# TOWARDS CLOSED-LOOP SLEEP MONITORING IN PARKINSON'S DISEASE: SELF-SUPERVISED LEARNING STRATEGIES FOR SLEEP STAGE CLASSIFICATION

KRISTAL DOGA MENGUC

A THESIS SUBMITTED TO THE FACULTY OF GRADUATE STUDIES IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTERS OF SCIENCE

GRADUATE PROGRAM IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE YORK UNIVERSITY TORONTO, ONTARIO

NOVEMBER 2025

© KRISTAL DOGA MENGUC, 2025

# ABSTRACT

Parkinson's disease (PD) involves severe sleep disturbances that may accelerate neurodegeneration. Closed-loop deep brain stimulation (DBS) is a promising therapeutic solution but requires accurate, real-time sleep-stage classification from subthalamic nucleus signals, a task where conventional models generalize poorly. To address pronounced class imbalance and improve cross-patient generalization, this work introduces a self-supervised transformer framework. The model employs a masked autoencoder strategy, pretrained on large public EEG/ECoG datasets from healthy subjects to learn balanced representations of all sleep stages, thereby improving discrimination of rare classes. While pretraining yielded limited overall improvement, it specifically enhanced N3 stage identification and next-sleep-stage prediction. Contrastive self-supervision (70%) significantly outperformed a reconstruction-based approach (62%). Furthermore, spectral feature extraction proved more effective than temporal CNN features for distinguishing commonly mispredicted stages. Future work will focus on hybrid reconstruction-contrastive losses, incorporating spectral feature usage to forecasting and extending the forecasting horizon to predict multiple subsequent stages for proactive neuromodulation.

# TABLE OF CONTENTS

# LIST OF FIGURES

# Chapter 1: Introduction

### 1.1 Parkinson's Disease and Sleep

Parkinson's disease (PD) is primarily recognized for its motor impairments—bradykinesia, tremor, and rigidity—but an equally debilitating dimension of the disorder lies in its non-motor symptoms, particularly sleep disturbances [1]. Even in the early stages of the disease, patients frequently experience insomnia, excessive daytime sleepiness, rapid eye movement (REM) sleep behavior disorder (RBD), and fragmented nocturnal sleep [2]. These disturbances often precede the onset of motor symptoms, suggesting that dysfunctions within the neural circuits governing sleep–wake regulation are integral to PD pathology rather than mere secondary effects of dopaminergic degeneration [2].

The implications of disturbed sleep in PD extend far beyond fatigue or reduced quality of life. Sleep plays a crucial role in maintaining neural homeostasis, particularly through the glymphatic system—a brain-wide clearance network that facilitates the removal of neurotoxic waste products such as misfolded proteins and metabolic byproducts [3]. During deep slow-wave sleep (N3 stage), the interstitial spaces of the brain expand, allowing cerebrospinal fluid to more effectively wash out accumulated toxins, including α-synuclein and β-amyloid [4, 5]. The pathological aggregation of these proteins is a defining feature of PD and Alzheimer's disease, respectively. Consequently, chronic sleep disruption can impair this clearance mechanism, accelerating neurodegenerative processes and exacerbating disease progression [2].

Longitudinal studies have demonstrated that sleep disturbances in PD tend to intensify as the disease advances. Over multi-year follow-up periods, patients have shown progressive declines in sleep efficiency, increased nocturnal awakenings, and reduced time spent in restorative sleep stages—even when motor symptoms are well-managed through pharmacological therapy [2]. Moreover, the severity of sleep dysfunction has been found to correlate strongly with both cognitive decline and the overall rate of disease progression [3]. These findings position sleep not only as a biomarker of neurodegenerative activity but also as a potential therapeutic target. Improving sleep quality in PD could therefore have a dual benefit—alleviating non-motor symptoms while slowing pathological protein accumulation and neuronal loss.

The restoration of normal sleep architecture in PD is thus of both clinical and neuroprotective importance. Addressing sleep dysfunction may offer a means to interrupt feedback loops between disrupted sleep,

neuroinflammation, and protein aggregation. This recognition provides the conceptual foundation for exploring neuromodulatory interventions, such as deep brain stimulation (DBS), which may hold potential to modulate sleep dynamics and restore physiological oscillatory balance in PD patients.

### 1.2 Deep Brain Stimulation and Sleep Modulation

DBS of the subthalamic nucleus (STN) was originally developed to alleviate the motor symptoms of PD. However, a growing body of evidence now suggests that nocturnal stimulation can also produce substantial, immediate improvements in sleep architecture. Several clinical studies have reported increases in total sleep time and reductions in wakefulness after sleep onset during high-frequency STN stimulation [6]. For example, one cohort of PD patients with severe insomnia exhibited a 47% increase in total sleep duration and markedly reduced nighttime akinesia after several months of chronic DBS treatment [6]. These findings indicate that conventional DBS—although primarily optimized for motor control—can inadvertently enhance sleep quality, possibly by modulating oscillatory activity within basal ganglia–thalamo–cortical networks involved in arousal and sleep regulation.

Nevertheless, the clinical picture remains nuanced. The improvement in sleep that accompanies conventional (continuous) DBS appears to be a secondary effect rather than a result of targeted modulation of sleep physiology. When stimulation is discontinued—even after extended use—patients often experience a rapid return of insomnia and sleep fragmentation, suggesting that continuous DBS does not induce lasting plastic changes in sleep-regulating circuits [6]. This transient benefit underscores the limitations of conventional DBS and motivates the development of adaptive deep brain stimulation (aDBS) paradigms that dynamically adjust stimulation parameters to optimize sleep rather than motor function alone.

Recent research in aDBS has explored the potential of tailoring stimulation to specific sleep stages, particularly N3 or slow-wave sleep, which plays a vital role in metabolite clearance and synaptic homeostasis [7]. In PD, reductions in slow-wave activity have been associated with faster disease progression, raising the possibility that enhancing N3 sleep could have neuroprotective value [7]. A proof-of-concept study showed that N3-targeted stimulation accurately modulated slow-wave activity in the majority of detected epochs, with minimal erroneous stimulation during non-N3 periods [7]. Classifier performance also remained stable under stimulation, suggesting that DBS artifacts did not meaningfully interfere with stage detection. These findings suggest that optimizing DBS to enhance slow-wave sleep could improve rest quality and potentially slow disease progression in PD.

However, implementing sleep-targeted DBS outside the laboratory or hospital setting requires a reliable, automated sleep-stage classifier that can operate in real time from implanted sensing electrodes. Mounting evidence suggests that this is feasible using STN local field potentials (LFPs) alone. Both animal studies and intracranial human recordings have revealed distinct STN spectral signatures across vigilance states [8]. In non-human primates, for example, STN power in delta (1–4 Hz), theta (4–8 Hz), alpha (8–13 Hz), and beta (13–32 Hz) bands systematically varies across N1, N2, N3, REM, and wake stages. Classifiers trained on these spectral features have achieved accuracies ranging from 62%–75% in macaques, approaching the performance of simultaneous EEG-based models [8]. These results demonstrate that STN signals contain sufficient stage-relevant information to support automated classification, despite being spatially localized and potentially noisier than cortical EEG.

Building on these insights, two key studies have attempted automatic sleep-stage classification in PD patients using STN-LFP recordings [9, 10]. Both employed traditional machine learning approaches based on engineered spectral inputs. The frequency-domain features were extracted via Short-Time Fourier Transform (STFT) and Welch's method to compute power spectral densities from one-second LFP windows and binned into eight separate frequency bands. Because N2 and N3 displayed overlapping frequency characteristics, the authors merged these stages and trained support vector machine (SVM) and decision tree (DT) classifiers for binary and multiclass sleep labeling [9]. Personalized, single-subject models achieved accuracies of approximately 80.8% (SVM) and 75.6% (DT), while cross-subject "unified" models dropped to 65.6% (SVM) and 61.4% (DT), indicating poor generalization to unseen patients [9]. When N3 was treated separately, multiclass SVM accuracy fell further to around 58%, reflecting the difficulty of distinguishing N2 from N3 and the scarcity of N3 epochs [9].

The second study employed a feedforward neural network with two hidden layers of 1,000 ReLU units each, trained on the eight spectral bands ranging from delta to high-frequency oscillations (up to 350 Hz) [10]. Their results mirrored those of the SVM study: strong within-patient performance but substantial degradation under leave-one-patient-out testing. Both studies reported consistent confusion between N1 and N2–N3, as well as poor classification of N3 due to its rarity. Collectively, these studies demonstrate that although STN-LFPs carry information relevant for sleep staging, conventional classifiers like SVMs and feedforward networks fail to model the temporal and cross-frequency structure inherent to PD sleep.

### 1.3 Deep Learning for EEG and Intracranial Signal Classification

To address the challenges conventional classifiers face, recent work has increasingly focused on deep learning (DL) architectures capable of learning data-driven representations from neural signals. Rather than relying on fixed spectral features, DL models can uncover hierarchical temporal and spectral

structure directly from raw or minimally processed data. This adaptability is particularly beneficial for PD recordings, where variability, noise, and stimulation artifacts often obscure sleep-related activity. By learning representations end-to-end, DL approaches have shown improved robustness and generalization in both EEG and intracranial signal classification tasks [11, 12].

Convolutional neural networks (CNNs), for example, can automatically extract relevant frequency and spatial patterns from raw or minimally processed signals. These models have achieved notable success in scalp EEG-based sleep classification, emotion decoding, and seizure detection [11]. However, CNNs have inherent limitations when applied to sleep staging: while they excel at capturing local correlations in time or frequency, they struggle to model long-range temporal dependencies that characterize the progression of sleep cycles over minutes [12].

To address these limitations, recent research has increasingly adopted transformer architectures—originally developed for natural language processing but now widely applied to time series and neural signal analysis [12]. Transformers employ self-attention mechanisms that enable each time step in a sequence to dynamically attend to all others, removing the locality and sequential processing constraints inherent to convolutional and recurrent models. This design allows transformers to capture both short- and long-range temporal dependencies, making them well-suited for sleep stage classification, where subtle and extended transitions between stages unfold over several minutes. A detailed overview of the transformer architecture and its multi-head attention mechanism is provided in Chapter 2.2.

More recent hybrid architectures, such as Conformers, integrate convolutional and transformer modules to combine the strengths of both paradigms. The convolutional layers perform localized feature extraction in the spectral or spatial domain, while the transformer layers model global temporal dependencies and aggregate these representations hierarchically. Such hybrid models represent a conceptual bridge between earlier CNN-based approaches and the newer generation of attention-driven temporal frameworks, offering a unified means of capturing both local spectral patterns and long-range temporal structure in neural activity [12].

### 1.4 Self-Supervised Representation Learning in Neural Data

While transformer-based models have improved accuracy and generalization, their success still depends heavily on large labeled datasets—an impractical requirement in clinical settings. Labeling sleep data, particularly intracranial recordings from PD patients, is labor-intensive and requires expert scoring. To overcome this bottleneck, researchers have increasingly turned to self-supervised learning (SSL), which

enables representation learning from unlabeled data by defining auxiliary tasks that do not require manual annotations.

The success of SSL in EEG has inspired its application to intracranial neural data, where electrode placement variability and noise make cross-subject generalization challenging. BrainBERT pioneered the use of transformer-based masked spectrogram modeling for neural data [13]. Adapted from language models like BERT, BrainBERT learns contextualized embeddings by reconstructing masked regions of neural spectrograms. Pretraining across multiple subjects and electrode configurations enables the model to generalize effectively to new patients and recording conditions [13].

BrainBERT demonstrated that self-supervised transformers can capture multiscale temporal–spectral dependencies and learn biologically meaningful representations without explicit labels. Its design acknowledges the fractal, scale-free nature of neural dynamics, aligning closely with the hierarchical structure observed in subthalamic and cortical activity during sleep.

Building on this idea, Brant introduced a dual-transformer encoder—one temporal and one spatial—to simultaneously model cross-channel correlations and long-term dynamics [14]. By aggregating timestamps into patches, Brant efficiently captures both fine-grained temporal details and global trends, performing well across diverse downstream tasks, including seizure prediction, signal imputation, and phase forecasting.

Most recently, NeuroNet, a hybrid self-supervised transformer framework for single-channel EEG, has pushed the boundaries of SSL in sleep stage classification [15]. By integrating both masked reconstruction and contrastive pretext tasks, NeuroNet achieves superior transferability across datasets and subjects, setting a new benchmark for scalable, label-efficient sleep-stage modeling.

### 1.5 Proposed Framework

Building on the insights from previous sections, this work introduces a self-supervised transformer framework for sleep stage classification using intracranial recordings from PD patients. The model is designed with closed-loop DBS compatibility in mind, operating directly on minimally preprocessed STN-LFP signals to preserve physiologically relevant temporal dynamics.

Unlike prior studies that relied on handcrafted spectral features such as power in fixed frequency bands, this work initially focused on temporal feature learning directly from raw neural signals. This decision was motivated by evidence that intersubject variability in Parkinson’s patients confounds the use of a

single spectral profile capable of consistently distinguishing sleep and wake states across individuals [16]. While some patients display comparable spectral sleep-stage patterns, others exhibit subject-specific frequency combinations, suggesting that strictly frequency-based representations may not generalize well across patients [16].

However, after extensive experimentation with raw inputs, spectral information was later incorporated to enhance the model's discriminative capacity. Unlike earlier approaches that reduced the spectrum to a limited set of frequency bins (e.g., eight bands), this framework leverages the full spectrogram representation without binning, preserving finer-grained spectral dynamics relevant to sleep staging.

Given the pronounced class imbalance in PD sleep data—particularly the scarcity of deep sleep (N3)—the framework employs a masking-based self-supervised pretraining strategy, similar in concept to the NeuroNet model. Pretraining initially employed a reconstruction loss, allowing the model to learn intrinsic temporal–spectral dependencies by reconstructing masked segments of the signal without reliance on labels . Subsequently, a contrastive objective based on the Wav2Vec2 framework was tested to evaluate the impact of alternative self-supervised objectives on neural time-series representation learning (see Chapter 2.2 for details) [17].

The pretraining is performed on large, balanced datasets from healthy subjects in public EEG and ECoG sleep repositories [18,19]. This exposure is expected to enable the model to learn the full spectrum of sleep dynamics prior to fine-tuning on PD recordings, enhancing cross-subject generalization and allowing adaptation to the altered neural oscillations characteristic of Parkinson's disease, while preserving accurate discrimination between stages, particularly N2 and N3.

Additionally, preliminary work is underway to apply the framework to sleep-stage forecasting, predicting the next epoch's stage. This task has practical relevance for closed-loop neuromodulation, as anticipating upcoming sleep transitions could allow intervention to encourage or discourage specific stages. Empirical studies have identified early-warning signals in EEG preceding critical sleep transitions, suggesting that predictive modeling of stage changes is feasible [20].

Overall, by combining minimal preprocessing, self-supervised masked learning, and transfer learning from healthy subjects to PD patients, the proposed framework addresses three key limitations of prior approaches: reliance on scarce expert labels, weak cross-subject generalization, and difficulty detecting rare stages (specifically N3). The resulting model holds the potential to provide a robust, patient-independent sleep-stage classifier for integration with closed-loop DBS systems.

# Chapter 2: Methods

## 2.1 Datasets

The primary dataset used in this study is the DBS dataset, which contains intracranial recordings from PD patients. In addition to this dataset, two publicly available datasets were used for self-supervised pretraining. These auxiliary datasets contain recordings from healthy subjects and provide larger, more balanced samples, allowing the model to learn generalizable neural representations before being fine-tuned on PD data. Although they differ from the DBS dataset in both recording modality (EEG and ECoG rather than subthalamic LFPs) and anatomical origin, they represent the best publicly available large-scale neural datasets and capture sleep patterns relatively free from pathological influences.

### 2.1.1 DBS Dataset

STN local field potential (LFP) recordings were obtained from 14 PD patients across Stanford University, the University of Pennsylvania, and the University of Nebraska Medical Center [10]. One patient was excluded due to missing electrode contact annotations, resulting in a final cohort of 13 participants (Fig. 1). For most patients, three nights of recordings were available; however, subjects 2_STAN and 2_UNMC included only two and one night of data, respectively.

Each recording included concurrent LFPs from the STN and polysomnography (PSG) signals following the standard AASM montage (F3–C3, P3–O1, F4–C4, P4–O2, EOGL–A2, EOGR–A1, chin EMG). Sleep stages were manually scored by certified sleep technicians in 30-second epochs based on the PSG recordings rather than the STN LFPs, and labeled as Wake, REM, N1, N2, or N3.

For analysis, only STN channels were used. Unlike conventional bipolar preprocessing, which subtracts adjacent contacts to form differential signals, this study used only the positive contacts. This choice was motivated by the close anatomical spacing of the DBS contacts—where bipolar subtraction may inadvertently cancel genuine neural events, such as transient beta or slow-wave bursts, rather than suppressing noise.

Recordings were resampled to 250 Hz, concatenated across nights, and z-score normalized per patient. Epochs labeled as “unknown” or unlabeled were excluded. Minimal preprocessing was applied to preserve the raw temporal dynamics of the signals. This choice was made to maintain compatibility with

DBS data, which often require minimal external denoising or filtering, and to allow the model to learn directly from the underlying neural activity rather than from feature-engineered representations. Specifically, instead of using manually defined spectral features such as the commonly employed eight frequency bins, the data were initially kept in their raw form to prevent potential information loss and to enable the transformer to discover latent temporal–spectral patterns autonomously. However, spectral processing was later explored as an alternative feature extraction method, where STFT representations were used without subsequent frequency binning. In this setup, the STFT served as a replacement for CNN-based feature extraction rather than as a manual feature-engineering step.

For evaluation, a leave-one-subject-out strategy was adopted: in each iteration, data from one patient served as the test set while the remaining 12 patients were used for training and validation. This ensured that the test set always contained unseen subjects, allowing assessment of cross-patient generalization.

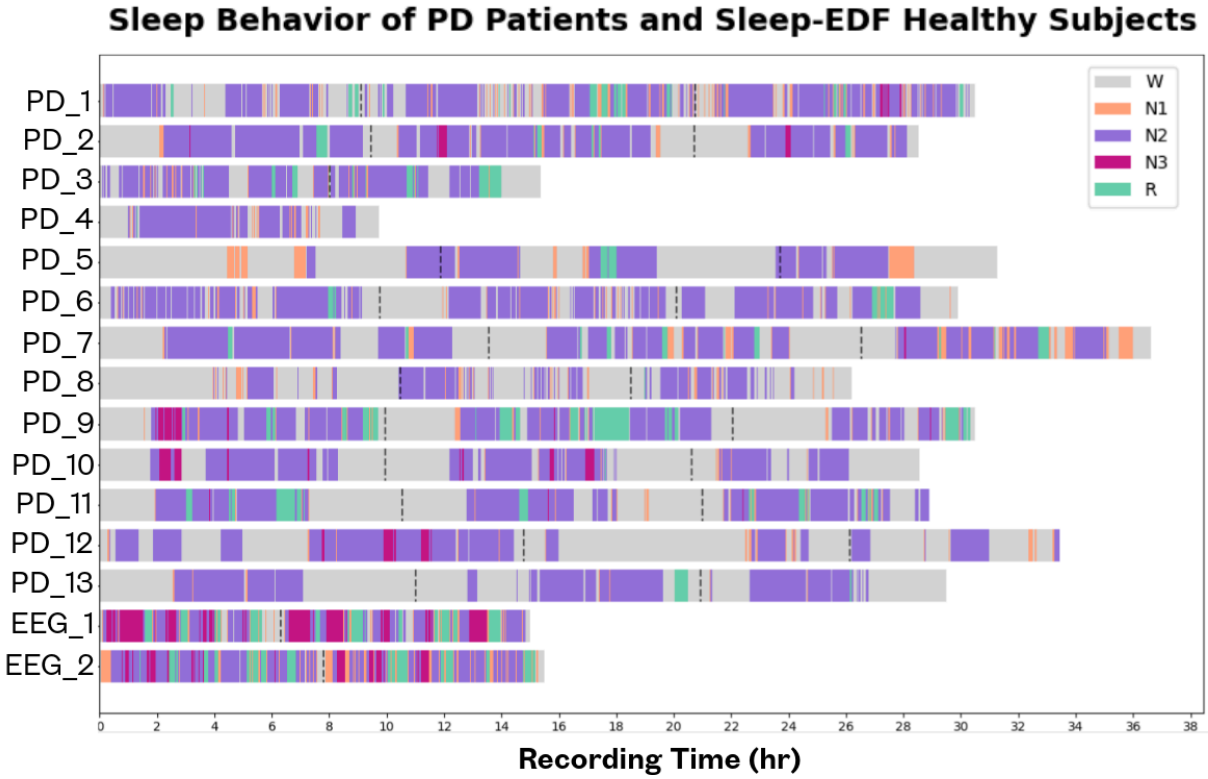


**Fig. 1.** Sleep hypnogram distributions of the Parkinson's disease (PD) patients (top 13) and two Sleep-EDF healthy subjects (bottom 2) across their recorded nights (concatenated). Black dotted lines indicate transitions between consecutive night sessions. Colors represent distinct sleep stages: Wake, NREM1, NREM2, NREM3, and REM. The Sleep-EDF subjects are included as a reference for typical healthy sleep architecture, providing a visual comparison to the PD patients' disrupted sleep patterns.

### 2.1.2 Sleep-EDF Dataset

To introduce a large dataset for pretraining, EEG recordings were sourced from the publicly available Sleep-EDF Expanded Dataset [18], which contains 197 full-night PSG studies. Each study includes EEG, EOG, chin EMG, and event markers, with some subjects also featuring respiration and body temperature channels. Sleep stages were scored manually by trained technicians following the Rechtschaffen and Kales (R&K) standard, yielding hypnograms labeling each 30-second epoch as Wake (W), REM (R), or non-REM stages 1–4, as well as movement (M) and unknown (?) periods.

For this work, only the Sleep Cassette (SC*) subset was used. This subset includes healthy adults aged 25–101 who were not taking sleep-related medication, providing physiologically representative recordings across the full range of natural sleep stages. EEG signals were acquired from two derivations—Fpz–Cz (frontal–central) and Pz–Oz (parietal–occipital)—at a sampling rate of 100 Hz. These channels were specifically chosen to capture cortical brain activity, as opposed to eye movements (EOG) or muscle tone (EMG), which are less relevant for the development of a neural sleep classifier.

Each recording was trimmed to begin at the annotated “lights off” marker and to end 15 minutes after the final labeled sleep stage period, removing pre- and post-sleep noise. The continuous EEG recordings were then segmented into 30-second epochs following AASM standards. The R&K sleep labels were converted to the modern AASM system, merging NREM3 and NREM4 into a single N3 stage for consistency with the DBS dataset.

Signals were z-score normalized per subject and per channel to reduce inter-subject variability while preserving within-subject amplitude structure. For training and testing, a subject-wise greedy split was used—subjects were added to the training set until approximately 85% of the total recording time was reached, and the remainder were assigned to the test set. This ensured that all test data came from previously unseen individuals, allowing subject-independent evaluation.

A key preprocessing step involved channel concatenation. For the classification tasks, the Fpz–Cz and Pz–Oz EEG channels were concatenated sequentially along the temporal axis for each 30-second epoch, forming a composite representation of both frontal and occipital activity. However, for the forecasting experiments, the concatenation was modified to maintain long-term temporal continuity across the entire session—each channel’s full-night sequence was appended end-to-end, rather than alternating between channels every 30 seconds. This approach preserved realistic temporal progression, which is crucial when modeling future-state predictions.

### 2.1.3 ECoG Dataset

To incorporate a second intracranial dataset with higher spatial resolution, ECoG recordings were obtained from the Open iEEG Dataset (CITE). This dataset includes interictal iEEG recordings from 185 subjects collected at the UCLA Mattel Children's Hospital and the Children's Hospital of Michigan. The recordings contain high-quality intracranial signals captured during sleep, offering detailed views of cortical network dynamics.

Only ECoG channels were used for this work. All signals were resampled to 250 Hz to match the DBS recordings, and segmented into 30-second epochs to maintain consistency across datasets. Each subject's data were z-score normalized independently, and a subject-wise greedy split was performed (≈85% train / 15% test, with no overlap). The channel concatenation strategy mirrored that of the Sleep-EDF data.

## 2.2 Model Architecture

The overall model architecture is inspired by the masked autoencoder (MAE) framework, which was originally developed for vision/image classification [21]. Recent studies have adapted this approach to time-series data, such as EEG, including the NeuroNet model. The key idea behind masked autoencoders is to randomly mask portions of the input, feed the remaining unmasked segments into a transformer encoder, and then reconstruct the masked segments using a transformer decoder. In this work, we adopt and extend this approach for neural sleep data. The main architecture and major modifications used are described below, while smaller experimental variations and modifications for the forecasting task are detailed in Section 2.3.

### 2.2.1 Feature Extraction

Transformers can, in principle, process raw neural signals directly, but very long sequences can be computationally expensive and challenging for self-attention to model effectively. To make the input more manageable and highlight relevant temporal patterns, the raw signal is first transformed into structured representations.

One simple approach is to segment the input into "patches" and feed each patch (as in the original MAE paper), along with its positional embedding, into the transformer encoder. A more flexible and automated approach is to use CNNs to extract hierarchical temporal features, which are then passed to the transformer. Alternatively, STFT can be applied to generate spectral features if frequency-domain information is desired.

#### 2.2.1.1 CNN

In this work, I primarily use 1D convolutions (Conv1d in PyTorch) with a single input channel representing the raw neural signal. Each convolutional layer contains learnable kernels (filters) that slide along the temporal dimension of the input. At each position, the layer computes a weighted sum between the kernel and the local segment of the input, producing an activation that reflects the presence of specific temporal patterns. The output of a convolutional layer consists of multiple activation maps stacked along the channel dimension, forming a hierarchical representation of the input [22].

The CNN consists of three convolutional layers, each followed by GELU activations and batch normalization. The output channel dimension for all three layers is 64, allowing the network to capture multiple temporal patterns at each layer. The number of layers and the corresponding kernel and stride sizes were determined through a hyperparameter search to optimize performance on raw neural signals. Different input lengths required adjustments to the kernel and stride sizes; the specific configurations for each scenario are provided in Appendix A.

For the forecasting task, a deconvolutional network (ConvTranspose1d in PyTorch) is used to upsample the transformer decoder features back to the input length for the calculation of the reconstruction loss. The deconvolutional network mirrors the CNN architecture as closely as possible, with three layers and each layer outputting the hidden state size in reverse order of the CNN. This design allows the model to preserve fine-grained temporal structure in the learned representations.

#### 2.2.1.2 STFT

The STFT is a method for analyzing how the frequency content of a signal changes over time. It works by dividing the signal into short, overlapping windows and computing the Fourier transform within each window. This produces a two-dimensional time–frequency representation, where the amplitude of different frequencies can be tracked across time. STFT provides a constant absolute bandwidth, allowing harmonic components to be identified while maintaining consistent resolution across the time–frequency plane [23].

In this work, the STFT was applied using 2-second Hamming windows with 1-second overlap, with each window transformed via FFT to generate the corresponding spectral representation. These parameters were chosen to match previous studies using the PD dataset. However, unlike those studies, I did not group the frequency outputs into eight predefined bins (delta, theta, alpha, low beta, high beta, low gamma, high gamma, and high-frequency oscillations averaged over 30 s). Instead, I retained the full spectral resolution to allow the transformer to capture more nuanced information. Previous literature has

shown that PD patients exhibit subject-specific spectral profiles, so aggregating frequencies into broad bins could obscure subtle, individual-specific patterns that are important for distinguishing between sleep stages. By keeping the full frequency resolution, the model has the opportunity to learn these unique spectral nuances directly.

### 2.2.2 Transformer Fundamentals

The Transformer architecture, first introduced in *Attention Is All You Need*, follows an encoder–decoder framework designed to model long-range dependencies efficiently. The encoder maps an input sequence into a set of continuous latent representations, while the decoder autoregressively generates an output sequence conditioned on these latent states. Each encoder and decoder block contains two main components: a multi-head self-attention and a position-wise feed-forward network. Both are surrounded by residual connections and layer normalization (Fig. 2.A) [24].

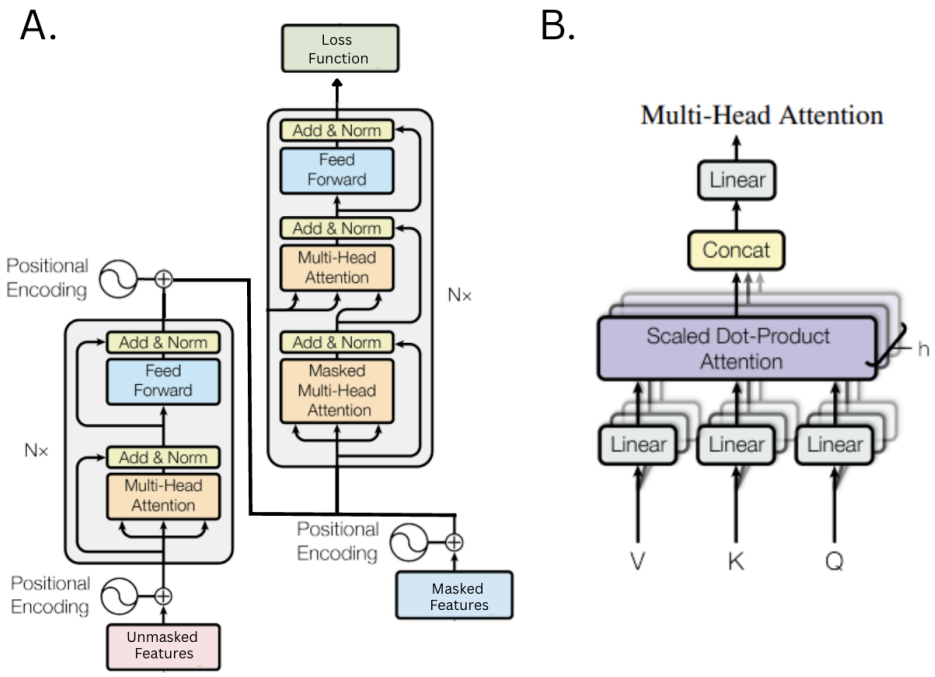


**Fig. 2** Transformer and multi-head attention architecture. Figures are adapted from *"Attention is All You Need"* and modified to illustrate their implementation in the current model **A**. Illustration of the encoder–decoder framework, showing how input sequences are processed through attention, feed-forward, and normalization layers to produce contextualized representations **B.** Visualization of multi-head attention, where multiple self-attention heads operate in parallel to capture diverse temporal–spectral dependencies across different representational subspaces.

**Attention Mechanism**

The attention mechanism determines how much each feature segment in the neural representation should attend to others within the sequence. Given query (Q), key (K), and value (V) projections derived from the CNN feature embeddings, the scaled dot-product attention is defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where $d_k$ denotes the key dimension [24]. In this context, each segment corresponds to a temporally localized neural feature extracted by the CNN (or the STFT). The query represents what that segment seeks to integrate from other parts of the sequence, the key encodes what information it can provide, and the value carries the underlying neural feature representation. The softmax normalization ensures that attention weights across all segments sum to one, enabling the model to dynamically capture dependencies between temporally and functionally related neural patterns.

To capture richer information from multiple representational subspaces, the Transformer employs multi-head attention, where several attention "heads" operate in parallel. Each head applies independent learned projections to transform the inputs, and their outputs are concatenated and linearly projected (Fig. 2.B). This structure allows the model to attend to information from different perspectives simultaneously [24]. In my model, each transformer layer has six attention heads.

**Positional Encodings**

Transformers rely on positional embeddings to provide information about the order of elements in a sequence, since the architecture itself does not inherently encode temporal relationships [24]. In this work, learnable absolute positional embeddings are used, which are initialized randomly and trained jointly with the model parameters. These embeddings help the network keep track of how tokens relate to one another in a temporal sense, allowing it to weigh contextual information appropriately.

Positional information is crucial for the decoder, which reconstructs masked regions of the signal. To accurately predict each masked segment, the decoder must understand how it relates to the surrounding unmasked tokens and which contextual embeddings to rely on. To support this, positional embeddings are added at the encoder input, re-added to the encoder outputs before passing them to the decoder, and also applied to the masked tokens themselves (Fig. 2.2A). Without these additions, the decoder would lack explicit temporal context, impairing reconstruction quality.

### 2.2.3 Masked Autoencoder Adaptation: Reconstruction Objective

Unlike the original MAE for images, which operates on 2D patches, this work adapts the framework to 1D neural time series as mentioned previously. The raw input signal is first processed through a CNN-based feature extractor, which transforms neural waveforms into latent temporal embeddings. In the STFT variant, the CNN is replaced with the STFT, but the remaining pipeline remains identical.

A random masking strategy is applied along the temporal axis, where approximately 65% of the tokens are replaced by trainable mask embeddings. These masked tokens are passed to the four layer transformer encoder with attention disabled, explicitly preventing the model from attending to them. This ensures that the encoder learns contextual representations solely from the visible (unmasked) tokens, which are provided along with positional embeddings to preserve temporal relationships (Fig. 3).

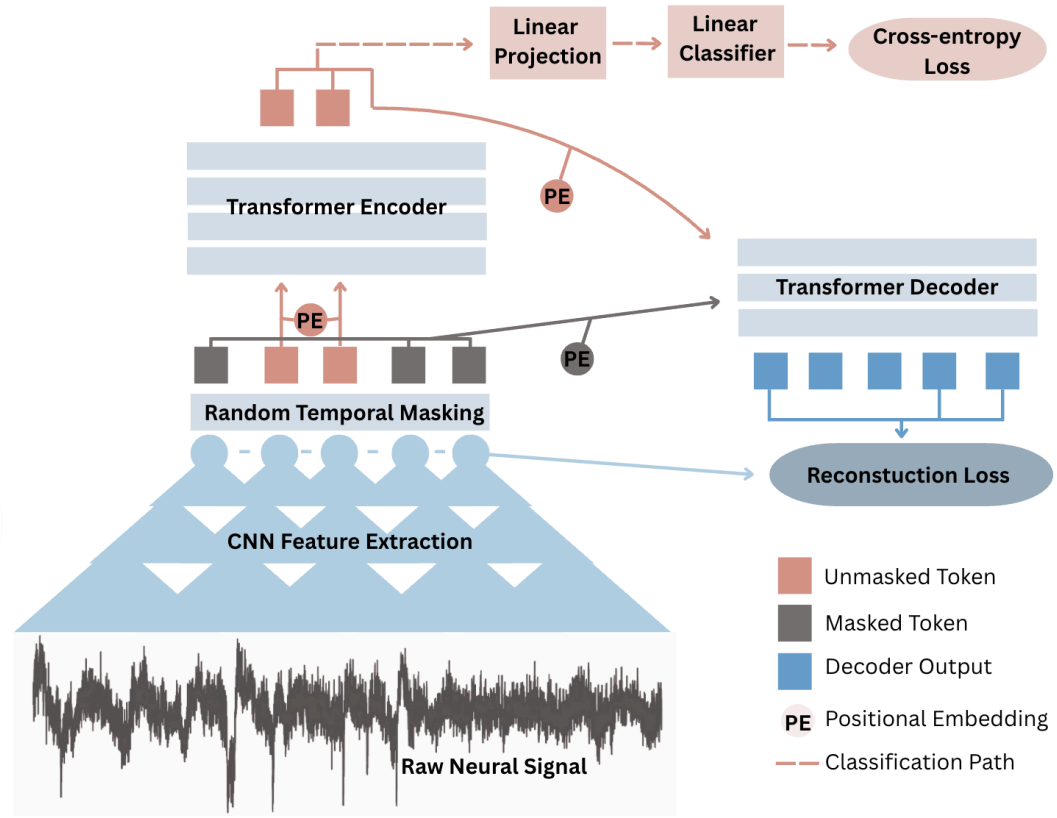


**Fig. 3.** Overview of the masked autoencoder architecture. The solid arrows depict the pretraining process, where the model reconstructs masked portions of the input signal. The dashed arrows represent the fine-tuning framework for classification, where the pretrained encoder is used to predict sleep stage labels. Important note for the finetuning: none of the tokens are masked in the usual model design.

During pretraining, the decoder receives both the encoder outputs and the masked tokens, each augmented with positional embeddings. The decoder is tasked with reconstructing the full input sequence. Importantly, the reconstruction loss is computed only over the masked tokens, encouraging the model to focus on predicting missing information. The loss is defined as the mean absolute error (MAE) between the decoder's output $\hat{x}$ and the ground-truth CNN feature embeddings $x$ for the masked indices:

$$\mathcal{L}_{\text{MAE}} = \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} |\hat{x}_i - x_i|$$

MAE is preferred over MSE because it is less sensitive to outliers and inter-patient variability, both of which are prevalent in Parkinson's datasets (as reflected in the sleep-stage distribution in Figure 1).

For fine-tuning/classification, only the encoder outputs are used. The input tokens are not masked during fine-tuning, since reconstruction is no longer required. The encoder representations are passed through a linear projection layer, followed by a linear classifier that outputs logits corresponding to the five sleep stage labels. To address class imbalance, the weighted cross-entropy loss is applied during supervised training:

$$\mathcal{L}_{\text{CE}} = -\sum_{i=1}^{C} w_i \cdot y_i \cdot \log(\hat{y}_i)$$

where $C$ is the number of classes, $y_i$ is the ground-truth label for class $i$, $\hat{y}_i$ is the predicted probability, and $w_i$ is the inverse frequency weight for class $i$.

#### 2.2.4 Wav2Vec 2.0: Contrastive Objectives

Self-supervised learning for neural signals can be guided by multiple objectives, with the two primary strategies being reconstruction and contrastive learning. While the main model used in this project employs a reconstruction objective, we also explored a contrastive pretraining strategy by adapting Wav2Vec 2.0, a self-supervised speech model, to Parkinson's disease neural time series data [17].

The rationale for this choice is that speech and neural time series share key characteristics: both are continuous, structured waveforms where local context conveys high-level state information. In speech, context encodes phoneme transitions; in neural data, it captures oscillatory dynamics and sleep-state transitions.

The Wav2Vec 2.0 architecture consists of a feature encoder—a stack of temporal convolutional layers with GELU activations and layer normalization—followed by a Transformer context network that processes the resulting latent representations (Fig. 4). To maintain comparability with the reconstruction-based model, the CNN feature extractor was configured identically (three layers with the same kernel sizes and strides), while the Transformer encoder retained its original Wav2Vec 2.0 configuration of six layers, each with six attention heads. This also keeps things in a similar scale of the reconstruction model (four encoder layers + three decoder layers) while avoiding the need for a decoder.

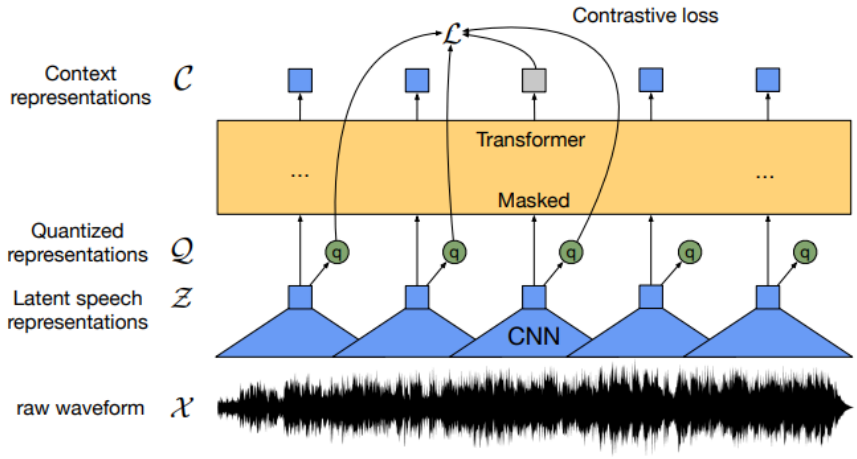


**Fig. 4.** Illustration of the Wav2Vec2 framework from wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations paper. The raw input signal is first processed by a convolutional feature encoder to extract latent representations, which are then quantized and passed into a Transformer encoder. The model is trained through a contrastive learning objective, where the true latent representations are distinguished from distractors to build contextualized representations.

During pretraining, a portion of the latent features is masked, and the model is trained to identify the correct quantized latent representation among distractors using a contrastive loss:

$$\mathscr{L}_{\mathrm{c}} = -\log \frac{\exp(\mathrm{sim}(c_t, q_t)/\kappa)}{\sum_{q' \in Q_t} \exp(\mathrm{sim}(c_t, q')/\kappa)}$$

where $c_t$ is the contextualized embedding at time step $t$, $q_t$ is the true quantized latent, $Q_t$ is the set of distractors, $sim(a,\ b)$ is the cosine similarity:

$$\mathrm{sim}(a, b) = \frac{a \cdot b}{\|a\|\|b\|}$$

and $\kappa$ is a temperature parameter that controls the sharpness of the similarity distribution.

To encourage diverse usage of latent representations, Wav2Vec 2.0 also incorporates a diversity loss:

$$\mathscr{L}_{\mathrm{d}} = \frac{1}{GV} \sum_{g=1}^{G} \sum_{v=1}^{V} \bar{p}_{g,v} \log \bar{p}_{g,v}$$

where $G$ is the number of codebooks, $V$ is the number of entries per codebook, and $\bar{p}_{g,v}$ is the average selection probability of codebook entry $v$ in codebook $g$. Each codebook serves as a learned dictionary of discrete latent vectors ("codewords"), and the diversity term encourages more uniform use across all codewords, preventing representational collapse.

The total pretraining objective is a weighted combination of these two terms:

$$\mathscr{L}_{\mathrm{t}} = \mathscr{L}_{\mathrm{c}} + \alpha \cdot \mathscr{L}_{\mathrm{d}}$$

where α=0.1, following the standard Wav2Vec 2.0 configuration. Because the contribution of diversity loss is significantly smaller compared to the contrastive loss, this model will be referred to as the contrastive objective in this thesis, rather than the combined contrastive + diversity objective.

The fine tuning of the model to sleep stage categorization task is kept identical to the reconstruction model for fair comparison purposes.

## 2.3 Experimental Modifications to the Model

Building upon the masked autoencoder framework described in Section 2.2.3 (Fig. 3), several targeted modifications are implemented to improve model robustness, address severe sleep-stage imbalance, and explore the model's capacity for temporal forecasting.

### 2.3.1 Strategies for Data Imbalance

Given the pronounced imbalance of sleep stages, particularly the underrepresentation of N3 and REM, the following strategies were applied to reduce bias (Fig. 1). Two primary approaches to handling dataset imbalance were explored:

**Unmasked encoder - Inverse Frequency Weight (original design)**

Input sequences are fed to the pretrained encoder without any temporal masking during fine-tuning, as described in Section 2.2.3. Class imbalance is addressed by incorporating inverse weight frequencies of the sleep stages into the cross-entropy loss calculation. These weights are computed based on the training dataset's class distribution.

**Masked encoder - Oversampling (alternative strategy)**

In this approach, underrepresented classes are oversampled until their quantities match the most frequent sleep stage in the training set. To mitigate repetition of the same input information, tokens are masked after feature extraction, similar to the pretraining procedure. This introduces a form of synthetic data augmentation, as the random masking ensures the encoder sees slightly different token sequences on each pass. Masking and oversampling are applied only to the training set; validation and test sets remain unaltered.

Additionally, a **penalized loss matrix** was incorporated to target the most prominent misclassifications. The highest penalty is applied when REM is misclassified as Wake (W), followed by N1 misclassified as W and N3 misclassified as N2 (Fig. 5). Reverse misclassifications are penalized to a lesser extent (see illustration below). Penalty magnitudes were determined based on the confusion matrix results from overall model performance (Chapter 3.2).

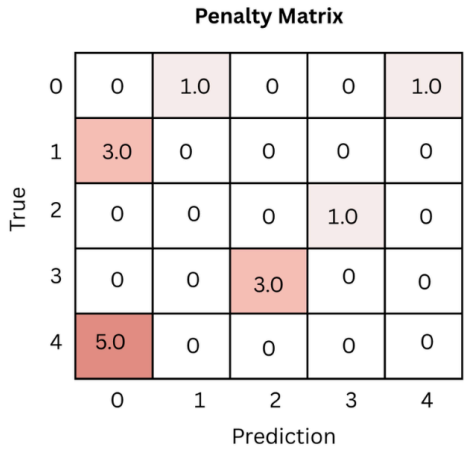


**Fig. 5.** Visualization of the penalty matrix. Rows correspond to true labels; columns to predicted labels (0=W, 1=N1, 2=N2, 3=N3, 4=R). Values indicate penalty weights added to cross-entropy loss for specific misclassifications. For example: misclassifying REM sleep as Wake (true=4, pred=0) incurs the highest penalty (5.0), reflecting the clinical severity of this error.

The penalized loss operates as follows: during loss calculation, the row corresponding to the true label is multiplied by the penalty weights for the predicted probabilities of that epoch. The resulting weighted

term is added to the cross-entropy loss, producing a **composite loss** that explicitly discourages high-impact mispredictions.

**2.3.2 Major Experimental Model Variants**:

1. **Feature extractor replacement:** The CNN was replaced with an STFT-based feature extractor to assess how spectral features compare to CNN-derived temporal features for classification (see Section 2.2 on the details of each method). As of now, this change has only been applied during fine-tuning and not for the reconstruction task.
2. **Contrastive pretraining objective:** Wav2Vec 2.0 was employed to pretrain the model using a contrastive objective instead of the reconstruction objective to see how different self-supervision methods compare. Architectural details are provided in Section 2.2.4.
3. **Pretraining dataset variations and control model:** Different datasets were used for pretraining, along with a control (or base) model that received no pretraining (the weights for the model were randomly initialized). For reconstruction pretraining, the Parkinson's disease (PD), ECoG, and Sleep-EDF datasets were used (dataset details in Section 2.1).

Notably, when pretraining is performed on the Sleep-EDF dataset, the PD dataset used for fine-tuning is downsampled to match Sleep-EDF's 100 Hz recording, ensuring compatibility of model weights. Furthermore, the CNN feature extractor weights are re-initialized during fine-tuning, since the Sleep-EDF and ECoG datasets require different feature extraction parameters compared to the PD dataset.

The number of pretraining epochs is also dataset-dependent: Sleep-EDF and ECoG, being substantially larger than the PD dataset, are pretrained for 30 epochs, as the loss had stabilized by this point and additional training was unnecessary. In contrast, the smaller PD dataset is pretrained for 100 epochs. Fine-tuning is consistently performed for 100 epochs to ensure sufficient learning, particularly given the limited size of the PD dataset after the train-test split.

**2.3.3 Forecasting Adaptations (Preliminary)**

In addition to classification, the architecture is extended to support temporal forecasting, enabling the model to predict future neural activity and subsequent sleep stages. For this task, the masking mechanism is redesigned from random to deterministic temporal masking: the final nnn seconds of each input sequence (currently 5 seconds) are consistently masked to simulate an unobserved future segment.

To provide sufficient context for forecasting, the input sequence length is increased to 60 seconds (compared to the original 30-second segments), as shorter windows may be insufficient for predicting the

upcoming 5 seconds. This 60-second window is constructed using a sliding window approach. To forecast the next 5 seconds of a given epoch, the model receives the 60 seconds of data immediately preceding it. This is implemented by concatenating the current 30-second epoch with the previous 30-second epoch, providing sufficient context to predict the subsequent masked 5-second segment.

Unlike the pretraining phase for the reconstruction objective, where masking is applied to CNN feature embeddings, the forecasting masking is applied directly to the raw waveform input. This ensures that the model learns genuine temporal extrapolation rather than relying on the feature extractor's prior encodings to fill in missing information.

Because the model now reconstructs raw waveform segments, a deconvolutional decoder is appended to the output of the transformer decoder (see Section 2.2 for architecture details). The deconvolutional decoder mirrors the CNN architecture as closely as possible, minimizing information loss. The reconstruction objective remains MAE, computed only over the masked future segments, training the model to forecast the unseen continuation of the neural signal.

After forecasting pretraining, the encoder is fine-tuned for next-epoch sleep stage prediction. In this phase, the model receives a 60-second input segment and is trained to predict the label of the subsequent epoch. No masking is applied during fine-tuning, and the upcoming 5 seconds of the sequence are not input; only the label of the next epoch is predicted.

## 2.4 Evaluation

To assess the performance of the proposed masked autoencoder and its fine-tuned variants, evaluation is conducted on held-out test data using multiple complementary metrics. For all classification experiments, including those with and without pretraining, the dataset is split into training, validation, and test sets (details on test set creation are provided in Section 2.1). The validation set comprises 20% of the training data, yielding an 80–20 train-validation split. A stratified, seeded random shuffle is applied to preserve class ratios across splits, ensuring that each sleep stage remains proportionally represented. The validation set is primarily used for early stopping, guiding model selection based on the weighted F1 score and preventing any leakage of information from the withheld test set into hyperparameter tuning.

**Weighted F1 Score**

The primary metric for evaluating model performance in this work is the weighted F1 score. The F1 score—the harmonic mean of precision and recall—was selected because it provides a balanced

assessment for each individual class, which is essential for this multi-class problem. Precision (the proportion of correct predictions for a class) is critical because a high rate of false positives could lead to inappropriate and potentially harmful stimulation. Recall (the proportion of actual class members correctly identified) is equally critical because a high rate of false negatives means the system misses opportunities to deliver therapy when it is most needed.

This balanced perspective is particularly important given the specific sleep fragmentation patterns in PD. Analysis of the hypnograms (Fig. 1) suggests that the overabundance of W is largely due to a failure to transition into stable N1 sleep, while the scarcity of N3 stems from a failure of N2 to transition into deeper sleep. Therefore, the model's core challenge is to navigate these ambiguous transitional states.

The clinical impact of misclassifications further justifies this focus. The most harmful errors involve stimulating at the wrong time: miscategorizing R or N1 as W could trigger stimulation that abruptly terminates a critical sleep stage, while miscategorizing N3 as N2 represents a missed opportunity for neuroprotective stimulation during slow-wave sleep. The F1 score, by balancing precision and recall, directly penalizes these high-impact false positives and false negatives.

Given the drastic class imbalance in the dataset, the weighted F1 score is used. It is calculated across all classes as:

$$\mathrm{P}_i = \frac{TP_i}{TP_i + FP_i} \qquad \mathrm{R}_i = \frac{TP_i}{TP_i + FN_i} \qquad F1_i = 2 \cdot \frac{\mathrm{P}_i \cdot \mathrm{R}_i}{\mathrm{P}_i + \mathrm{R}_i}$$

where $TP_i$, $FPi_i$, and $FN_i$ are the true positive, false positive, and false negative counts for class $i$.

The weighted F1 score across all classes is then calculated as:

$$F1_{\text{weighted}} = \frac{\sum_{i=1}^{C} N_i \cdot F1_i}{N_{\text{total}}}$$

where $N_i$ is the number of true instances in class $i$, and $N_{total}$ is the total number of samples in the test set. Weighted F1 is preferred over macro F1 for this application because some sleep stages, such as N3, are extremely rare in certain patients. Weighted F1 ensures that overall performance reflects class prevalence, providing a more accurate measure of model performance across all sleep stages.

The reported metrics in the results represent the median of all patients' test weighted F1 scores, with error bars indicating the minimum and maximum F1 values observed across patients.

**Statistical Significance**

To assess whether differences between models are statistically significant, the Wilcoxon signed-rank test is applied. This non-parametric test compares paired samples (e.g., patient-level F1 scores from two different models) and evaluates whether the observed differences are symmetric around zero. It is particularly suitable for small sample sizes and non-normal distributions, which applies for the PD sleep data.

**Confusion Matrices:**

Confusion matrices are computed to visualize the distribution of correct and incorrect predictions across sleep stages. For each patient, a normalized confusion matrix is created by dividing the count of predictions for each class by the total number of true instances in that class. Normalization allows comparison across patients with differing numbers of epochs per class.

To aggregate results across patients, the confusion matrices are medianized, producing a representative matrix for the model that is robust to outliers or highly skewed patient distributions. Further normalization ensures that the resulting matrix represents proportions rather than raw counts.

Some analyses report differences between confusion matrices of different models, which is done by simply subtracting the entries of one matrix from the corresponding entries of another. This highlights which classes see improved or worsened prediction rates under different models.

# Chapter 3: Experiment Results

### 3.1 Different Pretraining Models

Pretraining was performed on different datasets (Sleep-EDF and ECoG) using a reconstruction objective, in which the model learns to reconstruct masked regions of the input signal (see Section 2.2 for details). The motivation for exploring different pretraining datasets stems from the imbalanced distribution of sleep stages in the PD dataset. For instance, several patients exhibit very little or no N3 sleep (Fig. 1). To mitigate this limitation and expose the model to more balanced and healthy representations of sleep, the Sleep-EDF and ECoG datasets were included for pretraining. In addition, I trained a model on reconstructing the masked portions of the PD signals themselves, to assess whether pretraining on the fine-tuning dataset would yield any improvement compared to pretraining on external datasets. Finally, a randomly initialized model (without pretraining) was used as a baseline.

The results indicate no statistically significant difference between the pre-trained models and the randomly initialized model in terms of test weighted F1 accuracy. The PD reconstruction, ECoG reconstruction, Sleep-EDF reconstruction, and Random models achieved test F1 scores of approximately 62%, 59%, 60%, and 61%, respectively (Fig. 6.A).

However, one noteworthy observation is that the range of patient-level weighted F1 scores is slightly narrower for the Sleep-EDF model, meaning that its minimum and maximum scores are closer together. This trend is particularly evident in the validation F1 results (Fig. 6.B). A smaller range is an important property, as the main goal is to achieve consistent generalization across diverse patients. Even if a model does not achieve the highest median F1 score, its consistency across subjects can be of greater practical value, provided that its overall performance remains close to the top-performing model. Based on this, the Sleep-EDF pretraining model, having the smallest error bar, was selected for subsequent experiments alongside the ‘base’ random model.

Another notable finding is that the confusion matrices indicate a slightly higher prediction rate for the N3 stage in pretrained models, particularly the PD pretrained model. This stage is clinically significant due to its role in brain toxin clearance. Beyond this improvement, the confusion matrices reveal no substantial differences in prediction patterns between the models.

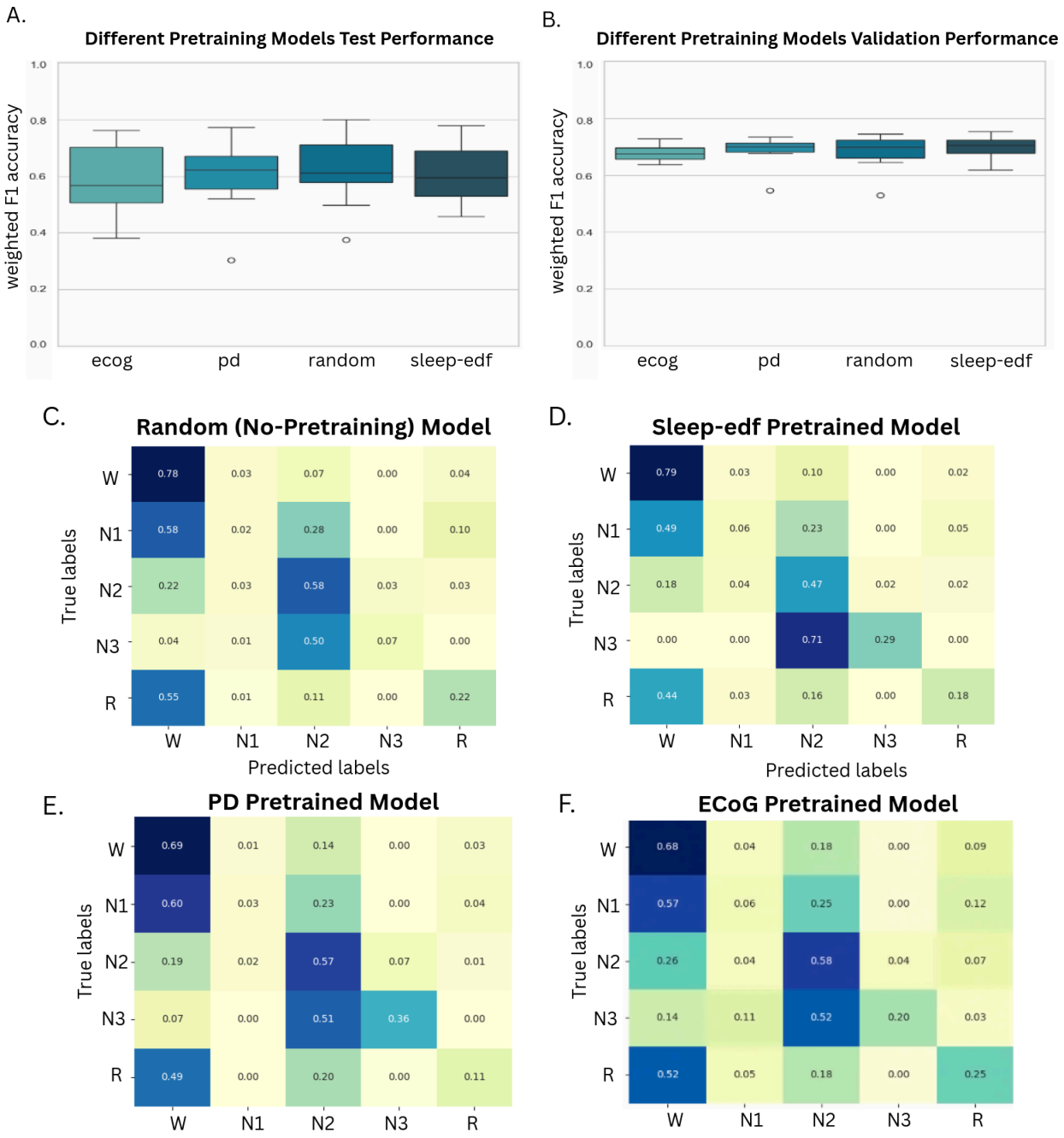


**Fig. 6.** Comparison of models pre-trained on different datasets based on their test and validation F1 scores. The F1 score, as described in Chapter 2.4, represents the median across all patients' weighted F1 scores. The y-axis indicates the weighted F1 score, while the x-axis shows different model types. Statistical significance is compared to the 'random' model, using Wilcoxon signed ranking. **A**. Test F1 scores for the withheld patient, reflecting the models' ability to generalize to unseen subjects **B.** Validation F1 scores, calculated on 20% of the remaining 12 patients (the other 80% used for training), representing within-subject performance **C.** Confusion matrix of randomly initialized model's true and predicted labels **D.** Confusion matrix of the Sleep-EDF reconstruction pretrained model's true and predicted labels **E.** Confusion matrix of PD reconstruction pretrained model's true and predicted labels **F.** Confusion matrix of ECoG reconstruction pretrained model's true and predicted labels.

### 3.2 Data Imbalance Strategies

One of the major challenges with the PD dataset, as discussed previously, is its severe class imbalance. To address this, I initially applied the inverse frequency weighting strategy. In this approach, classes that occur less frequently are assigned higher weights in the cross-entropy loss function, so the model is penalized more strongly when it misclassifies rare classes.

Subsequently, I implemented an alternative imbalance strategy. In this method, the underrepresented classes were oversampled to match the frequency of the most common class (typically the N2 stage). The model also retained the masking feature of the architecture, which serves as a form of synthetic data augmentation, allowing the network to view slightly different masked versions of the same input and improving its robustness to variation. Further details are provided in Chapter 2.3.

From the confusion matrix results depicted in Figure 7.C-F, there appears to be no significant difference between the two imbalance strategies, aside from a slight increase in the N3 stage for the Sleep-EDF inverse frequency weighted model. However, since this trend is not replicated in the random inverse frequency weighted model, it is unclear whether the inverse frequency weight configuration truly benefits N3 classification. The test F1 accuracies reinforce this observation, showing no significant difference in performance across all methods (Fig. 7.A). The only noticeable difference appears in the validation F1 accuracies, where the oversampling strategies tend to show slightly smaller error bars within each pretraining condition compared to the inverse frequency weight strategy (Fig. 7.B). However, since this effect does not extend to the test results, I chose to continue the subsequent experiments using the inverse frequency weight strategy, as it substantially reduces computational cost. By avoiding oversampling, the dataset remains smaller, enabling faster fine-tuning and training while achieving nearly identical performance.

Another consistent observation across all confusion matrices is the tendency of the models to misclassify R and N1 stages as W. The reverse (W being predicted as R or N1) also occurs, though less frequently. Additionally, N3 is often misclassified as N2 (Fig. 7.C–F). These misclassification patterns align with trends reported in previous literature, as discussed in Chapter 1 [7, 8].

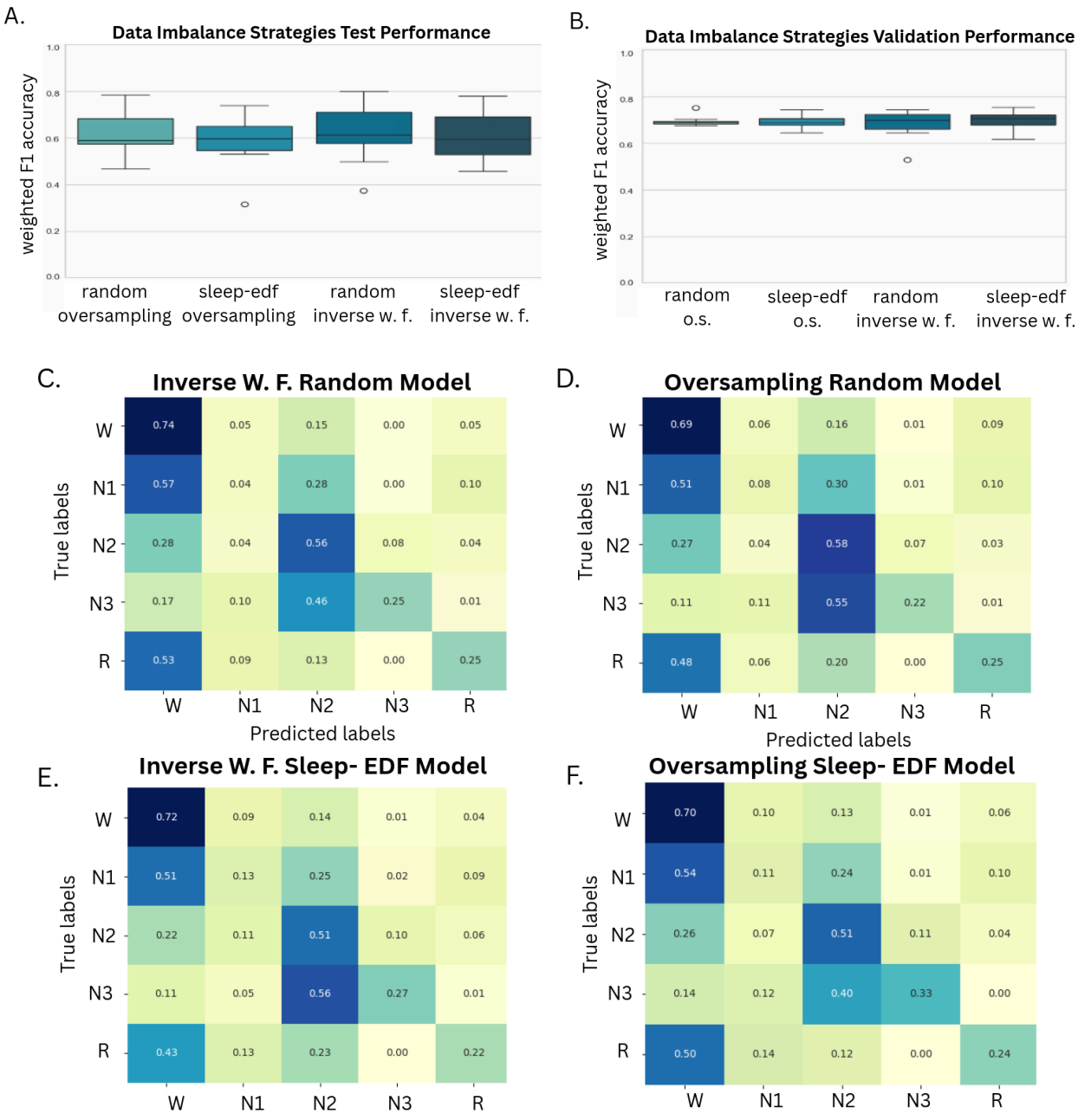


**Fig. 7.** Comparison of randomly initialized model and Sleep-EDF pretrained model using different data imbalance strategies. The "inverse w. f." represents the inverse frequency weighted cross-entropy strategy, while the "oversampling" or "o.s." represents the oversampled and masked input strategy. Statistical significance of the 'oversampling' version is compared to the 'inverse w. f' version for each model **A.** Shows the F1 accuracy scores for the test data **B.** Shows model performance for the validation data **C.** Shows the confusion matrix for the inverse frequency weighted random model (no pretraining) **D.** Shows the confusion matrix for the oversampled and masked input strategy random model **E.** Shows the confusion matrix for the Sleep-EDF reconstruction pre-trained model with the inverse frequency weight strategy **F.** Shows the confusion matrix for the Sleep-EDF reconstruction pre-trained model with the oversampling strategy.

### 3.3 Penalty Matrix for Mispredicted Classes

Given the specific misclassification patterns observed in the previous experiment, I introduced a penalty matrix into the loss calculation (see Fig. 5). The goal of this addition was to introduce a degree of supervision that would discourage the model from making the most frequent mispredictions, such as confusing W with R or N1, and N3 with N2. So if the model mispredicts N3 as W, for example, an additional penalty loss will be added on top of the cross-entropy loss. The reverse (W being miscategorized as N3) is also punished, but at a lower degree, as the matrix has smaller penalties for those situations (see Chapter 2.3 for more details).

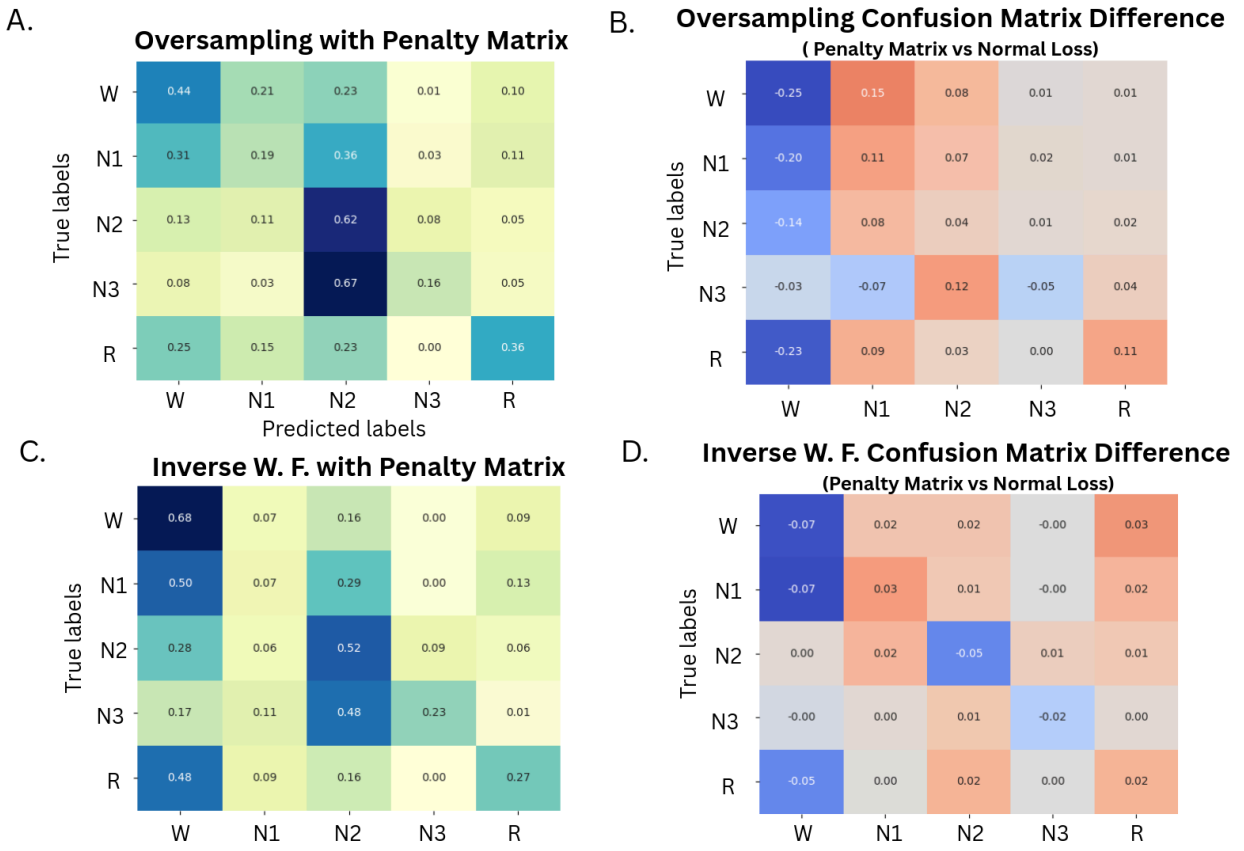


**Fig. 8.** Confusion matrices of the two imbalance strategies (oversampling and inverse frequency weight) with the penalty matrix added to the loss to address specific sleep stage misclassifications. The results are from the randomly initialized model. In panels B and D, red–orange hues indicate an increase in predictions for the sleep stages given the true label, while blue hues indicate a decrease. Grey represents no change **A.** Confusion matrix for the oversampling strategy with the penalty matrix **B.** Difference in confusion matrices for the oversampling strategy with and without the penalty matrix **C.** Confusion matrix for the inverse frequency weight strategy with the penalty matrix **D.** Difference in confusion matrices for the inverse frequency weight strategy with and without the penalty matrix.

The results in Figure 8 show that the penalty matrix reduced the W-stage mispredictions for the random model, especially for the oversampling strategy. The inverse frequency weight strategy, however, did not show substantial improvements—the most notable changes were a small decrease (approximately 7%) in the cases where N1 was misclassified as W, and a similar decrease for W being misclassified as N1.

Although the penalty matrix did lead to an increase in N1 predictions, these changes did not always occur in the intended direction. Specifically, the most pronounced increase in N1 predictions occurred when the true label was W, which is actually a penalized case (albeit lightly). This suggests that while the penalty matrix encouraged the model to predict N1 more frequently, it did not meaningfully improve the separation between N1 and W stages.

Additionally, the results show an increase in N2 and R-stage predictions, even in cases where those misclassifications were explicitly penalized (such as N3 being classified as N2). This indicates that, in practice, the penalty matrix strategy primarily discouraged the model from predicting the W stage, rather than correcting the specific misclassification relationships it was designed to address.

**3.4 STFT vs CNN Feature Extraction**

The next experiment revisited a method commonly used in previous literature, STFT. When designing the model, I initially chose to use the raw EEG signal instead of spectral features derived from STFT, since PD can produce sleep stages with atypical or mixed spectral patterns (as discussed in Chapter 1). Using raw signals was intended to help the model generalize better to new patients whose spectral compositions might differ from those seen during training [16]. Nevertheless, I decided to evaluate an STFT-based feature extractor as an alternative for the CNN feature extractor to directly compare their effects on model performance.

Although the test F1 accuracies did not differ significantly between the two feature extractors, the STFT-based model achieved a slightly higher median accuracy (68%) compared to the CNN-based model (61%) (Fig. 9.A). Furthermore, the STFT model exhibited a noticeably smaller error bar for validation F1 accuracy (Fig. 9.B), suggesting more stable performance across validation folds. This indicates that spectral feature extraction provided a modest improvement in within-patient performance, without reducing generalization ability—contrary to what I expected based on previous literature [16].

The most striking difference appeared in the confusion matrix patterns (Fig. 9.C–D). The STFT model produced a more balanced distribution of predicted sleep stages, achieving the kind of improvement expected from the penalty matrix (which it failed to produce as can be seen in the previous experiment). This effect was particularly evident for the R (4) stage predictions. However, while predictions for most

of the stages (N1, N2, N3, and R) improved, the accuracy for W (0) stage predictions decreased by 16%. This trade-off explains why the overall weighted F1 accuracy did not show a statistically significant change—the model improved in classifying less frequent stages but at the cost of accuracy in one of the most common stages.

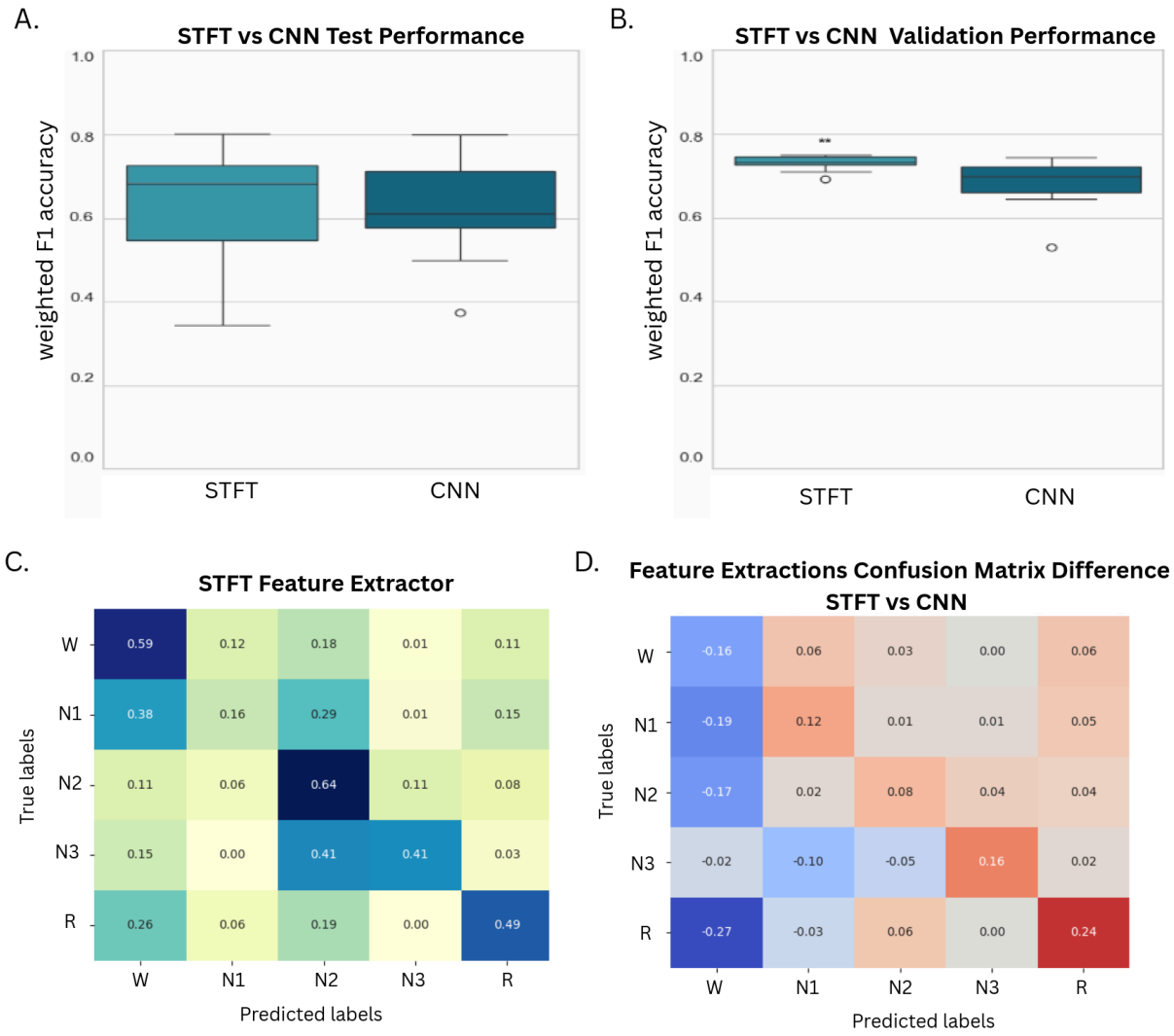


**Fig. 9.** Comparison of randomly initialized models with different feature extraction methods: STFT vs CNN **A.** Weighted F1 Accuracy median across patients for the test dataset **B.** Weighted F1 Accuracy median across patients for the validation dataset **C.** Confusion matrix for the predicted and true sleep labels with the STFT extractor model **D.** Difference in confusion matrices between the STFT and CNN feature extractor models.

### 3.5 Reconstruction vs Contrastive Objective

To investigate how different self-supervised learning objectives influence downstream performance, I compared two pretraining approaches: contrastive learning and reconstruction learning. In self-supervised learning, contrastive loss encourages the model to learn discriminative representations by bringing similar

samples (positive pairs) closer in the latent space while pushing dissimilar ones (negative pairs) apart. In contrast, reconstruction loss, which I have been using for the rest of the experiments, focuses on information preservation by training the model to accurately reconstruct masked or corrupted portions of the input signal. More details on these objectives are provided in Chapter 2.2.

For the contrastive setup, I used the Wav2Vec 2.0 model, originally designed for transforming raw audio waveforms into text representations. For the reconstruction setup, I selected the PD reconstruction-pretrained model, as it was pretrained on the same dataset as Wav2Vec 2.0, ensuring a fair comparison between the two objectives.

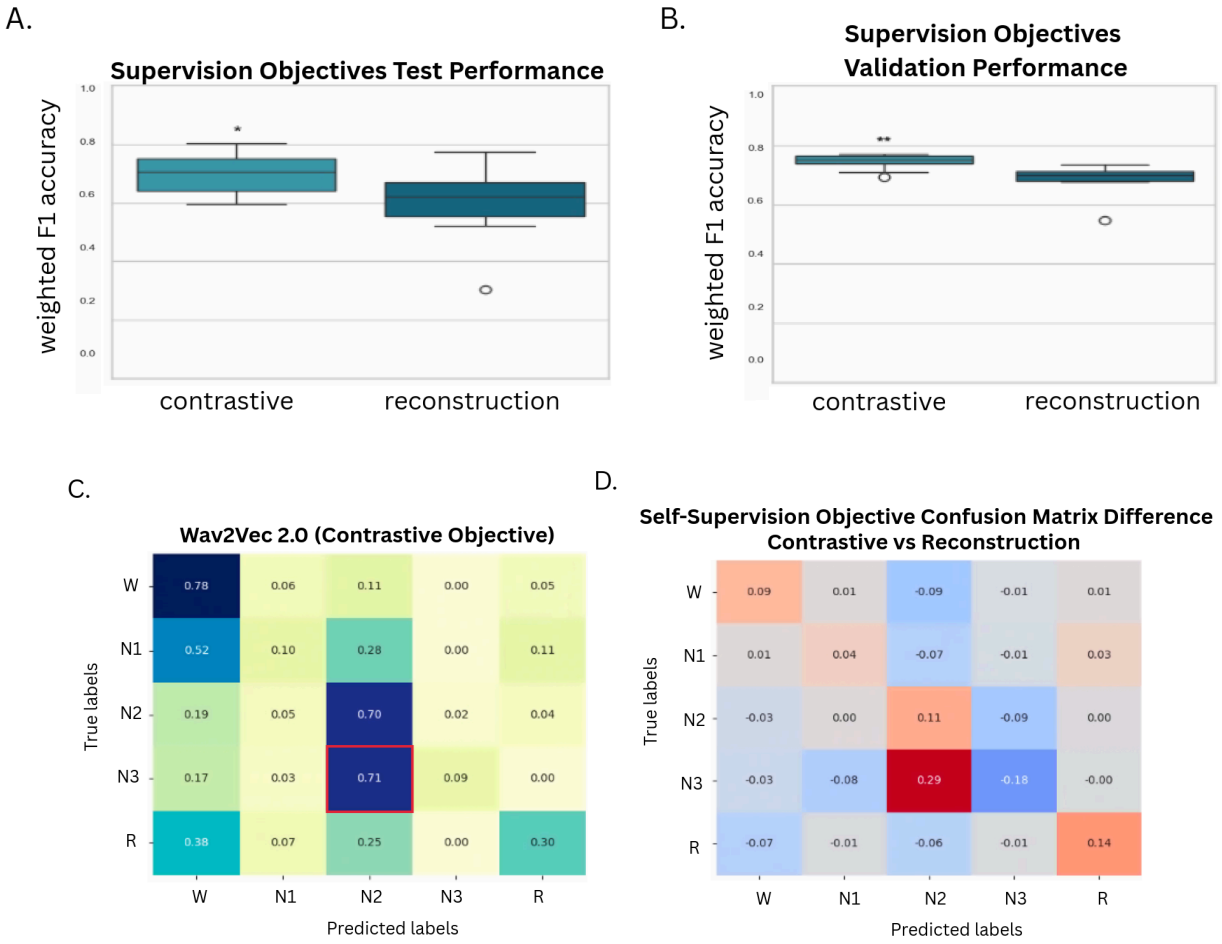


**Fig. 10.** Comparison of the Wav2Vec 2.0 model and the PD reconstruction-pretrained model for evaluating contrastive and reconstruction objectives **A.** Weighted F1 Accuracy median across patients for the test dataset **B.** Weighted F1 Accuracy median across patients for the validation dataset **C.** Confusion matrix for the predicted and true sleep labels with the Wav2Vec 2.0 (contrastive objective) **D.** Difference in confusion matrices between the contrastive and reconstruction objective models.

The results indicate that the contrastive learning approach (Wav2Vec 2.0) substantially outperforms the reconstruction-based model, achieving a test weighted F1 score of 70%, with notably smaller error bars (Fig. 10.A–B). When examining class-specific performance, the contrastive model also performs better across most classes, particularly for R stage classification (Fig. 10.D). However, one limitation of the contrastive approach is its reduced accuracy in predicting the N3 stage, which is the most clinically significant stage due to its role in brain toxin clearance. Interestingly, the reconstruction-pretrained model performs better in this category. This limitation suggests that the contrastive framework may not fully capture slow-wave sleep dynamics from raw neural signals. Future improvements could involve adapting the Wav2Vec 2.0 model to operate on STFT representations instead of raw signals, allowing the model to better capture frequency-domain information critical for accurate N3 detection.

### 3.6 Preliminary Forecasting Results

This experiment is still in its preliminary stages, as there are several variables yet to be explored for the forecasting task—such as input sequence length, the duration of the masked region, the number of tokens ahead to predict, the feature extraction method, and the choice of loss objective. So far, as described in Chapter 2.3.3, the reconstruction pretrained models are provided with 60 seconds of input data and trained to forecast the subsequent 5 seconds. For consistency, both the non-pretrained (randomly initialized) model and the fine-tuned versions use the same 60-second input length. During fine-tuning, the models are currently set to predict only the immediate next stage (+1). The feature extractor remains CNN-based, and the data imbalance strategy used is the unmasked inverse-frequency weighting, with no penalty matrix applied.

The current results indicate that forecasting future stages does not reduce the test F1 accuracy compared to the “current” token prediction task presented in Section 3.1. In fact, there is a slight improvement in the pretrained models: the PD reconstruction-pretrained model increased from 62% to 64%, and the Sleep-EDF reconstruction-pretrained model from 60% to 64% (Fig. 6.A, Fig. 11.A). However, similar to the earlier experiments, the improvements are not statistically significant, apart from a consistent reduction in the validation accuracy error bars—suggesting more stable training.

The more notable findings appear in the confusion matrix patterns (Fig. 11.C–F). The randomly initialized model, despite using the inverse-frequency weighting strategy, appears to overpredict the most frequent classes, resulting in a biased output distribution (Fig. 11.C). In contrast, the pretrained models display more balanced prediction behavior. The PD reconstruction-pretrained model demonstrates even coverage across stages, though with a relatively high number of misclassifications across categories (Fig. 11.D). The Sleep-EDF reconstruction-pretrained model, however, shows a significant improvement in N3 and R

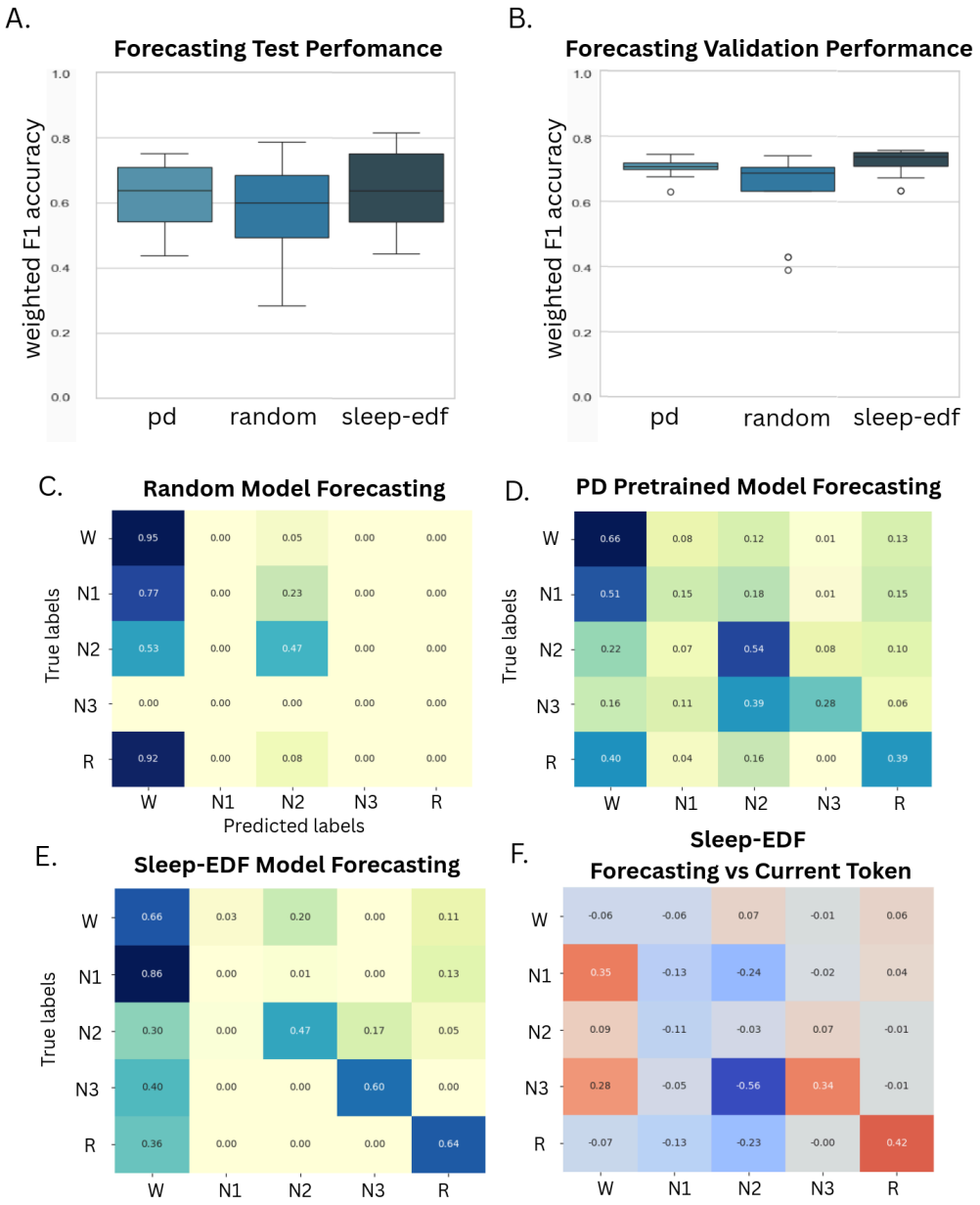


**Fig. 11.** Comparison of models with different pretrainings for the forecasting (next-token prediction) task. Statistical significance is compared towards the 'random' model (no-pretraining) **A.** Weighted F1 Accuracy median across patients for the test dataset **B**. Weighted F1 Accuracy median across patients for the validation dataset **C.** Confusion matrix for the predicted and true sleep labels for the randomly initialized model (no pretraining) on the next-token prediction task **D.** Confusion matrix for the predicted and true sleep labels for the PD reconstruction pretrained model on the next-token prediction task **E.** Confusion matrix for the predicted and true sleep labels for the Sleep-EDF reconstruction pretrained model on the next-token prediction task **F.** Difference in confusion matrices between the Sleep-EDF reconstruction-pretrained model's next-token prediction and its current-token prediction.

predictions compared to the 'current' stage prediction (34% and 42% improvement, respectively), with misclassifications occurring primarily for the W stage rather than across multiple classes. Notably, N3 is no longer misclassified as N2, representing a meaningful step toward stage differentiation (Fig. 11.E–F).

These preliminary findings suggest that pretraining provides greater benefit for forecasting (next-token) classification compared to current-token classification, especially when it comes to N3 stage predictions. The results also highlight that if the W-stage mispredictions can be further reduced, the model's overall performance could improve substantially. Future work will involve testing an STFT-based feature extractor in place of CNNs to examine whether spectral representations mitigate this W-stage confusion. Additionally, I plan to explore longer input sequences, varied reconstruction mask lengths, and predictions further into the future (e.g., +2 or +3 tokens ahead). Finally, I intend to experiment with contrastive loss as an alternative pretraining objective to evaluate whether a discriminative self-supervision signal can enhance forecasting performance.

# Chapter 4: Discussion

### 4.1 Effects of Pretraining on Classification Task

Contrary to the initial expectation that models pretrained on larger and more balanced datasets would outperform randomly initialized models, the results showed no significant improvement from pretraining for the PD sleep stage classification task when it came to the weighted F1 test scores (see Chapter 3.1). One possible explanation is that because the larger pretraining datasets were recorded from different brain regions, there was a domain mismatch between the features pretrained model extracted and the ones needed for PD dataset. The reconstruction-based pretraining might have extracted features that were representative within the source domain but not directly transferable to the STN recordings used in the PD dataset. This effect is particularly noticeable for the ECoG-pretrained model, which not only performed slightly worse than the other configurations (albeit not significantly), but also exhibited higher variance across runs (Fig. 6.A).

Interestingly, inspection of the confusion matrices revealed that pretrained models tended to perform slightly better on the N3 stage, one of the most clinically significant stages associated with deep sleep and neurotoxin clearance (Fig. 6.C-F). The difference, though modest, suggests that pretraining may have helped the model capture low-frequency oscillatory features characteristic of deep sleep, even if these representations did not translate into overall accuracy gains across all stages.

When it comes to the PD pretrained model, it showed a more balanced distribution across sleep stages in the confusion matrix (like the other pretrained models) but did not yield a higher test F1 score compared to the randomly initialized baseline. This suggests that the pretraining may have helped the model generalize slightly better across underrepresented classes, even if it did not translate into a measurable improvement in overall accuracy. The lack of significant F1 improvement may be attributed to the small size of the PD dataset. With only 13 patients and highly imbalanced sleep stage distributions—such as the complete absence of N3 sleep in some recordings—the model likely lacked sufficient examples to learn robust signal patterns during pretraining. As a result, the pretrained model did not extract substantially more useful representations than the randomly initialized model trained for 100 epochs.

Interestingly, pretrained models performed relatively better on the forecasting task, especially the Sleep-EDF pretrained model. This might be related to physiological differences in how sleep states evolve

over time. Sleep transitions are regulated by sleep-promoting neurons in the hypothalamus and GABAergic activity in the brainstem [25]. Since the PD recordings are obtained from the STN—a subcortical structure closer to these regulatory centers—there may exist a temporal shift in sleep dynamics between the PD and Sleep-EDF recordings. The forecasting objective, which requires the model to reconstruct or predict upcoming sleep epochs, may have allowed the pretrained model to capture these time-sensitive transitions more effectively than direct classification.

Although one might argue that fine-tuning on PD also involves predicting the "next" sleep stage, it is important to note that sleep scoring in PD patients is based on PSG data, which operates on a similar timescale to Sleep-EDF recordings. Therefore, the "current token" prediction in the PD dataset effectively corresponds to an n–1 step relative to the temporal progression of the neural signal, reducing the alignment between pretraining and fine-tuning objectives.

Another factor contributing to the stronger performance of the Sleep-EDF pretrained model in the forecasting task is the longer temporal context used (60 seconds versus 30 seconds). This extended input window likely provides richer information about transitional periods between sleep stages, enabling the model to learn features more relevant for predicting upcoming states. In contrast, the shorter input sequences used during classification may limit the model's ability to capture these transitions, reducing the performance gap between pretrained and randomly initialized models.

However, a critical limitation of the Sleep-EDF pretrained model is its failure to correctly predict N1 sleep, with near-zero accuracy for this stage. This presents a significant concern for potential clinical applications, as the model consistently misclassifies N1 as W. In practice, this could result in inappropriate interventions, such as stimulating the brain under the false assumption that the patient is awake when they are actually in N1, potentially disrupting natural sleep progression. Strategies to address this limitation are discussed in the section below.

### 4.2 Dealing with Imbalance in Sleep Stage Predictions and Misclassification

A major challenge observed in the models, as evident from the confusion matrices in Chapter 3.2, is the consistent misclassification of certain sleep stages and a strong bias toward predicting prevalent stages, such as W. These misclassification patterns align with previous literature and reflect known trends in Parkinson's disease (PD) sleep pathology, where disease-related disruptions alter the signal features of specific stages. For example, most false positives corresponded to misclassification of N2 sleep, which has an overlapping spectral profile with N3 [10]. Disease progression further reduces slow-wave activity

in non-REM stages, making it more difficult for the model to distinguish between N2 and N3 slow-wave patterns [7].

To mitigate these issues, in addition to standard class imbalance strategies such as inverse frequency weighting and oversampling, a penalty matrix was implemented to penalize misclassifications between physiologically dissimilar stages. However, as reported in Section 3.4, this approach did not yield measurable improvement. Interestingly, replacing the CNN-based feature extractor with STFT led to a modest improvement in classification balance. This suggests that the challenge lies less in the model's learning capacity and more in the representation of input data: spectral features appear to facilitate the discrimination of slow-wave information by emphasizing relevant frequency bands.

Notably, our STFT approach differs from prior studies. Instead of binning spectral data into a small number of frequency bands (e.g., 8 bins), we provided the full frequency range directly to the transformer. This allows the model to leverage richer spectral information and detect subtle variations, which may explain why our results surpass previous studies despite the known heterogeneity of PD spectral patterns [16].

Another model that demonstrated unexpected improvements in detecting rare stages such as N3 was the Sleep-EDF pretrained forecasting model. This enhancement likely stems from the factors discussed in Section 4.1—namely, better temporal alignment between Sleep-EDF and PD recordings and the longer input duration (60 seconds) used during forecasting. Longer sequences likely allow the model to capture more gradual transitions between sleep stages, highlighting the importance of how data is fed into the model rather than solely relying on architectural modifications. Features such as extended temporal context, well-aligned labels, or spectral inputs with rich frequency resolution can guide the transformer to extract more discriminative information efficiently.

Therefore, future directions for improving rare sleep stage classification should focus on optimizing feature extraction and temporal alignment. Strategies could include increasing sequence length (e.g., 75–90 seconds), applying label shifts similar to the forecasting task, or combining forecasting with spectral representations (using STFT instead of CNN embeddings) to provide both temporal and frequency-domain cues for better stage discrimination (especially for the N1 and N3 stages).

### 4.3 Self-Supervised Objectives: Contrastive vs. Reconstruction

The two self-supervised objectives—contrastive and reconstruction—differ fundamentally in the aspects of the signal they emphasize and the types of representations they encourage the model to learn. The

contrastive loss, as implemented in Wav2Vec 2.0, is designed to maximize separability between temporally distinct or contextually different signal segments. By explicitly encouraging the model to distinguish between positive and negative pairs, contrastive pretraining promotes highly discriminative feature representations. This property proved particularly advantageous for the classification task, as can be seen from the results in Section 3.5. The focus on discriminability for more frequently occurring stages allowed the Wav2Vec 2.0 model to outperform the PD reconstruction-pretrained model, achieving the highest weighted F1 accuracy among all tested configurations—approximately 70%. This result highlights the effectiveness of contrastive pretraining for maximizing class separability in sleep stage classification.

However, this discriminative focus comes with notable limitations. Specifically, contrastive objectives tend to prioritize patterns that are most informative for separating classes with high occurrence rates, while being less sensitive to rare or low-frequency features. In this dataset, rare stages such as N3 are characterized by subtle, slow-wave patterns that may not provide sufficient contrastive signal for the model to learn effectively. As reflected in the confusion matrix, the contrastive model misclassified N3 as N2 in 71% of instances. While this misclassification did not significantly impact the weighted F1 score—due to the low prevalence of N3—it reveals a vulnerability: the model may overlook physiologically meaningful details necessary for accurate representation of rare stages. N1 predictions were similarly weak, though this is a challenge observed across most models, suggesting a general need for improved feature representations for transitional or low-frequency stages.

In contrast, the reconstruction objective encourages the model to learn information-preserving and temporally coherent representations by reconstructing masked segments of the input signal. This process inherently emphasizes the continuity of the neural signal and the global structure of sleep dynamics. Consequently, reconstruction pretraining provides a relative advantage for detecting underrepresented or physiologically complex stages such as N3, where capturing slow-wave dynamics and temporal context is essential for accurate classification. Unlike contrastive loss, the reconstruction objective enforces the model to encode nuanced signal features that may not be immediately discriminative but are crucial for understanding the underlying neural processes.

Taken together, these results illustrate a clear trade-off between representation richness and class discriminability. Contrastive loss excels in maximizing overall classification accuracy for frequently occurring stages, whereas reconstruction loss better preserves physiologically meaningful information for rare stages, at the cost of slightly lower performance on dominant classes. This trade-off suggests that future work could benefit from hybrid approaches that integrate both objectives, such as the

contrastive–reconstruction framework explored in NeuroNet. Such a hybrid strategy could allow the model to simultaneously achieve discriminative power for common classes while maintaining sensitivity to rare, clinically significant stages like N3, thereby improving both the generalization and physiological fidelity of sleep stage predictions.

### 4.4 Overall Model Performance and Future Directions

The median weighted F1 score across patients ranged from 59% to 70%, with the highest performance achieved by the model pretrained using the contrastive loss objective (70%). At first glance, this improvement over prior studies attempting to generalize across unseen PD patients—which reported approximately 65% accuracy—may seem modest [9]. However, this study explicitly distinguishes between N2 and N3 sleep stages, which are notoriously difficult to separate in PD patients due to the rarity and disrupted characteristics of N3 sleep [7]. In fact, when previous studies included N3, overall accuracy dropped to 58% [9], highlighting the meaningful enhancement achieved here. Comparisons to classification tasks performed on healthy STN data from macaques, which achieved 62–75% accuracy with only two subjects [8], further emphasize the significance of these results in the more challenging PD population.

Although the overall weighted F1 score did not increase dramatically, confusion matrices reveal improvements in the classification of rarer stages, suggesting that the models are learning more balanced decision boundaries across all sleep stages. Nevertheless, expert human scorers typically achieve 76.8–82% consensus in sleep stage classification [10], indicating that further refinements are necessary to approach expert-level performance.

To further enhance model performance, aside from the strategies discussed in the previous sections, one promising direction is the adoption of an ensemble learning approach. By training multiple models on different subsets of patients and aggregating their predictions for a withheld patient, it would be possible to reduce individual model biases and minimize stage-specific misclassifications. Since each model might overfit to different aspects of the data, combining their outputs through a voting or averaging mechanism could yield a more robust and generalizable classifier, particularly for underrepresented or ambiguous sleep stages.

Another avenue for improvement involves using pretraining datasets recorded from anatomically closer regions to the STN. While direct recordings from healthy human STN are understandably unavailable, non-human primate studies—such as recordings from macaque STN—could provide valuable pretraining

data [8]. Although such datasets are typically limited in size (often consisting of recordings from only one or two subjects), they could still serve as a domain-relevant prior, helping the model to recognize sleep-related patterns in PD recordings that resemble healthy STN activity. This could ultimately improve the model's sensitivity to subtle pathological deviations in sleep-stage transitions.

Finally, an important consideration concerns the practical deployment of the model. In an ideal scenario, the system would operate in real time within a closed-loop DBS device, dynamically classifying ongoing neural signals and adjusting stimulation parameters to encourage or suppress specific sleep transitions. However, the transformer-based architecture used in the current work may pose engineering challenges, particularly regarding computational complexity, memory footprint, and inference latency. Future work should therefore explore model distillation, pruning, or lightweight architectural variants to reduce computational demands while maintaining accuracy. These optimizations would make the model more suitable for embedded, real-time DBS applications, bridging the gap between algorithmic development and clinical deployment.

# Bibliography

[1] J. E. González-Naranjo *et al.*, “Analysis of Sleep Macrostructure in Patients Diagnosed with Parkinson’s Disease,” *Behav Sci (Basel)*, vol. 9, no. 1, p. 6, Jan. 2019, doi: 10.3390/bs9010006.

[2] Z. Xu *et al.*, “Progression of sleep disturbances in Parkinson’s disease: a 5-year longitudinal study,” *J Neurol*, vol. 268, no. 1, pp. 312–320, Jan. 2021, doi: 10.1007/s00415-020-10140-x.

[3] G. W. Arendash, “The Brain Toxin Cleansing of Sleep Achieved During Wakefulness,” *Journal of Clinical Medicine*, vol. 14, no. 3, p. 926, Jan. 2025, doi: 10.3390/jcm14030926.

[4] O. C. Reddy and Y. D. van der Werf, “The Sleeping Brain: Harnessing the Power of the Glymphatic System through Lifestyle Choices,” *Brain Sciences*, vol. 10, no. 11, p. 868, Nov. 2020, doi: 10.3390/brainsci10110868.

[5] A. K. Patel, V. Reddy, K. R. Shumway, and J. F. Araujo, “Physiology, Sleep Stages,” in *StatPearls*, Treasure Island (FL): StatPearls Publishing, 2025. Accessed: Oct. 05, 2025. [Online]. Available: http://www.ncbi.nlm.nih.gov/books/NBK526132/

[6] I. Arnulf *et al.*, “Improvement of sleep architecture in PD with subthalamic nucleus stimulation,” *Neurology*, vol. 55, no. 11, pp. 1732–1734, Dec. 2000, doi: 10.1212/wnl.55.11.1732.

[7] C. Smyth, M. F. Anjum, S. Ravi, T. Denison, P. Starr, and S. Little, “Adaptive Deep Brain Stimulation for sleep stage targeting in Parkinson’s disease,” *Brain Stimulation*, vol. 16, no. 5, pp. 1292–1296, Sep. 2023, doi: 10.1016/j.brs.2023.08.006.

[8] N. Barbe *et al.*, “Toward an Automatic Classification of the Different Stages of Sleep: Exploring Patterns of Neural Activity in the Subthalamic Nucleus,” *Eur J Neurosci*, vol. 61, no. 7, p. e70107, Apr. 2025, doi: 10.1111/ejn.70107.

[9] Y. Chen *et al.*, “Automatic Sleep Stage Classification Based on Subthalamic Local Field Potentials,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 2, pp. 118–128, Feb. 2019, doi: 10.1109/TNSRE.2018.2890272.

[10] K. Carver, K. Saltoun, E. Christensen, A. Abosch, J. Zylberberg, and J. A. Thompson, “Towards automated sleep-stage classification for adaptive deep brain stimulation targeting sleep in patients with Parkinson’s disease,” *Commun Eng*, vol. 2, no. 1, p. 95, Dec. 2023, doi: 10.1038/s44172-023-00150-8.

[11] X. Lun, Z. Yu, T. Chen, F. Wang, and Y. Hou, “A Simplified CNN Classification Method for MI-EEG via the Electrode Pairs Signals,” *Frontiers in Human Neuroscience*, vol. 14, 2020, Accessed: Oct. 02, 2023. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fnhum.2020.00338

[12] Y. T. Yew, J. Hong Puah, C. K. Chan, N. Mohd Zain, C. K. Toa, and S. Kuan Goh, “EEG Sleep Stages Classification using Transformer-based Models: A Review,” in *2024 International Conference on Computing Innovation, Intelligence, Technologies and Education (CIITE)*, Sep. 2024, pp. 1–6. doi: 10.1109/CIITE62244.2024.10987610.

[13] C. Wang *et al.*, “BrainBERT: Self-supervised representation learning for intracranial recordings,” Feb. 28, 2023, *arXiv*: arXiv:2302.14367. doi: 10.48550/arXiv.2302.14367.

[14] D. Zhang, Z. Yuan, Y. Yang, J. Chen, J. Wang, and Y. Li, “Brant: Foundation Model for Intracranial Neural Signal”.

[15] C.-H. Lee, H. Kim, H. Han, M.-K. Jung, B. C. Yoon, and D.-J. Kim, “NeuroNet: A Novel Hybrid Self-Supervised Learning Framework for Sleep Stage Classification Using Single-Channel EEG,” May 13, 2024, *arXiv*: arXiv:2404.17585. doi: 10.48550/arXiv.2404.17585.

[16] J. A. Thompson *et al.*, “Sleep patterns in Parkinson’s disease: direct recordings from the subthalamic nucleus,” *J Neurol Neurosurg Psychiatry*, vol. 89, no. 1, pp. 95–104, Jan. 2018, doi: 10.1136/jnnp-2017-316115.

[17] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations,” Oct. 22, 2020, *arXiv*: arXiv:2006.11477. Accessed: Sep. 26, 2023. [Online]. Available: http://arxiv.org/abs/2006.11477

[18] B. Kemp, A. H. Zwinderman, B. Tuk, H. A. C. Kamphuisen, and J. J. L. Oberye, “Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the EEG,” *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 9, pp. 1185–1194, Sep. 2000, doi: 10.1109/10.867928.

[19] Y. Zhang, A. Daida, L. Liu, N. Kuroda, Y. Ding, S. Oana, T. Monsoor, C. Duan, S. A. Hussain, J. X. Qiao, N. Salamon, A. Fallah, M. S. Sim, R. Sankar, R. J. Staba, J. Engel, E. Asano, V. Roychowdhury, and H. Nariai, "Open iEEG Dataset," *OpenNeuro*, 2024. [Online]. Available: https://doi.org/10.18112/openneuro.ds005398.v1.0.1

[20] S. M. M. de Mooij, T. F. Blanken, R. P. P. P. Grasman, J. R. Ramautar, E. J. W. Van Someren, and H. L. J. van der Maas, “Dynamics of sleep: Exploring critical transitions and early warning signals,” *Computer Methods and Programs in Biomedicine*, vol. 193, p. 105448, Sep. 2020, doi: 10.1016/j.cmpb.2020.105448.

[21] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, “Masked Autoencoders Are Scalable Vision Learners,” Dec. 19, 2021, *arXiv*: arXiv:2111.06377. doi: 10.48550/arXiv.2111.06377.

[22] K. O’Shea and R. Nash, “An Introduction to Convolutional Neural Networks,” Dec. 02, 2015, *arXiv*: arXiv:1511.08458. doi: 10.48550/arXiv.1511.08458.

[23] D. Goyal and B. S. Pabla, “Condition based maintenance of machine tools—A review,” *CIRP Journal of Manufacturing Science and Technology*, vol. 10, pp. 24–35, Aug. 2015, doi: 10.1016/j.cirpj.2015.05.004.

[24] A. Vaswani *et al.*, “Attention Is All You Need,” Aug. 02, 2023, *arXiv*: arXiv:1706.03762. doi: 10.48550/arXiv.1706.03762.

[25] “Brain Basics: Understanding Sleep | National Institute of Neurological Disorders and Stroke.” Accessed: Oct. 21, 2025. [Online]. Available: https://www.ninds.nih.gov/health-information/public-education/brain-basics/brain-basics-understanding-sleep

# Appendix A

### Convolution Information:

| Data Type | Kernel | Stride |
|---|---|---|
| PD | (1250, 625, 100) | (3,2,2) |
| ECoG | (1250, 625, 100) | (3,2,2) |
| Sleep-EDF | (100, 112, 40) | (2, 2, 2) |

**Configuration Changes for Different Feature Extractors**

| Extractor Type | Masking_time_length | Extracted_feature_dim |
|---|---|---|
| CNN | 50 | 64 |
| STFT | 1 | 251 |