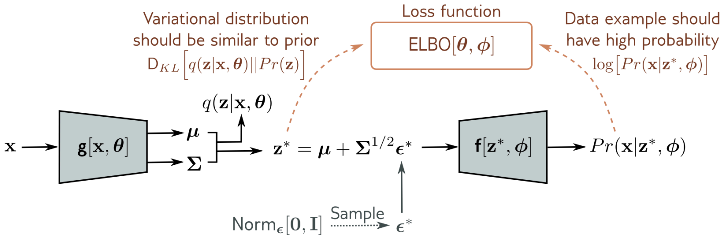

Variational distribution
should be similar to prior
$D_{KL}\left[q(\mathbf{z}|\mathbf{x},\boldsymbol{\theta})||Pr(\mathbf{z})\right]$
Loss function
$\text{ELBO}[\boldsymbol{\theta},\phi]$
Data example should
have high probability
$\log\left[Pr(\mathbf{x}|\mathbf{z}^*,\phi)\right]$
$q(\mathbf{z}|\mathbf{x},\boldsymbol{\theta})$
$\mathbf{x}$
$\mathbf{g}[\mathbf{x},\boldsymbol{\theta}]$
$\boldsymbol{\mu}$
$\boldsymbol{\Sigma}$
$\mathbf{z}^* = \boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{\epsilon}^*$
$\mathbf{f}[\mathbf{z}^*,\phi]$
$Pr(\mathbf{x}|\mathbf{z}^*,\phi)$
$\text{Norm}_{\epsilon}[\mathbf{0},\mathbf{I}]$
Sample
$\boldsymbol{\epsilon}^*$

Figure 2.4: Schematic overview of the variational autoencoder framework. The encoder produces the mean and covariance of the variational distribution $q(z \mid x, \theta)$,from which latent samples are drawn using the reparameterization trick. The decoder then generates a reconstruction, and training maximizes the ELBO to balance data likelihood and prior regularization. Reproduced from Prince [33] under CC BY-NC-ND 4.0

2.5 Classifiers

In general, a classifier is a function that maps an input to one of a finite number of class labels, with the goal of learning a decision rule from data that generalizes well to previously unseen samples.

Since our task is to distinguish latent representations associated with seizure versus non-seizure activity, we focus on the binary classification setting. The presentation and notation in this chapter follow the standard treatment of Bishop [7].

2.5.1 Logistic Regression

Logistic regression is a simple and widely used method for binary classification. Its goal is to learn a linear decision boundary in the input space that separates two distinct classes. By optimizing the cross-entropy loss, logistic regression adjusts its parameters such that the predicted labels align closely with the underlying truth.

Let the latent encodings be collected in a multiset $\mathcal{E} = \{(\mathbf{z}^{(i)}, y^{(i)})\}_{i=1}^{N}$, where each latent encoding is $\mathbf{z}^{(i)} \in \mathbb{R}^k$ and corresponding label is $y^{(i)} \in \{0, 1\}$, where $y^{(i)} = 1$ indicates seizure activity and $y^{(i)} = 0$ indicates background activity. Logistic regression models the relationship between the latent encodings and the binary target by forming a linear combination of the inputs, $\ell^{(i)} = \mathbf{w}^\top \mathbf{z}^{(i)} + b$, where $\mathbf{w} \in \mathbb{R}^k$ and $b \in \mathbb{R}$. This linear combination is then mapped to a probability via the sigmoid function, defined as:

$$\hat{y} = \sigma(\ell) = \frac{1}{1 + e^{-\ell}}.$$

The resulting conditional probabilities are given by $\Pr(y^{(i)} = 1 \mid \mathbf{z}^{(i)}) = \sigma(\ell^{(i)}) =$

$\hat{y}^{(i)}$, and $\Pr(y^{(i)} = 0 \mid \mathbf{z}^{(i)}) = 1 - \hat{y}^{(i)}$. Consequently, if the true label is $y^{(i)} = 1$, we want $\hat{y}$ to be as close to 1 as possible, and vice versa.

To achieve this, logistic regression minimizes the binary cross-entropy loss. It is defined as:

$$\mathcal{L}(y, \hat{y}) = -y \log \hat{y} - (1 - y) \log(1 - \hat{y}).$$

So the total loss can be defined as follows:

$$L(\mathbf{w}, b) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(y^{(i)}, \hat{y}^{(i)}).$$

The parameters $\mathbf{w}$ and b are optimized via gradient descent to minimize this loss. Finally, classification is performed by applying a threshold to the predicted probability:

$$\hat{y}^{(i)} > \tau, \quad 0 \leq \tau \leq 1,$$

the sample is then classified as seizure if the inequality holds, and as background otherwise.

In summary, logistic regression provides a simple way to test whether seizure-related information is present in the latent encodings, but it relies on the assumption of linear separability, which may not always hold. To complement logistic regression, we introduce a simple non-linear classifier in the next chapter.

2.5.2 k-Nearest-Neighbors

The k-nearest neighbors (k-NN) algorithm is a simple non-linear classification method. Given a latent encoding $\mathbf{z} \in \mathbb{R}^k$, the algorithm identifies the k closest samples in latent space according to some distance metric. The predicted label is then assigned based on the majority class among these neighbors. Unlike logistic regression, k-NN makes no assumptions about linear separability, allowing it to capture more complex, local decision boundaries in the latent space.

Having introduced the theoretical foundations necessary to understand our approach, we now turn to the practical aspects of this work. The following section describes the dataset and preprocessing steps, as well as the model architecture details, training procedure, and evaluation methods employed in our experiments.

Chapter 3

Material and Methods

3.1 Dataset

For this work, the TUH EEG Seizure Corpus [36](TUSZ) was selected as the dataset. TUSZ is a widely used benchmark for seizure detection in machine learning and is publicly available upon request. It consists of EEG recordings collected at Temple University Hospital in Philadelphia, Pennsylvania. It was developed with the explicit goal of advancing state of the art algorithms for automated seizure analysis. TUSZ is a curated subset of the broader TUH EEG Corpus, containing recordings that are known to include seizure activity. To ensure balance, the dataset also provides recordings of normal background EEG without seizures. In total, the corpus comprises more than 7,000 recordings from 674 patients, amounting to approximately 81 GB of data. Recordings vary in length, ranging from a few seconds to several minutes, and individual patients may appear in multiple recording sessions. The dataset reflects the natural imbalance of EEG recordings, where seizures occur less frequently than non-seizure activity. Out of the total 1,474.60 hours of recordings, approximately 76.02 hours correspond to seizure activity, which amounts to a ratio of 5% seizure to 95% non-seizure activity. EEG signals were recorded from scalp electrodes placed according to the international 10–20 system, with sampling rates varying between 250 Hz and 1,000 Hz. Each recording was reviewed by neurologists, who annotated the presence of seizures and, when applicable, marked their onset and offset times.

3.2 Data preprocessing

Before training the model on seizure data, several preprocessing steps were applied in order to transform the raw EEG recordings into a standardized form. Our preprocessing pipeline was inspired by Zhao et al. [40], but since their work relied on a different, non-public dataset, we adapted it to the characteristics of the TUSZ dataset. A schematic overview of the preprocessing steps is shown in Fig. 3.1.

The TUSZ dataset contains $N_{\text{rec}} = 7361$ EEG recordings. Each recording can be represented as a matrix, with rows corresponding to the channels and columns to the time points, sampled at a fixed frequency over the duration of the recording. Since the sampling frequency varied across recordings all signals were resampled to a common rate of 256 Hz. Channels were selected according to the standard 10–20 system, with the average reference applied.

To obtain frequency-specific information, we applied finite impulse response filters. By setting appropriate lower and upper filter cutoff frequencies, we separated the signal into five distinct frequency bands:

$$\delta : 1-4\,\text{Hz}, \quad \theta : 4-8\,\text{Hz}, \quad \alpha : 8-13\,\text{Hz}, \quad \beta_{\text{low}} : 13-20\,\text{Hz}, \quad \beta_{\text{high}} : 20-30\,\text{Hz}.$$

Note that, as mentioned in Section 2.1, frequencies relevant for seizure detection typically extend 30 Hz. To ensure consistency with the preprocessing pipeline of Zhao et al. [40], these higher frequencies, commonly referred to as the gamma band, are excluded in this work. Recording gamma oscillations with EEG is particularly challenging, due to contamination from cranial, ocular, and distant muscle activity [31]. The filtering yielded five band-limited versions of each recording, which were transformed further in the subsequent preprocessing steps. The following segmentation into fixed-length windows was performed on each band-pass filtered signal.

To avoid initialization artifacts, the first 60 seconds of each recording were discarded. The remaining signal was divided into non-overlapping segments of five seconds each to extract signals of heterogeneous length. Each segment was included only if its amplitude did not exceed a fixed threshold of 400 μV, thereby excluding segments containing artifacts that exceed typical EEG signal amplitudes. To mitigate overrepresentation from long recordings, the number of extracted segments per recording was limited to a maximum of 500, with longer recordings subsampled accordingly. Conversely, recordings contributing insufficient data were excluded if they yielded fewer than 200 segments.

Applying this pipeline repeatedly to all recordings, and using the same segment boundaries for each band-pass filtered signal, yields one dataset for each of the five frequency bands:

$$\mathcal{D}_b = \{\mathbf{S}_b^{(i)}\}_{i=1}^{N}, \quad b \in \{\delta, \theta, \alpha, \beta_{low}, \beta_{high}\},$$

where N is the total number of extracted segments across all recordings and b is the band index. Each segment can be expressed as a matrix of the form $\mathbf{S}^{(i)} \in \mathbb{R}^{C\times T}$, where $C = 19$ denotes the number of channels, and $T = f \cdot L = 1280$ is the number of time points, with a sampling frequency of $f = 256\,\text{Hz}$ and a recording duration of $L = 5\,\text{s}$, respectively.

With this pipeline, we transformed the raw EEG recordings into a structured and standardized dataset, segmented and filtered across five frequency bands. In the next section, we describe the architecture of the autoencoder frameworks designed to model these preprocessed signals.

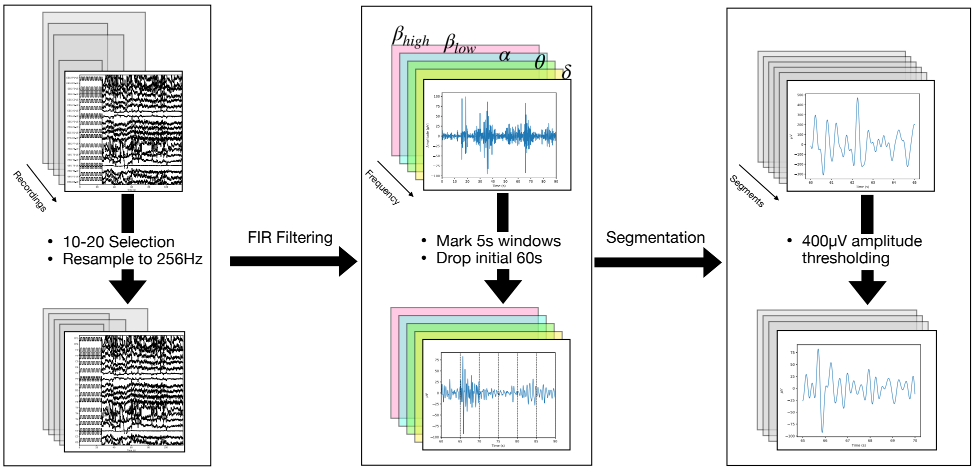


Figure 3.1: Schematic overview of preprocessing

3.3 Model Architecture

Since we implemented two types of autoencoders — a VAE and a DAE — for comparison, it is important to clarify their relationship. The DAE follows the same architecture as the VAE, but instead of sampling from a latent distribution, it directly uses the encoders mean as the latent representation. In the following, we describe the architecture of the VAE, noting that the DAE can be viewed as its deterministic special case.

The design was inspired by Zhao et al. [40] and named a Variational Autoencoder for Electroencephalography (VAEEG) by its authors. In section 3.2, we described the general preprocessing setup for multichannel EEG. In contrast, Zhao et al. [40] simplified the task by choosing single channels for each segment sequentially, rather than retaining all channels at once, yielding a dataset for each frequency band of the form $\mathcal{D}_b = \{\mathbf{s}_b^{(j)}\}_{j=1}^{CN}$, where each band-passed, single-channel segment can be expressed as a vector $\mathbf{s}_b^{(j)} \in \mathbb{R}^T$. C, N, T, b denote the number of channels, the number of extracted segments, the number of time points, and the band index, respectively, as described above. This single-channel approach enlarges the dataset and simplifies both the architecture and the learning task of the VAEEG, and was therefore adopted in our work, following Zhao et al. [40]. It should be noted, however, that seizures often manifest in spatially localized patterns in addition to temporal dynamics [3]. Thus, training a model on multichannel data may capture additional relevant information — a possibility which we will revisit in Section 5. In the following, we describe the architecture of the VAEEG used in our experiments.

A schematic overview of the VAEEG architecture is shown in Fig. 3.2. For each frequency band, a dedicated encoder processes a single-channel segment. A Gaussian distribution is used as the prior in the latent space, and latent variables are sampled accordingly. Each encoder maps the input to the distribution parameters $\mu, \sigma^2 \in \mathbb{R}^k$. The decoder, which mirrors the encoder structure, then reconstructs the signal by mapping the sampled latent representation back into the input space.

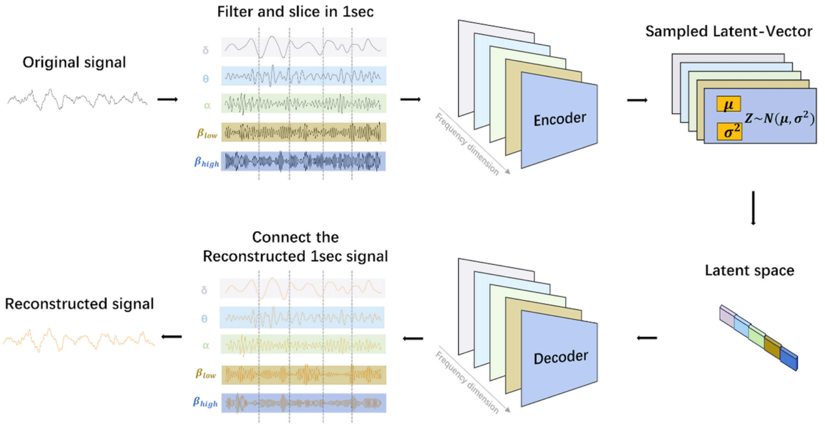


Figure 3.2: Schematic overview of the VAEEG workflow. Band-pass filtered and segmented EEG signals are passed through the encoder to obtain the mean and variance of the latent distribution. Latent representations are then sampled and processed by the decoder to reconstruct the original band-pass filtered segments. Reproduced from Zhao et al. [40] under CC BY 4.0.

The comprehensive VAEEG architecture is shown in Figure 3.3. In this diagram, *Linear* denotes a fully connected layer, *BatchNorm* refers to batch normalization, and *Conv1d* represents a convolutional layer as described in Section 2.3.6. As activation function, *Leaky ReLU* is chosen. It is defined as

$$\text{LeakyReLU}(x) = \begin{cases} x & \text{if } x \geq 0, \\ \alpha \cdot x & \text{otherwise} \end{cases}$$

where $\alpha > 0$ is a small constant that controls the slope in the negative half-plane. *ResBlock* is a subnetwork known to work well in deep neural networks and to alleviate the vanishing or exploding gradient problem. Aside from batch normalization and convolutional layers it also integrates a residual connection. The *Multi-Head CNN* is a subnetwork used exclusively in the encoder. Its core idea is to process the same input through multiple independent convolutional layers — here referred to as heads — and then concatenate their outputs before applying a final convolutional layer to fuse the combined representation. Recent work has shown that such parallel multi-head architectures can be advantageous, particularly when using different kernel sizes to capture multiple receptive fields of varying scales simultaneously [22, 37, 39]. This design can be understood as a way of encouraging the network to extract complementary features from the same EEG signal, since each head starts from an independent initialization and may converge to detecting distinct patterns. The subsequent convolution over the concatenated outputs then serves to integrate these diverse representations into a unified feature map for further processing.

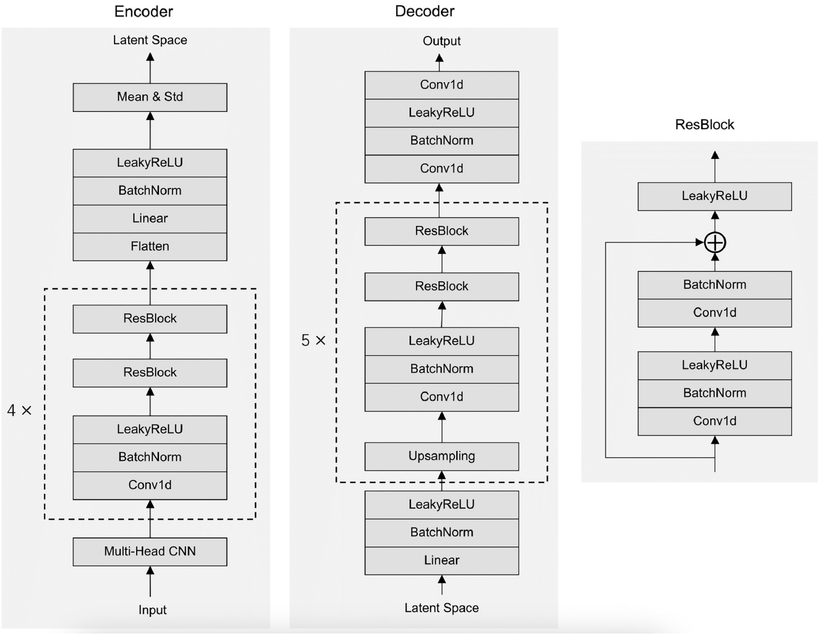


Figure 3.3: Architecture of the proposed VAEEG model. The encoder processes the input through a multi-head CNN, residual blocks, and linear layers to produce the mean and standard deviation of the latent distribution. Using the reparameterization trick, latent samples are passed to the decoder, which applies linear layers, convolutional layers, and residual blocks to reconstruct the original EEG segment. Reproduced from Zhao et al. [40] under CC BY 4.0.

Having introduced the architecture of the VAEEG and its components, we now turn to the training procedure, where we explain hyperparameter choices and optimization setup used to train the models on the preprocessed EEG data.

3.4 Training Procedure

Training procedure follows established practices in machine learning, with particular emphasis on efficiency and reproducibility. Key aspects include the choice of hyperparameters, the strategy for splitting the data into training and test sets, and the methods used for evaluation.

In our experiments, we trained one autoencoder per frequency band. To assess the relative merits of probabilistic and deterministic formulations, we compared a VAE with a DAE. For parameter optimization, we minimized the negative ELBO, which can be expressed as:

$$-\mathrm{ELBO}(\phi, \psi) \approx \|\mathbf{x} - \hat{\mathbf{x}}\|^2 + D_{\mathrm{KL}}\big(q(\mathbf{z} \mid \mathbf{x}, \psi) \| \Pr(\mathbf{z})\big).$$

It should be noted that all hyperparameters were chosen in accordance with Zhao et al. [40]. The dataset was randomly split into a training set and a test set, with each set consisting of individual segments. We used 90% of the data for training and the remaining 10% for testing. The training set was used to iteratively optimize the model parameters, while the test set provided an independent evaluation of generalization performance. Training proceeded in multiple epochs, with one epoch defined as a full pass through the training data. Each model was trained for a maximum of 50 epochs. In line with the approach outlined in Section 2.3.5, we trained the models using mini-batches of size 1024 to efficiently exploit parallel computation. Although the preprocessing pipeline, as described in Section 3.2, generates fixed 5 s segments, during each training iteration the data loader randomly samples a 1 s sub-segment from the 5 s window. This procedure serves as a form of data augmentation, increasing the signal variability encountered by the model and thereby improving generalization without enlarging the stored dataset. Optimization was performed using the RMSprop algorithm, which dynamically adapts the learning rate during training. The initial learning rate was set to 0.001, a commonly effective starting point that balances convergence stability and training speed [19]. Checkpoints were saved after each epoch, allowing us to retain the model with the lowest training loss. We set $\beta = 1$ for the VAE, while the DAE objective focused solely on reconstruction. The activation function used throughout was LeakyReLU with a negative slope coefficient of 0.2, a commonly adopted setting that balances representational capacity and stability. We chose 50 as the number of latent dimensions. Four partitions of an NVIDIA A100 GPU were used for training. Each autoencoder required approximately 3.5 hours of training, resulting in a total training time of about 17.5 hours for all frequency bands.

Having described the training setup, we now introduce the evaluation measures used to assess the models. These include metrics for reconstruction quality as well as analyses of the latent representations with respect to seizure-related information.

3.5 Evaluation

3.5.1 Reconstruction

Reconstruction performance of the autoencoder frameworks was assessed using several metrics on a held-out test set. It was first quantified using the mean squared error (MSE) between the original and reconstructed EEG signal segments. Secondly, as a complementary measure, we recorded the Pearson correlation coefficient (PCC) between each input and its reconstruction, to provide a scale-invariant measure of signal similarity.

The MSE between the original signals and their reconstructions is defined as:

$$\text{MSE}(\mathbf{x}, \hat{\mathbf{x}}) \;=\; \frac{1}{d} \sum_{i=1}^{d} (x_j - \hat{x}_j)^2 \,,$$

where $\mathbf{x}, \hat{\mathbf{x}} \in \mathbb{R}^d$ denote the original and reconstructed signals, respectively. The PCC between two signals is defined as:

$$\text{PCC}(\mathbf{x}, \hat{\mathbf{x}}) = \frac{\sum_{j=1}^{d} (x_j - \mu)\,(\hat{x}_j - \hat{\mu})}{\sqrt{\sum_{j=1}^{d} (x_j - \mu)^2}\,\sqrt{\sum_{j=1}^{d} (\hat{x}_j - \hat{\mu})^2}},$$

where μ and $\hat{\mu}$ denote the arithmetic mean of the original signal and the arithmetic mean of the reconstruction, respectively.

Since we trained a separate autoencoder for each frequency band, we obtained individual reconstructions for each band and could therefore combine them to approximate the original full-band EEG signal. To assess reconstruction quality on the full signal, we summed the band-specific reconstructions to reconstruct the unfiltered signal and then computed both the MSE and PCC on these full-band reconstructions across the test set.

PCA was included as a baseline for evaluating the reconstruction performance of the autoencoder frameworks. We chose the number of PCs to match the number of latent dimensions used in the autoencoder frameworks, and we therefore also set it to 50. Reconstructions obtained from PCA were evaluated with the same metrics as mentioned above.

3.5.2 Latent Space Analysis

Beyond reconstruction quality, we further analyzed the latent representations to investigate whether they capture information relevant for distinguishing seizure from non-seizure activity. If the latent representations would encapsulate such information, we would expect to observe systematic differences between the two groups in the latent space.

As a first exploratory step, we applied simple clustering analyses to probe potential separability. To assess whether the latent encodings exhibit global clustering behavior, we employed a classical multivariate framework based on the ratio of between-class to within-class scatter [7]. Specifically, for seizure and non-seizure encodings, we computed the within-class scatter matrices $\mathbf{S}_0$ and $\mathbf{S}_1$, as well as the between-class scatter matrix $\mathbf{S}_B$, defined as

$$\mathbf{S}_c = \sum_{i \in \mathcal{I}_c} \big(\mathbf{z}^{(i)} - \boldsymbol{\mu}_c\big)\big(\mathbf{z}^{(i)} - \boldsymbol{\mu}_c\big)^{\top}, \quad c \in \{0, 1\},$$

$$S_B = \sum_{c \in \{0,1\}} n_c \big(\boldsymbol{\mu}_c - \boldsymbol{\mu}\big)\big(\boldsymbol{\mu}_c - \boldsymbol{\mu}\big)^{\top},$$

where $\mathbf{z}^{(i)} \in \mathbb{R}^k$ denotes a latent encoding, $\boldsymbol{\mu}_c$ the class-specific mean in latent space, and $\boldsymbol{\mu}$ the global mean across all latent representations. $\mathcal{I}_c$ represents the set of indices belonging to class c, and $n_c = | \mathcal{I}_c |$ the number of latent encodings in that class. As a compact numerical summary of class separation, we used the trace ratio

$$R = \frac{\mathrm{tr}(\mathbf{S}_B)}{\mathrm{tr}(\mathbf{S}_W)}, \qquad \mathbf{S}_W = \mathbf{S}_0 + \mathbf{S}_1,$$

where $\mathrm{tr}(\cdot)$ denotes the matrix trace, i.e., the sum of diagonal entries. This ratio provides a scalar measure of how well the classes are separated, with larger values indicating clearer clustering in the latent space.

Additionally, we examined potential class separability by computing the distances between the arithmetic means of two randomly drawn subsets of latent encodings. The subsets were either both taken from seizure or background encodings, or one from each. For each case, we performed 10,000 random samplings, with the subset size set to half the number of seizure segments. Formally, given two subsets with arithmetic means $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_j$, their distance is given by:

$$d_{\mu_i \mu_j} = \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|_2.$$

Next, we visually inspected the distributions of seizure and non-seizure signals along each latent dimension to examine whether their marginal distributions differ.

For further visualization, we applied linear discriminant analysis (LDA) to project the latent encodings into a one-dimensional space. Similar to PCA, LDA is a linear dimensionality reduction technique [7]. However, unlike PCA, LDA is supervised and explicitly seeks projections that maximize class separability. In the binary case, LDA can produce only a single discriminant axis, since there is only one direction that maximally separates two classes.

As a final measure of separability, we trained simple classifiers on the latent encodings. We chose logistic regression and k-NN due to their simplicity and interpretability. For k-NN, we set the hyperparameter to $k = \sqrt{N}$, where N is the number of samples in the dataset. This heuristic is a well-established starting point for k-NN [15, 21]. To fit a classifier, we used 75% of the test data, while the remaining 25% served as an evaluation set. The split was stratified to preserve the proportion of seizure and background activity in both sets. Balanced class weights were applied to counter the overrepresentation of background activity, ensuring that misclassifications of seizures were penalized more strongly. The performance of the classifiers was evaluated using the area under the receiver operating characteristic curve (AUC), a widely used and robust performance measure for binary classification [14]. The receiver operator curve depicts the true positive rate against the false positive rate across different decision thresholds. The AUC summarizes this curve into a single value, where scores closer to 1 indicate better performance of the classifiers. To compare the separability of different latent representations, we applied DeLong's test to assess the statistical significance of differences in AUC scores obtained by the classifiers [13].

In the next section, we present our experimental results, starting with a comparison of the autoencoder frameworks and PCA in terms of their EEG reconstruction performance. We then analyze the latent representations learned by these models to assess their suitability as embeddings for distinguishing seizure from non-seizure activity.

Chapter 4

Results and Discussion

4.1 Reconstruction Performance

Evaluating how well the models are able to reconstruct EEG signals provides an initial indication of whether they can capture and learn the underlying temporal dynamics of the data.

Table 4.1 summarizes the reconstruction performance across all models and frequency bands, using 50 latent dimensions for all models. The DAE achieved excellent reconstruction across all frequency bands. On the held-out test set, the PCC between input and reconstruction exceeded 0.95 for every band. In contrast, the VAE showed weaker reconstruction performance for higher frequencies. Specifically, for the beta bands the PCC dropped to 0.903 for the low beta band and as low as 0.718 for the high beta band. This shows that the VAE struggles to accurately reconstruct higher frequencies, which can form more complex oscillatory patterns. When considering the PCC computed over the full reconstructions, all models achieved nearly perfect similarity between input and output, as exemplified in Figure 4.2. The DAE achieved higher similarity across all bands than the VAE. This difference could be attributed to the additional regularization of the latent space through the KL divergence for VAEs. Both approaches achieved a higher PCC than 0.97. PCA achieved the highest PCC on all bands with nearly perfect similarity between original and reconstruction. PCA also outperformed both the VAE and AE with respect to MSE, yielding a value of effectively zero across all frequency bands. In comparison, both autoencoder frameworks showed higher reconstruction errors, with the VAE consistently performing worst among the three methods. For the VAE, the delta band exhibited the highest MSE (2.130), despite its nearly perfect PCC (0.990). Notably, PCA and the deterministic autoencoder achieved even higher PCC values for the delta band (0.998 and 0.999, respectively) while yielding considerably lower MSE values (0.208 and 0.006, respectively). This discrepancy might be attributed to MSE's dependence on scale. As illustrated in Figure 4.1, the delta band exhibits the largest amplitudes, so even small deviations result in comparatively large squared errors. The DAE achieved lower reconstruction errors than the VAE in every band, but remained inferior to PCA. The elevated MSE values for the full signal reconstructions can be explained by the inevitably lost high-frequency variations that were removed during band-pass filtering in preprocessing.

The nearly perfect reconstruction performance of PCA may seem surprising, given its simplicity and its linear restriction to latent spaces. However, its effectiveness can be better understood by examining the loadings obtained from PCA. As shown in Fig. 4.3, PCA identifies loadings that resemble sinusoids of increasing frequency. Although this outcome is not an obvious property of PCA by construction, it is consistent with findings by Broomhead and King [9], who showed that applying PCA to delay-embedded time-series can extract the dominant oscillatory modes that govern the systems qualitative dynamics. PCA effectively recovers low-dimensional oscillatory structure, which explains its strong reconstruction performance on EEG signals.

Overall, all models showed strong reconstruction performance; however, both autoencoder frameworks were less accurate than PCA, with the DAE performing slightly better than the VAE. In the following, we examine whether the latent representations learned by the autoencoder frameworks — although less accurate in reconstruction than PCA — capture more meaningful structure that reflects the separability of seizure and non-seizure activity.

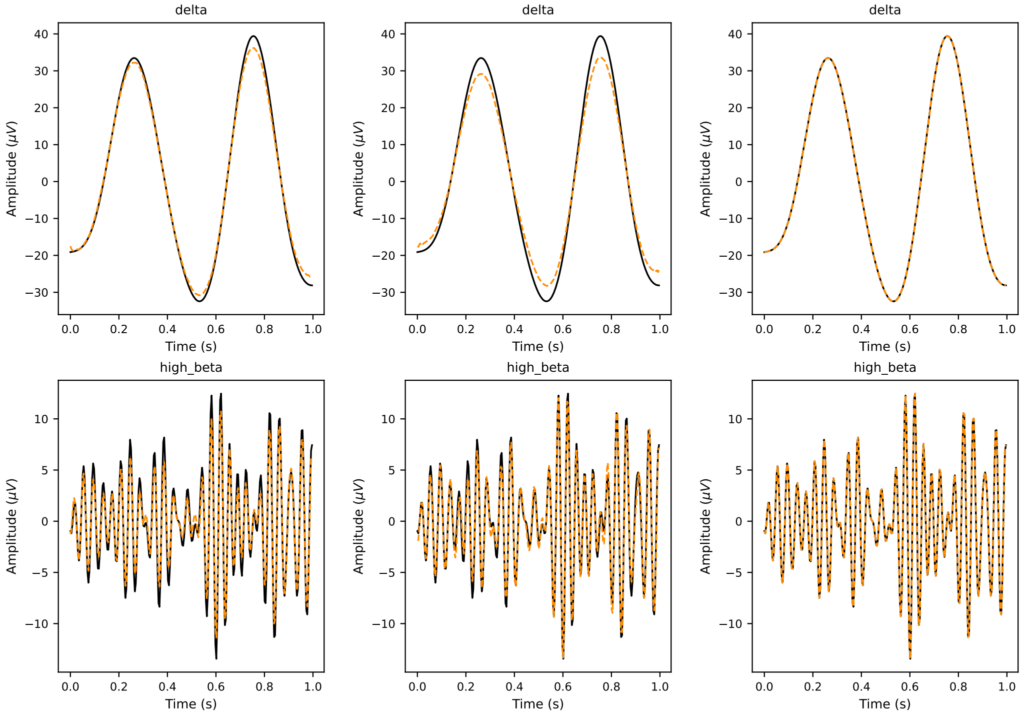
Band	VAE		AE		PCA	
	MSE	PCC	MSE	PCC	MSE	PCC
Delta	2.130	0.990	0.280	0.998	0.006	0.999
Theta	0.541	0.981	0.079	0.999	0.001	0.995
Alpha	0.332	0.965	0.099	0.996	0.001	0.984
Low Beta	0.814	0.903	0.646	0.966	0.003	0.986
High Beta	0.575	0.718	0.207	0.970	0.005	0.970
Full	7.870	0.973	6.613	0.989	4.966	0.993

Table 4.1: Test set performance of the variational autoencoder (VAE), deterministic autoencoder (DAE), and PCA. MSE denotes the Mean Squared Error and PCC the Pearson Correlation Coefficient. Values are rounded to three digits after the decimal point.

4.2 Latent Space Seizure Detection

4.2.1 Exploration of Class Separability

To gain initial insights into class separability in the latent space, we analyzed the variability and clustering behavior of seizure and background encodings produced by the different models.

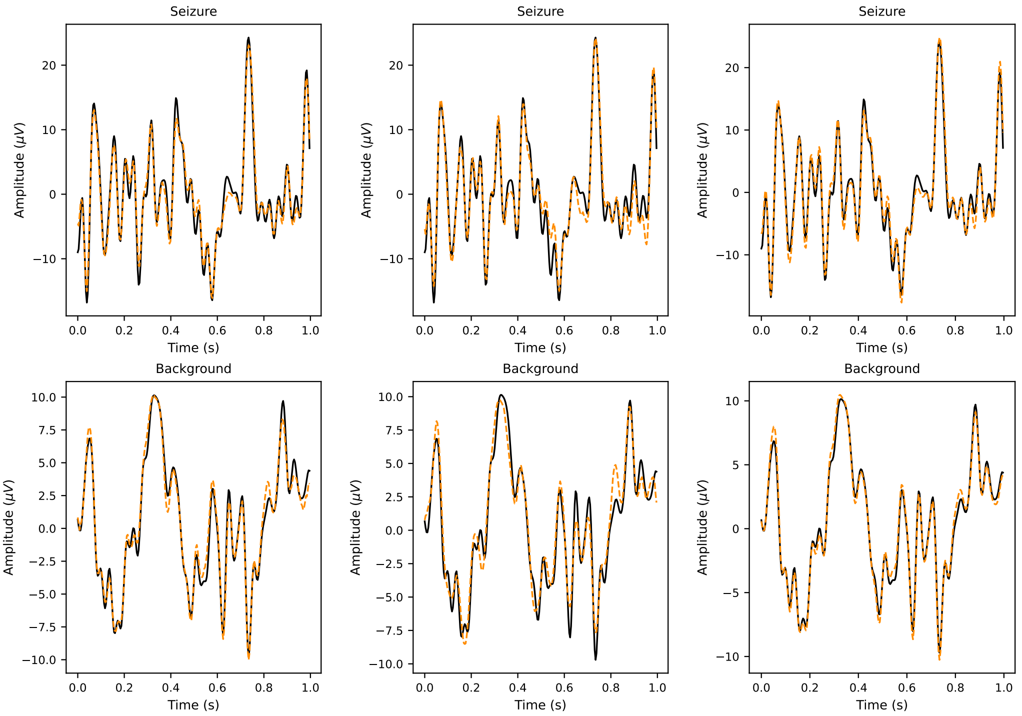


(a) (b) (c)

Figure 4.1: Exemplary original EEG signals and their reconstructions for the delta and high beta bands. Subfigure (a) shows the reconstruction obtained with the traditional autoencoder, (b) shows the reconstruction obtained with the variational autoencoder, and (c) shows the reconstruction obtained with PCA. In all plots, the original signal is depicted as a solid black line, while the reconstructed signal is shown as a dotted orange line.

From Table 4.2, it is evident that the ratios of between-class to within-class scatter are smaller than 1 for all methods and frequency bands. This indicates that the variability within each class is consistently larger than the difference between the class means, reflecting high overlap between seizure and background latent representations. For the VAE, the highest ratios occur in the theta and delta bands (4.648×10^{-3} and 2.464×10^{-3}, respectively). For the DAE, the ratios are generally lower, except in the high beta band where it reaches 5.426×10^{-3}. In contrast, PCA exhibits ratios closest to zero across all bands (ranging from 0.001×10^{-3} to 0.008×10^{-3}). This indicates that seizure and background representations largely overlap across all latent spaces, with no frequency band showing clear separation; however, the degree of overlap is several orders of magnitude smaller in the autoencoder frameworks than in PCA.

From Table 4.3, we observe that for both autoencoder frameworks the average distances between centroids of subsets drawn exclusively from seizure activity are an order of magnitude smaller than the distances between centroids of subsets drawn from different activities. This pattern is consistent across all frequency



(a) (b) (c)

Figure 4.2: Exemplary EEG signals and their reconstructions. Subfigure (a) shows results from the traditional autoencoder, (b) from the variational autoencoder, and (c) from PCA. The upper row depicts seizure activity and the lower row background activity. Original signals are shown as solid black lines, reconstructions as dotted orange lines.

Band	VAE	AE	PCA
Delta	2.464×10^{-3}	0.191×10^{-3}	0.005×10^{-3}
Theta	4.648×10^{-3}	0.182×10^{-3}	0.008×10^{-3}
Alpha	1.797×10^{-3}	1.256×10^{-3}	0.004×10^{-3}
Low Beta	2.514×10^{-3}	0.387×10^{-3}	0.001×10^{-3}
High Beta	1.444×10^{-3}	5.426×10^{-3}	0.002×10^{-3}

Table 4.2: Ratio of the traces of between-class to within-class scatter matrices for seizure and background signals. Scatter matrices were computed from the latent representations obtained with the variational autoencoder (VAE), the deterministic autoencoder (DAE), and PCA.

bands, and, for the VAE, also holds true for subsets drawn exclusively from background activity. For example, in the theta band, VAE within-seizure distance is 0.035, whereas the between-class distance is 0.870. Similarly, for the AE in

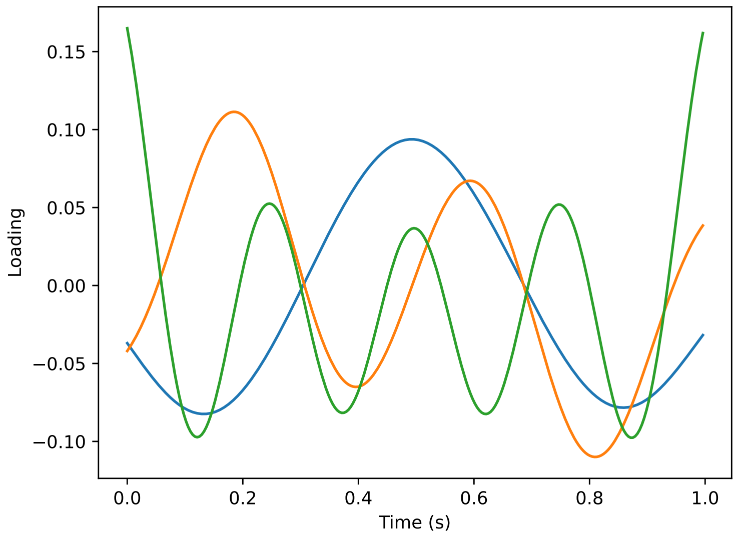


Figure 4.3: Exemplary PCA loadings of the delta band

the delta band, within-seizure distance of 0.098 contrasts with a between-class distance of 0.502. This indicates that seizure representations are more tightly clustered among themselves than in relation to background representations. For PCA encodings, however, no such pattern is evident. The average distances within seizure subsets are of the same order of magnitude as, or even close to, the distances between seizure and background subsets. For instance, in the delta band the seizure distance is 1.659, compared to a between-class distance of 2.138, suggesting that seizure and background representations do not form distinct clusters to the same extent as in the latent spaces learned by the autoencoder frameworks.

Overall, these results suggest that both autoencoder frameworks are more effective than PCA at capturing seizure-related structure in latent space.

4.2.2 Visualization of Latent Dimensions and Linear Discriminant Analysis

To further investigate the structure of the latent spaces and assess whether individual dimensions or linear combinations thereof carry discriminative information, we visualize histograms of single latent dimensions and apply LDA.

Figure 4.4 shows exemplary distributions of latent dimensions. For the VAE, the latent variables follow smooth, Gaussian-like distributions, which is expected since the KL-divergence term in the loss explicitly encourages this behavior (Fig. 4.4b). Interestingly, the DAE also produces relatively smooth distributions (Fig. 4.4a), whereas PCA yields distributions with sharp peaks concentrated around the mean (Fig. 4.4c). This illustrates a limitation of PCA: while it can reconstruct data well, it does not produce smooth latent dimensions that

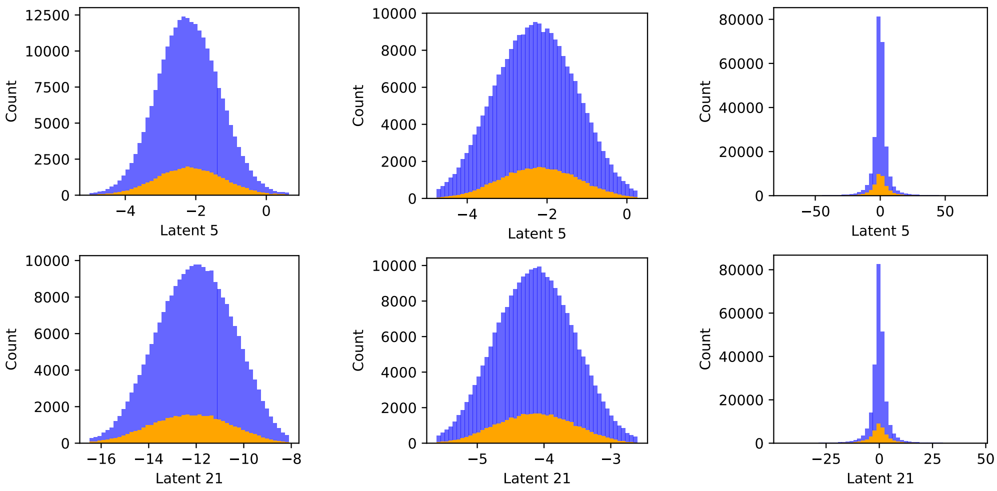
Band	VAE			AE			PCA		
	Seiz	Bkgd	Both	Seiz	Bkgd	Both	Seiz	Bkgd	Both
Delta	0.045	0.054	0.851	0.098	0.113	0.502	1.659	1.821	2.138
Theta	0.035	0.040	0.870	0.089	0.095	0.418	0.949	0.972	1.306
Alpha	0.020	0.039	0.534	0.011	0.051	0.601	0.667	0.818	0.913
Low Beta	0.046	0.062	0.968	0.091	0.110	0.715	0.584	1.301	1.090
High Beta	0.043	0.059	0.683	0.065	0.089	2.033	0.628	0.863	0.836

Table 4.3: Average distances between subset means of latent encodings obtained with the variational autoencoder (VAE), the deterministic autoencoder (DAE), and PCA. *Seiz* denotes subsets drawn exclusively from seizure activity, *Bkgd* from background activity, and *Both* from one seizure and one background subset. Values are rounded to three digits after the decimal point.

would support sampling of new, plausible EEG segments. Across all three frameworks, however, visual inspection of the latent-dimension histograms revealed no apparent separation between seizure and non-seizure activity. This suggests that individual latent dimensions alone do not capture the relevant differences between the two conditions.

To assess whether a linear combination of latent dimensions can better separate seizure and background activity, we next applied LDA and visualized the distributions along the discriminant axis. For both autoencoder frameworks, the distributions of seizure and background encodings along the discriminant axis show a clear but imperfect separation (Fig. 4.5a,b). In most frequency bands, seizure and background histograms form partially overlapping but distinct peaks, indicating that a single linear projection already captures meaningful differences between the two classes. In contrast, for PCA representations the histograms of seizure and background activity nearly coincide across all frequency bands, often collapsing to a narrow peak centered around zero (Fig. 4.5c). This reflects the lack of discriminative structure in the PCA latent space, as linear combinations of its dimensions fail to separate the two activities.

In summary, these visualizations highlight that the latent spaces learned by both autoencoder frameworks contain more class-relevant information than those derived from PCA, when reduced linearly to a single discriminant dimension. Having observed indications of linear separability in the latent spaces, we next examine this more systematically by applying the linear classifier logistic regression and comparing its performance to the non-linear classifier k-nearest neighbors.



(a) (b) (c)

Figure 4.4: Exemplary distributions of latent dimensions for seizure (orange) and non-seizure (blue) activity. Subfigure (a) shows results from the deterministic autoencoder, (b) from the variational autoencoder, and (c) from principle component analysis. The top row correponds to the delta band, the bottom row to the low beta band.

4.2.3 Fitting Classifiers

Table 4.4 reports the AUC values obtained by applying logistic regression to the latent representations. Both autoencoder frameworks achieved clearly higher scores than PCA, for which logistic regression performed at a level comparable to random classification. The differences in predicted probabilities between logistic regression applied to VAE encodings and PCA, as well as between logistic regression applied to deterministic autoencoder encodings and PCA, were highly significant according to DeLongs test. By contrast, the comparison between VAE and deterministic autoencoder encodings showed no statistically significant difference. Although the AUC values do not reach state-of-the-art levels for seizure detection, these results provide further evidence that the latent spaces learned by autoencoder frameworks capture seizure-related information.

Table 4.5 reports the AUC scores obtained by applying k-NN to the latent encodings. No statistically significant differences were observed between the frameworks, and applying k-NN directly to the original band-pass filtered EEG signals yielded similar AUC scores. The performance of k-NN was comparable to that of logistic regression applied to the autoencoder encodings. This suggests that k-NN, by design, can exploit the local similarity structure of the data, performing reasonably well on both raw signals and latent encodings, regardless of whether they are derived from PCA or autoencoders. However, the autoencoder encodings provide an additional advantage: they enable a significantly better linear separation of seizure and non-seizure activity than PCA.

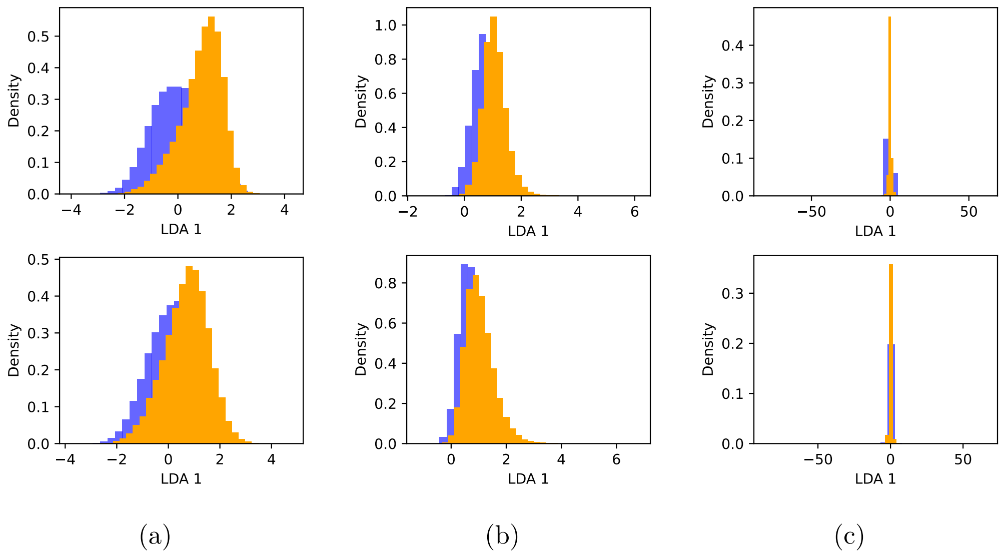


Figure 4.5: Linear discriminant analysis applied to latent encodings of the variational autoencoder (a), the deterministic autoencoder (b) and of principle component analysis (c). Top row is for theta band and bottom row is for delta band.

This highlights that autoencoders transform the data into a representation where linear classifiers can become effective.

Band	VAE (AUC)	AE (AUC)	PCA (AUC)
Delta	0.644***	0.660***	0.500
Theta	0.748***	0.745***	0.502
Alpha	0.726***	0.727***	0.499
Low Beta	0.714***	0.711***	0.506
High Beta	0.662***	0.670***	0.500

Table 4.4: Area under the receiver operator curve (AUC) scores for seizure vs. non-seizure classification using a logistic regression classifier trained on latent representations, obtained from the variational autoencoder (VAE), deterministic autoencoder (DAE), and PCA. Values are rounded to three digits after the decimal point. *** indicates $p < 0.001$ in DeLong's test comparing the AUCs of the respective autoencoder with PCA.

Band	VAE (AUC)	AE (AUC)	PCA (AUC)
Delta	0.722	0.722	0.710
Theta	0.743	0.744	0.747
Alpha	0.727	0.729	0.740
Low Beta	0.716	0.715	0.718
High Beta	0.674	0.676	0.674

Table 4.5: Area under the receiver operator curve (AUC) scores for seizure vs. non-seizure classification using a k-nearest-neighbor classifier trained on latent representations, obtained from the variational autoencoder (VAE), vanilla autoencoder (AE), and PCA. Values are rounded to three digits after the decimal point.

Chapter 5

Conclusion and Future Work

This thesis investigated whether unsupervised autoencoder-based methods can accurately compress and reconstruct EEG signals, and whether the obtained latent representations support the separation of seizure and non-seizure activity. We trained and compared two autoencoder frameworks to reconstruct EEG signals and analyzed their latent representations with respect to seizure activity. The experimental results revealed that both autoencoder frameworks had excellent reconstruction capacity, and that their latent spaces captured seizure-related structure more effectively in comparison to PCA. Statistical measures, visualization, and classification outcomes all indicated that the autoencoder encodings exhibited clearer separation between seizure and non-seizure activity. Notably, logistic regression achieved significantly higher classification performance on the autoencoder encodings. This suggests that the latent spaces learned by reconstruction-based neural networks uncover seizure-related structure in a way that can be effectively exploited by linear classifiers. Reconstruction performance and seizure separability in latent space did not differ substantially between the VAE and the vanilla autoencoder. Consequently, our work revealed that the probabilistic regularization of the VAE did not result in measurable advantages over its deterministic counterpart.

Despite these promising findings, several limitations of the present work must be acknowledged. First, the autoencoders were trained on single-channel, band-limited inputs, which neglect cross-channel spatial information and omit higher-frequency components such as the gamma band. Second, the architectural choices and the training procedure were fixed, without systematic hyperparameter search or exploration of alternative network designs. Finally, reconstruction and latent-space evaluation were both applied to the same dataset, which may limit generalizability.

These observations point to several promising directions for future research. A natural next step is to explore alternative architectural choices, including multi-channel convolutional autoencoders, recurrent models, or transformer-based architectures that may capture richer temporal and spatial structure. Moreover, future work should also investigate the impact of preprocessing, including alternative frequency handling or end-to-end approaches that bypass band-pass filtering altogether. An interesting direction for future work is to train the autoencoder frameworks solely on background activity and subsequently apply them to recordings that contain seizures, with the aim of identifying seizure segments as outliers in latent space. Furthermore, using heterogeneous datasets and conducting systematic hyperparameter optimization could improve model performance and enhancing the generalizability of future approaches.

In summary, this work shows that autoencoders can faithfully reconstruct EEG signals and, in doing so, learn unsupervised latent representations that encode seizure-related structure more meaningfully than classical linear approaches. This provides a foundation for using autoencoders as reliable tools for EEG compression and reconstruction, while also offering a first indication that such models may support the discovery of meaningful biomarkers for seizure activity.

Bibliography

[1] Abdullah Z. Al-Marridi, Amr Mohamed, and Aiman Erbad. 2018. Convolutional Autoencoder Approach for EEG Compression and Reconstruction in m-Health Systems. In *Proceedings of the 2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, Limassol, Cyprus, pp. 370–375. DOI: 10.1109/IWCMC.2018.8450511.

[2] Asanagi. 2010. Electrode locations of International 10–20 system for EEG recording. https://commons.wikimedia.org/wiki/File:21_electrodes_of_International_10-20_system_for_EEG.svg. Public domain image, Wikimedia Commons. (2010).

[3] Gerold Baier, Neil Cain, Christian Elger, and Klaus Lehnertz. 2012. The importance of modeling epileptic seizure dynamics as spatiotemporal evolving networks. *Frontiers in Neurology*, 3, 73. DOI: 10.3389/fneur.2012.00073. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3429055/.

[4] Dor Bank, Noam Koenigstein, and Raja Giryes. 2020. Autoencoders. *arXiv preprint arXiv:2003.05991*. https://arxiv.org/abs/2003.05991.

[5] Anthony J. Bell and Terrence J. Sejnowski. 1995. An InformationMaximization Approach to Blind Separation and Blind Deconvolution. *Neural Computation*, 7, 6, 1129–1159. DOI: 10.1162/neco.1995.7.6.1129. https://pubmed.ncbi.nlm.nih.gov/7584893/.

[6] Sándor Beniczky, Eugen Trinka, Elaine Wirrell, Fatema Abdulla, Raidah Al Baradie, Mario Alonso Vanegas, Stéphane Auvin, Mamta Bhushan Singh, Hal Blumenfeld, Alicia Bogacz Fressola, Roberto Caraballo, Mar Carreno, Fernando Cendes, Augustina Charway, Mark Cook, Dana Craiu, Birinus Ezeala-Adikaibe, Birgit Frauscher, Jacqueline French, M. V. Gule, Norimichi Higurashi, Akio Ikeda, Floor E. Jansen, Barbara Jobst, Philippe Kahane, Nirmeen Kishk, Ching S. Khoo, Kollencheri Vinayan Puthenveettil, Lieven Lagae, Kheng-Seang Lim, Angelica Lizcano, Aileen McGonigal, Katerina Tanya Pérez-Gosiengfiao, Philippe Ryvlin, Nicola Specchio, Michael R. Sperling, Hermann Stefan, William Tatum, Manjari Tripathi, Elza Márcia Yacubian, Samuel Wiebe, Jo Wilmshurst, Dong Zhou, and J. Helen Cross. 2025. Updated classification of epileptic seizures: Position paper of the International League Against Epilepsy. *Epilepsia*, 66, 6, 1804–1823. DOI: 10.1111/epi.18338. https://onlinelibrary.wiley.com/doi/10.1111/epi.18338.

[7] Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning*. Springer, New York, NY, USA. ISBN: 978-0-387-31073-2.

[8] Bernd Bromm and Edgar Scharein. 1982. Principal component analysis of pain-related cerebral potentials to mechanical and electrical stimulation in man. *Electroencephalography and Clinical Neurophysiology*, 53, 94–103. DOI: 10.1016/0013-4694(82)90105-3. https://pubmed.ncbi.nlm.nih.gov/6176025/.

[9] D. S. Broomhead and G. P. King. 1986. Extracting Qualitative Dynamics from Experimental Data. *Physica D: Nonlinear Phenomena*, 20, 2–3, 217–236. DOI: 10.1016/0167-2789(86)90031-X.

[10] Steven L. Brunton and J. Nathan Kutz. 2019. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control.* (1st ed.). Cambridge University Press, Cambridge. ISBN: 978-1108422093.

[11] A. Chaddad, D. Ben Ayed, H. El Amri, L. Mahmoudi, S. Chbili, and M. Hamdi. 2023. Electroencephalography Signal Processing. *Sensors*, 23, 17, 7667. DOI: 10.3390/s23177667. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10385593/.

[12] Wenna Chen, Yixing Wang, Yuhao Ren, Hongwei Jiang, Ganqin Du, Jincan Zhang, and Jinghua Li. 2023. An automated detection of epileptic seizures EEG using CNN classifier based on feature fusion with high accuracy. *BMC Medical Informatics and Decision Making*, 23, 96. DOI: 10.1186/s12911-023-02180-w. https://doi.org/10.1186/s12911-023-02180-w.

[13] Elizabeth R. DeLong, David M. DeLong, and Daniel L. Clarke-Pearson. 1988. Comparing the areas under two or more correlated receiver operating characteristic curves: A nonparametric approach. *Biometrics*, 44, 3, 837–845. DOI: 10.2307/2531595. https://pubmed.ncbi.nlm.nih.gov/3203132/.

[14] Tom Fawcett. 2006. An introduction to ROC analysis. *Pattern Recognition Letters*, 27, 8, 861–874. DOI: 10.1016/j.patrec.2005.10.010.

[15] Ahmad Basheer Hassanat, Mohammad Ali Abbadi, Ghada Awad Altarawneh, and Ahmad Ali Alhasanat. 2014. Solving the Problem of the k Parameter in the k-Nearest Neighbor Classifier Using an Ensemble Learning Approach. *arXiv preprint arXiv:1409.0919.* DOI: 10.48550/arXiv.1409.0919. https://arxiv.org/abs/1409.0919.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. DOI: 10.1109/CVPR.2016.90. https://doi.org/10.1109/CVPR.2016.90.

[17] Sepp Hochreiter. 1998. The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6, 2, 107–116. DOI: 10.1142/S0218488598000094. https://www.worldscientific.com/doi/10.1142/S0218488598000094.

[18] Harold Hotelling. 1933. Analysis of a Complex of Statistical Variables into Principal Components. *Journal of Educational Psychology*, 24, 417–441 and 498–520.

[19] Hideaki Iiduka. 2020. Appropriate Learning Rates of Adaptive Learning Rate Optimization Algorithms for Training Deep Neural Networks. *arXiv preprint arXiv:2002.09647*. https://arxiv.org/abs/2002.09647.

[20] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pp. 448–456. http://proceedings.mlr.press/v37/ioffe15.html.

[21] M. Jirina and M. Jirina Jr. 2011. Classifiers Based on Inverted Distances. In *New Fundamental Technologies in Data Mining*. Volume 1. K. Funatsu, (Ed.) InTech. Chapter 19, pp. 369–387.

[22] Yasin Kaya and Ercan Gürsoy. 2023. A novel multihead CNN design to identify plant diseases using the fusion of RGB images. *Ecological Informatics*, 75, 101998. DOI: 10.1016/j.ecoinf.2023.101998. https://www.sciencedirect.com/science/article/abs/pii/S1574954123000274.

[23] Diederik P. Kingma and Max Welling. 2013. Auto-Encoding Variational Bayes. *arXiv preprint arXiv:1312.6114*. https://arxiv.org/abs/1312.6114.

[24] S. López, A. Gross, S. Yang, M. Golmohammadi, I. Obeid, and J. Picone. 2016. An Analysis of Two Common Reference Points for EEGs. In *IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*. DOI: 10.1109/SPMB.2016.7846854. https://pmc.ncbi.nlm.nih.gov/articles/PMC5479869/.

[25] S. López, G. Suárez, D. Jungreis, I. Obeid, and J. Picone. 2015. Automated Identification of Abnormal Adult EEGs. In *IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*. PMID:27195311. IEEE. DOI: 10.1109/SPMB.2015.7405423.

[26] MathWorks. 2024. Autoencoders. Accessed: 2025-10-19. https://www.mathworks.com/discovery/autoencoder.html.

[27] Ibomoiye D. Mienye and Theo G. Swart. 2024. A Comprehensive Review of Deep Learning: Architectures, Recent Advances, and Applications. *Information*, 15, 12, 755. DOI: 10.3390/info15120755.

[28] Matiur Rahman Minar and Jibon Naher. 2018. Recent Advances in Deep Learning: An Overview. *arXiv preprint arXiv:1807.08169*. https://arxiv.org/abs/1807.08169.

[29] Ernst Niedermeyer and Fernando H. Lopes da Silva. 2004. *Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*. (5th ed.). Lippincott Williams & Wilkins, New York.

[30] Soheyl Noachtar and Jan Rémi. 2009. The role of EEG in epilepsy: A critical review. *Epilepsy & Behavior*, 15, 1, 22–33. DOI: 10.1016/j.yebeh.2009.02.035.

[31] Srividya Pattisapu and Supratim Ray. 2023. Stimulus-induced narrow-band gamma oscillations in humans can be recorded using open-hardware low-cost EEG amplifier. *PLoS ONE*, 18, 1, e0279881. DOI: 10.1371/journal.pone.0279881. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0279881.

[32] Karl Pearson. 1901. On Lines and Planes of Closest Fit to Systems of Points in Space. *Philosophical Magazine*, 2, 11, 559–572.

[33] Simon J. D. Prince. 2023. *Understanding Deep Learning*. MIT Press, Cambridge, MA. https://udlbook.github.io/udlbook/.

[34] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. Learning Internal Representations by Error Propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1*. David E. Rumelhart, James L. McClelland, and the PDP Research Group, (Eds.) MIT Press, Cambridge, MA, USA, pp. 318–362.

[35] Sani Saminu, Guizhi Xu, Zhang Shuai, Isselmou Abd El Kader, Adamu Halilu Jabire, Yusuf Kola Ahmed, Ibrahim Abdullahi Karaye, and Isah Salim Ahmad. 2021. A Recent Investigation on Detection and Classification of Epileptic Seizure Techniques Using EEG Signal. *Brain Sciences*, 11, 5, 668. DOI: 10.3390/brainsci11050668. https://www.mdpi.com/2076-3425/11/5/668.

[36] Vinay Jay Shah, Iyad Obeid, Joseph Picone, Satya Krishna Yadav, et al. 2018. The Temple University Hospital Seizure Detection Corpus. *Frontiers in Neuroinformatics*, 12, 83. DOI: 10.3389/fninf.2018.00083.

[37] Mingxing Tan and Quoc V. Le. 2019. MixConv: Mixed Depthwise Convolutional Kernels. *arXiv preprint arXiv:1907.09595*. https://arxiv.org/abs/1907.09595.

[38] Xiashuang Wang, Yuhang Wang, Dong Liu, Yicheng Wang, and Zhiyong Wang. 2023. Automated recognition of epilepsy from EEG signals using a combining spacetime algorithm of CNN-LSTM. *Scientific Reports*, 13, 1, 14876. DOI: 10.1038/s41598-023-41537-z. https://doi.org/10.1038/s41598-023-41537-z.

[39] Jiahong Zhang, Meijun Qu, Ye Wang, and Lihong Cao. 2022. A Multi-Head Convolutional Neural Network With Multi-path Attention Improves Image Denoising. *arXiv preprint arXiv:2204.12736*. https://arxiv.org/abs/2204.12736.

[40] Tong Zhao, Yi Cui, Taoyun Ji, Jiejian Luo, Wenling Li, Jun Jiang, Zaifen Gao, Wenguang Hu, Yuxiang Yan, Yuwu Jiang, and Bo Hong. 2024. Variational autoencoder for extracting EEG representation. *NeuroImage*, 304, 120946. DOI: 10.1016/j.neuroimage.2024.120946.

[41] Xia Zhao, Limin Wang, Yufei Zhang, Xuming Han, Muhammet Deveci, and Milan Parmar. 2024. A review of convolutional neural networks in computer vision. *Artificial Intelligence Review*, 57, 1–43. DOI: 10.1007/s10462-024-10721-6.