Titre: Title:	A Portable and Personalized Closed-Loop Brain Stimulation System
Auteur: Author:	Milo Sobral
Date:	2024
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Sobral, M. (2024). A Portable and Personalized Closed-Loop Brain Stimulation System [Master's thesis, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/58325/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/58325/
Directeurs de recherche: Advisors:	Giovanni Beltrame
Programme: Program:	Génie informatique

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

A Portable and Personalized Closed-Loop Brain Stimulation System

MILO SOBRAL
Département de génie informatique et génie logiciel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Génie informatique

Avril 2024

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

A Portable and Personalized Closed-Loop Brain Stimulation System

présenté par **Milo SOBRAL**
en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Benjamin DE LEENER, président
Giovanni BELTRAME, membre et directeur de recherche
Karim JERBI, membre externe

ACKNOWLEDGEMENTS

First of all, I want to thank Giovanni Beltrame for giving me the opportunity to work on this project and for all the help and guidance along the way.

I also wish to thank Yann Bouteiller and Ehsan Marjani, whose contributions were essential to this work. I am grateful to everyone at the MIST lab for their collaborative spirit and friendship, which I cherished during my time there.

Thank you also to Hugo Jourde and Emily Coffey for their continuous support and insightful feedback, both on neuroscience-related questions and beyond.

Finally, I want to thank my family and friends for their unwavering support throughout my studies.

RÉSUMÉ

Les fuseaux de sommeil, brefs pics d'activité cérébrale, jouent un rôle important dans la consolidation de la mémoire et les fonctions cognitives. Leur détection précise pendant le sommeil est essentielle pour les chercheurs qui étudient le sommeil et les processus associés. Cependant, les études traditionnelles sur le sommeil nécessitent souvent un équipement coûteux et complexe, et obligent les participants à se rendre dans des laboratoires spécialisés, ce qui peut être contraignant, onéreux et limiter la participation d'une grande partie de la population.

Ce mémoire propose une solution à ce défi en présentant la conception, le développement et l'évaluation du Portiloop, un système d'enregistrement d'électroencéphalogramme (EEG) portable et pratique, ainsi que des améliorations de son modèle sous-jacent d'apprentissage profond pour la détection des fuseaux de sommeil.

Une innovation clé de ce travail est l'intégration d'une approche d'apprentissage continu dans le modèle d'apprentissage profond du Portiloop. Les modèles d'apprentissage automatique traditionnels nécessitent généralement de vastes ensembles de données pour l'entraînement, et leurs performances peuvent être affectées lorsqu'ils rencontrent des données nouvelles ou inconnues. Les algorithmes d'apprentissage continu répondent à cette limitation en permettant au modèle d'apprendre et de s'améliorer en permanence en fonction des nouvelles informations. Dans le contexte du Portiloop, cela permet au modèle de s'adapter aux habitudes de sommeil individuelles des utilisateurs et d'améliorer sa précision dans la détection des fuseaux de sommeil au fil du temps. Ceci est particulièrement avantageux dans des environnements réels où les variations des habitudes de sommeil propres à chaque participant sont inévitables. L'efficacité de notre approche d'apprentissage continu a été validée en utilisant des données MASS et des enregistrements Portiloop en situation réelle. Cette validation a démontré des améliorations significatives de la performance de détection des fuseaux, à la fois au cours d'une seule nuit et de manière cumulative sur plusieurs nuits.

La conception du Portiloop est prometteuse pour diverses applications de recherche sur le cerveau basées sur l'EEG, pouvant conduire à des découvertes importantes dans notre compréhension du sommeil et du fonctionnement du cerveau.

ABSTRACT

Sleep spindles, transient bursts of brain activity, are associated with memory consolidation and cognitive function. Accurately detecting these events during sleep is crucial for researchers investigating sleep and its related processes. However, traditional sleep studies often require expensive and complex equipment and require participants to visit specialized sleep labs, which can be inconvenient, expensive, and limit the participation of a large portion of the population.

This thesis addresses this challenge by presenting the design, development, and evaluation of the Portiloop, a portable and user-friendly electroencephalography (EEG) recording system, as well as improvements to its underlying deep-learning based model for sleep spindle detection.

A key innovation of this work is the incorporation of a continual learning approach within the Portiloop's deep learning model for sleep spindle detection. Traditional machine learning models typically require large datasets for training, and their performance can suffer when encountering new or unseen data. Continual learning algorithms address this limitation by enabling the model to learn and improve continuously based on new information. In the context of the Portiloop, this allows the model to adapt to individual user sleep patterns and enhance its accuracy in detecting sleep spindles over time. This is particularly advantageous in real-world settings where participant-specific variations in sleep patterns are inevitable. The efficacy of our continual learning approach was validated using MASS data and real-world Portiloop recordings. This validation demonstrated significant improvements in spindle detection performance, both within a single night and cumulatively across multiple nights.

The Portiloop's design holds promise for various EEG-based brain research applications, potentially leading to significant discoveries in our understanding of sleep and brain function.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

EEG	Electroencephalography
CLS	Closed-Loop Stimulation
CLAS	Closed-Loop Auditory Stimulation
BCI	Brain-Computer Interface
TPU	Tensor Processing Unit
ADC	Analog-Digital Converter
NREM	Non-Rapid Eye Movement
DL	Deep Learning
SO	Slow Oscillations
MODA	Massive Online Data Annotation
MASS	Montreal Archive of Sleep Studies
PSASD	Personalized Semi-Automatic Sleep Spindle Detection
FPGA	Field-Programmable Gate Array
PCB	Printed Circuit Board
ADC	Analog to Digital Converter
GPIO	General Purpose Input-Output
GUI	Graphical User Interface
LSL	Lab Streaming Layer
CSV	Comma-Separated Values
MEG	Magnetoencephalography

CHAPTER 1 INTRODUCTION

This thesis documents the research conducted at the MIST Lab from September 2021 to March 2024. The focus of this work was the development of the Portiloop, a novel smart EEG device developed in collaboration with the CLASP Lab at Concordia University. This document is in the format of a thesis by articles, with one submitted contribution presented in Chapter 4.

- Milo Sobral, Hugo Jourde, Emily Coffey & Giovanni Beltrame, “Advancing Closed-Loop Brain Stimulation Continual Learning for Subject-Specific Sleep Spindle Detection,” *IEEE Transactions on Biomedical Engineering*, 2024.

1.1 Context and Problem definition

The brain remains a significant mystery, essential for understanding human cognition, behavior, and overall health. It acts as the body’s control center, managing all aspects of human experience, from basic physiological processes to complex cognitive tasks. By studying how the brain works, we can uncover insights into neurological disorders, mental health conditions, and possibly enhance cognitive functions. Electroencephalography (EEG) plays a key role in researching brain activity, providing a non-invasive way to track the electrical signals produced by neurons in real-time. Its flexibility and ease of use make EEG a vital tool for neuroscientists, facilitating the study of brain function in various states such as sleep, wakefulness, and altered states of consciousness [6].

Studies simply examining EEG data can link certain patterns to brain function, but they are limited as they cannot establish causality. One of the most popular and common ways to delve deeper into the inner workings of the brain is through the method of Closed Loop Stimulation (CLS) [7, 8]. This involves detecting a pattern of interest and stimulating the brain at the same time to study the brain response. Slow Oscillations (SO), known to be related to memory consolidation, have been the subject of many CLS studies [9–11]. Faster oscillations called the sleep spindles are believed to collaborate with SOs but are difficult to target due to the individual variability and short duration [12] [13]. Current CLS tools are slow, expensive, and inflexible in terms of detection and stimulation options. Additionally, they lack the ability to run powerful Deep Learning (DL) models for quick brain event targeting. Sleep studies are difficult due to the costs associated with lab or hospital settings and the tendency for people to sleep poorly away from home. To tackle these issues, the

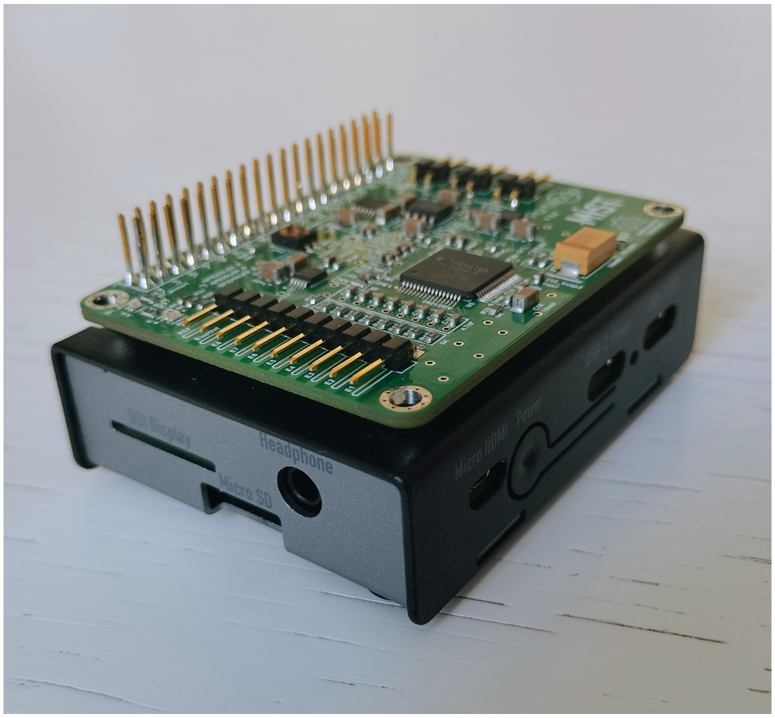


Figure 1.1 Photo of the Portiloop V2.1, the latest revision of the Portiloop at the time of writing.

Portiloop was created—a DL-based device that captures and analyzes EEG data in real-time to detect brain activity patterns for closed-loop stimulation [5].

The Portiloop concept originated in early 2020. Over the subsequent two years, Yann Bouteiller, Nicolas Valenchon, and Xavier L'Heureux built the initial prototypes and developed the first machine learning model designed to detect spindles in real-time on the Portiloop. While these prototypes demonstrated functionality, they were not yet suitable for real-world experimentation. Key considerations for the final Portiloop design included practicality, ease of use for both neuroscientists and research participants in a home setting, and scalability to accommodate multi-device experiments.

While Xavier L'Heureux's work yielded a custom printed circuit board (PCB), its functionality remained unverified due to the absence of a software platform for data acquisition and device interaction. Additionally, the deep learning model designed for sleep spindle detection was limited by its training on a restricted dataset and its inability to adapt to individual subjects.

1.2 Research Objectives

The research work in this thesis pursues the following objectives:

1. **Development of the Portiloop**: make the Portiloop into a fully functional hardware and software platform;
2. **Model Validation**: evaluate the performance of the existing sleep spindle detection model using real-world data collected by the Portiloop;
3. **Spindle Detection Enhancement**: refine the existing model to achieve improved accuracy in sleep spindle detection;
4. **Adaptation**: Develop a method that leverages the subject-specific nature of sleep spindles and enables the model to continually improve its performance over time.

1.3 Methods and Thesis Organisation

Our approach has two areas of focus: the development of the Portiloop ecosystem and the enhancement of its detection capabilities. The development of the Portiloop ecosystem focused on improving both software and hardware to make it usable by neuroscientists and research participants, to ensure scalability, and to provide a streamlined procedure to update and add new functionality to the system. The enhancement of the Portiloop's detection capabilities consisted of refining its existing model for spindle detection to improve its accuracy. This entailed the development of techniques to leverage the subject-specific nature of sleep spindles and ultimately enable the model to continually improve its performance over time.

1.3.1 Portiloop as a platform

Although the Portiloop started as an open-source project, the initial prototypes were not made with usability in mind, especially by people without a training in computer science and embedded systems. Transforming these prototypes into a user-friendly tool for users across disciplines (Objective 1) demanded extensive communication with neuroscientists. This iterative process involved understanding their specific needs and tailoring the device for optimal practicality within their research settings. Effective communication also proved critical for achieving real-world validation of the Portiloop, i.e. its use on human participants and the verification of its capabilities (Objective 2): the neuroscientists needed confidence that a system under heavy development was behaving as intended, requiring the development of a semi-automated signal quality and functionality testbench, described in section 3.4.

To achieve Objectives 1 and 2, we followed the following key steps:

1. **Hardware Assembly**: We began by assembling the Portiloop hardware, building upon the foundational work established by Xavier (reference citation).
2. **Software Development**: A dedicated software platform was developed to facilitate communication between neuroscientists and the Portiloop device.
3. **Test Bench Construction**: To ensure proper functionality prior to releasing new hardware or software revisions, a specialized test bench was constructed to validate Portiloop behavior.
4. **Validation**: A rigorous validation strategy was established to record data from human subjects and assess the Portiloop's ability to record EEG signals and detect sleep spindles.

It is important to acknowledge that this methodology was not strictly linear. As the development process progressed, shortcomings in the initial design were identified, necessitating iterative improvements to both hardware and software components. These refinements, including major revisions, are detailed in Chapter 3.

1.3.2 Improving detection performance

We used a two-pronged approach to improve the Portiloop's sleep spindle detection performance (Objective 3). First, the hardware advancements enabled the utilization of larger, higher capacity models. These models theoretically offer enhanced spindle detection accuracy compared to their predecessors.

Second, a method was devised to facilitate subject-specific model adaptation over time (Objective 4). This approach aims to achieve both improved performance for individual subjects and continuous improvement of the Portiloop detector's over time. However, as with all deep learning endeavors, both methods are heavily reliant on data, a critical resource. To realize the objective of improved model performance, we required a larger dataset with annotated spindle events. The overall methodology employed to fulfill Objectives 3 and 4 comprised the following steps:

1. **Redefining Baselines and Data Annotation**: in collaboration with neuroscientists, we established a new baseline for spindle detection performance;
2. **Model Training**: leveraging the increased data volume, we trained a larger deep learning model aiming for higher detection accuracy;
3. **Adaptation Implementation**: we developed an adaptation framework that enables participant-specific and lifelong performance improvements;
4. **Real-World Validation**: we evaluated the performance of the adaptation framework using real data collected by the Portiloop device.

The resulting system is presented in Chapter 4, which was submitted to the IEEE Transactions on Biomedical Engineering as the paper titled "Advancing Closed-Loop Brain Stimulation: Continual Learning for Subject-Specific Sleep Spindle Detection".

1.3.3 Thesis Organization

This thesis is structured into five chapters that contribute to the research objectives outlined earlier (Section 1.2). Chapter 2 establishes the context of the research by providing a review of relevant literature. Chapter 3 details the transformation of the initial Portiloop prototypes into a user-friendly hardware and software platform suitable for real-world experimentation and discusses the challenges encountered and the solutions implemented to ensure the Portiloop's usability neuroscientists. Chapter 4 presents the specific methods employed to improve the accuracy of sleep spindle detection (Objectives 3 and 4). This latter chapter takes the form of a dedicated research article focused on subject-specific model adaptation, enabling continuous improvement of the Portiloop detector and its performance for individual subjects.

Finally, Chapter 5 summarizes the key findings of the research, discusses the limitations inherent to the current work, and proposes promising avenues for future research that builds upon this thesis.

CHAPTER 2 LITERATURE REVIEW

We look at the current state of the art in three areas related to this thesis: existing portable EEG-capable devices, a background on sleep signal detection and stimulation, and the field of continual learning.

2.1 Portable EEG Hardware

Electroencephalography (EEG) is a non-invasive technique that measures electrical activity in the brain using electrodes placed on the scalp. These oscillations are fundamental to human perception, cognition, and behavior [6]. EEG data has a wide range of applications, including Closed-Loop Systems (CLS) and Brain-Computer Interfaces (BCI), both of which necessitate real-time signal processing with minimal delay. While traditional medical-grade EEG systems can be expensive to maintain and require specialized training, this work focuses on portable and affordable solutions that democratize access to BCI and EEG research for the broader public. Several research-oriented portable systems have been developed [14–16], and commercially available options exist such as OpenBCI [2] and Muse ([1]). OpenBCI focuses on developping research oriented hardware designed for EEG experimentations for both neuroscience and BCI research (see Figure 2.1 (a)). Muse provides a commercially available product which focuses on sleep improvement technology based on closed-loop brain stimulation (see Figure 2.1 (b)). However, these systems often separate data acquisition from processing, which introduces additional delays when real-time responses are required. The Portiloop [5] addresses this limitation by employing the same EEG capture approach (utilizing the same onboard EEG chip) as some existing systems, but it innovatively integrates a Tensor Processing Unit (TPU) for on-device, real-time inference of Deep Learning (DL) models.

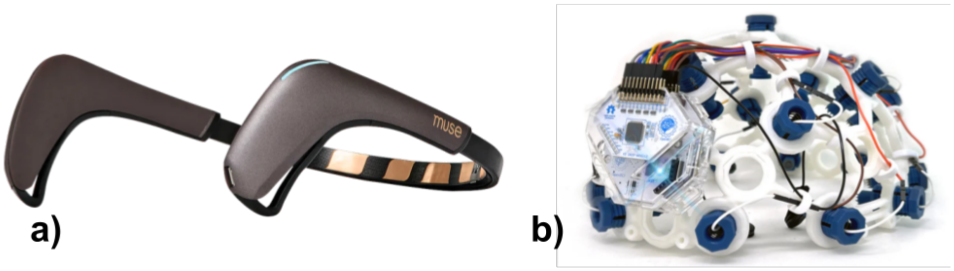


Figure 2.1 (a) Muse 2 headband [1]. (b) OpenBCI Biosensing R&D bundle [2]

2.2 Sleep

Humans need sleep to think; without it, our basic cognitive processes go awry and we cannot function normally and safely [17–19]. But sleep also is thinking; sleep produces cerebral conditions that support unique, coordinated neural activities [20]. Some of the human brain's nocturnal pursuits are about memory consolidation, allowing reorganization of memory traces from their temporary store in the hippocampus to more permanent storage in the neocortex, integrating them with existing knowledge, and extracting gist information [21–24]. Although much research implicates sleep-related brain oscillations in memory consolidation, we do not understand their distinct roles in different learning processes.

Memory consolidation processes seem to depend on several types of oscillatory neural activity: slow oscillations and sleep spindles in non rapid eye-movement (NREM) sleep, and theta band oscillations in rapid eye-movement (REM) sleep. Slow oscillations (SOs), which are highly related to (and perhaps indistinguishable from) K-complexes [25–27], refer to low frequency brain activity <2 Hz6. These high-amplitude fluctuations in cortical and subcortical excitability occur during NREM sleep stages (N2 and N3) and appear to be involved in both memory and non-memory-related processes (e.g. synaptic rescaling, cleaning waste produced during wakeful activity) [22, 28–30]. Sleep spindles are transient (<2.5 s) 11-16 Hz bursts of neural activity that are generated through thalamocortical interactions. They may be evoked by sensory stimulation [31, 32], and are linked to both declarative (i.e., facts and episodes) and procedural (i.e., skills and procedures) memory consolidation processes, particularly when spindles occur in groups [33] (see [34] for a comprehensive review). Both slow oscillations and spindles are also associated with declarative learning [35–37], with some work suggesting it is their co-occurrence or 'coupling' that is critical to memory [38, 39]. Theta band oscillations (4-8 Hz) instead occur both in wakefulness and during REM sleep. They are related to executive functions like working memory [40], consolidation of memories with an emotional component [41–44], and abstraction of rules [45]. Sleep quality, duration, and number and amplitude of SOs and sleep spindles changes with age, and these changes are associated with increased risk of cognitive impairment and decline [46, 47], making improving our mechanistic understanding of sleep oscillations critical for maintaining the health of our ageing population.

Despite general agreement that sleep plays an active role in learning and memory [20, 21, 23, 24, 48], there remain many open questions, such as: what are the unique contributions of each neural event to different types of memory, and by extension, of each sleep stage in which they are found? Is it their number or amplitude that drives successful memory consolidation, or their interrelation, such as precisely timed cross-frequency coupling [24, 39, 44, 49, 50] or clustering into trains [33, 51]? Which brain regions are involved in sleep-dependent memory processes, and how can their function be enhanced?

2.3 Closed-loop Brain Stimulation

The majority of human sleep studies have been correlational in nature, which can leave doubts as to the causal relationships between oscillations and memory function; a non-invasive tool capable of manipulating oscillations is needed. Therefore, we propose to apply closed-loop brain stimulation techniques.

In 2013, Ngo et al. demonstrated a new means of manipulating sleep oscillations in intact humans [31]. By introducing quiet sound bursts at precise moments during sleep, i.e., concurrently with SO up-states (as recorded with EEG), the researchers were able to enhance SOs and sleep spindles. Neural events generated by this technique, which is known as 'closed-loop auditory stimulation' (CLAS), were related to better performance on a simple declarative memory task. CLAS has since yielded performance gains on arithmetic tasks, serial motor reaction time and motor sequence tasks, and spatial navigation tasks (reviewed in [31, 52]), although as with other forms of brain stimulation, effectiveness can vary considerably across subjects [53]. The possibility of causally manipulating sleep and memory processes non-invasively has been hailed as a game-changer in basic human sleep research and has great potential for restoring function in ageing [37, 54]. In fact, new evidence suggests that CLAS over many nights may increase SOs and restore deep sleep in people suffering from Alzheimer's disease [55]. To date, CLAS has focused on stimulating SOs, as these high-amplitude slow waves are relatively easy to detect and stimulate, but the inability to target other frequencies limits both basic science and applications.

CLAS has been proposed to target sleep spindles, which are thought to collaborate with slow oscillations during NREM sleep stages 2 and 3 (N2 and N3) to facilitate the transfer of recently encoded memories into long-term storage [9–11]. This hypothesis is supported by observations of increased spindle density following learning tasks and correlations between age-related changes in spindle activity and overnight memory consolidation gains [56]. Sleep spindles are thought to participate in memory consolidation. However, leveraging CLAS to effectively target sleep spindles presents several significant challenges.

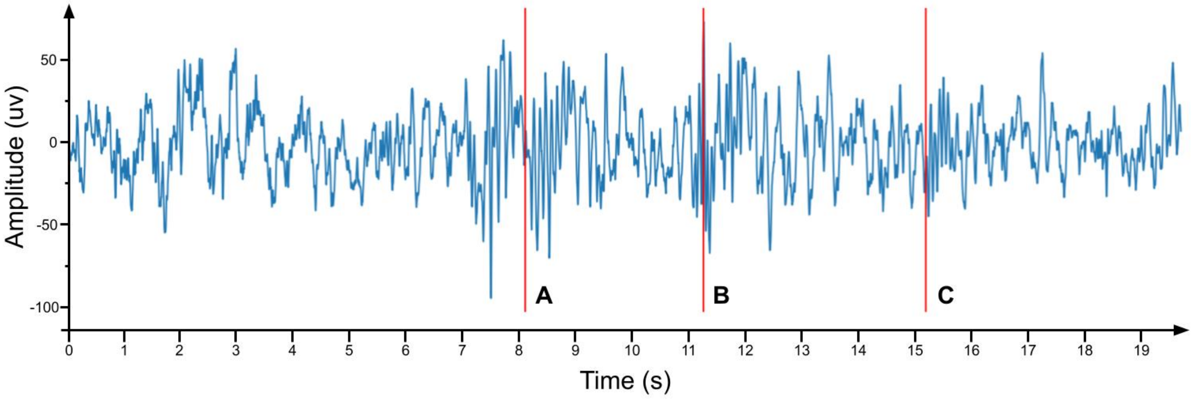


Figure 2.2 Example of 20 seconds of EEG signal where spindles are detected by our online neural network. Red vertical lines show the time step where the model first detects the spindle. The burst of sigma activity can clearly be seen on spindle A but is not as evident in spindles B and C.

2.4 Sleep Spindle Detection

2.4.1 Challenges of Sleep Spindle detection for Closed-loop Brain Stimulation

First, sleep spindles are characterized by their brief duration and high degree of intra-individual variability [12, 13]. This means that the characteristics of spindles for one individual might be different to the characteristics of another individual. While the initial Portiloop iteration demonstrated success in targeting sleep spindles across younger and older adults, its approach to accounting for age-related spindle variations relied on group identification. Separate models were trained on data from younger and older participants [5]. This approach captured average group differences but failed to address the significant variability within each group. Consequently, many individuals likely received suboptimal stimulation.

Ideally, spindle detection should be tailored to each participant's unique physiology. A personalized closed-loop brain stimulation approach targeting neural oscillations could bridge a critical knowledge gap: ensuring consistent and high-quality stimulation across individuals with diverse neural characteristics. This paradigm shift towards adaptive closed-loop and personalized stimulation is predicted to be an important step in advancing fundamental research and optimizing future clinical applications [57–59].

Furthermore, for the purpose of CLS, spindles need to be detected while they are happening [60–62], which necessitates real-time detection algorithms with exceptional temporal resolution and underlines the importance of the ability to adapt to individual differences. The requirement for real-time processing adds an additional layer of complexity to the task.

Also, the manual identification of sleep spindles is a laborious and time-consuming process, with even trained experts facing difficulties in precisely delineating their boundaries. This underscores the importance of developing automated detection methods to streamline this process and reduce associated costs [62].

In the following section, we will examine existing datasets and algorithms that hold promise for facilitating the automated detection of sleep spindles, an important step towards the implementation of effective CLS protocols for memory enhancement.

2.4.2 Datasets

The availability of large EEG datasets is critical for advancing sleep spindle research. However, identifying datasets with comprehensive annotations for spindle activity across entire nights remains a significant challenge.

One prominent example is the Montreal Archive of Sleep Studies (MASS) [63]. While MASS offers high-quality data from 200 nights of polysomnography recordings from various subjects, its spindle annotations are based on the independent labeling of two experts. A spindle is only marked in the dataset if detected by at least one expert. Unfortunately, the agreement between these expert annotations is demonstrably low, resulting in significant uncertainty regarding spindle presence [63].

To address this issue, the Massive Online Data Annotation (MODA) project [62] employed five sleep experts to independently annotate spindles within short (115-second) EEG segments from the MASS dataset. Additionally, confidence scores were assigned to each annotation. These scores were then aggregated to create a more reliable final sleep spindle dataset, boasting an inter-scorer agreement F1-score of 0.67 [62]. Consequently, MODA has been instrumental in the development of offline detection algorithms and the initial stages of the Portiloop model [5, 64].

Despite its contributions, MODA suffers from a key limitation: its annotations only encompass a small number of brief signal sequences per subject within the MASS dataset. This restricts its applicability to studies investigating the impact of adaptation methods on full-night spindle detection. To effectively replicate real-life scenarios where such methods would be employed, ground truth data encompassing spindles across entire nights and various sleep stages is key.

While MODA remains the best currently available expert-annotated dataset for sleep spindles, its restricted scope renders it unsuitable for our investigation. Furthermore, alternative datasets such as [65] and [66] either lack spindle annotations altogether or offer only short sleep EEG sequences, proving equally inadequate for our full-night testing goals.

2.4.3 Algorithms

While significant progress has been made in automatic sleep spindle detection algorithms, most existing approaches operate offline [64, 67–69]. Offline algorithms analyze the complete EEG data from a subject before classifying brain oscillations as spindles. This approach is incompatible with our CLS application, where data is acquired as a continuous stream during sleep recordings.

Previous attempts at online spindle detection have utilized deep learning techniques [70–73]. SpindleNet [70] is the best known of all these approaches and uses a combination of multiple representations (raw EEG signal, bandpass filtered EEG signal and power features in the sigma band) of EEG to learn representations for spindle classification. This results in a large model and requires some extra processing and filtering of the data prior to feeding it into the model. Most deep learning based methods [71–73] are limited by two key factors. Firstly, they were trained on the MASS dataset, where the agreement between each expert's annotations is demonstrably low. Secondly, they did not consider the computational constraints of the Portiloop device, which necessitates lightweight and efficient algorithms for real-time processing.

The Personalized Semi-Automatic Sleep Spindle Detection (PSASD) framework [74] offers a promising approach for subject-specific spindle detection. PSASD tailors its detection process to individual sleep patterns, potentially enhancing accuracy. However, PSASD remains an offline method and requires human intervention for spindle annotation. While it demonstrates potential for subject-specific adaptation, its reliance on offline processing and human interaction renders it unsuitable for real-time implementation on the Portiloop.

Furthermore, sleep staging plays a critical role in spindle detection, as spindles are primarily observed during NREM sleep stages 2 and 3 (N2 and N3). Real-time sleep stage estimation is critical for improving spindle detection accuracy and determining appropriate times for retraining algorithms to optimize subject-specific detection.

Several prior studies have explored real-time sleep staging using deep learning models [75–78]. However, for our specific application, only N2 and N3 stage classification is required as these are the sleep stages where spindles are known to be present. Utilizing separate models for both spindle detection and sleep staging would introduce unwanted latency, hindering CLS functionality. To address these limitations, we propose a novel deep learning model capable of performing both sleep staging (specifically classifying N2 and N3) and spindle detection within a single model, eliminating the need for separate models and minimizing processing delays.

2.5 Continual Learning

Continual learning, also known as online or lifelong learning, is a branch of machine learning dedicated to developing methods for incrementally training deep learning models on continuous data streams. Deep learning, a subfield of machine learning, has become a dominant force in scientific discovery due to its exceptional ability to model complex relationships within massive datasets (LeCun et al., 2015). Deep learning models, also known as artificial neural networks, are comprised of stacked layers that progressively extract features from the input data.

Activation functions, like the rectified linear unit (ReLU) [79], introduce non-linearity within these layers, allowing the network to learn complex patterns. Training these deep models relies on optimization algorithms that iteratively adjust the model's internal parameters to minimize a loss function, ultimately guiding the learning process. Stochastic gradient descent (SGD) and its variants [80] are popular choices for optimization. However, these algorithms can struggle with diminishing learning rates in later training stages. To address this, Adam was introduced [81], incorporating adaptive learning rates for each parameter. Building upon Adam, AdamW [82] further improves training efficiency by decoupling weight decay from the gradient update, leading to faster convergence and potentially better generalization performance.

Within the realm of deep learning architectures, two prominent types stand out for our purposes: convolutional neural networks (CNNs) and recurrent neural networks (RNNs). CNNs excel at identifying local patterns within grid-like data, making them particularly well-suited for tasks like image recognition [83]. RNNs, on the other hand, are adept at handling sequential data, incorporating information from previous steps to analyze sequences like text or time series data. Gated Recurrent Units (GRUs), a type of RNN architecture, address limitations of standard RNNs by introducing gating mechanisms that control information flow within the network, enabling them to better capture long-term dependencies within sequences [84]. This combination of powerful architectures, non-linear activation functions, efficient optimization algorithms like AdamW, and ongoing advancements empowers deep learning to unlock complex patterns within EEG signals, leading to significant breakthroughs in our understanding of brain activity.

In traditional machine learning, the entire training dataset is available upfront. Continual learning, however, updates the model's knowledge progressively as new data arrives, aiming for continual improvement [85]. This approach is particularly suited for our scenario, where the model can progressively adapt to individual sleep patterns over multiple nights without human intervention.

However, a significant challenge arises when applying deep learning models to continual learning tasks: catastrophic forgetting [86]. Deep learning models are highly sensitive to the distribution of their training data. In continual learning settings where this distribution changes over time, the model may prioritize the most recent data, forgetting previously learned information.

Several approaches have been proposed to address catastrophic forgetting in continual learning [87], categorized as:

- Replay-based approaches
- Regularization-based approaches
- Optimization-based approaches
- Representation-based approaches
- Architecture-based approaches

Unfortunately, most of these solutions are not directly applicable to our application due to two key limitations:

1. Task Discretization: Many approaches rely on task discretization, which assumes distinct boundaries between different data distributions [87]. In our case, we aim for the model to learn a generalizable representation across all subjects, rendering task discretization unsuitable.
2. Computational and Memory Constraints: Other approaches, such as replay-based and representation-based methods, are computationally expensive and require significant memory resources [87]. These limitations are impractical for the resource-constrained Portiloop device.

For instance, replay-based approaches necessitate storing a large amount of data for retraining, exceeding the Portiloop's memory capacity and introducing significant training delays [87].

Given these constraints, regularization-based methods emerge as the most promising avenue for our application. These techniques aim to mitigate catastrophic forgetting by carefully controlling model updates as the data distribution evolves [87]. Regularization generally helps prevent overfitting and enhances model generalizability. Considering our hardware limitations, we focus on a simple yet effective form of regularization: averaging.

Studies have shown that weight averaging offers improved accuracy and generalization performance with minimal computational overhead [88]. Averaging the weights of multiple models requires minimal time and memory, making it particularly suitable for our application on the Portiloop device.

CHAPTER 3 PORTILOOP DEVELOPMENT: BUILDING HARDWARE AND SOFTWARE FOR A USABLE PORTILOOP

This chapter details the transformation of the Portiloop from a proof-of-concept prototype into a user-friendly device that meets the research needs of neuroscientists (Objective 1). This transformation necessitated concurrent development efforts in both hardware and software. Continuous communication with neuroscientists ensured alignment with their specific requirements. In this chapter, we show how we fulfilled objectives 1 and 2 as described in section 1.2. The chapter begins with an overview of the initial Portiloop hardware prototype. It then details the hardware improvements implemented over the past few years to address limitations and enhance functionality. Subsequently, the chapter outlines the software development process undertaken to create a user-friendly Portiloop platform. Finally, it describes the methods employed to validate the Portiloop's usability and ensure its proper functioning for real-world research applications.

3.1 Initial State of the Project

The initial Portiloop hardware stemmed from a proof-of-concept design developed by Nicolas Valenchon using a Field Programmable Gate Array (FPGA) board. While this prototype demonstrated the feasibility of the Portiloop concept, it lacked the portability, user-friendliness, and functionality required for real-world research settings. During the summer of 2021, Xavier L'Heureux designed the Portiloop V1, a custom printed circuit board (PCB) incorporating a Google Coral System on a Module (SOM) with a Tensor Processing Unit (TPU). However, component shortages hindered the development of a fully functional board for EEG experiments.

To achieve our objectives with the Portiloop device, we have done the following:

- Hardware Functionality Enhancement: Ensuring the existing hardware became fully operational.
- User Interface Development: Creating an intuitive interface offering all necessary functionalities for neuroscientists.
- Test Bench Validation: Validating device performance on a dedicated test setup.
- Real-World Subject Validation: Evaluating device functionality with human subjects in real-world experiments.

Throughout this development process, continuous collaboration with neuroscientists from multiple labs ensured that the Portiloop's design and functionalities aligned with their research needs. Their valuable feedback played a crucial role in transforming the Portiloop into a usable research tool.

3.2 Hardware

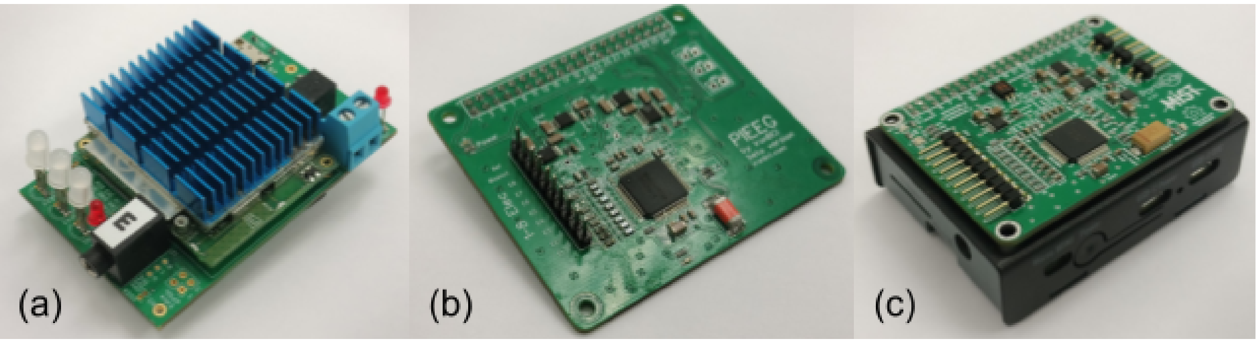


Figure 3.1 **Evolution of the Portiloop hardware**. (a) PCB of the Portiloop V1, (b) EEG shield designed by PiEEG, (c) complete Portiloop V2.

3.2.1 Portiloop V1

The Portiloop V1 (see Figure 3.1(a)) was designed as a custom PCB that provided all necessary interfaces for a Google Coral Mini SOM. To facilitate EEG processing, the PCB incorporated a Texas Instruments ADS1299 chip [89]. This design placed the responsibility for most board functions, including power supply and audio output (critical for Closed-Loop Systems (CLS)), on the custom PCB.

The initial power solution for the Portiloop V1 utilized multiple 3.7V lithium batteries. The device required approximately 7 volts to function, so a method was devised using four batteries in series-parallel configuration within plastic battery holders. This approach aimed to deliver sufficient power for extended use. However, it necessitated balanced charging of each lithium cell, as uneven charge distribution could lead to current inversion and potential burning of battery holder connectors. To mitigate this risk, electrical diodes were added between each cell, allowing current flow in only one direction. While this battery configuration successfully powered the Portiloop V1 for initial experiments and software development, several limitations became evident:

- Flimsy Construction: Although functional, the hand-soldered battery holders were fragile and susceptible to damage during transport.

- Limited Power Management: The design lacked a power management circuit, making it impossible to monitor battery charge state. This made it cumbersome to track individual cell recharge needs.
- Thermal Issues: The Google SOM design exhibited known thermal issues, causing device crashes during prolonged intensive workloads, such as running the Portiloop detection algorithm throughout entire nights. Attempts to improve thermal performance with a heatsink proved inadequate.

Despite these limitations, the Portiloop V1 fulfilled its purpose by validating the functionality of our design and model. It was even employed in two separate neuroscience labs to record sleep data (over 50 naps) from various subjects. However, the aforementioned shortcomings hindered the Portiloop V1's ability to fully meet the project's capability requirements. This led to the development of the Portiloop V2.

3.2.2 Portiloop V2

The Portiloop V2 leverages the same core components as its predecessor, the Portiloop V1: the Texas Instruments ADS1299 analog-to-digital converter (ADC) chip [89] and a Google Coral Mini, a single-board computer variant of the SOM used in V1.

The Portiloop V2 drew inspiration from an open-source design by PiEEG [4] (see Figure 3.1(b)). This design utilizes a simplified board to interface electrodes with the ADS1299 converter, which can then connect to any board equipped with a 40-pin GPIO header. Building upon the original design, we modify the PCB to achieve a more practical size and accommodate readily available components during construction (see Figure 3.1(c)).

Given the requirement for a TPU to facilitate real-time deep learning model execution, we opt for a Google Coral Mini as the core of the Portiloop V2. This cost-effective development board houses processing units comparable to those in the SOM used within the Portiloop V1. Transitioning to a single-board computer addresses both critical limitations identified in the Portiloop V1. The Coral Mini is powered via USB, enabling the use of any USB charger or portable battery. This significantly enhances device safety, practicality, and battery monitoring, as standard portable phone batteries can now power the Portiloop for a full night's experiment. Furthermore, while possessing less processing power than its SOM counterpart, the Coral Mini avoids the thermal issues that plagued the Portiloop V1, eliminating the frequent crashes experienced with the previous version. The portability, ease of use, and real-time deep learning-based detection capabilities for extended duration (entire nights) make the Portiloop V2 a device that fully meets the project's capability requirements.

3.3 Software

Beyond functional hardware, the Portiloop's utility hinges on its software components. This section details the various software elements of the Portiloop, along with the engineering decisions made during development and the identified limitations. This chapter delves into the core functionalities of the Portiloop system.

We begin by exploring the on-device processing pipeline, addressing real-time data acquisition, pre-processing, model execution, and the optimizations undertaken to balance efficiency and accuracy. Following this, we explore the user interface, the intuitive control panel designed to facilitate interaction with the system. Next, we introduce an innovative offline emulator, a valuable tool that empowers researchers to conduct experiments using pre-recorded data, even in settings without live EEG acquisition capabilities. Finally, we examine the Portiloop's commitment to user-friendliness through its open-source nature and streamlined installation process.

3.3.1 Processing Pipeline

This chapter explores the intricacies of the Portiloop's on-device processing pipeline, detailing real-time EEG data acquisition, pre-processing steps, model execution on the embedded platform, and the challenges encountered in optimizing efficiency and accuracy.

On-device processing involves real-time communication with the ADS1299 chip to acquire EEG data. This data is then processed and filtered before being passed through the spindle detection model. While Python is not known for its processing efficiency on embedded devices, it was chosen for the initial development of the processing pipeline due to its familiarity to the development team and its rapid iteration capabilities. Initial tests using Python with multiprocessing tools demonstrated the ability to capture real-time EEG data at up to 1 kHz. This sampling rate proved sufficient for our purposes, considering the model's training on 250 Hz data. However, future iterations targeting enhanced efficiency or higher sampling rates may necessitate exploring alternative programming languages or frameworks optimized for embedded systems.

A significant challenge encountered during processing pipeline development was determining the optimal configuration for the ADS1299 analog-to-digital converter (ADC) chip. Notably, the ADS1299 offers the option to utilize a bias drive amplifier. This amplifier electronically calculates the average signal across multiple electrodes, inverts it, and injects it back into the patient's body through a dedicated bias electrode. This technique serves to remove common noise across all channels, thereby improving data quality. To address this configuration decision, we employed an iterative approach involving experimentation and discussions to evaluate data quality. Consequently, the software is designed to provide users with the option to enable the bias drive amplifier and select the channels contributing to the derivation of the bias signal.

Another challenge involves preparing the deep learning model for execution on the Coral Mini's TPU. This hardware component is optimized for TensorFlow Lite frameworks and requires models using a specific data format (INT8 precision) for inference. Since our model was originally developed using the PyTorch framework, a two-step process is necessary. First, we convert the model weights to a TensorFlow-compatible format. However, certain model layers present compatibility issues with standard conversion tools. To address this, we recreate an identical model architecture in TensorFlow and manually transfer the weights layer by layer.

The second step involves model quantization, a technique that reduces the data format precision of the model's weights (from training precision to inference precision). In simpler terms, quantization aims to represent numerical values within the model using less data (e.g., converting from 64-bit to 8-bit representation). This optimization technique reduces memory usage and computational demands while striving to maintain acceptable model accuracy.

For our implementation on TensorFlow Lite, we employ a method called post-training quantization. This approach leverages a representative dataset of real-world data to define the range and scaling factors for each weight within the model. The weights are then converted using this formula to achieve the desired data format (INT8) compatible with the TPU:

$$\text{Int8\_weight} = \frac{\text{Float64\_weight} - \text{min\_range}}{\text{scale\_factor}}, \tag{3.1}$$

where min_range is the minimum of the range found for that weight on the inference dataset, and scale_factor is the overall scaling factor found over the dataset.

Then, we ensure that the model's performance remain comparable after quantization, acknowledging the inherent precision loss associated with converting weights from float64 to int8 formats. Our evaluation yields positive results, demonstrating an accuracy close to 97% relative to the pre-quantized model. In simpler terms, the model's classification behavior remains consistent for nearly 97% of the evaluated inputs, indicating an acceptable trade-off between efficiency and accuracy.

The final challenge within the processing pipeline involves ensuring accurate sound stimulation for detected sleep spindles. This necessitates minimizing delays throughout the sound delivery process to prevent missed stimulations during these brief events. The specific sound

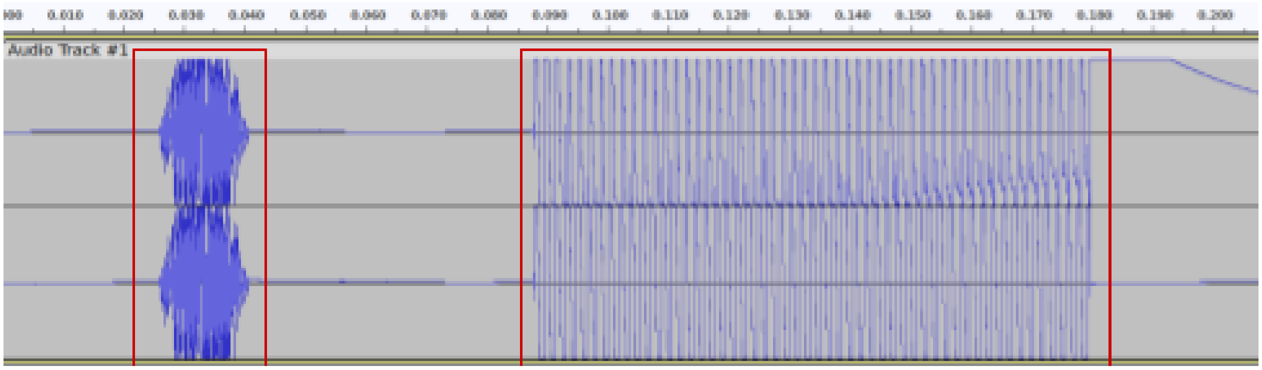


Figure 3.2 Sound waves captured with Audacity [3] showing the different sounds produced by the Portiloop for CLAS.

stimuli used can vary between studies but typically consist of short (15-100 milliseconds) bursts of pink noise with tapered on/off periods (see Figure 3.2 for examples).

To achieve this time-sensitive delivery, a separate process was employed for sound output. This approach leverages the multi-core processing capabilities of the Google Coral Mini to prioritize sound delivery, ensuring it does not interfere with other processing tasks. Verification of this functionality is elaborated upon in Section 3.4.

3.3.2 Graphical User Interface

As a standalone computer, the Portiloop necessitated the development of a user interface for communication, experiment initiation, and monitoring. The interface design prioritized the following key requirements:

1. Platform Independence: The interface should eliminate the need for software installation on the control device. This approach aimed to streamline user experience by avoiding versioning issues and allowing immediate use upon connection, without requiring additional external software.
2. Graphical User Interface (GUI): A graphical interface was essential to cater to users with limited command-line experience. This decision held particular importance for at-home experiments where subjects would interact directly with the Portiloop.
3. Experiment Flexibility: The interface was designed for easy modification to accommodate diverse experimental paradigms.
4. Data Redundancy: The interface needed to ensure data redundancy, safeguarding against data loss in case of device failure during an experiment.

To establish user communication and experiment management functionalities, two primary interface components are developed: a communication interface and a Graphical User Interface (GUI). The communication interface focuses on establishing a connection to the Portiloop device while adhering to the aforementioned requirements.

Our approach involves configuring the Portiloop as a Wi-Fi access point. This eliminates reliance on external networks and enables connection from any Wi-Fi-enabled device. The Coral Mini's dual Wi-Fi chip antennas allows for simultaneous operation of an access point (for user connection) and a client connection (for internet access crucial for development and code updates via Git version control). While susceptible to interference in specific environments like busy labs or close proximity of multiple Portiloops, this approach offers the greatest versatility by enabling experiment execution independent of external communication infrastructure.

To ensure data redundancy as outlined in the initial interface requirements (requirement 4), two data storage options were implemented:

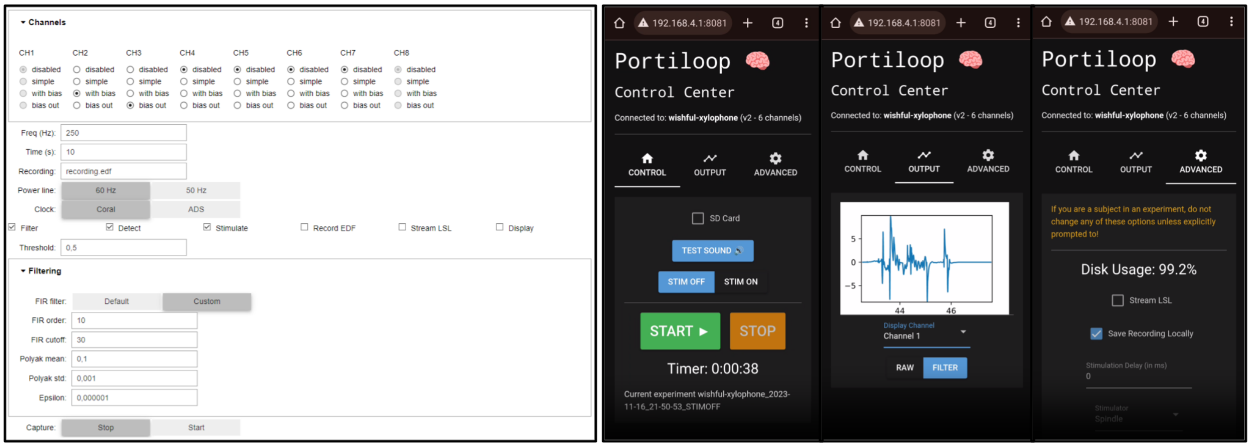
- Lab Streaming Layer (LSL): This framework facilitates real-time streaming of time-series data over a network connection. This approach allows for data storage on an external computer, providing a network-based backup.
- Local Storage (CSV): The Portiloop can also store data locally as comma-separated values (CSV) files. However, this method necessitates the use of an external SD card with sufficient storage capacity to accommodate full-night recordings.

This dual-pronged storage strategy safeguards against data loss by creating a network-based copy (LSL) and a local backup (CSV) on the Portiloop itself.

Desktop GUI

The Portiloop's Graphical User Interface (GUI) development heavily emphasizes user-centered design principles. Frequent interactions with target scientists ensured that the interface prioritized functionalities most relevant to their workflows and established intuitive interactions between its components. Two distinct GUI versions were explored throughout the development process.

Our initial approach utilizes the device as a Jupyter server [90] and presents the GUI within a Jupyter notebook (see Figure 3.3(a)). Users can access this interface by simply connecting to the Portiloop's Wi-Fi access point and navigating to a designated IP address in their web browser. This approach offers functionalities like file browsing on the device and a customizable Jupyter notebook specifically designed for Portiloop control. The GUI is built using widgets within Jupyter cells, allowing for user interaction and experiment control.



▾ Channels

CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8
● disabled	○ disabled	○ disabled	● disabled	● disabled	● disabled	● disabled	● disabled
○ simple	○ simple	○ simple	○ simple	○ simple	○ simple	○ simple	○ simple
○ with bias	● with bias	○ with bias	○ with bias	○ with bias	○ with bias	○ with bias	○ with bias
○ bias out	○ bias out	● bias out	○ bias out	○ bias out	○ bias out	○ bias out	○ bias out

Freq (Hz): 250
Time (s): 10
Recording: recording.edf
Power line: [✓ 60 Hz] 50 Hz
Clock: [✓ Coral] ADS

☑ Filter ☑ Detect ☑ Stimulate ☐ Record EDF ☐ Stream LSL ☐ Display

Threshold: 0,5

▾ Filtering

FIR filter: Default [✓ Custom]
FIR order: 10
FIR cutoff: 30
Polyak mean: 0,1
Polyak std: 0,001
Epsilon: 0,000001

Capture: [✓ Stop] Start

(a) Notebook based Interface

192.168.4.1:8081

Portiloop 🧠

Control Center

Connected to: **wishful-xylophone** (v2 - 6 channels)

☐ SD Card

Timer: 0:00:38

Current experiment wishful-xylophone_2023-11-16_21-50-53_STIMOFF

Display Channel: Channel 1

If you are a subject in an experiment, do not change any of these options unless explicitly prompted to!

Disk Usage: 99.2%

☐ Stream LSL

☑ Save Recording Locally

Stimulation Delay (in ms): 0

Stimulator: Spindle

(b) NiceGUI based Interface

Figure 3.3 **Both versions of the Portiloop Graphical User Interface.** (a) shows an early version of the Jupyter notebook based GUI (desktop interface), (b) shows the NiceGUI based interface (mobile interface) and its different tabs.

Additionally, real-time EEG data visualization is integrated within the Jupyter cell during operation.

However, this solution demonstrated limitations for extended experiments. Jupyter notebooks are not optimized for long-term server operation, and frequent connection drops or server restarts were observed. Furthermore, Jupyter widgets lacked the development flexibility required for complex GUIs, resulting in increasingly convoluted code as functionalities were added. Although convenient for developers, the Jupyter server interface presented usability challenges for non-technical users. Navigating to the correct file, initiating specific Jupyter notebook cells, and understanding the extensive set of constantly displayed options proved cumbersome for researchers.

3.3.3 Mobile GUI

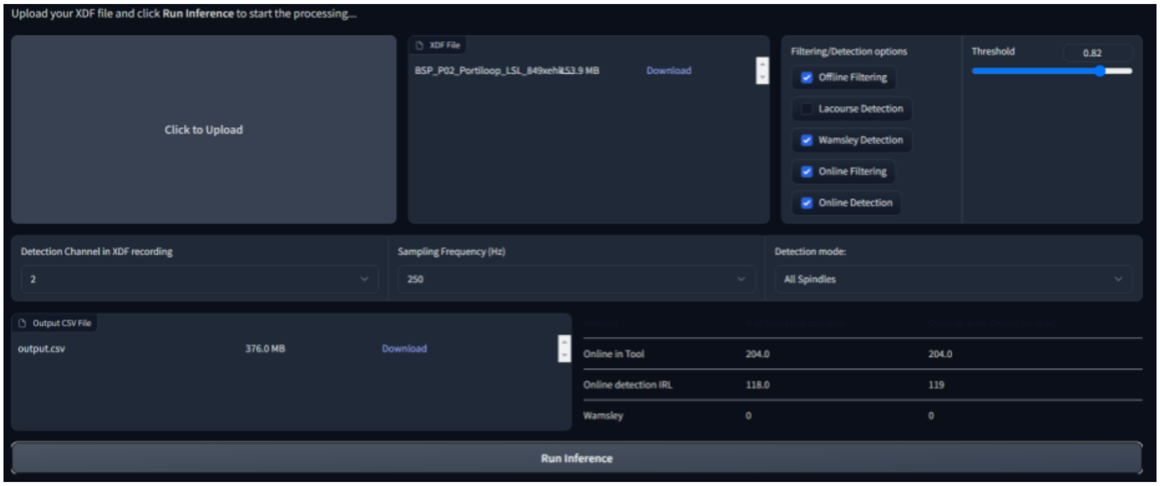
In response to the limitations identified within the initial Jupyter-based GUI version, a second iteration was developed leveraging the NiceGUI framework [91]. This framework streamlines the creation of web applications and offers pre-built components for constructing complex GUIs. Similar to its predecessor, this interface is accessible by connecting to the Portiloop's Wi-Fi network and navigating to a designated IP address within a web browser.

The primary focus of this second version is to enhance user-friendliness, particularly for at-home experiments conducted by subjects with minimal Portiloop experience (see Figure 3.3(b)). Recognizing the prevalence of smartphone usage, the GUI prioritizes mobile-device optimization, catering to the practical scenario of initiating experiments before bedtime. While the interface retains the capability for advanced options and real-time data visualization, the core emphasis remains on simplicity. The main page displays only necessary functionalities, facilitating intuitive control for anyone operating the Portiloop. This user-centered design philosophy has demonstrably improved the Portiloop's usability, with successful data collection reported from multiple subjects conducting at-home experiments.

3.3.4 Offline Tools to emulate the Portiloop

A critical factor influencing model performance is the selection of an appropriate threshold for spindle detection. The machine learning model outputs a value between 0 and 1, and spindle presence is determined by whether this value surpasses a predefined threshold. While an intuitive threshold might be 0.5, studies conducted concurrently with model development demonstrated that this was often not optimal [5].

Prior to the development of the automatic threshold adaptation model presented in Chapter 4, neuroscientists relied on previously recorded Portiloop data to identify the most effective threshold. To streamline this process, an offline emulation tool replicating the Portiloop's functionality was developed and deployed. This tool allows researchers to execute the same data while adjusting various parameters to analyze their impact on detection performance.



Upload your XDF file and click **Run Inference** to start the processing...

XDF File: BSP_P02_Portiloop_LSL_849xeh853.9 MB Download

Filtering/Detection options
☑ Offline Filtering
☐ Lacourse Detection
☑ Wamsley Detection
☑ Online Filtering
☑ Online Detection

Threshold: 0.82

Detection Channel in XDF recording: 2

Sampling Frequency (Hz): 250

Detection mode: All Spindles

Output CSV File: output.csv 376.0 MB Download

Method	# of Detected spindles	Overlap with Online (in tool)
Online in Tool	204.0	204.0
Online detection IRL	118.0	119
Wamsley	0	0

Figure 3.4 Screenshot of the offline tool used to test and compare the Portiloop's behavior on the same data with different settings. Here, this experiment was run with a lower threshold than the recorded data which results in more spindles found (204 vs 118).

The emulation tool is hosted on a Hugging Face space [92], an online platform featuring machine learning demos with user-friendly graphical interfaces to ensure it fit the same requirements for usability of our software detailed in section 3.3.2. The tool additionally computes spindle detection using various offline algorithms (Lacourse's A7 [67] and Wamsley's spindle detection method [64]) on the same data, enabling rapid comparison of detection results. This tool not only facilitates the determination of optimal thresholds but also allows for the development and validation of experimental options, such as delaying stimulation based on specific detection characteristics. This offline tool can be found here and is shown in Figure 3.4.

3.3.5 Quality-of-Life Features for Easy Setup and Use

Recognizing that the Portiloop primarily targets users with limited software experience, a core focus during development was user-friendliness. This emphasis led to the creation of numerous quality-of-life features designed to simplify setup and operation for non-technical users.

In line with its open-source nature, all Portiloop hardware schematics and software code are freely available online. Additionally, users can opt for pre-assembled hardware components, further minimizing technical hurdles.

To ensure a seamless setup experience, a single-command installation procedure managed by a dedicated Makefile was developed. This script, when executed on the Portiloop, performs the following functions:

- **Network Configuration**: Prompts the user for desired Wi-Fi network name (SSID) and password to configure all networking interfaces. this includes running all the commands to allow the Portiloop to serve as an access point and connect to the Internet simultaneously, as well as WiFi pass-through to any connected device.
- **Software Dependency Installation:** Installs all necessary dependencies to run the Portiloop software, supporting both desktop and mobile versions of the GUI.
- **Service Startup:** Configures services to ensure automatic execution of the software upon system startup. This makes usage of the Portiloop easier by simply requiring the device to be turned on to be able to access the desired GUI.

This streamlined installation process empowers users with limited technical backgrounds to effortlessly set up their Portiloop devices. It further facilitates the scaling of experiments to multiple Portiloops, enhancing the project's accessibility for a broader audience.

3.4 Validating the Portiloop

Validating that both the data quality and spindle detection performance of the Portiloop were working properly was done over time by two different groups (Julien Doyon's at the Neuro at McGill and Emily Coffey's CLASP at Concordia) by collecting data on human subjects. The confident deployment of the Portiloop in real-world subject studies and across diverse laboratories necessitated the development of a comprehensive testing methodology. Given the inherent complexity of the Portiloop, encompassing both software and hardware components with lab-based device preparation, a multitude of potential failure points exist. To ensure the delivery of fully functional devices to neuroscientists, encompassing all functionalities from EEG recording hardware to sound output, a standardized test suite is established. This section details this test suite.

To guarantee the execution of these tests with every new code update, a testing manifest is integrated into each release. This manifest documents the specific tests conducted and their corresponding results, providing a transparent record of device functionality and makes sure that all the following functionalities are tested:

- Sound output
- Sound delay
- Signal quality
- Desktop GUI
- WiFi interface

To mitigate potential hardware failures and ensure consistent device functionality, a dedicated test bench was constructed within the MIST laboratory. This bench facilitates the systematic testing of all hardware contact points within the Portiloop. The test bench's primary function is to evaluate the quality of signals acquired by the Portiloop and the fidelity of sound output.

Signal quality assessment is conducted using a signal generator (Figure 3.5, Box 1). This instrument generates sinusoidal electrical signals. Due to the significant difference in scale between the signal generator's minimum amplitude (1 mV) and typical EEG signal ranges (microvolt level), a breadboard with resistors is employed to attenuate the signal amplitude before feeding it into the Portiloop. By connecting the signal generator to each channel of the Portiloop, we are able to verify signal quality on individual channels. Figure 3.6 illustrates the results of such a test, where the signal generator is sequentially connected to each of the four available channels on a specific device. The output clearly demonstrates proper functioning of channels 1, 3, and 4, while channel 2 exhibits a defect, as evidenced by the distorted sine wave.

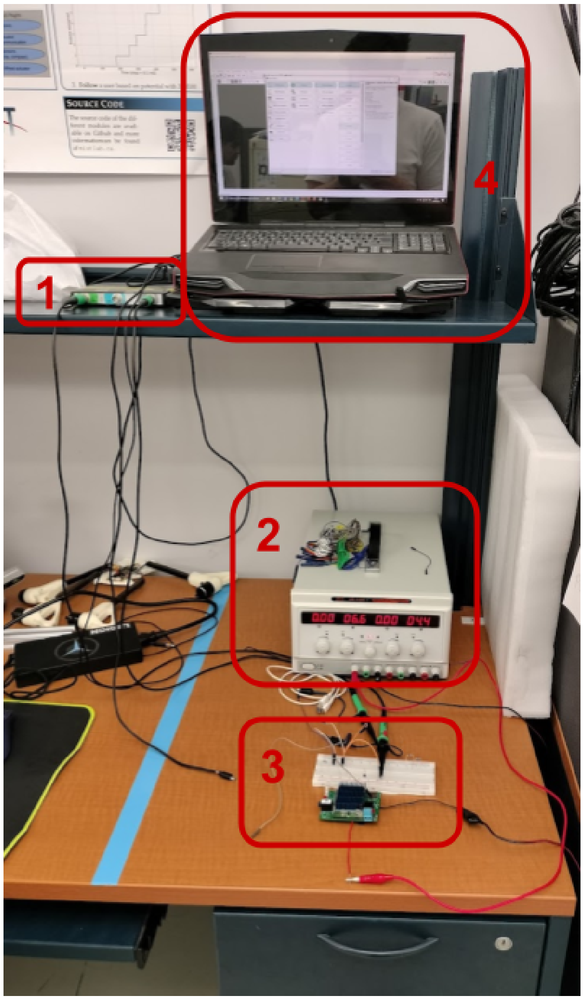


Figure 3.5 **Portiloop test bench at MIST lab.** Box 1 is the signal generator, box 2 is the power generator used to power the Portiloop during tests, box 3 shows the breadboard used to divide the signal to simulate EEG, box 4 is the computer used to record audio and check the audio ouput of the Portiloop.

The evaluation of the Portiloop's sound output employs a direct connection from the device's 3.5 mm audio jack to a computer using a male-to-male cable. The computer (Figure 3.5, Box 4) running Audacity software [3] recorded the sound directly. This configuration facilitates testing of sound production fidelity by the Portiloop using pre-recorded audio containing spindle events identified by our model. Additionally, this setup enables the detection of potential delays between spindle detection and corresponding sound stimulation.

Presenting a significant challenge, software testing aims to comprehensively evaluate all possible combinations of GUI options within the Portiloop notebook. To address this challenge and ensure efficient testing, a script was developed that harnessed the capabilities of Se-

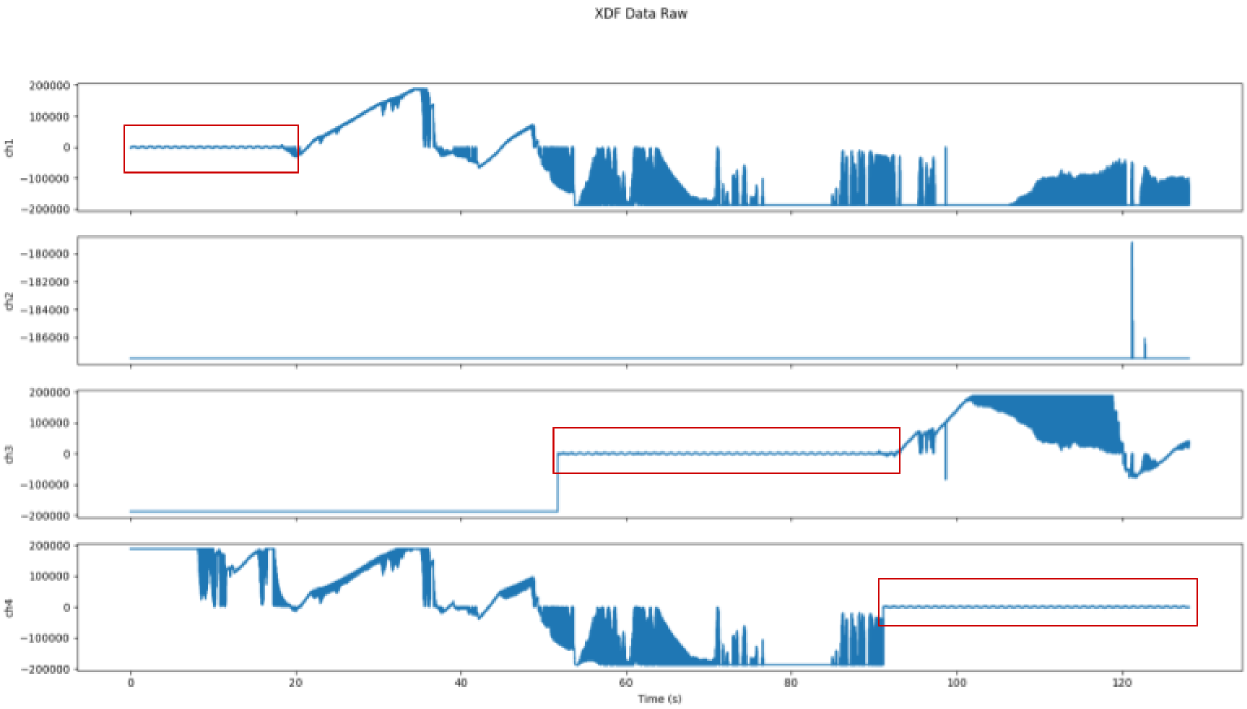


Figure 3.6 **Output of the Portiloop signal test**. Each plot represents one of the Portiloop's recording channel which are sequentially connected to an electrode sending simulated EEG signal in the form of a sine wave. We can see here that channel 2 is broken on that specific device.

lenium [93]. Selenium is an established open-source framework designed for automating interactions with web browsers, particularly for functional testing of web applications. By leveraging Selenium, the script facilitates automated UI interaction and verification of proper output across all Portiloop options. This approach significantly improves testing efficiency compared to manual selection of each option. Consequently, the script enables the detection and rectification of UI and underlying functionality bugs prior to the deployment of new Portiloop software versions.

CHAPTER 4 ARTICLE 1: ADVANCING CLOSED-LOOP BRAIN STIMULATION: CONTINUAL LEARNING FOR SUBJECT-SPECIFIC SPINDLE DETECTION

The contribution of student Milo Sobral involved design, implementation, experimentation, analysis and bibliographical research in the writing of the following article titled: **Advancing Closed-Loop Brain Stimulation: Continual Learning for Subject-Specific Sleep Spindle Detection**, submitted to IEEE transaction on Biomedical Engineering on the 12/04/2024. Authors: Milo Sobral (85%), Hugo Jourde (5%), Emily Coffey (5%), Giovanni Beltrame (5%).

4.1 Abstract

Objective: Personalized closed-loop brain stimulation, in which algorithms used to detect neural events adapt to a user's unique neural characteristics, may be crucial to enable optimized and consistent stimulation quality for both fundamental research and clinical applications. Precise stimulation of sleep spindles—transient patterns of brain activity that occur during non rapid eye movement sleep that are involved in memory consolidation-presents an exciting frontier for studying memory functions; however, this endeavor is challenged by the spindles' fleeting nature, inter-individual variability, and the necessity of real-time detection. *Methods:* This paper introduces an approach to tackle these challenges, centered around a novel continual learning framework. Using a pre-trained model capable of both online classification of sleep stages and spindle detection, we implement an algorithm that refines spindle detection, tailoring it to the individual throughout one or more nights without manual intervention. *Results:* Our methodology achieves accurate, subject-specific targeting of sleep spindles and enables advanced closed-loop stimulation studies. *Conclusion:* While fine-tuning alone offers minimal benefits for single nights, our approach combining weight averaging demonstrates significant improvement over multiple nights, effectively mitigating catastrophic forgetting. *Significance:* This advancement represents a crucial step towards personalized closed-loop brain stimulation, potentially leading to a deeper understanding of sleep spindle functions and their role in memory consolidation. It holds the promise of deepening our understanding of sleep spindles' role in memory consolidation for cognitive neuroscience research and therapeutic applications.

4.2 Introduction

Patterns of electrical brain activity can be captured non-invasively through sensors placed on the scalp, a technique which is known as electroencephalography (EEG). In the century since its invention, EEG has been used to document the full array of oscillatory frequency ranges and transient events produced by the brain, and link them to brain states, perception, cognition, and behaviour. While such correlational studies link EEG patterns to cognitive functions, they cannot establish causality, prompting interest in non-invasive brain stimulation. These techniques enable researchers to safely study how experimentally-altered neural activity causally affects behaviour in the healthy brain [94]. Closed-loop brain stimulation (CLS), in which stimulation aims to enhance or disrupt naturally-generated neural events, is particularly well-suited to the task [95]. It offers promising avenues for understanding the roles of endogenous brain activity [96] and in some cases, improving quality of life for patients with neurological disorder or age-related cognitive decline by restoring faulty processes [97, 98].

Closed-loop brain stimulation aimed at neural oscillations that are specific to the sleep state represents a particularly compelling research frontier [8]. Events known as slow oscillations and sleep spindles are closely associated with memory, and decreases in the frequency of their occurrence and amplitude parallel cognitive decline in older adulthood [99]. While there is now considerable evidence that closed-loop brain stimulation to slow oscillations has benefits for sleep and memory (see [9, 10]), little work has focused on faster neural oscillations. Sleep spindles are transient neural oscillations with frequencies between 11 and 16 Hz ('sigma band') lasting 0.5 to 2.5 seconds and observed in non-rapid eye movement (NREM) sleep stages 2 and 3 (N2 and N3). They are thought to be involved in reactivating recently learned memories and transferring them to long-term memory in a process known as memory consolidation (see [100] for a primer). Evidence supporting the role of sleep spindles in memory consolidation includes increased spindle density after learning and correlations between age-related spindle changes and overnight performance gains [56]. Precise closed-loop stimulation of these events holds potential for further exploring and enhancing memory function. However, accurately targeting spindles in real-time presents significant challenges due to their short duration, and that their characterization and identification in literature depends on measurements that encompass their temporal evolution (i.e., they are best identified once they have terminated, but to target them with stimulation, they must be identified beforehand) [61, 62]. As an additional complication, spindle properties exhibit considerable variability among subjects, and particularly so in older adults [12].

In our previous work, we developed an EEG-based deep learning-based tool for closed-loop brain stimulation, and validated its application to targeting sleep spindles [5]. The Portiloop enables the detection of patterns of interest for research purposes and delivers low-latency stimulation responses. It features both an in-built sound card to deliver auditory stimulation directly, and can be configured to trigger external stimulation devices (e.g., transcranial magnetic stimulation), according to research needs. In its latest iteration, the device combines a custom EEG printed circuit board (PCB) with a neural accelerator (Google Coral Mini with TPU), boasting increased memory capacity and computing power compared to earlier versions. This enhancement allows for the implementation of more complex algorithms and larger models, thereby improving performance and versatility.

The Portiloop's initial iteration achieved some success in targeting spindles both in younger and older adults [5]. A simple approach to tailoring detection for age-related differences in spindle characteristics would be to use a classifier on them which has been trained on a group of people with shared characteristics. This approach incorporates average group differences in spindles, but does not account for the considerable variability found within those groups, meaning that many subjects would be stimulated sub-optimally. Ideally, spindle detection should be tailored to each participant's unique spindle characteristics. Such a paradigm shift towards adaptive closed-loop stimulation and personalized stimulation approaches has been identified as a critical step for facilitating fundamental research and optimizing future clinical applications [58]. A personalized approach to closed-loop brain stimulation of neural oscillations could address a key gap in current knowledge: how to ensure high and consistent stimulation quality across participants with varying neural characteristics.

This study aims to explore different approaches for online adaptation to improve subject-specific performance of spindle detection. We study approaches which facilitate continuous improvement in spindle detection performance resulting in enhanced spindle detection performance for overnight and multi-night experiments and eliminating the need for manual data annotation or manual model retraining on new data. Overall, this work introduces the following contributions[1]:

- a dual-task model, capable of simultaneously classifying sleep spindles and limited sleep-staging (N2-N3), designed for real-time use for CLS;
- an online threshold adaptation algorithm which chooses the best subject- and night-specific spindle detection threshold to perform within-night adaptation;
- a fine-tuning method which incrementally improves the model's performance as more data becomes available night after night.

[1] Code is open source on `https://github.com/Portiloop`

4.3 General Background

4.3.1 Existing Spindle Datasets and Detection Algorithms

Detecting and stimulating spindles in real time poses significant challenges as spindles exhibit shorter duration and faster frequencies compared to slow oscillations, spindle characteristics vary significantly between subjects, particularly among older populations [12], and experts often disagree on the definition and visual appearance of spindles [62, 67].

Many existing sleep spindle detection algorithms operate in "offline" conditions [64, 67, 69]. In this configuration, the algorithm processes the complete EEG recording of a subject to identify and classify brain oscillations as spindles. In contrast, closed-loop stimulation systems require a different approach: they analyze EEG data in real-time, initiating stimuli immediately upon detecting the onset of a spindle, before the spindle has fully formed. This necessity for immediate response renders traditional offline algorithms unsuitable for use in closed-loop stimulation. For this reason, the Portiloop [5] implements a neural network, trained on an annotated dataset, to identify the onset of spindles.

While large EEG datasets exist, finding numerous full nights with complete spindle annotations is challenging. One such dataset, the Montreal Archive of Sleep Studies (MASS, [63]), contains high-quality data from 200 nights across 200 different subjects. Manual spindle annotations in MASS are provided by two experts, with a spindle being labeled if detected by either expert. However, their agreement F1-score (Appendix 4.8.1[2]) of only 0.54 indicates that these annotations are not particularly reliable for either model training or performance evaluation.

To address this problem, the Massive Online Data Annotation (MODA) project recruited five sleep experts to independently annotate spindles and provide confidence metrics [62]. These confidence scores were then aggregated to create a more reliable final sleep spindle dataset, resulting in an inter-scorer agreement F1-score of 0.67. Consequently, MODA has been instrumental in the development of various offline algorithms and the initial iteration of the Portiloop model [5, 67]. One such algorithm, Lacourse's A7, combines four parameters (absolute sigma power, relative sigma power, correlation, and covariance of the sigma band-passed signal to the original signal) along with knowledge about spindle length and frequency to detect spindle presence [67]. Lacourse's A7, or 'LA7', outperforms all other offline algorithms on MODA, achieving an F1-score of 0.70. However, a significant drawback of MODA is its limited scope—it annotates only two to four 115-second sequences per subject within

[2] All appendices and supplementary material can be found at `https://mistlab.ca/papers/2024/ContinualLearningSpindes`

the MASS dataset. To explore the effectiveness of adaptation methods in full-night studies, our ground truth needs to include spindle annotations throughout the entire night and across all sleep stages, reflecting the real-world conditions in which the adaptation methods would be applied. This means that despite MODA being the premier expert-annotated dataset for sleep spindles currently available, its restricted coverage makes it inadequate for our purposes. To address this limitation and obtain full-night annotations, we opted to annotate the entirety of MASS using LA7: in theory LA7 outperforms all other offline detectors as well as individual human annotators, giving annotations that are more reliable than those derived from a single expert.

There are some attempts in literature to use deep learning for online spindle detection [70–73]. These approaches assume large desktop computers with powerful Graphics Processing Units (GPUs), making them hardly portable and unsuitable for the smaller and more limited Portiloop hardware (see Appendix 4.8.2 for details). The Personalized Semi-Automatic Sleep Spindle Detection (PSASD) framework [74] tailors spindle detection to the specific characteristics of each subject. However, PSASD is an offline detector and requires manual annotation of spindles after each night to fine-tune their model. Despite yielding promising subject-specific results, this approach is unsuitable for closed-loop stimulation, in which detection and adaptation must occur in real time.

It is worth noting that spindles are present only in sleep stages N2 and N3 (N2N3): being able to identify these stages in real-time to enable detection could improve performance and limit the amount of false positive stimulations occurring in other sleep stages. Accurate real-time sleep stage estimation is therefore instrumental for enhancing spindle detection accuracy. Moreover, distinguishing between spindle-rich periods (N2-N3), where detection should be active; and spindle-free sections; where detection is not needed; provides us with a great temporal window for adaptation procedures. While prior research has explored real-time sleep staging using deep learning approaches [75–78], our application only requires classifying N2N3 and non-N2N3. Using separate models for spindle detection and sleep staging would introduce undesirable latency, which is detrimental for CLS applications. Therefore, we leverage a single deep learning model for both tasks, employing separate classifiers within the model to generate the respective outputs from a unified EEG data representation.

4.3.2 Continual Learning

Continual Learning, also known as online learning or lifelong learning, constitutes a sub-field of machine learning dedicated to studying the various methods of incrementally training deep learning models on data streams. Unlike the traditional machine learning framework, which assumes that all training data is known from the outset, continual learning involves updating the model's knowledge over time as more data becomes available, aiming for gradual improvement [85]. Continual learning adapts the model to new data, which would allow to learn a subject-specific representation and would improve the model over time without the need of human intervention for multi-night experiments. Deep learning models are heavily dependent on the distribution of their training data. In a continual learning paradigm where this distribution changes over time, a phenomenon known as "catastrophic forgetting" occurs, wherein the model adapts to the most recent data distribution it has encountered [86]. To address catastrophic forgetting, several approaches have been proposed in continual learning [87], which can be categorized as: (a) replay-based; (b) regularization-based; (c) optimization-based; (d) representation-based; (e) and architecture-based. However, most of these approaches and their associated solutions suffer from two primary drawbacks:

1. they rely on dividing the problem into various non-continuous tasks, which is not applicable in our case as the learning from one night or subject needs to carry over to other subjects in a continual manner (Architecture-Based, Optimization-Based).
2. they are computationally intensive or require significant memory resources (Replay-Based, Representation-Based, Optimization-Based). This is impractical for lightweight devices like the Portiloop.

For instance, replay-based approaches involve storing a large amount of data on the device to retrain on a general dataset and mitigate catastrophic forgetting [87]. However, this solution necessitates large amounts of memory and storage, which the Portiloop lacks, and also leads to longer training times as it involves training on both new data and a sample of replay data. Among the approaches, regularization-based methods show the most promise for our application. These techniques aim to minimize catastrophic forgetting by balancing model updates as the data distribution changes over time [87]. Regularization, in general, helps prevent overfitting and enhances the model's ability to generalize to unseen data. Given our hardware constraints, we focus on a simple form of regularization: averaging. Studies have demonstrated that weight averaging generally improves accuracy and generalization performance with minimal computational cost [88]. Averaging the weights of multiple models requires little time and memory, making it particularly suitable for our application.

4.4 Methods

4.4.1 Datasets

To both train and validate our models and algorithms, we use two datasets: the Montreal Archive of Sleep Studies (MASS) data, and data from a study employing the Portiloop device [5]. The MASS dataset has a large number of high quality recordings and includes sleep stage annotations, which are needed to validating both the accuracy of our model's sleep staging and the performance of SLA7 across scenarios with varying sleep staging data availability. We implemented a stratified 5-fold cross-validation approach, which involves partitioning the dataset into five subsets. We reserve 28 subjects from these subsets for testing our adaptation algorithms. The remaining 112 subjects are divided into a training set of 100 subjects, used to train the model, and a validation set of 12 subjects, employed for early stopping and model selection based on optimized performance.

The Portiloop data comprised 41 nights from 8 participants (4-6 nights each). It is used to extend the findings to examine performance in real-world in-home conditions, as well as the behaviour of adaptation strategies over multiple nights. These data lack sleep staging information (as standard sleep staging requires a full polysomnographic setup), and the data quality is more variable, leading to inconsistencies in group truth (i.e. the LA7 algorithm used with MASS did not detect sleep spindles in some nights of Portiloop data).

The secondary use of MASS archival data for use in this study was approved by Concordia University's Human Research Ethics Committee (#30014351, Initial approval date 11 Jan 2021), and the experimental protocol for the Portinight data collection was also approved by Concordia University's Human Research Ethics Committee (#30012713, Initial approval date 22 Dec 2020). Both approvals were recognized by the Human Research Ethics Committee of Polytechnique Montreal.

4.4.2 Dual-task Model

To achieve real-time adaptation during nighttime sleep, we exploit temporal intervals when sleep spindles do not occur, such as during N1 or wakefulness. To accomplish sleep stage detection with minimal additional resources, we propose training a single model for both sleep spindle detection and sleep stage classification. This model builds upon the structure of the original Portiloop model [5] but incorporates additional layers and increased size to enhance performance and representation capacity.

The model architecture combines a sequence of Convolutional Neural Networks (CNNs) with a sequence of Recurrent Neural Networks (RNNs). These layers generate embeddings [83] that are then used by two classifiers with the same structure (a detailed explanation of the model structure can be found in Appendix 4.8.5). Each classifier consists of two linear layers with a non-linear ReLU [79] activation function in between. Our experiments suggest that incorporating this non-linearity improves performance during model fine-tuning.

Both classifiers are binary, with a single output layer compared to a predefined threshold to determine the predicted class. Consequently, we employ the same binary cross-entropy loss function ($\mathcal{L}_{\text{head}_k}$) for each classifier ($k = 1, 2$):

$$\mathcal{L}_{\text{head}_k}(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right] \tag{4.1}$$

where:

- y represents the true labels,
- $\hat{y}$ represents the predicted probabilities,
- N is the number of samples,
- y_i and $\hat{y}_i$ are the true label and predicted probability for the i-th sample, respectively.

To train a unified model with shared embeddings for simultaneous execution of both tasks, we average the two losses:

$$\mathcal{L} = \alpha \cdot \mathcal{L}_{\text{head}_1} + (1 - \alpha) \cdot \mathcal{L}_{\text{head}_2} \tag{4.2}$$

with α a hyperparameter of the system, set to 0.5 in our experiments. This approach encourages the creation of generalized embeddings that encapsulate sufficient information for both sleep spindle detection and sleep stage classification. A breakdown of the training parameters is provided in Table 4.1.

Table 4.1 Training Hyperparameters

Parameter name	**value**
Learning rate	0.0001
Batch size	64
Sequence length	50
Optimizer	AdamW [82]
weight decay	0.001
betas	(0.9, 0.999)

To fully utilize all 140 subjects available in MASS, we train 5 models, one for each fold. Each of these models is trained on 100 subjects, with 12 subjects allocated for validation and 28 reserved for adaptation experiments (see Section 4.5). Achieving a high-performing model for sleep spindle classification necessitates balanced data. Sleep spindles are relatively rare events during the night, making random data sampling ineffective for learning a robust spindle representation. Therefore, it is essential to evenly sample the data between spindles and non-spindles during training. However, we perform the validation steps without balancing considerations: we examine the EEG sequence as it would appear in a real deployment, splitting it in batches for efficiency.

We use the validation split to select each model at its peak performance for each fold. To assess this performance, we consider both sleep staging and sleep-staging performance, integrating the sleep spindle detection F1-score and the sleep staging accuracy (see Appendix 4.8.1 for more details about the metrics used). The final results of each model may vary for each fold.

4.4.3 Adaptation

We investigate two complementary approaches to model adaptation within the context of Portiloop experiments. The first approach involves dynamically adjusting the classification threshold employed by the model to identify sleep spindles. This optimization aims to identify the threshold that yields the best performance for the current recording session, enhancing overnight detection accuracy. The second approach entails fine-tuning the model itself using data collected throughout the night. This data is annotated with sleep spindles identified by an offline algorithm.

The approach we take for both methods of adaptation follows the same principle (shown in Figure 4.1): we store the data seen so far during the night in a buffer in order to compute an online ground truth on which to base our adaptation. Periodically and when we are in a non-spindle sleep stage, we run the adaptation. This strategy leverages the sleep staging detection feature of the implemented dual model to identify suitable training windows (see Algorithm 1). This starts by running an offline algorithm, Lacourse's A7, on the buffer [67]. Because of the segmented nature of the data on which the algorithm is run, we hereafter differentiate it as 'SLA7'. This 'online ground truth' will then be used by both our adaptation methods to learn some night specific features of the signal.

Adaptative Thresholding

Achieving optimal detection performance for sleep spindles hinges on identifying the most suitable threshold for the model. This optimal threshold inherently varies across individuals

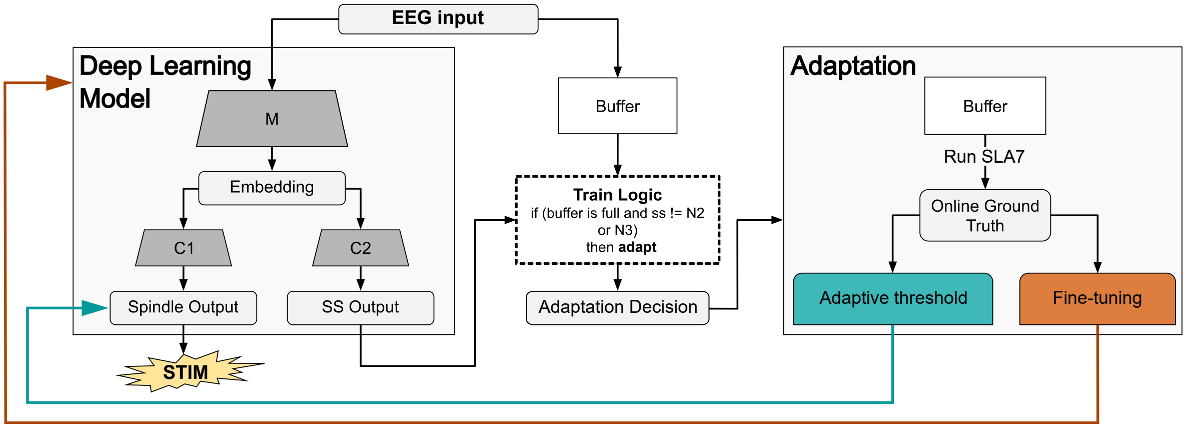


Figure 4.1 **Overview of our spindle detection model and adaptation mechanism.** EEG input is streamed both to the pre-trained deep learning model, which detects spindles, begins to send stimulation, and classifies sleep stages. During non-N2 and N3 sleep stages, data in the buffer is used to calculate a online ground truth for the individual, which are used to adapt the model during the recording. *SS Output* denotes the sleep staging prediction of our model. *M* is the EEG embedding section of our model, common for both classifiers: *C1* for spindle detection and *C2* for sleep stage classification.

```
Input: New point $p$
Output: Buffer with point and model output
$outSpindle, outSleepStage$ = runModel($p$);
Add $p$, $outSpindle$, $outSleepStage$ to buffer;
if $buffer.lengthInMinutes > 60$ then
    $doneAdapt = False$;
    while $outSleepStage$ not N2-N3 or not $doneAdapt$ do
        $doneAdapt$ = runAdapt();
        _, $outSleepStage$ = runModel($p$);
    end
end
```

Algorithm 1: Adaptation Decision Procedure. *runAdapt()* runs the adaptation procedures (adaptive threshold selection) and interrupts after a set amount of time, returning True if the procedures are done and false if they are not. The next time this function is called, it resumes from where it left off.

(see Appendix 4.8.4 for sample distributions). Previously, Portiloop experiments employed a time-consuming and labor-intensive approach. This method involved analyzing a diverse collection of nights from multiple subjects, running the data through the model with various thresholds, and ultimately selecting the threshold that yielded the best average performance across the entire subject pool. This approach does not guarantee the selection of the most appropriate threshold for each individual participant.

To overcome the limitations of the traditional threshold selection method, we propose an adaptive thresholding algorithm. This algorithm efficiently approximates the optimal threshold specific to each subject. It achieves this by maintaining a buffer of the raw network output, allowing for a dynamic exploration of potential thresholds. The algorithm employs a hierarchical search within this buffer to identify the threshold that would have resulted in the best performance based on ground truth data. This ground truth data is generated by applying the SLA7 offline algorithm to the data buffer.

The efficiency of our search strategy is crucial, aiming to achieve superior performance within a limited time frame. For certain subjects, the optimization metric, F1-score (see Appendix 4.8.1), exhibits a non-monotonic behavior, with multiple peaks. This characteristic renders approaches like binary search in the F1-score space ineffective. To address this challenge, we propose a hierarchical search algorithm. This algorithm iteratively partitions the search space, progressively refining its precision in regions exhibiting promising performance. This strategic allocation of search time ensures focused exploration, concentrating efforts on areas with the highest likelihood of containing the optimal threshold value. The search process culminates in an approximation of the subject-specific optimal threshold, a critical parameter for our model's spindle detection capabilities.

This approach achieves a well-balanced trade-off between accuracy and computational efficiency. It allows for a rapid and reliable approximation of the optimal threshold for each subject without modifying the model weights. The identified threshold remains in effect until new data becomes available. At that point, the algorithm triggers a reevaluation and refinement of the threshold to ensure continuous improvement in detection accuracy. Through the combined application of the adaptive thresholding algorithm and its hierarchical search mechanism, we establish a robust methodology that not only optimizes performance efficiently but also dynamically adapts to the unique brain activity patterns of each subject during a recording. This process guarantees the ongoing refinement of the spindle detection threshold on an individual basis.

Fine-tuning

This study explores a second adaptation approach that involves fine-tuning the model weights. This process aims to improve both the model's representation of EEG data through its embeddings and the overall performance of the sleep spindle classifier. We leverage online ground truth data for spindle detection, generated by the SLA7 algorithm, to guide this fine-tuning process. The primary objective is to endow the model with subject-specific or experiment-specific knowledge while preserving its generalized capabilities, thereby achieving long-term performance gains. This necessitates extracting maximum information from the limited ground truth data available while mitigating the risk of catastrophic forgetting, a phenomenon where the model loses previously learned information during adaptation.

We first investigate the effects of fine-tuning the model during the night to capture subject-specific information. Then, we examine the transferability of learned information from one night to the next by training the model at the end of each recording session. These experiments aim to assess the potential benefits of continual model improvement and the potential drawbacks of catastrophic forgetting when retraining on recent nighttime data. The fine-tuning process is evaluated under three distinct configurations:

1. Fine-tuning the entire model on SLA7-annotated data, employing the same configuration used for initial training.
2. Maintaining a copy of the original model weights throughout each experiment and averaging the weights after training.
3. Freezing the embedding layers (convolutional neural network (CNN) and recurrent neural network (RNN) layers) of the model and exclusively fine-tuning the weights associated with the sleep spindle classifier.

For each configuration, we adopt the same data balancing technique used during initial training, ensuring a 50% spindle composition within the training data. To do so, we randomly sample non-spindle data from our buffer. Furthermore, we maintain training hyperparameters consistent with those employed for the original model development. Details regarding these hyperparameters are provided in Table 4.1.

4.5 Results

4.5.1 Establishing performance of the dual-task model

Our analysis first focuses on evaluating the performance of the dual-task model, using the MASS dataset. Due to the stratified 5-fold cross-validation approach, each fold utilizes distinct training and validation subject sets. Consequently, the validation results exhibit some variation across folds (final validation results for each fold along with an explanation of the model selection methods are provided in Appendix 4.8.3).

Table 4.2 presents the results achieved by our initial model on the testing set, encompassing all subjects across all folds.

Table 4.2 Descriptive Statistics of Dual Model on Testing Set

	Staging Accuracy		Spindle F1-score	
	Older	Younger	Older	Younger
Median	0.871	0.925	0.538	0.548
Mean	0.843	0.918	0.520	0.538
Std. Deviation	0.095	0.40	0.124	0.077
Minimum	0.402	0.759	0.118	0.350
Maximum	0.940	0.972	0.711	0.670

To better understand the model's behaviour, we explored the effect of age group on online sleep staging accuracy and spindle detection performance (F1-score; noting that the baseline model is trained on a dataset that includes both younger and older adults). We conducted a Mann-Whitney U statistical test (due to a violated assumption of normality). This result suggests a significant relationship between age category and sleep-staging accuracy ($U = 832$, $p < 0.001$, rank-biserial correlation (effect size) $= -0.6$), indicating that our model more accurately identifies sleep stages in younger adults. We did not, however, observe a significant difference between younger and older adults on spindle detection performance as measured by F1-score ($U = 2074$, $p = 0.86$).

4.5.2 Validating the Online Baseline: SLA7

Next, we evaluated the utility of incorporating sleep stage information into the spindle detection as a means of deriving online ground truth. As both of our proposed adaptation approaches utilize SLA7 as the online baseline, we analyze the algorithm's behavior under three configurations:

1. SLA7 with perfect sleep staging (Ground Truth)
2. SLA7 with no sleep staging (None)
3. SLA7 with our online sleep staging (Online)

To assess the performance variations across these configurations, we employ a repeated-measures ANOVA test. The within-subjects factor is comprised of the three sleep staging

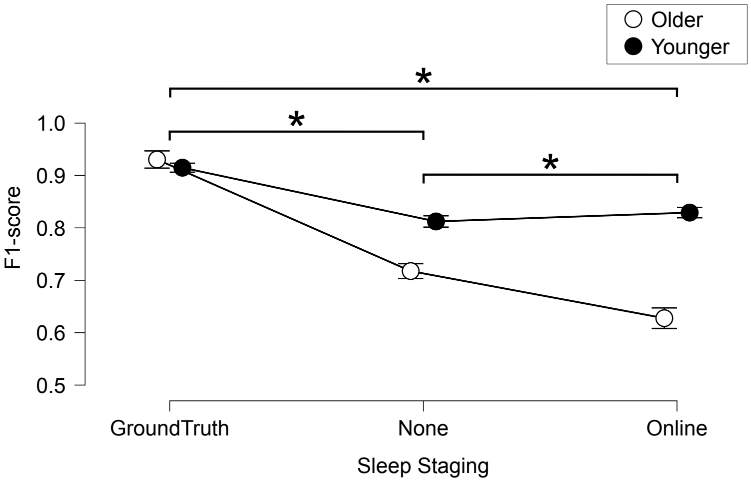


Figure 4.2 **Impact of sleep staging on SLA7 F1-score compared to LA7 in younger and older adults.** Overall, the online sleep staging does not improve F1-scores. Error bars show the standard error of the mean. '*' indicates post-hoc comparisons in which sleep-staging configuration leads to significant differences in F1-score.

configurations (Ground Truth, None, and Online). Age is included as a between-subjects factor.

The results of the ANOVA (see Figure 4.2) reveal a statistically significant main effect for the sleep staging configurations ($F = 111.437$, $p < 0.001$), indicating that the presence and quality of sleep stage information significantly impacts the F1-score of SLA7 compared to LA7. Furthermore, a significant main effect of age was observed ($F = 32.110$, $p < 0.001$; see Appendix 4.8.6 for more information).

Post-hoc tests revealed that each sleep staging configuration differed from the others $p < 0.001$. When using Ground Truth sleep staging, age does not exert a significant influence on F1-score (mean difference = 0.016, $p = 0.779$). However, for both configurations lacking sleep stage information (None and Online), younger subjects exhibit significantly higher F1-scores compared to older subjects (None: MeanDiff= -0.095, $p < 0.001$; Online: MeanDiff= -0.201, $p < 0.001$). Interestingly, for older subjects, the configuration with no sleep staging results in a higher F1-score compared to the configuration with online sleep staging (MeanDiff= 0.090, $p < 0.001$). Conversely, younger subjects do not show a statistically significant improvement in F1-score when using online sleep staging compared to no sleep staging (MeanDiff= -0.017, $p = 0.7794$).

These findings suggest that while our online sleep staging algorithm achieves good absolute performance (close to 90% accuracy in classifying N2-N3 vs. other sleep stages, see Table

4.2), it is not sufficiently accurate to fully replicate the performance of the Ground Truth scenario when used within SLA7. Consequently, incorporating our online sleep staging into the online ground truth computation for adaptation would not improve results for younger subjects and might even deteriorate them for older subjects.

Therefore, we propose to utilize the sleep staging component of our model solely to determine suitable training periods during nighttime sleep. The online spindle ground truth will be computed by SLA7 without incorporating any sleep stage information.

4.5.3 Within-night Adaptation

Despite demonstrating a significant performance difference between SLA7 and LA7 under imperfect sleep staging conditions, SLA7 remains the sole practical option for online ground truth during real-time night-time adaptation. This section explores three distinct approaches for achieving performance improvement over the course of a night, without the need for human intervention:

1. **Adaptive Threshold (Threshold)**: This approach uses our proposed algorithm for dynamically adjusting the threshold based on incoming data;
2. **Fine-tuning:** This approach involves progressively refining the model's weights as new data becomes available throughout the night via SLA7;
3. **Combined:** This approach combines both Threshold and Fine-Tuning, aiming to leverage the strengths of each method for enhanced performance.

To evaluate the effectiveness of the proposed adaptation methods, we compare the F1-scores achieved by SLA7 with each adaptation approach (Threshold, Fine-tuning, Combined) against the baseline LA7 performance on the MASS dataset.

We conduct a repeated-measures ANOVA test with a single within-subjects factor: the adaptation configuration (Baseline, Threshold, Fine-tuning, Combined). Age category (Older vs. Younger) is included as a between-subjects factor to investigate potential age-related effects. All detailed results of the ANOVA tests can be found in the Appendix 4.8.7.

The results (see Figure 4.3) reveal a statistically significant main effect for the adaptation configuration ($F = 37.581$, $p < 0.001$), indicating that the choice of adaptation method impacts F1-score. However, the between-subjects effect of age did not reach significance ($p = 0.058$), warranting further exploration through post-hoc tests (see Table 4.3)).

Examining the interaction effects between configurations reveals that both the adaptive threshold (Threshold) and the combined approach (Combined) achieve statistically significant improvements in F1-score compared to the baseline configuration. However, the fine-

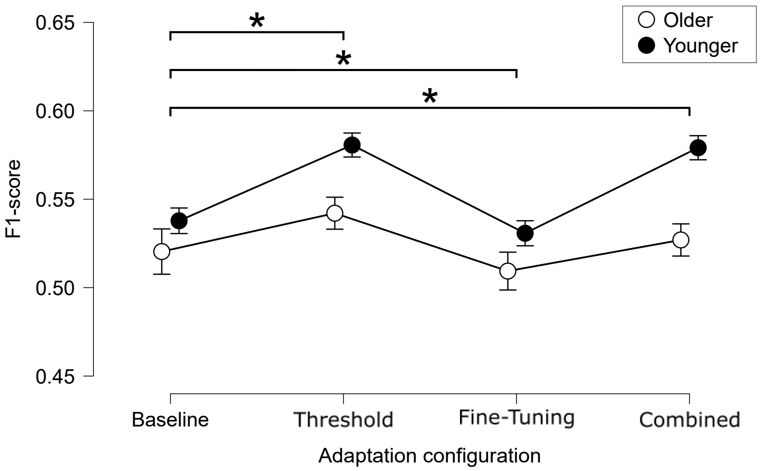


Figure 4.3 **Impact of adaptation configuration on F1-score.** Error bars show the standard error of the mean. '*' indicates post-hoc comparisons in which adaptation strategy leads to improvements in F1-score over baseline (see also Table 4.3).

Table 4.3 Post-hoc Comparisons - Config

		Mean Diff.	SE	t	p_{bonf}
Baseline	Threshold	−0.032	0.004	−7.177	< .001
	Fine-tuning	0.009	0.004	2.010	0.271
	Combined	−0.024	0.004	−5.323	< .001
Threshold	Fine-tuning	0.041	0.004	9.187	< .001
	Combined	0.008	0.004	1.854	0.387
Fine-tuning	Combined	−0.033	0.004	−7.333	< .001

tuning approach does not demonstrate a significant improvement in F1-score over a single night. Interestingly, Combined does not yield a statistically significant benefit over using Threshold alone. This suggests that the performance gains observed in Combined are primarily attributable to the adaptive threshold mechanism. Consequently, fine-tuning the model throughout the night appears to be an ineffective strategy for subject-specific adaptation within a single night.

Furthermore, the interaction between configuration and age category presents a noteworthy finding. The adaptive threshold approach (Threshold) leads to significant improvements over baseline for both younger subjects (mean difference = -0.043, standard error = 0.006, $p < 0.001$) and older subjects (mean difference = -0.022, standard error = 0.006, $p = 0.018$). However, the magnitude of improvement is greater for younger subjects, as evidenced by the larger mean difference for the same standard error.

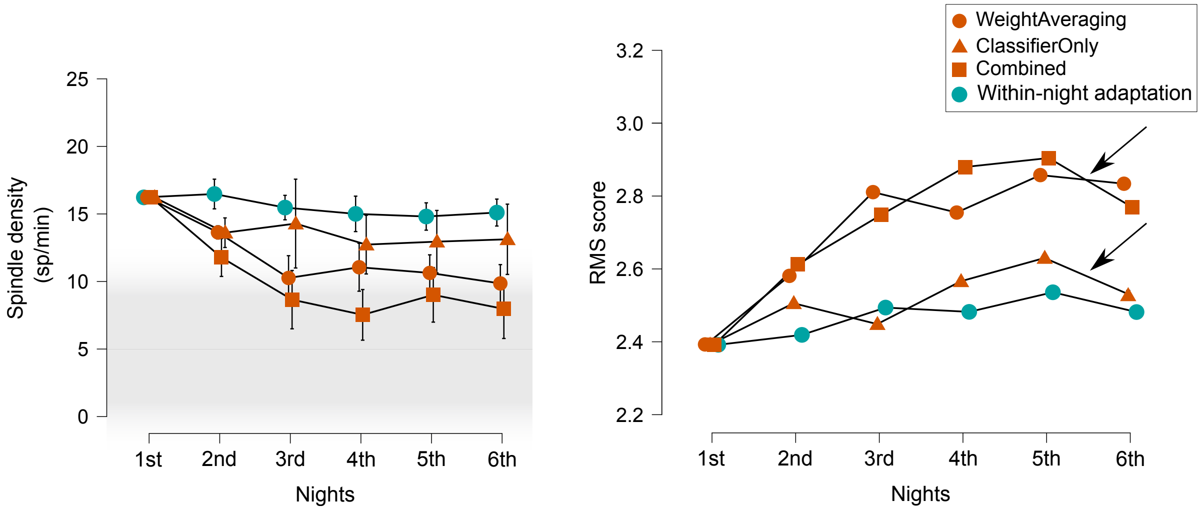


Figure 4.4 **Spindle density and RMS score comparing within-night adaptation (Threshold) and different end-of-night training strategies (Combined, WeightAveraging, ClassifierOnly).** The left panel displays the mean spindle density (sp/min) across nights. The error bars represent the standard error of the mean (SEM). The shaded gray area indicates the expected range of spindle density in N2 and N3 sleep as reported in literature, for reference (2-10 sp/min) [4]. The right panel displays the mean RMS score across nights for the same fine-tuning strategies and Threshold method, as an index of spindle strength. Note that SEM error bars are not visible, due to their small size. The black arrows indicate catastrophic forgetting in the Combined and ClassifierOnly models, which is avoided in WeightAveraging.

In conclusion, these results demonstrate that dynamically adapting the threshold based on spindles detected during a single night significantly improves sleep spindle detection metrics. Conversely, fine-tuning the model throughout the night does not yield significant advantages. These findings suggest that the adaptive threshold approach offers a promising strategy for real-time adaptation within a closed-loop brain stimulation system over a single night.

4.5.4 End-of-night Training

Our previous results indicate that fine-tuning the model on a single night yields no significant improvement. Here, we explore fine-tuning over multiple nights, training only at the end of each night, while maintaining the constraints of no human interaction for data annotation and no replay buffer.

Real-world, multi-night data collected by participants using the Portiloop in their homes is used in this experiment. Due to the absence of sleep stage annotations and lower data quality compared to lab settings, F1-score comparisons with LA7 spindles are deemed unsuitable. To assess spindle detection quality, we revert to the spindle definition of burst sigma band activity and use the following metrics (please see Appendix 4.8.1 for more details):

- **Spindle Density:** this metric measures the number of detected spindles per minute.
- **RMS Score:** this metric quantifies the quality of the spindles detected, reflecting the strength of the sigma band activity and potentially indicating spindle presence.

Prior research has documented substantial inter-subject variability in sleep spindle density, with reported values ranging from approximately 2 spindles per minute (sp/min) to over 10 sp/min [4]. We include a visual indicator of this range as a reference in Figures 4.4 and 4.5, though noting that this range does not represent a strict threshold for acceptable values. Instead, it serves as an indicator of whether the detected spindle densities are approaching a range generally considered to encompass typical physiological values.

Our first analysis explores the difference between training on a random array of subjects as opposed to data from the same subject. To do so, we split our experiments into two categories:

- Random Experiments: a random sequence of 6 nights was selected from random subjects;
- Same-Subject Experiments: subjects with at least 5 nights of data were used, resulting in 6 total subjects.

We compared spindle densities across experiment types and adaptation configurations with a repeated-measures ANOVA test (see Appendix 4.8.8). These configurations include:

- Baseline (no adaptation)
- Adaptive Threshold (method described previously)
- Fine-tuning (fine-tuning the model at night's end)
- WeightAveraging (combining adaptive thresholding and fine-tuning with averaging)

The results reveal a significant difference among configurations ($F = 10.634$, $p < 0.0001$, $\eta = 0.042$), but no significant difference between experiment types ($F = 0.054$, $p = 0.917$, $\eta = 2.324\text{e-}4$). Post-hoc tests confirm no significant differences between experiment types for any configuration (baseline, Threshold, WeightAveraging, Fine-tuning; all $p = 1.000$). This suggests that training on data from the same subject versus another subject yields statistically similar results.

Next, we investigate the performance of various adaptation methods over our two metrics, spindle density and RMS score. To do so, we employ two different ANOVA tests:

- **Repeated measures ANOVA over time**: This analysis compares the spindle density for each training configuration as a factor and night number as a between-subject factor, investigating the evolution of different adaptation methods over time (details in Appendix 4.8.9).

- **ANOVA for RMS score**: This analysis uses the RMS score of all detected spindles as the dependent variable, with configuration and night number as fixed factors (details in Appendix 4.8.10).

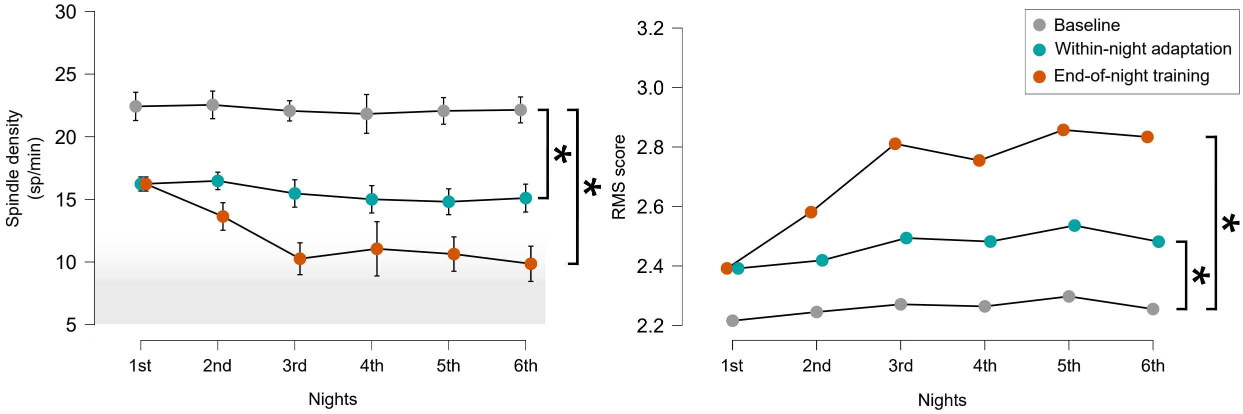


Figure 4.5 **Spindle density and RMS score across nights comparing adaptation strategies with the baseline model.** The left panel displays the mean spindle density (sp/min) across nights for our adaptation strategies. End-of-night training includes Threshold combined with fine-tuning and WeightAveraging, and Within-night adaption includes Threshold. The error bars represent the standard error of the mean (SEM). The shaded gray area highlights expected range of spindle density in N2 and N3 sleep as reported in literature, for reference (2-10 sp/min) [4]. The right panel displays the mean RMS score across nights. Note that SEM error bars are not visible, due to their small size. '*' indicates post-hoc comparisons in which the adaptation method leads to significant improvement compared to baseline.

We look at the Fine-Tuning method of end-of-night training, which does not involve looking threshold adaptation, compared with our within-night adaptation method: Threshold. Across 6 nights, we did not observe any significant difference in spindle density between fine-tuning and Adaptive Threshold ($p = 0.119$). Similar results are found with RMS scores. Although RMS scores generally improve compared to the baseline, they perform worse than the Adaptive Threshold algorithm (MeanDiff= -0.133, $p < 0.001$). This suggests that fine-tuning alone is insufficient to surpass the Adaptive Threshold algorithm over 6 nights. Consequently, the remainder of the analysis focuses on comparing configurations that combine the Threshold method with Fine-tuning, to explore how fine-tuning influences performance over time.

Next, we compare different fine-tuning strategies to examine the impact of catastrophic forgetting and determine the optimal approach:

- **Combined**: Training the model on the entire night's data.
- **WeightAveraging**: Averaging the model weights between the beginning and end of the night.
- **ClassifierOnly**: Freezing the weights of the embeddings to assess whether learning originates from the embeddings or classifiers.

Results for both spindle density and RMS score are presented in Figure 4.4. Our key findings are:

- ClassifierOnly yields the worst overall performance and does not outperform Threshold in terms of spindle density (MeanDiff=1.694, $p = 0.207$). This suggests that training the embeddings plays a crucial role in maximizing performance;
- WeightAveraging and Combined methods outperform Threshold in terms of spindle density (WeightAveraging: MeanDiff= 3.571, $p = 0.003$; Combined: MeanDiff= 5.320, $p < 0.001$). However, they are not significantly different ($p = 0.207$).

The analysis of RMS scores reveals: (a) faster convergence with no averaging: Combined converges faster towards a higher RMS score initially (night 4: MeanDiff= 0.227, $p < 0.001$); (b) catastrophic forgetting with no averaging: Combined displays signs of catastrophic forgetting after night 5, with performance dropping and falling below WeightAveraging (night 5: MeanDiff= 0.065, $p = 0.007$). These findings demonstrate that our averaging method (WeightAveraging) effectively mitigates catastrophic forgetting, even when training only on the previous night's data once.

Finally, we combine Threshold with fine-tuning and WeightAveraging into End-of-night training, and compare it with Threshold as our Within-night adaptation method, as well as with our baseline (Figure 4.5). End-of-night training outperforms the baseline in spindle density (MeanDiff= 10.231, $p < 0.001$) and RMS Score (MeanDiff= 0.447, $p < 0.001$). Furthermore, end-of-night training surpasses Within-night adaptation in spindle density (MeanDiff= 3.571, $p = 0.003$) and RMS Score (MeanDiff= 0.238, $p < 0.001$). These results demonstrate end-of-night training's ability to improve over time by fine-tuning on the limited new information provided, despite SLA7's limitations.

4.6 Discussion and Future Work

In this work, we explore different methods for subject-specific adaptation of sleep spindle detection over a single night, and continual improvement of the underlying model over multiple nights. These methods, designed in the context of closed-loop brain stimulation and intended to run in real time on lightweight hardware, show promise to improve the feasibility of providing consistently high quality stimulation across study participants who have varying neural characteristics, without adding resource-intensive retraining or manual inspection steps.

Although fine-tuning the model does not appear to yield advantages over a single night for this application, we show that it does yield better results when run on multiple consecutive nights. We compared fine-tuning alone to the our adaptation threshold method and found no significant difference in spindle density across six nights. While RMS scores generally improved with fine-tuning, they remained lower than those achieved with the adaptation threshold algorithm. This suggests that fine-tuning alone is insufficient to surpass that approach.

We then investigated the impact of catastrophic forgetting on different fine-tuning strategies. Freezing the embedding layer weights during fine-tuning resulted in the worst performance, highlighting the importance of training the embeddings for optimal performance. Both WeightAveraging and Combined, which involved training on the entire night's data, outperformed Threshold in terms of spindle density. However, no significant difference was observed between these two methods.

To determine the superior approach, we examined RMS scores across nights. The Combined method exhibited faster initial convergence towards a higher RMS score, indicating its ability to learn quickly from new data. However, it also displayed signs of catastrophic forgetting after the fifth night, suggesting a loss of previously learned information. In contrast, WeightAveraging, which involved averaging weights between the beginning and end of the night, effectively mitigated catastrophic forgetting. This approach maintained performance even when training only on the previous night's data once.

Overall, the combination of Threshold with fine-tuning and WeightAveraging achieves the best performance. It significantly improves upon the baseline and the adaptive threshold method in both spindle density and RMS score. These observations demonstrate the model's ability to learn and adapt over time with limited new information.

These findings demonstrate the potential of continual learning strategies for enhancing real-world sleep spindle detection in a multi-night setting and offer practical guidance to researchers seeking to optimize spindle detection in their work. Specifically, using an online sleep staging model as input to compute the online ground truth (using SLA7) does not provide a spindle detection benefit as regards F1 score improvement (although the online sleep staging algorithm works reasonably well and can be practically useful to differentiate NREM from REM sleep for the purposes of sleep stage-specific stimulation). As for the methods of adaptation, their use is indicated based on the amount of individual data that will be present in the study design. If only a single nights' data will be available, as might be the case in the majority of fundamental research studies on learning and memory, using the threshold algorithm is the best choice to improve F1 scores (in both younger and older adults). If 2-4 nights of data will be available for each participant, further benefits (as measured by the sleep spindle quality indicators spindle density and RMS score) may be derived from combining threshold adaptation with fine-tuning strategies. Finally, if 5 or more nights are available, as might be possible in longitudinal studies and in potential therapeutic applications, catastrophic forgetting can occur, but is avoidable by using a weighted averaging approach (see Figure 4.5). Beyond the specific use-can of sleep spindles online detection, this work also provides a template for the process of developing fit-to-purpose adaptive, personalized detection of any other neural events measurable with EEG that might be targets of interest to researchers and clinicians.

A fundamental consideration for developing adaptive neural event detection algorithms turns out to be the definition of a 'ground truth', and the degree to which it can be computed effectively online within an available data window (noting that some detection algorithms consider the longer-term statistical properties of the signal in their decision, e.g., [101]). In the case of sleep spindles, there is only modest agreement amongst expert observers and amongst automatic offline detection algorithms as to whether or not a spindle has taken place; the closest we have to a ground truth is a modest consensus amongst expert sleep scorers, and algorithms that approximate their opinions [62]. The issue is not necessarily that expert observers or detection algorithms are making errors (as is implied by the use of F1 scores), nor that the data are too noisy to capture the phenomena of interest (though data quality does matter), but rather the problem is that there is no real ground truth in the same way that there would be if, for example, an ML classifier was being trained to detect whether or not a photograph contained the image of a cat. It is worth considering why defining whether or not a spindle exists is a different kind of problem (noting that similar problems are inherent to detecting most other neural events).

Sleep spindles are generated in thalamocortical brain circuits. The cortex refers to the superficial layer of the brain. The thalamus is a deep brain structure that comprises nuclei which are specialized via their functional connections to specific cortical regions and sub-thalamic nuclei to relay and filter specific sensory and motor signals, regulate states of consciousness, and critically here, to help reorganize new, temporally-stored memories into longer-lasting ones during sleep spindles. Spindles result from circuits comprised of inhibitory and excitatory connections between different areas of the thalamus and the cortex (see [100] for details). Parallel thalamocortical loops, containing cells with regionally-specific and heterogenous electrical properties connect pairs of thalamic and cortical regions [102]. Relevant to our purposes, spindle properties (frequency, amplitude, location), interaction with other neural oscillations like slow oscillations, and likely function, varies across the brain according to connectivity patterns and regional differences in cellular properties in the thalamus itself [102]. Furthermore, spindles can stay localized in the cortex or spread more widely [103], and some spindles travel in the brain [104].

It is these complex neurophysiological processes, unfolding in a three-dimensional structure and propagating through tissues, which we observe through the one-dimensional lens of an electrode pair placed on the scalp surface. The strongest, clearest spindles that experts and spindle detection algorithms aim to identify in EEG likely originate in medial dorsal cortex (see Figure 5 in [105]) and represent a small subset of highly heterogeneous spindle-like neural events in which properties vary in multiple dimensions [106–108]. Returning to the matter of defining a set of ground truth labels, we must keep in mind that the decision as to whether or not a spindle has taken place relies on binarization of a series of continuous variables and multidimensional processes. While the specifics of defining spindles are somewhat arbitrary, we caution against the interpretation that detecting sleep spindles is meaningless and futile; the specific subset of sleep spindles that experts and algorithms are specialized to detect do have clear correlations to behaviour and memory [100].

Considering the neurophysiological context of spindle generation leads to several interesting ideas: first, since we fine-tune the model based on an imperfect baseline (the SLA7 algorithm), our improvements as measured by F1 scores are inherently capped by the performance of this baseline. F1 scores are useful metrics of algorithmic adaptation to the extent that the offline algorithm which determines ‘ground truth’ labels is successful at identifying a subset of neural activity of interest for a given application. In the absence of a conclusive ground truth, other metrics such as spindle density or RMS score can provide researchers with insights into algorithms’ biases and how adaptation is influencing the characteristics of the subset of detected spindles. Algorithms can be tuned for different purposes; for example, scientists wishing to be sure they are stimulating only a certain type of spindle might opt for a restrictive detection, whereas in a clinical context it might be preferable to stimulate more liberally so as overall to maximize stimulation. Second, research into the roles of sleep spindles to date has focused almost exclusively on those spindles which happen to have been most obvious to the naked eye in EEG recordings. AI-based detection tools which can adapt both to individuals’ physiological differences and to different ‘ground truths’ open a world of possibilities for precision detection and manipulation of this highly heterogeneous class of important neural events to discover their roles and unlock their therapeutic potential.

Another consideration concerns the fact that each Portiloop device operates independently; weights will diverge over time. An intriguing avenue for future research would be to devise a method for Portiloop devices to share newly adapted weights after a certain number of nights. Federated learning, a privacy-preserving machine learning technique [109], offers a promising solution to this challenge. By enabling secure communication between devices without sharing raw data, federated learning could facilitate the aggregation of these weight updates, leading to a more globally optimized model while preserving user privacy. The nature and dispersion of adaptation to a large population of subjects is a source of information that could be exploited to devise better models.

4.7 Conclusion

In this study, we explored subject-specific adaptation and continual model improvement for real-time sleep spindle detection on portable devices. While fine-tuning alone showed limited benefit for single nights, our combined approach with weight averaging yielded significant improvement over multiple nights, while avoiding catastrophic forgetting. This approach offers a significant step towards personalized closed-loop brain stimulation, holding promise for advancing our understanding of sleep spindle function and its role in memory consolidation. Future work should explore the impact of personalized spindle stimulation on memory consolidation in controlled settings and investigate the potential benefits for enhancing cognitive function, particularly in aging populations. By enabling personalized detection and stimulation of brain events, this work aligns with the emerging trend in closed-loop stimulation research, which emphasizes the importance of adaptive approaches to maximize therapeutic efficacy and generalizability of findings across diverse populations and neural events.

4.8 Supplementary Material

4.8.1 Defining our metrics

In this Appendix, we provide a brief overview of the spindle metrics used in this paper: F1-score, recall, precision, RMS score and spindle density.

Characterizing the performance of any spindle detector can be done in two different ways:

1. comparing to a ground truth spindle detection for the same data and report metrics of detection performance. This requires a trustworthy ground truth which can be used for comparison to compute metrics like the F1-score;
2. showing evidence that the spindles detected are in fact spindles and that their distribu-

tion approximates the expected values for humans, with metrics such as spindle density and RMS score of detected spindles.

F1-score, recall and precision are commonly used metrics in classification tasks. These metrics are especially useful when the class distributions are imbalanced which leads to other common metrics like the accuracy being biased towards the most common class. We choose these metrics as they do not take into account the True Negatives in their computation as opposed to other metrics like specificity, which would be biased by the rarity of spindles during sleep and would not be a good indicator of performance. However, these metrics require comparison to some ground truth. When such a ground truth is not available, we opt to report RMS score in sigma power and spindle density.

F1-Score

F1-score is a metric that combines both precision and recall into a single value. It is particularly useful in scenarios where the classes are imbalanced. The formula for F1-score is given by:

$$F1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

where precision is the ratio of true positive predictions to the total number of positive predictions, and recall is the ratio of true positive predictions to the total number of actual positive instances.

Recall

Recall, also known as sensitivity or true positive rate, measures the ability of a classifier to correctly identify positive instances out of all actual positive instances. The formula for recall is given by:

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

where True Positive (TP) represents the number of correctly identified positive instances, and False Negative (FN) represents the number of positive instances incorrectly classified as negative.

Precision

Precision measures the proportion of true positive predictions out of all positive predictions made by the classifier. The formula for precision is given by:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

where True Positive (TP) represents the number of correctly identified positive instances, and False Positive (FP) represents the number of negative instances incorrectly classified as positive.

Given that our objective is to detect and stimulate spindles for CLS, we use a by-event evaluation of performance. This means that an event is considered a True Positive if our model detects a spindle the ground truth spindles, as opposed to making sure that every single sample is correctly identified as a spindle or non-spindle.

RMS Score

The RMS (Root Mean Square) score is a metric we defined in the context of this study to assess the quality of candidate spindle detections. It quantifies the sigma activity at a specific time compared to a baseline period. It is calculated by dividing the RMS value of the signal filtered in the sigma band (11-16 Hz) at the time of detection (from 0 to 0,5s post detection) by the RMS value of the same filtered signal 2 second prior to the detection (from -2 to -1.5s pre detection). As spindles, when occurring in trains, are often distant by 4 seconds, measuring spindle activity 2s prior to the current detection ensures a neutral baseline value.

Let $x(t)$ denote the sigma signal at time t within the specific time-window of interest. The sigma power of the segment is then calculated as the Root Mean Square (RMS) of the filtered signal:

$$RMS = \sqrt{\frac{1}{T}\sum_{t=1}^{T} x(t)^2} \tag{4.3}$$

where T is the length of the segment.

Finally, let RMS_{post} represent the RMS value of the sigma signal in the 0.5 seconds following a spindle detection, and RMS_{pre} represent the RMS value in a 0.5 seconds-long time window occuring 2 s prior to the same detection. The RMS score (RMSscore) is calculated as follows:

$$\text{RMSscore} = \frac{RMS_{\text{post}}}{RMS_{\text{pre}}} \tag{4.4}$$

Spindle Density

Sleep spindle density is a critical metric utilized in sleep research due to its ability to capture the frequency of spindle occurrences within a specified period, providing valuable insights into the temporal distribution of spindle activity. Unlike metrics solely based on spindle presence or absence, spindle density offers a more comprehensive understanding of spindle dynamics by accounting for variations in spindle occurrence over time. Mathematically, spindle density (SD) is computed as the number of spindles (N_{spindles}) detected within a defined epoch duration (T_{epoch}), typically expressed per unit of time (e.g., per minute). Therefore, the spindle density (SD) is calculated as follows:

$$SD = \frac{N_{\text{spindles}}}{T_{\text{epoch}}} \tag{4.5}$$

where N_{spindles} represents the total number of spindles detected within the epoch duration T_{epoch}.

In addition to the RMS score, spindle density serves as a valuable metric for evaluating the efficacy of our spindle detection algorithm in terms of both the quality and quantity of detected spindles, removing the necessity for a ground truth reference for comparison.

4.8.2 Portiloop Hardware Specifications

Table 4.4 Technical Specifications (`https://coral.ai/products/dev-board-mini`

Component	**Specification**
CPU	MediaTek 8167s SoC (Quad-core Arm Cortex-A35)
ML Accelerator	Google Edge TPU coprocessor: 4 TOPS (int8); 2 TOPS per watt
RAM	2 GB LPDDR3
Flash Memory	8 GB eMMC

4.8.3 Validation Results

This Appendix describes the process employed to select the optimal model for cross-validation, considering the 24-hour training time constraint. It is crucial to reiterate that the primary focus of this investigation lies in evaluating the effectiveness of adaptation methods, rather than achieving the absolute best possible final model performance. Consequently, the inherent quality of the final model holds less significance compared to the improvements observed through the application of each adaptation method.

Table 4.5 presents the comprehensive results for each fold, encompassing both the chosen evaluation metrics: sleep staging accuracy and sleep spindle detection F1-score. The epoch that maximizes the sum of these two metrics is selected as the optimal model for subsequent analysis.

Table 4.5 Validation Result for each fold

	Spindle F1-score	**Sleep-staging Accuracy**	**Combined (sum)**
Fold 1	0.5982	86.14	1.460
Fold 2	0.5051	93.09	1.436
Fold 3	0.4902	88.59	1.376
Fold 4	0.4263	87.78	1.304
Fold 5	0.4913	86.87	1.360
Average	**0.5022**	**88.49**	**1.387**

4.8.4 Threshold Distribution

Find a figure showing the variance in threshold distribution in figure 4.6

4.8.5 Model Structure

This Appendix provides a comprehensive explanation of the dual-task model architecture illustrated in Figure 4.7. The model is built using the PyTorch deep learning framework [110], and it incorporates various layers to process the input electroencephalogram (EEG) data to achieve the two objectives of sleep stage classification and online sleep spindle detection. Here is the detailed description of each layer used:

- Conv1D Layers (Conv1d): This sequence of convolutional layers with 1-dimensional kernels is responsible for extracting features from the raw EEG data. The number of filters and kernel sizes used in these layers (denoted as *Conv1D(inChannels, outChannels, kernelSize)*) are crucial for capturing relevant temporal and spectral features from the EEG signal.
- MaxPool1D Layers (MaxPool1d): These layers perform downsampling along the temporal dimension of the data, reducing its dimensionality while preserving important

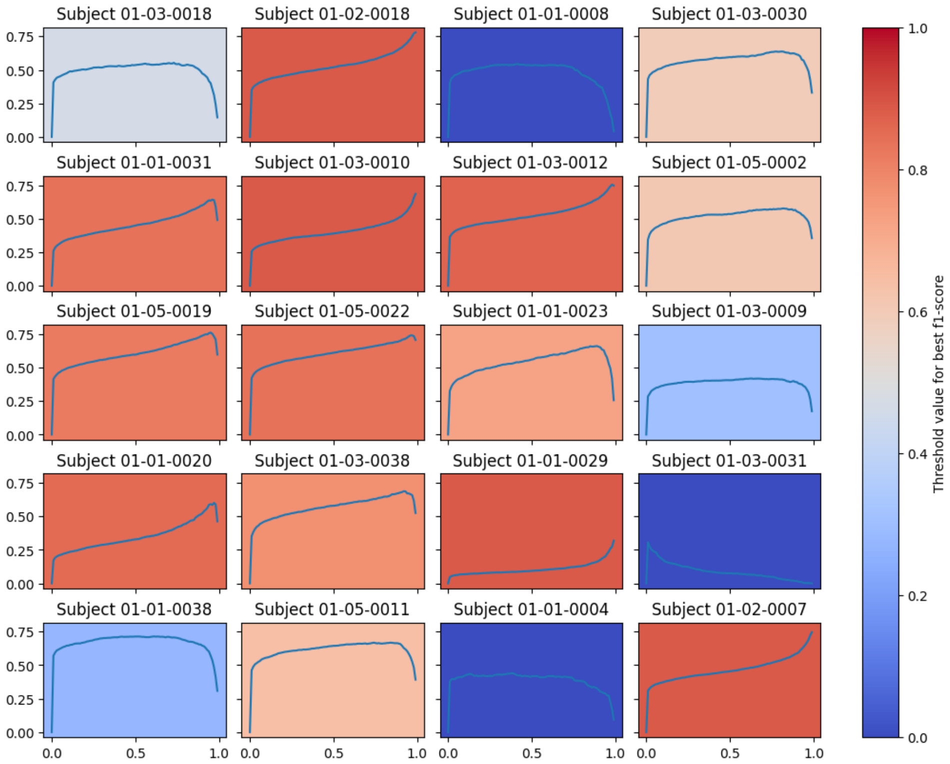


Figure 4.6 Plot of the F1-score depending on threshold for the entire night for 20 random subjects. Although the model is trained to label either 0 or 1, the threshold of 0.5 is rarely the best threshold for any subject as was the case with the previous Portiloop model [5].

features. The kernel size controls the amount of downsampling applied (denoted as *MaxPool1d(kernelSize)*).

- GRU Layer (GRU): This Gated Recurrent Unit (GRU) layer is a type of recurrent neural network (RNN) that effectively captures temporal dependencies within the EEG data. The number of units in the GRU layer (denoted as *GRU(inputsize, hiddenSize)* determines its capacity to learn complex temporal relationships.
- Linear Layers (Linear): A sequence of fully-connected linear layers performs further feature extraction and transformation on the combined representation. The number of units in each linear layer (denoted as *Linear(inFeatures, outFeatures)* for input and output dimensions respectively) determines its complexity and capacity to learn higher-level features.

The classification models generate a single floating-point value as output. To ensure these outputs range between 0 and 1, a sigmoid activation function is applied as the final layer. This transformation allows for a probabilistic interpretation of the model's predictions. A predefined threshold is then employed to convert the continuous output into a binary classi-

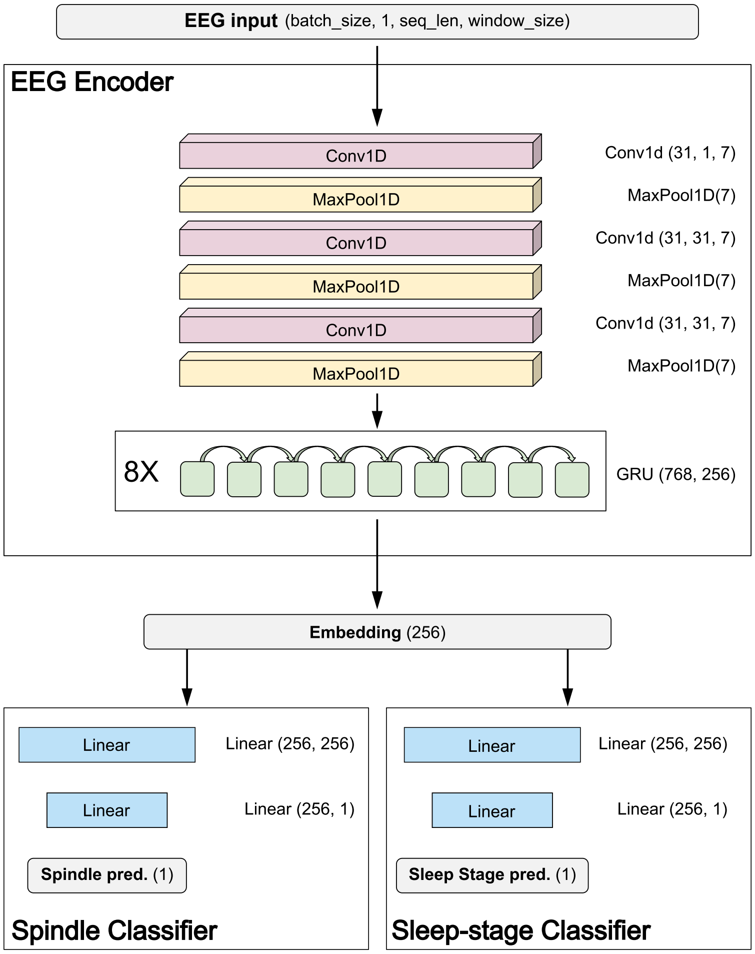


Figure 4.7 Dual-Task Model Architecture

fication (positive or negative).

4.8.6 Repeated Measures ANOVA - Sleep Staging Configuration * Age

Here, we show the full results of the ANOVA test performed between the results of SLA7 compared to LA7 depending on each sleep staging configuration to determine the significance of sleep staging in the computation of our online ground truth spindles.

This includes tables 4.6 4.7 4.8

Note: the assumption of sphericity is violated so we use the Greenhouse-Geisser sphericity correction.

Table 4.6 Within Subjects Effects - Sleep Staging Configuration

Cases	Sum of Squares	df	Mean Square	F	p	η^2
Sleep Staging	2.877	1.862	1.545	111.437	< .001	0.247
Sleep Staging * Age Cat	0.795	1.862	0.427	30.807	< .001	0.068
Residuals	3.434	247.675	0.014			

Table 4.7 Between Subjects Effects - Sleep Staging Configuration

Cases	Sum of Squares	df	Mean Square	F	p	η^2
Age Cat	0.885	1	0.885	32.110	< .001	0.076
Residuals	3.666	133	0.028			

Table 4.8 Post Hoc Comparisons - Sleep Staging

		Mean Difference	SE	t	p_{bonf}	p_{holm}
GroundTruth	None	0.158	0.014	11.406	< .001	< .001
	Online	0.194	0.014	14.044	< .001	< .001
None	Online	0.036	0.014	2.638	0.026	0.009

4.8.7 Repeated Measures ANOVA - Adaptation Configurations * Age for single night experiments

This section presents detailed for our Repeated Measures ANOVA comparing Adaptation configuration with age for single nights experiments. It includes tables 4.9, 4.10 and 4.11

Table 4.9 Within Subjects Effects - Adaptation configuration

Cases	Sum of Squares	df	Mean Square	F	p	η^2
Config	0.148	2.002	0.074	37.581	< .001	0.027
Config * Age Cat	0.025	2.002	0.013	6.375	0.002	0.004
Residuals	0.505	256.312	0.002			

Note: the assumption of sphericity is violated so we use the Greenhouse-Geisser sphericity correction.

Table 4.10 Between Subjects Effects - Adaptation configuration

Cases	Sum of Squares	df	Mean Square	F	p	η^2
Age Cat	0.136	1	0.136	3.654	0.058	0.024
Residuals	4.773	128	0.037			

Table 4.11 Post Hoc Comparisons - Age Cat * Config

		Mean Difference	SE	t	p_{bonf}
Older, Baseline	Younger, Baseline	−0.017	0.018	−0.978	1.000
	Older, Threshold	−0.022	0.006	−3.440	0.018
	Younger, Threshold	−0.060	0.018	−3.384	0.025
	Older, Fine-tuning	0.011	0.006	1.746	1.000
	Younger, Fine-tuning	−0.010	0.018	−0.582	1.000
	Older, Combined	−0.007	0.006	−1.042	1.000
	Younger, Combined	−0.059	0.018	−3.297	0.034
Younger, Baseline	Older, Threshold	−0.004	0.018	−0.241	1.000
	Younger, Threshold	−0.043	0.006	−6.685	< .001
	Older, Fine-tuning	0.028	0.018	1.597	1.000
	Younger, Fine-tuning	0.007	0.006	1.102	1.000
	Older, Combined	0.011	0.018	0.609	1.000
	Younger, Combined	−0.041	0.006	−6.445	< .001
Older, Threshold	Younger, Threshold	−0.039	0.018	−2.165	0.894
	Older, Fine-tuning	0.033	0.006	5.186	< .001
	Younger, Fine-tuning	0.011	0.018	0.637	1.000
	Older, Combined	0.015	0.006	2.398	0.475
	Younger, Combined	−0.037	0.018	−2.078	1.000
Younger, Threshold	Older, Fine-tuning	0.071	0.018	4.002	0.003
	Younger, Fine-tuning	0.050	0.006	7.787	< .001
	Older, Combined	0.054	0.018	3.014	0.084
	Younger, Combined	0.002	0.006	0.240	1.000
Older, Fine-tuning	Younger, Fine-tuning	−0.021	0.018	−1.201	1.000
	Older, Combined	−0.018	0.006	−2.788	0.156
	Younger, Combined	−0.070	0.018	−3.916	0.004
Younger, Fine-tuning	Older, Combined	0.004	0.018	0.213	1.000
	Younger, Combined	−0.048	0.006	−7.546	< .001
Older, Combined	Younger, Combined	−0.052	0.018	−2.928	0.110

4.8.8 Repeated Measures ANOVA - Spindle Density of adaptation configuration * Experiment Type

This Appendix presents the detailed results of our ANOVA analysis comparing the spindle density of various adaptation configurations (Baseline, Threshold, WeightAveraging, and Train) with the two experiment types (Random and SameSubject). It includes tables 4.12 and 4.13

Table 4.12 Within Subjects Effects - Spindle Density * Experiment

Cases	Sum of Squares	df	Mean Square	F	p	η^2
Config	1673.858	1.887	887.022	10.634	$< .001$	0.042
Config * experiment_type	226.324	1.887	119.935	1.438	0.240	0.006
Residuals	18101.349	217.011	83.412			

Note: the assumption of sphericity is violated so we use the Greenhouse-Geisser sphericity correction.

Table 4.13 Between Subjects Effects - Spindle Density * Experiment

Cases	Sum of Squares	df	Mean Square	F	p	η^2
experiment_type	9.216	1	9.216	0.054	0.817	2.324×10^{-4}
Residuals	19646.473	115	170.839			

4.8.9 Repeated Measures ANOVA - Spindle Density of adaptation configuration * Night Number

This section presents the detailed results of our ANOVA comparing spindle density with adaptation configuration for our multi night experiment. It includes tables 4.14, 4.15, and 4.16

Table 4.14 Within Subjects Effects - Spindle Density of adaptation configuration * Night Number

Cases	Sum of Squares	df	Mean Square	F	p	η^2
Config	10828.127	2.168	4994.211	34.993	$< .001$	0.134
Config * night_num	916.044	10.841	84.501	0.592	0.832	0.011
Residuals	34347.400	240.663	142.720			

Note: the assumption of sphericity is violated so we use the Greenhouse-Geisser sphericity correction.

Table 4.15 Between Subjects Effects - Spindle Density of adaptation configuration * Night Number

Cases	Sum of Squares	df	Mean Square	F	p	η^2
night_num	1751.581	5	350.316	1.187	0.320	0.022
Residuals	32767.598	111	295.204			

Table 4.16 Post Hoc Comparisons - Spindle Density of adaptation configuration * Night Number

		Mean Difference	SE	t	p_{holm}
Baseline	Threshold	6.660	1.030	6.465	< .001
	Fine-tuning	4.417	1.030	4.288	< .001
	WeightAveraging	10.231	1.030	9.931	< .001
	Combined	11.981	1.030	11.629	< .001
	ClassifierOnly	8.354	1.030	8.109	< .001
Threshold	Fine-tuning	−2.243	1.030	−2.177	0.119
	WeightAveraging	3.571	1.030	3.466	0.003
	Combined	5.320	1.030	5.164	< .001
	ClassifierOnly	1.694	1.030	1.644	0.207
Fine-tuning	WeightAveraging	5.814	1.030	5.644	< .001
	Combined	7.563	1.030	7.342	< .001
	ClassifierOnly	3.937	1.030	3.821	0.001
WeightAveraging	Combined	1.749	1.030	1.698	0.207
	ClassifierOnly	−1.877	1.030	−1.822	0.207
Combined	ClassifierOnly	−3.627	1.030	−3.520	0.003

4.8.10 ANOVA - RMS score compared to night number and configuration

This section shows the detailed results of the ANOVA tests performed to compare RMS score with night number and different adaptation configurations in our multi-night study. It includes tables 4.17 and 4.18

Table 4.17 ANOVA - rms_score

Cases	Sum of Squares	df	Mean Square	F	p	η^2
night_num	14659.143	5	2931.829	970.807	$< .001$	0.003
config	48077.866	5	9615.573	3183.975	$< .001$	0.009
night_num * config	7133.757	25	285.350	94.487	$< .001$	0.001
Residuals	$5.161 \times 10^{+6}$	1708899	3.020			

Table 4.18 Post Hoc Comparisons - config

		Mean Difference	SE	t	p_{tukey}
WeightAveraging	ClassifierOnly	0.193	0.005	37.106	$< .001$
	Combined	-0.013	0.006	-2.287	0.199
	Fine-tuning	0.371	0.005	76.972	$< .001$
	Baseline	0.447	0.005	97.462	$< .001$
	Threshold	0.238	0.005	47.803	$< .001$
ClassifierOnly	Combined	-0.205	0.005	-37.559	$< .001$
	Fine-tuning	0.178	0.005	37.883	$< .001$
	Baseline	0.254	0.004	56.979	$< .001$
	Threshold	0.045	0.005	9.231	$< .001$
Combined	Fine-tuning	0.383	0.005	74.991	$< .001$
	Baseline	0.460	0.005	93.870	$< .001$
	Threshold	0.250	0.005	47.604	$< .001$
Fine-tuning	Baseline	0.076	0.004	18.992	$< .001$
	Threshold	-0.133	0.004	-29.936	$< .001$
Baseline	Threshold	-0.209	0.004	-49.887	$< .001$

CHAPTER 5 CONCLUSIONS

This thesis project pursues a two-pronged approach:

1. **Development of a User-Friendly Portiloop System**: The project aimed to design and implement a practical Portiloop system, encompassing both hardware and software components, to facilitate its use in real-world experimental settings.
2. **Enhancement of Deep Learning Spindle Detection**: The project also investigated and evaluated methods to refine the Portiloop's deep learning-based spindle detection algorithm.

Objectives 1 and 2 two from section 1.2 are demonstrably fulfilled by the work presented in Chapter 3. Multiple laboratories have successfully adopted both versions of the Portiloop device to conduct CLAS sleep-spindle experiments. Nowadays, all components of the Portiloop system are open-source and available to any user seeking a cost-effective EEG recording solution. Furthermore, Chapter 4 establishes the feasibility of a continual learning, adaptable algorithm for progressive improvement of model performance over time, which in turn fulfills objectives 3 and 4 presented in section 1.2.

5.1 Summary of Works

Chapter 3 details the development process of the Portiloop's hardware and software. This iterative journey began with a prototype that eventually evolved into the functional Portiloop V1. While V1 served to validate the core design principles, it also revealed significant limitations that necessitated further improvements. Consequently, Portiloop V2 was developed to address these shortcomings and enhance user-friendliness. This refined version facilitated the execution of multiple overnight experiments, allowing subjects to take the Portiloop home for data collection.

In tandem with hardware development, the project entailed the creation of user-friendly software for seamless interaction with the Portiloop. This software constitutes an integral component, ensuring accessibility for users with limited technical expertise. The software underwent multiple iterations to arrive at its current form, meticulously incorporating user feedback to provide a comprehensive and user-centric feature set.

This thesis significantly contributes improvements to the deep learning model employed for sleep spindle detection. The project investigates the potential of leveraging continual learning paradigms to achieve incremental model improvement. Chapter 4 details the validation of an offline sleep spindle detection algorithm for online deep learning model enhancement. This strategy establishes an online performance baseline that facilitates subject-specific optimization and continuous model improvement across nights.

A key innovation of this work is the development of a dual-task model that utilizes a shared learned embedding for the classification of both sleep spindles and sleep stages. This combined approach lays the foundation for a continually learning algorithm, and the results demonstrate significant performance improvements on Portiloop-collected data.

5.2 Limitations

Chapter 4's proposed approach presents limitations in both design and validation. The initial experiments utilizing Portiloop data were constrained by the volume of available data per subject. Nightly recordings were insufficient to comprehensively evaluate the algorithm's behavior beyond three nights. Future studies with larger datasets could address this limitation. Additionally, a core motivation for developing this adaptation algorithm was to improve performance for older subjects. However, all Portiloop data employed in the experiments involved younger participants, hindering validation for the target demographic.

Another inherent design limitation stems from the Portiloop's distributed nature. The device is intended to facilitate parallel studies, where each subject takes the Portiloop home for data collection across nights. While our approach demonstrates improvement within a single device, it does not explore how multiple devices interact and share learned information across subjects participating in parallel experiments. Consequently, each Portiloop would optimize based on its specific data distribution, potentially deviating from the distribution observed by other devices. This divergence could lead to performance discrepancies between individual Portiloops over time. The current study's focus on a single device limits the generalizability of the findings to real-world scenarios with multiple devices in operation.

5.3 Future Research

The limitations identified in section 5.2 illuminate promising avenues for future investigation. Firstly, the efficacy of the adaptation approach for older subjects warrants exploration. Given the inherent challenges associated with sleep spindle detection in this population, subject-specific adaptation holds promise for enhanced performance. Expanding data collection efforts to include older participants would enable the validation of this hypothesis.

Secondly, real-world scenarios necessitate a mechanism for distributed performance improvement across Portiloop devices. Federated learning, a machine learning paradigm that facilitates collaborative model training on decentralized devices without sharing raw data, presents itself as a viable solution. Integrating the proposed adaptation approach with federated learning techniques would optimize data utilization from Portiloops while adhering to privacy constraints and eliminating the need for manual model retraining on individual devices. This would significantly enhance the applicability of the proposed approach to real-world, large-scale experiments.

The Portiloop's online detection algorithm demonstrates potential for applications beyond the realm of EEG. Ongoing efforts are exploring the integration of the Portiloop with magnetoencephalography (MEG) machines. By directly accessing data from the MEG interface, the Portiloop will gain access to source-localized information in the form of virtual sensors. This real-time data stream will then be fed into the Portiloop's model to guide the stimulation of spindles which would enable targeted stimulation of sleep spindles originating from specific brain regions.

Beyond its current application in sleep spindle detection, the Portiloop's versatile architecture inherently positions it for adaptation to various research domains, including brain-computer interfaces (BCI). The combination of its EEG sensing capabilities and deep learning-compatible hardware makes the Portiloop an ideal platform for the development of EEG-based input interfaces. The Portiloop has the potential to emerge as a powerful tool for furthering our understanding of brain-computer interaction and brain-robot interaction, mediated by brainwave signals.

REFERENCES

[1] [Online]. Available: https://ca.choosemuse.com/pages/shop

[2] [Online]. Available: https://shop.openbci.com/collections/frontpage

[3] [Online]. Available: https://www.audacityteam.org/

[4] L. M. Fernandez and A. Lüthi, “Sleep spindles: Mechanisms and functions,” *Physiological Reviews*, vol. 100, no. 2, p. 805–868, Apr 2020.

[5] N. Valenchon, Y. Bouteiller, H. R. Jourde, X. L’Heureux, M. Sobral, E. B. J. Coffey, and G. Beltrame, “The portiloop: A deep learning-based open science tool for closed-loop brain stimulation,” *PLOS ONE*, vol. 17, no. 8, pp. 1–20, 08 2022. [Online]. Available: https://doi.org/10.1371/journal.pone.0270696

[6] M. Tudor, L. Tudor, and K. I. Tudor, “Hans berger (1873-1941)–the history of electroencephalography,” *Acta medica Croatica: casopis Hravatske akademije medicinskih znanosti*, vol. 59, no. 4, pp. 307–313, 2005.

[7] J. Choi, M. Kwon, and S. C. Jun, “A systematic review of closed-loop feedback techniques in sleep studies—related issues and future directions,” *Sensors*, vol. 20, no. 10, p. 2770, 2020.

[8] S. M. Romanella, D. Roe, R. Paciorek, D. Cappon, G. Ruffini, A. Menardi, A. Rossi, S. Rossi, and E. Santarnecchi, “Sleep, noninvasive brain stimulation, and the aging brain: challenges and opportunities,” *Ageing research reviews*, vol. 61, p. 101067, 2020.

[9] H.-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. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0896627313002304

[10] S. Ruch, F. J. Schmidig, L. Knüsel, and K. Henke, “Closed-loop modulation of local slow oscillations in human nrem sleep,” *NeuroImage*, vol. 264, p. 119682, 2022.

[11] M. Navarrete, J. Schneider, H.-V. V. Ngo, M. Valderrama, A. J. Casson, and P. A. Lewis, “Examining the optimal timing for closed-loop auditory stimulation of slow-wave sleep in young and older adults,” *Sleep*, vol. 43, no. 6, p. zsz315, 2020.

[12] S. M. Purcell, D. S. Manoach, C. Demanuele, B. E. Cade, S. Mariani, R. Cox, G. Panagiotaropoulou, R. Saxena, J. Q. Pan, J. W. Smoller, S. Redline, and R. Stickgold, "Characterizing sleep spindles in 11,630 individuals from the national sleep research resource," *Nature Communications*, vol. 8, p. 15930, 2017. [Online]. Available: https://doi.org/10.1038/ncomms15930

[13] K. R. Peters, L. B. Ray, S. Fogel, V. Smith, and C. T. Smith, "Age differences in the variability and distribution of sleep spindle and rapid eye movement densities," *PLOS ONE*, vol. 9, no. 3, pp. 1–11, 03 2014. [Online]. Available: https://doi.org/10.1371/journal.pone.0091047

[14] A. von Lühmann, J. Addesa, S. Chandra, A. Das, M. Hayashibe, and A. Dutta, "Neural interfacing non-invasive brain stimulation with nirs-eeg joint imaging for closed-loop control of neuroenergetics in ischemic stroke," in *2017 8th International IEEE/EMBS Conference on Neural Engineering (NER)*, 2017, pp. 349–353.

[15] C.-T. Lin, Y.-C. Chen, T.-Y. Huang, T.-T. Chiu, L.-W. Ko, S.-F. Liang, H.-Y. Hsieh, S.-H. Hsu, and J.-R. Duann, "Development of wireless brain computer interface with embedded multitask scheduling and its application on real-time driver's drowsiness detection and warning," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 5, pp. 1582–1591, 2008.

[16] C. M. McCrimmon, J. L. Fu, M. Wang, L. S. Lopes, P. T. Wang, A. Karimi-Bidhendi, C. Y. Liu, P. Heydari, Z. Nenadic, and A. H. Do, "Performance assessment of a custom, portable, and low-cost brain–computer interface platform," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 10, pp. 2313–2320, 2017.

[17] A. Galván, "The need for sleep in the adolescent brain," *Trends in cognitive sciences*, vol. 24, no. 1, pp. 79–89, 2020.

[18] B. Miner and M. Kryger, "Sleep in the aging population," *Sleep Medicine Clinics*, vol. 15, pp. 311–318, 06 2020.

[19] U. Ziemann and H. R. Siebner, "Inter-subject and inter-session variability of plasticity induction by non-invasive brain stimulation: boon or bane?" *Brain Stimulation: Basic, Translational, and Clinical Research in Neuromodulation*, vol. 8, no. 3, pp. 662–663, 2015.

[20] S. Brodt, M. Inostroza, N. Niethard, and J. Born, "Sleep—a brain-state serving systems memory consolidation," *Neuron*, vol. 111, no. 7, pp. 1050–1075, 2023.

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

[22] J. G. Klinzing, N. Niethard, and J. Born, "Mechanisms of systems memory consolidation during sleep," *Nature neuroscience*, vol. 22, no. 10, pp. 1598–1610, 2019.

[23] W. Li, L. Ma, G. Yang, and W.-B. Gan, "REM sleep selectively prunes and maintains new synapses in development and learning," *Nature neuroscience*, vol. 20, no. 3, pp. 427–437, 2017.

[24] M. Strauss, L. Griffon, P. Van Beers, M. Elbaz, J. Bouziotis, F. Sauvet, M. Chennaoui, D. Léger, and P. Peigneux, "Order matters: sleep spindles contribute to memory consolidation only when followed by rapid-eye-movement sleep," *Sleep*, vol. 45, no. 4, p. zsac022, 2022.

[25] H. Jourde, R. Merlo, M. Brooks, M. Rowe, and E. Coffey, "The neurophysiology of closed-loop auditory stimulation in sleep: A magnetoencephalography study," *The European journal of neuroscience*, vol. 59, 09 2023.

[26] V. Latreille, N. von Ellenrieder, L. Peter-Derex, F. Dubeau, J. Gotman, and B. Frauscher, "The human k-complex: insights from combined scalp-intracranial eeg recordings," *Neuroimage*, vol. 213, p. 116748, 2020.

[27] B. A. Riedner, B. K. Hulse, M. J. Murphy, F. Ferrarelli, and G. Tononi, "Temporal dynamics of cortical sources underlying spontaneous and peripherally evoked slow waves," *Progress in brain research*, vol. 193, pp. 201–218, 2011.

[28] C. N. Oyanedel, E. Durán, N. Niethard, M. Inostroza, and J. Born, "Temporal associations between sleep slow oscillations, spindles and ripples," *European Journal of Neuroscience*, vol. 52, no. 12, pp. 4762–4778, 2020.

[29] N. L. Hauglund, C. Pavan, and M. Nedergaard, "Cleaning the sleeping brain–the potential restorative function of the glymphatic system," *Current Opinion in Physiology*, vol. 15, pp. 1–6, 2020.

[30] M. Ulander, F. Rångtell, and J. Theorell-Haglöw, "Sleep measurements in women," *Sleep Medicine Clinics*, vol. 16, no. 4, pp. 635–648, 2021.

[31] J. Nicolas, B. R. King, D. Levesque, L. Lazzouni, E. Coffey, S. Swinnen, J. Doyon, J. Carrier, and G. Albouy, "Sigma oscillations protect or reinstate motor memory

depending on their temporal coordination with slow waves," *Elife*, vol. 11, p. e73930, 2022.

[32] M.-A. Savard, A. G. Sares, E. B. Coffey, and M. L. Deroche, "Specificity of affective responses in misophonia depends on trigger identification," *Frontiers in Neuroscience*, vol. 16, p. 879583, 2022.

[33] A. Boutin, E. Gabitov, B. Pinsard, A. Bore, J. Carrier, and J. Doyon, "Temporal cluster-based organization of sleep spindles underlies motor memory consolidation," *Proceedings of the Royal Society B*, vol. 291, 01 2024.

[34] L. M. Fernandez and A. Lüthi, "Sleep spindles: mechanisms and functions," *Physiological reviews*, vol. 100, no. 2, pp. 805–868, 2020.

[35] D. P. Brunner, D.-J. Dijk, I. Tobler, and A. A. Borbély, "Effect of partial sleep deprivation on sleep stages and eeg power spectra: evidence for non-rem and rem sleep homeostasis," *Electroencephalography and clinical neurophysiology*, vol. 75, no. 6, pp. 492–499, 1990.

[36] M. Barakat, J. Doyon, K. Debas, G. Vandewalle, A. Morin, G. Poirier, N. Martin, M. Lafortune, A. Karni, L. Ungerleider *et al.*, "Fast and slow spindle involvement in the consolidation of a new motor sequence," *Behavioural brain research*, vol. 217, no. 1, pp. 117–121, 2011.

[37] D. Grimaldi, N. A. Papalambros, P. C. Zee, and R. G. Malkani, "Neurostimulation techniques to enhance sleep and improve cognition in aging," *Neurobiology of disease*, vol. 141, p. 104865, 2020.

[38] B. E. Muehlroth, M. C. Sander, Y. Fandakova, T. H. Grandy, B. Rasch, Y. L. Shing, and M. Werkle-Bergner, "Precise slow oscillation–spindle coupling promotes memory consolidation in younger and older adults," *Scientific reports*, vol. 9, no. 1, p. 1940, 2019.

[39] Z. Muñoz-Torres, U. Jiménez-Correa, and C. J. Montes-Rodríguez, "Sex differences in brain oscillatory activity during sleep and wakefulness in obstructive sleep apnea," *Journal of sleep research*, vol. 29, no. 4, p. e12977, 2020.

[40] S. M. Romanella, D. Roe, R. Paciorek, D. Cappon, G. Ruffini, A. Menardi, A. Rossi, S. Rossi, and E. Santarnecchi, "Sleep, noninvasive brain stimulation, and the aging brain: challenges and opportunities," *Ageing research reviews*, vol. 61, p. 101067, 2020.

[41] R. Boyce, S. D. Glasgow, S. Williams, and A. Adamantidis, “Causal evidence for the role of rem sleep theta rhythm in contextual memory consolidation,” *Science*, vol. 352, no. 6287, pp. 812–816, 2016.

[42] R. Boyce, S. Williams, and A. Adamantidis, “Rem sleep and memory,” *Current Opinion in Neurobiology*, vol. 44, pp. 167–177, 2017.

[43] J. N. Cousins and G. Fernández, “The impact of sleep deprivation on declarative memory,” *Progress in brain research*, vol. 246, pp. 27–53, 2019.

[44] I. C. Hutchison and S. Rathore, “The role of rem sleep theta activity in emotional memory,” *Frontiers in Psychology*, vol. 6, 2015. [Online]. Available: https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2015.01439

[45] S. M. Fogel, L. B. Ray, L. Binnie, and A. M. Owen, “How to become an expert: a new perspective on the role of sleep in the mastery of procedural skills,” *Neurobiology of Learning and Memory*, vol. 125, pp. 236–248, 2015.

[46] D. Campos-Beltrán and L. Marshall, “Changes in sleep eeg with aging in humans and rodents,” *Pflügers Archiv-European Journal of Physiology*, vol. 473, no. 5, pp. 841–851, 2021.

[47] J. R. D. Espiritu, “Aging-related sleep changes,” *Clinics in geriatric medicine*, vol. 24, no. 1, pp. 1–14, 2008.

[48] J. G. Klinzing, M. Mölle, F. Weber, G. Supp, J. F. Hipp, A. K. Engel, and J. Born, “Spindle activity phase-locked to sleep slow oscillations,” *Neuroimage*, vol. 134, pp. 607–616, 2016.

[49] R. F. Helfrich, B. A. Mander, W. J. Jagust, R. T. Knight, and M. P. Walker, “Old brains come uncoupled in sleep: Slow wave-spindle synchrony, brain atrophy, and forgetting,” *Neuron*, vol. 97, no. 1, pp. 221–230.e4, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0896627317310735

[50] S. Datta, C. M. Knapp, R. Koul-Tiwari, and A. Barnes, “The homeostatic regulation of rem sleep: A role for localized expression of brain-derived neurotrophic factor in the brainstem,” *Behavioural Brain Research*, vol. 292, pp. 381–392, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166432815300668

[51] P. Champetier, C. André, F. Weber, S. Rehel, V. Ourry, A. Laniepce, A. Lutz, F. Bertran, N. Cabé, A.-L. Pitel, G. Poisnel, V. Sayette, D. Vivien, G. Chételat, and

G. Rauchs, “Age-related changes in fast spindle clustering during nrem sleep and their relevance for memory consolidation,” *Sleep*, vol. 46, 11 2022.

[52] J. Choi, M. Kwon, and S. Jun, “A systematic review of closed-loop feedback techniques in sleep studies—related issues and future directions,” *Sensors*, vol. 20, p. 2770, 05 2020.

[53] M. Navarrete, J. Schneider, H.-V. Ngo-Dehning, M. Valderrama, A. Casson, and P. Lewis, “Examining the optimal timing for closed loop auditory stimulation of slow wave sleep in young and older adults,” *Sleep*, vol. 43, 12 2019.

[54] T. Rosinvil, J. Bouvier, J. Dubé, A. Lafrenière, M. Bouchard, J. Cronier, N. Gosselin, J. Carrier, and j.-m. Lina, “Are age and sex effects on sleep slow waves only a matter of eeg amplitude?” *Sleep*, vol. 44, 09 2020.

[55] R. Vertes and K. Eastman, *The case against memory consolidation in REM sleep*, 01 2001, pp. 75–84.

[56] S. M. FOGEL and C. T. SMITH, “Learning-dependent changes in sleep spindles and stage 2 sleep,” *Journal of Sleep Research*, vol. 15, no. 3, pp. 250–255, 2006. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2869.2006.00522.x

[57] K. Nasr, D. Haslacher, E. Dayan, N. Censor, L. G. Cohen, and S. R. Soekadar, “Breaking the boundaries of interacting with the human brain using adaptive closed-loop stimulation,” *Progress in Neurobiology*, vol. 216, p. 102311, 2022.

[58] G. Soleimani, M. A. Nitsche, T. O. Bergmann, F. Towhidkhah, I. R. Violante, R. Lorenz, R. Kuplicki, A. Tsuchiyagaito, B. Mulyana, A. Mayeli *et al.*, “Closing the loop between brain and electrical stimulation: towards precision neuromodulation treatments,” *Translational psychiatry*, vol. 13, no. 1, p. 279, 2023.

[59] F. Frohlich and L. Townsend, “Closed-loop transcranial alternating current stimulation: towards personalized non-invasive brain stimulation for the treatment of psychiatric illnesses,” *Current Behavioral Neuroscience Reports*, vol. 8, pp. 51–57, 2021.

[60] B. Rasch and J. Born, “About sleep’s role in memory,” *Physiological Reviews*, vol. 93, no. 2, pp. 681–766, 2013, pMID: 23589831. [Online]. Available: https://doi.org/10.1152/physrev.00032.2012

[61] U. Hassan, G. B. Feld, and T. O. Bergmann, “Automated real-time eeg sleep spindle detection for brain-state-dependent brain stimulation,” *Journal of sleep research*, vol. 31, no. 6, p. e13733, 2022.

[62] K. Lacourse, B. Yetton, S. Mednick, and S. Warby, “Massive online data annotation, crowdsourcing to generate high quality sleep spindle annotations from eeg data,” *Scientific Data*, vol. 7, 06 2020.

[63] 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. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/jsr.12169

[64] E. J. Wamsley, M. A. Tucker, A. K. Shinn, K. E. Ono, S. K. McKinley, A. V. Ely, D. C. Goff, R. Stickgold, and D. S. Manoach, “Reduced sleep spindles and spindle coherence in schizophrenia: Mechanisms of impaired memory consolidation?” *Biological Psychiatry*, vol. 71, no. 2, p. 154–161, Jan 2012.

[65] Y. Gu, F. Han, L. E. Sainburg, M. M. Schade, and X. Liu, “"simultaneous eeg and fmri signals during sleep from humans",” 2023.

[66] ——, “"simultaneous eeg and fmri signals during sleep from humans",” 2023.

[67] K. Lacourse, J. Delfrate, J. Beaudry, P. Peppard, and S. C. Warby, “A sleep spindle detection algorithm that emulates human expert spindle scoring,” *Journal of Neuroscience Methods*, vol. 316, pp. 3–11, 2019, methods and models in sleep research: A Tribute to Vincenzo Crunelli. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165027018302504

[68] F. Ferrarelli, “Reduced sleep spindle activity in schizophrenia patients,” *American Journal of Psychiatry*, vol. 164, no. 3, p. 483, Mar 2007.

[69] M. Mölle, L. Marshall, S. Gais, and J. Born, “Grouping of spindle activity during slow oscillations in human non-rapid eye movement sleep,” *The Journal of Neuroscience*, vol. 22, no. 24, p. 10941–10947, Dec 2002.

[70] P. M. Kulkarni, Z. Xiao, E. J. Robinson, A. S. Jami, J. Zhang, H. Zhou, S. E. Henin, A. A. Liu, R. S. Osorio, J. Wang, and Z. Chen, “A deep learning approach for real-time detection of sleep spindles,” *Journal of Neural Engineering*, vol. v 16, n 3, 2019.

[71] N. I. Tapia and P. A. Estevez, “Red: Deep recurrent neural networks for sleep eeg event detection,” in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, Jul. 2020. [Online]. Available: http://dx.doi.org/10.1109/IJCNN48605.2020.9207719

[72] N. Yasuhara, T. Natori, M. Hayashi, and N. Aikawa, "A study on automatic detection of sleep spindles using a long short-term memory network," *2019 IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS)*, vol. 2019-August, pp. 45–48, 8 2019.

[73] D. Tan, R. Zhao, J. Sun, and W. Qin, "Sleep spindle detection using deep learning: A validation study based on crowdsourcing," *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference*, vol. 2015, p. 2828—2831, August 2015. [Online]. Available: https://doi.org/10.1109/EMBC.2015.7318980

[74] M. Kafashan, G. Gupte, P. Kang, O. Hyche, A. Luong, G. Prateek, Y.-E. S. Ju, and B. J. A. Palanca, "A personalized semi-automatic sleep spindle detection (psasd) framework," *Journal of Neuroscience Methods*, p. 110064, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165027024000098

[75] A. Koushik, J. Amores, and P. Maes, "Real-time sleep staging using deep learning on a smartphone for a wearable eeg," 2018.

[76] E. Bresch, U. Großekathöfer, and G. Garcia-Molina, "Recurrent deep neural networks for real-time sleep stage classification from single channel eeg," *Frontiers in Computational Neuroscience*, vol. 12, 2018. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fncom.2018.00085

[77] A. Patanaik, J. L. Ong, J. J. Gooley, S. Ancoli-Israel, and M. W. L. Chee, "An end-to-end framework for real-time automatic sleep stage classification," *Sleep*, vol. 41, no. 5, p. zsy041, 03 2018. [Online]. Available: https://doi.org/10.1093/sleep/zsy041

[78] A. Koushik, J. Amores, and P. Maes, "Real-time smartphone-based sleep staging using 1-channel eeg," in *2019 IEEE 16th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, 2019, pp. 1–4.

[79] A. F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018.

[80] H. Robbins and S. Monro, "A Stochastic Approximation Method," *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400 – 407, 1951. [Online]. Available: https://doi.org/10.1214/aoms/1177729586

[81] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[82] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” 2019.

[83] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[84] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” 2014.

[85] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, “Continual lifelong learning with neural networks: A review,” *Neural Networks*, vol. 113, pp. 54–71, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608019300231

[86] R. M. French, “Catastrophic forgetting in connectionist networks,” *Trends in cognitive sciences*, vol. 3, pp. 128–135, 1999.

[87] L. Wang, X. Zhang, H. Su, and J. Zhu, “A comprehensive survey of continual learning: Theory, method and application,” 2024.

[88] M. Wortsman, G. Ilharco, S. Y. G. R. Roelofs, R. Gontijo-Lopes, A. S. Morcos, H. N. A. Farhadi, Y. Carmon, S. Kornblith, and L. Schmidt, “Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time.”

[89] [Online]. Available: https://www.ti.com/product/ADS1299

[90] [Online]. Available: https://jupyter.org/

[91] [Online]. Available: https://nicegui.io/

[92] [Online]. Available: https://huggingface.co/spaces

[93] [Online]. Available: https://www.selenium.dev/

[94] R. Polanía, M. A. Nitsche, and C. C. Ruff, “Studying and modifying brain function with non-invasive brain stimulation,” *Nature neuroscience*, vol. 21, no. 2, pp. 174–187, 2018.

[95] M. Ramot and A. Martin, “Closed-loop neuromodulation for studying spontaneous activity and causality,” *Trends in cognitive sciences*, vol. 26, no. 4, pp. 290–299, 2022.

[96] C. Zrenner, P. Belardinelli, F. Müller-Dahlhaus, and U. Ziemann, “Closed-loop neuroscience and non-invasive brain stimulation: A tale of two loops,” *Frontiers in Cellular Neuroscience*, vol. 10, 2016. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fncel.2016.00092

[97] A. Vassileva, D. van Blooijs, F. Leijten, and G. Huiskamp, “Neocortical electrical stimulation for epilepsy: Closed-loop versus open-loop,” *Epilepsy Research*, vol. 141, pp. 95–101, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0920121117300578

[98] A. N. Belkacem, N. Jamil, S. Khalid, and F. Alnajjar, “On closed-loop brain stimulation systems for improving the quality of life of patients with neurological disorders,” *Frontiers in human neuroscience*, vol. 17, p. 1085173, 2023.

[99] P. Champetier, C. André, F. D. Weber, S. Rehel, V. Ourry, A. Laniepce, A. Lutz, F. Bertran, N. Cabé, A.-L. Pitel *et al.*, “Age-related changes in fast spindle clustering during non-rapid eye movement sleep and their relevance for memory consolidation,” *Sleep*, vol. 46, no. 5, p. zsac282, 2023.

[100] L. M. Fernandez and A. Lüthi, “Sleep spindles: mechanisms and functions,” *Physiological reviews*, vol. 100, no. 2, pp. 805–868, 2020.

[101] E. J. Wamsley, M. A. Tucker, A. K. Shinn, K. E. Ono, S. K. McKinley, A. V. Ely, D. C. Goff, R. Stickgold, and D. S. Manoach, “Reduced sleep spindles and spindle coherence in schizophrenia: Mechanisms of impaired memory consolidation?” *Biological Psychiatry*, vol. 71, no. 2, pp. 154–161, 2012, functional Consequences of Altered Cortical Development in Schizophrenia.

[102] G. Vantomme, A. Osorio-Forero, A. Lüthi, and L. M. Fernandez, “Regulation of local sleep by the thalamic reticular nucleus,” *Frontiers in neuroscience*, vol. 13, p. 461194, 2019.

[103] H. Bastuji, P. Lamouroux, M. Villalba, M. Magnin, and L. Garcia-Larrea, “Local sleep spindles in the human thalamus,” *The Journal of Physiology*, vol. 598, no. 11, pp. 2109–2124, 2020.

[104] C. W. Dickey, A. Sargsyan, J. R. Madsen, E. N. Eskandar, S. S. Cash, and E. Halgren, “Travelling spindles create necessary conditions for spike-timing-dependent plasticity in humans,” *Nature communications*, vol. 12, no. 1, p. 1027, 2021.

[105] H. R. Jourde, R. Merlo, M. Brooks, M. Rowe, and E. B. Coffey, “The neurophysiology of closed-loop auditory stimulation in sleep: A magnetoencephalography study,” *European Journal of Neuroscience*, vol. 59, no. 4, pp. 613–640, 2024.

[106] G. Piantoni, E. Halgren, and S. S. Cash, “Spatiotemporal characteristics of sleep spindles depend on cortical location,” *Neuroimage*, vol. 146, pp. 236–245, 2017.

[107] C. Gonzalez, X. Jiang, J. Gonzalez-Martinez, and E. Halgren, “Human spindle variability,” *Journal of Neuroscience*, vol. 42, no. 22, pp. 4517–4537, 2022.

[108] F. D. Weber, G. G. Supp, J. G. Klinzing, M. Mölle, A. K. Engel, and J. Born, “Coupling of gamma band activity to sleep spindle oscillations–a combined eeg/meg study,” *NeuroImage*, vol. 224, p. 117452, 2021.

[109] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, “A survey on federated learning,” *Knowledge-Based Systems*, vol. 216, p. 106775, 2021.

[110] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.