Adaptive Chirplet Transform and Deep Learning Algorithms for EEG based Sleep Stage Detection

by

João Pedro Bicalho Andrade

Supervisor: Steve Mann

April 7th, 2025

B.A.Sc. Thesis

Division of Engineering Science
UNIVERSITY OF TORONTO

Acknowledgements

I would like to express my sincere gratitude to all those who supported and contributed to this thesis.

I am especially grateful to my supervisor, Prof. Steve Mann. Your expertise and mentorship played a big role in shaping the direction and depth of this research.

A heartfelt thank-you goes to Nishant Kumar, whose support and technical knowledge were instrumental to this work.

Finally, I extend my appreciation to my friends and family for their continuous support.

Contents

List of Figures

1 Abstract

Electroencephalography (EEG) is a powerful tool for analyzing brain activity, particularly in the context of sleep stage detection. However, traditional signal processing methods like the Fourier and Wavelet Transforms often struggle to capture the non-stationary and time-varying nature of EEG signals. This thesis explores the Adaptive Chirplet Transform (ACT) as a novel feature extraction technique for EEG-based sleep stage classification using deep learning. The ACT offers a dynamic time-frequency representation by matching signal segments to Gaussian chirplets. A GPU-accelerated implementation of the ACT was developed and applied to the Bitbrain Open Access Sleep Dataset, processing over 10,000 epochs from three nights of sleep. Feature vectors extracted by the ACT were reshaped into 5×5 matrices and used as input to a hybrid Convolutional Neural Network (CNN) and Gated Recurrent Unit (GRU) model. The model achieved a training accuracy of 91.5% and a test accuracy of 57.9%, demonstrating the ACT's potential to encode relevant features with a 153.6:1 compression ratio. The findings suggest that, with further optimization and scaling, the ACT could become the next state of the art signal processing and feature extraction method for EEG.

2 Introduction

Building on the diverse methodologies for EEG analysis, this thesis specifically focuses on the application of the Adaptive Chirplet Transform combined with deep learning to detect sleep stage. Sleep stage detection is a critical area of research due to its significance in diagnosing sleep disorders, understanding brain function during rest, and developing interventions to improve sleep quality. Deep learning offers a transformative approach by automating the extraction of meaningful features from complex, high-dimensional EEG data. Unlike traditional manual scoring methods or simpler classification techniques, deep learning models do well at capturing sophisticated temporal and spatial patterns within EEG signals. By using data from publicly available datasets, this study aligns with broader scientific practices for benchmarking and comparability while avoiding the logistical and ethical complexities of collecting new EEG data.

Techniques such as the Fourier Transform and Wavelet Transform have traditionally dominated time-frequency analysis in EEG research. The Fourier Transform decomposes signals into their constituent frequencies, providing valuable insights into rhythmic brain activity. Wavelet Transforms extend this by combining frequency and temporal information, enabling the analysis of dynamic changes over time. However, these methods face limitations when capturing complex, non-stationary signal features, such as rapid frequency modulations or localized transient events. This paper adopts the Adaptive Chirplet Transform (ACT) as a promising alternative. [1] [2,3] Unlike the fixed basis functions of Fourier and Wavelet Transforms, the ACT dynamically adapts to the signal's local characteristics, offering a more flexible and precise representation of time-varying EEG patterns. These qualities make the ACT particularly suitable for sleep stage detection, where subtle physiological changes manifest as transient shifts in frequency content.

Despite its potential, the application of the ACT to EEG data remains under-explored, with only a few studies demonstrating its capabilities. [2,4–12] This gap presents an opportunity to advance EEG research by integrating the ACT with state-of-the-art deep learning architectures. The ACT's ability to provide a detailed, dynamic representation of EEG signals enhances the input to deep learning models, potentially improving their classification accuracy and robustness. Sleep stage transitions, which involve nuanced changes in frequency and amplitude, are especially well-suited for analysis using the ACT.

The primary objective of this thesis is to address this research gap by employing the ACT as the feature extraction algorithm for the EEG data and feed it to a Deep Learning architecture for the application of sleep stage detection.

3 Background and Literature Review

3.1 Introduction to EEG Analysis

Electroencephalography (EEG) is a widely utilized technique that measures the electrical activity generated by the brain, allowing for brain wave visualization. Specifically, it detects minute differences in electric potential at the scalp, which result from the collective activity of post-synaptic potentials produced by neurons within the cortical layers.

EEG data is used extensively to analyze brain activity, but EEG signals are highly complex due to their non-stationary nature, low signal to noise ratio, and the presence of artifacts from non-neural sources such as muscular activation. Even after processing, the data is still extremely intricate and convoluted, making it difficult to interpret. To extract meaningful information from EEG signals, the data needs to go through rigorous processing and analysis.

3.2 Traditional Methods in EEG Analysis

The analysis of EEG signals spans diverse methodologies, each suited to different applications and offering unique insights into brain activity.

3.2.1 Event-Related Potentials (ERPs)

Event-Related Potentials (ERPs) represent one of the most established approaches, involving the identification and analysis of brain responses that are time-locked to specific events, such as sensory stimuli or cognitive tasks. Examples include the P300, a positive deflection occurring roughly 300 ms after an expected stimulus, often used in BCIs and studies on attention and decision-making, and Steady-State Visual Evoked Potentials (SSVEPs), which leverage repetitive visual stimulation for applications like assistive technology and gaming. ERPs are widely used in cognitive neuroscience due to their ability to isolate specific neural processes. However, they require significant manual preprocessing and are often limited by their reliance on predefined stimuli and rigid temporal windows, making them less adaptable to complex, real-world scenarios.

3.2.2 Spectral Analysis

Spectral analysis, another cornerstone of EEG research, focuses on the frequency content of brain signals to reveal insights into oscillatory activity across delta, theta, alpha, beta, and gamma bands. Each frequency band correlates with distinct cognitive and physiological states, such as delta waves linked to deep sleep or alpha waves associated with relaxation and attention. By employing time-frequency techniques like Wavelet Transforms or Chirplet Transforms, spectral analysis bridges the gap between traditional ERPs and modern deep learning by enabling the study of dynamic changes in oscillatory activity. It finds extensive use in both clinical and research settings, such as monitoring epilepsy, understanding attention dynamics, and tracking meditation progress. [13]

3.2.3 EEG Data Pre-processing, Filtering and Artifact Removal

Noise and artifacts from non-neural sources can be broken down into physiological noise and environmental noise. Physiological sources contributing the most noise come from eyeball movement, which is known as electrocorticogram (EOG), the cardiac signal, known as electrocardiogram (ECG), and muscular contractions, known as electromyography (EMG). Environmental noise can encompass electromagnetic fields in the vicinity caused by any other non biological sources, such as those caused from AC power lines or electronic devices in the room.

Signal processing methods are employed to address these challenges, enhancing the interpretability of EEG data. Signal Processing is a broad field and active area of research, and in the context of EEG signals it encompasses several different tasks, most notably pre-processing, feature extraction and analysis and interpretation of data. Pre-processing includes data filtering for noise reduction and artifact removal, segmenting the data into appropriate epochs and normalizing the data.

Filters such as low-pass, high-pass, band-pass, and notch filters are commonly used to retain relevant brainwave frequencies while suppressing noise from sources like eye movements or power line interference. By focusing on frequency-specific manipulations, data filtering provides a clean, noise-reduced signal, forming the basis for subsequent, more sophisticated signal processing techniques.

Preprocessing EEG data is a critical step for the success of deep learning models. Filtering techniques include bandpass filtering to eliminate frequencies outside the range of interest (e.g., 0.5–50 Hz for most EEG tasks) and notch filtering to remove powerline interference such as 50 or 60 Hz noise. Artifact removal methods are employed to mitigate non-brain artifacts, utilizing techniques such as independent component analysis (ICA) to separate mixed signals and isolate artifacts, regression-based methods to subtract artifacts like eye blinks using reference signals, and consensus filtering, which applies multiple filtering methods to ensure signal integrity. Signal standardization, including normalization or z-scoring, ensures consistency and improves convergence during training. [13]

3.3 Adaptive Chirplet Transform (ACT)

The ACT was originally developed by S. Mann and S. Haykin in 1991 and since then several papers have proposed modifications or possible improvements to it. [3] Some notable examples include the "Adaptive Linear Chirplet Transform" by Guan et Al. the "Multi-Scynchrosqueezing Chirplet Transform" by Zhu et Al. [14] the "Enhanced Adaptive Linear Chirplet Transform" (EALCT) by Lopez et Al. [7] and lastly, developed by Chui et Al., the Chirplet Transform-based signal separation scheme (CT3S). [15]

Given its flexibility and capability to capture information on signals that vary both in time and frequency, the ACT has also been used across a wide range of applications. From aircraft bearings fault diagnosis, to P300, VEPs and epileptic seizure detection with EEG, target recognition from RADAR technology, and more. [2, 9, 16–18]

After a thorough analysis of several different versions and models of the ACT it was clear to see that the paper which aligned the most with the goals set out by this thesis was the work of Bhargava et Al. Besides using the ACT to analyze P300 signals from EEG data, Bhargava also published an open source Python implementation of the ACT optimized with the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm.

The implementation works by creating an ACT class that is able to perform the Adaptive Chirplet Transform on any given signal. The class initializes by either generating a dictionary of Gaussian chirplet functions or loading one from the cache if one has already been generated. The size of the dictionary is variable, and it depends on some of the parameters passed in by the user when creating an instance of the class.

The equation for a general Gaussian Chirplet is shown below:

$$g(t) = \frac{1}{\sqrt[4]{\pi}\sqrt{\Delta_t}} \exp\left\{-\frac{1}{2}\left(\frac{t-t_c}{\Delta_t}\right)^2 + j \cdot 2\pi\left[c(t-t_c) + f_c(t-t_c)\right]\right\} \tag{1}$$

The Gaussian chirplet above has a center time t_c, duration Δ_t, frequency modulation (chirp) rate c, and center frequency f_c. These four parameters are what is needed to fully characterize a Gaussian chirplet function. The dictionary then is constructed by generating different Gaussian chirplets varying these parameters over a range and step size defined by the user when instantiating the class.

Once the dictionary has been generated, the ACT compares the given signal to the Gaussian Chirplets in the dictionary, and it picks the one which most closely matches the signal, using the BFGS optimization algorithm to do so. This step is called a first order Chirplet Transform, and this step can be repeated as many times as desired, achieving a higher order ACT. More concisely, this method represents a signal as a linear combination of *chirplets*, which are Gaussian-windowed, frequency-modulated basis functions.

For an N order ACT, this implementation returns several lists. First a list of coefficients, where the n^{th} element of the lists represents the coefficient that scales the best match Gaussian Chirplet chosen in the n^{th} iteration of the algorithm. Followed by this it returned a list of size $4 * n$, where the elements are the four parameters necessary to identify each of the chosen Gaussian Chirplets from the dictionary. For the purposes of applying the ACT, these two lists are the most important, but it also provids a residue list and a raw error list, both of which could be used as a measure of how accurately the signal was reconstructed by the chirplet transform at different orders.

3.4 EEG and Deep Learning

Deep learning has emerged as a powerful tool for EEG analysis because it can automatically extract meaningful patterns and features from raw data without requiring extensive manual processing or domain-specific feature engineering. It has enabled breakthroughs in tasks that were previously difficult or impossibly unpractical to achieve using conventional methods. Furthermore, recent advances in deep learning architectures and computational resources keep accelerating progress in this field.

Deep learning has been widely applied to various EEG classification tasks, which can be broadly categorized into domains such as sleep stage classification, mental state and emotion recognition, disease detection and diagnosis, and brain-computer interfaces (BCIs).

Sleep stage classification involves identifying distinct sleep stages, such as NREM and REM, based on EEG patterns, facilitating research in sleep medicine and sleep disorders.

Mental state and emotion recognition tasks focus on classifying cognitive states (e.g., focused vs. relaxed) and emotions (e.g., happiness, stress), often utilized in human-computer interaction applications.

Disease detection and diagnosis tasks aim to identify neurological disorders like epilepsy, Alzheimer's disease, and Parkinson's disease, with seizure detection from EEG signals being a prominent application.

Lastly, BCIs enable control of external devices using EEG signals, involving tasks such as motor imagery classification and control signal generation.

These applications showcase the versatility and transformative potential of deep learning in EEG analysis.

3.4.1 Input to Neural Networks

The choice of input data for neural networks is a critical factor in determining the performance of EEG analysis models. Different research approaches vary in their use of raw EEG signals, preprocessed data, and transformed representations to optimize neural network training.

While some studies utilize raw EEG signals as input relying simply on the feature extraction power of the Neural Network, the vast majority of research emphasizes preprocessing and filtering as essential steps. Techniques such as bandpass filtering, notch filtering to remove powerline interference, and independent component analysis (ICA) for ocular and muscular artifact removal are widely employed. These methods enhance signal quality and ensure cleaner inputs for neural networks.

However, beyond initial preprocessing and filtering, there is little consensus among researchers on the extent of artifact removal or the specific techniques to employ. For instance, some studies apply rigorous artifact removal to eliminate non-neuronal signals entirely, while others retain certain artifacts to preserve additional features of the raw signal.

A recent review of 89 studies on deep learning for EEG classification highlights this lack of consensus and variability in pre-processing strategies. [19] The accompanying visual graph from the review illustrates the proportion of studies adopting different artifact removal methods, noting that the majority of studies either did not specify or did not use removal methods.

Transformations of EEG data into alternative domains further diversify input preparation strategies. Frequency-domain techniques, such as the Fourier Transform, decompose the signal into its spectral components, providing insights into frequency-specific brain activity. The Wavelet Transform, offering a time-frequency representation, is frequently utilized to capture transient and non-

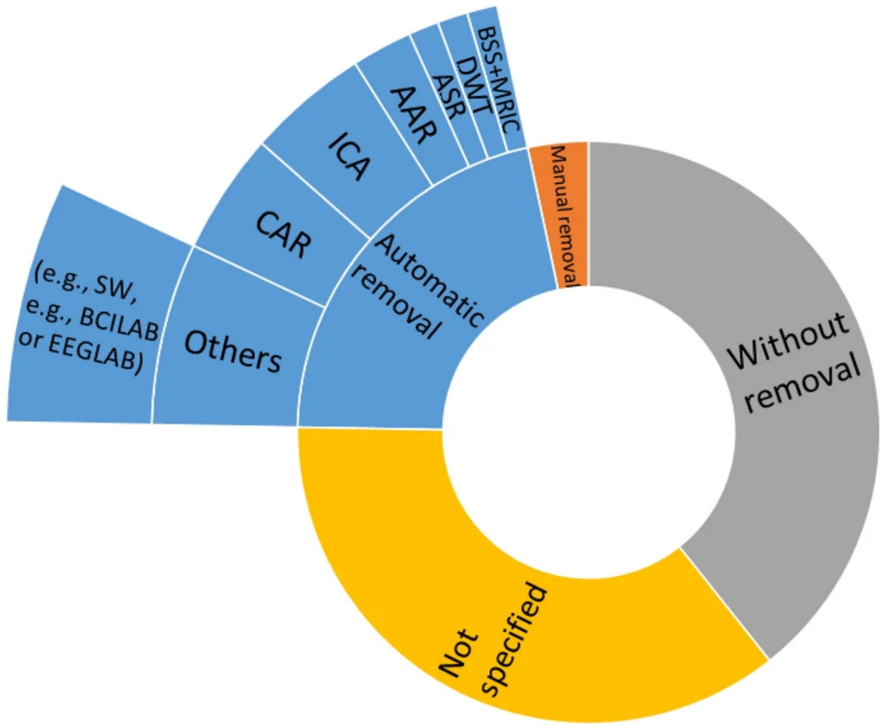


Figure 1: Artifact removal methods in Deep Learning EEG classification, as summarized in a review of 89 studies.

stationary patterns in EEG signals. These transformed representations can augment neural network models by emphasizing specific aspects of brain activity relevant to the classification task. [20]

Figure 1 is complemented by another systematic review that analyzed 154 studies on deep learning-based EEG analysis. [21] This review explored different input strategies and neural network architectures, shedding light on trends across various methodologies. The insights from these reviews reinforce the importance of aligning input preparation strategies with specific research objectives.

This research will focus on using the Adaptive Chirplet Transform (ACT) as the chosen signal processing method due to its promising results in capturing time-frequency characteristics in other domains. The ACT's ability to provide detailed temporal and spectral representations aligns well with the goals of this study, offering a powerful alternative to traditional signal processing and feature extraction techniques. By applying the ACT, the aim is to enhance the quality of the inputs to the neural networks and explore its potential to improve EEG and Deep Learning based sleep stage classification performance.

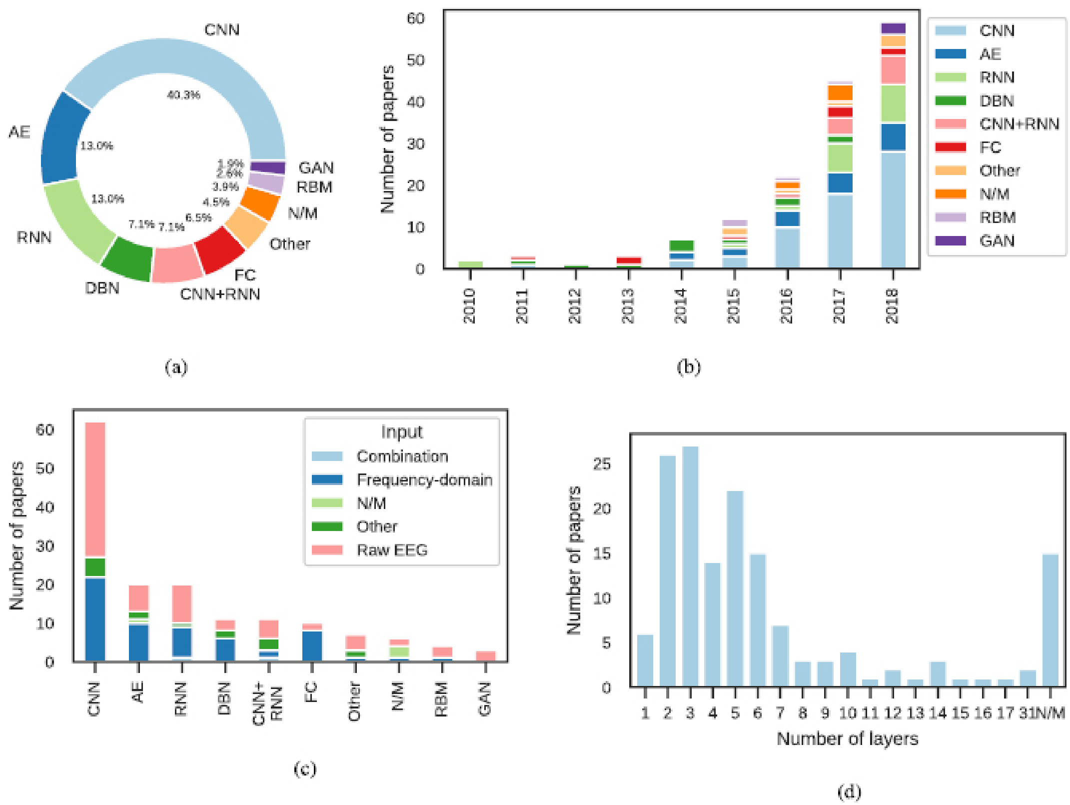


Figure 2: Deep learning architectures across 154 studies analyzed in "Deep learning-based electroencephalography analysis: a systematic review". 'N/M' stands for 'Not mentioned' and accounts for papers which have not reported the respective deep learning methodology aspect under analysis. (a) Architectures. (b) Distribution of architectures across years. (c) Distribution of input type according to the architecture category. (d) Distribution of number of neural network layers.

3.4.2 Deep Learning Architectures

Deep learning architectures play a crucial role in determining the performance and adaptability of models used for sleep stage detection with EEG data. The aforementioned review article of 89 studies on deep learning for EEG classification provides valuable insights into the variety of architectures employed in the field. [19] The findings are summarized in Figure 3, which illustrates the distribution of different deep learning architectures across these studies.

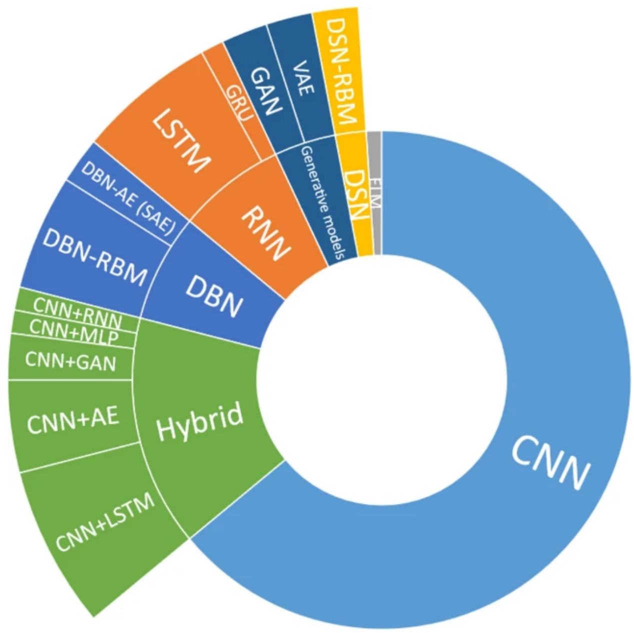


Figure 3: Deep learning architectures utilized in EEG sleep stage detection.

The figure above highlights the diversity of approaches, including CNNs, RNNs, LSTMs, GANs, VAEs, and hybrid models, among others.

The landscape of deep learning applications for EEG-based sleep stage detection is marked by a variety of innovative approaches. Each architecture offers unique strengths and capabilities, reflecting the diverse challenges associated with analyzing EEG data. From capturing intricate spatial patterns to modeling temporal dynamics, the choice of architecture is often influenced by the specific requirements of the task, the nature of the data, and the desired level of interpretability. By exploring the relative merits and limitations of these architectures, researchers aim to uncover optimal strategies for improving classification performance and advancing the state of the field.

Convolutional Neural Networks (CNNs) are widely used for their ability to automatically extract spatial features from EEG data. [22] They excel in handling structured grid-like data and are particularly effective for time-series and spectrogram-based EEG representations. Many studies employed CNNs in pure form or hybridized with other architectures to enhance performance. Recurrent Neural Networks (RNNs), designed for sequential data, capture temporal dependencies in EEG signals and are particularly suited for tasks requiring the modeling of temporal dynamics, such as sleep stage transitions. Long Short-Term Memory Networks (LSTMs), a specialized type of RNN, address the vanishing gradient problem, enabling them to capture long-term dependencies in sequential data. Similarly, the Gated Recurrent Unit (GRU) is a specialized type of RNN, simpler than an LSTM with fewer parameters and faster training but typically achieving similar results as LSTMs. [23] These are frequently used in hybrid models to complement the spatial feature extraction capabilities of CNNs.

Deep Belief Networks (DBNs), comprising layers of stacked restricted Boltzmann machines, were among the earlier deep learning architectures applied to EEG data. Although less commonly used in recent years, they have shown utility in unsupervised feature learning.

Autoencoders are particularly useful for dimensionality reduction and feature extraction, learning compact representations of EEG data, and are often used as a preprocessing step before classification.

Hybrid models combine multiple architectures such as CNN-LSTM hybrids. These models are designed to capture both spatial and temporal dynamics of EEG signals, thereby improving classification accuracy.

While CNNs and their hybrid variations dominated the reviewed studies, the lack of consensus on the optimal architecture underscores the exploratory nature of this field.

Moreover, it is important to note that this review was conducted prior to the advent of Transformers, a groundbreaking architecture that has revolutionized AI by beating state of the art performance in applications that capture long-range dependencies and attention-based modeling. Transformers have already demonstrated immense potential in domains beyond EEG analysis, and their application to sleep stage detection represents a promising area for future research. [24]

The diversity of architectures highlighted in the review emphasizes the ongoing evolution of deep learning methodologies for EEG-based sleep stage detection.

3.5 Summary of Research Gaps and Objectives

3.5.1 Identified Gaps

Despite advancements, the application of the ACT to EEG data remains limited, with few studies exploring its full potential. Similarly, the integration of advanced feature extraction methods with deep learning is still an emerging field.

There is a lack of studies integrating the ACT with deep learning for EEG analysis, particularly in sleep stage detection. The potential of the ACT for capturing subtle physiological changes has not been fully realized.

Furthermore, there is no consensus on the optimal deep learning architecture for sleep stage detection. The field of Machine Learning is extremely active, with novel architectures being developed on a weekly basis.

4 Methods

A Python pipeline was developed for this thesis with the purpose of evaluating the Adaptive Chirplet Transform as a novel feature extraction method for EEG based sleep stage prediction.

Below follows a diagram of the full pipeline from raw EEG signal to sleep stage prediction:

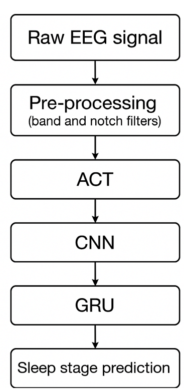


Figure 4: Diagram of pipeline from raw EEG data to final sleep stage prediction

4.1 Pre-Processing

As aforementioned, one of the key parts of any study using EEG data is to pre-process the EEG signals in an attempt to remove as much noise in the data as possible. For this thesis a Python pipeline for EEG data pre-processing was developed. The first step in the pipeline is to download all of the data as an MNE raw data class. MNE is the most commonly used Python library for analysis of EEG and MEG data and it is used to implement the filters used in this pipeline. [25] The pre-processing pipeline includes bandpass filtering to isolate relevant frequency bands associated with sleep stages and notch filtering to eliminate powerline interference, a common artifact in EEG data. The bandpass filter between 1 to 40 Hz was applied using a zero-phase FIR filter with a Hamming window, followed by a notch filter at 50 Hz to eliminate power line interference; both filters were implemented using MNE-Python's `raw_data.filter` and `raw_data.notch_filter` functions with a `firwin` design. The firwin design was a chosen parameter which has been deemed ideal for EEG signal processing. From Widmann et Al. *Digital filter design for electrophysiological data - a pratical approach*: "FIR filters are easier to control, are always stable, have a well-defined passband, can be corrected to zero-phase without additional computations, and can be converted to minimum-phase. We therefore recommend FIR filters for most purposes in electrophysiological data analysis." [26]

These preprocessing steps are essential for ensuring that the EEG signals have some reduction from noise and artifacts while retaining the physiological features necessary for accurate sleep stage classification. The pipeline was designed to process EEG data in a consistent and scalable manner, facilitating future experimentation with both small-scale and large-scale datasets.

Another significant area of progress is the meticulous design of data segmentation and epoching strategies. Recognizing that the accurate detection of sleep state transitions requires both temporal precision and computational feasibility, epoch lengths were carefully chosen. These lengths were optimized to strike a balance between capturing rapid changes in brain activity and maintaining manageable computational loads for the ACT and subsequent analyses. This alignment ensures that the data preparation phase supports the broader objective of precise sleep stage detection.

4.1.1 Windowing and Overlap Strategy

Following the initial cleaning of the data, the EEG signal was segmented into overlapping epochs of 15 seconds. Windowing was applied to each segment to reduce spectral leakage, which are artifacts introduced by the assumption that the signal is periodic across window boundaries.

A variety of window functions were considered, each balancing trade-offs in main-lobe width, side-lobe attenuation, computational cost and ease of implementation.

The following windows were considered: Dirichlet, Bartlett, Hann, Hamming, Blackman and Kaiser. The **rectangular window** (Dirichlet window) applies no tapering, preserving amplitude but leading to significant spectral leakage due to high and slowly decaying side lobes. The **Bartlett window** (triangular) offers slightly reduced leakage but lower energy retention. **Hann** and **Hamming** windows are cosine-based functions commonly used in signal processing. Hamming has a slightly narrower main lobe and better side-lobe attenuation than Hann. The **Blackman window** provides even greater side-lobe suppression at the cost of a wider main lobe and increased computational load. Lastly, the **Kaiser window**, which is based on the Bessel function, offers flexible trade-offs by adjusting the shape parameter β.

Table 1: Comparison of Common Window Functions

Window Type	Main-lobe Width	Stopband Attenuation	Features
Rectangular	Narrow	~13 dB	High leakage
Bartlett (Triangular)	Moderate	~26 dB	Simple, low leakage
Hann (Hanning)	Moderate	44 dB	Smooth cosine taper
Hamming	Moderate	53 dB	Better passband performance than Hann
Blackman	Wide	74 dB	Excellent attenuation, more computation
Kaiser ($\beta = 6$–8)	Adjustable	Flexible	Tunable trade-off via β

For this thesis, the **Hamming window** was selected for its favorable balance of computational efficiency, main-lobe width, and side-lobe suppression. It is also the default window in MNE-Python's FIR filter implementation, which was used for the band-pass filter and notch filter that were applied in the previous step of the pipeline.

To mitigate the known issue of edge degradation, where signal quality is poorer at the edges of a window, a 75% overlap was used between consecutive epochs. This ensures that each time point is well represented in at least one central region of a window, where approximation quality is highest. The impact of windowing on signal quality is illustrated in Figure 5.

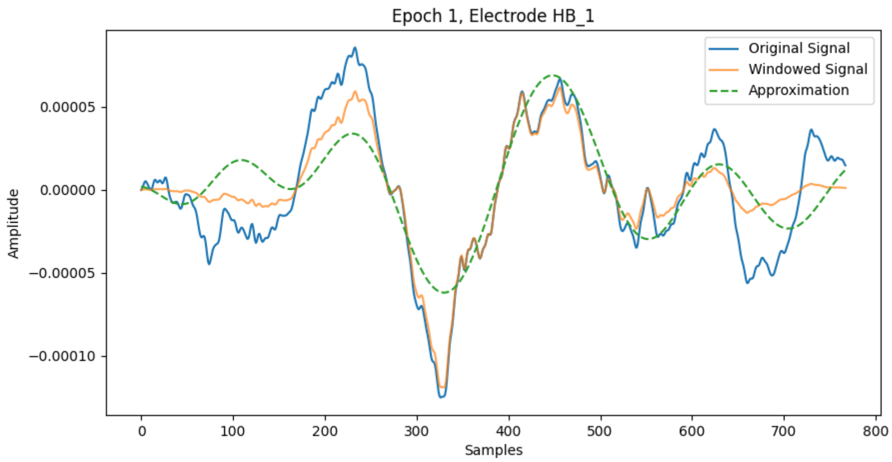


Figure 5: Effect of Hamming Window on Signal Approximation

The image above shows just how much difference the application of a Hamming window made when compared to a Dirichlet window in terms od approximation accuracy. Using the same exact ACT parameters and ACT dictionary, the signal reconstruction under the Hamming window is visibly considerably superior.

4.2 Dataset

To further strengthen the study, the Bitbrain Open Acess Sleep Database was selected for analysis. [27] This dataset was chosen due to its high quality labeling procedure and abundant amount of sleep data, which provided enough samples to train a machine learning model on it. Providing a more reliable foundation for the thesis experiments and a better ground-truth to compare the model's accuracy against.

The dataset consists of 128 full nights of sleep data from 108 participants, including Polysomnography (PSG) data, EEG data, and a sleep stage label.

The sleep stage label included Wake, NonREM sleep stages one through three, REM sleep, PSG disconnection, and lastly a label for artifact or missing data.

Additionally, the labelling was done by three expert sleep scorers independently, and they followed the criteria developed by the Academy of Sleep Medicine. [28] Labels were provided for epochs of 30 seconds, and label had to be agreed upon by at least two of the scorers. In cases where a consensus was not achieved between the three experts, a fourth one was brought in to make the final decision. This was done to reduce possible human error and disagreement in sleep scoring, as there is inherent variability between experts when it comes to classifying sleep stages, with an estimated inter-scorer agreement of approximately 85%. [29]

The selected dataset was subjected to the preprocessing pipeline, downloading both EEG channels into an MNE object and then applying the band pass and notch filters, followed by data segmentation and epoching and the application of the ACT.

4.3 Adaptive Chirplet Transform

Building upon the theoretical background of the algorithm presented in the background, this section details how the ACT was implemented and adapted for use in this thesis.

4.3.1 Dictionary parameter tuning

The ACT Python open source implementation from Barghava was used as a baseline. [9] However, early testing, particularly informed by a conference paper published by our research group, revealed many changes were necessary to the initialization parameters for the ACT class and dictionary generation. [30]

The chirplet dictionary is created by generating Gaussian chirplets over the following user-defined ranges: $t_c \in [t_{\text{min}}, t_{\text{max}}]$ in steps of δt, $f_c \in [f_{\text{min}}, f_{\text{max}}]$ in steps of δf, $\log \Delta t \in [\log \Delta t_{\text{min}}, \log \Delta t_{\text{max}}]$ in steps of $\delta(\log \Delta t)$, and $c \in [c_{\text{min}}, c_{\text{max}}]$ in steps of δc.

Each point in this 4D parameter space corresponds to a unique chirplet, which is generated and stored as a row in a dictionary matrix. Given an input signal, the transform iteratively selects chirplets that best approximate the signal via dot product projection, followed by local parameter refinement using numerical optimization. The result is a compact, interpretable time-frequency representation consisting of a set of optimized chirplet parameters and their associated coefficients.

The defined parameter ranges have to be carefully selected for the given problem. For example, Bhargava used this implementation to study a P300 signal. The P300 is an event related potential which has a delay between stimulus and response of roughly 250 to 500ms. Thus, when using the

ACT to analyze P300, the $[t_{\min}, t_{\max}]$ parameter may range between 0 to 500ms.

However, this time range would not work for sleep stage research, as discussed in background the scientific consensus for analyzing EEG data for sleep staging is 15 second epochs. This is roughly the amount of time that shows enough transient features in the brainwave data such that the differences between sleep stages can be captured. With shorter epochs there might not be enough frequency changes and patterns that would allow an expert or machine learning model to properly figure out which sleep stage the epoch corresponds to.

Besides this, running the ACT on 500ms alone for sleep would require thirty times more epochs to be run, which would be computationally prohibitive for processing a full night of sleep.

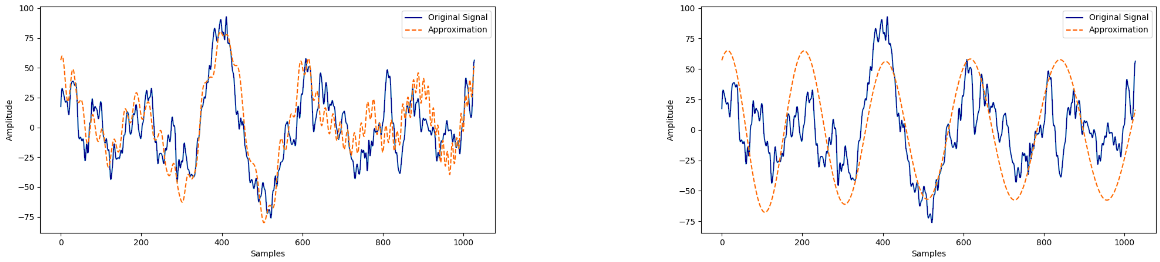
When it came to the initialization parameters used for the ACT for this study there were two majors factors two consider. First and foremost was the fact that there was a significant trade-off between computational requirements and accuracy of signal approximation.

An incredibly near perfect signal approximation and reconstruction can be achieved with the Adaptive Chirplet Transform, but it requires a dictionary so extensive that the computation time is simply too large and the RAM requirements alone make it so that it can't be run on most modern laptops.

The second factor to consider was physiological. The time ranges should match the rough time duration where enough change has happened within the EEG signal such that sleep scoring could be achieved, which is roughly between 5 to 30 seconds. Similarly, the frequency range combined with the chirp rate should match known values of frequency ranges of brainwaves, typically listed as 0.5 to 40 Hz. These provide the bounds and guidelines on the minimum and maximum ranges of each of the parameters.

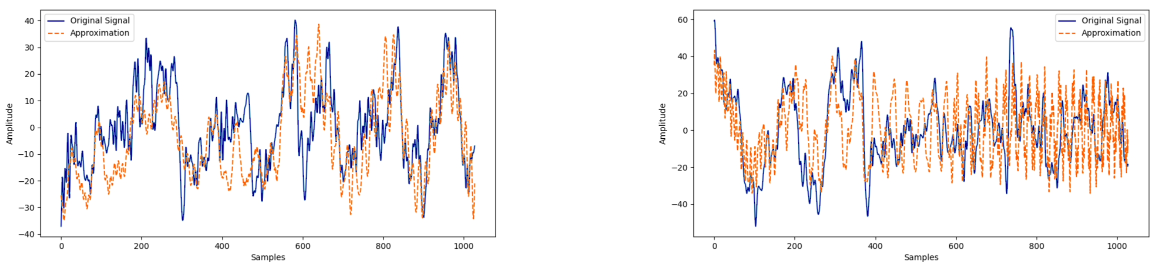
The step size, however, is purely motivated by the above mentioned trade off of computational requirements and overall signal quality. Figures below demonstrate the effect of varying step size and parameters on the accuracy of the approximation.

From the images below, it is clear to see that the smaller the step size, the better the approximation becomes. Note that the difference between 0.1 to 0.5 in frequency step size was the largest, where with 0.1 the reconstructed signal is a very close match of the original, especially in the middle, with some leakage from the sides causing a slightly less accurate reconstruction. This is in part due to the fact that the image above was constructed using a Dirichlet window, which has



(a) 0.1 (b) 0.5

Figure 6: Impact of the Frequency Step Size on Approximation of the ACT.



(a) 20 (b) 200

Figure 7: Impact of the Time Step Size on the Approximation of the ACT

this known effect. The final parameters were constructed using a Hamming window which greatly minimizes the seen leakage effect at the boundary.

The visualization code that generated the graphs above was used to analyze epoch by epoch of several different parameter step sizes and after trial and error a final parameter range was decided upon. The final parameters are shown below:

Final Parameters: t_c: (0, 3840, 64); f_c: (0.6, 15, 0.2)

The time scale is in samples rather than seconds, since the sampling rate of the EEG recording device used was 256Hz, 3840 represents 15 seconds worth of sleep EEG data. A number that was chosen to follow the literature consensus on sleep stage analysis.

4.3.2 Parallelization of code

The original implementation of the Adaptive Chirplet Transform (ACT) was not well-suited for large-scale processing. It relied on CPU-based loops and NumPy operations, which made it impractical for analyzing full-length EEG recordings, especially when using longer epochs, high overlap, or finely-sampled parameter dictionaries. After reviewing performance limitations, I rewrote the pipeline to take advantage of GPU computing and parallel processing.

To run the transform more efficiently, the ACT code was adapted to work with CuPy, which mirrors the interface of NumPy but runs operations on the GPU. The new implementation was run on a Lambda Labs server equipped with an NVIDIA A100 SXM4 GPU with 40 GB of VRAM. The EEG data, along with the generated chirplet dictionary and intermediate arrays like window functions, were all transferred to GPU memory so that the ACT could be computed without moving data between devices during processing.

One of the main changes involved rewriting how epochs were created and processed. Instead of slicing each epoch on the CPU, the entire EEG signal was transferred to GPU memory at once. Epoch segmentation, windowing, and chirplet projections were then done in parallel on the GPU. The transform itself, especially the dictionary matching and optimization steps, saw major speedups due to the GPU's ability to handle thousands of operations at once. Only the final results, such as the fitted parameters and coefficients, were transferred back to CPU memory for saving and plotting on csv files. For reference, a csv file containing one full night of sleep was on average 900 Mega bytes long.

These changes made the ACT usable for full-night recordings and also opened the door to running more detailed parameter sweeps and real-time variants in future work. The entire GPU pipeline was integrated into a single script that could batch process subjects and automatically save results in a format suitable for later use in neural network training.

Despite these improvements, with the ACT dictionary parameters used, and the 75% window overlap, it still took 33.4 hours to run 3 out of the 128 full nights of sleep, costing a total of USD$43.11. With an average time of eleven hours of GPU time to process eight hours worth of EEG. The computational complexity of the Adaptive Chirplet Transform remains the biggest roadblock preventing it from having the capacity of becoming the standard EEG feature extraction method and achieving widespread usage in EEG data analysis and other domains and fields of signal processing.

Due to financial constraints and not having access to GPUs in house, this research was limited to only processing three out of the one hundred and twenty eight nights of sleep. Due to the sheer amount of epochs, in part due to the 75% overlap, three nights of data was still enough to provide over twelve thousand epochs. However, after processing it so it would be suitable for the Neural Network, it ended up resulting in 10,633 samples.

4.4 Deep Learning Based Sleep Stage Detection

This section details every step taken from preparing the data post ACT to making it suitable to be used as an input into the Neural Network, and the design and architecture of the Neural Network itself.

4.4.1 Input preparation for Neural Network

The implementation of the ACT used returns a csv file which contains per row, a list of coefficients (of length 5 since a 5^{th} order transform was employed), a list of parameters (of length 20), a list of residue, original signal, and error.

A separate Python pipeline was coded to handle the data preparation for the Neural Network, starting from downloading the csv files containing the results from the ACT.

The pipeline begins by extracting only the coefficients and parameters of each epoch, and transforming it into a 5x5 matrix. Where the rows of the matrix represent a each respective ACT order approximation. Such that each coefficient was joined with its respective function defining four parameters, and the first column of the matrix stood for the coefficients, wheres the remaining four columns were filled with the parameters.

This restructuring was done to make use of the power of two-dimensional Convolutional Neural Networks, which have been established as the most commonly used architectures for EEG data. Furthermore, this ordering made the structure of the data more explicit with the hopes it would be easier for the Neural Network to identify the most important features of the data faster.

For example, within this ACT resultant 5x5 matrix, the first column will always be more important than the second and so forth. This information could be given to the Neural Network with careful selection of initialization of weights.

Another crucial step with the data was to relabel it carefully. As previously mentioned, the original EEG data from Bitbrain contained labels per every 30 second epoch. However, since the Hamming window was applied with 75% overlap, the epochs no longer spanned only single labeled epochs. Rather, there was a large amount of epochs that spanned two different labels. To ensure no sample points were unnecessarily lost, an algorithm was written to calculate the exact time of each new ACT processed epoch and map it back to the original labeled epochs. Then, the algorithm checked for every epoch that spanned two labels (due to the duration none spanned more than two), and checked whether the overlapped labels were the same or different. If the sample was overlapping a sleep stage transition period (meaning it spanned two different labels) it was discarded from the dataset.

Following this, epochs assigned to labels -2 (PSG disconnected), 3 (Non REM sleep stage 3), and 8 (artifact) were removed from the classification. This was because within the 3 subjects data alone there was essentially no data points for these labels (less than 0.1% of the data). This was not enough for the model to learn to distinguish these patterns but if included it made the learning process of the model much slower and had a significant negative effect on the performance and accuracy of the model. Regardless, two of the stages represent mal-labeled data, so removing it altogether from the dataset was an improvement in the overall data quality used for training the model and made it so that the model focused solely on the standard sleep stages, Wake, N1, N2, and REM.

The dataset was then split into training, validation, and test sets using a stratified splitting strategy. Stratification was done to preserve the original label distribution across each of the 4 categories, which was especially important because some categories had considerably more data points than others. The data was split such that 70% was used for training and the remaining 30% was split evenly between validation and training.

To mitigate the issue of massive class imbalance, the training data was oversampled to balance class count. The oversampling was done by duplicating samples for the underrepresented classes until all classes had the same number of examples. This ensures the model would not bias towards more frequent stages during training.

Oversampling improved the accuracy of the model considerably, almost doubled it in fact. However, it is only done for the training data and not for the validation or test. This is a common practice in the field of Machine Learning, as the validation and test are meant to be representative of how the model would generalize and perform with real world data, oversampling is generally avoided and frowned upon for validation and test sets.

After the oversampling, all feature tensors were standardized using StandardScalar from the sklearn library. This transforms the tensors such that they have a mean of zero and unit variance.

Lastly, the input tensors were grouped into small batches of a given sequence length. The sequence length used for this particular model was four, this value was a hyperparameter of the model. The purpose of these batches was so that temporal information could be passed into the model and to take advantage of the fact that typically sleep stages remain constant for much longer than the period of an individual epoch. Therefore, if one knows the sleep stages of a given amount of previous epochs, this information can be used to influence the decision of the model. If the previous three epochs were all at a given sleep stage, then the probability that the current epoch is at that same sleep stage should be higher.

These batches were shuffled before generating the validation/train/test splits to help with generalization of the model.

4.4.2 CNN + GRU Hybrid Model

To model the spatial and temporal structure of the chirplet-transformed EEG data, a hybrid neural network architecture combining Convolutional Neural Networks (CNNs) with Gated Recurrent Units (GRUs) was used.

The CNN component is applied independently to each time step in the input sequence. Each input matrix has shape 1×5×5 (channels × height × width), and is passed through a single convolutional layer with 16 output channels and a kernel size of 3×3 (with padding to preserve spatial dimensions). This layer captures local patterns between chirplet features while maintaining a manageable number of parameters. The resulting activation for each time step has shape 16×5×5, which is then flattened into a vector of size 400. These 400-dimensional vectors are stacked in temporal order, forming a [batch size, sequence length, 400] tensor passed to the GRU.

The GRU layer contains 2 stacked layers, each with a hidden size of 64. It processes the temporal sequence of CNN-extracted features in a recurrent manner, learning to model dependencies and transitions across time. The final hidden state from the GRU is used as a summary representation of the entire sequence. This final state is passed through a fully connected linear layer to produce class scores for sleep stage classification.

The CNN-GRU model was trained using cross-entropy loss with class weights to compensate for class imbalance, and optimized using the Adam optimizer. Class weights were computed from the training data after oversampling to maintain fairness across classes.

5 Results and Analysis

Below is the confusion matrix from running the CNNGRU model described above for 100 epochs, with a learning rate of 0.001, batch size of eight and sequence length of four on the Chirplet Transformed data from the first 3 nights of sleep of the Bitbrain Dataset.

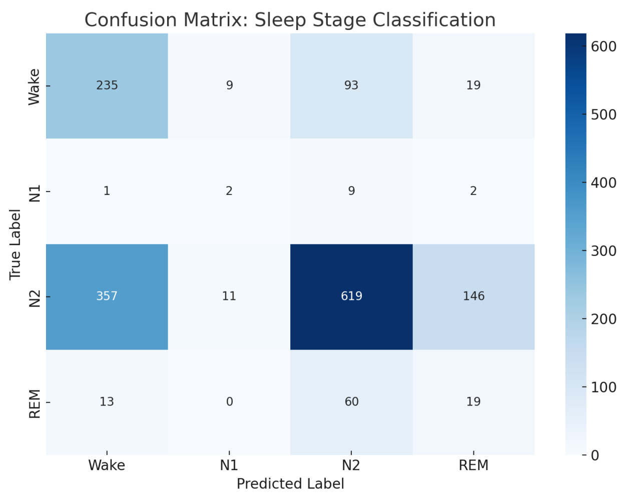


Figure 8: Confusion Matrix - Final Model Result

The model achieved a final training accuracy of 91.54%, indicating that it was able to effectively learn the relationships between the input features and the sleep stage labels during training. This high training performance suggests that the Adaptive Chirplet Transform (ACT) is a strong candidate for feature extraction in EEG-based sleep stage classification, thus validating the initial hypothesis of this thesis.

Since the ACT captures both time and frequency domain features with high precision, it allowed the CNN-GRU model to extract the most meaningful discriminative patterns for each sleep stage.

Furthermore, it does this by also achieving incredible compression as it approximates a 15 second epoch of raw signal containing 3840 sample points (stored as floats) using 25 floats instead, comprised of the chirplet coefficients and parameters.

The compression ratio (CR) is defined as the ratio of the original signal length to the compressed one:

$$\text{Compression Ratio} = \frac{\text{Original size}}{\text{Compressed size}} = \frac{3840}{25} = 153.6 : 1 \tag{2}$$

Equivalently, this corresponds to a compression rate of:

$$\text{Compression Rate} = \frac{25}{3840} \approx 0.00651 \quad \Rightarrow \quad 0.651\% \text{ of the original signal retained} \tag{3}$$

This indicates that the ACT representation retains only about 0.651% of the original signal data per epoch, highlighting its effectiveness for compact, interpretable signal encoding.

However, the validation (52.76%) and test accuracy (57.93%) were significantly lower. This gap can be attributed to the fact that only the training data was oversampled to balance class distributions, while validation and test sets still had their original imbalances. The model may have generalized poorly to underrepresented classes it didn't see as frequently in validation/testing, highlighting the detrimental impact of class imbalance on final performance evaluation.

A closer look at the confusion matrix reveals important patterns. N2 sleep was the most prevalent class in the dataset (with 1133 test samples) and had the highest number of correct predictions (682), which contributed heavily to the overall accuracy. Wake was often confused with N2, as 115 of the Wake samples were misclassified as N2. REM sleep was also frequently confused with N2 (58 instances), while N1 showed poor classification performance overall, with only 1 correctly classified sample and most misclassified as N2.

This suggests a clear prediction bias toward N2, which is likely due to its overwhelming presence in the dataset. The network, trained on oversampled data but evaluated on imbalanced splits, defaulted toward the dominant class when uncertain.

Furthermore, while the overall test accuracy is lower than other results in the area of Deep learning categorization using EEG data, it is important to note that the particular issue of sleep staging is complicated and there is intervariability even amongst human experts, with results of about 85% agreement. This gives important context to the results.

6 Conclusion and Future Work

The results achieved with this thesis show that the Adaptive Chirplet Transform has great potential to be an excellent method of feature extraction for EEG based Deep Learning applications. It manages to extract the most meaningful features of the data and it does so with astounding compression rate. A feature that could be very useful for Brain Chip Interface (BCI) applications where memory tends to be limited.

The relatively low accuracies displayed in the validation and test sets are to some extent explained by the large class imbalance in the data used. This can be likely vastly improved if the full pipeline is run on the full Bitbrain dataset, going through one hundred and twenty eight full nights of sleep rather than just three.

The largest roadblock at this time remains the computational complexity of the Adaptive Chirplet Transform. With the final ACT generation parameters used, the dictionary contains over 1.4 million Chirplet Gaussian functions, which makes it so that each epoch has to go through many dot products before finding the optimal chirplets. This currently takes around eleven hours to process eight hours worth of EEG data on A100 GPUs, making it almost but not quite suitable for real-time performance.

It is quite possible that this time complexity could be greatly reduced by the generation of a more optimized dictionary. For example, instead of generating every single Chirplet Gaussian from the minimum and maximum values for the four parameters at the given step size, the dictionary could be crafted by the Chirplet Gaussian functions that are most widely seen and used. Further research is required to determine a reasonable dictionary size, and exactly how much this can speed up the ACT computation time. Once again, the optimal answer will undoubtedly come down to a trade-off between reconstruction accuracy and computational complexity.

A possible approach would be to run the current ACT version for a handful more hours' worth of data while keeping track of all of the chirplet functions that were chosen. Then create a dictionary solely from these chosen functions. This approach could also be done adaptively.

As for direct future steps for this research, given access to GPUs and the computational resources. The remainder of the Bitbrain dataset could be run with the current ACT implementation and the model could be trained with a much larger and more balanced dataset. This would test and hopefully confirm the hypothesis that the high training accuracy but low validation and test accuracies can be largely explained by the class imbalance in the current validation and test datasets.

References

[1] L. Hu and Z. Zhang, *EEG signal processing and feature extraction*. Springer, 2019.

[2] J. Cui and W. Wong, "The adaptive chirplet transform and visual evoked potentials," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 7, pp. 1378–1384, 2006.

[3] S. Mann and S. Haykin, "Adaptive chirplet: an adaptive generalized wavelet-like transform," in *Adaptive Signal Processing* (S. Haykin, ed.), vol. 1565, pp. 402 – 413, International Society for Optics and Photonics, SPIE, 1991.

[4] Y. Jiang, W. Chen, M. Li, T. Zhang, and Y. You, "Synchroextracting chirplet transform-based epileptic seizures detection using eeg," *Biomedical Signal Processing and Control*, vol. 68, p. 102699, 2021.

[5] J. Cui and W. Wong, "Investigation of short-term changes in visual evoked potentials with windowed adaptive chirplet transform," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 4, pp. 1449–1454, 2008.

[6] M. Kaur, R. Upadhyay, and V. Kumar, "A hybrid deep learning framework using scaling-basis chirplet transform for motor imagery eeg recognition in brain–computer interface applications," *International Journal of Imaging Systems and Technology*, vol. 34, no. 4, p. e23127, 2024.

[7] C. López and K. J. Moore, "Enhanced adaptive linear chirplet transform for crossing frequency trajectories," *Journal of Sound and Vibration*, vol. 578, p. 118358, 2024.

[8] G. Zhang, Y. Cui, Y. Zhang, H. Cao, G. Zhou, H. Shu, D. Yao, Y. Xia, K. Chen, and D. Guo, "Computational exploration of dynamic mechanisms of steady state visual evoked potentials at the whole brain level," *NeuroImage*, vol. 237, p. 118166, 2021.

[9] A. Bhargava and S. Mann, "Adaptive chirplet transform-based machine learning for p300 brainwave classification," in *2020 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES)*, pp. 62–67, 2021.

[10] S. S. Shinde and S. S. Kamthekar, “Person authentication using eeg signal that uses chirplet and svm,” in *Computational Vision and Bio-Inspired Computing: ICCVBIC 2019*, pp. 338–344, Springer, 2020.

[11] J.-W. Qiu, J. K. Zao, P.-H. Wang, and Y.-H. Chou, “Consistent sparse representations of eeg erp and ica components based on wavelet and chirplet dictionaries,” in *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, pp. 4014–4019, IEEE, 2010.

[12] Y. Lu, R. Demirli, G. Cardoso, and J. Saniie, “A successive parameter estimation algorithm for chirplet signal decomposition,” *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 53, no. 11, pp. 2121–2131, 2006.

[13] S. Motamedi-Fakhr, M. Moshrefi-Torbati, M. Hill, C. M. Hill, and P. R. White, “Signal processing techniques applied to human sleep eeg signals—a review,” *Biomedical Signal Processing and Control*, vol. 10, pp. 21–33, 2014.

[14] X. Zhu, Z. Zhang, Z. Li, J. Gao, X. Huang, and G. Wen, “Multiple squeezes from adaptive chirplet transform,” *Signal Processing*, vol. 163, pp. 26–40, 2019.

[15] C. K. Chui, Q. Jiang, L. Li, and J. Lu, “Analysis of a direct separation method based on adaptive chirplet transform for signals with crossover instantaneous frequencies,” *Applied and Computational Harmonic Analysis*, vol. 62, pp. 24–40, 2023.

[16] Y. Qin, R. Yang, H. Shi, B. He, and Y. Mao, “Adaptive fast chirplet transform and its application into rolling bearing fault diagnosis under time-varying speed condition,” *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–12, 2023.

[17] M. Alaee and H. Amindavar, “Chirplet-based target recognition using radar technology,” in *2008 5th IEEE Sensor Array and Multichannel Signal Processing Workshop*, pp. 451–454, 2008.

[18] Y. Jiang, W. Chen, M. Li, T. Zhang, and Y. You, “Synchroextracting chirplet transform-based epileptic seizures detection using eeg,” *Biomedical Signal Processing and Control*, vol. 68, p. 102699, 2021.

[19] H. Altaheri, G. Muhammad, M. Alsulaiman, *et al.*, “Deep learning techniques for classification of electroencephalogram (eeg) motor imagery (mi) signals: a review,” *Neural Computing and Applications*, vol. 35, pp. 14681–14722, 2023.

[20] M.-P. Hosseini, A. Hosseini, and K. Ahi, “A review on machine learning for eeg signal processing in bioengineering,” *IEEE Reviews in Biomedical Engineering*, vol. 14, pp. 204–218, 2021.

[21] Y. Roy, H. Banville, I. Albuquerque, A. Gramfort, T. H. Falk, and J. Faubert, “Deep learning-based electroencephalography analysis: a systematic review,” *Journal of Neural Engineering*, vol. 16, no. 5, p. 051001, 2019.

[22] K. O’Shea and R. Nash, “An introduction to convolutional neural networks,” 2015.

[23] A. Sherstinsky, “Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network,” *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, Mar. 2020.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” 2023.

[25] A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, R. Goj, M. Jas, T. Brooks, L. Parkkonen, and M. Hämäläinen, “Meg and eeg data analysis with mne-python,” *Frontiers in Neuroscience*, vol. 7, 2013.

[26] A. Widmann, E. Schröger, and B. Maess, “Digital filter design for electrophysiological data – a practical approach,” *Journal of Neuroscience Methods*, vol. 250, pp. 34–46, 2015. Cutting-edge EEG Methods.

[27] E. López-Larraz, M. Sierra-Torralba, S. Clemente, G. Fierro, D. Oriol, J. Minguez, L. Montesano, and J. G. Klinzing, “"bitbrain open access sleep dataset",” 2025.

[28] R. Berry, S. Quan, and A. Abreu, *The AASM Manual for the Scoring of Sleep and Associated Events: Rules, Terminology and Technical Specifications*. Darien: American Academy of Sleep Medicine, version 2.6 ed., 2020.

[29] R. S. Rosenberg and S. Van Hout, “The american academy of sleep medicine inter-scorer reliability program: sleep stage scoring,” *Journal of Clinical Sleep Medicine*, vol. 9, pp. 81–87, 2013.

[30] S. Mann, N. Kumar, J. P. Bicalho, M. Sibai, and C. Leaver-Preyra, “Adaptive chirplet transform-based sleep state detection,” in *2025 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 1–6, 2025.