Universidad Zaragoza

Master's Thesis

Self-supervised learning for EEG-based automatic sleep staging

Author

Emilio Estevan Tomás

Supervisor

Luis Montesano del Campo

Master in Robotics, Graphics and Computer Vision

ESCUELA DE INGENIERÍA Y ARQUITECTURA

2025

Acknowledgements

I would like to express my deepest gratitude and appreciation to all the members of the Bitbrain team who have contributed to and supported the completion of this thesis. Your expertise, feedback, and encouragement have been invaluable throughout this journey.

Especially, I want to thank my supervisors, Luis Montesano and María Sierra, for the opportunity to work under your mentorship and participate in such an innovative project, as well as for your unwavering guidance, insights, and encouragement during the research period. It is a truly honor to keep growing and learning alongside you.

Last but certainly not least, I extend my heartfelt gratitude to my family, friends, and partner for their understanding and support through the thesis and, more broadly, throughout all the ups and downs of my academic and personal life. I am deeply grateful to have you by my side as I move forward into the next chapter of my life.

Abstract

Sleep is crucial within the spectrum of human health. A significant number of people suffer from sleep-related disorders, representing a growing public health concern. Polysomnography (PSG) is the gold standard for sleep evaluation, which begins with the sleep staging procedure. PSG recordings are segmented into 30-second intervals and categorized into one of the well-defined sleep stages. However, it is a very expensive process performed manually by expert technicians, and is highly sleep-invasive on patients due to the large number of sensors placed on the body.

Recent advancements in deep learning are revolutionizing automatic sleep staging, achieving, and even surpassing, expert-level accuracy, thereby overcoming some limitations of the manual scoring process. Given the drawbacks of traditional sleep monitoring and the potential of learning-based sleep staging approaches, Bitbrain Technologies has developed a fully textile-based, wearable, and user-friendly EEG garment to serve as a scalable solution for sleep analysis, also including a deep learning system with medical-grade precision trained on labeled datasets. Nevertheless, deep learning methods are notoriously data-hungry, necessitating from large volumes of labeled data for effective training. The process of signal annotation is as resource-intensive as the manual scoring procedure carried by technicians. Therefore, the mass adoption of home sleep monitoring enabled with this novel device generates a massive amount of EEG data that is impossible to label, resulting in enormous datasets that Bitbrain cannot leverage for supervised training of its models.

As a consequence, the objective of this work is to develop and evaluate self-supervised learning (SSL) methods as a pre-training step within the learning pipeline to learn electroencephalogram representations from unlabeled data, reducing the costs of the manual annotation process. To achieve this, state-of-the-art deep learning and self-supervised methods for automatic sleep staging are reviewed, implemented, and evaluated across different scenarios, comprising two different datasets recorded using Bitbrain's device: the HOGAR study, an unlabeled dataset acquired under domestic conditions to quantify cognitive function in populations at risk of dementia, and the BOAS dataset, a labeled resource collected in a controlled laboratory environment with the aim of closing the gap between traditional clinical PSG technologies and portable EEG solutions.

As a result, the transferability between both datasets was thoroughly assessed and successfully demonstrated, obtaining significant improvements by incorporating SSL techniques into the learning pipeline. This is further supported by a feature visualization analysis performed using t-SNE and UMAP tools. On the other side, a comprehensive understanding of the performance and limitations of SSL relative to the labeled data available was provided, exhibiting particularly pronounced accuracy gains in low-data regimes. Consequently, this work serves as a critical step forward in the design and validation of label-efficient, self-administered, and large-scale automatic sleep monitoring systems in home environments under uncontrolled conditions, advancing in a scalable solution for the substantial proportion of the word population suffering from serious sleep disorders that require medical attention.

Index

1 Introduction **1**

1.1 Sleep architecture . . . 3

1.2 Electroencephalography (EEG) . . . 4

1.2.1 Frequency bands . . . 4

1.2.2 Technical features . . . 5

1.3 Objectives and scope of the project . . . 6

2 Related work **7**

2.1 Automatic sleep scoring . . . 7

2.1.1 Shallow learning . . . 7

2.1.2 Deep learning . . . 7

2.2 Self-supervised learning . . . 9

2.2.1 Contrastive learning techniques . . . 10

2.2.2 Masked prediction techniques . . . 11

2.2.3 Self-supervised learning for sleep staging . . . 11

3 Methods **13**

3.1 Deep learning architecture . . . 13

3.2 Self-supervised learning methods . . . 14

3.2.1 Data augmentation techniques . . . 14

3.2.2 Contrastive learning methods . . . 16

3.2.3 Masked prediction methods . . . 20

3.3 Evaluation . . . 23

3.3.1 Datasets . . . 23

3.3.2 Performance . . . 24

4 Results and discussion **29**

4.1 Semi-supervised learning . . . 29

4.2 Linear evaluation . . . 36

4.3 Feature visualization . . . 40

5 Conclusions **43**

5.1 Future work . . . 44

Bibliography **52**

Appendices **53**

A Project management **54**

A.1 Software and hardware tools 54

A.2 Planning 54

B Artifact detection system **55**

C SSL ablation study **56**

D Electroencephalographic properties of sleep **60**

List of Figures

1.1 Illustration of the different configurations for sleep monitoring. 2
1.2 Hypnogram associated with the typical distribution of sleep stages during the night. 4
1.3 Classification of EEG waveforms based on their frequency band. 5
1.4 Placement of the standard electrodes of the 10-20 system. 6

2.1 Sequence-to-sequence scheme for automatic sleep staging. Adapted from [31]. . . 8
2.2 Neural network typical architecture. Adapted from [61]. 10

3.1 Deep learning model architecture. 13
3.2 Epoch encoder sub-model architecture. 14
3.3 Overview of the augmentation sets T_1 and T_2. 15
3.4 SimCLR framework overview. 16
3.5 BYOL framework overview. 17
3.6 SimSiam framework overview. 18
3.7 Barlow Twins framework overview. 19
3.8 ContraWR framework overview. 19
3.9 BENDR framework overview. 21
3.10 MAEEG framework overview. 22
3.11 MAEEG masking logic. Adapted from [76]. 22
3.12 Datasets available for evaluating automatic sleep staging. 24
3.13 Deep learning pipeline. 25
3.14 Self-supervised learning evaluation methodologies. 25
3.15 Evaluation scenarios for automatic sleep staging. 26
3.16 Feature visualization pipeline using t-SNE [84] and UMAP [85] tools. 28

4.1 Accuracy evolution of the different approaches corresponding to test *SEMI00*. . . 30
4.2 Accuracy evolution of the different approaches corresponding to test *SEMI01*. . . 31
4.3 Accuracy evolution of the different approaches corresponding to test *SEMI02*. . . 35
4.4 Accuracy evolution of the different approaches corresponding to test *SEMI03*. . . 35
4.5 Accuracy evolution of the different approaches corresponding to test *LINEV00*. . 37
4.6 Accuracy evolution of the different approaches corresponding to test *LINEV01*. . 38
4.7 Accuracy evolution of the different approaches corresponding to test *LINEV02*. . 39
4.8 Accuracy evolution of the different approaches corresponding to test *LINEV03*. . 40

4.9 t-SNE feature visualization results (● Wake, ● N1, ● N2, ● N3, ● REM). 41
4.10 UMAP feature visualization results (● Wake, ● N1, ● N2, ● N3, ● REM). 41

A.1 Gantt chart corresponding to the project schedule. 54

B.1 Data preprocessing pipeline comprising the reshaping, resampling, and filtering steps. 55

D.1 EEG properties of Wake sleep stage. 60
D.2 EEG properties of N1 sleep stage. 60
D.3 EEG properties of N2 sleep stage. 61
D.4 EEG properties of N3 sleep stage. 61
D.5 EEG properties of REM sleep stage. 61

List of Tables

1.1 Sleep stages and associated durations. 4

2.1 Overview of deep learning models for automatic sleep staging. 9
2.2 Overview of self-supervised learning techniques for sleep staging. 12

3.1 Overview of evaluation tests. 28

4.1 Accuracy and standard deviation comparison of the different approaches carried within the full HOGAR → BOAS scenario with distinct percentages of labeled data (X in Figure 3.15a), featuring a constant and large test set, evaluated in a semi-supervised procedure (*SEMI00*). Bold formatting highlights the top results. 30
4.2 Accuracy and standard deviation comparison of the different approaches carried within the full HOGAR → BOAS scenario with distinct percentages of labeled data (X in Figure 3.15b), featuring a 10-fold cross-validation, evaluated in a semi-supervised procedure (*SEMI01*). Bold formatting highlights the top results. . . . 31
4.3 Accuracy and standard deviation comparison of the different approaches carried within the BOAS → BOAS scenario with distinct percentages of labeled data (X in Figure 3.15c), featuring a 10-fold cross-validation, evaluated in a semi-supervised procedure (*SEMI02*). In addition, the second part shows the results for the HOGAR → BOAS scenario, where pre-training is performed on the HOGAR dataset using the exact same number of subjects as in the BOAS dataset (*SEMI03*). The percentages correspond to the BOAS dataset and are proportionally adjusted to manage the HOGAR pre-training data. Bold formatting highlights the top results. 34
4.4 Accuracy and standard deviation comparison of the different approaches carried within the full HOGAR → BOAS scenario with distinct percentages of labeled data (X in Figure 3.15a), featuring a constant and large test set, linearly evaluated (*LINEV00*). Bold formatting highlights the top results, and underlining is used for the second-best. 36

4.5 Accuracy and standard deviation comparison of the different approaches carried within the full HOGAR → BOAS scenario with distinct percentages of labeled data (X in Figure 3.15b), featuring a 10-fold cross-validation, linearly evaluated (*LINEV01*). Bold formatting highlights the top results, and underlining is used for the second-best. 37

4.6 Accuracy and standard deviation comparison of the different approaches carried within the BOAS → BOAS scenario with distinct percentages of labeled data (X in Figure 3.15c), featuring a 10-fold cross-validation, linearly evaluated (*LINEV02*). The second part shows the results for the HOGAR → BOAS scenario, where pre-training is performed on the HOGAR dataset using the exact same number of subjects as in the BOAS dataset (*LINEV03*). The percentages correspond to the BOAS dataset and are proportionally adjusted to manage the HOGAR pre-training data. Bold formatting highlights the top results, and underlining is used for the second-best. 39

C.1 SimCLR ablation study results, where τ denotes the temperature parameter of the NT-Xent loss, λ is the weight decay applied in the optimizer, and η corresponds to the learning rate. 56

C.2 BYOL ablation study results, where λ denotes the weight decay applied in the optimizer, and η corresponds to the learning rate. 57

C.3 SimSiam ablation study results, where λ denotes the weight decay applied in the optimizer, and η corresponds to the learning rate. 57

C.4 Barlow Twins ablation study results, where λ_{loss} is the hyperparameter trading off the importance between loss terms, λ_{opt} denotes the weight decay applied in the optimizer, and η corresponds to the learning rate. 58

C.5 ContraWR ablation study results, where λ denotes the weight decay applied in the optimizer, η corresponds to the learning rate, δ is the empirical margin of the loss, σ is the standard deviation of the Gaussian kernel, and τ is the temperature of the loss. 59

Chapter 1

Introduction

Sleep is crucial within the spectrum of human health, supporting a wide range of physiological functions, including immune, metabolic, and cardiovascular systems [1]–[3]. Additionally, sleep influences cognitive processes as memory consolidation or emotional regulation. A significant number of people suffer from sleep-related disorders, such as sleep apnea, insomnia and narcolepsy, representing a growing public health concern due to their impact on long-term health outcomes [4]. Effective and practical sleep assessment is essential for identifying sleep problems and enabling timely interventions. However, fewer than 20% of cases are properly diagnosed and treated [5].

Polysomnography (PSG) is regarded as the gold standard for sleep evaluation and, consequently, for the assessment and diagnosis of sleep disorders. It consists of a multi-parametric measurement of a wide range of physiological signals in parallel, including electroencephalogram (EEG), electrooculogram (EOG), electromyogram (EMG), electrocardiogram (ECG), blood oxygen saturation, and respiration. This proceeding is typically conducted in a controlled hospital setting, where patients sleep overnight for monitoring.

Sleep evaluation begins with the sleep staging procedure. PSG recordings are segmented into 30-second intervals, known as *epochs*, and every one of these is categorized into one of the well-defined sleep stages, resulting in a complete representation of the sleep phases present throughout the recording. It is a very time-consuming process performed manually by expert technicians, requiring the visual analysis of the recorded time series. Noise in data and the complexity of brain processes difficult the interpretation and annotation of the recorded physiological signals. As a consequence, reported inter-scorer variability (i.e., the accuracy between professionals scoring the same recoding) is about 82% [6], [7], while the individual scoring against a consensus group has a lower limit of 74% and an upper limit of 85% (with a mean of 81.3%) [8]. Indeed, the intra-scorer agreement (i.e., the accuracy of the same technician scoring the same recoding) reaches approximately 90% [9]. In addition, the PSG recording setup is considered to be highly sleep-invasive on patients due to the large number of sensors placed on the body, often disrupting natural sleep patterns. As a result, this technique is both time- and money-intensive, making it impossible to serve as a scalable solution for the substantial proportion of the word population suffering from serious sleep disorders and requiring medical attention.

Recent advancements in machine learning, specifically within deep learning, are revolutionizing the automatic sleep staging paradigm, achieving (and even surpassing) expert-level accuracy in terms of inter-scorer variability [10]–[12]. These models enable effective and efficient sleep classification while overcoming the limitations of the manual scoring process described before, serving as a step toward sleep technologies that democratize access to sleep studies and diagnosis.

Bitbrain Technologies is a brain technology company that combines neuroscience, artificial intelligence, and hardware to develop human monitoring technologies, specially EEG brain sensing devices. In order to address the limitations existing in the current sleep evaluation process and provide a scalable solution for people suffering from sleep pathologies and requiring medical diagnosis, it has developed the first fully textile-based, wearable garment capable of measuring brain activity with medical-grade precision [13]. The device, illustrated in Figure 1.1, addresses the most tedious aspects present in a complex PSG recording setup in terms of the excessive number of sensors, the impact on the patient's comfort during the night, and the associated economic expenses. This novel and cheap technology works in conjunction with a deep learning system that enables self-administered and large-scale automatic sleep monitoring in home environments under uncontrolled conditions. The ultimate objective of this platform extends beyond sleep disorder diagnosis to also perform closed-loop interventions aimed at rehabilitating lost functions and enhancing cognitive capabilities [14], [15].

(a) Polysomnography (PSG). (b) Wearable EEG device.

Figure 1.1: Illustration of the different configurations for sleep monitoring.

In this scenario, signals included in PSG, such as EOG, EMG or ECG, that are also used to classify sleep, are unavailable, making sleep evaluation harder and necessitating the training of specialized experts to operate in this novel environment. Additionally, these signals exhibit higher noise levels and artifacts compared to PSG recordings due to the simplified ecological setup, which supposes an additional challenge. In this sense, Bitbrain has made promising advancements in training supervised models on reduced labeled datasets recorded with its wearable device in laboratory settings, which include synchronized PSG monitoring used by technicians for sleep staging. As a result, expert-level accuracy in automatic sleep staging is achieved, effectively addressing some of the previously mentioned limitations.

Nevertheless, deep learning methods are notoriously data-hungry, which implies having large amounts of labeled data for effectively training them. The process of annotation is as resource-intensive as the manual scoring procedure carried by technicians, requiring up to 2 hours to analyze a complete 8-hour PSG recording [16]. As specified in [17], obtaining accurate annotations on physiological data is quite expensive, time-consuming, or simply impossible. Relying on human labels can also compromise performance since brain signals are inherently difficult to interpret and annotate, leading to a strong dependency on the labeler's subjective criteria and highlighting the issue of low inter-scorer agreement. Moreover, a significant part of the neuroscience research occurs in a low-labeled data regime, where large-scale annotated studies are rare in comparison with fields such as computer vision. This can also limit the performance of conventional deep learning, and therefore its success in real-world applications.

The mass adoption of home sleep monitoring enabled with the wearable device generates a massive amount of EEG data that is impossible to fully analyze and label by technicians. This results in enormous datasets that Bitbrain cannot leverage for supervised training (i.e., using annotations) of its models. Within this context, there exists the necessity of making use of these large home-recorded datasets without relying on expert annotations. As a consequence, the objective of this work is to develop and evaluate self-supervised learning (SSL) methods as a pre-training step to learn electroencephalogram representations from unlabeled data, exploiting the inherent structure of brain signals, and reducing the costs of the manual annotation process. SSL generates better-than-random initial parameters for the supervised training of the model, boosting its convergence and performance.

In this way, unlabeled signals have a significant role in the learning pipeline, addressing the data-hungry characteristic of deep learning by adding more data, fundamental in low-data regimes, while also mitigating the impact of low inter-scorer agreement and the ambiguity of labels. This will help determine the minimum amount of data that Bitbrain needs to label when incorporating SSL pre-training on unlabeled datasets, potentially reducing the elevated expenses of manual scoring.

Moreover, this poses an important transferability challenge, as pre-training is conducted under data coming from domestic conditions without experts oversight, and supervised training is carried on signals recorded in a controlled laboratory environment along with parallel PSG monitoring, employed by expert technicians for sleep classification. This may create a potential gap in data properties and the model's capability to generalize across both scenarios, highlighting the need for further investigation to better understand the variability and limitations between the aforementioned recording conditions, specially in the home settings, which represent the target scenario for the Bitbrain wearable EEG device.

To achieve this, state-of-the-art deep learning and self-supervised methods for automatic sleep staging will be reviewed, implemented, and evaluated across different scenarios, analyzing and understanding their capabilities and limitations within the context presented. These aspects will be detailed in the following chapters. Previously, a theoretical background will be provided to ensure a proper understanding of the work, covering the architecture and structure of sleep (Section 1.1), and the fundamentals of electroencephalography (Section 1.2).

1.1 Sleep architecture

Healthy sleep is typically characterized by a duration of 7-9 hours, consistent regularity, and the absence of significant disruptions. The structure of this period is divided into two primary phases: Non-Rapid Eye Movement (NREM) sleep and Rapid Eye Movement (REM) sleep, which alternate throughout the night. The first part of the night (*early sleep*) is characterized by slow-wave sleep (SWS), whereas REM sleep is prevalent during the second half (*late sleep*) [18].

The NREM phase was initially composed of four distinct stages (*Stages I, II, III, and IV*) according to the Rechtschaffen and Kales (R&K) standard [19], established in 1968. However, in 2007, the American Academy of Sleep Medicine (AASM) redefined NREM sleep as consisting of three stages (*N1, N2, and N3*) [20], resulting in the convention adopted in studies on automatic sleep classification [21] and subsequently in this project. Table 1.1 summarizes the different sleep stages along with their associated durations.

Sleep stage	Type of sleep	Typical length	Total
N1 (Stage I)	NREM	5-10 minutes	2-5%
N2 (Stage II)	NREM	20-25 minutes	50%
N3 (Stages III - IV)	NREM	20-40 minutes	20-50%
REM (Stage V)	REM	10-60 minutes	20-25%

Table 1.1: Sleep stages and associated durations.

Considering this structure, the classification of sleep is conducted by assigning one of these stages to each 30-second segment (*epoch*) of the patient's sleep recording. The graphical representation of the distribution and temporal evolution of the aforementioned sleep epochs throughout the night is referred to as a *hypnogram* (see Figure 1.2), which serves as a valuable tool for analyzing sleep patterns and diagnosing potential disorders.

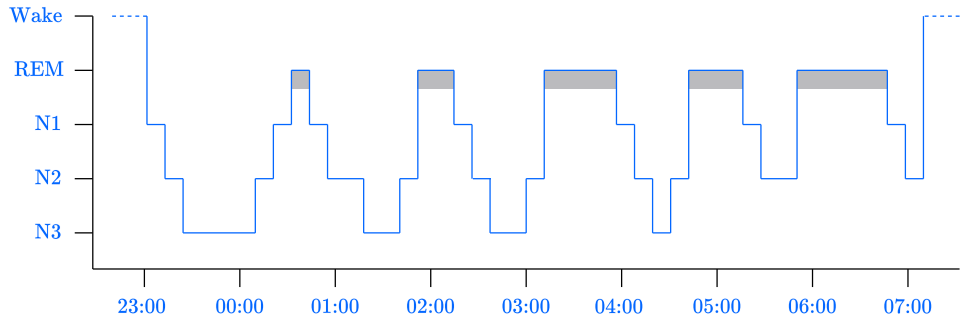


Figure 1.2: Hypnogram associated with the typical distribution of sleep stages during the night.

1.2 Electroencephalography (EEG)

Electroencephalography (EEG) is a non-invasive measurement of the electrical activity in the human brain. Scalp-placed electrodes record postsynaptic potentials resulting from slow currents subsequent to neurotransmitter release at axon terminal boutons [22]. The amplitude of these voltage signals is on the order of microvolts (μV), and reflects the potential differences between a recording electrode and a designated reference electrode. First introduced in 1929 [23], this technique has evolved into various forms and is now widely employed for diverse purposes, including diagnosing neurological disorders, monitoring brain function, and advancing scientific research. In addition, Subsection 1.2.1 explains the primary waveforms observed in these brain signals, while Subsection 1.2.2 addresses the fundamental factors influencing the configuration of an EEG device.

1.2.1 Frequency bands

Brain waves collected through EEG are measured in hertz (Hz), or cycles per second, and are classified based on their frequency range. The literature identifies the following four primary frequency bands. Delta bands (1-4 Hz) predominate during deep sleep, while theta waves (4-8 Hz) are involved in memory encoding and retrieval. Alpha bands (8-12 Hz) contribute to various motor and cognitive processes. Finally, beta waves (12-30 Hz) indicate cortical transmission and active concentration. Figure 1.3 illustrates the distinct waveforms that define EEG signals.

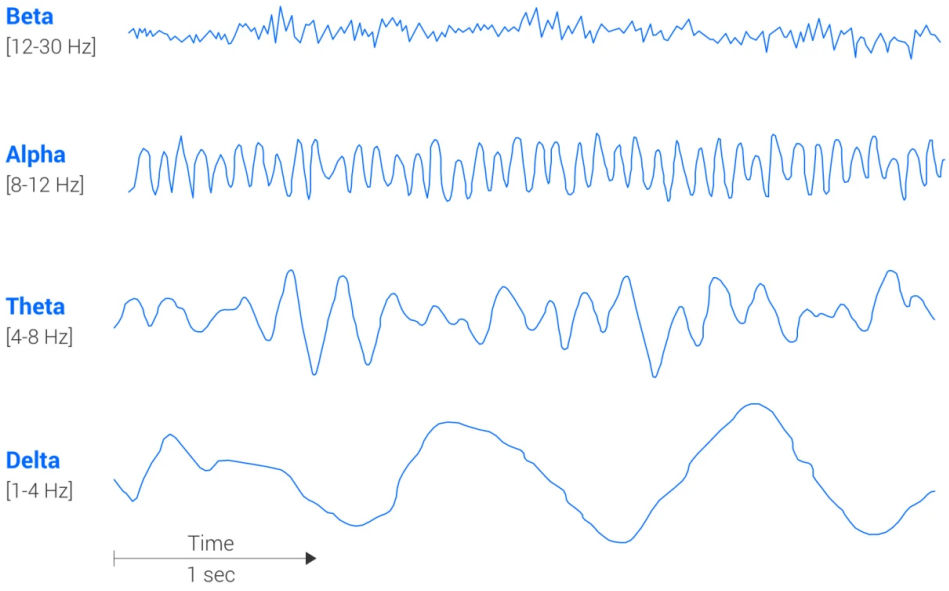


Figure 1.3: Classification of EEG waveforms based on their frequency band.

1.2.2 Technical features

A key consideration when analyzing a set of EEG signals is the technical specifications and parameters comprising the recording hardware. An EEG system can be categorized into three main areas: the acquisition layer, the sensor layer, and the connectivity layer. Firstly, the main component of the acquisition layer is the amplifier, which is responsible for three critical functions: accommodating, amplifying, and converting the analog electrical signals captured by the sensors into digital signals that can be processed by a computer. It comprises features such as the sampling rate (the number of times that the signal is measured per unit of time), resolution (the number of bits used to encode the analog EEG signal voltage values into discrete numbers), and input range (the maximum signal amplitude that can be recorded before saturation).

The sensor layer consists of scalp-placed electrodes that record electrical brain activity, serving as the sensor-body interface. For this component, the following aspects are worth noting:

- **Number of electrodes**: determines the amount of detailed information that can be measured from the brain, commonly ranging between 8 and 128. This number refers to the available *recording* electrodes, along with a *reference* electrode, essential for subtracting the common mode noise from recording electrodes, and a *ground* electrode, necessary for stabilizing the system and minimizing electrical interference.
- **Placement of sensors**: describes the location of scalp electrodes and the underlying area of the brain. It typically adheres to the International 10-20 system (see Figure 1.4), where 10 and 20 indicate that the distances between adjacent electrodes are either 10% or 20% of the total front-back or right-left length of the skull. In this sense, the 10-5, 10-10, and 10-20 systems are distinguished, ensuring a standardized testing protocol for reliable reproducibility and accurate analysis. The brain regions are categorized as pre-frontal (Fp), frontal (F), central (C), temporal (T), parietal (P), and occipital (O). Odd-numbered electrodes (1,3,5,7) correspond to electrodes placed on the left hemisphere, even-numbered electrodes (2,4,6,8) to those on the right hemisphere, and electrodes over the midline (zero line) are denoted by the letter z.

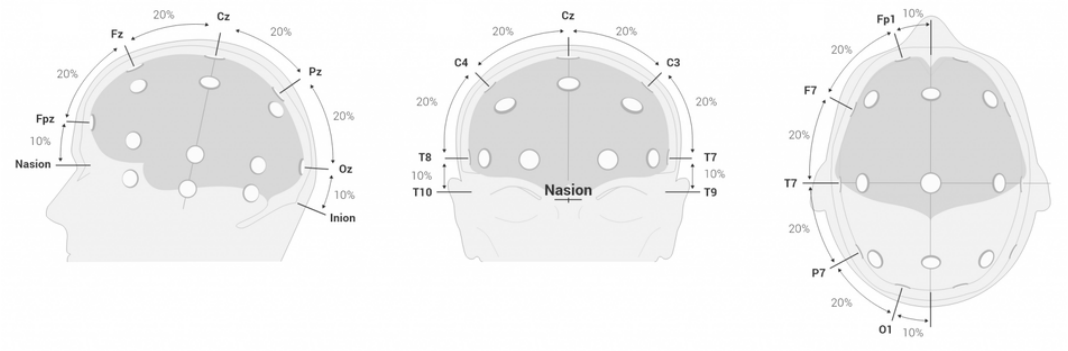


Figure 1.4: Placement of the standard electrodes of the 10-20 system.

- **Electrode contact**: depending on the contact with skin, EEG electrodes are classified into two main groups. Dry EEG electrodes establish direct contact with the scalp without requiring any electrolytic substance. In contrast, wet EEG electrodes require the application of an electrolytic medium (such as gels, saline solutions, or just tap water) to improve the contact impedance.

Finally, the connectivity layer defines additional features that are also critical for the type of experiment and the quality of EEG results. These include the power supply (cable-powered or battery-powered), the connectivity between the amplifier and the computer (wired or wireless), and the input/output interfaces and formats required for physiological sensors.

1.3 Objectives and scope of the project

The primary objective of this work is to develop and evaluate self-supervised learning (SSL) methods as a pre-training step to learn electroencephalogram representations from unlabeled data, thereby exploiting the inherent structure of brain signals to boost purely supervised learning performance and reduce the costs of the manual sleep staging process carried by technicians. To achieve this, the following specific objectives are established:

- Conduct a thorough analysis of the state-of-the-art techniques for automatic sleep staging, along with the study of general-purpose and sleep-specific self-supervised learning methods (Chapter 2).
- Design and develop a robust and efficient supervised deep learning model for automatic sleep scoring (Chapter 3, Section 3.1).
- Implement self-supervised learning techniques to pre-train the deep learning model on unlabeled data (Chapter 3, Section 3.2).
- Perform a comprehensive evaluation of the learning pipeline using Bitbrain datasets, recorded in both laboratory and home environments, with an emphasis on the transferability of learned representations and performance variation relative to the amount of available labeled data (Chapter 3, Section 3.3 and Chapter 4).
- Analyze and validate the results obtained from the evaluation pipeline, including an exploration of limitations and practical applications (Chapter 4).

Chapter 2

Related work

Artificial intelligence (AI) revolution is currently reshaping the world. Machine learning (ML), a key subfield of AI, enables learning and improving from experience with minimal human intervention, being able to replicate complex human processes with remarkable accuracy. This approach can be applied to the manual sleep staging procedure carried by technicians, which is both highly time-consuming and repetitive. Consequently, Section 2.1 reviews the state-of-the-art methods for automatic sleep scoring based on purely supervised frameworks, while Section 2.2 explores self-supervised learning techniques, which are capable of learning from unlabeled datasets, according to the context presented in the previous chapter.

2.1 Automatic sleep scoring

Two main different fields are distinguished within the automatic sleep scoring. Firstly, shallow learning, or traditional machine learning (Subsection 2.1.1), relies on hand-crafted feature extraction and classifier selection. In contrast, deep learning (Subsection 2.1.2) automatically obtains learned features from raw data through an end-to-end pipeline.

2.1.1 Shallow learning

Shallow learning begins with data preprocessing and filtering to obtain clear and reliable signals. Then, feature extraction is performed, followed by the selection of meaningful features that serve as input to the corresponding classifier. These typically include time-domain, frequency-domain, and time-frequency domain features [24]. Several works have employed classifiers such as Support Vector Machine (SVM) [25], Random Forest (RF) [26], K-means clustering [27], Bootstrap Aggregating [28], K-Nearest Neighbors (KNN) [29], and AdaBoost [30].

However, the feature extraction procedure requires a certain level of expertise, limiting its usability to trained specialists. Additionally, these methods are mainly tested under low-data conditions with high subject homogeneity, potentially limiting their performance in real-world applications. For this reasons, the following subsection examines the success of deep learning methods, which are able to automatically learn features from data and overcome many of the limitations of traditional machine learning techniques.

2.1.2 Deep learning

Deep learning systems rely on artificial neural networks (ANNs), which can be understood as complex mathematical models that learn to approximate functions in order to optimize a specific objective metric. They are primary defined by their architecture, which includes the number of neurons, the type of layers (i.e., the mathematical operations applied to data), the connection

between layers, as well as the activation and loss functions. As previously introduced, their main capability lies in learning representations from large volumes of raw data under heterogeneous conditions, making them particularly effective for sleep staging.

As reviewed in [31], the first attempts at deep learning for automatic sleep staging utilized simple networks with short input contexts of one to a few sleep epochs around a target epoch. This systems included architectures such as deep neural networks (DNNs) [32], [33], convolutional neural networks (CNNs) [34]–[36], and recurrent neural networks (RNNs) [37]. While these models successfully learned meaningful features from input signals, they were unable to capture long-range dependencies between sleep stages given the limited input context, fundamental due to the inherent slow transitions of the physiological processes underlying sleep stages.

In consequence, some works proposed models that encoded each epoch into an epoch-wise feature vector, and separately trained in a second stage to employ a long sequence of feature vectors preceding the target epoch to be classified, which is known as two-stages training. This procedure significantly increased performance [10], but presented some limitations. Firstly, the independent training of the components was suboptimal because the epoch encoder did not consider the subsequent sequential modeling. Secondly, the sequence classifier was limited to encode the left-side context of the target epoch for prediction, leading to lower overall accuracy.

Given these results, a set of architectures with a focus on long-term modeling raised within a common framework named sequence-to-sequence (see Figure 2.1). In this approach, the first part encodes the input epochs, typically raw signals or time-frequency images, into epoch-wise feature vectors (green dots). Subsequently, the sequence classifier generates a new sequence of vectors (red dots) that are employed to predict the labels corresponding to sleep stages. This end-to-end scheme overcomes the limitations of earlier proposals by jointly optimizing both components, enabling its interaction. Also, it is tasked to solve a sequence-to-sequence sleep staging problem, where epochs influence others depending on their position in the sequence. As a result, the method achieved an improvement in performance [38], [39].

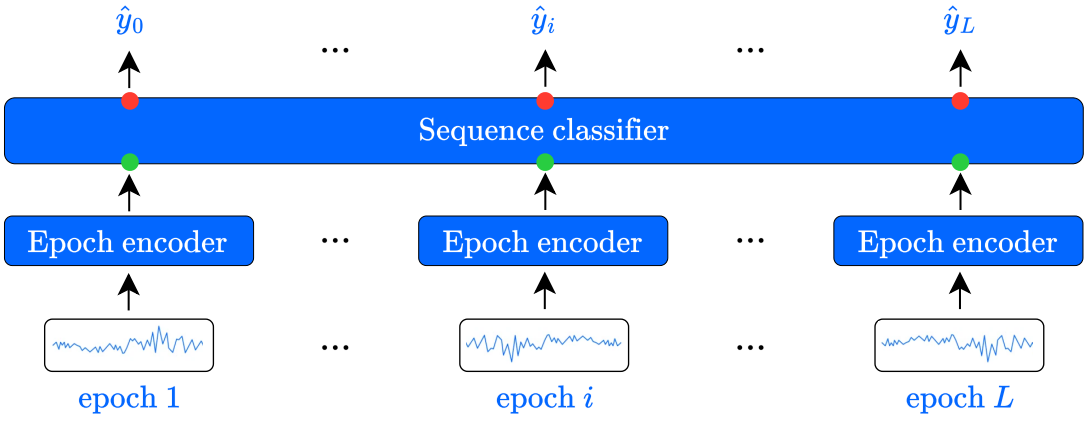


Figure 2.1: Sequence-to-sequence scheme for automatic sleep staging. Adapted from [31].

In recent years, there has been an increasing trend toward applying the described sequence-to-sequence scheme using different architectures, with the Transformer [40] standing out because of its notable success in natural language processing and computer vision fields with models such as BERT [41] and GPT [42].

Within this context, Bitbrain has made significant contributions to the literature by exploring and evaluating different neural architectures, including CNNs, RNNs, and Transformers [43], [44]; proposing a platform to enable closed-loop memory enhancement during sleep using automatic scoring [45]; investigating the effects of electrode setup, temporal scope, and population characteristics on the algorithm performance [46]; and publishing a complete EEG dataset [47].

To provide an overview of some relevant works in automatic sleep staging, Table 2.1 presents the model, year, architecture, and accuracy results in SleepEDF [48], [49], MASS [50], and SHHS [51], [52] datasets, respectively. The metric shown does not pretend to justify the superiority of one model over another, but rather provide a quantitative insight about the performance archived under the specific data and configurations in which they were evaluated.

Model	Year	Architecture	SleepEDF-20	SleepEDF-78	MASS	SHHS
DeepSleepNet [10]	2017	CNN + BiLSTM	82.0	-	86.2	-
SeqSleepNet [38]	2019	BiLSTM + BiLSTM	-	82.6	82.8	86.5
TinySleepNet [53]	2020	CNN + LSTM	85.4	83.1	87.5	-
XSleepNet [39]	2020	RNN + RNN	83.9	80.3	85.2	87.6
AttnSleep [54]	2021	CNN + Multi-Head Self-Attention	84.4	81.3	-	84.2
AT-BiLSTM [55]	2021	BiLSTM + Attention	83.78	-	-	-
SleepTransformer [11]	2022	Transformer	-	81.4	-	87.7
Cross-Modal Transformer [56]	2022	CNN + Transformer	-	84.0	-	-
L-SeqSleepNet [57]	2023	BiLSTM + Attention	86.1	-	-	88.4
SleepTNT [12]	2023	Transformer	-	83.1	-	-
SleepExpertNet [58]	2023	CNN + Transformer + BiLSTM	-	90.8	-	-
Coon and Ogg [59]	2024	CNN + Transformer	-	80.5	-	-
CTCNet [60]	2024	CNN + Transformer + CapsNet	86.2	82.5	-	85.7

Table 2.1: Overview of deep learning models for automatic sleep staging.

Nevertheless, as mentioned in the introduction, supervised deep learning techniques experiment some limitations related with their data-hungry nature. A significant portion of sleep-related research operates in low-data regimes, limiting the availability of signals. Annotating large datasets is both time-intensive and expensive, becoming impossible depending on the context. In addition, relying only on human labels introduces dependency on the technician's subjective criteria, as the inherent ambiguity of physiological signals often results in low inter-scorer agreement. These aspects degrade performance and potentially hinder their introduction to sleep therapies. To address these issues, the following section explores self-supervised learning methods, which are able to learn from massive amounts of unlabeled data, boosting traditional deep learning performance in low-data scenarios and reducing reliance on human annotations.

2.2 Self-supervised learning

Self-supervised learning (SSL) is an unsupervised learning methodology that learns representations from unlabeled datasets, exploiting the inherent structure of data [17]. As covered in [61], SSL comprises two main tasks: *pretext* and *downstream*. Firstly, the pretext task aims to generate better-than-random initial parameters and learn feature representations from the input data. These features capture general characteristics of the data rather than task-specific properties. Such representations are typically extracted from the last or near-final layers of the

pretext model (see Figure 2.2). There exists an intermediate task that slightly modifies the pretext model to be compatible with the task-specific problem (e.g., by adding a softmax function on top for classification purposes), transforming it into the downstream model.

On the other hand, the downstream task involves fine-tuning the downstream model employing the available labeled data. Some approaches train the entire model, while others freeze all but the last few layers. This downstream model presents the same architecture as if a fully supervised pipeline with random initial parameters were adopted. The key distinction lies in the initialization of the downstream model with the weights learned during the pretext task, which are derived from a significantly larger amount of data, providing a more informed starting point for training.

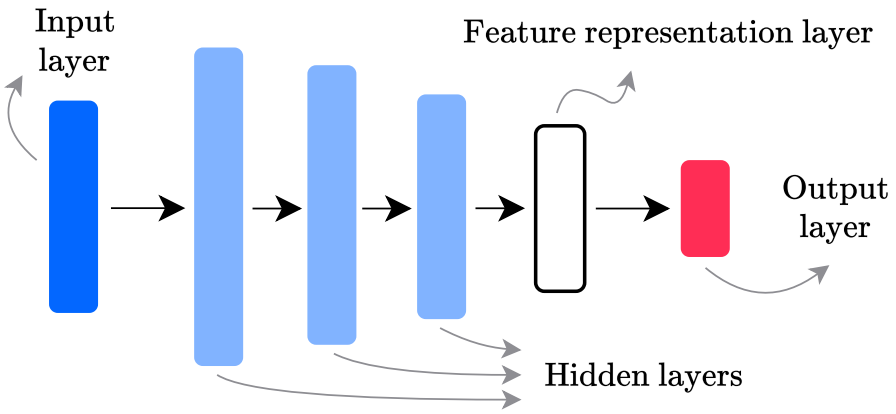


Figure 2.2: Neural network typical architecture. Adapted from [61].

The recent success of SSL in fields such as computer vision and natural language processing highlights its potential for application in signal processing, where much of the research occurs in low-data regimes. Therefore, it can reduce the bias inherent in smaller studies by pre-training models on larger repositories, benefiting from the general learned features across different conditions. Based on their functioning, SSL research methods can be categorized into two main paradigms: *contrastive learning techniques* (Subsection 2.2.1) and *masked prediction techniques* (Subsection 2.2.2) [62]. Additionally, Subsection 2.2.3 reviews the state-of-the-art of SSL within the automatic sleep staging domain.

2.2.1 Contrastive learning techniques

Contrastive self-supervised learning aims to identify the differences between distinct augmentations of input data. Data augmentation techniques involve applying slight modifications to data points without altering their semantic nature. Operations such as rotation, adding Gaussian noise or random color distortion are common within the computer vision field [63]. Two augmented versions of the same input data are treated as a contrastive pair, which the pretext model is tasked to identify. In this way, a repository of augmentations is first created, followed by model training using an appropriate loss function. The downstream model benefits from the general representations learned during this process through weight transfer, enhancing its performance on specific tasks.

The following contrastive learning frameworks represent some of the most significant contributions to this paradigm. SimCLR [63] learns representations by maximizing the agreement

between differently augmented views of the same data example via a contrastive loss in the latent space, using large batch sizes. BYOL [64] relies on two neural networks, one updated using a slow-moving average of the other (momentum encoder), trained to predict the same latent representation from two augmented views. MOCO [65] builds a dynamic dictionary with a queue and a moving-averaged encoder. SimSiam [66] explores siamese networks to maximize the similarity between two augmentations without requiring a momentum encoder but utilizing a stop-gradient operation. Barlow Twins [67] calculates the cross-correlation matrix between the embeddings of two augmented views, and tries to make this matrix close to the identity.

2.2.2 Masked prediction techniques

This line of research focuses on applying masking operations to data. It aims to extract strong features by training the model to reconstruct the masked portions and compare them with the original input. Within this definition, MAE [68] develops a ViT-based asymmetric autoencoder architecture, featuring an encoder that operates only on the non-masked subsets of pixels, along with a lightweight decoder that reconstructs the original image from the generated latent representations. Other relevant frameworks, such as BEiT [69] and Data2Vec [70], also build upon this approach.

Recent studies are exploring the combination of contrastive and masked methods, leveraging the strengths of both worlds. For example, CMAE [71] consists of two branches: an online branch as an asymmetric autoencoder that reconstructs the input images, and a momentum encoder that improves feature discriminability through contrastive learning. It is worth mentioning that other authors classify the SSL techniques into *contrastive*, which employ data augmentations, and *generative*, which rely exclusively on original data [61]. Inside this definition, it is also possible to distinguish between supervised and unsupervised approaches depending on the usage of pseudolabels.

2.2.3 Self-supervised learning for sleep staging

Most applications of SSL have focused on domains where large labeled repositories are available, resulting in relatively few studies published in the fields of biosignal and sleep [17]. On one hand, some studies adapt approaches originally developed for computer vision or natural language processing. On the other side, other authors propose domain-specific methods for time series, leveraging their properties to implement effective approaches.

With respect to the first group, authors brought the SimCLR framework to the sleep staging problem in [72], while an adaptation of BYOL was presented in [73]. BENDR [74], inspired by wave2vec 2.0 [75], leverages contrastive learning comparing reconstructed features generated through a Transformer, some of them masked, with the original features obtained from a previous convolutional encoder. MAEEG [76], influenced by the masked autoencoder architecture from [68], builds on the BENDR framework, adding two layers to map the Transformer output back to the original signal dimensions, but minimizing a reconstruction loss between the input signal and the reconstructed EEG. NeuroNet [77], which contains a Transformer autoencoder on top of a convolutional encoder, adopts an hybrid approach that combines contrastive learning and masked prediction tasks.

In the context of sleep-related approaches, ContraWR [78] applies contrastive learning by employing global representations from the datasets to differentiate signals associated with sleep stages. TS-TCC [79] combines efficient data augmentations with a temporal contrastive module

based on a cross-view prediction task and a contextual contrastive component that maximizes similarity between different contexts of the same sample. mulEEG [80] utilizes multiple views, including raw signals and time-frequency images, and incorporates a combination of diversity and contrastive losses. CoSleep [81] learns generalizable representations with a co-training scheme from multiples views (time and frequency) to mine positive samples, in conjunction with a queue and a momentum-based encoder to build a memory bank of negative samples.

A wide variety of different evaluation procedures is carried by these studies. The common denominator is to follow either a semi-supervised approach, combining SSL pre-training with fine-tuning on labeled data, or a linear evaluation process, where a linear classifier is trained on top of frozen representations extracted from the unlabeled data. Results also vary depending of the work, being common to find an increase in accuracy of 1% to 5% when incorporating SSL pre-training compared to a purely supervised pipeline. Nonetheless, this gain in performance typically diminishes when more labeled data is added to training, which specially highlights the capabilities of these methods under low-data conditions. Table 2.2 summarizes the accuracy results from various studies in the sleep staging field, detailing the method, year, evaluation type, and accuracy metric across various datasets, respectively. The first value represents the accuracy achieved with supervised training alone, while the second reflects the improvement obtained when including SSL pre-training. If only one value is shown, it corresponds to the latter case.

Method	Year	Evaluation	SleepEDF-20	SleepEDF-78	SHHS
Jiang, Xue, et al. [72]	2021	Semi-supervised	86.60 - 88.16	82.07 - 84.42	-
ContraWR [78]	2021	Linear	-	84.98 - 86.90	75.61 - 77.97
TS-TCC [79]	2021	Linear	83.41 - 83.00	-	-
CoSleep [81]	2021	Linear	71.60	-	-
BootstrapNet [73]	2021	Linear	80.80 - 85.50	-	-
mulEEG [80]	2022	Linear	-	79.08 - 78.06	82.62 - 81.21
NeuroNet [77]	2024	Linear	-	76.74	84.13
		Semi-supervised	-	85.24	86.88

Table 2.2: Overview of self-supervised learning techniques for sleep staging.

However, the approaches reviewed reveal some unexplored aspects. Firstly, none of these methods have been evaluated using a commercial device that could serve as the scalable solution for sleep studies. In addition, no studies have analyzed the effectiveness of self-supervised pre-training on home-recorded signals by end users, followed by a supervised training in controlled laboratory settings. Regarding results, there is a lack of clear understanding about the limitations of SSL concerning the available amount of data and, therefore, about its recommended applications depending on the context. As a consequence, some of the general and biosignal-specif SSL techniques will be implemented and evaluated in the following chapters, addressing the aforementioned aspects, among others.

Chapter 3

Methods

After reviewing the current state-of-the-art in sleep stage scoring and self-supervised learning, this chapter delves into the deep learning model architecture employed (Section 3.1), along with the SSL techniques developed (Section 3.2), aiming to address the challenges outlined in previous chapters. In addition, Section 3.3 provides a comprehensive description of the evaluation procedure carried to asses the effectiveness and robustness of the aforementioned methods across various scenarios.

3.1 Deep learning architecture

The deep learning model architecture, depicted in Figure 3.1, largely follows the sequence-to-sequence scheme presented in the previous chapter, and it is based on previous work [45], [46]. The input of the network comprises a window of L raw sleep epochs (i.e., 30-second segments), rather than time-frequency images, represented as a tensor of shape (L, *channels*, *samples*), where *channels* denotes the number of EEG channels, and *samples* refers to the number of sampled time points in the recording, which depends on the sampling frequency. The network outputs a vector of L labels, each corresponding to the sleep stage of the respective input epoch.

The network architecture consists of two main components. An epoch encoder f_θ performs feature extraction at epoch level, followed by a sequence encoder that transforms the entire encoded sequence leveraging temporal properties into logits, which are then utilized for sleep stage labeling. Additionally, the supervised parameter optimization is performed by stochastic gradient descent with the Adam optimizer ($lr = 0.0001,\ \beta_1 = 0.9,\ \beta_2 = 0.999,\ \epsilon = 1^{-8}$), which minimizes the cross-entropy loss between the output logits and the target labels.

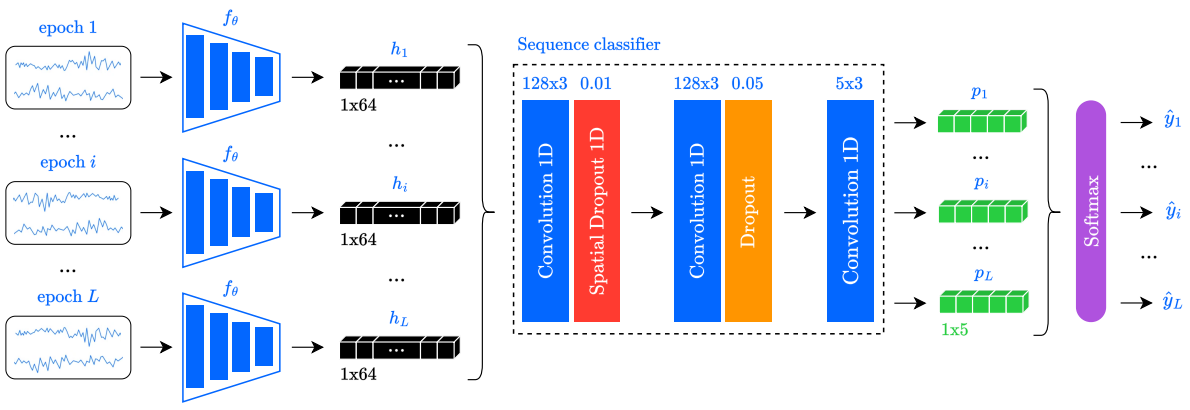


Figure 3.1: Deep learning model architecture.

More specifically, the epoch encoder sub-model f_θ follows a convolutional network architecture (see Figure 3.2). It receives a single sleep epoch with shape (*channels*, *samples*) as input, which is outputted as a 1D epoch-wise feature tensor h_i of size 64, with $i \in (1, ..., L)$. The network begins with 1D convolutions comprising 16 filters, kernel sizes of 5, strides of 1, and ReLU as the activation function, followed by spatial dropout and pooling layers. This structure is then repeated with kernel sizes of 3 and an increasing number of filters. The time axis is progressively reduced during the processing pipeline, while increasing the feature depth dimension of the data through the use of convolutional filters. A Global Max Pooling layer transforms the 2D structure of data into a 1D feature vector, after which a non-linear dense layer with 64 neurons is applied. The primary objective of this module is to perform robust, temporal-invariant feature extraction, capturing the intrinsic properties of the signal itself, regardless of when they occur within the night and without accounting for the context of other epochs in the input window.

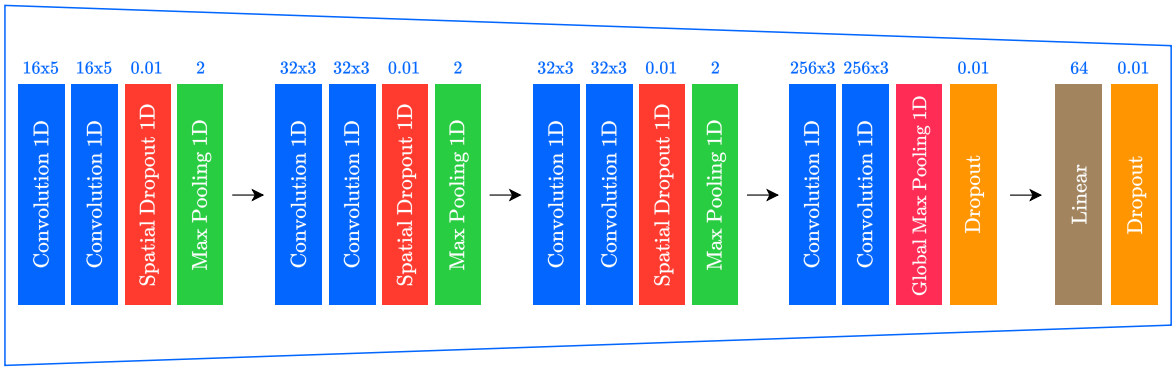


Figure 3.2: Epoch encoder sub-model architecture.

On the other side, the sequence classifier (see Figure 3.1) processes the sequence of encoded epochs $(h_1, ..., h_L)$ generated by the epoch encoder. It adopts a temporal convolutional architecture, consisting of two convolutional layers with 128 filters, kernel sizes of 3, no padding, and ReLU as the activation function, followed by dropout layers. A final convolution with 5 filters is performed to produce the sequence of logits vectors $(p_1, ..., p_L)$. The core function of this component is to model the dependencies between different sleep epochs, enabling a sequence-to-sequence sleep stage classification, and thereby imitating the approach carried by technicians during manual scoring to account for the inherent slow transitions of the physiological processes underlying sleep stages. Finally, a softmax operation is applied, along with a selection of the maximum value to determine the sleep labels $(\hat{y}_1, ..., \hat{y}_L)$.

3.2 Self-supervised learning methods

This section aims to describe the data augmentation techniques (Subsection 3.2.1) utilized in the contrastive self-supervised learning methods developed in Subsection 3.2.2. Furthermore, Subsection 3.2.3 elaborates on the SSL methods implemented within the masked prediction paradigm.

3.2.1 Data augmentation techniques

A data augmentation technique involves applying operations to data that slightly modify it without altering its semantic meaning. It is common to find transformations such as rotation, adding Gaussian noise or random color distortion within the computer vision field [63]. Biosignal processing includes specific augmentations designed for the time series format. In this work, two

different sets of transformations are employed, as summarized in Figure 3.3. On one hand, the first set of transformations T_1, inspired by [78], contains the following operations:

- **Bandpass filtering**: in order to reduce noise, first-order Butterworth filter is applied, using a frequency interval of $(1, 5)$ and $(30, 50)$.
- **Noising**: add high- and/or low-frequency noise to each channel. The high-frequency noise is sampled from an uniform distribution modulated by a noise degree value, following $high\_freq\_noise = d * a * uniform\_random\_seq$, where d is the noise degree, a is the amplitude range of the original signal, and $uniform\_random\_seq$ is an independent and identically distributed sequence with the same length as the original signal, generated from a uniform distribution in the range $(-1, 1)$. The low-frequency noise is obtained using the same equation but sampling the random noise sequence with 1% of the signal's length, followed by an interpolation to match the original length, converting the noise into a low-frequency component.
- **Channel flipping**: flip the EEG input channels as an augmentation method.
- **Time shifting**: horizontal rotation of the signal, which is split into two pieces and then resembled.

In addition, a second set of augmentations T_2, adapted from [72] and [73], includes these transformations:

- **Permutation**: the signal is randomly divided into $n \in [5, 20]$ segments of unequal length, shuffled and concatenated.
- **Crop and resize**: a random segment of length $m \in [0.25, 0.75]$ of the original signal is cropped. Then, a linear interpolation is performed to restore the signal to its original length.
- **Cutout and resize**: the signal is randomly divided into $n \in [5, 20]$ segments of unequal length, and one of them is discarded. The remaining segments are concatenated and linearly interpolated to match the original length.
- **Random masking**: the signal is randomly divided into $n \in [5, 20]$ segments of unequal length. Then, a proportion of $m \in [0.25, 0.75]$ of the total segments are selected and masked with zeros.

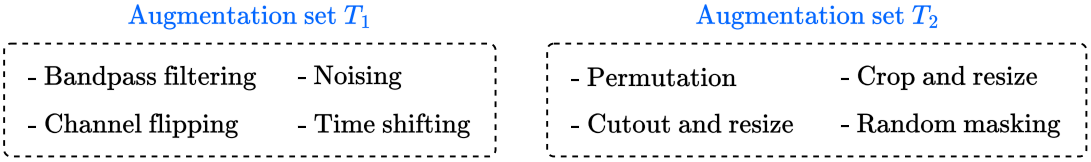


Figure 3.3: Overview of the augmentation sets T_1 and T_2.

These data augmentation techniques are individually applied to each EEG channel within a sleep epoch, except for the channel flipping operation from T_1, which inherently utilizes the existing channels. For a given set of augmentations T, a random operation $t \sim T$ is selected with equal probability to be applied to the input data.

3.2.2 Contrastive learning methods

This subsection delves into the contrastive self-supervised learning techniques that have been developed, which employ the data augmentations described in the previous subsection as a key component for training with unlabeled data. The first four approaches originate from the computer vision field, while the last one is specifically designed for the sleep staging paradigm.

SimCLR

Given a raw sleep epoch x, SimCLR [63] begins by sampling two data augmentations, $t \sim T$ and $t' \sim T$, to produce two augmented views $v = t(x)$ and $v' = t'(x)$ of the same example, which are considered a positive pair. Then, a neural base encoder f_θ, regarded as the *representation head*, outputs representation vectors $h_\theta = f_\theta(v)$ and $h'_\theta = f_\theta(v')$ from the augmented data examples. This network follows the epoch encoder architecture described in Section 3.1. A small neural *projection head* g_θ maps the representations to the latent space $z_\theta = g_\theta(h_\theta) = W^{(2)}\sigma(W^{(1)}h_\theta)$, being σ a ReLU non-linearity, and $z'_\theta = g_\theta(h'_\theta)$, where contrastive loss is applied after a final batch normalization. Once training is completed, the projection head g_θ is thrown away, using the encoder network f_θ and representation h_θ for downstream tasks. This is done because authors demonstrated that the hidden layer encoding h_θ before the projection head is a better representation than the layer after $z_\theta = g_\theta(h_\theta)$. This procedure is illustrated in Figure 3.4.

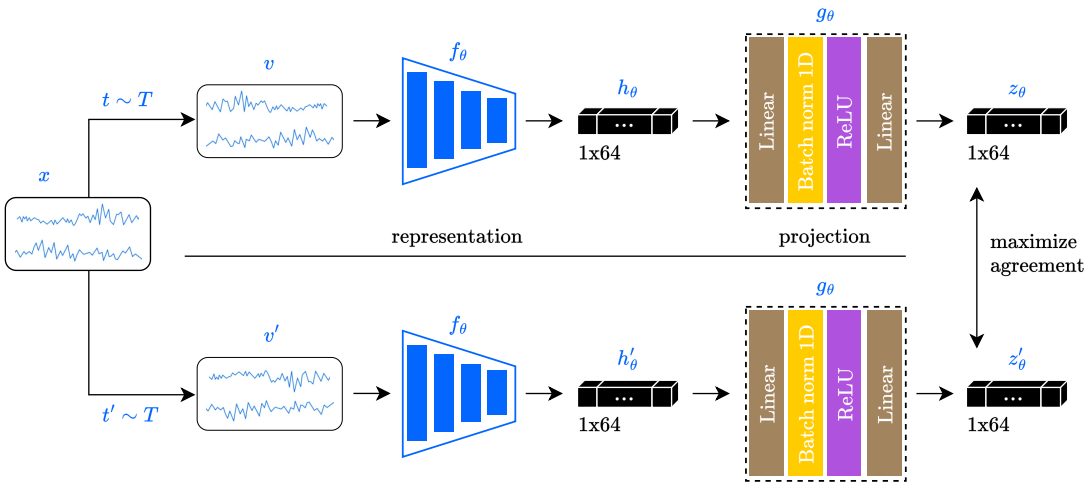


Figure 3.4: SimCLR framework overview.

The contrastive loss function is known as *NT-Xent* (the normalized temperature-scaled cross entropy loss). A random minibatch of N examples from a dataset D results in $2N$ data points after the data augmentation process. Given a positive pair, the other $2(N-1)$ augmented examples within the minibatch are treated as negative examples. Therefore, the loss function for a positive pair of examples (i, j) is defined in Equation 3.1:

$$\ell_{i,j} = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(z_i, z_k)/\tau)} \tag{3.1}$$

where $\text{sim}(u, v) = u^\top v / \|u\| \|v\|$ is the dot product between ℓ_2 normalized u and v (i.e., cosine similarity), $\mathbb{1}_{[k \neq i]} \in \{0, 1\}$ is an indicator function evaluating to 1 if $k \neq i$, and τ denotes a temperature parameter. The final loss is calculated across all positive pairs (i, j) and (j, i) along the minibatch.

BYOL

BYOL framework [64], shown in Figure 3.5, employs two neural networks: the *online* and *target* networks. As before, each receives an augmented view of the same input example x, denoted as $v = t(x)$ and $v' = t'(x)$, where $t \sim T$ and $t' \sim T$, respectively. Firstly, the online network, defined by the set of weights θ, follows the same architecture as SimCLR but includes an additional *prediction head* q_θ, outputting $q_\theta(z_\theta)$. On the other side, the target network also mirrors the structure of SimCLR, thereby matching the online network until the projection head, although utilizes separate parameters ξ that are an exponential moving average (EMA) of the online weights θ, updated after each training step following $\xi \leftarrow \tau\xi + (1-\tau)\theta$, where $\tau \in [0, 1]$ is the target decay rate. This makes the architecture asymmetric between the online and target pipelines.

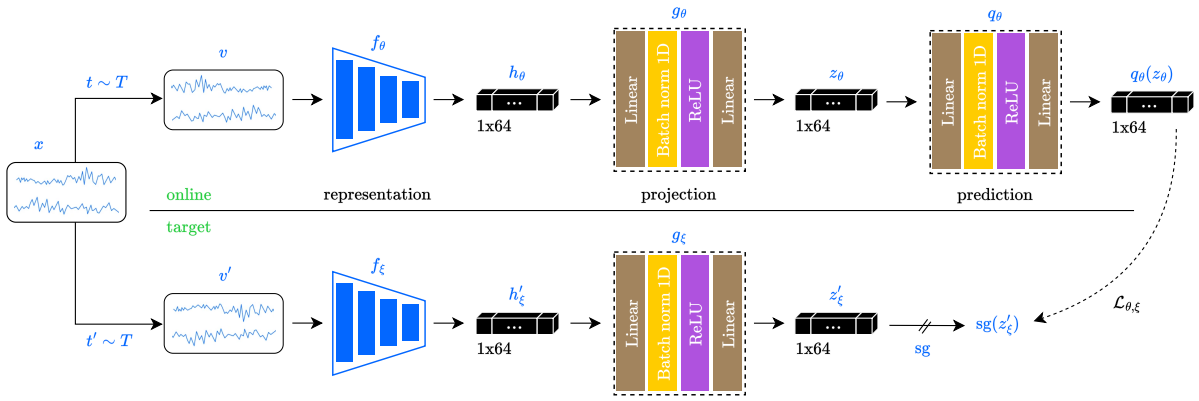


Figure 3.5: BYOL framework overview.

The online and target network outputs $q_\theta(z_\theta)$ and z'_ξ, respectively, are ℓ_2-normalized to $\overline{q_\theta}(z_\theta) = q_\theta(z_\theta)/\|q_\theta(z_\theta)\|_2$ and $\overline{z}'_\xi = z'_\xi/\|z'_\xi\|_2$. The loss function $\mathcal{L}_{\theta,\xi}$ follows Equation 3.2, minimizing the mean squared error between the normalized predictions and target projections. In addition, it is symmetrized by separately feeding v' to the online network and v to the target network to compute $\widetilde{\mathcal{L}}_{\theta,\xi}$, performing a stochastic optimization to minimize $L^{\text{BYOL}}_{\theta,\xi} = \mathcal{L}_{\theta,\xi} + \widetilde{\mathcal{L}}_{\theta,\xi}$ with respect to θ but not ξ, as indicated by the stop-gradient $\text{sg}(z'_\xi)$ in Figure 3.5. At the end of the training, the online encoder f_θ and representation h_θ are kept for downstream tasks. Note that this method does not rely on negative pairs, being focused on maximizing agreement between positive pairs, which eliminates the need for large batch sizes or memory banks.

$$\mathcal{L}_{\theta,\xi} = \left\|\overline{q_\theta}(z_\theta) - \overline{z}'_\xi\right\|_2^2 = 2 - 2 \cdot \frac{\langle q_\theta(z_\theta), z'_\xi \rangle}{\|q_\theta(z_\theta)\|_2 \cdot \|z'_\xi\|_2} \tag{3.2}$$

SimSiam

Similar to previous approaches, SimSiam [66] takes two randomly augmented views $v = t(x)$ and $v' = t'(x)$ from the same input example x, which are processed by the same encoder f_θ, producing $h_\theta = f_\theta(v)$ and $h'_\theta = f_\theta(v')$. Then, the projection head g_θ outputs the representations $z_\theta = g_\theta(h_\theta)$ and $z'_\theta = g_\theta(h'_\theta)$. Finally, a prediction head q_θ is applied to one of the two views, getting $q_\theta(z_\theta)$. This method can be thought of as BYOL without the momentum encoder, or as SimCLR without negatives pairs since it shares the weights between the two branches. It

is worth noting that the projection head contains a deeper architecture compared to previous techniques, as depicted in Figure 3.6. Additionally, the output of this head is always batch normalized, whereas the output of the prediction head is not.

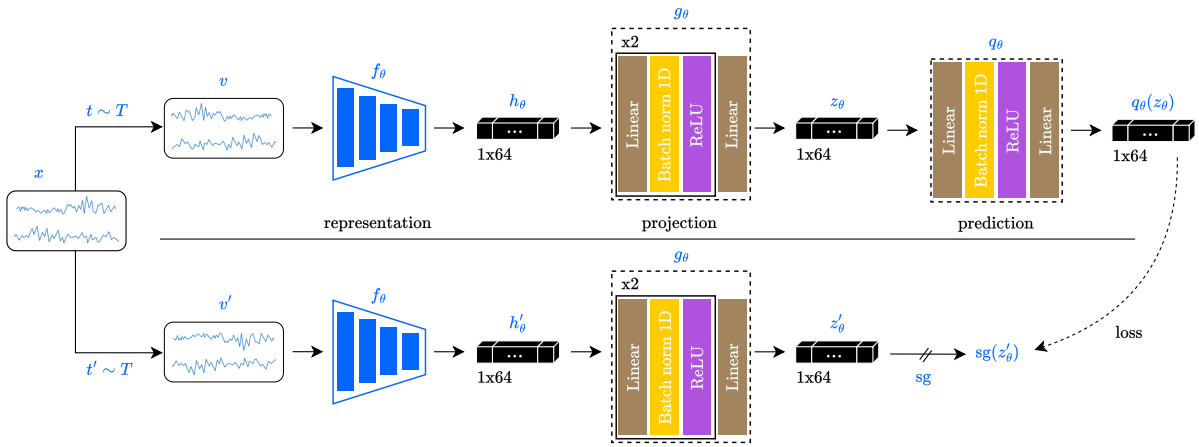


Figure 3.6: SimSiam framework overview.

The loss function minimizes the negative cosine similarity between the output vectors $q_\theta(z_\theta)$ and z'_θ, as illustrated in Equation 3.3. As previously performed by BYOL, it is symmetrized by separately feeding v' and v to both branches. The total final loss is averaged over all input samples within the minibatch, without relying on negative examples. It is important to note the stop-gradient operation applied to the second branch, which makes the network receive no gradient from z'_θ.

$$D(q_\theta(z_\theta), z'_\theta) = -\frac{q_\theta(z_\theta)}{\|q_\theta(z_\theta)\|_2} \cdot \frac{z'_\theta}{\|z'_\theta\|_2} \tag{3.3}$$

Barlow Twins

Barlow Twins [67] framework, depicted in Figure 3.7, follows the same pipeline than SimCLR, differing primarily in the depth of the projection head and the use of a distinct loss function, indicated in Equation 3.4:

$$\mathcal{L}_{\text{BT}} = \underbrace{\sum_i (1 - C_{ii})^2}_{\text{invariance term}} + \lambda \underbrace{\sum_i \sum_{j \neq i} C_{ij}^2}_{\text{redundancy reduction term}} \tag{3.4}$$

where λ is a positive constant trading off the importance of the first and second terms, and C is the cross-correlation matrix between the outputs of the two identical networks along the batch dimension:

$$C_{ij} = \frac{\sum_b z_{\theta,b,i} \cdot z'_{\theta,b,j}}{\sqrt{\sum_b \left(z_{\theta,b,i}\right)^2} \sqrt{\sum_b \left(z'_{\theta,b,j}\right)^2}} \tag{3.5}$$

where b indexes minibatch samples, and i, j index the vector dimension of the networks' outputs. C is a square matrix with dimensions equal to the feature size of the model, and comprises values ranging from -1 (i.e., perfect anti-correlation) to 1 (i.e., perfect correlation). The invariance term makes the embeddings invariant to the augmentations by driving the diagonal elements of the cross-correlation matrix C to 1. In contrast, the redundancy reduction term aims to decorrelate the different vector components of the embeddings by pushing the off-diagonal elements of C toward 0. As a result, this decorrelation minimizes redundancy between output units, ensuring they do not contain redundant information.

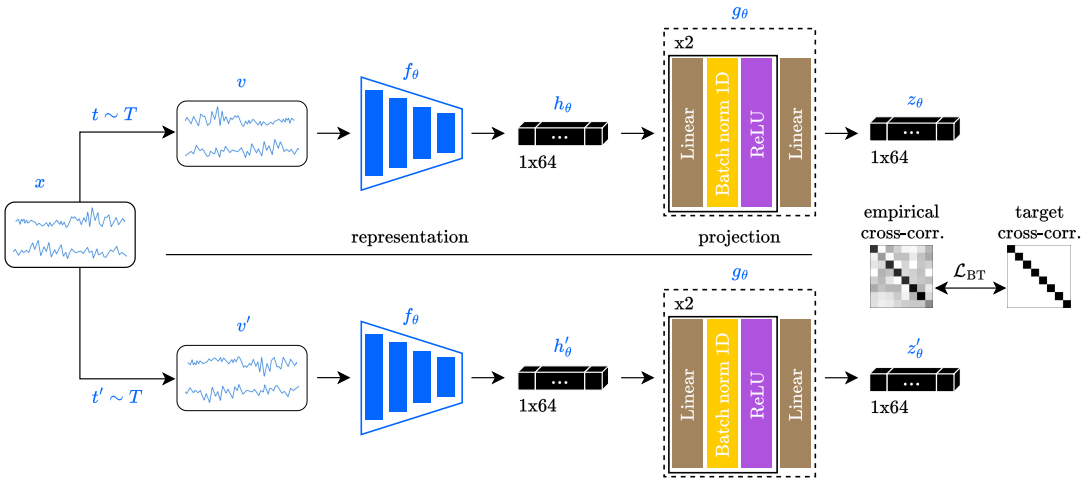


Figure 3.7: Barlow Twins framework overview.

ContraWR

ContraWR [78] introduces a specific framework for sleep staging, shown in Figure 3.8, though its application is not limited to this domain. In terms of network architecture, it follows a hybrid approach inspired by SimCLR and BYOL, featuring a symmetric structure between both branches with corresponding projection heads. It separates the parameters of the second network ξ, which are an exponential moving average of the weights from the first network θ.

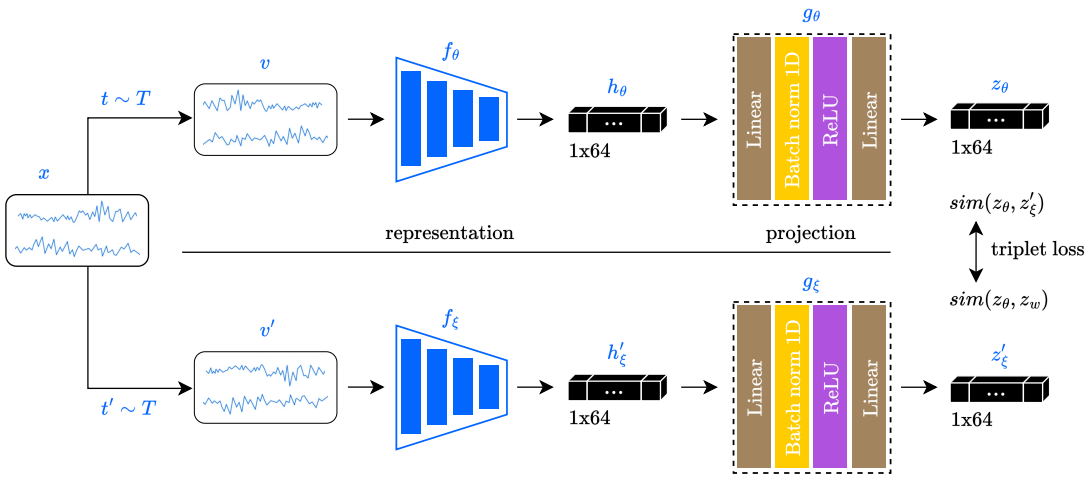


Figure 3.8: ContraWR framework overview.

The main contribution of this work is the loss function. It firstly proposes a world representation $z_w = \mathbb{E}_{k\sim p(\cdot)}[z_k]$ as the average representation of the dataset, being $k \sim p(\cdot)$ the sample distribution over the dataset, which is approximated by Monte Carlo method within each minibatch (i.e., using the average value over the batch $z_w = \frac{1}{M}\sum_{i=1}^{M} z_i$, where M is the batch size). Then, given two embeddings (z_i, z_k) and a hyperparameter σ, a Gaussian kernel is defined as the similarity measure:

$$\text{sim}(z_i, z_k) = \exp\left(-\frac{\|z_i - z_k\|^2}{2\sigma^2}\right) \tag{3.6}$$

Therefore, the triplet loss function follows Equation 3.7, which aims to establish stronger similarities between a positive pair (z_θ, z'_ξ) than the similarity among z_θ and the world representation z_w.

$$\mathcal{L}_{\theta,\xi} = \left[\text{sim}(z_\theta, z_w) + \delta - \text{sim}(z_\theta, z'_\xi)\right]_+ \tag{3.7}$$

where $\delta > 0$ is a hyperparameter indicating the empirical margin. Based on this idea, authors introduce a novel weighted averaged world representation by modifying the sampling distribution to be instance-specific. Consequently, for a given embedding z_θ, the instance-aware world representation becomes:

$$z_w = \mathbb{E}_{k\sim p(\cdot|z_\theta)}[z_k] = \frac{\mathbb{E}_{k\sim p}\left[\exp\left(\langle z_k, z_\theta\rangle/\tau\right)\cdot z_k\right]}{\mathbb{E}_{k\sim p}\left[\exp\left(\langle z_k, z_\theta\rangle/\tau\right)\right]} \tag{3.8}$$

where $p(\cdot|z_\theta)$ is the instance-aware sampling distribution of z_θ, and τ indicates a temperature parameter. As before, z_w is approximated by Monte Carlo sampling within the batch, and the triplet loss function also adheres to Equation 3.7, which is calculated for each embedding along the minibatch. This method does not symmetrize the loss as in BYOL, although it remains as a possible option. In this thesis, the contrast with instance-aware world representation version is implemented as the definitive approach of the ContraWR framework.

3.2.3 Masked prediction methods

This subsection details the masked prediction techniques that have been implemented, which rely on applying masking operations to unlabeled data instead of data augmentations used in contrastive learning by previous methods.

BENDR

BENDR [74] largely follows the architecture of wave2vec 2.0 [75], and is comprised of two different stages (see Figure 3.9). Firstly, a convolutional encoder f'_θ downsamples the input data into a new sequence of vectors referred to as BENDR, enlarging the feature dimension while reducing the time axis. This sub-network matches the encoder architecture proposed earlier (see Figure 3.2) up to the last convolutional layer before the Global Max Pooling operation, aiming to generate the required 2D output structure with feature depth instead of a one-dimensional vector outputted by default (note the notation f'_θ instead of f_θ). Therefore, the time axis is downsampled through the stride of convolutional layers and pooling operations, whereas the feature dimension is defined by the number of filters of the last convolution (256 in this implementation).

On the other side, the second component consists of a Transformer encoder that mirrors the standard implementation of Vaswani et al. [40], but without internal batch normalization layers and using a weight initialization scheme known as T-Fixup [82]. This module comprises $N = 8$ layers, each with 8 heads, defining a model dimension of the feature depth multiplied by a factor of 3 ($256 * 3$), which is obtained through a linear transformation. As wave2vec 2.0, GELU activations are employed, along with LayerDrop and Dropout with probabilities 0.01 and 0.15, respectively. Position information is encoded using an additive grouped convolution layer with a receptive field of 25 and 16 groups before upscaling the feature dimension. Finally, a linear layer maps the Transformer output (contextual features) back to the BENDR feature size.

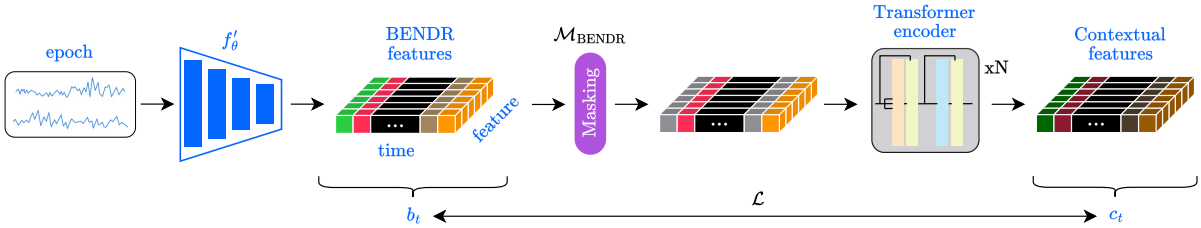


Figure 3.9: BENDR framework overview.

Regarding the learning procedure, this method aims to reconstruct the masked positions in the latent space by applying the NT-Xent contrastive loss between BENDR features (encoder output) and contextual features (Transformer output). Therefore, given a position t within the time dimension, the loss is defined as:

$$\mathcal{L} = -\log \frac{\exp(\text{sim}(c_t, b_t)/\tau)}{\sum_{b_i \in B_D} \exp(\text{sim}(c_t, b_i)/\tau)} \tag{3.9}$$

where c_t is the contextual vector outputted from the Transformer at position t, b_i is the BENDR vector at some index i, B_D is a set of 20 uniformly selected distractors from the same sequence (plus b_t), sim is the cosine similarity, and τ denotes a temperature parameter. Consequently, this function tries to adjust the Transformer output at position t to be most similar to the encoded representation by f'_θ at t, despite of being masked. In addition, the mean squared activation of the BENDR features is added to the loss with a weight of 1. Although authors state that the loss is applied only to masked positions, the implementation provided computes $\mathcal{L}$ for each b_t within the input sequence.

The masking logic $\mathcal{M}_{\text{BENDR}}$ replaces contiguous sections of size 10 with probability $p_{mask} = 0.065$, allowing overlap, with a single learned mask vector of the same length as the feature dimension depth. This occurs right after the encoder outputs the BENDR features, but before relative positioning and linear upscaling, respectively. The number of negatives in B_D for each b_t is set to 20 and is uniformly sampled from the same input sequence as b_t. As a result, the layers of the encoder f_θ that correspond to those in f'_θ are initialized with their pre-trained weights, thereby employing the learned BENDR representations for downstream tasks.

MAEEG

MAEEG [76], depicted in Figure 3.10, builds upon the previous work but adapts a masked autoencoder scheme from computer vision to EEG data. This framework learns representations

through reconstruction loss between input and output EEG signals. To achieve this, the architecture simply introduces two layers at the end of the Transformer to map contextual features back to the raw EEG input dimensions, enabling both temporal and spatial reconstruction. The linear layer that previously reduced the contextual feature dimension to the BENDR feature size after the Transformer is removed. Then, the reconstruction loss is calculated by comparing the reconstructed EEG $\hat{x}$ and the input EEG x along the channel dimension, defined as:

$$\mathcal{L} = 1 - \frac{\hat{x} \cdot x}{\|\hat{x}\|\|x\|} \tag{3.10}$$

Consequently, while BENDR uses contrastive learning between convolved and contextual features, MAEEG learns representations by minimizing a reconstruction loss on the raw signal space, mainly reconstructing the mean of the short-timescale signals and capturing the long-timescale fluctuations.

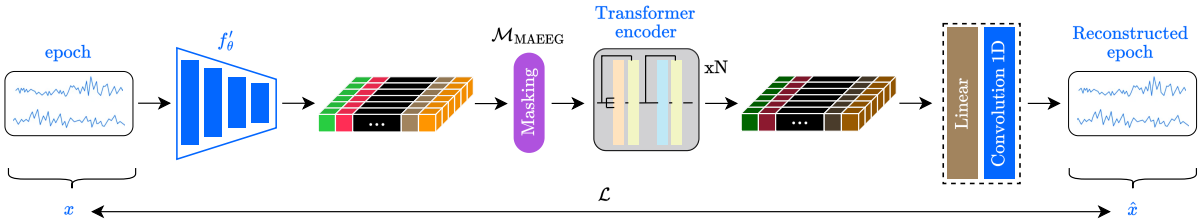


Figure 3.10: MAEEG framework overview.

The MAEEG masking strategy $\mathcal{M}_{\text{MAEEG}}$ differs from the BENDR masking $\mathcal{M}_{\text{BENDR}}$. Authors in [76] state that $\mathcal{M}_{\text{BENDR}}$ can vary a lot because in some cases most tokens are masked with overlap, while in others no tokens are masked at all, which can cause instability in representation learning. To address this, a more systematic way for generating the mask is introduced, shown in Figure 3.11, by defining the overall mask size using a rate value relative to the total sequence size, and a second parameter that corresponds to the number of mask chunks (i.e., the number of contiguous masked segments without overlap). In this sense, authors found that masking a high percentage of tokens (e.g., 75%) performed better than masking a smaller rate (e.g., 25%), and that models with a single chunk outperformed those with multiple chunks. Consequently, one chunk with 75% of the tokens masked is employed within this thesis.

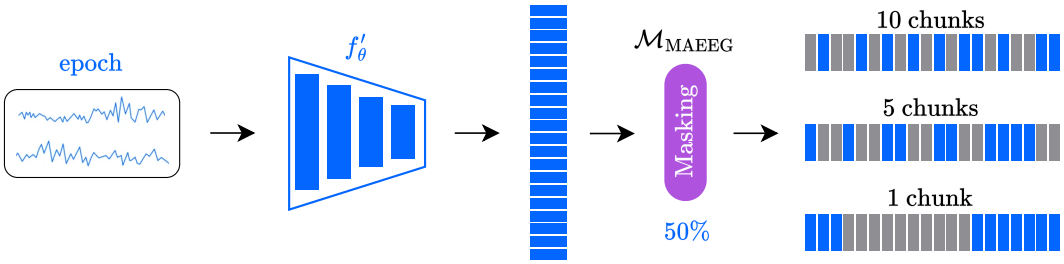


Figure 3.11: MAEEG masking logic. Adapted from [76].

3.3 Evaluation

In this section, the datasets employed within this work are described in Subsection 3.3.1, whereas Subsection 3.3.2 explains the evaluation pipeline, including the corresponding tests and their implications in the presented context.

3.3.1 Datasets

The datasets used for evaluating automatic sleep staging are as follows:

- **Bitbrain Open Access Sleep (BOAS) dataset** [47]: the dataset comprises 128 nights in which healthy participants were monitored simultaneously using both PSG and the wearable EEG headband presented in Chapter 1, enabling direct comparison between the two devices. It includes sleep labels obtained from three different sleep scorers who annotated the PSG recordings following AASM guidelines [20]. A consensus label was derived from these annotations by a fourth expert to ensure robust and reliable sleep staging given the variability in the classification process across distinct scorers, which is reflected by the low inter-scorer threshold, established around 82% [6], [7]. These consensus labels were also applied to the corresponding wearable EEG recordings, leveraging the simultaneous data acquisition. Within the PSG data, EEG (F3, F4, C3, C4, O1, and O2 channels), electrooculogram (EOG), electromyography (EMG), breathing activity, respiratory airflow, and photoplethysmographic activity recordings are available.

  The headband data includes EEG recordings through two frontal electrodes positioned at measuring locations equivalent to the AF7 and AF8 channels, motion via an accelerometer and gyroscope, and pulse using a PPG sensor. For this work, EEG frontal signals sampled at 256 Hz and recorded by the wearable headband (AF7 and AF8) are selected to accomplish the requirements outlined in previous sections, focusing on the exploration of emerging wearable EEG technologies.

- **HOGAR dataset** [83]: novel dataset acquired by the Bitbrain team as part of a research project aimed at developing tools to improve the diagnosis and early treatment of aging-related diseases. The primary objective of the dataset is the validation of a home-use instrument for quantifying cognitive function in a population at risk of dementia. One of the key distinction with the previous dataset lies in the recording setup, which involves home environments under uncontrolled conditions where users completely manage device configuration during the night. To date, it contains 239 recordings sampled at 256 Hz and collected though the same wearable EEG headband at measuring locations equivalent to the AF7 and AF8 channels.

  In this scenario, signals such as EOG or EMG are not acquired due to the lack of simultaneous PSG monitoring. Sleep technicians require this information to effectively perform manual sleep staging, thereby necessitating the complex training of specific experts to operate in this alternative environment where only the headband signals are accessible. In addition, these recordings exhibit higher noise levels and artifacts due to the simplified ecological setup and the absence of professional supervision when users self-place the wearable device or do so with the assistance of a family member. Therefore, the dataset consists of a collection of unlabeled data that continues to increase over time, especially with the massive adoption of emerging wearable EEG devices that is currently taking place.

As a consequence, the first dataset serves as a rich source of reliable labeled data recorded under controlled conditions, while the second provides a substantial volume of unlabeled recordings suitable for representation learning. It is worth mentioning that the sampling frequency is downsampled from 256 to 128 Hz for both datasets in order to reduce the model's input size, alleviating storage and processing requirements, along with the application of a band-pass filter ranging from 0.5 to 45 Hz with a view to compensate the low signal-to-noise ratio (SNR).

An artifact detection system, developed by Bitbrain, further filters the EEG signals, identifies non-numerical values, detects flat signal segments, flags high-amplitude noise, and analyzes noise in specific frequency ranges (see Appendix B for more details). The thresholds for the different detectors were estimated empirically with the goal of optimizing artifact removal to maximize model's performance while minimizing the amount of discarded data. As a result, robust and clear signal quality is ensured for downstream processing. The integration of these datasets, summarized in Figure 3.12, into the learning pipeline is detailed in the following subsection.

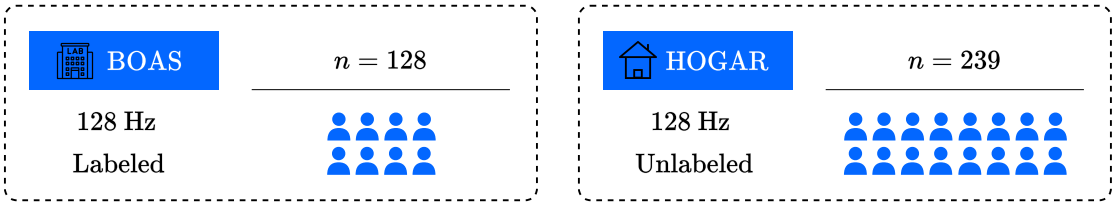


Figure 3.12: Datasets available for evaluating automatic sleep staging.

3.3.2 Performance

The deep learning pipeline is illustrated in Figure 3.13, and consists of three distinct stages. Firstly, as previously described, the data preprocessing module resamples and reshapes the raw EEG signals from the datasets introduced in the prior subsection. Each recording is additionally reshaped into a 3D matrix containing the number of 30-second epochs, the number of time points per epoch ($30 * 128$ in these specific datasets), and the number of EEG channels (AF7 and AF8). Depending on the derived test, a certain data partition is performed to feed the subsequent stages of the pipeline.

The next step is defined by the pretext task, which involves Z-scoring the input data followed by a self-supervised learning module that extracts feature representations from unlabeled signals using one of the SSL techniques from Section 3.2. Therefore, this part aims to generate better-than-random initial parameters that boost model convergence and performance in subsequent tasks.

Finally, the third step consisting of the downstream task involves supervised learning for sleep classification. To achieve this, the labeled input data is divided into training, validation and test sets, whose percentages are defined by the specific test scenario (see Figure 3.15). Also, a Z-score is applied using only the training data partition. Supervised learning is subsequently performed, including the possibility of initializing the feature encoder f_θ with the weights learned during the pretext task, providing a more informed starting point for training. The accuracy metric is employed as the primary performance indicator, consistent with many studies in the literature [78]–[81], although additional metrics such as precision, recall and F1-score can also be calculated as part of the classification report.

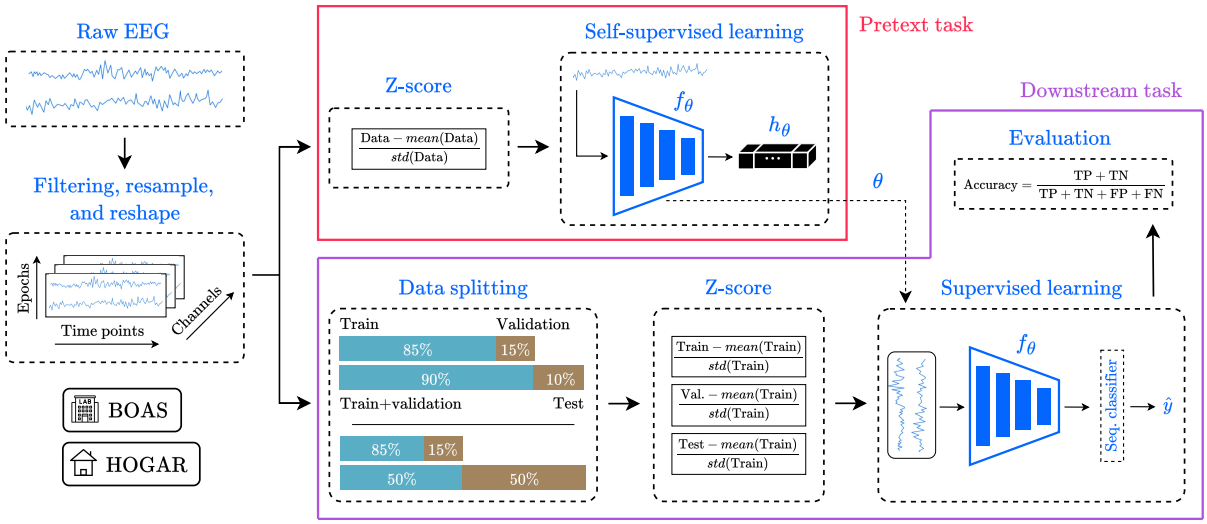


Figure 3.13: Deep learning pipeline.

Two different self-supervised evaluation procedures, depicted in Figure 3.14, are commonly distinguished within the described pipeline. On the one hand, a semi-supervised approach consists of pre-training the feature encoder (Figure 3.2) on unlabeled data, followed by fine-tuning the entire model (Figure 3.1), or just some specific layers, with available labeled data. This methodology aims to refine the general representations obtained during pre-training with task-specific properties (i.e., sleep-related electroencephalographic hallmarks), leveraging the strengths of both learning steps to improve model performance, particularly under low-data conditions. It represents the practical application of self-supervised learning, which complements supervised learning to optimize model weights and surpass the performance of a purely supervised pipeline.

On the other side, a linear evaluation approach involves training a linear classifier (instead of a sequence classifier) on top of the frozen encoder, which was pre-trained on unlabeled data. The objective of this evaluation is to assess the linear separability of the general representations learned during the self-supervised step when applied to the downstream task. It serves as a theoretical measure of the feature quality, indicating the adaptability of the representations to the task-specific domain. It is widely adopted in the computer vision field to compare the effectiveness of the features learned by SSL methods [63]–[67].

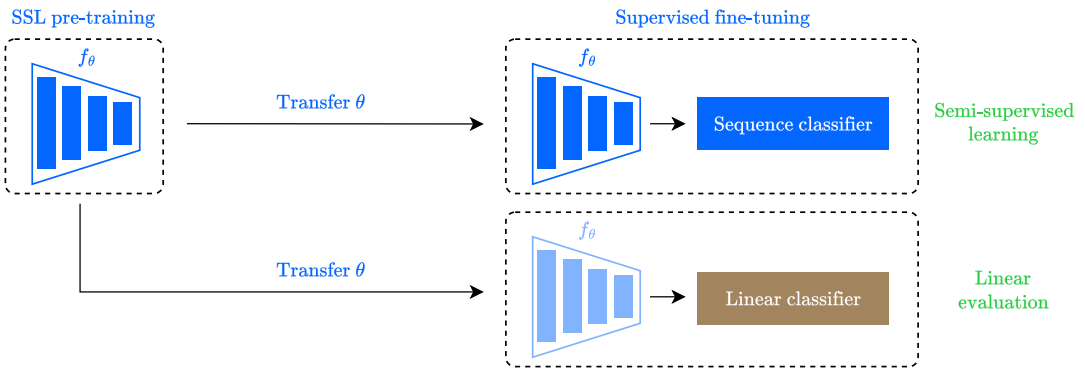
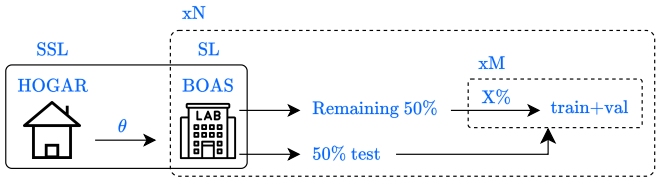


Figure 3.14: Self-supervised learning evaluation methodologies.

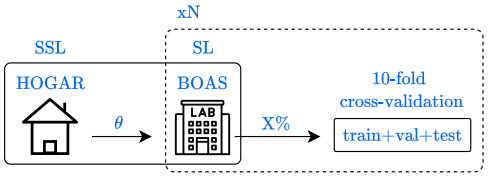
The different evaluation scenarios proposed for automatic sleep staging, following the aforementioned deep leaning pipeline, are shown in Figure 3.15. The first scenario (Figure 3.15a) involves learning general features with SSL on the entire HOGAR dataset. Subsequently, the BOAS dataset is split into a large, constant test set representing the 50% of the total data. The remaining half is used for training and validation, whose recordings are incrementally selected for supervised training. This approach ensures a constant and representative test set in which the model can be properly evaluated, regardless of the proportion of training and validation data used. In order to obtain consistent results, the test partition is randomly selected N times, and the percentage of training data is repeated M times within the corresponding partition.

The next scenario is illustrated in Figure 3.15b, and consists of the same pre-training step on the whole HOGAR dataset but followed by supervised training using a partition X to perform a 10-fold cross-validation, which already contains the training, validation, and test sets. Unlike the previous scenario, this procedure introduces higher variability in the test set as the number of subjects and their evaluation depend on the percentage X. However, it represents a more accurate scenario where datasets with varying sizes are individually evaluated, rather than relying on a fixed test set, which may not represent the practical application of sleep-related research studies. As before, to ensure consistent results, the partition X is randomly selected N times.

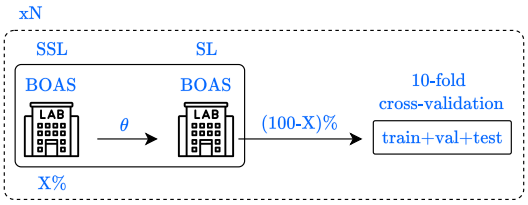
Finally, the last scenario involves both self-supervised pre-training and supervised training performed entirely on the BOAS dataset (see Figure 3.15c). This approach selects a percentage of the available data X for SSL, employing the remaining recordings for supervised training through a 10-fold cross-validation procedure. In this way, an evaluation of the learned representations during the pretext task is derived within the same datasets, enabling a direct comparison with the home-recorded signals that comprise the HOGAR dataset. To ensure consistency, this scenario is also repeated N times, selecting a different data partition for each run.



(a) HOGAR → BOAS scenario with a large and constant test set.



(b) HOGAR → BOAS scenario with a 10-fold cross-validation evaluation procedure.



(c) BOAS → BOAS scenario with a 10-fold cross-validation evaluation procedure.

Figure 3.15: Evaluation scenarios for automatic sleep staging.

All things considered, the general deep learning pipeline presented (Figure 3.13) can be semi-supervised or linearly evaluated (Figure 3.14), following a data partition scheme defined by the corresponding testing scenario (Figure 3.15). These configurations are designed to provide a complete set of tests that addresses the following aspects, among others, under distinct conditions:

- Assessing the effectiveness of the transferability between learned SSL representations from recordings collected in domestic conditions by end users (HOGAR), and labeled signals acquired in a controlled laboratory environment (BOAS). The quality of the features learned from each individual dataset can also be further analyzed.
- Providing a clear understanding of the limitations of SSL concerning the available amount of data is essential, as it helps determine the minimum amount of labeled data required to achieve medical-grade sleep staging accuracy. In this sense, the potential of SSL pretraining to compensate the data-hungry nature of deep learning can also be examined regarding the available data, crucial in low-data regimes.

Consequently, this approach could significantly reduce the costs associated with the manual annotation process and mitigate the impact of low inter-scorer agreement contributing to label variability. Evaluating Bitbrain's novel device within the context presented will help bridge the gap between gold-standard clinical sleep monitoring and emerging wearable EEG technologies, advancing the development of a scalable solution for people requiring medical attention.

Table 3.1 summarizes the suggested tests for evaluating the learning pipeline, which are categorized into semi-supervised and linear evaluation approaches. The results obtained within each configuration will be compared against a purely supervised procedure to provide a comprehensive understanding of the benefits of SSL. Regarding the available computational resources, tests *SEMI03* and *LINEV03* are conducted exclusively with contrastive learning methods due to the demanding execution time required by masked prediction techniques.

Regarding the hyperparameters, Appendix C addressed an ablation study to determine the configuration followed in each contrastive method. The transformation set T_1 is utilized across all contrastive approaches, with the exception of BYOL framework, which demonstrated superior performance when employing the augmentation set T_2. For masked prediction techniques, the proposed values in their respective works have been adopted due to the extensive execution time required, with a few exceptions: Transformer encoder depth (reduced to 4 instead of 8), batch size (set to 64), Adam optimizer ($lr = 0.0001,\ \lambda = 1^{-4}$), and the number of epochs (established to 200).

Beyond numerical insights, the quality of the feature representations will be additionally examined by projecting them onto the 2D space using t-SNE [84] and UMAP [85], two widely used tools for dimensionality reduction and feature exploration that enable the identification of patterns, clusters, and relationships within the data (see Figure 3.16). They aim to provide a deeper understanding of the learned representations and their separability in the feature space.

Id	SSL data	SL data	Evaluation	Motivation
Semi-supervised				
SEMI00	HOGAR	BOAS	Constant test set (50%)	Study performance when increasing BOAS labeled data using HOGAR SSL representations + transferability analysis in a constant test set
SEMI01	HOGAR	BOAS	10-fold cross-validation	Study performance when increasing BOAS labeled data using HOGAR SSL representations + transferability analysis in a 10-fold CV evaluation
SEMI02	BOAS	BOAS	10-fold cross-validation	Study performance with complementary SSL and SL data partitions within the same dataset (BOAS) in a 10-fold CV evaluation
SEMI03	HOGAR	BOAS	10-fold cross-validation	Equal to *SEMI01* but using the same amount of pre-training data as *SEMI02* to compare the representations learned between datasets
Linear evaluation				
LINEV00	HOGAR	BOAS	Constant test set (50%)	Study linear performance when increasing BOAS data using HOGAR SSL representations + transferability analysis in a constant test set
LINEV01	HOGAR	BOAS	10-fold cross-validation	Study linear performance when increasing BOAS data using HOGAR SSL representations + transferability analysis in a 10-fold CV evaluation
LINEV02	BOAS	BOAS	10-fold cross-validation	Study linear performance with complementary SSL and SL data partitions within the same dataset (BOAS) in a 10-fold CV evaluation
LINEV03	HOGAR	BOAS	10-fold cross-validation	Equal to *SEMI03* but following a linear evaluation procedure to compare the general, frozen representations learned between both datasets

Table 3.1: Overview of evaluation tests.

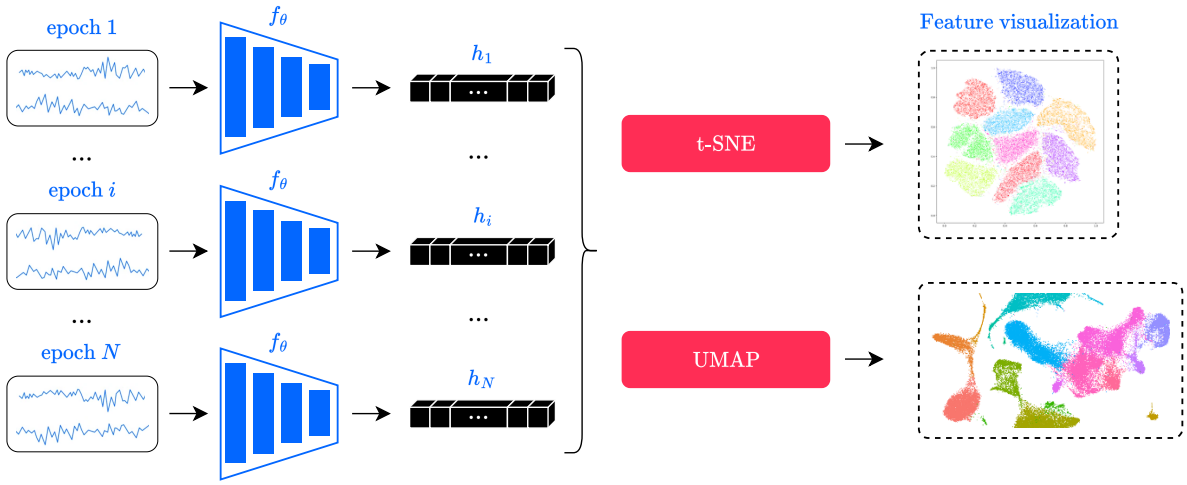


Figure 3.16: Feature visualization pipeline using t-SNE [84] and UMAP [85] tools.

Chapter 4

Results and discussion

This chapter delves into the results of semi-supervised learning in Section 4.1 and linear evaluation procedure in Section 4.2, following the evaluation pipeline stated in previous sections. Additionally, Section 4.3 analyzes the feature visualization plots generated through t-SNE and UMAP to provide a better understanding of the learned representations in the feature space.

4.1 Semi-supervised learning

The first classification report corresponds to test *SEMI00*, shown in Table 4.1 and evaluated in the scenario depicted in Figure 3.15a, where the amount of labeled data used from the BOAS dataset is progressively increased. The first row corresponds to training the model only on the percentage of labeled data indicated, without SSL pre-training, while subsequent rows indicate the SSL method employed for pre-training on the entire HOGAR dataset, followed by the supervised learning step (i.e., executing the entire learning pipeline from Figure 3.13).

Overall, all SSL approaches surpass supervised results in most cases, with contrastive methods outperforming masked prediction techniques. More specifically, SimCLR and Barlow Twins show the best performance, while MAEEG ranks as the least effective, probably because authors in [76] empirically observed no significant improvements when using the convolved representations from the signal encoder f'_θ (see Figure 3.10), whereas features outputted by the Transformer demonstrated superior performance. Employing the contextual features from the Transformer in the downstream task was not feasible due to the demanding execution requirements associated to that module.

The advantages of SSL are particularly evident in low-data regimes, experimenting the most significant accuracy increase when the percentage of labeled data is low. For example, SimCLR reaches a +6.75% accuracy improvement with just 2.5% of labeled data for training and a +1.03% with 100% of labels, while Barlow Twins achieves a +5.61% increase with 2.5% of labeled data and a +1.38% improvement with 100% of labels. The pre-training step effectively compensates for the data-hungry nature of deep learning models, boosting their performance in low-labeled scenarios, thereby addressing one of the primary objectives raised in this thesis.

In addition, these results further highlight the capability of SSL approaches to learn generalizable representations from the HOGAR dataset (recorded at home under uncontrolled conditions) that are highly useful for the downstream task of sleep staging, performed on the BOAS dataset (acquired in a controlled laboratory environment). Indeed, when training with 20% of labeled data, SimCLR achieves medical-grade sleep staging accuracy (82.48%), which is not reached when omitting the SSL pre-training (78.64%). This effectively demonstrates the success of an-

alyzing the transferability between both datasets, crucial for Bitbrain to additionally validate the quality of its novel device and the corresponding data collected, solidifying its potential as a scalable solution for both research and real-world applications. It highlights that the HOGAR dataset, despite being acquired in self-administer conditions, serves as a valuable resource for such learning pipelines, with potential applications extending to other tasks.

	Percentage of labeled data							
	2.5	5	10	20	40	60	80	100
Only supervised	63.35±2.14	69.42±0.59	74.46±1.01	78.64±0.75	82.46±0.51	84.14±0.70	84.47±0.61	85.02±0.90
SimCLR	**70.10±0.61**	**76.15±2.26**	**79.78±1.59**	**82.48±0.96**	**83.95±0.65**	**85.19±0.71**	85.36±0.58	86.05±1.31
BYOL	65.48±0.83	72.49±2.41	75.92±0.77	79.81±1.22	83.17±0.80	84.64±0.98	85.10±1.06	85.92±1.08
SimSiam	65.45±1.27	73.07±1.48	77.49±0.84	80.53±0.99	83.35±0.27	84.30±1.06	84.85±0.71	85.65±1.07
Barlow Twins	68.96±1.93	75.34±2.14	79.04±1.38	82.37±0.97	83.92±0.55	85.03±0.53	**85.55±0.85**	**86.40±0.80**
ContraWR	67.38±1.71	74.61±2.65	78.40±1.95	81.10±0.92	83.92±0.30	84.84±0.74	85.37±0.78	85.77±0.73
BENDR	66.80±1.68	73.94±1.80	78.11±1.33	80.79±1.13	83.49±0.71	84.46±0.74	84.62±0.72	85.52±0.46
MAEEG	65.14±0.93	71.43±1.68	74.82±1.33	78.74±0.34	82.24±0.78	83.61±0.79	84.62±0.62	85.09±0.98

Table 4.1: Accuracy and standard deviation comparison of the different approaches carried within the full HOGAR → BOAS scenario with distinct percentages of labeled data (X in Figure 3.15a), featuring a constant and large test set, evaluated in a semi-supervised procedure (*SEMI00*). Bold formatting highlights the top results.

To provide a more visual representation of the classification results, Figure 4.1 illustrates the accuracy evolution for each percentage of labeled data and method. As previously stated, the most noteworthy improvements occur in low-data regimes, where the accuracy gains are particularly significant. Then, the figure reveals a plateau in high-data scenarios, in which the benefits of SSL become less pronounced. Furthermore, all approaches exhibit a similar trend in performance evolution, which further reinforces the capabilities of SSL to effectively boost model accuracy.

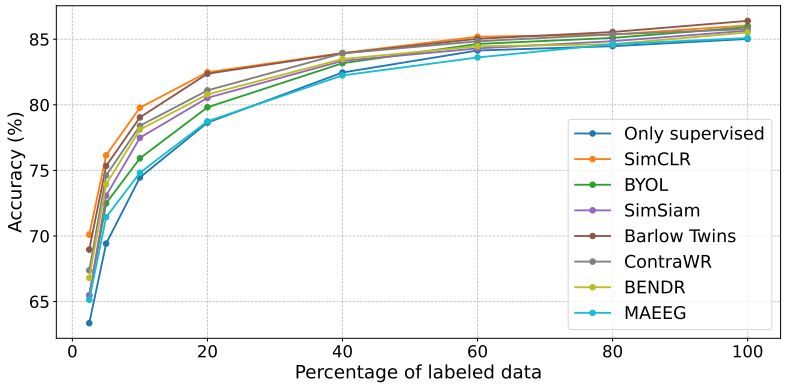


Figure 4.1: Accuracy evolution of the different approaches corresponding to test *SEMI00*.

Following with the subsequent test *SEMI01*, Table 4.2 presents the classification report as a result of the evaluation scenario from Figure 3.15b, in which a 10-fold cross validations is conducted within the supervised learning step on the BOAS dataset. The overall analysis aligns with the previous scenario, emphasizing the accuracy improvements of SSL methods in low-data

regimes. Specifically, Barlow Twins demonstrates a +8.08% accuracy improvement when training with 7.5% of labeled data, and SimCLR reaches a +0.78% gain with 100% of labels. In this case, ContraWR approach also classifies among the top three best methods, alongside SimCLR and Barlow Twins.

This evaluation scenario experiments higher variability in accuracy values, as reflected by the standard deviation, due to the fluctuations in the test set, which was the motivation to design the previous scenario. This effect is mitigated as the amount of labeled data increases, stabilizing evaluation outcomes. The last column of Table 4.2 serves as the definitive metric in a standard classification study, with SimCLR reaching a final accuracy of 87.21%, which surpasses the inter-scorer agreement threshold and therefore acquires a high medical-grade sleep scoring precision. This methodology also demonstrates that useful features are learned during the pretext task on the HOGAR dataset. A similar analysis can be derived from Figure 4.2, where the most interesting zone corresponds to low percentage values, with the trend leveling off into a plateau as the percentage of labeled data increases.

	Percentage of labeled data				
	7.5	15	20	60	100
Only supervised	72.11±5.97	80.35±2.33	81.34±1.49	84.97±1.06	86.43±0.05
SimCLR	79.22±4.86	83.15±1.63	**84.32±0.55**	85.49±1.14	**87.21±0.15**
BYOL	74.60±6.24	80.73±1.66	83.25±1.30	85.18±0.62	87.17±0.30
SimSiam	76.98±5.25	81.96±1.61	83.66±1.00	85.50±0.92	86.56±0.38
Barlow Twins	**80.19±3.97**	82.91±1.83	84.14±1.34	**85.73±1.19**	87.08±0.14
ContraWR	79.38±4.91	**83.37±1.53**	84.09±1.08	85.37±1.33	87.04±0.19
BENDR	79.95±2.96	81.55±0.71	83.34±0.76	85.38±0.75	86.48±0.03
MAEEG	75.87±4.95	80.00±0.74	82.13±1.64	85.01±1.19	86.26±0.09

Table 4.2: Accuracy and standard deviation comparison of the different approaches carried within the full HOGAR → BOAS scenario with distinct percentages of labeled data (X in Figure 3.15b), featuring a 10-fold cross-validation, evaluated in a semi-supervised procedure (*SEMI01*). Bold formatting highlights the top results.

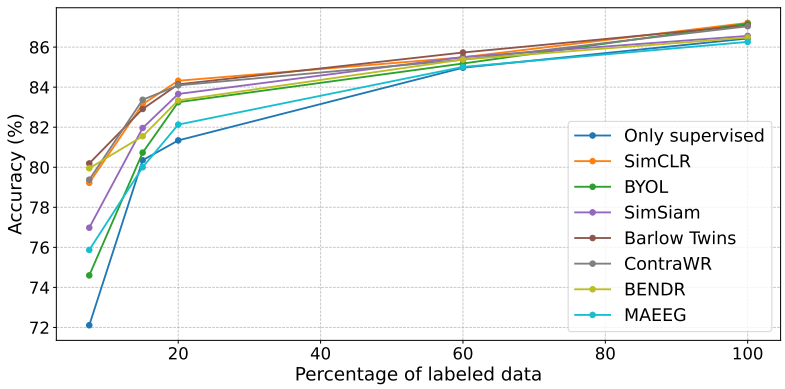


Figure 4.2: Accuracy evolution of the different approaches corresponding to test *SEMI01*.

Finally, the first part of Table 4.3 indicates the results of test *SEMI02*, where the BOAS dataset is complementary divided into portions for pre-training and supervised training. In addition, it includes the accuracy obtained by pre-training on an equivalent number of subjects selected from the HOGAR dataset, comprising test *SEMI03*. This setup enables a direct comparison between the general features learned from both datasets and their applicability to the sleep task. As previously observed, the most significant improvements corresponds to low-data scenarios, where SimCLR reaches a +8.9% and +9.98% accuracy improvement by pre-training on the BOAS and HOGAR datasets, respectively, with just 7.5% of the BOAS labeled data for supervised training. There are some cases in which incorporating SSL fails to yield performance gains with respect to the fully supervised procedure. In this sense, BYOL is the most impacted approach, exhibiting accuracy levels slightly below those of the supervised pipeline. Masked prediction techniques are not evaluated due to the extensive execution time required regarding the computational resources available, although its performance is not expected to surpass that of contrastive learning techniques based on previous results.

However, increasing the amount of labeled data in this scenario reduces the available unlabeled data for SSL pre-training. Consequently, the final columns of the table show performance levels similar to those of a purely supervised pipeline, even penalizing some approaches (e.g., BYOL) due to the limited pre-training data. The comparison between *SEMI02* and *SEMI03* tests reveals that the general representations learned during pre-training across datasets are equally robust, producing similar results. This finding is particularly promising given the differing recording conditions of each dataset. It demonstrates the feasibility of self-administered automatic sleep monitoring, serving as a proof-of-concept for a scalable, high-quality sleep solution. All these concepts are clearly reflected in Figures 4.3 and 4.4, which show a significant gain in accuracy in the low-data percentage range, followed by a plateau as the labeled and unlabeled data fractions become more balanced.

Taking everything into account, the following general insights can be drawn from the semi-supervised learning evaluation procedure. Firstly, the benefits of SSL are particularly pronounced in low-data regimes, effectively compensating the data-hungry nature of deep learning models. Conversely, when enough data is available to train models with expert-level accuracy, the additional gains from including SSL are not significant. Some authors argue that neither human scorers nor machine learning models can ever achieve 100% accuracy due to the aleatoric uncertainty (i.e., the inherent ambiguity in brain signals) [86]. Indeed, achieving a performance above the inter-scorer agreement (+82% [6], [7]) or the individual-against-consensus upper threshold (+85% [8]) is often regarded as an upper bound limit (see Table 2.1), representing the peak performance of these methodologies. This theoretical ceiling may explain why models, including those leveraging SSL, struggle to achieve further improvements in high-data environments.

More specifically, contrastive methods outperform masked prediction techniques. This could be explained due to the fact that contrastive approaches focus on distinguishing meaningful differences between augmented views in the latent space, directly enhancing feature separability, whereas masked techniques prioritize the reconstruction of low-level details, which might not effectively capture task-specific discriminative features between different input samples. Furthermore, contrastive methods directly employ the output vector from the feature encoder, which perfectly aligns with the presented network architecture in this work (see Figure 3.1), as the full backbone can be pre-trained. Among the contrastive methods, SimCLR and Barlow Twins emerged as the top performers, closely followed by ContraWR. Their success can be attributed

to the use of batch-wise comparisons with negative examples in the loss function, which appears to enhance the separability of feature representations. Conversely, BYOL and SimSiam exhibited comparatively lower performance, possibly because of the lack of negative examples in the training process, which seems to hinder their ability to learn robust representations within the model configurations and scenarios evaluated.

Regarding the masked approaches, BENDR showed superior results compared to MAEEG. BENDR is designed to learn features through a convolutional encoder (see Figure 3.9), which better suits the architecture utilized in this thesis. It also incorporates a contrastive loss function over the masked positions, which appears to perform more effectively than a straightforward reconstruction. On the other hand, MAEEG tries to adapt a masked autoencoder strategy but replaces the Transformer decoder with two lightweight layers (see Figure 3.10), potentially limiting the potential of the model to learn strong representations. As mentioned before, authors in [76] found that convolved representations from the signal encoder f'_θ did not yield improvements in the MAEEG framework, which can provide further context to understand its behavior.

These insights can be compared with findings reported in the literature that implement some of these techniques. For instance, authors in [78] observed comparable performance among the contrastive approaches developed, experimenting accuracy differences depending on the dataset tested. Similarly, Lee, Cheol-Hui, et al. [77] developed both SSL paradigms and reported similar performance regarding contrastive methods, with Barlow Twins and ContraWR as the most effective approaches. They also implemented BENDR framework, which results as one of the worst techniques. Both works demonstrated that BYOL and SimSiam, particularly the former one, can perform as well as or even better than approaches such as SimCLR. Therefore, further hyperparameter optimization and environment configuration could potentially yield higher accuracy values for these methods. This thesis does not aim to establish definitive conclusions about the theoretical superiority of one technique over another but rather pretends to contribute by evaluating a diverse range of methods within each SSL paradigm, providing a comprehensive perspective on their application and performance.

Finally, it has been demonstrated that general representations can be effectively learned from a home-recorded dataset (HOGAR) and successfully transferred to a dataset acquired in laboratory settings (BOAS), resulting in improved performance. Moreover, these SSL-derived features produce similar results when trained with the same amount of unlabeled data, even collected from different scenarios, further underscoring the promising potential obtained with wearable EEG devices such as the Bitbrain headband.

	Percentage of unlabeled-labeled data				
	92.5 - 7.5	80 - 20	70 - 30	60 - 40	50 - 50
Only supervised	69.46±4.04	82.44±1.29	83.70±0.95	85.25±0.36	85.70±0.56
BOAS → BOAS					
SimCLR	78.36±5.63	83.65±0.36	84.82±1.43	84.89±0.69	85.34±1.22
BYOL	67.74±7.34	75.55±10.93	82.91±0.53	84.00±1.15	84.56±1.19
SimSiam	73.01±7.56	82.09±1.08	84.76±0.58	85.23±0.13	85.61±0.32
Barlow Twins	77.72±5.24	**84.57±1.00**	**85.11±0.68**	85.10±0.12	85.81±0.21
ContraWR	76.35±4.67	83.24±1.10	84.30±1.18	85.16±0.17	85.96±0.01
HOGAR → BOAS					
SimCLR	**79.44±4.69**	83.76±1.05	84.76±0.97	84.92±0.13	85.60±0.62
BYOL	67.23±6.18	81.82±0.99	83.18±1.20	84.00±0.80	84.11±1.18
SimSiam	76.68±5.74	83.27±0.62	84.14±1.26	85.14±0.03	85.70±1.15
Barlow Twins	77.08±7.79	84.21±1.31	84.65±0.82	**85.33±0.55**	**86.22±0.11**
ContraWR	77.34±4.83	82.84±2.24	84.65±0.62	85.17±0.53	85.38±1.37

Table 4.3: Accuracy and standard deviation comparison of the different approaches carried within the BOAS → BOAS scenario with distinct percentages of labeled data (X in Figure 3.15c), featuring a 10-fold cross-validation, evaluated in a semi-supervised procedure (*SEMI02*). In addition, the second part shows the results for the HOGAR → BOAS scenario, where pre-training is performed on the HOGAR dataset using the exact same number of subjects as in the BOAS dataset (*SEMI03*). The percentages correspond to the BOAS dataset and are proportionally adjusted to manage the HOGAR pre-training data. Bold formatting highlights the top results.

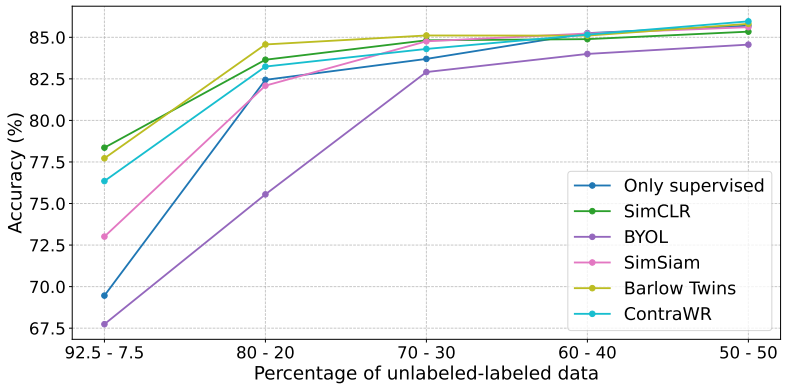


Figure 4.3: Accuracy evolution of the different approaches corresponding to test *SEMI02*.

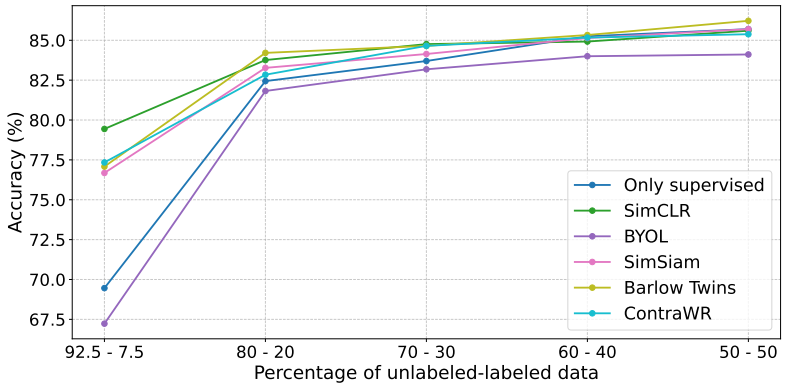


Figure 4.4: Accuracy evolution of the different approaches corresponding to test *SEMI03*.

4.2 Linear evaluation

In this section, the same tests previously reviewed are executed using a linear evaluation procedure. The results of the first test (*LINEV00*) are presented in Table 4.4. On the one hand, the first row indicates a fully supervised pipeline (the feature encoder is also trained on labeled data), while the second represents a frozen untrained encoder linearly evaluated. These pipelines serve as upper and lower bounds on performance, respectively. Subsequent rows correspond to the results of SSL pre-training approaches.

The features learned through the supervised approach outperform the frozen representations obtained with SSL, which can be attributed to several factors. In supervised learning, the model is able to learn more sleep-specific features due to the availability of labels. Moreover, both encoder and linear head are jointly optimized for classification, ensuring that learned features are directly aligned with the requirements of the downstream task. In contrast, the representations extracted through SSL are designed to be general-purpose, which might be less discriminative compared to those learned via direct supervision. Indeed, since the encoder remains frozen during linear evaluation, the features cannot be fine-tuned to better suit the downstream task.

However, this does not imply that the learned representations with SSL are suboptimal, as all methods surpass supervised performance in the semi-supervised evaluation (see Table 4.1). In this sense, techniques with stronger results (SimCLR, Barlow Twins, and ContraWR) experiment an increase in accuracy by training the linear head with more labeled data, thereby optimizing it to better understand the general features learned through SSL (since these representations remain identical and frozen in all tests). On the other hand, although methods ranking lower exhibit an accuracy gain in Table 4.1, they perform comparable to the untrained encoder. This may indicate that the linear layer alone may be insufficient to fully exploit the encoded features, particularly when the representations are not strongly separable, potentially creating a bottleneck. Nevertheless, a clear correlation exists between the approaches achieving the highest accuracy gains when evaluated in semi-supervised settings and the linear separability of their features. All these ideas are depicted in Figure 4.5, which highlights the positive trend described by the top-performing methods and the stagnation of the lower-ranked approaches.

	Percentage of labeled data							
	2.5	5	10	20	40	60	80	100
Only supervised	**62.89±1.37**	**66.36±1.34**	**69.78±2.29**	**74.48±1.28**	**78.92±0.91**	**80.47±0.84**	**81.23±1.17**	**81.96±0.83**
Untrained encoder	55.74±3.68	60.14±1.25	61.29±0.76	60.66±0.38	60.71±0.64	60.62±0.50	60.64±0.58	60.47±0.36
SimCLR	58.22±2.31	60.61±0.35	60.73±0.62	62.59±1.30	70.03±1.14	73.26±0.84	75.20±0.95	76.24±0.65
BYOL	50.64±2.89	59.48±1.91	60.47±0.38	60.45±0.38	60.45±0.35	60.44±0.38	60.46±0.32	60.48±0.33
SimSiam	54.55±3.98	60.87±0.75	61.17±0.64	62.08±0.54	62.53±0.48	63.07±0.51	63.31±0.50	64.10±0.67
Barlow Twins	56.80±4.05	60.55±0.55	60.59±0.60	60.88±0.75	64.40±1.06	68.18±1.09	70.37±1.38	71.80±1.18
ContraWR	56.12±4.22	61.29±0.66	62.44±0.70	67.29±1.46	72.58±1.30	73.54±1.41	74.44±0.71	74.83±0.71
BENDR	58.53±2.61	60.25±0.40	60.44±0.37	60.43±0.35	60.44±0.38	60.45±0.38	60.45±0.38	60.45±0.38
MAEEG	55.85±2.21	59.40±1.29	60.41±0.37	60.45±0.38	60.51±0.31	60.47±0.40	60.45±0.38	60.45±0.38

Table 4.4: Accuracy and standard deviation comparison of the different approaches carried within the full HOGAR → BOAS scenario with distinct percentages of labeled data (X in Figure 3.15a), featuring a constant and large test set, linearly evaluated (*LINEV00*). Bold formatting highlights the top results, and underlining is used for the second-best.

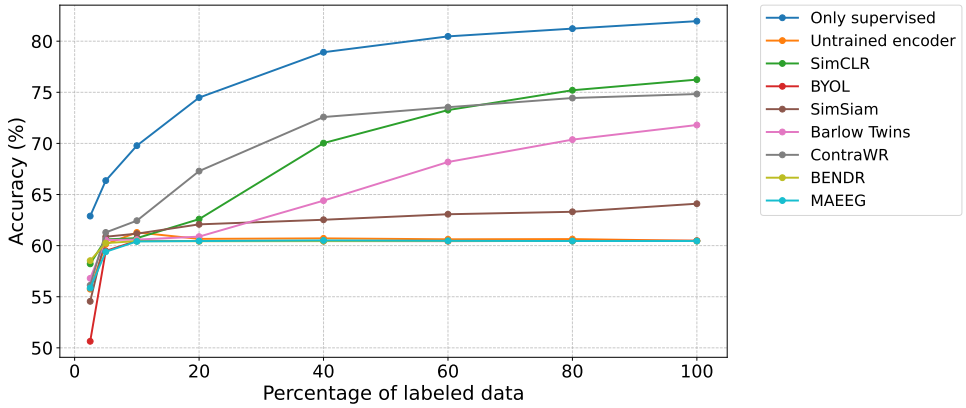


Figure 4.5: Accuracy evolution of the different approaches corresponding to test *LINEV00*.

The results of the test *LINEV01* are shown in Table 4.5, which mirrors the scenario of test *SEMI01* but is evaluated using a linear procedure. Similar to the previous discussion, the purely supervised pipeline outperforms the other approaches in this modality, but helps improve performance when evaluated in semi-supervised settings (see Table 4.2). Furthermore, ContraWR achieves the best accuracy in low-data regimes, being relatively close to fully supervised results, while SimCLR excels when the amount of labeled data is high. Figure 4.6 illustrates the values provided, exhibiting consistent insights with those observed in the previous test.

	Percentage of labeled data				
	7.5	15	20	60	100
Only supervised	**66.74±5.72**	**76.30±1.62**	**78.08±3.18**	**81.92±0.54**	**83.39±0.07**
Untrained encoder	60.66±2.63	60.79±0.95	60.31±1.14	60.64±0.14	60.32±0.16
SimCLR	60.67±2.94	64.77±1.94	68.28±2.05	76.58±0.16	78.15±0.05
BYOL	60.70±2.77	60.72±1.05	60.28±1.16	60.59±0.09	60.18±0.01
SimSiam	61.81±2.06	62.01±1.48	62.10±0.67	64.28±0.06	64.68±0.24
Barlow Twins	61.11±2.55	61.42±0.89	62.69±1.72	71.61±0.02	73.88±0.26
ContraWR	65.77±3.73	71.28±2.12	72.75±0.98	74.87±0.30	75.35±0.04
BENDR	60.72±2.69	60.77±0.95	60.28±1.16	60.60±0.10	60.18±0.01
MAEEG	60.22±3.71	60.75±0.99	60.29±1.17	60.60±0.10	60.18±0.01

Table 4.5: Accuracy and standard deviation comparison of the different approaches carried within the full HOGAR → BOAS scenario with distinct percentages of labeled data (X in Figure 3.15b), featuring a 10-fold cross-validation, linearly evaluated (*LINEV01*). Bold formatting highlights the top results, and underlining is used for the second-best.

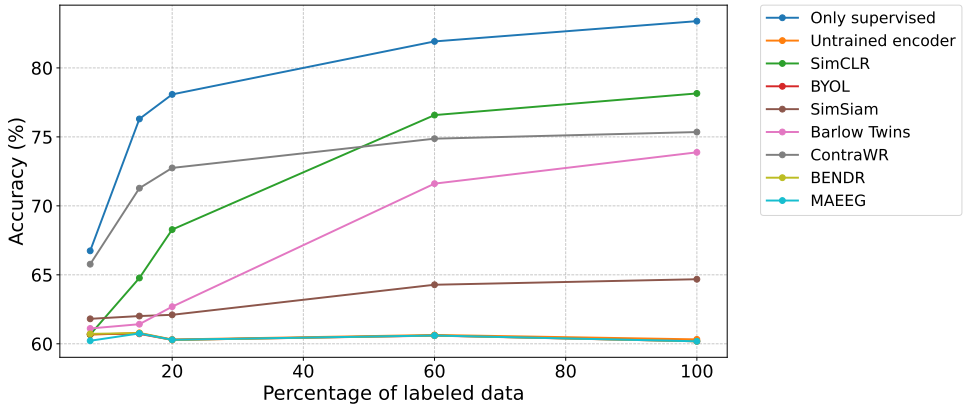


Figure 4.6: Accuracy evolution of the different approaches corresponding to test *LINEV01*.

Finally, Table 4.6 provides a linear comparison between the features extracted from each dataset, corresponding to tests *LINEV02* and *LINEV03*, respectively. Similar to the semi-supervised experiments, masked approaches are excluded from this scenario due to their high execution time requirements. The supervised pipeline outperforms the general features learned by SSL methods, with SimCLR achieving the best results. Barlow Twins and ContraWR also perform reasonably well, in contrast to BYOL and SimSiam. In relation with previous results, the smallest gap between SSL and supervised outcomes occurs in the scenario with a lowest amount of labeled data, which underscores the strength of SSL under low-data conditions. Significant differences are not observed when comparing the features learned from the two datasets, resulting in similar performance and further validating the HOGAR dataset (self-recorded by users) as a valuable resource for learning purposes. Figures 4.7 and 4.8 depict these concepts in terms of accuracy evolution.

Everything taken into consideration, the linear evaluation procedure reveals several key observations. Firstly, the sleep-specific features learned through supervised pipelines outperform the general-purpose representations learned by SSL methods, specially when the amount of labeled data increases. This trend aligns with findings in the literature (see Table 2.2), being common to experience more pronounced accuracy gains in linear evaluation under low-data conditions. For instance, in the ContraWR paper [78], authors discovered that when labeled data exceeded 5%, the supervised pipeline outperformed most SSL methods. In this sense, further reducing the amount of labeled data in the previous scenarios would likely result in SSL methods surpassing the supervised pipeline. This notion is supported by the observation that the smallest performance gap occurs when the percentage of labeled data is reduced, additionally highlighting the capabilities of SSL in these environments. However, these results do not imply that SSL methods fail to capture useful information. As determined in the previous section, all SSL approaches consistently surpass supervised performance when fine-tuning is incorporated into the semi-supervised evaluations, with contrastive learning methods over masked techniques.

In addition, it has been further confirmed that meaningful information can be effectively learned from the HOGAR dataset and successfully transferred to the BOAS dataset. Indeed, the features extracted from each dataset yield similar results, thereby exhibiting comparable robustness. The primary objectives of the analysis carried is to provide a deeper understanding of how the developed SSL methods work under an additional evaluation procedure commonly used in the literature. Therefore, semi-supervised settings represent the practical application of self-supervised learning, whose results serve as a benchmark for a real-world implementation.

	Percentage of unlabeled-labeled data				
	92.5 - 7.5	80 - 20	70 - 30	60 - 40	50 - 50
Only supervised	**65.52±6.38**	**78.38±2.54**	**80.89±1.06**	**81.89±0.71**	**82.15±0.06**
Untrained encoder	60.75±2.29	60.95±1.02	60.86±0.36	60.70±0.07	60.85±0.70
BOAS → BOAS					
SimCLR	63.40±1.54	72.30±0.19	72.63±0.28	73.49±1.88	74.01±2.79
BYOL	60.52±2.25	61.27±1.44	60.70±0.12	60.86±0.37	61.18±0.28
SimSiam	60.60±1.44	61.03±1.15	60.67±0.10	60.88±0.23	60.82±0.67
Barlow Twins	62.35±1.24	71.37±2.19	72.26±0.36	72.29±0.28	73.84±1.72
ContraWR	62.66±2.47	67.97±1.33	68.76±1.74	70.23±1.12	71.30±1.08
HOGAR → BOAS					
SimCLR	62.49±2.68	71.01±1.06	73.40±0.48	73.85±1.09	74.44±0.73
BYOL	60.47±2.53	60.91±1.08	60.75±0.22	60.87±0.36	60.81±0.64
SimSiam	60.44±2.13	60.83±1.00	60.78±0.03	60.84±0.39	61.33±1.39
Barlow Twins	63.20±2.40	68.44±3.94	70.49±2.26	71.60±0.52	71.86±1.13
ContraWR	61.65±1.11	68.76±1.39	69.15±0.34	69.53±0.03	70.02±0.02

Table 4.6: Accuracy and standard deviation comparison of the different approaches carried within the BOAS → BOAS scenario with distinct percentages of labeled data (X in Figure 3.15c), featuring a 10-fold cross-validation, linearly evaluated (*LINEV02*). The second part shows the results for the HOGAR → BOAS scenario, where pre-training is performed on the HOGAR dataset using the exact same number of subjects as in the BOAS dataset (*LINEV03*). The percentages correspond to the BOAS dataset and are proportionally adjusted to manage the HOGAR pre-training data. Bold formatting highlights the top results, and underlining is used for the second-best.

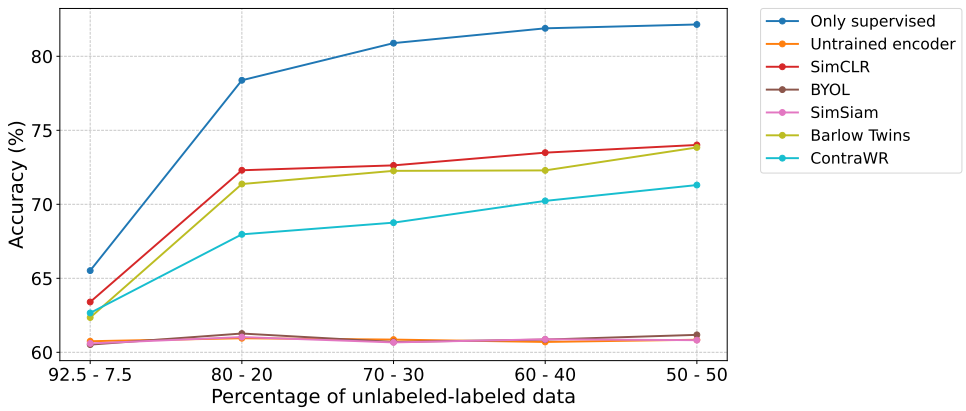


Figure 4.7: Accuracy evolution of the different approaches corresponding to test *LINEV02*.

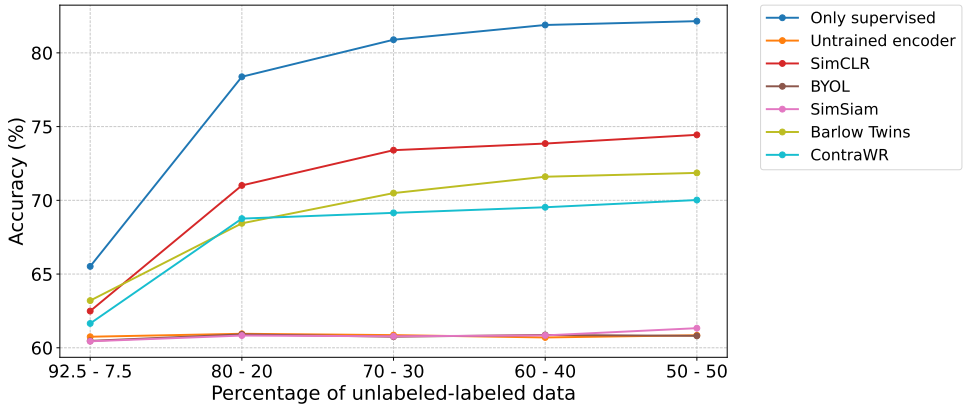


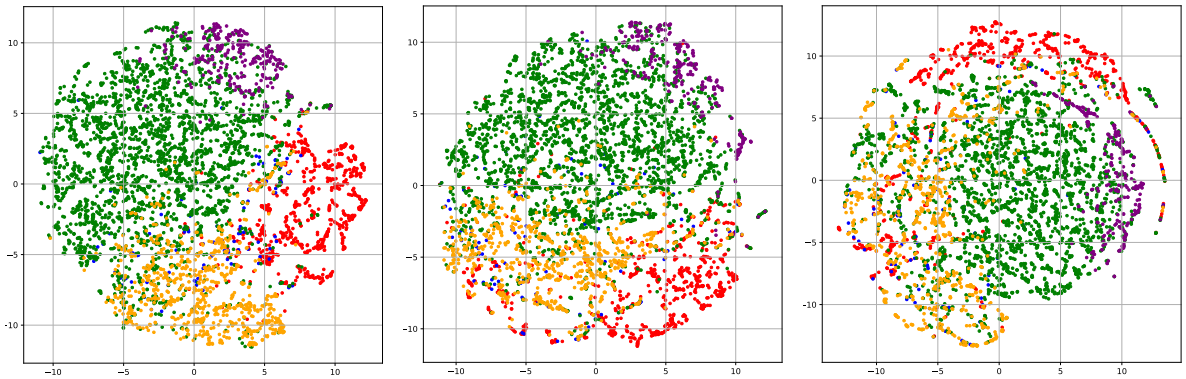
Figure 4.8: Accuracy evolution of the different approaches corresponding to test *LINEV03*.

4.3 Feature visualization

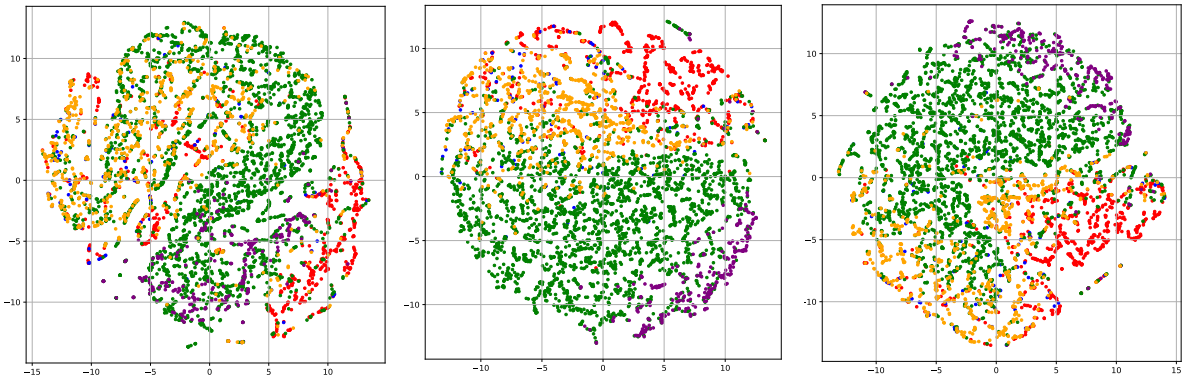
This section evaluates the quality of the features learned by projecting them onto the 2D space using t-SNE and UMAP tools (see Figure 3.16 from the previous chapter). Masked prediction methods are excluded from this procedure because they generate a two-dimensional output rather than a feature embedding (note f'_θ in Figures 3.9 and 3.10), which is required for both approaches. Based on the characteristics of each sleep stage, reviewed in Appendix D, it is expected that N1 samples will cluster near wake embeddings since N1 is considered a transition period between wakefulness and sleep. Similarly, REM samples are anticipated to appear in close proximity to wake and N1 clusters since their EEG features resemble those seen in the awake state, but tend to be a bit slower and higher in amplitude. N2 embeddings are likely to diverge from these stages, reflecting the descent in brain activity. Finally, N3 cluster, which corresponds to the deepest sleep stage, is expected to be the most distant group in the plot, revealing its characteristic low-frequency delta waves.

To begin with, Figure 4.9 illustrates the plots generated with t-SNE, where the models were trained on the entire HOGAR dataset using a certain SSL technique and then visually tested on the embeddings of the first five labeled subjects from the BOAS dataset, without fine-tuning. The supervised approach, whose features were additionally fine-tuned on the BOAS dataset to provide a reference plot, exhibits the clearest separation of features in the 2D space, aligning with the expected patterns discussed before. SimCLR, Barlow Twins and ContraWR demonstrate the best clustering, with wake, N1 and REM samples showing slight overlap compared to the supervised features. Conversely, BYOL and SimSiam exhibit a poorer inter-class separation, consistent with their lower numerical performance in the previous evaluation tests.

On the other hand, Figure 4.10 depicts the 2D feature plots obtained using the UMAP tool with the same models and data. As before, the fully supervised approach achieves the best feature separation, further reinforcing the inherent differences between the EEG hallmarks of each sleep stage. Among the SSL techniques, SimCLR demonstrates comparable clustering performance, comprising the most effective result. Interestingly, BYOL performs notably well with UMAP, although not reaching the highest accuracy improvements in earlier tests. Barlow Twins and ContraWR present good feature separability, whereas SimSiam indicates the weakest performance in this evaluation.

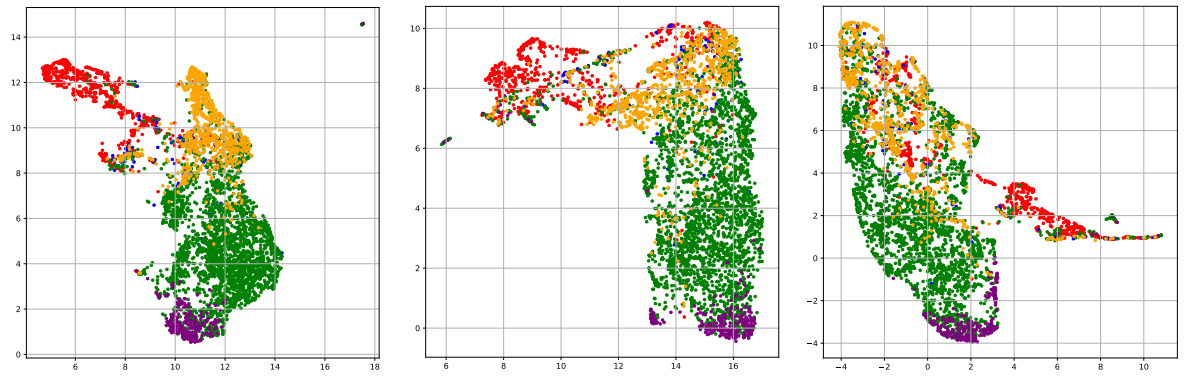


(a) Supervised. (b) SimCLR. (c) BYOL.

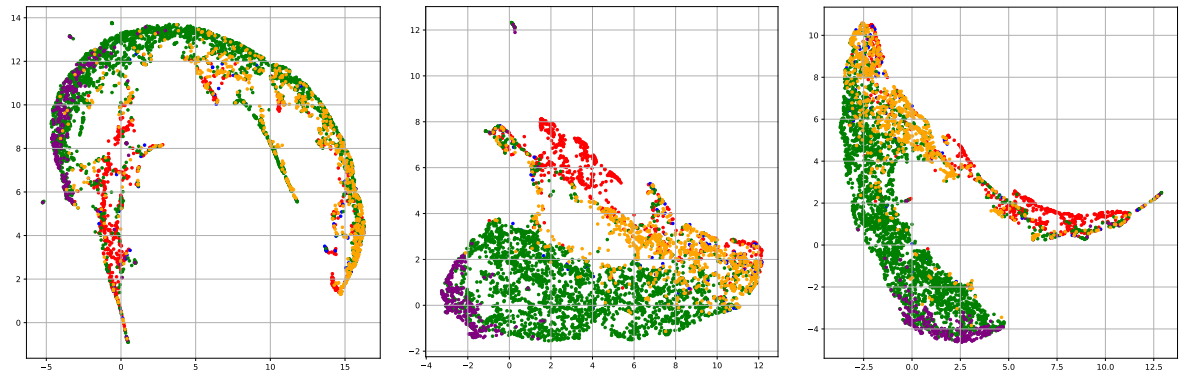


(d) SimSiam. (e) Barlow Twins. (f) ContraWR.

Figure 4.9: t-SNE feature visualization results (● Wake, ● N1, ● N2, ● N3, ● REM).



(a) Supervised. (b) SimCLR. (c) BYOL.



(d) SimSiam. (e) Barlow Twins. (f) ContraWR.

Figure 4.10: UMAP feature visualization results (● Wake, ● N1, ● N2, ● N3, ● REM).

On the whole, the results shown in this section aim to provide a visual perspective of the general features learned through SSL techniques and their separability within the feature space. In the absence of labels, some methods demonstrate comparable clustering performance to that of the fully supervised pipeline, further reinforcing the feasibility of extracting meaningful features from an unlabeled dataset (HOGAR) and effectively transferring this general knowledge to domain-specific tasks, such as those represented by the BOAS dataset.

Chapter 5

Conclusions

This thesis has successfully design and develop self-supervised learning (SSL) methods as a pre-training step within the learning pipeline to extract meaningful electroencephalogram (EEG) representations from unlabeled data, exploiting the inherent structure of brain signals. A total of five contrastive learning approaches and two masked prediction techniques were implemented and evaluated, providing a complete exploration of different SSL paradigms. By incorporating these methods, the data-hungry nature of deep learning models is effectively addressed through the utilization of additional data, potentially reducing the elevated costs of the manual scoring process carried by technicians, while also mitigating the impact of low inter-scorer agreement and the ambiguity of human-generated labels.

A comprehensive set of tests was designed and executed, consisting of evaluation procedures commonly used in the literature, such as semi-supervised learning and linear evaluation, alongside diverse data partitioning schemes, including a constant test set and 10-fold cross-validation. These tests were applied across different scenarios based on the datasets and the proportions employed for self-supervised and supervised training, respectively.

More specifically, the transferability between the HOGAR dataset, self-recorded by users under domestic conditions without experts oversight, and the BOAS dataset, acquired in a controlled laboratory environment, was thoroughly assessed and successfully demonstrated. As a result, significant improvements were achieved by incorporating SSL techniques (+9.98% accuracy gain in test *SEMI03*, see Table 4.3), effectively leveraging the general representations learned from the HOGAR dataset during supervised fine-tuning on the BOAS dataset. Indeed, the intra-dataset SSL feature extraction comparison revealed that the representations derived from each dataset are equally robust, producing similar results. This finding is particularly promising given the differing recording conditions, validating the HOGAR dataset as a useful resource for learning purposes. Moreover, the feature visualization analysis further supports this outcome and contributes to the interpretability of the system, where some methods demonstrated comparable clustering performance to that of the fully supervised pipeline, even in the absence of labels.

On the other side, this work provides a clear and complete understanding of the performance of SSL relative to the labeled data available. Consequently, executing the aforementioned evaluation scenarios with different percentages of data raised that the advantages of SSL in terms of accuracy gains are particularly pronounced in low-data regimes, whereas the improvements in high-data environments are comparatively modest (+6.75% gain with 2.5% labeled data and +1.38% increase with 100% labels in test *SEMI00*, see Table 4.1). This demonstrates the potential of SSL to address the data-hungry characteristic of deep learning by effectively leveraging

unlabeled data. To the best of our knowledge, no prior study offers such a comprehensive understanding of SSL performance across distinct amounts of labeled data, thereby providing valuable insights into the practical applications and limitations of self-supervised automatic sleep scoring in real-world implementations.

Current state-of-the-art deep learning methods for sleep scoring are already achieving, and even surpassing, expert-level accuracy when evaluated on large, annotated, and high-quality datasets (see Table 2.1). As some authors claim, neither human scorers nor machine learning models can ever achieve 100% accuracy due to the aleatoric uncertainty (i.e., the inherent ambiguity in brain signals) [86]. Achieving a performance above the inter-scorer agreement (+82% [6], [7]) or the individual-against-consensus upper threshold (+85% [8]) is often regarded as an upper bound limit. This theoretical ceiling may explain why models leveraging SSL struggle to achieve further improvements in high-data environments.

As a consequence, since a significant part of the neuroscience research operates in low-labeled regimes, this study additionally contributes by exploring SSL techniques that significantly enhance performance under data-restricted conditions, as demonstrated by the reviewed results. Such low-data regimes remain underexplored in the literature, making this research a valuable step toward addressing these gaps and advancing practical applications in the field. However, this does not mean that it lacks additional uses. In scenarios such as deploying a new headband or a different type of sensor, where re-labeling data is expensive, it would not be necessary to include as many subjects to achieve considerable medical-grade sleep scoring accuracy around 80%, as obtained in tests *SEMI00* with 10% and 20% of labels (see Table 4.1), and *SEMI01* with 7.5% of labeled data (see Table 4.2).

The self-supervised learning methods developed in this work are inherently agnostic to the domain-specific task of sleep scoring. The pretext task comprising the representational learning step could be easily adapted to pre-train different deep learning models. Therefore, this work serves as a proof-of-concept that demonstrates the feasibility of learning general-purpose representations from the HOGAR dataset that not only enhance outcomes in sleep scoring but also hold the potential to improve downstream performance in other biosignal processing tasks.

All things considered, this thesis makes significant contributions to the design and validation of label-efficient sleep staging systems, providing a comprehensive analysis of the transferability between two distinct datasets and the performance under different data availability conditions. Bitbrain leverages this knowledge to enhance the efficiency of its deep learning models by incorporating the growing volume of EEG data that is impossible to fully analyze and annotate, eliminating the need for further training specialized experts to operate in this novel environment. Therefore, this work serves as a critical step forward in the efficiency of self-administered and large-scale automatic sleep monitoring in home environments under uncontrolled conditions, advancing in a scalable solution for the substantial proportion of the word population suffering from serious sleep disorders that require medical attention.

5.1 Future work

Given the promising results obtained, the following open research lines are stated:

- Further explore the data augmentation techniques employed in contrastive learning methods. As highlighted in the literature, the choice of data transformations significantly

impacts downstream performance, making it crucial to apply augmentations that better suit to each SSL approach and dataset [63], [72], [78]. In this sense, investigating non-deterministic, learning-based techniques, such as autoencoders or GANs, could offer compelling alternatives to traditional hand-crafted augmentations.

- Expand the evaluation of the learning pipeline to encompass additional scenarios. For instance, assessing the impact in performance of varying the amount of HOGAR data used during pre-training could provide deeper insights into the role of the unlabeled-based part of the pipeline beyond the provided understanding of the supervised step with labels. Furthermore, testing the proposed pipeline using typical state-of-the-art datasets could additionally serve as valuable procedure for result comparison. Related with this, employing public datasets along with the HOGAR dataset as a huge database for representational learning could improve performance, leading to a inter-dataset analysis within the pre-training phase.

  Another example corresponds to supervised training with the HOGAR dataset, whose labels could be established by Bitbrain experts, and inference with the BOAS dataset to evaluate if the accuracy reaches the $\sim 86\%$ achieved when training directly with the BOAS dataset. Obtaining this would be a significant advantage, as the former dataset uses fewer sensors (no PSG), lacks professional assistance, and does not include a consensus process (a single evaluator performs the labeling). This would eliminate the need for data labeling by three individuals and an additional expert to derive a consensus. If the accuracy falls short, semi-supervised learning could be introduced to help close the gap.

- Increase the scale of the learning system by expanding both the model size and dataset scope. The HOGAR dataset is currently growing due to the accessible and user-friendly recording conditions enabled by Bitbrain's wearable EEG headband. Consequently, these techniques could be tested in a future within an unlabeled dataset comprising thousands of recordings rather than the current scale of a few hundred, potentially leading to even greater SSL performance improvements. This could also make the model more robust against data from non-healthy individuals, who are often more costly and challenging to label, by leveraging the subjects diagnosed with dementia in the HOGAR dataset, whose number is currently limited, since this resource aims to improve the diagnosis and early treatment of aging-related diseases.

  In addition, the availability of such amount of training data would allow for the deployment of larger and more advanced deep learning models, which are known to offer superior feature representation capabilities. This future step may include a hyperparameter tuning of the employed network, which was initially optimized with a smaller subset of the BOAS dataset, or using the Transformer network developed in a previous work [44]. While both architectures demonstrated similar performance in the past, the increased availability of data (and the prospect of even more in the future) could potentially raise the upper performance limit by uncovering separations in data distributions that were previously undetectable due to insufficient data. Alternatively, the SSL pipeline could also be leveraged to reach this upper limit with the minimum possible amount of labeled data utilized for the supervised learning step.

- Maintain an active exploration of emerging techniques in the literature. Learning from massive volumes of data remains an active area of research within the machine learning community, powered by the current advancements in artificial intelligence. Given the potential of the learning environment established for automatic sleep scoring and the fu-

ture expansion of the system's scale mentioned before, incorporating novel approaches could be considered to further improve system performance and broaden the scope of the project. For example, the method proposed in [87] leverages big self-supervised models in a semi-supervised setting with an additional knowledge distillation step, where a fine-tuned network acts as a teacher to generate labels for training a student network. Furthermore, a recent work introducing NeuroNet [77] adopts an hybrid approach that combines contrastive learning and masked prediction tasks, demonstrating superior performance over existing SSL approaches and latest supervised learning methodologies within the automatic sleep scoring field. Adopting similar strategies could unlock new opportunities for additionally optimizing automatic sleep monitoring systems.

Bibliography

[1] L. Besedovsky, T. Lange, and M. Haack, "The sleep-immune crosstalk in health and disease," *Physiological Reviews*, vol. 99, no. 3, pp. 1325–1380, 2019.

[2] F. P. Cappuccio and M. A. Miller, "Sleep and cardio-metabolic disease," *Current Cardiology Reports*, vol. 19, no. 11, 2017.

[3] E. C. Harding, N. P. Franks, and W. Wisden, "Sleep and thermoregulation," *Current Opinion in Physiology*, vol. 15, pp. 7–13, 2020.

[4] M. M. Ohayon, "Epidemiological overview of sleep disorders in the general population," *Sleep Medicine Research*, vol. 2, no. 1, pp. 1–9, 2011.

[5] M. M. Ohayon, "Prevalence and comorbidity of sleep disorders in general population," *La Revue du Praticien*, vol. 57, no. 14, pp. 1521–1528, 2007.

[6] R. S. Rosenberg and S. V. Hout, "The american academy of sleep medicine inter-scorer reliability program: Sleep stage scoring," *Journal of Clinical Sleep Medicine*, vol. 09, no. 01, pp. 81–87, 2013.

[7] Y. J. Lee, J. Y. Lee, J. H. Cho, and J. H. Choi, "Interrater reliability of sleep stage scoring: A meta-analysis," *Journal of Clinical Sleep Medicine*, vol. 18, no. 1, pp. 193–202, 2022.

[8] J. B. Stephansen, A. N. Olesen, M. Olsen, *et al.*, "Neural network analysis of sleep stages enables efficient diagnosis of narcolepsy," *Nature communications*, vol. 9, no. 1, p. 5229, 2018.

[9] L. Höller and H. Riemer, "Comparison of visual analysis and automatic sleep stage scoring (oxford medilog 9000 system)," *European Neurology*, vol. 25, no. 2, pp. 36–45, 1986.

[10] A. Supratak, H. Dong, C. Wu, and Y. Guo, "DeepSleepNet: A model for automatic sleep stage scoring based on raw single-channel EEG," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 11, pp. 1998–2008, 2017.

[11] H. Phan, K. Mikkelsen, O. Y. Chen, P. Koch, A. Mertins, and M. D. Vos, "SleepTransformer: Automatic sleep staging with interpretability and uncertainty quantification," *IEEE Transactions on Biomedical Engineering*, vol. 69, no. 8, pp. 2456–2467, 2022.

[12] M. Kim, K. Jung, and W. Chung, "Automatic sleep stage classification method based on transformer-in-transformer," in *2023 11th International Winter Conference on Brain-Computer Interface (BCI)*, IEEE, 2023.

[13] E. López-Larraz, C. Escolano, A. Robledo-Menéndez, L. Morlas, A. Alda, and J. Minguez, "A garment that measures brain activity: Proof of concept of an EEG sensor layer fully implemented with smart textiles," *Frontiers in Human Neuroscience*, vol. 17, 2023.

[14] H.-V. V. Ngo, T. Martinetz, J. Born, and M. Mölle, "Auditory closed-loop stimulation of the sleep slow oscillation enhances memory," *Neuron*, vol. 78, no. 3, pp. 545–553, 2013.

[15] L. Marshall, H. Helgadóttir, M. Mölle, and J. Born, "Boosting slow oscillations during sleep potentiates memory," *Nature*, vol. 444, no. 7119, pp. 610–613, 2006.

[16] R. Vallat and M. P. Walker, "An open-source, high-performance tool for automated sleep staging," *eLife*, vol. 10, 2021.

[17] H. Banville, O. Chehab, A. Hyvärinen, D.-A. Engemann, and A. Gramfort, "Uncovering the structure of clinical eeg signals with self-supervised learning," *Journal of Neural Engineering*, vol. 18, no. 4, p. 046 020, 2021.

[18] S. Diekelmann and J. Born, "The memory function of sleep," *Nature Reviews Neuroscience*, vol. 11, no. 2, pp. 114–126, 2010.

[19] J. A. Hobson, "A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects," *Electroencephalography and Clinical Neurophysiology*, vol. 26, p. 644, 1969.

[20] C. Iber, S. Ancoli-Israel, A. Chesson, and S. Quan, "The aasm manual for the scoring of sleep and associated events: Rules, terminology and technical specifications," *Westchester, IL: American Academy of Sleep Medicine*, 2007.

[21] L. Fiorillo, A. Puiatti, M. Papandrea, *et al.*, "Automated sleep scoring: A review of the latest approaches," *Sleep Medicine Reviews*, vol. 48, p. 101 204, 2019.

[22] A. Biasiucci, B. Franceschiello, and M. M. Murray, "Electroencephalography," *Current Biology*, vol. 29, no. 3, R80–R85, 2019.

[23] H. Berger, "Über das elektrenkephalogramm des menschen," *Archiv für Psychiatrie und Nervenkrankheiten*, vol. 87, no. 1, pp. 527–570, 1929.

[24] G. Fu, Y. Zhou, P. Gong, P. Wang, W. Shao, and D. Zhang, "A temporal-spectral fused and attention-based deep model for automatic sleep staging," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 31, pp. 1008–1018, 2023.

[25] M. Čić, J. Šoda, and M. Bonković, "Automatic classification of infant sleep based on instantaneous frequencies in a single-channel EEG signal," *Computers in Biology and Medicine*, vol. 43, no. 12, pp. 2110–2117, 2013.

[26] P. Memar and F. Faradji, "A novel multi-class EEG-based sleep stage classification system," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 1, pp. 84–95, 2018.

[27] M. Diykh, Y. Li, and P. Wen, "EEG sleep stages classification based on time domain features and structural graph similarity," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 11, pp. 1159–1168, 2016.

[28] A. R. Hassan and A. Subasi, "A decision support system for automated identification of sleep stages from single-channel EEG signals," *Knowledge-Based Systems*, vol. 128, pp. 115–124, 2017.

[29] P. Chriskos, D. S. Kaitalidou, G. Karakasis, *et al.*, "Automatic sleep stage classification applying machine learning algorithms on EEG recordings," in *2017 IEEE 30th International Symposium on Computer-Based Medical Systems (CBMS)*, IEEE, 2017.

[30] L. Siyuan, L. Jingyuan, G. Hangping, and R. Minhua, "Sleep staging prediction model based on xgboost," in *2021 International Conference on Electronic Information Engineering and Computer Science (EIECS)*, 2021, pp. 350–353.

[31] H. Phan and K. Mikkelsen, "Automatic sleep staging of EEG signals: Recent development, challenges, and future directions," *Physiological Measurement*, vol. 43, no. 4, 2022.

[32] H. Dong, A. Supratak, W. Pan, C. Wu, P. M. Matthews, and Y. Guo, "Mixed neural network approach for temporal sleep stage classification," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 2, pp. 324–333, 2018.

[33] R. Wei, X. Zhang, J. Wang, and X. Dang, "The research of sleep staging based on single-lead electrocardiogram and deep neural network," *Biomedical Engineering Letters*, vol. 8, no. 1, pp. 87–93, 2017.

[34] O. Tsinalis, P. M. Matthews, Y. Guo, and S. Zafeiriou, *Automatic sleep stage scoring with single-channel eeg using convolutional neural networks*, 2016.

[35] A. Vilamala, K. H. Madsen, and L. K. Hansen, "Deep convolutional neural networks for interpretable analysis of EEG sleep stage scoring," in *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, IEEE, 2017.

[36] H. Phan, F. Andreotti, N. Cooray, O. Y. Chen, and M. D. Vos, "DNN filter bank improves 1-max pooling CNN for single-channel EEG automatic sleep stage classification," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, 2018.

[37] H. Phan, F. Andreotti, N. Cooray, O. Y. Chen, and M. D. Vos, "Automatic sleep stage classification using single-channel EEG: Learning sequential features with attention-based recurrent neural networks," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, 2018.

[38] H. Phan, F. Andreotti, N. Cooray, O. Y. Chen, and M. D. Vos, "SeqSleepNet: End-to-end hierarchical recurrent neural network for sequence-to-sequence automatic sleep staging," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 3, pp. 400–410, 2019.

[39] H. Phan, O. Y. Chen, M. C. Tran, P. Koch, A. Mertins, and M. D. Vos, "XSleepNet: Multi-view sequential model for automatic sleep staging," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

[40] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, *Attention is all you need*, 2017.

[41] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, 2018.

[42] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, *et al.*, "Improving language understanding by generative pre-training," 2018.

[43] M. Sierra Torralba and L. Montesano del Campo, "Clasificación automatizada de las fases del sueño mediante el uso de bioseñales.," 2022.

[44] E. Estevan Tomás, M. Sierra Torralba, and L. Montesano del Campo, "Clasificación automática de las etapas del sueño mediante técnicas de aprendizaje profundo.," 2023.

[45] M. Esparza-Iaizzo, I. Álvarez, J. Klinzing, L. Montesano, J. Minguez, and E. López-Larraz, "Sleepbci: A platform for memory enhancement during sleep based on automatic scoring," in *Proceedings of the XXXIX annual congress of the spanish society of biomedical engineering, Valladolid*, 2021.

[46] M. Esparza-Iaizzo, M. Sierra-Torralba, J. G. Klinzing, J. Minguez, L. Montesano, and E. López-Larraz, "Automatic sleep scoring for real-time monitoring and stimulation in individuals with and without sleep apnea," *bioRxiv*, pp. 2024–06, 2024.

[47] E. López-Larraz, M. Sierra-Torralba, S. Clemente, *et al.*, *"bitbrain open access sleep dataset"*, OpenNeuro, 2024.

[48] A. L. Goldberger, L. A. N. Amaral, L. Glass, *et al.*, "PhysioBank, PhysioToolkit, and PhysioNet," *Circulation*, vol. 101, no. 23, 2000.

[49] B. Kemp, A. Zwinderman, B. Tuk, H. Kamphuisen, and J. Oberye, "Analysis of a sleep-dependent neuronal feedback loop: The slow-wave microcontinuity of the EEG," *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 9, pp. 1185–1194, 2000.

[50] C. O'Reilly, N. Gosselin, J. Carrier, and T. Nielsen, "Montreal archive of sleep studies: An open-access resource for instrument benchmarking and exploratory research," *Journal of Sleep Research*, vol. 23, no. 6, pp. 628–635, 2014.

[51] G.-Q. Zhang, L. Cui, R. Mueller, *et al.*, "The national sleep research resource: Towards a sleep data commons," *Journal of the American Medical Informatics Association*, vol. 25, no. 10, pp. 1351–1358, 2018.

[52] S. F. Quan, B. V. Howard, C. Iber, *et al.*, "The Sleep Heart Health Study: Design, Rationale, and Methods," *Sleep*, vol. 20, no. 12, pp. 1077–1085, 1997.

[53] A. Supratak and Y. Guo, "Tinysleepnet: An efficient deep learning model for sleep stage scoring based on raw single-channel eeg," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, 2020, pp. 641–644.

[54] E. Eldele, Z. Chen, C. Liu, *et al.*, "An attention-based deep learning approach for sleep stage classification with single-channel eeg," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 29, pp. 809–818, 2021.

[55] M. Fu, Y. Wang, Z. Chen, *et al.*, "Deep learning in automatic sleep staging with a single channel electroencephalography," *Frontiers in Physiology*, vol. 12, 2021.

[56] J. Pradeepkumar, M. Anandakumar, V. Kugathasan, *et al.*, *Towards interpretable sleep stage classification using cross-modal transformers*, 2022.

[57] H. Phan, K. Lorenzen, E. Heremans, *et al.*, *L-seqsleepnet: Whole-cycle long sequence modelling for automatic sleep staging*, 2023.

[58] C.-H. Lee, H.-J. Kim, Y.-T. Kim, H. Kim, J.-B. Kim, and D.-J. Kim, "Sleepexpertnet: High-performance and class-balanced deep learning approach inspired from the expert neurologists for sleep stage classification," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 6, pp. 8067–8083, 2023.

[59] W. G. Coon and M. Ogg, "Laying the foundation: Modern transformers for gold-standard sleep analysis and beyond," *bioRxiv*, 2024.

[60] W. Zhang, C. Li, H. Peng, H. Qiao, and X. Chen, "Ctcnet: A cnn transformer capsule network for sleep stage classification," *Measurement*, vol. 226, p. 114 157, 2024.

[61] M. H. Rafiei, L. V. Gauthier, H. Adeli, and D. Takabi, "Self-supervised learning for electroencephalography," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 2, pp. 1457–1471, 2022.

[62] C.-H. Lee, H. Kim, H.-j. Han, M.-K. Jung, B. C. Yoon, and D.-J. Kim, "Neuronet: A novel hybrid self-supervised learning framework for sleep stage classification using single-channel eeg," *arXiv preprint arXiv:2404.17585*, 2024.

[63] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*, PMLR, 2020, pp. 1597–1607.

[64] J.-B. Grill, F. Strub, F. Altché, *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *Advances in neural information processing systems*, vol. 33, pp. 21 271–21 284, 2020.

[65] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9729–9738.

[66] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 15 750–15 758.

[67] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," in *International conference on machine learning*, PMLR, 2021, pp. 12 310–12 320.

[68] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 000–16 009.

[69] H. Bao, L. Dong, S. Piao, and F. Wei, "Beit: Bert pre-training of image transformers," *arXiv preprint arXiv:2106.08254*, 2021.

[70] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, "Data2vec: A general framework for self-supervised learning in speech, vision and language," in *International Conference on Machine Learning*, PMLR, 2022, pp. 1298–1312.

[71] Z. Huang, X. Jin, C. Lu, *et al.*, "Contrastive masked autoencoders are stronger vision learners," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[72] X. Jiang, J. Zhao, B. Du, and Z. Yuan, "Self-supervised contrastive learning for eeg-based sleep staging," in *2021 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2021, pp. 1–8.

[73] X. Mai and T. Yu, "Bootstrapnet: An contrastive learning model for sleep stage scoring based on raw single-channel electroencephalogram," in *2021 2nd International Conference on Artificial Intelligence and Computer Engineering (ICAICE)*, IEEE, 2021, pp. 303–308.

[74] D. Kostas, S. Aroca-Ouellette, and F. Rudzicz, "Bendr: Using transformers and a contrastive self-supervised learning task to learn from massive amounts of eeg data," *Frontiers in Human Neuroscience*, vol. 15, p. 653 659, 2021.

[75] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "Wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.

[76] H.-Y. S. Chien, H. Goh, C. M. Sandino, and J. Y. Cheng, "Maeeg: Masked auto-encoder for eeg representation learning," *arXiv preprint arXiv:2211.02625*, 2022.

[77] C.-H. Lee, H. Kim, H.-j. Han, M.-K. Jung, B. C. Yoon, and D.-J. Kim, "Neuronet: A novel hybrid self-supervised learning framework for sleep stage classification using single-channel eeg," *arXiv preprint arXiv:2404.17585*, 2024.

[78] C. Yang, D. Xiao, M. B. Westover, and J. Sun, "Self-supervised eeg representation learning for automatic sleep staging," *arXiv preprint arXiv:2110.15278*, 2021.

[79] E. Eldele, M. Ragab, Z. Chen, *et al.*, "Time-series representation learning via temporal and contextual contrasting," *arXiv preprint arXiv:2106.14112*, 2021.

[80] V. Kumar, L. Reddy, S. Kumar Sharma, *et al.*, "Muleeg: A multi-view representation learning on eeg signals," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2022, pp. 398–407.

[81] J. Ye, Q. Xiao, J. Wang, H. Zhang, J. Deng, and Y. Lin, "Cosleep: A multi-view representation learning framework for self-supervised learning of sleep stage classification," *IEEE Signal Processing Letters*, vol. 29, pp. 189–193, 2021.

[82] X. S. Huang, F. Perez, J. Ba, and M. Volkovs, "Improving transformer optimization through better initialization," in *International Conference on Machine Learning*, PMLR, 2020, pp. 4475–4483.

[83] E. López-Larraz, A. Robledo-Menéndez, E. Jubera-García, *et al.*, "The hogar study: Home-based brain monitoring with a self-managed eeg to study cognitive decline in the aging population," in *17th Clinical Trials on Alzheimer's Disease (CTAD) Conference*, Madrid, Spain, 2024.

[84] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, no. 11, 2008.

[85] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[86] H. van Gorp, I. A. Huijben, P. Fonseca, R. J. van Sloun, S. Overeem, and M. M. van Gilst, "Certainty about uncertainty in sleep staging: A theoretical framework," *Sleep*, vol. 45, no. 8, zsac134, 2022.

[87] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. E. Hinton, "Big self-supervised models are strong semi-supervised learners," *Advances in neural information processing systems*, vol. 33, pp. 22 243–22 255, 2020.

Appendices

Appendix A

Project management

A.1 Software and hardware tools

The deep learning models and training strategies were implemented in Python (v3.9.19) using the PyTorch machine learning library (v2.3.1) as the core framework for algorithm development. In addition, NumPy (v1.26.4) and Scikit-learn (v1.5.1) libraries were also employed to support data processing and evaluation tasks. The supervised pipeline followed (see Section 3.3), along with the deep learning model architecture (see Section 3.1), was already designed and implemented in previous work [45], [46]. They were completely translated from Keras to PyTorch and properly adapted to encompass the self-supervised learning pre-training step. The data-loading logic, augmentation module, and SSL methods (see Sections 3.2 and 3.3) were entirely developed from scratch. Moreover, the storage, management, and sharing of the project's code were carried out on the GitLab platform. On the other hand, the execution of the algorithms and corresponding tests was conducted on the Linux Ubuntu 22.04.5 LTS operating system using an NVIDIA GeForce RTX 3080 Ti GPU with CUDA v12.4.

A.2 Planning

The distribution of the main tasks that have comprised the project over time is reflected in the Gantt chart of Figure A.1.

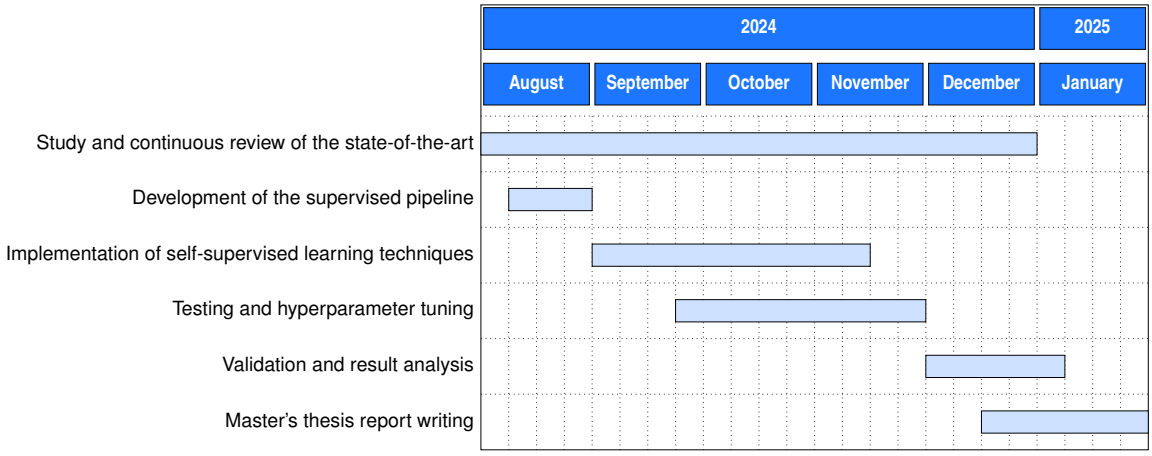


Figure A.1: Gantt chart corresponding to the project schedule.

Appendix B

Artifact detection system

The decoder developed by Bitbrain, illustrated as a key module of the data preprocessing pipeline from Figure B.1, comprises advanced algorithms designed to automatically estimate noise and evaluate the overall quality of EEG signals, effectively identifying artifacts in recordings captured with their wearable textile headband. These algorithms detect a variety of artifacts stemming from different sources, including:

- **High-amplitude artifacts**, which may arise due to:
  - Temporary loss of contact between the EEG sensor and the scalp, caused by physical interaction with the sensor or spontaneous changes in electrode-skin contact.
  - Movement of cables connecting the electrodes to the amplification system.
- **High-frequency noise** (30-45 Hz), potentially originating from:
  - Electrical activity produced by the contractions of muscles, referred to as electromyography (EMG). Common examples include jaw clenching or forehead tension during frowning.
  - AC electrical and electromagnetic interference, often due to inadequate wire shielding.
  - Poor electrode-skin contact.
- **Low-frequency noise** (0.2-4 Hz), primarily caused by the perspiration from skin glands, where small drops of sweat alter the electrical baseline of the electrodes. Intense sweating may even create shorts between electrodes.
- **Empty signal segments** resulting from Bluetooth connection losses.
- **Flat-line segments**, mainly caused by instrumental errors, such as the loss of contact between electrodes and the skin.

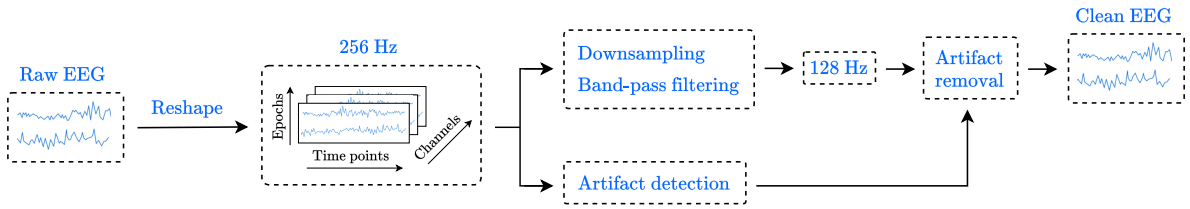


Figure B.1: Data preprocessing pipeline comprising the reshaping, resampling, and filtering steps.

Appendix C

SSL ablation study

This appendix presents the ablation study conducted on the implemented contrastive learning techniques: SimCLR (Table C.1), BYOL (Table C.2), SimSiam (Table C.3), Barlow Twins (Table C.4), and ContraWR (Table C.5). In general, variations in hyperparameters do not significantly impact model performance, where the Barlow Twins approach shows the greatest increase of accuracy (+1.56%) when incrementing the number of pre-training epochs. It is worth mentioning that these tests were carried on reduced versions of the HOGAR and BOAS datasets, each containing 127 and 102 subjects, respectively. For masked prediction techniques BENDR and MAEEG, the proposed values in their respective works have been adopted due to the extensive execution time required, with a few exceptions detailed at the end of Subsection 3.3.2.

Epochs	Batch size	τ	λ	η	Accuracy
100	512	0.1	1^{-4}	0.0001	82.04±0.76
150					81.73±0.67
200					81.95±0.93
250					82.31±0.95
300					82.08±0.63
350					**82.37±0.74**
350	768				81.94±0.72
350	1024				82.18±0.75
350	1280				81.93±0.48
350	1536				81.44±0.68
350		0.05			82.02±0.59
350		0.5			81.27±0.97
350		1.0			81.38±0.73
350			1^{-5}		81.72±0.73
350			1^{-6}		81.76±0.46
350			1^{-7}		82.31±0.42
350				0.00001	80.84±1.04
350				0.001	81.70±1.21

Table C.1: SimCLR ablation study results, where τ denotes the temperature parameter of the NT-Xent loss, λ is the weight decay applied in the optimizer, and η corresponds to the learning rate.

Epochs	Batch size	EMA decay	λ	η	Accuracy
100	512	0.999	1^{-4}	0.0001	80.80±1.14
150					80.25±1.53
200					80.93±0.68
250					80.27±1.10
300					80.87±1.19
350					80.84±0.50
200	768				80.35±1.08
200	1024				80.02±1.03
200	1280				80.52±0.97
200		0.9995			80.93±0.77
200		0.99			80.94±1.34
200		0.9			**81.84±0.88**
200		0.9	1^{-5}		80.15±1.09
200		0.9	1^{-6}		80.54±1.10
200		0.9	1^{-7}		81.15±0.86
200		0.9		0.00001	80.21±1.31
200		0.9		0.001	81.62±0.76

Table C.2: BYOL ablation study results, where λ denotes the weight decay applied in the optimizer, and η corresponds to the learning rate.

Epochs	Batch size	λ	η	Accuracy
100	512	1^{-4}	0.0001	80.55±0.93
150				79.80±0.71
200				80.23±0.96
250				79.71±0.95
300				79.19±1.04
350				79.22±1.22
	768			80.39±0.70
	1024			80.33±0.87
	1280			79.45±0.55
		1^{-5}		80.60±0.94
		1^{-6}		**80.90±1.02**
		1^{-7}		80.49±1.05
		1^{-6}	0.00001	78.91±1.38
		1^{-6}	0.001	80.24±0.81

Table C.3: SimSiam ablation study results, where λ denotes the weight decay applied in the optimizer, and η corresponds to the learning rate.

Epochs	Batch size	λ_{loss}	λ_{opt}	η	Accuracy
100	512	0.005	1^{-4}	0.0001	80.67±1.31
150					81.45±0.93
200					81.11±1.38
250					82.02±0.79
300					81.52±0.93
350					**82.33±0.73**
400					82.08±0.85
450					81.76±0.93
500					81.27±1.11
350	768				81.96±1.07
350	1024				81.16±1.16
350	1280				80.93±1.15
350	1536				81.50±0.94
350		0.007			82.28±0.75
350		0.01			81.25±1.06
350		0.015			81.44±1.05
350			1^{-5}		81.43±1.08
350			1^{-6}		81.16±0.86
350			1^{-7}		81.30±1.23
350				0.00001	79.89±1.10
350				0.001	82.24±0.79

Table C.4: Barlow Twins ablation study results, where λ_{loss} is the hyperparameter trading off the importance between loss terms, λ_{opt} denotes the weight decay applied in the optimizer, and η corresponds to the learning rate.

Epochs	Batch size	EMA decay	λ	η	δ	σ	τ	Accuracy
100	512	0.999	1^{-4}	0.0001	0.5	2.0	2.0	80.04±0.89
150								79.42±0.66
200								78.83±1.11
250								79.57±1.40
300								78.89±1.27
350								78.72±1.21
	768							79.69±1.09
	1024							79.33±1.29
	1280							79.61±1.16
	1536							79.70±1.11
		0.9995						79.39±0.82
		0.99						80.04±1.13
		0.9						79.78±0.75
			1^{-5}					80.46±0.87
			1^{-6}					80.50±1.46
			1^{-7}					80.61±0.88
			1^{-8}					80.12±1.24
			1^{-7}	0.00001				79.22±1.21
			1^{-7}	0.001				79.91±0.92
			1^{-7}		0.1			81.20±0.93
			1^{-7}		0.25			80.62±1.00
			1^{-7}		1.0			79.98±0.98
			1^{-7}		0.1	4.0		81.02±1.06
			1^{-7}		0.1	6.0		80.47±1.24
			1^{-7}		0.1	8.0		80.65±1.03
			1^{-7}		0.1		1.0	**81.30±1.05**
			1^{-7}		0.1		4.0	80.36±0.71
			1^{-7}		0.1		6.0	80.24±0.89

Table C.5: ContraWR ablation study results, where λ denotes the weight decay applied in the optimizer, η corresponds to the learning rate, δ is the empirical margin of the loss, σ is the standard deviation of the Gaussian kernel, and τ is the temperature of the loss.

Appendix D

Electroencephalographic properties of sleep

Expert technicians classify sleep stages following the rules defined by the AASM [20], which outline the wave patterns and EEG frequency characteristics observed during each sleep phase. These can be summarized as follows:

- **Wake**: characterized by the presence of more than 50% alpha frequencies, typically over the occipital region, along with eye-blinking events and rapid or reading-related eye movements (see Figure D.1).

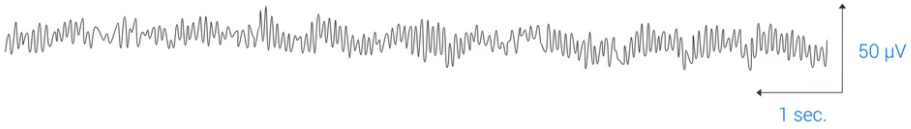


Figure D.1: EEG properties of Wake sleep stage.

- **N1**: exhibits alpha frequency bands that should not exceed 50% of the total spectrum, along with the appearance of low-amplitude theta waves and slow eye movements (see Figure D.2).

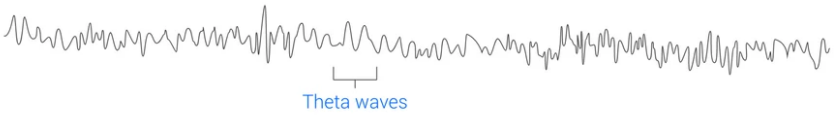


Figure D.2: EEG properties of N1 sleep stage.

- **N2**: shows the so-called sleep spindles (11-15 Hz), composed of waxing and waning waves lasting approximately 0.5 seconds, along with K-complex events, which consist of negative deflections (downstate) followed by a less intense positive deflection (upstate), with an approximate duration of 1 second (see Figure D.3).

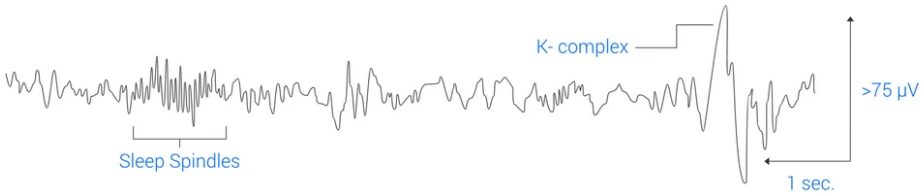


Figure D.3: EEG properties of N2 sleep stage.

- **N3**: delta waves dominate more than 20% of the epoch, with the possible occurrence of sleep spindles and K-complexes (see Figure D.4).

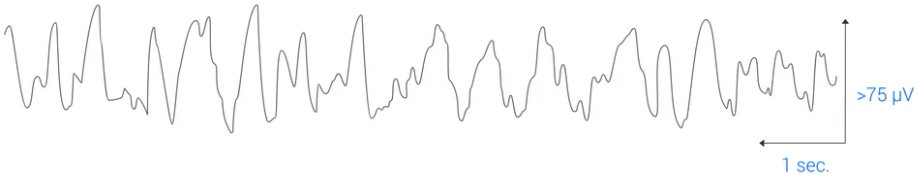


Figure D.4: EEG properties of N3 sleep stage.

- **REM**: experiments rapid, low-voltage activity characterized by EEG flattening in the alpha and theta frequency range. Additionally, it reveals a specific type of theta waves known as sawtooth waves, named for their resemblance to a saw blade, which precede the onset of the rapid eye movements that name this stage (see Figure D.5).

Figure D.5: EEG properties of REM sleep stage.