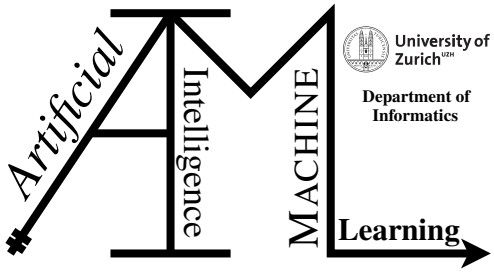
Towards Scalable Foundation Models for Sleep EEG and Polysomnography

Master Thesis

Daniel Lutziger
18-642-648

Submitted on
December 9, 2025

Thesis Supervisor
Prof. Dr. Manuel Günther, Dr. André Anjos
Dr. Olivier Pallanca



Declaration of Independence for Written Work

I hereby declare that I have **composed** this work independently and without the use of any aids other than those declared (including generative AI such as ChatGPT) – the use of generative AI to **improve** my composed work was permitted by the thesis supervisor. I am aware that I take full responsibility for the scientific character of the submitted text myself, even if AI aids were used. All passages taken verbatim or in sense from published or unpublished writings are identified as such. The work has not yet been submitted in the same or similar form or in excerpts as part of another examination.

Place, Date

Daniel Lutziger

Master Thesis

Author: Daniel Lutziger, daniel.lutziger@uzh.ch

Project period: June 9, 2025 - December 9, 2025

Artificial Intelligence and Machine Learning Group
Department of Informatics, University of Zurich

Acknowledgements

I would like to express my sincere gratitude to my supervisors, Professor Dr. Manuel Günther, Dr. André Anjos and Dr. Olivier Pallanca, for their invaluable guidance and support throughout this thesis. I could always turn to them with any question or doubt, and their feedback and advice consistently helped me find the right direction. I am also grateful to Professor Günther for his Deep Learning course, which sparked my interest in this field and motivated me to explore it further in my research.

I also deeply appreciate my family and friends for their constant encouragement and support. Their understanding, patience and confidence in me have been a continuous source of motivation throughout this journey. Knowing that they believed in me, even at times when I doubted myself, has given me strength and made this work possible. For this, I am truly grateful. Thank you.

Abstract

Automated analysis of polysomnography (PSG) remains challenging: recordings differ in montages, label distributions, and clinical tasks, making it hard to train models that generalize across cohorts. This thesis studies scalable foundation models for sleep PSG, building on a multimodal set-then-sequence Transformer encoder pretrained contrastively on about 13'000 subjects from ten public cohorts. We freeze this encoder and train lightweight sequence heads for two downstream tasks on a held-out SHHS cohort: sleep staging and sleep apnea detection. We systematically vary pretraining data scale, loss functions for class imbalance, downstream modality sets, and contrastive objectives. Pretraining on this large, heterogeneous pool yields strong downstream performance, reaching 76.2% balanced accuracy for BAS-only sleep staging on SHHS, while maintaining competitive apnea detection performance and showing that respiratory channels are critical for apnea but of limited value for staging. Overall, the experiments indicate that large PSG foundation models can be reused across tasks and montages, but that their benefits depend on the available modalities.

Contents

Chapter 1

Introduction

Sleep is a fundamental biological need with wide-ranging consequences for cognition, health, and society. Insufficient or disordered sleep is associated with impaired memory, reduced productivity, and higher risks for chronic diseases, and insomnia alone imposes large economic burdens through health care costs and lost productivity (Hafner et al., 2023). Sleep disorders include hypersomnia, circadian rhythm disorders, parasomnias and others, affecting quality of life and daytime functioning (Mahowald and Schenck, 2005).

Overnight polysomnography (PSG) — a multichannel recording including electroencephalogram (EEG), electrooculography (EOG), electromyography (EMG) and respiratory signals — remains the gold standard for objectively characterizing sleep architecture and many sleep disorders (Silber et al., 2007). An overview of the PSG setup and a representative 30-second segment of the recorded signals is shown in Figure 1.1. Clinically, PSG recordings are segmented into discrete 30-second epochs for scoring, and practitioners annotate abnormal events (*e. g.* apneas, arousals, atypical movements), providing clinically actionable information for conditions such as obstructive sleep apnea and parasomnias (Silber et al., 2007; Mahowald and Schenck, 2005). Typical clinical and research tasks based on PSG include sleep staging, arousal detection, characterization of sleep-disordered breathing, and estimation of indices such as the apnea-hypopnea index (AHI).

Despite its centrality, manual PSG analysis is resource-intensive. Trained practitioners assign a sleep stage to each 30-second epoch according to standardized rules such as the Rechtschaffen and Kales (R&K) and American Academy of Sleep Medicine (AASM) manuals, a manual procedure that is time-consuming and costly, creating bottlenecks for both clinics and research. These manuals define the canonical stages Wake, N1, N2, N3, and rapid eye movement (REM) sleep. Inter-scorer agreement is imperfect and stage-dependent, with especially low consensus for light transitional sleep (stage N1), which complicates benchmarking of algorithms and clinical interpretation (Lee et al., 2022). Agreement for respiratory event scoring also varies across scorers and centers (Magalang et al., 2013). Together, the labor costs and inter-scorer variability motivate automation of PSG analysis.

Deep learning has advanced automatic sleep staging by learning features directly from raw signals. Convolutional and recurrent neural network models can reach human-level performance on public datasets under controlled conditions (Supratak et al., 2017). When temporal context is modeled explicitly and training spans multiple cohorts, performance further improves and approaches expert agreement (Brunini, 2023). However, two gaps limit clinical translation. First, models trained on one cohort or montage can perform poorly on others, which restricts generalizability. Second, the decision processes of deep neural networks are often opaque, which reduces clinician trust and slows adoption (Montavon et al., 2018).

Recent work explores *foundation models* (FMs): large, pretrained encoders learned from diverse, large-scale corpora and adapted to many downstream tasks. Analogous to vision and

language (He et al., 2022), PSG FMs aim to learn generalizable sleep representations that transfer to staging, arousal detection, sleep-disordered breathing, and even long-term risk prediction. SleepFM pretrains multimodal encoders (EEG, ECG, respiration) and reports strong transfer to standard sleep tasks and disease prediction across more than 500'000 hours of PSG (Thapa et al., 2024, 2025). These results point to scalable, cohort-robust solutions but raise questions about objective choice, and channel/montage flexibility when adapting such models to new datasets and tasks.

1.1 Problem Setting

We target sleep EEG and PSG models that generalize across sites and populations. Achieving such generalization is challenging due to between-site differences in sensor montages and amplifier characteristics, demographic variation, and shifts in the prevalence and presentation of sleep disorders (Montavon et al., 2018). Foundation-style pretraining on heterogeneous datasets offers a path toward robustness and label efficiency. In our setting, *pretraining* refers to learning a generic encoder on large amounts of unlabeled PSG data, which is then reused for specific supervised tasks that we refer to as *downstream tasks*. Designing such a pretraining pipeline requires careful choices of objective, for example contrastive or masked reconstruction objectives, as well as channel-flexible modeling and rigorous cross-cohort evaluation (Thapa et al., 2024, 2025). Broadly, contrastive objectives encourage similar embeddings for different views of the same sample and dissimilar embeddings for different samples, often using an InfoNCE-style loss, as detailed in Section 2.3.1 and Section 3.3.1. Masked reconstruction objectives remove or corrupt parts of the input and train the model to reconstruct them, which encourages rich internal representations of signal structure, as discussed in Section 2.3.2. These families of objectives are not mutually exclusive and are both actively explored in the sleep and time-series literature. In this work, we focus on contrastive objectives with modality-aware dropout and summarize reconstruction-based approaches for context.

We investigate scalable deep models for PSG. The models are pretrained on heterogeneous multimodal signals, are designed to transfer to EEG-only downstream tasks, remain robust under class imbalance and montage variation, and provide transparent analyses that are aligned with physiological features. We focus on two downstream tasks that are central in clinical sleep medicine, namely *sleep staging* and *sleep apnea detection*.

We leverage public cohorts (*e. g.* MASS, Sleep-EDF, MESA) and insights from SleepFM and related work (Thapa et al., 2025) to study data efficiency, objective choice, and modality contributions, always with an eye to cross-cohort robustness.

1.2 Clinical Background

In the following sections, we will focus on the clinical background for sleep analyses and our downstream tasks. We take a closer look at PSG data and apnea.

1.2.1 Polysomnography (PSG)

PSG is an overnight multiparametric test that synchronously records various biophysical signals during sleep and wake to assess sleep architecture and detect sleep disorders. It is considered the gold-standard diagnostic tool for conditions like obstructive sleep apnea (OSA) and related sleep disorders, providing a comprehensive evaluation of sleep stages and physiological events (Carmel, 2023). A full PSG montage typically includes channels monitoring brain activity, cardiac

rhythm, muscle tone, and respiratory parameters, among others. These signals are acquired non-invasively via surface electrodes or sensors placed according to standardized configurations to ensure reproducibility (Carmel, 2023). Below, we briefly outline the core PSG signals and their significance in sleep monitoring.

EEG

Electroencephalography (EEG) in PSG records the brain's electrical activity and is the primary basis for sleep stage classification. Multiple EEG leads (usually frontal, central, occipital derivations with a reference) are placed according to the 10-20 system, which ensures consistent scalp electrode positions (Carmel, 2023). EEG allows identification of characteristic waveforms associated with different sleep stages (frquency bands): *alpha waves* (8-13Hz) dominate relaxed wakefulness, *theta* activity (3-7Hz) emerges in stage N1, and distinctive features such as sleep spindles in the *sigma* band (12-14Hz bursts) and K-complexes mark stage N2 (Carmel, 2023). In stage N3 (deep slow-wave sleep), high-amplitude *delta* waves (0.5-2Hz) prevail, whereas rapid eye movement (REM) sleep shows low-amplitude mixed-frequency EEG resembling wakefulness (Carmel, 2023). By capturing these frequency bands and waveform changes, the EEG is indispensable for determining sleep onset, differentiating non-REM stages (N1-N3), and detecting REM sleep. Figure 1.2 showcases the placement of the electrodes for EEG.

ECG

During an overnight sleep study (polysomnography), a single-lead ECG, typically derived from two torso electrodes, is recorded to monitor heart rate and cardiac rhythm. ECG is used to identify rate changes and rhythm abnormalities that can appear during sleep and around breathing events (Berry et al., 2017). In obstructive apneas, it is common to see brief slow-fast heart-rate swings tied to the pause and subsequent arousal, and the ECG channel flags more sustained issues if they occur (Berry et al., 2017).

EMG

Electromyography (EMG) in PSG measures muscle activity using surface electrodes, most importantly a chin (submental) EMG for sleep staging. The chin EMG signal is crucial for identifying REM sleep, which is characterized by an almost complete loss of muscle tone (atonia) (Carmel, 2023). A high chin EMG amplitude indicates muscle tension (seen in wake or arousals), whereas a low or absent chin EMG denotes the REM atonia or very relaxed muscle tone of deep non-REM sleep (Carmel, 2023). In addition to the chin, EMG leads may be placed on the anterior tibialis muscles of the legs to capture periodic limb movements or other movement disorders during sleep (Carmel, 2023). The EMG recordings complement EEG/EOG signals by confirming stage transitions (*e. g.* the drop in chin EMG at REM onset) and by documenting abnormal movements that have clinical significance.

Respiration

Respiratory monitoring is a central component of PSG, aimed at detecting apnea, hypopnea, and related breathing abnormalities. Airflow is measured using a nasal pressure transducer (for sensitive detection of flow limitation and hypopneas) and an oronasal thermal sensor (thermistor/thermocouple) which reliably detects absence of airflow (apneas) (Carmel, 2023). Respiratory effort is recorded via thoracic and abdominal inductance plethysmography belts, which gauge chest and abdominal movement to distinguish obstructive events (effort present) from central events (effort

absent) (Carmel, 2023; Berry et al., 2017). Pulse oximetry on the finger continuously tracks blood oxygen saturation, revealing oxygen desaturations associated with apneas/hypopneas (Berry et al., 2017). Snoring sounds are often captured by a microphone or vibration sensor, and an end-tidal CO_2 monitor or transcutaneous CO_2 may be used in select cases to assess hypoventilation (Berry et al., 2017). By integrating these measures, PSG respiration channels characterize the presence, duration, and severity of breathing disturbances during sleep. They allow precise identification of apneas (complete airflow cessations), hypopneas (partial flow reductions), and respiratory effort-related arousals, which form the diagnostic basis for sleep-disordered breathing like OSA.

1.2.2 Staging Standards

The AASM guidelines provide standardized rules for sleep staging that are widely used in both clinical practice and research. Overnight PSG is divided into 30-second epochs, and each epoch is assigned one of five stages: Wake, N1, N2, N3, or REM sleep (Berry et al., 2017). Stage definitions are based on combinations of EEG, EOG, and chin-EMG features.

In this thesis we adopt these specific guidelines as the basis for all supervised experiments. Sleep staging labels are treated as ground truth when training and evaluating downstream models. Respiratory events and indices such as the AHI are likewise defined according to the AASM rules described in Section 1.2.3 and form the targets for our apnea related analysis.

1.2.3 Sleep-Disordered Breathing

Sleep-disordered breathing is scored from airflow, respiratory effort and oxygen saturation signals according to AASM rules (Berry et al., 2017). In this thesis we distinguish apnea, hypopneas and respiratory effort related arousals (RERAs), and we summarize their severity with the apnea-hypopnea index (AHI). Below we collect the definitions and the AHI convention used throughout the rest of the thesis.

Apnea

An apnea is the complete cessation of airflow for a sufficient duration during sleep. Per AASM criteria, an apnea is scored when there is a $\geq 90\%$ drop in airflow amplitude from baseline, as measured by the oronasal thermal sensor (or equivalent flow signal), lasting at least **10 seconds** (Berry et al., 2017). During an obstructive apnea (OSA), respiratory effort continues, reflecting an occluded upper airway. In a central apnea, by contrast, there is no respiratory effort (flat effort belts) because neural drive to breathe transiently stops (Berry et al., 2017). A mixed apnea begins as central (no effort at onset) and ends as obstructive (effort resumes before airflow returns) (Berry et al., 2017). All apneas, regardless of type, cause a pause in ventilation that often leads to blood oxygen desaturation and cortical arousal once breathing resumes. Figure 1.3 illustrates how an apnea event in a PSG recording looks like. From the Figure we can clearly see the drop in SpO_2 and the lack of effort in the signals.

Hypopnea

A hypopnea is a partial reduction of airflow during sleep, less severe than an apnea but still physiologically consequential. The AASM defines a hypopnea in adults as a period of diminished breathing lasting ≥ 10 seconds with a clear decrease in airflow amplitude (by at least 30% relative to baseline) accompanied by either an oxygen desaturation $\geq 3\%$ or an arousal from sleep (Berry et al., 2017). In practice, hypopneas are identified on the nasal pressure channel as transient

flattenings or amplitude drops of the waveform, indicating reduced inspiratory flow (Berry et al., 2017). If the event meets or exceeds the 30% flow reduction threshold and triggers the requisite SpO_2 drop or arousal, it is scored as a hypopnea. Figure 1.3 illustrates how a hypopnea event in a PSG recording can look like.

RERA

A respiratory effort-related arousal (RERA) is a sequence of breaths that does not meet the apnea or hypopnea criteria but still culminates in an arousal due to increased respiratory effort. According to AASM guidelines, a RERA is scored when there is a period of increasing ventilatory effort or flow limitation (*e. g.* progressive flattening of the nasal pressure waveform) lasting ≥ 10 seconds, followed by an abrupt EEG arousal, in the absence of criteria qualifying as an apnea or hypopnea (Berry et al., 2017). In other words, a RERA represents a subtle breathing abnormality that fragments sleep (via arousal) despite only a minor reduction in airflow. Inclusion of RERAs thus provides a more comprehensive assessment of sleep-disordered breathing events that disturb sleep continuity.

Normal Sleep

In healthy individuals, sleep is uninterrupted by frequent respiratory events, and ventilation remains stable throughout the night. It is normal for breathing to slow during non-REM sleep and even exhibit occasional brief pauses (up to 10 seconds) as part of periodic breathing, especially in transitional sleep. However, these pauses do not cause appreciable oxygen desaturation or arousal. During normal sleep, oxygen saturation stays generally above 95%, and sleep architecture is continuous (Shrivastava et al., 2014).

1.2.4 Severity Metrics

The AHI is a key quantitative measure used to assess the severity of sleep apnea syndromes. AHI is defined as the total number of apneas A plus hypopneas observed H, divided by the total sleep time T in hours, yielding a frequency of respiratory events per hour of sleep (Berry et al., 2017):

$$\text{AHI} = \frac{A + H}{T} \tag{1.1}$$

By convention, an AHI < 5 events is considered within normal limits and indicates no clinically significant sleep apnea (Berry et al., 2017). Higher AHIs correlate with greater risk of neurocognitive impairment and cardiovascular morbidity (Quan et al., 1997). Clinicians use AHI thresholds to classify OSA severity into *mild* (AHI 5-14), *moderate* (AHI 15-30), and *severe* (AHI >30) (Berry et al., 2017). These cut-offs, endorsed by the AASM and other societies, reflect an increasing measurement of disease: higher AHI values generally indicate more frequent oxygen desaturations and arousals, which can lead to fragmented sleep. That said, AHI is an imperfect metric of severity in isolation, as it does not capture the depth of desaturations or duration of events. As seen in Equation (1.1), only the events count and neither the duration nor the depth are quantified. Therefore, two patients with the same AHI can have very different clinical outcomes if one has short shallow hypopneas and the other has long apneas with profound desaturations. Nonetheless, AHI remains the most widely used summary index for the measurement of sleep apnea severity in both clinical practice and research due to its simplicity and objectivity (Quan et al., 1997).

1.3 Research Questions and Hypotheses

We organize this thesis around five research questions that link pretraining scale and objectives to downstream transfer, modality design, and imbalance-aware training. Recent PSG foundation models based on large-scale multimodal contrastive pretraining demonstrate that such encoders can learn strong, transferable representations (Thapa et al., 2024, 2025; Fox et al., 2025), but they leave open how much unlabeled data is actually needed, how to handle label imbalance during fine-tuning, and when additional modalities beyond BAS (EEG+EOG) provide a tangible benefit. The research questions and hypotheses below spell out how we address these issues in the remainder of the thesis.

RQ1 - Data efficiency of pretraining.

- **RQ1:** *How does downstream sleep-staging performance change when we vary the amount of data used for contrastive pretraining of the foundation encoder?*
- **Scope:** Pretraining uses multimodal PSG (BAS, EKG, EMG, respiratory); for this RQ, downstream tasks use BAS-only inputs.
- **H1a (Pretraining scale):** Increasing the pretraining set size improves linear-probe and downstream task performance monotonically across $\{1\text{k}, 2\text{k}, 3\text{k}, 5\text{k}, 9\text{k}, \text{all}\}$ subjects.

RQ2 - Transfer learning of FM embeddings to downstream tasks (BAS-only).

- **RQ2:** *For a single multimodal PSG foundation encoder pretrained with a contrastive self-supervised objective, to what extent do frozen BAS-only embeddings improve sleep staging and sleep apnea detection on a held-out cohort compared with training the same downstream architectures from scratch?*
- **Scope:** Pretraining uses multimodal PSG; downstream tasks use BAS-only.
- **H2a (Staging, pretrained vs scratch):** Full fine-tuning on frozen FM embeddings achieves higher F1, Cohen's κ, and balanced accuracy on the held-out cohort for sleep staging than a supervised LSTM trained from scratch (same BAS input).
- **H2b (Apnea, pretrained vs scratch):** For apnea detection, a classifier trained on frozen FM embeddings outperforms the same architecture trained from scratch, in terms of balanced accuracy, AUROC and AUPR.

RQ3 - Handling class imbalance in downstream sleep staging.

- **RQ3:** *When training sleep-staging models on imbalanced PSG data, how do different loss functions influence performance on minority versus majority sleep stages?*
- **Scope:** Downstream sleep-staging task heads; compares class-weighted cross-entropy and focal loss.
- **H3a (Loss weighting):** Compared to class-weighted cross-entropy, using focal loss improves recall and F1 for minority stages (especially N1) while keeping overall F1 and balanced accuracy at a comparable level.

RQ4 - Contribution of additional modalities beyond BAS.

- **RQ4:** *Given BAS-only downstream benchmarking, do added modalities (EOG/EMG/respiratory) improve performance, and for which tasks?*
- **Scope:** Pretraining uses multimodal PSG; downstream experiments compare BAS-only and BAS+{EKG, EMG, respiratory} inputs.
- **H4a (Staging):** Adding EKG, EMG, and respiratory channels to BAS improves REM balanced accuracy and reduces Wake/N1 confusions.
- **H4b (Apnea):** For apnea detection, adding EKG/EMG/respiratory channels improves balanced accuracy, AUROC and AUPR over BAS-only embeddings.

RQ5 - Pretraining objectives and modality-aware training.

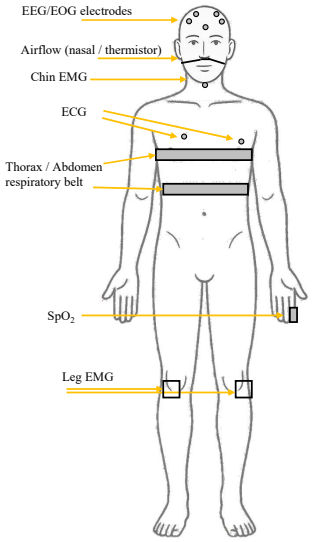
- **RQ5:** *How does the pretraining objective affect downstream transfer when pretraining uses all four modalities without discarding incomplete data?*
- **Scope:** Pretraining uses multimodal PSG with modality dropout; downstream tasks follow the BAS-only setup of RQ2.
- **H5a (SimCLR augmentation):** Adding a SimCLR-style contrastive term to the pretraining objective, together with modality dropout, yields higher F1, Cohen's κ, and balanced accuracy than using the leave-one-out contrastive (LOOC) objective alone under identical data, architecture, and compute budgets.

1.4 Thesis Outline

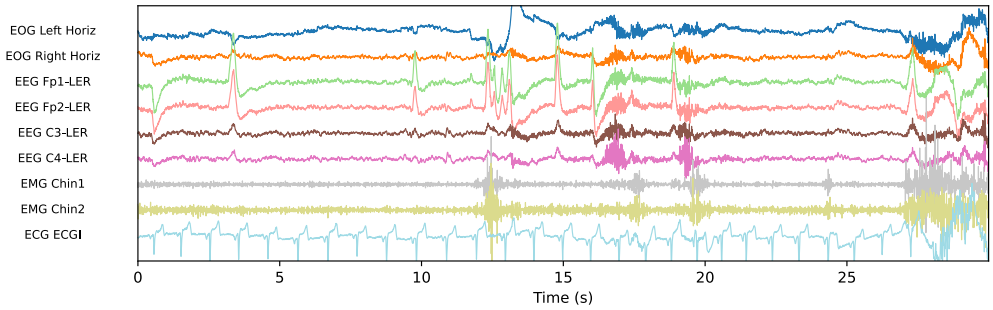
This master thesis is structured as follows. Chapter 2 reviews related work from classical pipelines to foundation models, with emphasis on cross-cohort generalization and interpretability. Chapter 3 summarizes PSG signals, staging standards, attention-based tokenization, and evaluation metrics that are required to understand our methods. Chapter 4 describes the datasets and harmonization procedures. Chapter 5 presents our foundation-model encoder, baseline models, self-supervised pretraining objectives, and the training and evaluation pipelines. Chapter 6 explains how these components are instantiated in concrete experiments, including pretraining setups, data splits, and downstream protocols for each research question. Chapter 7 reports the empirical results organized by research question and hypothesis. Chapter 8 interprets these findings, discusses limitations, situates them within the broader context and foundation models, and outlines directions for future work. Finally, Chapter 9 synthesizes the main conclusions and summarizes as a whole.

1.5 Usage of AI

Generative AI tools were used in this thesis in a limited, assistive role. In particular, ChatGPT supported the author in editing, paraphrasing, and refining the text, and Claude was used as a coding assistant. The scientific ideas, experimental designs, analyses, and conclusions were developed by the author. Whenever AI tools proposed alternative formulations, summaries, or claims, these outputs were manually reviewed and, where relevant, checked against the original sources before inclusion. Responsibility for the correctness and interpretation of all content remains with the author.



(a) Schematic illustration of key PSG sensors, including EEG and EOG electrodes, chin and leg EMG, ECG leads, a nasal airflow sensor, thoracic and abdominal respiratory belts, and a finger pulse oximeter (SpO_2).



(b) Example 30-second PSG segment showing corresponding EEG, EOG, EMG, ECG, airflow, and respiratory effort channels.

Figure 1.1: OVERVIEW OF POLYSOMNOGRAPHY SENSORS AND SIGNALS. *This figure summarizes the polysomnography setup and a representative recording segment. Subfigure (a) shows the main PSG sensors used to monitor brain activity, eye movements, muscle tone, cardiac rhythm, airflow, respiratory effort, and oxygen saturation. Subfigure (b) presents an example 30-second PSG segment from MASS that illustrates how EEG, EOG, EMG, ECG, airflow, and respiratory effort signals appear in the raw recording.*

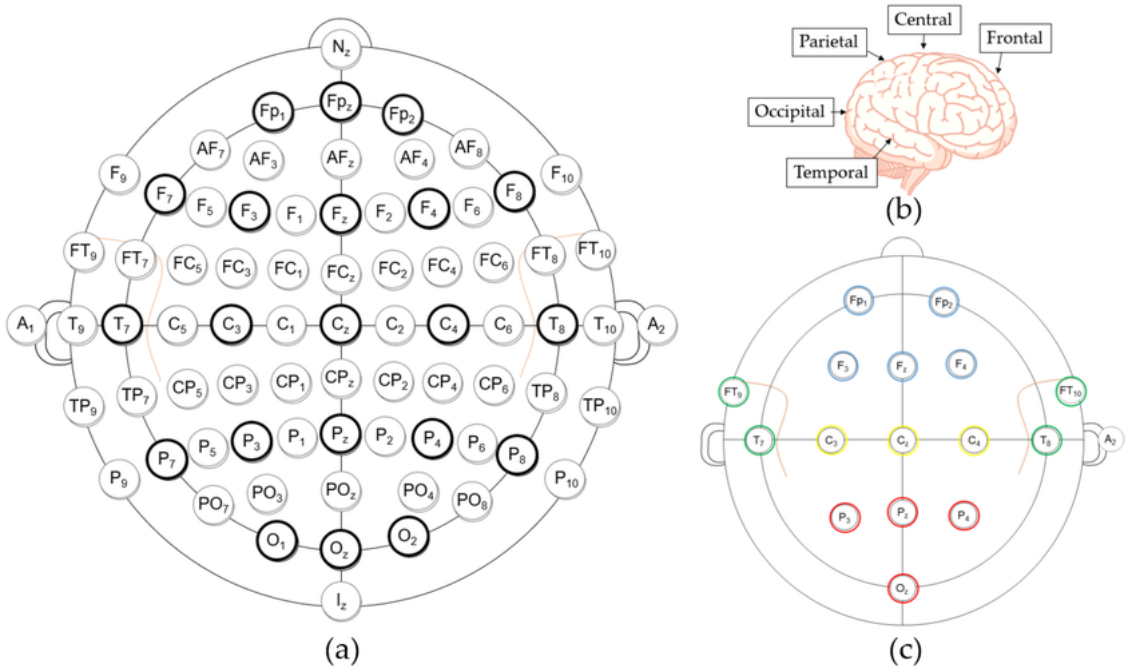


Figure 1.2: ELECTRODE PLACEMENT AND LOBES OVERVIEW. *(a) Electrode placement system. (b) The cerebral lobes and central area. (c) The placement configuration of 16 electrodes (Hong and Baek, 2021).*

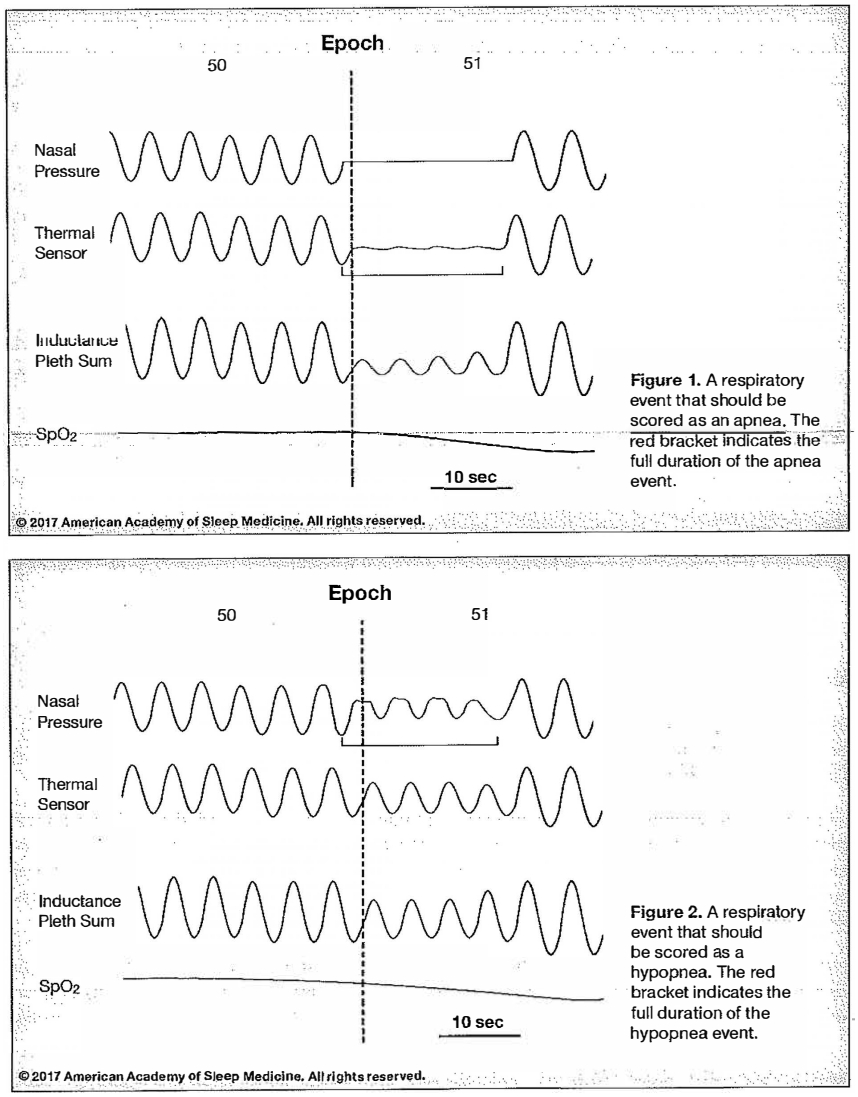


Figure 1.3: APNEA VS HYPOPNEA. *Comparison between apnea and hypopnea in a PSG recording. The upper recording showcases an apnea event, where no effort is visible, the lower recording showcases a hypopnea event, where effort is visible (Berry et al., 2017).*

Chapter 2

Related Work

Over decades, methods for automated sleep analysis have progressed from pipelines based on handcrafted features and traditional classifiers to end-to-end deep neural networks that learn directly from raw polysomnography (PSG) signals, and more recently to large-scale self-supervised and foundation models that provide general-purpose encoders (Craik et al., 2019; Supratak et al., 2017; Stephansen et al., 2018; Thapa et al., 2024, 2025). Throughout this evolution, researchers have grappled with cross-cohort generalization, robustness to varying channel configurations, and the interpretability of model decisions in clinical practice (Perslev et al., 2021; Guo et al., 2024; Montavon et al., 2018; Phan et al., 2022).

In this chapter we review manual scoring and traditional feature-based pipelines, deep neural architectures for PSG, self-supervised learning and foundation models for sleep, and the main downstream tasks such as sleep staging and apnea detection to which these models are applied.

2.1 Traditional Feature-Based Pipelines

The first wave of automatic sleep staging predates deep learning. Typical systems worked one 30-second epoch at a time: they extracted a fixed set of hand-crafted features and fed them into a conventional classifier such as an SVM, LDA, or a gradient-boosted decision tree, sometimes followed by a simple temporal smoother like a hidden Markov model (Craik et al., 2019). The feature sets were designed to mirror what human scorers look at. In the time domain they included basic statistics such as signal variance. In the frequency domain they computed absolute or relative power in the familiar EEG bands — *delta*, *alpha*, *theta*, and *sigma* — and ratios between them, together with time-frequency representations (wavelet or short-time Fourier transforms) and a handful of non-linear measures (entropies, Hjorth parameters) (Craik et al., 2019). In other words, these models operated on numerical summaries of the same *alpha*, *theta*, spindle-related *sigma*, and slow-wave *delta* activity that define the stages clinically (Carmel, 2023, see also the EEG overview 1.2.1).

A recent survey by Zong et al. (2025) points out that such pipelines still have a few clear strengths. Because many features have a direct physiological meaning (for example, high *delta* power in slow-wave sleep or a bump in *sigma* power when spindles are present), their decisions are comparatively easy to interpret. They can also be trained with relatively few labeled nights, which matters when expert scoring time is limited, and they are cheap to run computationally, making them suitable for standard hospital hardware or portable devices.

YASA is a widely used example of this feature-based approach. It combines interpretable spectral and statistical EEG features with a tree-based gradient boosting classifier and reaches human-level accuracy and inter-scorer agreement on several public datasets (Vallat and Walker, 2021). Because both the features and the trees are transparent, users can inspect, for a given epoch,

which quantities (for instance a drop in *alpha* power or a rise in *delta* power) drove the predicted stage. At the same time, fixed feature sets make such systems sensitive to montage and hardware differences: performance can drop when moving to a new cohort without re-tuning, and difficult boundaries such as Wake versus N1 often remain challenging (Craik et al., 2019; Zhang et al., 2024). These limitations motivated the shift toward end-to-end deep models that learn features directly from raw PSG.

2.2 Deep Architectures for PSG

As sleep staging datasets grew in size and diversity, deep learning models that learn features directly from raw signals became the state of the art (SOTA). These models aim to capture both the per-epoch signal patterns and the sequential context of sleep stage transitions. We review two main classes of deep architectures, namely convolutional/recurrent models and transformer-based attention models. Throughout, we note how these designs affect generalization across cohorts, computational efficiency, and interpretability.

2.2.1 CNN and CNN-RNN Baselines

Convolutional neural networks (CNNs) can learn time-frequency patterns from raw EEG or multi-sensor PSG signals, while recurrent or other sequence modules model the temporal dependencies across consecutive epochs. A representative example is *DeepSleepNet* (Supratak et al., 2017), which uses multi-scale convolutional feature extractors followed by a bidirectional long short-term memory (LSTM) layer to capture transitions between sleep stages. Supratak et al. (2017) report that DeepSleepNet achieved strong performance on public datasets (MASS and Sleep-EDF) without relying on any handcrafted features. Subsequent studies confirmed the value of incorporating more than just EEG. For instance, adding EOG and EMG channels alongside EEG led to measurable gains in accuracy (Chambon et al., 2018). This makes intuitive sense, as certain stages (like REM sleep) are easier to identify with eye movement and muscle tone information in addition to brain waves (Chambon et al., 2018). Another line of work focused on compact CNN designs that are parameter-efficient. For example, EEGNet is a lightweight CNN architecture originally developed for brain-computer interface signals. It achieves competitive accuracy on EEG tasks with far fewer parameters than typical deep models, which is valuable for real-time or mobile deployment (Lawhern et al., 2018).

In sleep staging, fully convolutional models have also been proposed to capture long temporal contexts without recurrence. U-Time and U-Sleep are two such architectures that adopt an encoder-decoder convolutional network (inspired by U-Net in vision) to label each epoch in a sequence (Perslev et al., 2019, 2021). Despite having no recurrent layers, these models use deep hierarchies and wide receptive fields to effectively model the sequence structure of sleep. Notably, Perslev et al. (2021) showed that U-Sleep trained on a large multi-dataset corpus generalized well to entirely different cohorts without fine-tuning. Overall, CNN-based and CNN+RNN hybrid models set strong baselines for sleep staging, especially in settings with limited computational resources or extremely long recordings (Supratak et al., 2017; Perslev et al., 2021). They establish a solid reference point in accuracy, against which newer architectures can be compared.

2.2.2 Transformers and Attention

The architectures above rely on local receptive fields or recurrent processing to aggregate context, which can make it challenging to capture very long-range dependencies. Transformer-based

models address this limitation by using self-attention mechanisms that can relate widely separated time points within a night's sleep and reveal which parts of the sequence the model considers most relevant through their attention weights (Phan et al., 2022).

Phan et al. (2022) introduced SleepTransformer, which applies a transformer encoder to sequences of EEG epochs and achieves sleep staging performance comparable to recurrent networks. The authors also show how visualizing attention weights highlights epochs that influenced the stage decision. Beyond single-channel EEG Dai et al. (2023) proposed a multi-channel transformer that operates on time-frequency representations from each channel and then fuses the information, leading to improved accuracy on public PSG datasets. In another recent development, Guo et al. (2024) present *FlexSleepTransformer*, a transformer model explicitly designed to handle variable input channel configurations. Rather than being tied to a single fixed PSG montage, FlexSleepTransformer can be trained on datasets with different channel sets and does not rely on a pre-specified channel order, making it more robust to differences in recording setups across clinics (Guo et al., 2024). Channel-flexible and permutation-invariant designs are discussed in more detail in Section 2.2.3. Other groups have explored adapting vision transformer ideas to sleep data as well. For example, the *Sleep ViT* model by Peng et al. (2023) tokenizes time-frequency representations of signals in a similar way to image patches, then applies a pure transformer encoder for staging.

While attention-based models are powerful in their ability to integrate information over long sequences and multiple channels, they also bring new challenges. In particular, transformers can be memory- and data-hungry: training them effectively may require very large datasets and careful consideration of sequence length (since self-attention complexity grows with sequence size) (Phan et al., 2019; Dai et al., 2023).

2.2.3 Permutation-Invariant Channel Modeling

PSG sensor montages differ across hospitals and studies. Earlier models for sleep staging typically assume a fixed set of input channels, and the network architecture is built around that configuration (Phan et al., 2022; Dai et al., 2023). A principled alternative is to treat the available input channels as *unordered set* and design the model to be permutation-invariant to channel ordering. One way to achieve this is by learning per-channel embeddings and then aggregating them with order-agnostic operations such as pooling or attention. Lee et al. (2019) provide a general blueprint in their *Set Transformer*, which uses a pooling-by-multihead-attention mechanism to capture interactions among elements of a set without assuming any particular sequence. Recent sleep studies have started to embrace this idea. For example Thapa et al. (2025) train a channel-flexible multimodal encoder across a heterogeneous PSG dataset, allowing the model to adapt seamlessly to site-specific montages during fine-tuning. Their approach, which introduces the *SleepFM* foundation model, shows that a single network can learn from EEG, EOG, EMG, and respiratory signals combined, even if some recordings lack one or two of those modalities.

2.3 Self-Supervised Learning

Creating expertly labeled data at scale is costly and time-consuming, which limits the amount of supervised data available for training. Self-supervised learning (SSL) addresses this by learning useful representations from large amounts for *unlabeled* physiological data. The learned encoder can then be fine-tuned or probed on specific tasks with relatively few labels. We focus here on two families of SSL approaches that have been applied to sleep data: contrastive learning and autoencoding (reconstruction-based) objectives.

2.3.1 Contrastive Learning

Contrastive methods learn by discriminating between different views or segments of the data. Typically, two correlated views of the same underlying signal (positives) are defined through augmentations or by drawing adjacent time windows, and compares them against other, unrelated signals (negatives). The training objective (an InfoNCE loss) encourages the model to produce similar embeddings for positive pairs and dissimilar embeddings for non-matching pairs. In EEG Banville et al. (2019) were among the first to show that contrastive SSL can substantially improve downstream performance, especially when labeled data are scarce or noisy. Subsequent works have tailored the contrastive paradigm to time-series data. For example Eldele et al. (2021) introduced TS-TCC, which constructs weak and strong augmented views and learns with two contrastive modules on top of a CNN encoder, yielding strong results on various sensor dataset. Similarly, Yue et al. (2022) proposed TS2Vec, a framework that builds a hierarchy of contextualized representations through contrastive learning at multiple time scales, further boosting performance on sequential data tasks.

Beyond these generic approaches, domain-informed augmentations can make contrastive learning even more effective. In the context of electrocardiography Kiyasseh et al. (2021) showed that incorporating physiological transformations into the contrastive pipeline led to pretrained encoders that significantly improved cardiac abnormality detection. This finding suggests that carefully chosen augmentations for PSG could likewise enhance clinically relevant features. Indeed, very recent work by Thapa et al. (2025) applied a contrastive, multimodal SSL scheme to whole-night PSG recordings. Their approach created positive pairs by taking synchronized multichannel segments from the same patient night (and negatives from different patients), while also using modality dropout (*i. e.* sometimes leaving out one modality) to ensure the encoder learns to handle missing channels. The resulting pretrained model, when transferred to tasks like sleep staging and even predicting health outcomes, outperformed models trained from scratch and was robust across large, heterogeneous cohorts. This evidence underlines that contrastive pretraining can yield general-purpose sleep representations, especially when the method respects the structure of physiological signals.

2.3.2 Masked Autoencoding

Another major branch of self-supervision is based on reconstructing missing or corrupted portions of the input signal. In masked reconstruction, the model is presented with an input where some fraction of the data has been removed or masked, and it learns to predict the missing content. The intuition is that to fill in the gaps, the model must capture meaningful patterns in the data.

Masked autoencoders (MAE), first developed for images (He et al., 2022), have been adapted to one-dimensional time-series including EEG and PSG. Key design choices in these methods include the masking ratio (how much of the signal is dropped) and the patch size or granularity of masking. For example, Cai and Zeng (2024) introduce an EEG Transformer with an MAE objective and find that masking around 30% of the input with patch-wise dropout yields a robust representation for downstream tasks. A challenge particular to multichannel sleep data is how to account for the relationships between channels when masking (Fu et al., 2024). One strategy is to incorporate knowledge of the electrode layout or to learn an adjacency graph for channels, then mask in a way that respect these structures. Fu et al. (2024) take this approach by combining graph neural network ideas with masked EEG pretraining, which improves cross-channel generalization (*i. e.* the model can better handle an electrode montage it was not explicitly trained on). On the scalability front, Ogg and Coon (2024) demonstrate that masked autoencoding can be trained on very large sleep EEG datasets. They pretrained a sleep EEG model on thousands of

hours of data using a combination of masked and unmasked segments, indicating the feasibility of learning high-capacity sleep encoders without any labels.

Variational autoencoders (VAEs) have also been explored as another way to learn from unlabeled EEG. Zhao et al. (2024) propose VAE-EEG, which learns a probabilistic latent space of EEG signals and could be applied to sleep data as well. It remains an open question how these reconstruction-based pretraining methods compare to contrastive methods for PSG; each approach has its strengths, and they are not mutually exclusive. In our work, we prioritize contrastive objectives with modality-aware dropout (inspired by the techniques above) and do not pursue masked pretext tasks, but we summarize the autoencoding literature here to provide context and completeness.

2.4 Foundation Models for Sleep

Foundation models (FMs) refer to large models trained on broad data sources that can be adapted to a wide range of downstream tasks. In the context of sleep, a foundation model typically means a deep network pretrained on diverse PSG data (potentially from many cohorts, with multiple signal types), which can then be fine-tuned for tasks such as sleep staging, event detection, or health outcome prediction. The goal is to learn general-purpose representation of sleep physiology that is not tied to any single dataset.

A pioneering example is *SleepFM* by Thapa et al. (2024), which introduced a multimodal PSG encoder encompassing brain (EEG), eye (EOG), muscle (EMG), and cardiac/respiratory signals. SleepFM was pretrained using a contrastive learning objective similar to those discussed above, with an emphasis on aligning representations across modalities and accommodating different channel combinations. The authors designed the model to be flexible with respect to input montage, allowing it to accept various subsets of the four modality types during fine-tuning. Remarkably, the embeddings from this foundation model proved effective for a variety of tasks. Not only could they be fine-tuned to perform sleep staging comparably to SOTA dedicated models, but they also transferred to clinical prediction tasks (predicting cardiovascular risk factors) with minimal adaptation. Building on this concept, Thapa et al. (2025) scaled up the training of such a model to over 500'000 hours of PSG data drawn from multiple cohorts. This massive training set yielded a model with exceptional generalization ability: it achieved high agreement when evaluated on external datasets, and it enabled accurate prediction of health outcomes like mortality, incident heart failure, and strokes using only simple linear classifiers on the learned features. These results underscore that large-scale pretraining can capture clinically relevant information that goes beyond the immediate task of staging.

Independent studies have arrived at similar conclusions. Fox et al. (2025) train full-night multichannel PSG transformers in a self-supervised manner and report hat the learned representations transfer well to sleep staging across different cohorts. There appear to be common patterns in sleep EEG and related signals that a sufficiently large model can learn and then recognize in new subjects and settings. At the same time, researchers are exploring more compact architectures to make foundation models practical. Dimofte et al. (2025) introduce a model called Compact Encoder for Representations of Brain Oscillations (CEReBrO) that uses an alternating attention mechanism to separately handle temporal dynamics within each EEG channel and inter-channel correlations across electrodes in an efficient way. This design substantially reduces memory and computation requirements (achieving roughly a two-fold speedup) while maintaining strong performance in transfer learning scenarios. Such innovations are important for translating foundation models from research to real-world clinical deployment, where GPU resources may be limited.

2.5 Downstream Tasks for PSG Foundation Models

Foundation models for PSG are typically evaluated by transferring them to supervised *downstream tasks*. A single encoder pretrained on large PSG corpora — such as SleepFM — can be adapted to a range of tasks, such as sleep staging and the assessment of sleep-disordered breathing (Stephansen et al., 2018; Thapa et al., 2024, 2025). In this work we focus on two tasks that are central in clinical sleep medicine and well-represented in our data: sleep staging and apnea detection.

2.5.1 Sleep Staging

In this thesis, sleep stages are scored according to the AASM rules, which assign one of five stages (Wake, N1, N2, N3, REM) to each 30-second PSG epoch based on EEG, EOG, and EMG patterns (Berry et al., 2017). Even under these standardized criteria, human scorers only reach moderate agreement on average, with particularly low consensus for the N1 stage (Lee et al., 2022). This variability makes sleep staging both a clinically important and methodologically challenging target for automation.

Traditional automated staging systems rely on hand-crafted features and classical classifiers (Craik et al., 2019), whereas modern approaches use deep CNN/RNN architectures and Transformers that operate directly on raw PSG and can reach human-level performance on benchmark datasets (Supratak et al., 2017; Brunini, 2023). Foundation models such as SleepFM extend this trend by pretraining multimodal encoders on large PSG corpora and then fine-tuning them for sleep staging (Thapa et al., 2025).

2.5.2 Apnea Detection

Sleep-disordered breathing, particularly obstructive sleep apnea, is characterized by repeated reductions or cessations of airflow during sleep (Carmel, 2023). Clinically, events are scored based on changes in airflow, respiratory effort, and blood oxygen saturation, and summarized by indices such as the apnea-hypopnea index (AHI) (Berry et al., 2017). We detail these definitions in Section 1.2.3. From a machine-learning perspective, apnea detection is typically formulated either as a per-window event detection task (apnea vs nonapnea) or as a severity estimation task, such as predicting the AHI.

Traditional automated approaches typically extract hand-crafted features from respiratory and oxygen-saturation signals and feed them into relatively simple classifiers tuned to be sensitive to apnea events (Moret-Bonillo et al., 2014). More recent deep-learning work uses CNNs and Transformer-based architectures on raw or lightly processed respiratory and related signals to perform apnea assessment and severity estimation (Hu et al., 2025). Several studies have shown that Transformers are effective for apnea screening and severity assessment from cardiorespiratory signals: for example, Chen et al. (2025) use a Transformer with learnable positional encodings on SpO_2 for oximetry-based OSA diagnosis, and Zhang et al. (2025) propose a multimodal multiscale Transformer that fuses ECG and SpO_2 for OSA detection and AHI estimation, outperforming earlier deep-learning baselines. As noted above, the same multimodal foundation model encoders can be fine-tuned for apnea related tasks, exploiting shared structures between EEG activity and respiratory dynamics (Thapa et al., 2024, 2025).

Chapter 3

Background

This chapter collects technical background that complements the related work in Chapter 2 and builds the technical background for our Methods (Chapter 5) and Experiments (Chapter 6). We summarize the attention-based tokenization and Transformer components that our foundation model builds on, followed by the evaluation metrics and loss functions used throughout the thesis. A complete nomenclature of symbols used in this thesis is given in Appendix A.

3.1 Attention and Tokenization Basics

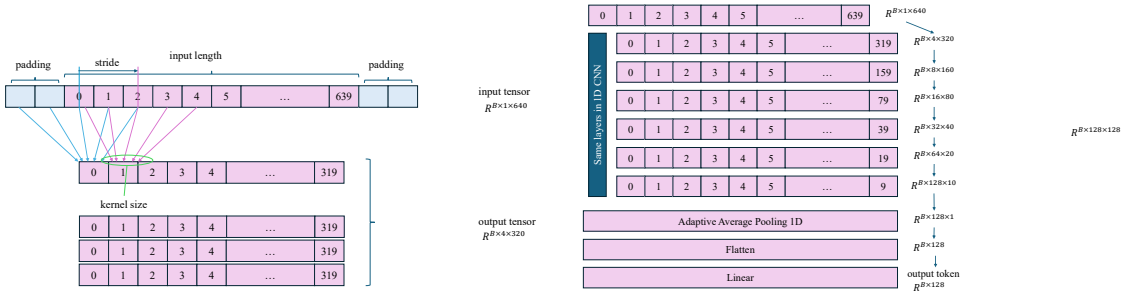
We focus on conceptual aspects needed to understand the architecture. The implementation details closely follow the designs cited in the related work and are not novel.

Modern Transformer models for time-series first compress long raw sequences into shorter sequences of *tokens* before applying self-attention. In particular, a 1D convolutional front-end can tokenize a multichannel signal like PSG into patch-level embeddings, which a Transformer encoder then processes. We review this tokenization process and the core Transformer operations (scaled dot-product attention, multi-head attention, feed-forward layers, and normalization) that follow. We also discuss how to encode positions in time, maintain permutation invariance across channels, and choose pooling strategies for sequence summarization.

3.1.1 1D CNN Tokenization for Patch Embeddings

Transformer encoders operate most effectively on sequences of compact, informative tokens rather than extremely long raw waveforms. In time-series applications (including PSG), feeding millions of raw samples into a Transformer would be computationally prohibitive. The core operation of a Transformer, self-attention (introduced in Section 3.1.2), compares each token with every other token in the sequence, and thus its memory and compute cost grow quadratically of the sequence length (Vaswani et al., 2017). A practical approach is therefore to *first* compress short windows of the signal using a lightweight one-dimensional convolutional encoder and *then* apply a Transformer only to the much shorter sequence of encoded tokens. For example, wav2vec 2.0 uses a strided 1D CNN front-end to produce latent token embeddings from raw audio before applying Transformer layers (Baevski et al., 2020).

Formally, let $X \in \mathbb{R}^{B\times C\times T}$ denote a mini-batch of multichannel sequences, where B is the batch size, C is the number of channels, and T is the number of time samples per sequence (Dosovitskiy et al., 2021). We divide the time axis of each sequence into non-overlapping patches of



(a) Patch extraction with kernel size 5, stride 2, and padding 2 from a 640-sample segment.

(b) Stacked 1D convolutions that compress each patch into a fixed-size token embedding.

Figure 3.1: 1D CNN TOKENIZER FOR PATCH EMBEDDINGS. *Simplified visualization of the convolutional tokenizer used in SleepFM. Subfigure (a) shows how a 640-sample input segment is covered by receptive fields using padding* $= 2$*, stride* $= 2$*, and kernel size* $= 5$*. Subfigure (b) illustrates the subsequent stack of six strided 1D convolutions that reduce temporal length and increase channel depth, followed by pooling and a linear layer to obtain a compact token representation of each patch (Thapa et al., 2025).*

length P, yielding

$$S = \left\lfloor \frac{T}{P} \right\rfloor \tag{3.1}$$

where any trailing segment shorter than P is discarded. This follows the setup used in Thapa et al. (2025): allowing a variable-length remainder would otherwise require special handling, whereas dropping it ensures all patches have consistent shape.

For each batch element $b \in \{1, \ldots, B\}$, channel $c \in \{1, \ldots, C\}$, and patch index $s \in \{0, \ldots, S-1\}$, we extract the s-th patch as

$$x_{b,c,s} \;=\; X_{b,c,\,sP:(s+1)P} \;\in\; \mathbb{R}^P \tag{3.2}$$

the raw P-dimensional waveform segment for that channel and time window (similar to patch extraction in Vision Transformers) (Dosovitskiy et al., 2021). A 1D convolutional tokenizer $f_\theta : \mathbb{R}^P \to \mathbb{R}^E$ that is shared across all batches, channels, and patch indices then maps each patch $x_{b,c,s}$ to an E-dimensional embedding (or "token"), producing a compact vector representation:

$$z_{b,c,s} \;=\; f_\theta\big(x_{b,c,s}\big) \;\in\; \mathbb{R}^E \tag{3.3}$$

where E is the token embedding dimension. Stacking these tokens yields a tensor $Z^{\text{ch}} \in \mathbb{R}^{B\times C\times S\times E}$.

Figure 3.1 illustrates, in simplified form, how the 1D CNN tokenizer acts on a 640-sample segment (5-seconds at 128Hz, *i.e.* $P = 640$). A convolutional filter with kernel size of 5 (*i.e.* we are looking at at local window of five consecutive time samples) slides across the input with stride of 2 (*i.e.* it moves two samples between successive convolution positions) and zero-padding of 2 samples on each side (*i.e.* we append two dummy zeros at the beginning and end of the sequence so that boundary positions are also covered in full 5-sample window; light blue sample in Figure 3.1), so that each output element depends on a local window of five input samples while stride-2 convolutions halves the temporal length at every layer.

In our implementation, following SleepFM (Thapa et al., 2025), f_θ is a stack of six such 1D convolutional layers: as the signal passes through the stack, the temporal dimension shrinks (from 640 samples to 10), the channel dimension grows (from 1 up to 128), and a global average pooling over the remaining temporal axis (`AdaptiveAvgPool1d(1)`) collapses each patch to a single

128-dimensional vector. A final linear layer then maps this vector to the token dimension E, which we also use as the Transformer model dimension (corresponding to d_{model} in the original Transformer notation). This convolutional front-end design is similar to those used in raw-audio Transformers and other tokenization-based models, where local convolutions create compact token embeddings that capture nearby structure (Baevski et al., 2020; Thapa et al., 2025).

By mapping each P-sample patch to a single embedding, the tokenizer reduces a long waveform to a much shorter sequence of tokens. As discussed in Section 3.1.2 below, the computational and memory cost of self-attention grows approximately quadratically to the token sequence length. The choice of the patch length P (and thus the number of tokens S via Eq. (3.1)) therefore trades off temporal resolution against computational efficiency: larger patches yield fewer, cheaper tokens but coarser timing, whereas smaller patches retain finer detail at higher computational cost.

3.1.2 Scaled Dot-Product and Multi-Head Self-Attention

After convolutional tokenization (Section 3.1.1), each example is represented as sequence of vector tokens. The Transformer encoder updates these tokens using *self-attention*, which lets each token attend to all others and build context-aware representations. In this subsection, we summarize scaled dot-product and multi-head self-attention mechanisms introduced by Vaswani et al. (2017), since they form the core of our temporal encoder.

Let $Z \in \mathbb{R}^{B\times S\times E}$ be a batch of token sequences, where B is the batch size, S is the sequence length (number of temporal patches), and E is the embedding dimension of each token. In the notation of Section 3.1.1, S is exactly the number of patches per epoch after aggregating the C channel-wise tokens at each patch index into a single token (see Section 3.1.4 for channel aggregation).

Linear Projections (Queries, Keys, Values)

Following Vaswani et al. (2017), the input token sequence Z is linearly projected to queries (Q), keys (K), and values (V) using learned matrices:

$$Q = Z\,W_Q \qquad K = Z\,W_K \qquad V = Z\,W_V \tag{3.4}$$

where $W_Q, W_K \in \mathbb{R}^{E\times d_k}$ and $W_V \in \mathbb{R}^{E\times d_v}$. Here d_k and d_v denote the query/key and value dimensions (Vaswani et al., 2017).

Scaled Dot-Product Attention

Given the projected queries, keys, and values, the attention operation computes a weighted sum of the value vectors for each query. Vaswani et al. (2017) define the weights by the dot products between each query and all keys, scaled by $\frac{1}{\sqrt{d_k}}$ (to counteract the effect of high dimensionality), and then normalize with a softmax:

$$\text{Attention}(Q, K, V) = \text{softmax}\Big(\frac{QK^\top}{\sqrt{d_k}}\Big)V \tag{3.5}$$

In practice, attention is almost always used with a binary mask that indicates which positions are valid to attend to. Let $m_{ij} \in \{0, 1\}$ denote whether query position i is allowed to attend to key position j (*e.g.* $m_{ij} = 0$ for padding tokens, *i.e.* dummy positions added to match the sequence length, or for future time steps in autoregressive models, and $m_{ij} = 1$ for valid tokens). We convert this to an additive mask M with $M_{ij} = 0$ when $m_{ij} = 1$ and $M_{ij} = -\infty$ when $m_{ij} = 0$,

and add it to the unnormalized attention scores. The resulting masked attention takes the form (Shi et al., 2021):

$$\text{Attention}(Q, K, V) = \text{softmax}\Big(\frac{QK^\top + M}{\sqrt{d_k}}\Big)V \tag{3.6}$$

By adding a large negative value before the softmax, we force the corresponding probabilities to zero, without changing the relative scale of the allowed logits.

Multi-Head Self-Attention

Instead of using a single set of projections W_Q, W_K, W_V, Transformers employ H parallel attention "heads" to jointly attend to information from different representation subspaces. As introduced by Vaswani et al. (2017), for each head $h = 1, \ldots, H$, let $W_Q^{(h)}, W_K^{(h)}, W_V^{(h)}$ be separate learned projections, and the h-th head computes its output as:

$$\text{head}^{(h)} = \text{Attention}\Big(ZW_Q^{(h)}, ZW_K^{(h)}, ZW_V^{(h)}\Big) \tag{3.7}$$

i.e. it applies Eq. (3.5) to $Q = ZW_Q^{(h)}$, $K = ZW_K^{(h)}$, and $V = ZW_V^{(h)}$. Each head output $\text{head}^{(h)}$ has shape $B \times S \times d_v$, where d_v is the per-head value dimension. In our implementation we choose $d_v = E/H$, so that concatenating all heads yields an embedding of size E. The multi-head attention (MHA) module then concatenates all head outputs along the last dimension (resulting in a tensor of shape $B \times S \times (Hd_v)$) and, as in the single-head case, projects the result back to dimension E with another learned matrix $W_O \in \mathbb{R}^{Hd_v \times E}$:

$$\text{MHA}(Z) = \big[\text{head}^{(1)}, \quad \cdots, \quad \text{head}^{(H)}\big]W_O \tag{3.8}$$

The use of multiple attention heads allows the model to capture different types of relationships or features from the token sequence in parallel, as each head can focus on a different subspace or scale of interactions (Vaswani et al., 2017).

Transformer Encoder Block

A standard Transformer encoder layer consists of two sub-layers: a multi-head self-attention block and a position-wise feed-forward network (FFN, two-layer MLP) block, each wrapped in a residual connection with layer normalization.

Following Ba et al. (2016), layer normalization normalizes each token's feature vector to have zero mean and unit variance, then applies a learned scale and bias. For an input token vector $x \in \mathbb{R}^E$ (*e.g.* one token embedding $Z_{b,s,:}$), we write:

$$\text{LN}(x) = \frac{x - \mu(x)}{\sqrt{\sigma^2(x) + \varepsilon}} \odot \gamma + \beta \tag{3.9}$$

where $\mu(x)$ and $\sigma^2(x)$ are the mean and variance of the E elements of x, $\varepsilon > 0$ is a small constant for numerical stability, $\odot$ denotes element-wise multiplication, and $\gamma, \beta \in \mathbb{R}^E$ are learned gain and bias parameters. Thus, LN standardizes the features of each token and then shifts/scales them with learnable parameters.

As specified by Vaswani et al. (2017), the position-wise FFN applies an identical two-layer perceptron to each token in the sequence. If $Z \in \mathbb{R}^{B \times S \times E}$ is the input to the FFN (with $Z_{b,s,:}$ the s-th token in the b-th sequence), the output is:

$$\text{FFN}(Z) = \phi\big(ZW_1 + b_1\big)W_2 + b_2 \tag{3.10}$$

where $W_1 \in \mathbb{R}^{E \times d_{\text{hid}}}$ and $W_2 \in \mathbb{R}^{d_{\text{hid}} \times E}$ are weight matrices (with d_{hid} the FFN's hidden-layer size), b_1, b_2 are biases, and ϕ is a non-linear activation function such as ReLU (Vaswani et al., 2017). The FFN maps each token from E to a hidden dimension d_{hid} through ϕ, then projects back to E. Dropout regularization is applied to the intermediate activations and the final output of the FFN (Vaswani et al., 2017).

Using these components, we can write the transformations in a single Transformer encoder layer. In the post-norm formulation of Vaswani et al. (2017), layer normalization is applied *after* adding the residual connection:

$$U^{\ell} = \text{LN}\Big(Z^{\ell} + \text{MHA}(Z^{\ell})\Big) \tag{3.11}$$

where U denotes the intermediate result. Next, U goes through the FFN sub-layer, is added back to U, and normalized to produce the output:

$$Z^{\ell+1} = \text{LN}\Big(U^{\ell} + \text{FFN}(U^{\ell})\Big) \tag{3.12}$$

An important architectural choice is whether to apply layer normalization after the residual connection (post-norm, as in Eqs. (3.11)-(3.12)) or before the attention and feed-forward sub-layers (pre-norm, *e. g.* $\text{MHA}(\text{LN}(Z^{\ell})) + Z^{\ell}$ and $\text{FFN}(\text{LN}(U^{\ell})) + U^{\ell}$). The original Transformer uses post-norm, while modern deep Transformers often adopt pre-norm for improved stability (Xiong et al., 2020). The approach stacks multiple such layers to form the full Transformer encoder: the residual pathways help preserve information, normalization keeps activations well-scaled, self-attention integrates information across the entire token sequence, and the FFN enriches each token representation based on that integrated context.

3.1.3 Temporal Positional Encodings

After tokenization (Section 3.1.1), the Transformer encoder processes a sequence of S tokens, represented as $Z \in \mathbb{R}^{S \times E}$. Self-attention alone is invariant to token order, so we must inject information about *temporal positions* for the model to know the sequence ordering (Irie, 2025). We therefore add explicit temporal positional encodings only along the time (sequence) dimension. As introduced in Section 3.1.1, each PSG example has C channels (*e. g.* EEG, EOG, EMG, respiratory); how we aggregate information across channels in a permutation-invariant way is described in Section 3.1.4.

A standard approach for encoding token positions is to use sinusoidal position embeddings that are added to the token representations (Vaswani et al., 2017). Before we apply self-attention, we offset the token vector in Z by a fixed positional encoding:

$$\widetilde{Z} = Z + P_{\text{pos}} \tag{3.13}$$

where $P_{\text{pos}} \in \mathbb{R}^{S \times E}$ is a matrix of positional encodings. The token index represents the time step $s = 0, 1, \ldots, S-1$ and i represents the feature index $i = 0, 1, \ldots, E-1$ (Vaswani et al., 2017).

$$P_{\text{pos},\, s,\, 2i} = \sin\Big(s/10000^{2i/E}\Big) \qquad P_{\text{pos},\, s,\, 2i+1} = \cos\Big(s/10000^{2i/E}\Big) \tag{3.14}$$

This deterministic function produces alternating sine and cosine waves that encode absolute positions without any learned parameters, and it is commonly used in Transformers for sequences (Vaswani et al., 2017). Alternatives such as learned relative positions by Shaw et al. (2018) or rotary positional embeddings by Su et al. (2024) exist, but we do not explore them here.

Because channel order should not matter, we leave the channel axis without any positional structure and rely instead on permutation-invariant channel pooling (Section 3.1.4) to aggregate information across sensors.

3.1.4 Permutation Invariance over Channels (Set Attention and Pooling)

After tokenization (Section 3.1.1), the 1D CNN produces an embedding $z_{b,c,s} \in \mathbb{R}^E$ for each batch element b, channel c and patch index s (see Eq. (3.2)). For a fixed example b and patch index s, collecting over channels gives a set of C token embeddings:

$$\{x_1, \ldots, x_C\} \;=\; \{z_{b,1,s}, \ldots, z_{b,C,s}\} \tag{3.15}$$

that all describe the same time window but originate from different sensors. Since the channels have no canonical order (and different datasets may use different channel montages), the aggregation function over channels should be permutation-invariant.

One general architecture for such a set function is DeepSets (Zaheer et al., 2017), which can be written as:

$$f(x_1, \ldots, x_C; \theta) = \rho\Big(\sum_{i=1}^{C} \phi(x_i)\Big) \tag{3.16}$$

where ϕ and ρ are learnable transformations (*e. g.* small neural networks applied before and after summation). Because summation is order-agnostic, $f(x_1, \ldots, x_C)$ is invariant to any permutation of the inputs (Zaheer et al., 2017).

When some channels are missing, it is convenient to work with an explicit binary mask and a normalized pooling operation. A simple special case is the masked mean. If each channel i has an indicator $m_i \in \{0, 1\}$ denoting whether that channel is present ($m_i = 1$) or absent ($m_i = 0$) for a given example, we define

$$f_{\text{mean}}\big(\{x_i, m_i\}\big) = \frac{\sum_{i=1}^{C} m_i x_i}{\max\{1, \sum_{i=1}^{C} m_i\}} \tag{3.17}$$

which averages the available channel embeddings (or returns 0 if none are present). Reordering the channels leaves both the numerator and denominator unchanged, so f_{mean} is permutation-invariant in the pairs (x_i, m_i).

Beyond simple pooling, the *Set Transformer* uses attention to model relationships among set elements while still ignoring their order (Lee et al., 2019). The idea is to first apply self-attention layer on the set $\{x_1, \ldots, x_C\}$ so that elements can exchange information (this step is permutation-equivariant: permuting the inputs permutes the outputs in the same way), and the apply a pooling-by-multi-head-attention (PMA) layer to obtain a fixed-size summary.

Concretely, Lee et al. (2019) propose PMA as follows: let $Z^{\text{set}} \in \mathbb{R}^{C \times E}$ contain the C channel embeddings (one row per channel), and let $Q_{\text{pool}} \in \mathbb{R}^{R \times E}$ be a small set of learnable query vectors. The PMA operation attends from these R queries to the set Z^{set}:

$$\text{PMA}_R(Z^{\text{set}}) = \text{Attention}\big(Q_{\text{pool}}, Z^{\text{set}}, Z^{\text{set}}\big) \tag{3.18}$$

where $\text{Attention}(\cdot)$ denotes the scaled dot-product attention of Eq. (3.5) applied with $Q = Q_{\text{pool}}$, $K = Z^{\text{set}}$, and $V = Z^{\text{set}}$. Each query in Q_{pool} learns to collect information from all elements of Z^{set}, producing R summary vectors of size E. Because the attention treats Z^{set} as an unlabeled collection of values and does not rely on channel indices, the result is permutation-invariant with respect to channel order (Lee et al., 2019). If some channels are missing for an example, we pass the corresponding mask to the attention (3.6) so that those rows of Z^{set} effectively contribute nothing (Vaswani et al., 2017; Lee et al., 2019).

In summary, at each patch index s we first obtain a set of C channel embeddings. We then collapse this set into a single summary token $z_{b,s}$ (or a fixed small set of tokens) using a permutation-invariant function such as the masked mean in Eq. (3.17) or the PMA layer in Eq. (3.18). Stacking

the summary tokens $(z_{b,0}, \ldots, z_{b,S-1})$ for all examples yields a tensor $Z \in \mathbb{R}^{B\times S\times E}$ of fused tokens that is fed into the temporal self-attention encoder. Because we pool across channels before this stage, the encoder focuses on *when* events happen (via temporal position encodings) while remaining agnostic to the arbitrary order of channels across different montages.

3.1.5 Attention Pooling vs CLS vs PMA (Design Trade-offs)

After producing a sequence of tokens (Section 3.1.1) and processing them with a Transformer encoder (Section 3.1.2), we need a fixed-size representation for tasks such as classification. Several standard strategies can be used to aggregate a variable-length sequence into a single vector: attention pooling with a learned query, adding a special learned token (usually denoted as [CLS]) whose final embedding serves as a global summary, or pooling by multi-head attention (PMA) as introduced in Section 3.1.4. These approaches differ in how they combine information and in the inductive biases they introduce.

Attention Pooling

Attention pooling with a single query is a straightforward learnable pooling method. We introduce a *learned* query vector $q \in \mathbb{R}^E$ that is a model parameter (initialized randomly and optimized together with the rest of the network), and let it attend over the entire sequence of S token embeddings. Formally, let $Z \in \mathbb{R}^{S\times E}$ stack the S token embeddings row-wise. Using the scaled dot-product attention from Section 3.1.2, we compute attention weights and a weighted sum of tokens as follows (Vaswani et al., 2017):

$$z_{\text{pool}} = \text{softmax}\Big(\frac{q\,Z^\top}{\sqrt{E}}\Big)\,Z \in \mathbb{R}^{1\times E}. \tag{3.19}$$

Here the softmax term produces attention weights over the S tokens, and their weighted sum yields z_{pool}, a pooled sequence representation. In practice, z_{pool} is then passed to a small task-specific head (for example, a fully connected layer) to produce the final prediction, so it should be viewed as an intermediate summary rather than the network's ultimate output.

Conceptually, this differs from the [CLS] approach below in that the Transformer encoder first processes the sequence as usual, and the extra attention pooling with q is applied only once on top of the final token representations.

[CLS] Token

A popular alternative in NLP and vision is to prepend a learned special token (often denoted as [CLS]) to the sequence and to use its final embedding as the summary (Devlin et al., 2019; Dosovitskiy et al., 2021). Concretely, a learned [CLS] embedding is added at the start of the input sequence, the full sequence (including [CLS]) is fed through all Transformer layers, and the output embedding of the [CLS] position is taken as the sequence-level representation.

In this setup, no additional pooling layer is introduced beyond the self-attention blocks themselves: at each layer the [CLS] token attends to the other tokens and aggregates information via the standard attention mechanism. This differs from attention pooling with a separate query q, where pooling happens in an explicit extra step after the encoder. When combined with standard positional encodings, the [CLS] embedding is sensitive to the order of tokens in the sequence (as in BERT and ViT), which is appropriate for time-series or text but not ideal for unordered sets unless positional information is omitted.

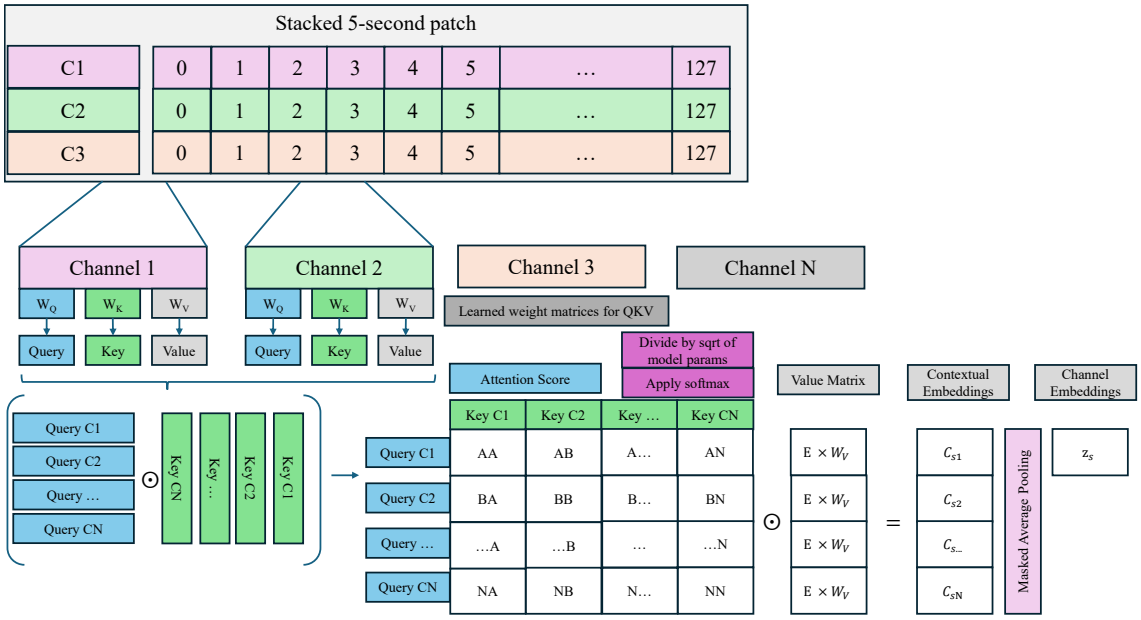


Figure 3.2: CHANNEL-WISE ATTENTION OVER CHANNELS. *Channel-wise attention over channels. Each input channel is tokenized independently and projected into query, key, and value vectors. Attention is applied across the set of channels to compute contextual embeddings at each time step, which are then pooled into a single fused token z_s. This operation is permutation-invariant with respect to channel order and enables flexible integration of multimodal inputs. Adapted from Vaswani et al. (2017) and Thapa et al. (2025).*

Pooling by Multi-Head Attention

For unordered sets of tokens, such as the per-time-step channel sets discussed in Section 3.1.4, PMA provides a permutation-invariant, learnable aggregation mechanism (Lee et al., 2019). As defined in Eq. (3.18), PMA introduces a small set of learned query vectors S that attend to the set Z^{set} using the same scaled dot-product attention as in Section 3.1.2. Each query can be interpreted as "asking" for a different aspect of the set's information; attending from S to Z^{set} yields a set of corresponding summary vectors that each combine information from all set elements while remaining invariant to their order.

Figure 3.2 illustrates this procedure: the top panel shows a stacked 5-second patch across channels, the middle panels show per-channel $Q/K/V$ projections and the channel-channel attention matrix, and the rightmost panels show the resulting contextualized channel embeddings and their masked average pooling into the fused token z_s. Because the attention operates on the set of channel embeddings without any positional encodings, the result is invariant to permutations of channel order and naturally handles missing channels via the attention mask (Section 3.1.4) (Lee et al., 2019).

Trade-offs

Each aggregation method has its own strengths. Attention pooling with a single learned query produces a global summary with minimal overhead and no change to the Transformer encoder architecture. The [CLS] token integrates pooling into the encoder itself and has been very successful in language and vision models, where an ordered sequence and positional encodings are assumed. PMA explicitly targets unordered sets, offering permutation-invariant pooling with learnable queries and linear cost in the number of set elements (Lee et al., 2019). In this thesis we adopt the simpler masked mean pooling as illustrated in Figure 3.2. Here, masked mean pooling refers to averaging the token embeddings only over positions marked as valid by a binary mask, so that missing channels or padding tokens do not contribute to the pooled representation.

3.2 SleepFM: Set-then-Sequence Foundation Encoder

The previous sections introduced the building blocks of our models: convolutional tokenization (Section 3.1.1), scaled dot-product attention and Transformer encoders (Section 3.1.2), permutation-invariant channel-set pooling (Section 3.1.4), and attention-based aggregation mechanisms (Section 3.1.5). We now describe how these components are combined in SleepFM, the multimodal PSG foundation encoder proposed by Thapa et al. (2025), which we adopt as the backbone throughout this thesis. The goal of this section is to summarize the published architecture and to fix notations used later in Chapter 5.

Figure 3.3 provides an overview. For each channel the raw signal is split into non-overlapping patches of length P samples. In our experiments $P = 640$, corresponding to 5-seconds at 128Hz. A shared 1D convolutional tokenizer maps each patch to an E-dimensional embedding. At each patch index s the model then forms a set of channel embeddings $\{z_{1,s}, \ldots, z_{C,s}\}$ and applies a small channel-set encoder with masked self-attention across channels followed by permutation-invariant *masked mean* pooling. This yields a single fused token $z_{b,s}$ per patch. The fused tokens $(z_{b,1}, \ldots, z_{b,S})$ form a temporal sequence. The sequence is augmented with positional encodings and processed by a Transformer encoder along the time axis. Finally, mean pooling over the S time steps produces a fixed-size embedding that serves as the input to the downstream prediction heads.

3.2.1 Channel-Set Encoder

Let $X \in \mathbb{R}^{B\times C\times T}$ denote a batch of PSG segments with C channels and T time samples per segment. As in Section 3.1.1 we first divide the time axis into $S = \lfloor T/P \rfloor$ non-overlapping patches. A single convolutional tokenizer f_θ with shared parameters is applied independently to every channel and patch:

$$z_{b,c,s} = f_\theta(x_{b,c,s}) \in \mathbb{R}^E, \tag{3.20}$$

which produces a tensor $Z^{\text{ch}} \in \mathbb{R}^{B\times C\times S\times E}$. In the notation of this section a time step corresponds to one patch index $s \in \{0, \ldots, S-1\}$ within an epoch, rather than a full 30-second epoch.

Recordings from different cohorts can have different subsets of channels. SleepFM therefore keeps a binary mask $m_{b,c} \in \{0, 1\}$ that indicates whether channel c is present for example b. For each patch index s and example b we consider the masked set $\{(z_{b,c,s}, m_{b,c})\}_{c=1}^{C}$ and feed it into a small SetTransformer-style block (Section 3.1.4), implemented as a single Transformer encoder layer across the C channel tokens followed by masked mean pooling over the channels using this mask (Section 3.1.5). Because the self-attention and pooling operate on the set of channels without positional encodings, the channel-set encoder is permutation equivariant before pooling

and permutation invariant after pooling: reordering the channels does not change the fused token $z_{b,s}$.

The output of this step is a sequence of fused tokens $Z \in \mathbb{R}^{B\times S\times E}$, where $Z_{b,s,:}$ summarizes the information available across all present channels during patch s for example b.

3.2.2 Modalities and Channel Sets

In the data pipeline we distinguish four modality groups

$$\mathcal{M} = \{\text{BAS}, \text{RESP}, \text{EKG}, \text{EMG}\} \tag{3.21}$$

where BAS collects all brain-activity signals (EEG and EOG), RESP comprises airflow and respiratory effort channels, EKG contains ECG leads, and EMG covers chin and limb EMG (Chapter 4). In the SleepFM implementation, a global channel inventory (`channel_groups.json`) maps every concrete signal label in the data (C3-A2, F4-M1, nasal pressure, thoracic belt, ...) to exactly one of these modalities and is shared across all cohorts (Thapa et al., 2025).

For each recording and chunk we look up which channels of each modality are actually present, stack them in a fixed modality order along the channel axis, and obtain an input tensor $X \in \mathbb{R}^{B\times C\times T}$ with $C = C_{\text{BAS}} + C_{\text{RESP}} + C_{\text{EKG}} + C_{\text{EMG}}$ for that example. Within each modality the channel order is arbitrary and may differ across cohorts. We do not encode any spatial layout. Configuration entries such as `BAS_CHANNELS=10` and `RESP_CHANNELS=7` define an upper bound C_m per modality. If a recording provides fewer than C_m channels, the remaining slots are zero padded and marked as absent in a binary mask $m_{b,c} \in \{0, 1\}$ that is passed through the encoder.

The backbone itself is modality-agnostic: it only operates on the union of channels and the mask. Consequently, the channel set processed by the set-attention block in Section 3.1.4 and Figure 3.3 is always a *mixture of modalities*, and permutation-invariant pooling ensures that neither the order nor the exact composition of BAS, RESP, EKG, and EMG channels affects the fused token.

3.2.3 Temporal Transformer Encoder and Epoch Representations

The fused tokens Z form an ordered sequence of length S. SleepFM adds sinusoidal positional encodings along the patch index, as introduced in Section 3.1.3, and passes the sequence through a stack of Transformer encoder layers described in Section 3.1.2. In our implementation the temporal encoder has $L_{\text{enc}} = 6$ layers, model dimension $E = 128$, and $H = 8$ attention heads, following Thapa et al. (2025). The output of the final layer is a tensor $H \in \mathbb{R}^{B\times S\times E}$.

We use two derived representations. For self-supervised pretraining we work with the patch-level representation $z_{\text{enc}} \in \mathbb{R}^{B\times S\times E}$ obtained from H (and ℓ_2-normalized) and apply the contrastive objectives from Section 3.3.1 to these embeddings. For downstream tasks we use an epoch-level representation $z_{\text{fm}} \in \mathbb{R}^{B\times E}$ obtained by mean pooling H over the patch dimension.

3.2.4 LSTM Sleep Staging Head

In addition to the foundation encoder, Thapa et al. (2025) introduce a sequence classifier for full-night sleep staging built on top of the epoch-level embeddings z_{fm} from Section 3.2.3. For a given night the encoder produces a sequence of embeddings

$$(z_{\text{fm},1}, \ldots, z_{\text{fm},L_{\text{night}}}), \tag{3.22}$$

one vector per 30-second epoch. The SleepFM downstream head, denoted SleepEventLSTMClassifier, takes this sequence as input and applies a bidirectional LSTM with hidden size $E/2$ per direction, followed by a linear layer that maps each time step to logits over the five AASM stages (W, N1, N2, N3, REM). A softmax yields per-epoch stage probabilities, and training uses cross-entropy loss.

This design combines the local representation learning of the foundation encoder (within each epoch) with longer-range temporal modeling over the night. Because the LSTM is bidirectional, it can use both past and future context when predicting the stage of a given epoch, which is appropriate for an offline scoring scenario.

3.3 Losses and Evaluation Metrics

The following loss functions and evaluation metrics are used consistently in our experiments and in the result plots of Chapters 7 and 8. For self-supervised pretraining we use contrastive objectives that shape a general embedding space.

For downstream classifiers we optimize class-weighted cross-entropy (and, in some experiments, focal loss) and evaluate with confusion matrices and class-sensitive metrics — per-class F1, F1, balanced accuracy, and Cohen's κ for sleep staging (to match SleepFM and handle class imbalance) — as well as AUROC and AUPR for apnea detection.

3.3.1 Contrastive Loss

Many modern self-supervised and supervised representation-learning methods use a contrastive loss to learn transferable features without heavy labeling (Khosla et al., 2020; Jaiswal et al., 2021). The idea is to shape the embedding space so that different views of the same sample are close and views from different samples are far apart, which produces discriminative, task-agnostic representations (Khosla et al., 2020). In this thesis, a *view* always refers to an alternative representation of the same underlying PSG segment. Depending on the objective, views are either different modality subsets of the same patch (*e. g.* EEG-only versus EEG+ECG for LOOC) or stochastically augmented versions of the same multichannel patch (for SimCLR).

InfoNCE-Loss

We follow the InfoNCE formulation of Oord et al. (2018). An encoder f_θ maps an input segment x to an embedding $z \in \mathbb{R}^E$, which is then ℓ_2-normalized so that z lies on the unit sphere (Chen et al., 2020). Similarity is measured with cosine similarity, which for unit vectors reduces to the dot product, $\mathrm{sim}(u, v) = u^\top v$.

For each training example x_i we choose one embedding z_i as the *anchor* and one embedding z_i^+ as its *positive* (a different view of the same underlying sample). All other embeddings in the current batch that do not correspond to this sample form the *negative* set $A(i)$. The temperature $\tau > 0$ rescales similarities before the softmax. The InfoNCE loss for anchor i is

$$\mathcal{L}_i = -\log \frac{\exp(\mathrm{sim}(z_i, z_i^+)/\tau)}{\sum_{a \in A(i)} \exp(\mathrm{sim}(z_i, z_a)/\tau)}. \tag{3.23}$$

The overall loss is the average of $\mathcal{L}_i$ over all anchors in the batch. In practice one often uses a symmetric variant where each positive also acts as an anchor in turn (Chen et al., 2020).

Leave-One-Out Contrastive Loss

SleepFM uses InfoNCE with a specific way of constructing anchors and positives across modalities (Thapa et al., 2025). For a given 30-second epoch and modality m (for example BAS, EOG or respiratory), let $z_n^{(m)}$ denote the embedding produced by the encoder for sample n. Let $\mathcal{M}$ be the set of all modalities and define a *leave-one-out* positive embedding by averaging over all modalities except m:

$$\bar{z}_n^{(-m)} = \frac{1}{|\mathcal{M}| - 1} \sum_{m' \in \mathcal{M} \setminus \{m\}} z_n^{(m')} \tag{3.24}$$

For modality m and sample n, leave-out-out contrastive loss (LOOC) treats $z_n^{(m)}$ as the anchor and $\bar{z}_n^{(-m)}$ as its positive, and uses all embeddings from other samples in the batch as negatives. Plugging these choices into Eq. (3.23) yields the LOOC loss described by Thapa et al. (2025). Intuitively, each modality embedding is encouraged to align with a summary of the remaining modalities for the same epoch, so that the shared representation captures information that is consistent across signals.

SimCLR Loss

SimCLR (Chen et al., 2020) is another instance of InfoNCE. Instead of using different modalities, SimCLR constructs two stochastic augmentations of each input sample (for example, different crops and distortions of an image) and treats these as two views of that sample. With a batch of N original samples, there are $2N$ augmented views and corresponding embeddings $z_1, \ldots, z_{2N}$. For each view i, the positive $z_{j(i)}$ is the embedding of the other augmentation of the same sample, and all remaining $2N - 2$ views in the batch are negatives. The SimCLR loss is

$$\mathcal{L}_{\text{SimCLR}} = \frac{1}{2N} \sum_{i=1}^{2N} -\log \frac{\exp(\text{sim}(z_i, z_{j(i)})/\tau)}{\sum_{r=1, r \neq i}^{2N} \exp(\text{sim}(z_i, z_r)/\tau)}. \tag{3.25}$$

Compared with LOOC, the underlying objective has the same InfoNCE form, but the notion of a view and of a positive pair is different: SimCLR uses two augmentations of a single modality, whereas LOOC uses one modality embedding and the averaged embedding of the remaining modalities from the same PSG segment.

3.3.2 Focal Loss

Focal loss reshapes cross-entropy so that well-classified examples contribute less to the loss, which emphasizes hard, misclassified samples under strong class imbalance (Lin et al., 2020). In this thesis we mainly use it for multi-class sleep staging, where the classes are the five AASM stages (Wake, N1, N2, N3, REM).

Let a downstream classifier take an encoder embedding and produce logits $a_i \in \mathbb{R}^K$ for sample i, where K is the number of classes. Class probabilities are obtained with a softmax

$$p_{i,k} = \frac{\exp(a_{i,k})}{\sum_{k'=1}^{K} \exp(a_{i,k'})} \tag{3.26}$$

and the one-hot target vector is $y_{i,k} \in \{0, 1\}$ with $y_{i,k} = 1$ if sample i belongs to class k. For a mini-batch of N samples, the multi-class focal loss is

$$\mathcal{L}_{\text{FL}} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} \alpha_k \, y_{i,k} \, (1 - p_{i,k})^{\gamma} \log p_{i,k} \tag{3.27}$$

where $\gamma \geq 0$ is the focusing parameter and $\alpha_k \geq 0$ is an optional class weight for class k. Setting $\gamma = 0$ and $\alpha_k = 1$ for all k recovers standard cross-entropy. Increasing γ down-weights easy, high-confidence examples and focuses training on hard ones. When we combine focal loss with class weighting, we set α_k equal to the class weights defined from label frequencies.

3.3.3 Class-Weighted Cross-Entropy

When classes are imbalanced, plain cross-entropy can favor common classes. Class-weighted cross-entropy counteracts this by giving a larger penalty to mistakes on rare classes (Cui et al., 2019). Using the same notation as in Section 3.3.2, with softmax probabilities $p_{i,k}$ and one-hot targets $y_{i,k}$, the weighted loss for a mini-batch of N samples is

$$\mathcal{L}_{\text{wCE}} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} \alpha_k \, y_{i,k} \, \log p_{i,k}, \tag{3.28}$$

where $\alpha_k > 0$ is the weight for class k. Setting $\alpha_k = 1$ for all k recovers the standard (unweighted) cross-entropy loss.

In our experiments we choose α_k inversely proportional to the class frequency. Let n_k be the number of training samples in class k and $N_{\text{train}} = \sum_{k=1}^{K} n_k$ the total number of training samples. We set

$$\alpha_k = \frac{N_{\text{train}}}{K \, n_k}, \tag{3.29}$$

so that rarer classes (small n_k) receive larger weights. This normalization keeps the average weight close to one, so the overall scale of the loss is comparable across datasets. The same α_k are used as class weights in the focal loss in Eq. (3.27).

3.3.4 Mean Squared Error

Mean squared error (MSE) is a standard loss for regression problems and penalizes the squared deviation between predicted and true targets. Given a batch of N samples with true continuous values y_i and model predictions $\hat{y}_i$, the MSE loss is

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2. \tag{3.30}$$

It is widely used for apnea severity regression tasks such as AHI prediction in recent deep-learning work on sleep apnea (Hu et al., 2025).

3.3.5 F1 and Per-Class F1

Per-Class F1

In multi-class sleep staging (Wake, N1, N2, N3, REM) and multi-class apnea detection, we report per-class effectiveness because classes are imbalanced and (especially N1 for sleep staging OSA for apnea) are intrinsically harder to score (see Figure 4.2). Given model probabilities $p_{i,k}$ for sample i and class k, we obtain a hard prediction by choosing the most likely class

$$\hat{y}_i = \arg\max_k p_{i,k},$$

which defines a standard multi-class confusion matrix.

Following Sokolova and Lapalme (2009); Powers (2008), we evaluate each stage k as a one-vs-rest detection problem: class k is treated as positive and all other stages as negative. This isolates how well the model detects that specific stage against all alternatives, without changing the underlying multi-class prediction rule. From the confusion matrix we define

$$\begin{aligned} \mathrm{TP}_k &= \#\{\text{true } = k,\ \text{predicted } = k\}, \\ \mathrm{FP}_k &= \#\{\text{true } \neq k,\ \text{predicted } = k\}, \\ \mathrm{FN}_k &= \#\{\text{true } = k,\ \text{predicted } \neq k\}, \\ \mathrm{TN}_k &= \#\{\text{true } \neq k,\ \text{predicted } \neq k\}. \end{aligned} \tag{3.31}$$

The precision and recall for class k are

$$P_k = \frac{\mathrm{TP}_k}{\mathrm{TP}_k + \mathrm{FP}_k}, \qquad R_k = \frac{\mathrm{TP}_k}{\mathrm{TP}_k + \mathrm{FN}_k}, \tag{3.32}$$

and the per-class F1 score is their harmonic mean,

$$F1_k = \frac{2P_kR_k}{P_k + R_k}. \tag{3.33}$$

If the model never predicts class k but there are true examples of k in the evaluation set, then $\mathrm{TP}_k = \mathrm{FP}_k = 0$ and $\mathrm{FN}_k > 0$. In this case we follow common practice and set $P_k = 0$ and hence $F1_k = 0$, reflecting a complete failure to detect that class. If class k does not appear at all in the evaluation set ($\mathrm{TP}_k = \mathrm{FP}_k = \mathrm{FN}_k = 0$), $F1_k$ is undefined.

F1

F1 summarizes multi-class performance by averaging the per-class F1 scores equally, regardless of class frequency (Sokolova and Lapalme, 2009; Powers, 2008):

$$\text{Macro-F1} = \frac{1}{K}\sum_{k=1}^{K} F1_k, \tag{3.34}$$

where K is the number of classes. Because each class contributes the same weight, F1 is sensitive to performance on minority classes and is less dominated by the majority classes than plain accuracy (see Section 3.3.6).

3.3.6 Balanced Accuracy

For completeness, we report plain accuracy, which measures the fraction of correctly classified classes. In strongly imbalanced data, this metric can be high even if the model performs poorly on rare classes.

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N} 1\{\hat{y}_i = y_i\} = \frac{1}{N}\sum_{k=1}^{K} \mathrm{TP}_k. \tag{3.35}$$

To better reflect performance on each class, we use balanced accuracy (BA), defined as the mean of per-class recalls (Sokolova and Lapalme, 2009; Brodersen et al., 2010). Using the recall R_k from Section 3.3.5:

$$\mathrm{BA} = \frac{1}{K}\sum_{k=1}^{K} R_k = \frac{1}{K}\sum_{k=1}^{K} \underbrace{\frac{\mathrm{TP}_k}{\mathrm{TP}_k + \mathrm{FN}_k}}_{\text{recall of class k}} \tag{3.36}$$

Balanced accuracy weights all classes equally, so low recall on minority classes directly lowers the score even if the model is very accurate on majority stages.

3.3.7 Cohen's k

Cohen's kappa (κ) measures how much two labelers agree, after correcting for the amount of agreement that would be expected by chance alone. In our case, the human annotations act as rater A and the model predictions as rater B. Unlike plain accuracy (see Section 3.3.6), κ subtracts the agreement that comes just from both raters following the same class distribution, which makes it better suited to imbalanced data (McHugh, 2012).

For a K-class problem we summarize predictions with a (non-normalized) confusion matrix $N = (n_{ij})$, where n_{ij} counts how many examples (epochs) have true class i and predicted class j. The total number of evaluated examples is

$$N_{\text{tot}} = \sum_{i=1}^{K} \sum_{j=1}^{K} n_{ij} \tag{3.37}$$

The observed agreement

$$p_o = \frac{1}{N_{\text{tot}}} \sum_{i=1}^{K} n_{ii} \tag{3.38}$$

is simply the fraction of epochs where prediction and ground truth coincide, *i. e.* the overall accuracy.

To estimate how much agreement we would see by accident, we use the marginal totals of the confusion matrix (McHugh, 2012). Let $n_{i:} = \sum_j n_{ij}$ be the number of true examples of class i (row sum) and $n_{:i} = \sum_j n_{ji}$ the number of predictions of class i (column sum). The expected chance agreement is

$$p_e = \frac{1}{N_{\text{tot}}^2} \sum_{i=1}^{K} n_{i:}\, n_{:i} \tag{3.39}$$

that is, the probability that both raters would independently assign the same class purely because they follow these empirical label frequencies. Cohen's kappa is then defined as

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{3.40}$$

which yields $\kappa = 1$ for perfect agreement, $\kappa = 0$ for chance-level agreement, and negative values if the model tends to disagree with the ground truth more than would be expected by chance.

3.3.8 AUROC and AUPR

For highly imbalanced binary problems such as apnea detection, ranking-based metrics are often more informative than accuracy (Powers, 2008). We therefore report the area under the receiver operating characteristic curve (AUROC) and the area under the precision-recall curve (AUPR).

Consider a binary classifier that outputs a continuous score s_i (*e. g.* predicted probability of apnea) for each example i. For any decision threshold t we obtain predicted labels and can compute TP, FP, TN, and FN (see Section 3.3.5). The true positive rate (TPR, or recall) and false positive rate (FPR) at threshold t are

$$\text{TPR}(t) = \frac{\text{TP}(t)}{\text{TP}(t) + \text{FN}(t)}, \qquad \text{FPR}(t) = \frac{\text{FP}(t)}{\text{FP}(t) + \text{TN}(t)} \tag{3.41}$$

The receiver operating characteristic (ROC) curve plots $\text{TPR}(t)$ against $\text{FPR}(t)$ as t varies from 1 to 0. The AUROC is the area under this curve and can be interpreted as the probability that

a randomly chosen positive example receives a higher score than a randomly chosen negative example (Powers, 2008). AUROC is insensitive to the overall class prevalence and summarizes how well the model ranks positives above negatives.

Precision-recall analysis focuses on the positive class and is often more sensitive to performance under extreme imbalance (Sokolova and Lapalme, 2009; Powers, 2008). For each threshold t, precision and recall are

$$\mathrm{Precision}(t) = \frac{\mathrm{TP}(t)}{\mathrm{TP}(t) + \mathrm{FP}(t)}, \qquad \mathrm{Recall}(t) = \frac{\mathrm{TP}(t)}{\mathrm{TP}(t) + \mathrm{FN}(t)} \tag{3.42}$$

The precision-recall (PR) curve plots $\mathrm{Precision}(t)$ against $\mathrm{Recall}(t)$. Its area, often summarized as average precision (AP) or AUPR, emphasizes how well the model identifies the minority (positive) class without being diluted by the many true negatives.

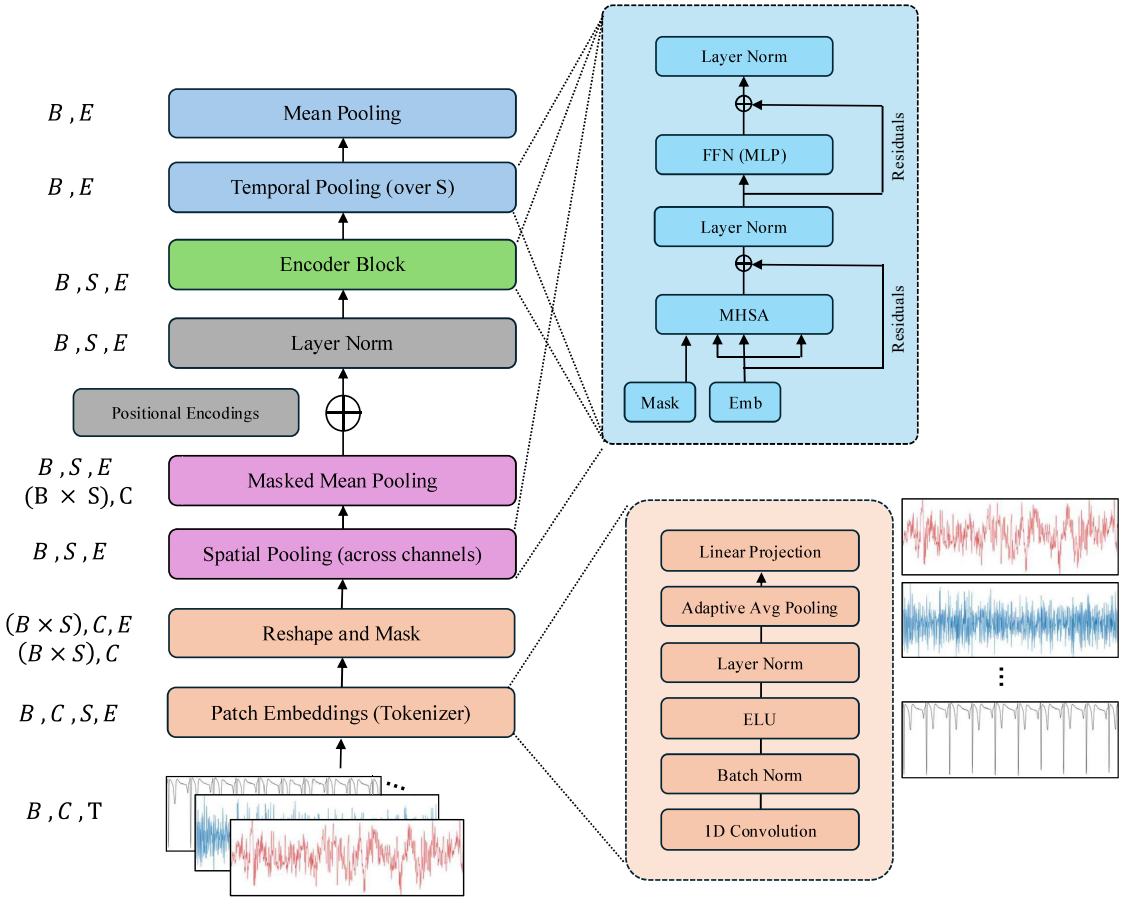


Figure 3.3: SLEEPFM SET-THEN-SEQUENCE ENCODER. *Overview of the SleepFM foundation encoder (Thapa et al., 2025). Raw PSG segments $X \in \mathbb{R}^{B\times C\times T}$ are tokenized by a shared 1D convolutional network into patch embeddings of shape $B \times C \times S \times E$. For each patch index s a channel-set encoder applies masked self-attention across channel embeddings and permutation-invariant masked mean pooling to produce a single fused token $z_{b,s} \in \mathbb{R}^E$. The resulting sequence of fused tokens is stacked into a tensor $Z \in \mathbb{R}^{B\times S\times E}$, augmented with temporal positional encodings and passed through a stack of Transformer encoder blocks. A final mean pooling over the S time steps yields an epoch-level representation z_{fm} that is used for self-supervised pretraining and for downstream prediction heads. Tensor shapes on the left indicate how the representation changes through the pipeline.*

Chapter 4

Data

We train and evaluate our models on large-scale PSG data drawn from multiple public cohorts. These datasets differ in montage, sampling rate, population, and prevalence of sleep-disordered breathing, which makes them well suited for studying cross-cohort generalization. Figure 4.1 visualizes the total recording hours per dataset after preprocessing.

For clarity we distinguish between pretraining datasets (Section 4.1), used for self-supervised foundation-model pretraining, and evaluation dataset (Section 4.2), used for all downstream supervised experiments in this thesis. Section 4.3 explains the common preprocessing and harmonization applied across cohorts.

4.1 Pretraining Datasets

We pretrain the foundation model on ten public PSG cohorts that span home and laboratory recordings, community samples and clinical populations, and a wide range of montages (Alvarez-Estevez and Rijsman, 2021; Blackwell et al., 2011; Chen et al., 2015; Goldberger et al., 2000; Kemp et al., 2000; O'Reilly et al., 2014; Redline et al., 1995; Stephansen et al., 2018; Terzano et al., 2001; Young et al., 2009; Zhang et al., 2018). Table 4.1 summarizes the cohorts used for self-supervised pretraining. Modalities are grouped as brain activity signals (BAS: EEG and EOG), electrocardiogram (ECG), electromyogram (EMG), and respiratory/oximetry (RESP). The table reports the number of files and the total recording hours after preprocessing.

For self-supervised training we ignore all manual sleep labels and draw unlabeled segments from this multi-cohort pool. When we vary the pretraining set size in Experiments (Chapter 6), we subsample subjects from this same pool, so that changes in performance can be attributed to data scale.

4.1.1 MESA

The Multi-Ethnic Study of Atherosclerosis (MESA) Sleep Exam provides unattended home PSG for more than two thousand middle-aged and older adults from four ethnic groups. The montage includes frontal and central EEG, bilateral EOG, chin and leg EMG, ECG, respiratory effort, airflow, oximetry, and body position (see Table A.1 for more detailed information about the channels) (Chen et al., 2015; Zhang et al., 2018).

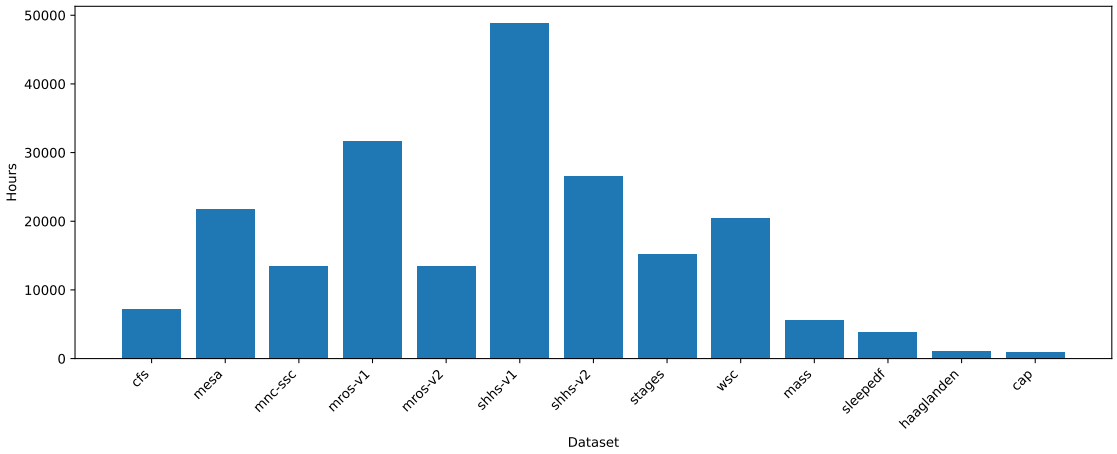


Figure 4.1: TOTAL HOURS OF RECORDING PER DATASET. *This figure shows the total PSG recording hours available per dataset after preprocessing.*

4.1.2 MrOS

The Osteoporotic Fractures in Men Sleep Study (MrOS) is a large unattended home PSG cohort of older men. Recordings use a montage with central EEG, bilateral EOG, chin and leg EMG, ECG, respiratory effort, airflow, and oximetry (Blackwell et al., 2011; Zhang et al., 2018). The channel configuration and sampling rates are detailed in the Appendix in Table A.2.

4.1.3 MASS

The Montreal Archive of Sleep Studies (MASS) aggregates in-lab PSG from several research protocols. We use the SS3 subset, which consists of overnight PSG from healthy adults with a dense EEG montage (up to 20 channels), bilateral EOG, chin and leg EMG, and ECG (O'Reilly et al., 2014). The corresponding modalities, channels, and sampling rates are summarized in Table A.3.

4.1.4 Sleep-EDF

The Sleep-EDF expanded database provides overnight PSG from healthy participants and patients, recorded either at home (cassette recordings) or in a telemetry setting. The recordings include EEG, EOG, and a chin EMG channel, with additional low-rate respiratory and temperature channels in some files (Table A.4) (Kemp et al., 2000; Goldberger et al., 2000).

4.1.5 WSC

The Wisconsin Sleep Cohort (WSC) follows adults across repeated in-lab PSG studies over many years. Later recordings employ a rich EEG montage, bilateral EOG, chin and leg EMG, ECG, airflow, effort belts, snore microphone, oxygen saturation, and body position, as detailed in Table A.5 (Young et al., 2009; Zhang et al., 2018).

Table 4.1: PRETRAINING DATASETS OVERVIEW. *Overview of datasets used for self-supervised pretraining. Modalities are grouped as brain activity signals (BAS: EEG+EOG), electrocardiogram (ECG), electromyogram (EMG), and respiratory/oximetry signals (RESP). Channel counts are summarized from the cohort-specific tables and give typical numbers per study night. #Subjects counts EDF files; Hours are total PSG hours after preprocessing (Alvarez-Estevez and Rijsman, 2021; Blackwell et al., 2011; Chen et al., 2015; Goldberger et al., 2000; Kemp et al., 2000; O'Reilly et al., 2014; Redline et al., 1995; Stephansen et al., 2018; Terzano et al., 2001; Young et al., 2009; Zhang et al., 2018).*

Name	Modalities and associated channels	#Subjects	Hours
MESA	BAS: 3 EEG + 2 EOG; ECG: 1; EMG: 2; RESP: 7	2'056	21'745.2
MrOS	BAS: 2 EEG + 2 EOG; ECG: 2; EMG: 5; RESP: 5	2'907	31'710.8
MASS (SS3)	BAS: 4-20 EEG + 2 EOG; ECG: 1; EMG: 2; RESP: 1-3 (thermistor, RIP, SpO_2)	653	5'540.4
Sleep-EDF	BAS: 2 EEG + 1 EOG; ECG: 0; EMG: 1; RESP: 0-1 (thermistor, subset only)	394	3'849.0
WSC	BAS: 6 EEG + 2 EOG; ECG: 1; EMG: 3; RESP: 4	2'570	20'520.2
MNC	BAS: 5 EEG + 2 EOG; ECG: 1-3; EMG: 5; RESP: up to 8	1'438	13'400.8
HMC	BAS: 4 EEG + 2 EOG; ECG: 1; EMG: 1; RESP: 0	302	1'144.2
CAP	BAS: 6 EEG + 2 EOG; ECG: 1; EMG: 2-3; RESP: 3-4	106	992.6
STAGES	BAS: $\geq$3 EEG + 2 EOG; ECG: 1; EMG: 2; RESP: $\geq$2	1'914	15'165.0
CFS	BAS: 2 EEG + 2 EOG; ECG: 2; EMG: 5; RESP: 4	730	7'218.5

4.1.6 MNC

The Mignot Nature Communications (MNC) dataset is a multi-site clinical PSG collection with a broad spectrum of sleep disorders, including narcolepsy, insomnia, hypersomnolence, and obstructive sleep apnea (Stephansen et al., 2018; Zhang et al., 2018). Recordings come from accredited sleep labs with standard clinical montages (multiple EEG derivations, bilateral EOG, chin and leg EMG, ECG, and respiratory channels), summarized in Table A.6.

4.1.7 HMC

The Haaglanden Medisch Centrum (HMC) database contains PSGs from patients referred for suspected sleep disorders (Alvarez-Estevez and Rijsman, 2021; Goldberger et al., 2000). The public release focuses on staging channels: four EEG derivations, bilateral EOG, chin EMG, and ECG, without respiratory signals (Table A.7).

4.1.8 CAP

The Cyclic Alternating Pattern (CAP) database consists of full-night PSGs from controls and patients scored for CAP, an EEG marker of sleep instability (Terzano et al., 2001; Goldberger et al., 2000). Recordings include several EEG derivations, EOG, chin and tibialis EMG, respiratory channels, oximetry, and ECG (see Table A.8).

4.1.9 STAGES

The Stanford Technology Analytics and Genomics in Sleep (STAGES) cohort is a multi-site, in-lab PSG study spanning adolescents and adults evaluated for sleep disorders. As can seen in

Table 4.2: DOWNSTREAM DATASET OVERVIEW. *Overview of the SHHS dataset used for downstream tasks and evaluation. Modalities are grouped as BAS (EEG and EOG), ECG, EMG, and RESP. Channel counts are summarized from the cohort-specific tables. #Subjects counts EDF files; Hours are total PSG hours after preprocessing (Quan et al., 1997; Zhang et al., 2018).*

Name	Modalities and associated channels	#Subjects	Hours
SHHS	BAS: 2 EEG + 2 EOG; ECG: 1; EMG: 1; RESP: 4	5'793	48'861.3

Table A.9, recordings include multiple EEG leads, bilateral EOG, chin and leg EMG, respiratory channels, and ECG, but montages vary slightly across sites (Zhang et al., 2018).

4.1.10 CFS

The Cleveland Family Study (CFS) is a family-based study of sleep apnea with overnight home PSGs across children and adults. The montage includes central EEG, bilateral EOG, chin and leg EMG, ECG, respiratory effort, airflow, snoring, and oximetry, summarized in Table A.10 (Redline et al., 1995; Zhang et al., 2018).

4.2 Evaluation Dataset

The evaluation dataset was solely used for downstream tasks and not included in the pretraining. Therefore, it acted as a hold-out set.

4.2.1 SHHS

The Sleep Heart Health Study (SHHS) is a multi-center community cohort designed to investigate sleep-disordered breathing and cardiovascular outcomes (Quan et al., 1997). It comprises two waves of unattended home PSG: Visit 1 enrolled 6'441 participants, yielding 5'793 usable overnight recordings, and Visit 2 restudied a subset of these individuals about five years later, adding roughly 2'600 further nights (Zhang et al., 2018). In this thesis we treat each overnight PSG as one subject-level recording. If a participant has two nights, they contribute two recordings but are always kept in the same split (see Section 4.4)

All SHHS studies use a standardized montage including a central EEG derivation (with backup), bilateral EOG, chin EMG, ECG, airflow via thermistor (plus nasal pressure in Visit 2), thoracic and abdominal effort belts, finger oximetry, body position, and leg-movement sensors (Table A.11).

As this is the downstream dataset, we adjusted the labels. Original sleep staging followed Rechtschaffen & Kales rules (predecessor from AASM) and we map these labels to the AASM scheme (essentially merging sleep stages 3 and 4 into N3) and harmonize sampling rates as described in Section 4.3. After preprocessing, SHHS serves exclusively as our downstream cohort for supervised sleep-staging and apnea-detection experiments. Table 4.2 summarizes the modalities and overall size of the SHHS corpus used in this work. Figure 4.2 shows the sleep stage distribution and the proportion of epochs involved in respiratory events versus normal breathing.

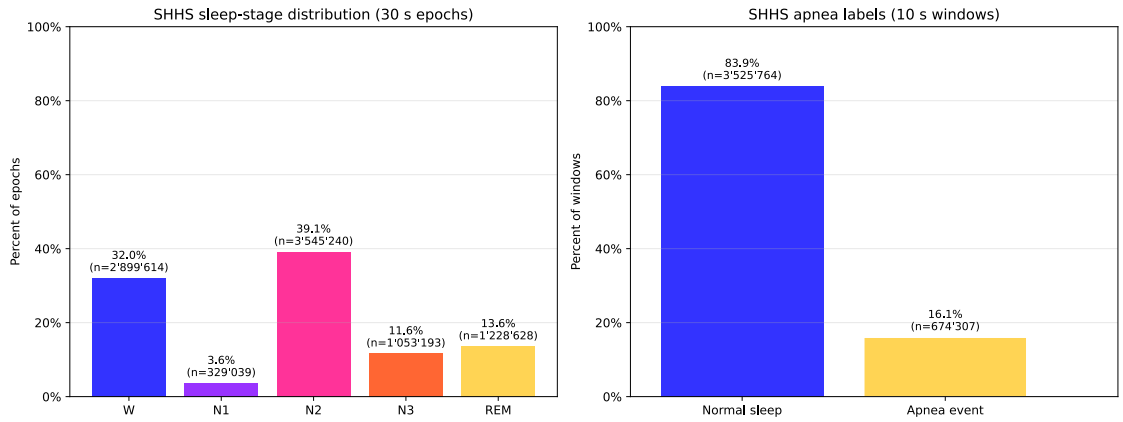


Figure 4.2: SHHS DOWNSTREAM LABEL DISTRIBUTIONS. *Sleep-stage distribution of 30-second epochs and proportion of apnea vs non-apnea epochs in SHHS after preprocessing.*

4.3 Preprocessing and Harmonization

To integrate data from all cohorts, we applied a uniform preprocessing pipeline. First, we resampled every signal to a common frequency of 128Hz. Original sampling rates varied widely between datasets (from as low as 100Hz in Sleep-EDF to 256Hz in MESA and others). Standardizing to 128Hz ensures that our model receives inputs with comparable time resolution and frequency content. When downsampling higher-rate signals, we applied an anti-aliasing filter (a 4th-order zero-phase Butterworth) to band-limit the signal below 64Hz prior to decimation. This software filter preserves the signal's important sleep frequencies (primarily in the 0.3-35Hz range for EEG/EOG) while preventing high-frequency noise from aliasing. After filtering, we perform resampling via linear interpolation to 128Hz. We also up-sampled any lower-frequency channels (*e. g.* SpO_2 at 1Hz) to 128Hz using interpolation, so that all signals share a synchronized timeline and sample count.

In addition, we address differences in signal referencing and channel layouts across datasets. Since each PSG dataset provides a different montage and reference scheme, we harmonized the schemes by mapping them to either one of four defined modalities: brain activity signals (BAS), electromyography (EMG), electrocardiogram (ECG), respiratory (RESP). In practice, we therefore selected a set of channels and assigned them to one modality (*e. g.* EEG channels such as C3/C4 were mapped to BAS). We handle these channel differences with the permutation-invariant set attention and pooling over channels described in Section 3.1.4, which treats the available channels at each time step as an unordered set. We further normalized each channel by subtracting its mean and dividing by its standard deviation (per recording). This z-score normalization makes the amplitude scale consistent across recordings and devices, removing variations in amplifier gain. Overall, our preprocessing and data harmonization minimized inter-dataset differences in signal characteristics, as recommended by previous multi-cohort sleep studies (Stephansen et al., 2018; Zhang et al., 2018; Thapa et al., 2025).

4.4 Subject-Wise Splits

All downstream supervised experiments in this thesis use SHHS only. None of the SHHS recordings are seen during self-supervised pretraining. To obtain reliable estimates of generalization to unseen individuals, we create *subject-wise* splits:

- We treat each SHHS participant as a unit. If a subject has two nights (Visit 1 and Visit 2), both nights are always assigned to the same split.
- We randomly partition subjects into 60% train, 20% validation, and 20% test. The same split is used for sleep staging and apnea detection.

Subject-wise splitting prevents information leakage across nights from the same individual: the model cannot learn a participant's EEG signature on one night and be evaluated on another night of that participant, which in turn yields a more realistic assessment of performance on truly unseen patients.

Chapter 5

Methods

This chapter describes the methods used in this thesis. We first introduce the foundation model architecture and self-supervised pretraining strategy, then outline the training and evaluation pipeline that links the research questions and hypotheses from Section 1.3 to the concrete experiments.

5.1 Foundation Model Architecture

All experiments in this thesis build on the SleepFM foundation encoder described in Section 3.2. SleepFM is a set-then-sequence Transformer that first aggregates information across channels within each short time patch and then models temporal dependencies across patches within an epoch and across the night (Thapa et al., 2025). In this section we specify the concrete instantiation used in our work and define the representations that the downstream heads operate on.

5.1.1 Input Representation and Channel Mapping

We follow Thapa et al. (2025) and group PSG channels into four modality blocks: baseline signals (BAS, consisting of EEG and EOG), ECG, EMG, and respiratory channels, as described in Section 3.2.2. After standard preprocessing such as filtering, resampling to 128 Hz, and epoching into 30-second windows each epoch is represented as $X \in \mathbb{R}^{C \times T}$ with $T = 3840$ samples.

The SleepFM tokenizer introduced in Section 3.2.1 is applied to every available channel. The same convolutional network f_θ is used for all channels, so the architecture does not depend on the specific montage. In our implementation we keep the tokenizer architecture and hyperparameters identical to Thapa et al. (2025) (patch length $P = 640$ samples, six stride-2 convolutional layers, token dimension $E = 128$). The only adaptation is the channel map. We extend the original configuration so that additional channels present in our cohorts, for example extra EMG or respiratory signals, are assigned to the appropriate modality group. No new weights are created for these channels and the encoder remains channel-flexible.

5.1.2 Temporal Encoder and Learned Representations

For each epoch and each channel the tokenizer produces $S = T/P = 6$ patch embeddings of dimension $E = 128$. The channel-set encoder from Section 3.2.1 compresses the C channel embeddings at each patch index into a single fused token, resulting in a sequence $Z \in \mathbb{R}^{S \times E}$ per

epoch. We add sinusoidal positional encodings over the patch index and process Z with a Transformer encoder that has $L_{\text{enc}} = 6$ layers, model dimension $E = 128$ and $H = 8$ attention heads, exactly as in Thapa et al. (2025). The encoder output is denoted by $H \in \mathbb{R}^{S \times E}$.

We define the patch-level representation z_{enc} as the ℓ_2-normalized rows of H. The contrastive losses in Section 3.3.1 are applied to z_{enc}. For downstream prediction we compute an epoch-level representation $z_{\text{fm}} \in \mathbb{R}^E$ by averaging H over the patch dimension. In whole-night models we use sequences $(z_{\text{fm},1}, \ldots, z_{\text{fm},L_{\text{night}}})$ constructed from consecutive epochs.

5.1.3 Self-Supervised Pretraining Strategy

We train the foundation encoder described in Section 5.1 using contrastive self-supervision. All objectives are InfoNCE losses in the sense of Section 3.3.1, applied to the patch-level embeddings z_{enc} (the ℓ_2-normalized rows of H from Section 5.1.2). Each training step builds a set of views for every 5-second patch and uses an InfoNCE loss to pull together views of the same patch and push apart views from different patches in the batch.

5.1.4 LOOC Baseline

Our default objective is the leave-one-out (LOOC) contrastive loss of Thapa et al. (2025), as formalized in Section 3.3.1. For a given 30-second epoch and modality m (BAS, ECG, EMG or RESP), the encoder produces a modality-specific embedding $z^{(m)}$ and an averaged embedding $\bar{z}^{(-m)}$ over all remaining modalities for the same epoch. LOOC treats $z^{(m)}$ as the anchor and $\bar{z}^{(-m)}$ as its positive, and uses embeddings from other samples in the minibatch as negatives via the InfoNCE loss in Eq. (3.23). In practice we instantiate this at the patch level: for each patch where at least two modalities are present we form pairs

$$\left(z_{\text{enc}}^{(m)},\ \bar{z}_{\text{enc}}^{(-m)}\right) \tag{5.1}$$

and apply LOOC exactly as in Section 3.3.1, masking out any modalities that are missing in a given cohort.

This construction encourages each modality-specific view to align with the shared representation of the remaining modalities for the same patch, so that the encoder learns cross-modal structure while remaining robust to missing channels.

5.1.5 Auxiliary SimCLR Term

To study objective design (RQ5) we also consider an augmentation-based SimCLR loss on top of LOOC, following Chen et al. (2020) and Section 3.3.1. For each patch we generate two stochastic augmentations of the same multichannel segment (small additive noise, mild amplitude scaling and a slight temporal jitter within the 5-second window) and pass both through the encoder. The resulting embeddings $z_{\text{enc}}^{(a)}$ and $z_{\text{enc}}^{(b)}$ form a positive pair, and all other augmented views in the minibatch act as negatives in the SimCLR loss of Eq. (3.25). In contrast to LOOC, this term does not depend on the number of available modalities and can be computed for every patch.

5.1.6 Combined Objective

When both objectives are used we simply add them

$$\mathcal{L}_{\text{SSL}} = \mathcal{L}_{\text{LOOC}} + \lambda\, \mathcal{L}_{\text{SimCLR}} \tag{5.2}$$

with $\lambda = 1$ in all RQ5 related experiments. LOOC enforces agreement between modality-specific and modality-averaged views, while SimCLR enforces invariance to small within-patch perturbations.

5.2 Downstream Evaluation Pipeline

All downstream experiments use the same protocol. We first train the foundation encoder with self-supervised objectives (Chapter 6) and then freeze its weights. For each downstream task we pass the labeled PSG data through the pretrained encoder to obtain fixed-size embeddings and train lightweight sequence heads on top of these embeddings. This section describes how those embeddings are constructed and how the sleep staging and apnea heads operate.

5.2.1 Embedding Extraction and Segmentation

Given a preprocessed PSG recording, we segment it at the label resolution of the downstream task and run each segment through the fixed foundation encoder from Section 5.1. For sleep staging we use the standard 30-second AASM epochs, and for apnea detection we use 10-second respiratory windows. In both cases the encoder produces a single E-dimensional vector per segment, which we denote by z_t.

For a full night with L_{night} labeled segments we therefore obtain a sequence

$$(z_1, z_2, \ldots, z_{L_{\text{night}}}) \in \mathbb{R}^{L_{\text{night}} \times E}$$

accompanied by a binary mask that marks padded time steps for nights of different lengths. These sequences are the sole inputs to all downstream heads. The raw PSG is not used during downstream training when the encoder is frozen.

5.2.2 Sleep Staging Head

For sleep stage classification we adopt the LSTM-based head from SleepFM (Section 3.2.4). The per-epoch embeddings $(z_1, \ldots, z_{L_{\text{night}}})$ correspond to consecutive 30-second AASM epochs in a night. The sequence is fed to a bidirectional LSTM with hidden size $E/2$ per direction and L_{seq} layers, followed by a linear layer that maps each time step to a five-dimensional logit vector over the AASM stages (W, N1, N2, N3, REM). A softmax over these logits defines per-epoch stage probabilities, and the predicted stage is the argmax of this distribution.

The staging head is trained with weighted cross-entropy loss on the SHHS training subjects (Section 4.4). Evaluation uses the subject-wise splits from Section 4.4 and reports F1, balanced accuracy, Cohen's κ, and confusion matrices over the five stages, as detailed in Section 3.3.

5.2.3 Apnea Detection Head

Architecturally, this head follows the recent trend of apnea models that use attention-based sequence encoders over full-night respiratory signals and train them in a multi-task fashion (Hu et al., 2025, Section 2.5.2). Apnea detection uses a multi-task Transformer head that operates on the sequence of 10-second embeddings for each night. The head first applies masked attention pooling across channels (as described in Section 3.1.5 and implemented in the MultiTaskApneaTransformer) to obtain a single token per time step. These tokens are then processed by a causal Transformer encoder, where self-attention is restricted to the current and past time steps so that the representation at time t only depends on the present and earlier 10-second windows.

The 10-second segmentation matches the AASM scoring rules, which define respiratory events as lasting at least 10 seconds (Berry et al., 2017).

On top of the Transformer outputs the head defines four prediction tasks:

1. A **4-class respiratory state** at each 10-second window (None, Hypopnea, Apnea, RERA) via a linear layer and softmax over 4 logits.
2. A **3-class apnea label** at each 10-second window (Rest, Obstructive, Central+Mixed) via a second linear layer and softmax over 3 logits.
3. A **night-level AHI regression head**, obtained by masked attention pooling over valid time steps followed by a linear layer that predicts a continuous AHI value.
4. A **night-level 4-class severity head** (Normal, Mild, Moderate, Severe), using the same pooled representation and a linear layer with 4 logits.

The token-level heads are trained with cross-entropy on SHHS apnea labels, while the AHI head uses a regression loss and the severity head uses cross-entropy. As with staging, the encoder remains frozen and only the apnea head parameters are updated.

Evaluation for apnea follows the metrics in Section 3.3. At the token level we compute confusion matrices, per-class recall, and macro balanced accuracy for the 4-class respiratory state and 3-class mechanism labels. At the night level we evaluate the AHI regression both as a continuous estimate and after thresholding at clinically relevant cutoffs (*e. g.* AHI ≥ 15) to obtain binary severe vs non-severe labels. For these we report balanced accuracy, AUROC, AUPR, and corresponding confusion matrices. The 4-class severity head is evaluated analogously to the multiclass token tasks.

5.3 Baseline Models for Comparison

Our baseline models use the same 1D-CNN front-end as our downstream models, creating and stacking 5-second patches in an identical way. However, unlike the downstream models, these baselines do not use the foundation-model encoder: instead, the 1D-CNN is applied directly to the preprocessed PSG. For sleep staging, we employ the LSTM model from Thapa et al. (2025), modified so that it can process raw data rather than encoder embeddings. For apnea detection, we reuse the same architecture as in our downstream setup, but add a wrapper that allows it to operate directly on the preprocessed PSGs instead of on embeddings from the encoder.

Chapter 6

Experiments

This chapter instantiates the methods from Chapter 5 to answer the research questions and test the hypotheses in Section 1.3. For each RQ we specify the pretraining configuration, the downstream tasks, and the evaluation protocol, using the foundation encoder from Section 5.1 together with the sleep staging and apnea heads from Sections 5.2.2 and 5.2.3. SHHS serves as the main downstream cohort, with subject-wise training, validation, and test splits as described in Section 4.4. Numerical results are deferred to Chapter 7.

6.1 Common Training Setup

6.1.1 Pretraining

Unless stated otherwise, all foundation models are pretrained on the multi-cohort PSG pool from Chapter 4, excluding SHHS. The full pretraining pool contains approximately 13'000 subjects across the non-SHHS cohorts. Pretraining uses the LOOC contrastive objective of Section 5.1.3 with the architecture and channel mapping specified in Section 5.1 (Thapa et al., 2025). We train for 10 epochs with Adam (learning rate 0.001, weight decay 0), batch size 32, dropout rate of 0.3, and a step-wise learning rate decay by a factor of 0.1 every two epochs. Unless stated otherwise, the temperature τ in the InfoNCE loss is a learnable scalar parameter initialized as in the original SleepFM implementation (Thapa et al., 2025).

6.1.2 Sleep Staging Head

For all sleep staging experiments we use the LSTM from Section 3.2.4, which takes a sequence of epoch-level embeddings $(z_1, \ldots, z_{L_{\text{night}}})$ as input and outputs logits for the five AASM stages. Nights are segmented into standard 30-second AASM epochs, so that each time step corresponds to one $z_t \in \mathbb{R}^{128}$. When used on frozen foundation model embeddings, we train only the LSTM layers and the final classification layer, and the encoder is fixed. Training uses Adam with learning rate 0.001, batch size 32, and up to 100 epochs. Unless stated otherwise, the loss is class-weighted cross-entropy with inverse-frequency weights computed from the SHHS training split and normalized to have mean 1 (Section 3.3).

6.1.3 Apnea Detection Head

Apnea detection uses the apnea Transformer head from Section 5.2.3 on top of the frozen foundation encoder. We train this head with Adam (learning rate 0.001, batch size 32) for up to

100 epochs. Input sequences consist of 10-second respiratory windows with corresponding embeddings, masked where necessary to account for variable night lengths and non-sleep segments. The loss is a weighted sum of class-weighted cross-entropy for the token-level 4-class and 3-class tasks, a regression loss on the night-level AHI, and cross-entropy on the night-level 4-class severity label.

6.1.4 Baselines Trained from Scratch

When comparing to models trained from scratch, we use the *same* sequence architectures as the downstream heads above: the LSTM for staging and the Transformer for apnea. In the scratch setting these models receive raw PSG inputs (with their own lightweight convolutional front-ends), whereas in the foundation-model setting they receive frozen embeddings z_{fm}. Training schedules, batch sizes, loss functions, and evaluation metrics are matched between the two settings.

6.1.5 Losses, Metrics, and Model Selection

All supervised staging models (downstream task models and scratch) are optimized with class-weighted cross-entropy as described in Section 3.3. Focal loss is only used in the dedicated imbalance experiment of RQ3. For sleep staging we treat balanced accuracy as the *primary* evaluation metric, complemented by F1 and Cohen's κ, following the definitions in Section 3.3.5.

For apnea detection we use the metrics from Section 3.3.8. We report token-level and night-level balanced accuracy, AUROC, and AUPR (with a particular focus on the binary AHI ≥ 15 decision), as well as confusion matrices for the multi-class tasks.

Across *all* experiments we select the checkpoint used for test evaluation by the lowest validation loss corresponding to the training objective of that model. The reported metrics are always computed once on this selected checkpoint and on the fixed SHHS test split.

6.2 RQ1: Data Efficiency of Pretraining

RQ1 asks how much unlabeled data is needed for contrastive pretraining to yield useful downstream representations. To answer this, we train separate foundation encoders on progressively larger subject pools from the multi-cohort pretraining set, keeping architecture and optimization hyperparameters fixed.

We start from the multi-cohort pretraining pool described in Chapter 4, excluding SHHS. To preserve the relative contribution of each cohort, we specify a fixed target cohort composition

$$\begin{aligned}\text{PCT} = \big[&(\text{MASS}, 1), (\text{MESA}, 15), (\text{MrOS}, 27), (\text{MNC}, 9), (\text{WSC}, 27),\\ &(\text{HMC}, 1), (\text{STAGES}, 14), (\text{CAP}, 1), (\text{CFS}, 5), (\text{Sleep-EDF}, 1)\big]\end{aligned}$$

where the numbers indicate relative subject counts per cohort. For a target pretraining size we then create subject pools of $\{1\text{k}, 2\text{k}, 3\text{k}, 5\text{k}, 9\text{k}, \text{all}\}$, where *all* denotes the full pretraining pool of 13'526 subjects. For each scale we draw three independent subject subsets using random seeds 34, 42, and 67 so that variability across runs reflects both the amount of data and differences in which subjects are seen. Each subset is used to train one foundation encoder with the LOOC objective (Section 5.1.3) for 10 epochs or until a cap of 15 hours runtime is reached, yielding three checkpoints per pretraining scale.

For each pretrained encoder we freeze its weights and train the sleep staging LSTM head on SHHS using the common staging setup from Section 6.1. The only varying factor is which pretrained encoder produced the embeddings. We evaluate on the fixed SHHS test split and track

how balanced accuracy, F1, and κ change as a function of pretraining set size and how stable they are across the three seeds at each scale. RQ1 is answered by examining these plots and determining whether larger pretraining pools consistently improve downstream staging performance.

6.3 RQ2: Transfer Learning vs Training from Scratch (BAS-only)

RQ2 investigates to what extent a single multimodal foundation encoder pretrained with contrastive self-supervision improves BAS-only downstream performance compared with training the same architectures from scratch.

For each downstream task (sleep staging and apnea detection) we compare two settings. In the *pretrained* setting we take the encoder trained on the full multi-cohort pool from Section 6.2, freeze its weights, and train only the LSTM staging head or the apnea Transformer head on SHHS embeddings. For sleep staging we restrict the encoder inputs to BAS-only channels, while for apnea detection we use BAS together with respiratory signals (BAS+RESP), reflecting the minimal clinically useful modality set. In the *scratch* baseline, we train the same LSTM and Transformer architectures end-to-end on SHHS, starting from random initialization and using raw PSG inputs with the same modality restrictions (BAS-only for staging, BAS+RESP for apnea).

In both settings we use the SHHS splits and common downstream training protocol from Section 6.1. This design isolates the effect of pretraining under matched downstream conditions, so that comparing pretrained and scratch models directly addresses RQ2.

6.4 RQ3: Handling Class Imbalance in Downstream Sleep Staging

RQ3 focuses on class imbalance in sleep staging and asks how different loss functions affect performance on minority versus majority sleep stages when training on the imbalanced SHHS label distribution.

We fix the foundation encoder to the model pretrained on the full pretraining pool and keep the SHHS splits and staging head architecture from Section 6.1. On top of this encoder we train two variants of the staging head that differ only in the loss function. The first variant uses our default class-weighted cross-entropy with inverse-frequency weights w_k computed from the training split and normalized to have mean 1 (Eq. (3.29)). The second variant replaces this with a focal loss that uses the same weights w_k as class-wise α_k coefficients and a focusing parameter $\gamma = 2$ (Section 3.3), thereby down-weighting easy, majority-class examples and emphasizing misclassified and minority-stage epochs.

Both configurations use the setup from Section 6.1. The main focus is on per-class F1 and recall, especially for N1 and REM, and on how much each loss improves minority-stage performance without degrading overall F1 and balanced accuracy. RQ3 is answered by comparing these per-class and aggregate metrics between the weighted cross-entropy and focal-loss variants.

6.5 RQ4: Contribution of Additional Modalities Beyond EEG

RQ4 asks whether additional modalities beyond EEG (EOG, EMG, respiratory signals) improve downstream performance after multimodal pretraining, and whether any gains are consistent across tasks.

All foundation models in this RQ are pretrained on the full multimodal PSG pool (BAS, RESP, EMG, EKG) as in Section 6.1. During downstream training on SHHS we vary which modalities are fed into the encoder to generate embeddings. As a reference condition we retain only the BAS channels, and then we add further modalities step-wise, forming multimodal configurations that include BAS together with RESP, EMG, and EKG modality groups where available.

For each choice of downstream modalities we train the same staging LSTM and apnea Transformer heads with identical hyperparameters, loss functions, and model-selection rule as in Section 6.1, keeping the encoder frozen. Differences in performance can therefore be attributed to the modalities present at downstream time.

For sleep staging we compare BAS-only versus multimodal embeddings on the SHHS staging task, using balanced accuracy (primary), F1, Cohen's κ, and confusion matrices as in Section 6.1.5. For apnea detection we perform the same comparison on the SHHS apnea task, using the apnea metrics from Section 6.1.5. RQ4 is answered by examining whether multimodal inputs consistently improve these metrics over BAS-only embeddings and by identifying for which tasks and modality combinations the gains are most pronounced.

6.6 RQ5: Pretraining Objectives and Modality-Aware Training

RQ5 investigates how the choice of pretraining objective and modality-aware training strategy affects downstream transfer when pretraining uses all four modalities but some datasets lack two or more of them.

We compare two pretraining objectives while keeping architecture, data, and optimization hyperparameters fixed. In the *LOOC-only* condition, the encoder is trained with the LOOC contrastive loss of Section 5.1.3 applied at the patch level, using modality dropout to randomly hide channels during training. In the *LOOC+SimCLR* condition we add an augmentation-based SimCLR term (Section 5.1.3) that generates two stochastic augmentations per patch (small additive noise, mild amplitude scaling, slight temporal jitter) and applies an InfoNCE loss between the two augmented views.

The motivation for adding SimCLR is that several cohorts contain only a single modality or very few channels (for example, no RESP and no EKG). For such records the LOOC objective cannot be formed because it requires at least three modalities per sample. If we used LOOC alone, these recordings would either contribute no gradient or have to be discarded. The SimCLR term instead provides a self-contrastive signal based purely on augmentations of the available channels, so that every subject — including single-modality cohorts — can participate in pretraining. Modality dropout continues to encourage robustness to missing channels, while SimCLR ensures that structurally missing-modality datasets still shape the representation. For a fair comparison we scale our contrastive loss with a temperature parameter, following models such as CLIP (Radford et al., 2021). In these experiments, instead of learning the temperature, we fix the temperature to $\tau = 20$ for both LOOC and SimCLR.

For each pretrained encoder we freeze the weights and train the same LSTM downstream

head on SHHS as in RQ4, using the common downstream setup from Section 6.1. RQ5 is answered by comparing the downstream staging from Section 6.1.5 between the LOOC-only and LOOC+SimCLR encoders, in particular balanced accuracy and F1 for staging.

Chapter 7

Results

7.1 RQ1: Data Efficiency of Pretraining

To address RQ1, we vary the number of unlabeled subjects used for contrastive pretraining and evaluate downstream sleep staging on SHHS. Figure 7.1 shows F1 on the SHHS test split, and Figure 7.2 shows the corresponding balanced accuracies. Each pretraining scale yields three points, one for each random subject sampling, so that both mean performance and variability across seeds can be inspected.

Across both metrics, downstream sleep staging performance improves when the encoder is pretrained on more subjects. Moving from the smallest pretraining pool to intermediate scales gives a clear boost in F1 and balanced accuracy, while further increasing the pool size leads to more modest but still consistent gains. The spread between seeds also tends to shrink as the pretraining set grows, indicating that larger unlabeled pools not only improve average performance but also make it less sensitive to the specific subject sampling.

To understand which stages benefit most from additional pretraining data, Figure 7.3 presents per-class F1 scores for the stages. The figure allows comparison of how performance on majority stages and minority stages evolves with pretraining scale. With more unlabeled data, F1 improves not only on the dominant stages but also on the minority stages, where gains are particularly noticeable.

Figure A.1 shows confusion matrices for selected pretraining scales, using the same SHHS test split. As the pretraining pool grows, misclassifications between adjacent non-REM stages and between REM and wake become less frequent, and the diagonal entries become more pronounced. The confusion patterns therefore mirror the per-class F1 improvements and indicate that larger pretraining pools reduce systematic confusions rather than only improving a single class.

Taken together, H1a is supported and these curves answer RQ1 by showing that contrastive pretraining on diverse multi-cohort PSG clearly benefits downstream sleep staging, and that the benefits grow with the number of unlabeled subjects, with diminishing returns at the largest scales rather than an abrupt saturation.

7.2 RQ2: Transfer Learning of Foundation Model Embeddings (BAS-only)

To address RQ2, we compare BAS-only downstream models that use frozen representations from a single multimodal foundation encoder against the same architectures trained from scratch on preprocessed SHHS signals, for both sleep staging and apnea detection.

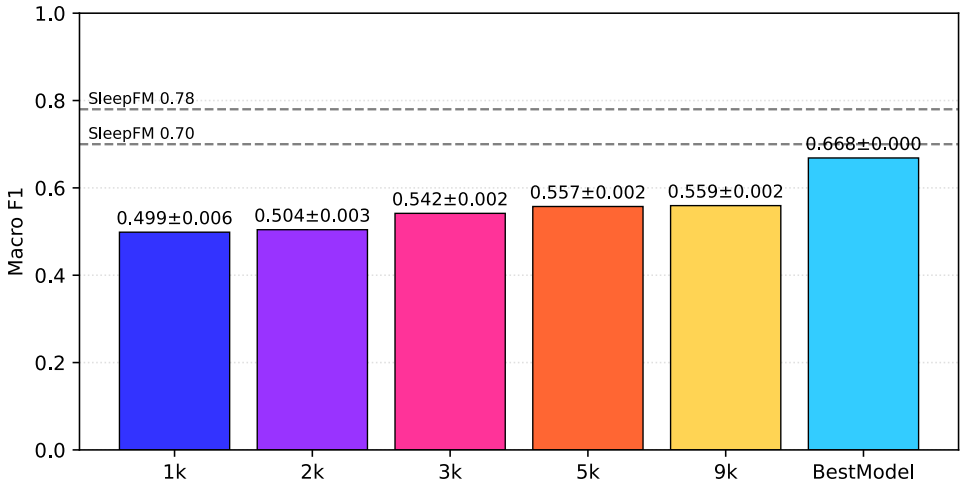


Figure 7.1: RQ1: F1 PER PRETRAINING SET SIZE. *F1 on the SHHS test split for the sleep staging LSTM with the number of unlabeled subjects used to pretrain the foundation encoder. Each scale is averaged over three random subject samplings and error bars show the standard deviation. BestModel is the model containing the whole pretraining dataset.*

7.2.1 Sleep Staging: Pretrained vs Scratch Model Performance

Figure 7.4 compares the downstream LSTM trained on frozen BAS-only embeddings to the LSTM baseline trained end-to-end on BAS-only PSG. On the SHHS hold-out subjects, the downstream model clearly outperforms the baseline. Using the foundation encoder as a feature extractor more than doubles both balanced accuracy and F1 compared with training the sequence model from scratch.

Per-stage F1 scores in Figure 7.5 show that this gain is consistent across all sleep stages. The downstream model attains high F1 on Wake and clearly better F1 on N1, N2, N3, and REM than the scratch baseline. The confusion matrices in Figure A.3 reflect the same pattern: the baseline model confuses most stages and almost never predicts minority stages correctly, whereas the downstream model substantially reduces these errors across the board.

Overall, H1a for sleep staging is strongly supported. Pretraining the multimodal encoder and then freezing it for BAS-only downstream training yields substantial improvements in balanced accuracy, F1, and per-stage performance compared with training the same architecture from scratch.

7.2.2 Apnea Detection: Pretrained vs Scratch Model Performance

For apnea detection, we compare a downstream Transformer head trained on BAS+RESP embeddings to a matched Transformer baseline trained from scratch on BAS+RESP PSG. Both models share the same multi-task architecture and output heads.

Figure 7.6 summarizes token-level and night-level metrics. At the subject level, the baseline model achieves stronger discrimination for the binary AHI decision, with higher AUROC and

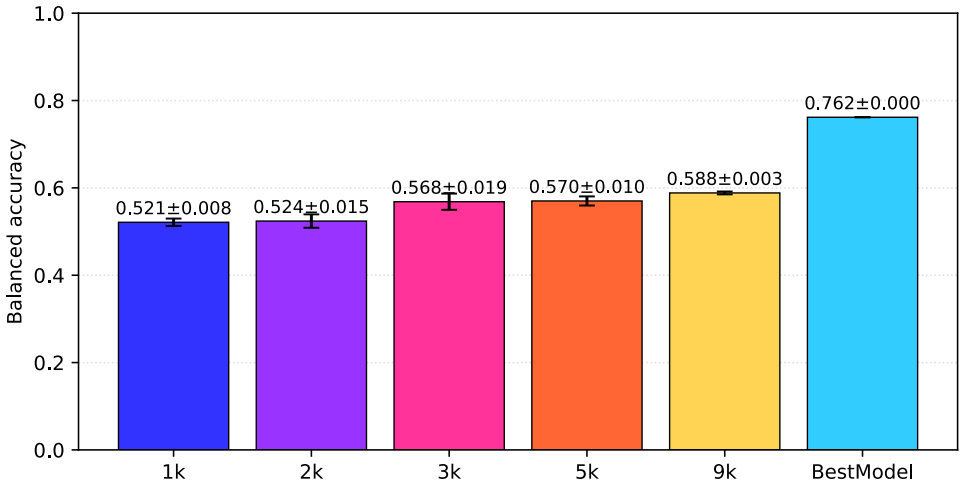


Figure 7.2: RQ1: BALANCED ACCURACY PER PRETRAINING SET SIZE. *Balanced accuracy on the SHHS test split for the sleep staging LSTM with the number of unlabeled subjects used to pretrain the foundation encoder. Each point corresponds to a pretrained encoder trained on one subject subset at that scale.*

AUPR at the clinical cutoff used in our experiments. This suggests that, under the current training recipe, the scratch model is better tuned to the heavily imbalanced AHI target than the downstream model.

The confusion matrices in Figure A.4 and the AHI curves in Figure 7.7 provide more nuance. Token-level performance on the 4-class breathing disorder classification and 3-class apnea type tasks is closer between the two models, and for some event types the downstream model reduces specific confusions, even though the baseline retains an overall advantage on the subject-level AHI classification. In other words, the foundation-based model does not yet consistently outperform the baseline once the task is aggregated to a single severity index per night.

H2b for apnea detection is not supported: with BAS+RESP inputs, the scratch Transformer baseline matches or exceeds the downstream model on most AHI-centric metrics, although token-level differences are more mixed. Therefore, overall RQ2 is partially supported.

7.3 RQ3: Handling Class Imbalance in Downstream Sleep Staging

To address RQ3, we train identical architectures on SHHS with different loss functions and compare per-class performance, focusing on how these losses impact minority versus majority sleep stages (Section 6.4 and Figure 4.2). Figure 7.8 compares the global staging metrics on the SHHS test split for the class-weighted CE model and the focal-loss model. Overall performance is similar: focal loss yields slightly higher F1 and Cohen's κ, while class-weighted CE attains slightly higher balanced accuracy, which is our primary metric.

Figure 7.9 shows how these aggregate differences break down by stage. Focal loss increases F1 for the minority stages N1 and REM, whereas Wake, N2, and N3 change only slightly relative

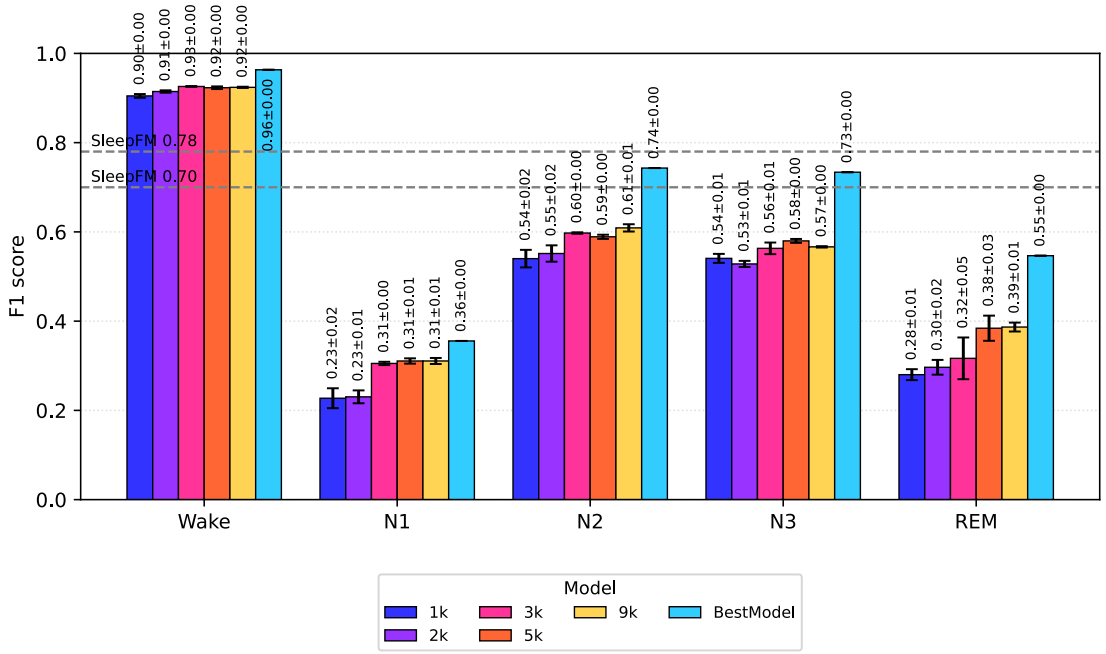


Figure 7.3: RQ1: PER-CLASS F1 PER PRETRAINING SET SIZE. *Per-class F1 scores for the five sleep stages on the SHHS test split grouped by the pretraining set size. Each group of points shows the performance of encoders pretrained at a given scale for one stage. The horizontal denotes the SleepFM results by Thapa et al. (2025) for the same task.*

to the CE baseline. The plot also includes per-stage F1 from SleepFM as horizontal reference markers. The confusion matrices in Figure A.5 show that these F1 gains for N1 and REM come with reduced recall for those stages, which explains the drop in balanced accuracy.

Overall, RQ3 shows that focal loss trades balanced accuracy for higher F1 and Cohen's κ, mainly by improving minority-stage precision at the cost of lower recall for N1 and REM.

7.4 RQ4: Contribution of Additional Modalities Beyond EEG

To address RQ4, we incrementally extend the input from BAS to BAS+respiratory, EMG, and EKG channels and measure downstream performance, comparing gains for sleep staging and apnea detection against the BAS-only baseline.

7.4.1 Sleep Staging

Figure 7.10 summarizes how balanced accuracy, F1 and Cohen's κ behave, when adding respiratory, EMG, and EKG channels on top of BAS. All multimodal configurations are very close to

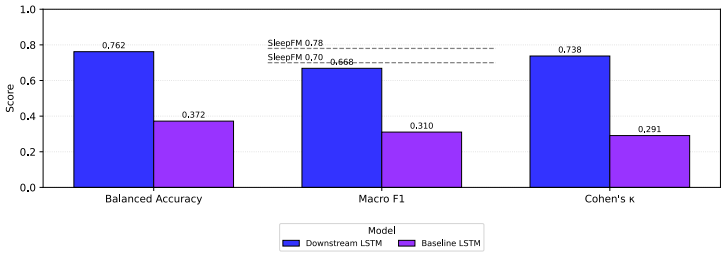


Figure 7.4: RQ2: SLEEP STAGING METRICS: PRETRAINED VS SCRATCH. *Balanced accuracy, F1, and Cohen's κ on the SHHS hold-out set for a model using frozen foundation-model embeddings (Downstream BAS) and the same end-to-end trained LSTM baseline from scratch on BAS-only inputs.*

BAS. Balanced accuracy decreases slightly when adding the respiratory modality to BAS and increases slightly when adding respiratory and EMG. Overall, the balanced accuracy stays within a narrow band for all combinations. F1 shows a similar pattern: the best configuration combines BAS, EKG and EMG, while adding RESP only yields the lowest F1. Cohen's κ behaves similarly, achieving the highest results with the combination of BAS, EKG, and EMG and with BAS, RESP, and EMG. Dropping EKG and RESP only remaining with BAS and EMG leads to slightly worse results.

Per-stage F1 scores in Figure 7.11 provide more detail. Wake, N2, and N3 are already strong with BAS alone and change little with extra modalities. The minority stage N1 shows small improvements in some modality settings, particularly EMG is present. For the second minority stage REM, the best F1 scores are obtained with BAS only. Overall, these changes are modest and not consistent across all configurations. The confusion matrices in Figure A.6 reflect the same behavior: the main error modes-confusions between N1 and N2 and between N3 and REM stages-remain essentially unchanged, with only slight shifts in recall across configurations.

Taken together, these results suggest that, after multimodal pretraining, BAS (EEG+EOG) already provides a strong representation for sleep staging. Additional EMG, respiratory, or EKG channels yield at most small, configuration-dependent gains. H4a is therefore not supported for sleep staging: extra modalities do not lead to a substantial or consistent improvement over BAS-only inputs.

7.4.2 Apnea Detection

Figure 7.12 depicts the effect of different modality combinations on night-level AHI estimation. BAS-only inputs yields the weakest performance. Using RESP alone already improves AHI discrimination, and adding BAS, EMG, or EKG on top of RESP further raises night-level performance. Configurations that lack RESP consistently sit at the bottom of the AHI metrics, whereas those that include RESP form a higher performing cluster.

Figure 7.13 reports the classification metrics for the token-level and night-level apnea tasks. For the 4-class breathing disorder and 3-class apnea type labels, balanced accuracy is similar across RESP-containing configurations and clearly reduced whenever RESP is omitted. For subject-level AHI-threshold and severity classification the same pattern appears: BAS-only is the weakest setting, RESP-only is markedly better, and combinations such as BAS+RESP (with or without EMG/EKG) yield the strongest results.

The confusion matrices in Figures A.7-A.9 make this more concrete. With BAS alone, many

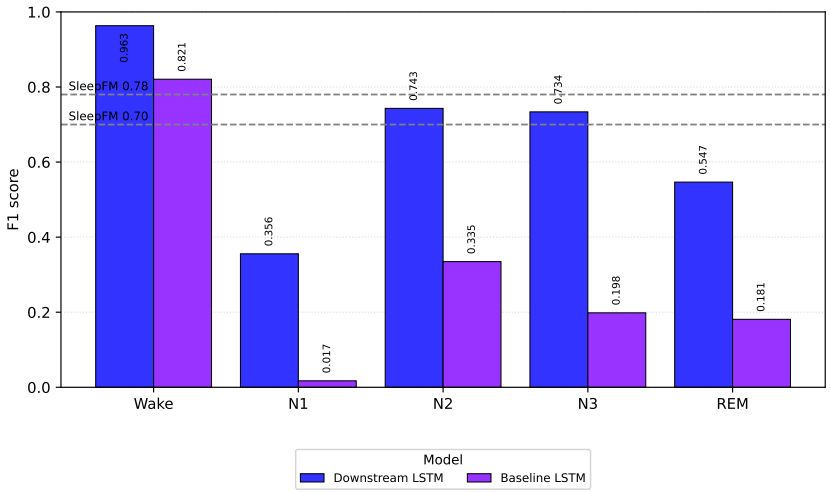


Figure 7.5: RQ2: PER-STAGE F1 FOR SLEEP STAGING MODELS. *Per-stage F1 scores on the SHHS hold-out set for the Downstream BAS model and the BAS-only LSTM baseline. Bars correspond to W, N1, N2, N3, and REM. The horizontal lines indicate the results from Thapa et al. (2025).*

apnea and hypopnea events are misclassified as normal breathing. Introducing RESP substantially increases the recall for apnea and hypopnea and reduces their confusion with the None class. Adding EMG and EKG on top of BAS+RESP refines the predictions further but does not qualitatively change the error structure.

In summary, H4b is supported for apnea detection: additional modalities — most importantly respiratory signals, and to a lesser extent EMG and EKG — do substantially improve downstream apnea performance compared with BAS-only inputs. Combined with the sleep staging results above, this shows that the RQ4 is partially supported and the benefit of extra modalities is task-dependent: small for staging, but large for apnea detection where respiration carries core task signal.

7.5 RQ5: Pretraining Objectives and Modality-Aware Training

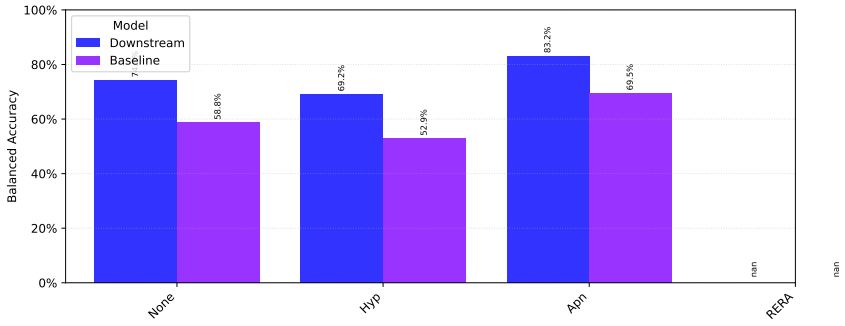
To address RQ5, we add an augmentation-based SimCLR term to the LOOC pretraining objective and compare downstream sleep staging performance under varying degrees of missing modalities across pretraining cohorts.

Figure 7.14 compares downstream sleep staging performance on SHHS for the two pretraining objectives: LOOC-only and LOOC+SimCLR, both evaluated with the frozen-encoder LSTM head from Section 5.2.2. At the aggregate level, balanced accuracy, F1, and Cohen's κ on the hold-out set are very similar for the two encoders, with differences well within one percentage point. Thus, adding a SimCLR term on top of LOOC does not yield a clear overall improvement in staging performance.

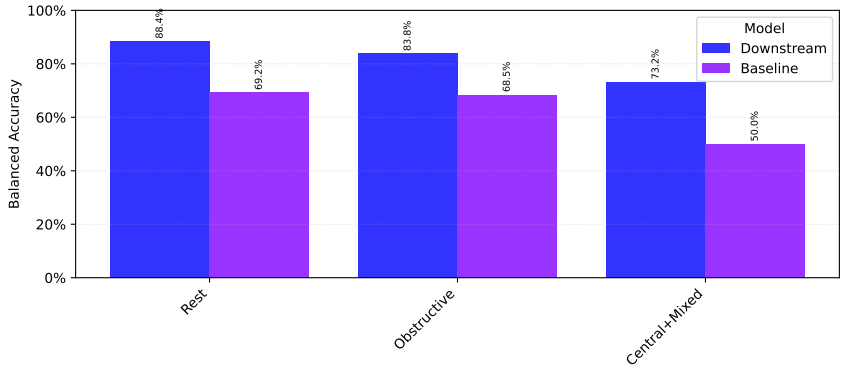
Per-stage F1 scores in Figure 7.15 show that the objectives mainly redistribute accuracy across

stages rather than producing gains. The LOOC-only model tends to score higher on the minority stage N1, on Wake, and N3 whereas the LOOC+SimCLR model yields improvements for N2 and REM. The confusion matrices in Figure A.10 reflect the same trade-off: SimCLR training reduces confusions among the deeper sleep and REM stages but at the cost of somewhat lower recall for Wake and N1.

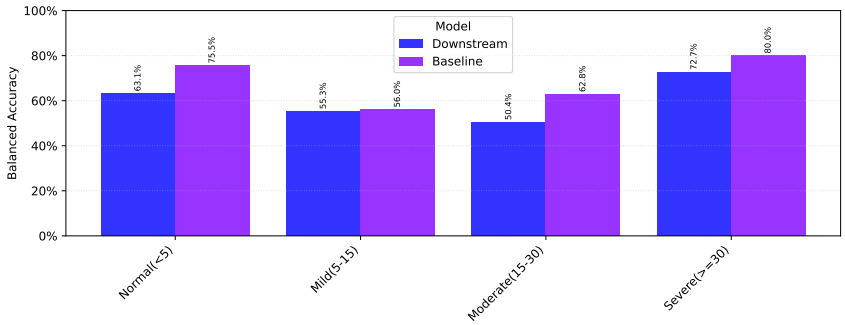
Overall, these results indicate that the SimCLR term does not substantially change downstream sleep staging accuracy compared with LOOC-only pretraining. Its main benefit in our setting is that it enables the use of single-modality and heavily channel-reduced cohorts during pretraining (Section 6.6), rather than providing a measurable gain in SHHS staging metrics. In this sense, H5a is not supported with respect to downstream performance improvements, however, RQ5 is partially supported because the modality-aware objective allows us to exploit more heterogeneous pretraining data.



(a) BA for Breathing Disorder Classification

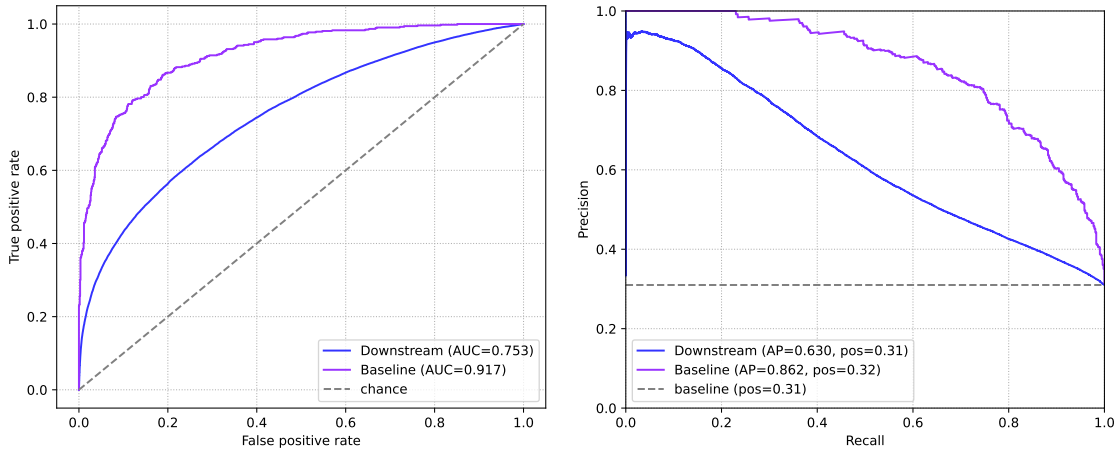


(b) BA for Apnea Type Classification



(c) BA for Severity Class

Figure 7.6: RQ2: APNEA DETECTION METRICS: PRETRAINED VS SCRATCH. *Per-class balanced accuracy (BA) on the SHHS hold-out set for apnea-related tasks, comparing the Downstream BAS+RESP model with a Transformer baseline trained from scratch on BAS+RESP inputs. Subfigure (a) shows BA for breathing disorder classification, Subfigure (b) shows BA for apnea type classification, and Subfigure (c) shows BA per severity class (night-level AHI categories).*



(a) ROC (subject) with AHI ≥ 15

(b) Precision-Recall (subject) with AHI ≥ 15

Figure 7.7: RQ2: SUBJECT-LEVEL AHI PERFORMANCE. *Subject-level ROC and precision-recall curves at the clinical AHI threshold used in our experiments, comparing the Downstream BAS+RESP model and the BAS+RESP Transformer baseline on the SHHS hold-out set. Subfigure (a) shows the ROC curves, and Subfigure (b) shows the corresponding precision-recall curves.*

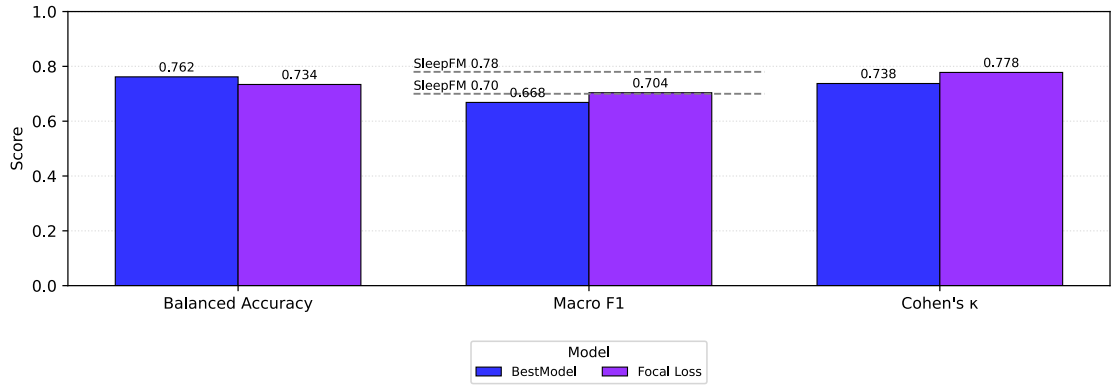


Figure 7.8: RQ3: GLOBAL STAGING METRICS FOR CE VS FOCAL LOSS. *Balanced accuracy, F1, and Cohen's κ on the SHHS hold-out set for the class-weighted CE model and the focal-loss model. The F1 score contains benchmarks from Thapa et al. (2025).*

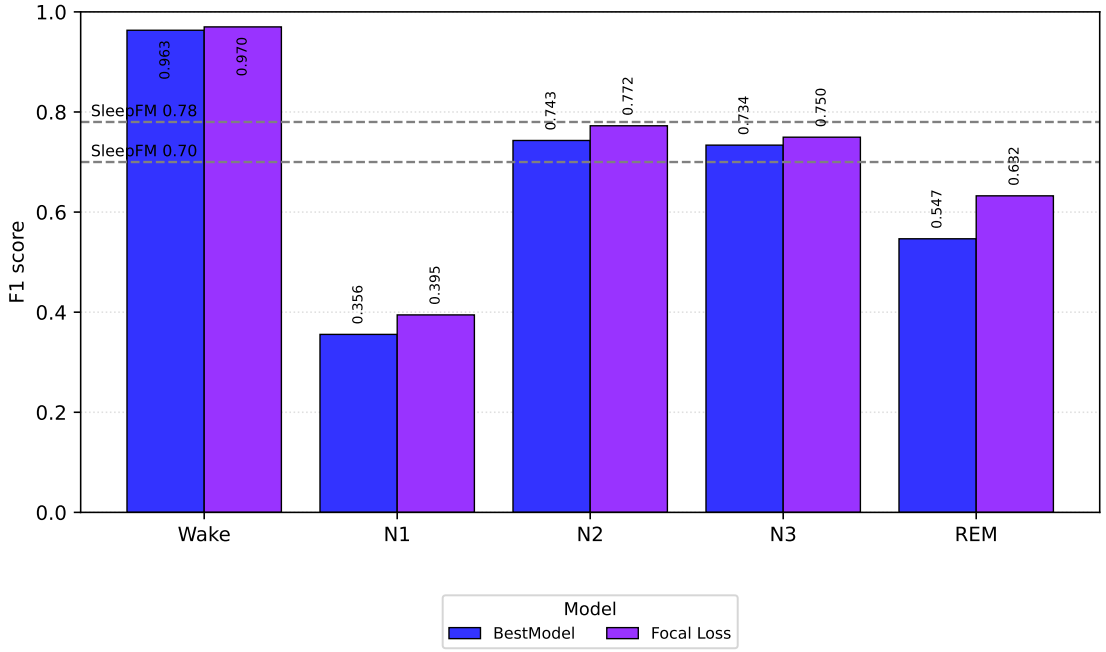


Figure 7.9: RQ3: PER-STAGE F1 FOR CE VS FOCAL LOSS. *Per-stage F1 scores on the SHHS hold-out set for the CE and focal-loss models. Horizontal markers indicate per-stage F1 of SleepFM, shown as a reference. The horizontal SleepFM line denotes the benchmarks from Thapa et al. (2025).*

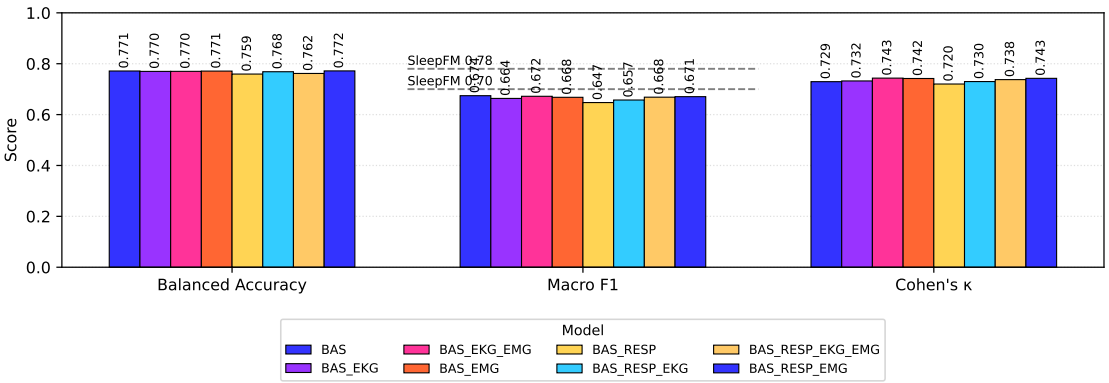


Figure 7.10: RQ4: SLEEP STAGING PERFORMANCE BY MODALITY CONFIGURATION. *Balanced accuracy and F1 on the SHHS hold-out set for different combinations of BAS, RESP, EMG and EKG inputs. Bars indicate overall scores per configuration, and lines and dashed bands show reference ranges for comparison (Thapa et al., 2025).*

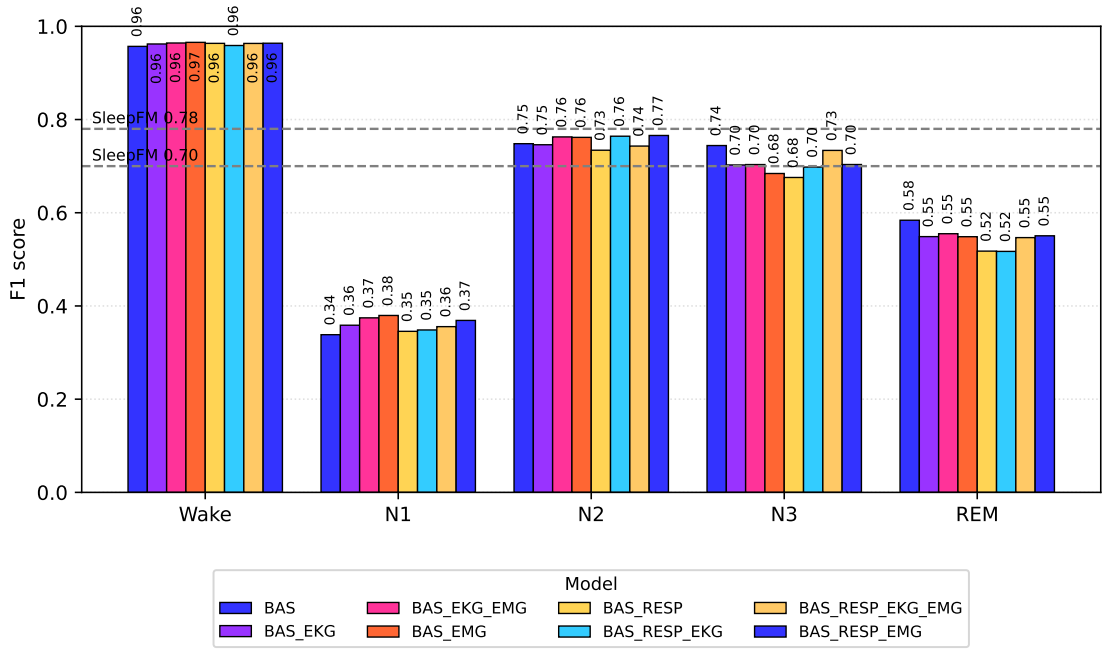
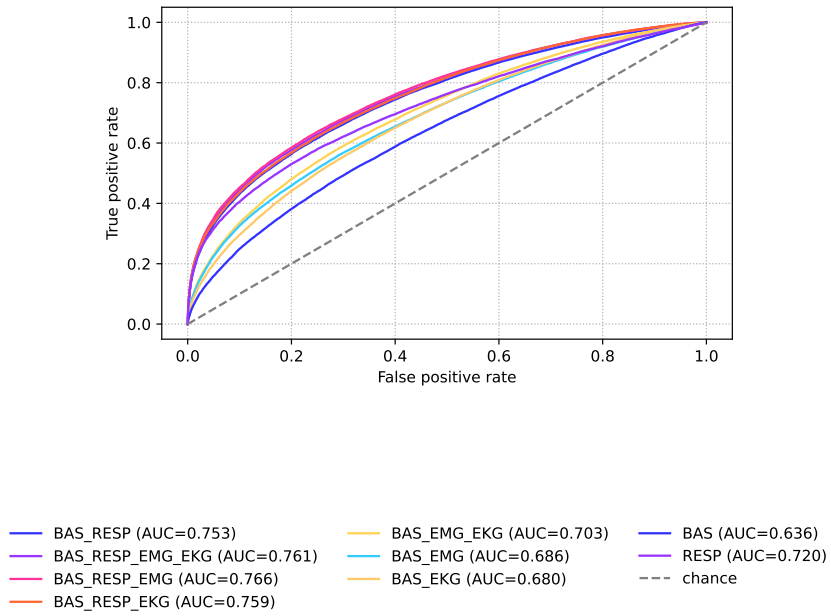
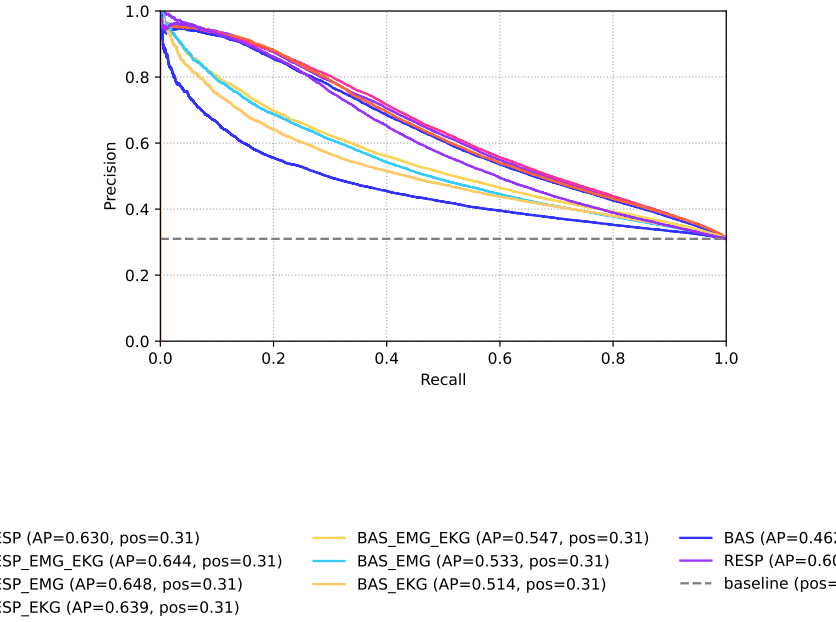


Figure 7.11: RQ4: PER-STAGE PERFORMANCE FOR DIFFERENT MODALITY CONFIGURATIONS. *Per-stage F1 scores for Wake, N1, N2, N3 and REM across modality combinations. Each group of bars corresponds to one configuration, and the colors indicate sleep stages.*

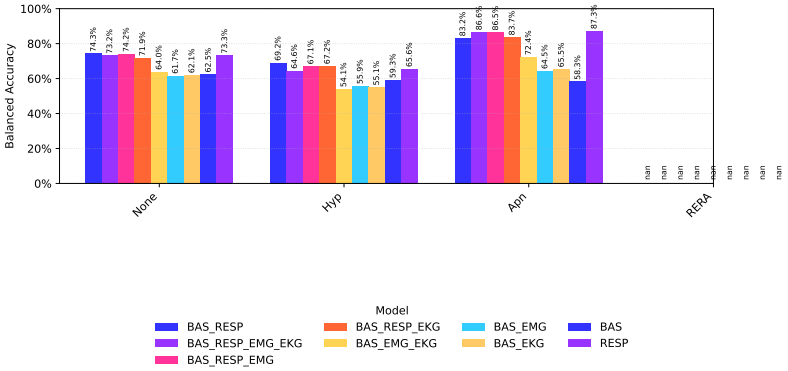


(a) ROC (subject) with AHI $\geq$ 15

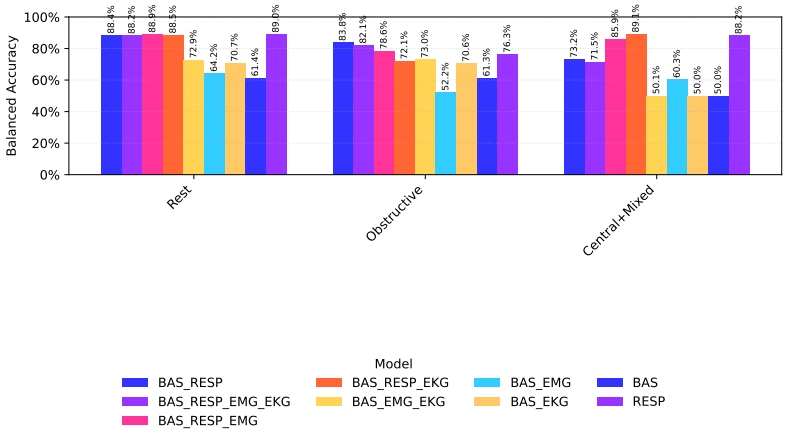


(b) Precision Recall (subject) with AHI $\geq$ 15

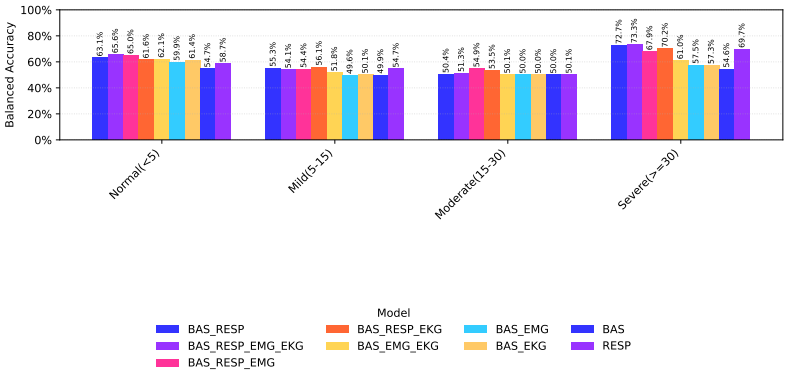
Figure 7.12: RQ4: NIGHT-LEVEL AHI SCREENING PERFORMANCE BY MODALITY. *ROC and precision-recall curves for screening at different AHI thresholds for each modality configuration. Subfigures (a) and (b) correspond to the clinical cut-off at AHI* $\geq$ *15 and compare the different multimodal input combinations.*



(a) Breathing Disorder BA (token)



(b) Apnea Type BA (token)



(c) Per-Class Severity BA (subject)

Figure 7.13: RQ4: APNEA DETECTION METRICS BY MODALITY CONFIGURATION. *Token-level and night-level performance metrics for apnea detection across modality combinations. Subfigures (a)-(c) show per-class balanced accuracy for breathing disorder, apnea type, and severity classes (night-level AHI categories), respectively, across all modality configurations.*

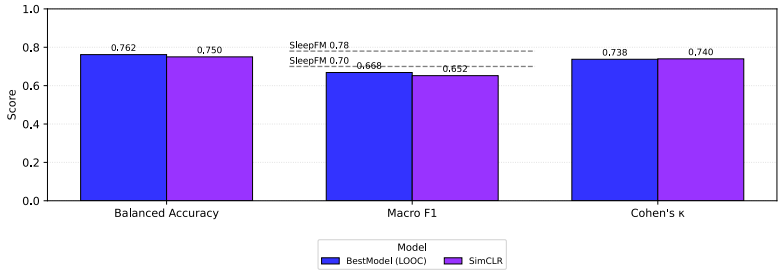


Figure 7.14: RQ5: SLEEP STAGE CLASSIFICATION METRICS BY CONTRASTIVE LOSS. *Epoch level performances on balanced accuracy, F1, and Cohen's κ for the different foundation model objectives. Horizontal lines on F1 indicate SleepFM results (Thapa et al., 2025).*

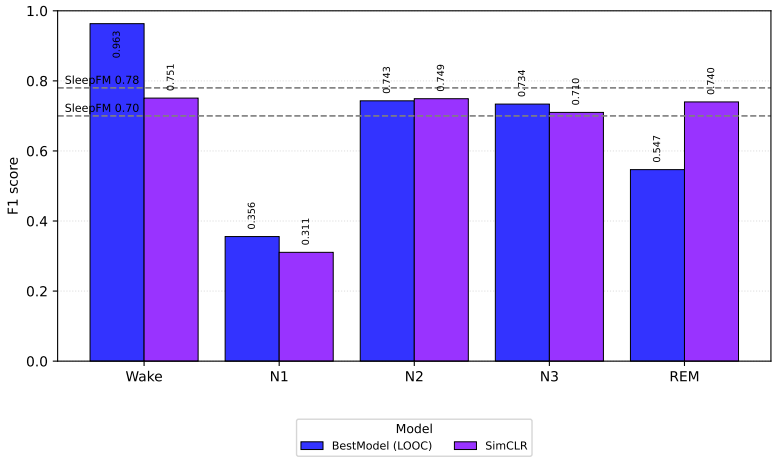


Figure 7.15: RQ5: PER-STAGE PERFORMANCE FOR DIFFERENT FOUNDATION MODEL ENCODER LOSS. *Per-stage F1 scores for Wake, N1, N2, N3 and REM on the SHHS hold-out set for two encoders: LOOC-only and LOOC+SimCLR. Each group of bars corresponds to one pretraining objective, colors indicate sleep stages, highlighting how the additional SimCLR term slightly redistributes performance across classes.*

Chapter 8

Discussion

This chapter interprets the findings of Chapter 7, ties them back to the research questions and puts them into context. We then highlight challenges and limitations of our work.

8.1 Interpretation of Experimental Findings

In the following, we revisit each research question in light of the results and discuss what they imply for scalable sleep foundation models.

8.1.1 Pretraining Scale

Experiments under RQ1 show that more unlabeled PSG generally leads to better downstream sleep staging. As the pretraining pool grows from 1k subjects to 13'526 subjects (BestModel), balanced accuracy and F1 on SHHS test split increase steadily (Figures 7.1-7.2). The confusion matrices and per-stage F1 curves indicate that larger pretraining sets mainly reduce errors around N1 and REM, where the model is otherwise prone to confuse light sleep with neighboring stages. This suggests that additional heterogeneous nights help the encoder build more reliable representations for rare and ambiguous states that are under-represented in the labeled downstream cohort. The scaling curves do not show a clear plateau within the explored range. Performance keeps improving up to 9k subjects and shows a further jump for the BestModel (full dataset of 13526 subjects), rather than flattening out. We therefore have no obvious "enough data" point. Even with more than ten thousand subjects, extra unlabeled PSG brings measurable gains. Similar "more data helps" trends have been reported for large foundation models in sleep and vision (He et al., 2022; Thapa et al., 2025).

These RQ1 findings must be read in light of our fixed pretraining recipe. All encoders share the same architecture, LOOC objective, optimizer, temperature, and training schedule (10 epochs or a 15-hour cap per model; Section 6.2), following the original SleepFM configuration (Thapa et al., 2025). We do not know whether smaller pools could close part of the gap with longer training or stronger regularization, or whether even larger pools would keep improving under a relaxed compute budget. In the present setup, a pragmatic conclusion is that using as much diverse unlabeled PSG as available consistently helps downstream sleep staging, especially for minority stages, and we have not observed a clear performance ceiling yet.

8.1.2 Transfer Learning vs Training from Scratch

The RQ2 results for sleep staging are clear: with BAS inputs, the LSTM on top of the pretrained encoder achieves substantially higher balanced accuracy and F1 than the BAS-only baseline (Section 7.2). Figures 7.4-7.5 and Figure A.3 show that these gains span all stages. Confusions between light non-REM stages and between REM and Wake shrink substantially, and minority stages that the scratch model almost never predicts become reliably detectable. This aligns with the motivation behind PSG foundation models: pretraining on diverse cohorts teaches the encoder general sleep patterns that transfer well to a new cohort and montage (Thapa et al., 2025; Fox et al., 2025). At the same time, the BAS-only baseline fails to predict several minority classes, suggesting poorly calibrated decision boundaries, possibly due to sensitivity to initialization and class imbalance, even though the training pipeline is identical across experiments. A more detailed analysis of this failure mode is left for future work due to time constraints.

For apnea detection the picture is more mixed. The downstream Transformer on BAS+RESP embeddings reaches similar token-level performance to the scratch baseline, but lags behind on subject-level AHI screening (Figures 7.6-7.7). A likely reason is that our LOOC objective is agnostic to apnea events: it aligns modalities at the epoch level but never uses event labels or the sharp thresholds that define clinical AHI. The scratch model, in contrast, can specialize directly to the SHHS label distribution and decision boundary, even if its internal representation is less reusable.

In our experiments, SleepFM pretraining is consistently beneficial for BAS-only sleep staging, and it also improves token-level apnea predictions (*e. g.* obstructive vs central vs rest) compared to training from scratch. However, the same pretrained encoder does not translate into better subject-level AHI screening: the strongest AHI results from our experiments still come from models trained end-to-end on respiratory channels with an apnea-focused loss. Thus, the value of pretraining is apnea-task dependent.

8.1.3 Handling Class Imbalance

Experiments on RQ3 compared two simple ways of handling the strong class imbalance in SHHS sleep staging: inverse-frequency class weighting in the cross-entropy loss and the same weighting combined with focal loss (Section 7.3). At the global level the models are close. The focal-loss variant yields slightly higher F1 and Cohen's κ, whereas class-weighted cross-entropy attains the best balanced accuracy (Figure 7.8). This already suggests a trade-off: focal loss spreads performance more evenly across stages, but does not increase overall recall when each stage counts equally.

Stage-wise metrics make this clearer. In Figure 7.9, focal loss raises F1 for the minority stages N1 and REM, while Wake, N2, and N3 change little. The confusion matrices in Figure A.5 show that the focal model predicts N1 and REM less often but with higher precision. Fewer false positives improve F1, yet the stricter behavior also reduces recall, which in turn hurts balanced accuracy. Thus focal loss sharpens the decision boundary for rare stages without fundamentally resolving their ambiguity.

Practically, this means that simple class weighting is already a robust default, and focal loss is mainly attractive when minority-stage precision matters more than overall balanced accuracy (Lin et al., 2020). Both variants share the same pretrained encoder, so most differences come from the loss and head. The residual confusion for N1 and REM is consistent with known inter-rater variability for these stages, suggesting that label noise and signal quality, not just the choice of loss, contribute to the remaining errors (Lee et al., 2022).

8.1.4 Value of Additional Modalities

RQ4 examined how much additional modalities beyond BAS contribute once the encoder has been pretrained multimodally. For sleep staging the effect is small. Across all modality combinations, balanced accuracy, F1, and Cohen's κ remain close to the BAS-only model (Section 7.4). Figure 7.10 shows no configuration that clearly surpasses BAS, and per-stage F1 in Figure 7.11 indicates that Wake, N2, and N3 are already modeled well. EMG gives slight gains for N1 in some settings, whereas respiratory and ECG channels bring no stable improvement. The confusion matrices in Figure A.6 are likewise similar, suggesting that extra sensors mostly add redundant or noisy information for staging.

Apnea detection tells a different story. Respiratory channels are central: modality settings that omit the RESP modality show the weakest performance on token-level events and night-level AHI or severity, whereas any configuration including RESP forms a clearly stronger cluster (Figures 7.12-7.13). Adding BAS, EMG, or ECG on top of RESP yields smaller gains, especially for separating apnea mechanisms and for subject-level screening. The confusion matrices in Figures A.7-A.9 illustrate how RESP sharply reduces misclassification of apneas and hypopneas as normal breathing, matching clinical practice where airflow, belts, and oximetry carry most diagnostic signal (Berry et al., 2017; Hu et al., 2025).

Together, these findings indicate that the marginal value of extra modalities is strongly task dependent. For staging, BAS-only models are attractive because they are simpler to deploy yet reach near-best performance. For apnea assessment, respiratory sensors are indispensable, and additional modalities mainly provide refinements. In a multi-task system it may therefore be reasonable to tailor the sensor set to the main clinical question instead of always using a maximal montage.

8.1.5 Pretraining under SimCLR

In RQ5, adding a SimCLR-style augmentation term on top of the LOOC loss leaves SHHS sleep staging performance almost unchanged (Section 7.5). Global metrics differ by around one percentage point, and per-stage F1 scores mainly show small shifts of accuracy between stages rather than a clearly better encoder (Figures 7.14-7.15 and Figure A.10).

We take this as a sign that the LOOC objective of Thapa et al. (2025) already provides a strong contrastive signal for multimodal PSG and that, in our current setup, adding SimCLR-style augmented pairs neither reliably improves nor harms transfer. Given these small, inconsistent effects, we cannot say whether any apparent advantage of one objective is meaningful or just noise in our data. Accordingly, H5a is not supported for sleep staging.

The SimCLR term still has a practical benefit. LOOC requires at least three modalities per epoch, so cohorts that provide only BAS (EEG/EOG) or otherwise lack two modality groups cannot contribute a LOOC loss and would otherwise be discarded during pretraining (Section 6.6). In contrast, the SimCLR loss can be computed as soon as at least one channel is present, so single-modality and heavily reduced montages can still influence the encoder. This allows us to include structurally incomplete cohorts such as HMC and CAP. Since our largest datasets already satisfy the LOOC requirement, the extra data from these smaller cohorts may simply be too limited to noticeably change SHHS performance. All pretraining hyperparameters were kept fixed, so we do not know whether different weightings or stronger augmentations would make SimCLR more beneficial. Under the present recipe, LOOC alone is sufficient for strong SHHS sleep staging, while SimCLR mainly serves as a modality-aware way to include more cohorts rather than a direct route to higher staging accuracy.

8.2 Cross-Dataset Challenges

A central motivation for this work was the observation that single-cohort models often fail to generalize well across sleep laboratories, hardware, and scoring conventions (Alvarez-Estevez and Rijsman, 2021). The pretraining pool in Chapter 4 therefore combines ten cohorts with different montages, sampling rates, and annotation standards, ranging from community-based studies to specialized clinical recordings. This heterogeneity poses an opportunity but also a challenge. Without careful harmonization, differences between cohorts can degrade cross-dataset performance (Alvarez-Estevez and Rijsman, 2021), and a naive pooled training setup may in practice overfit to one dominant cohort instead of learning truly cohort-agnostic features.

On the positive side, multi-cohort pretraining exposes the encoder to a much wider range of EEG morphologies, respiratory patterns, and sleep architectures than any single cohort would provide. The data-efficiency results of RQ1 suggest that scaling pretraining to several thousand subjects is beneficial for downstream staging, and that the encoder can learn a modality- and dataset-agnostic representation that transfers reasonably well to a new cohort with different demographics and hardware.

At the same time, our experiments highlight how strongly downstream performance still depends on the target dataset. All evaluations in Chapter 7 are carried out on SHHS, which is a large community-based cohort of adults aged 40 years and older (Quan et al., 1997). It remains unclear how well the same encoder and heads would perform on smaller datasets such as HMC when used as test sets rather than pretraining sources. Prior inter-database studies report that models which perform well on their local cohort can show substantial drops in performance when evaluated on external databases addressing the same task (Alvarez-Estevez and Rijsman, 2021), and nothing in our experiments guarantees that the foundation encoder fully overcomes these shifts.

Our modality experiments also tie into cross-dataset issues. Many cohorts use different sensor placements or lack some channels. The LOOC objective and channel-group mapping are designed to absorb some of this variation by operating at the level of modality sets rather than fixed channel identities, and the addition of SimCLR in RQ5 further helps by allowing datasets with at least one modality to participate in pretraining. Nevertheless, the fact that respiration is indispensable for apnea detection but largely irrelevant for staging means that source datasets without good respiratory signals are more informative for staging than for apnea.

Moreover, another challenge is the skewed population and pathology composition across cohorts. Our largest pretraining datasets are dominated by older adults, and MrOS in particular consists of community-dwelling men aged 65 years and older with a high prevalence of moderate-severe sleep-disordered breathing (Redline et al., 1995; Blackwell et al., 2011). The encoder is therefore exposed far more often to age-related sleep architectures than to recordings from younger or healthier individuals. In older adults, sleep is typically characterized by reduced deep sleep, lower sleep efficiency, and more awakenings compared to younger adults (Mander et al., 2017). This may bias the learned representations toward older, high-risk populations.

Lastly, since we use datasets that are not of the same size, we have a substantial dataset size imbalance that further skews learning. A few studies dominate the pool (see Figure 4.1). In our pretraining set, MrOS and MESA contribute most of the total recording time, while smaller datasets (CAP, Haaglanden, Sleep-EDF) have comparatively few nights. Without countermeasures, smaller cohorts have little influence on the loss and thus on the learned representation.

8.3 Limitations

This work has several limitations that qualify the scope of our conclusions. First, all downstream evaluations are performed on a single cohort. Although the encoder is pretrained on ten hetero-

geneous cohorts, our downstream results only quantify performance and robustness on SHHS, so claims about generalizability to other hospitals, age groups, or scoring practices remain indirect.

Second, the pretraining pool is strongly imbalanced. A few large cohorts such as MrOS and MESA contribute most of the total recording time, whereas several datasets (for example CAP, HMC, Sleep-EDF) provide comparatively few nights (see Figure 4.1). We sample uniformly over subjects and do not explicitly reweight datasets, so smaller cohorts have limited influence on the learned representation, and the encoder may implicitly specialize to the dominant studies and their populations.

Third, we restrict self-supervised pretraining to contrastive objectives (LOOC and an auxiliary SimCLR term). Masked reconstruction and related autoencoding objectives have shown promising results for EEG and time-series representation learning (Cai and Zeng, 2024), but are not explored here, so it remains unclear whether they would yield more robust or more interpretable sleep embeddings in our setting.

Fourth, our downstream experiments use a narrow set of architectures. All sleep-staging experiments use a single bidirectional LSTM head on top of frozen embeddings, and apnea detection relies on one specific Transformer head. We do not systematically compare alternative heads such as purely Transformer-based staging models, convolutional decoders, or lightweight architectures targeted at deployment. Part of the performance gap to prior work may therefore be due to head design rather than to the foundation encoder itself.

Finally, we evaluate only two downstream tasks, sleep staging and apnea detection. While these are central to clinical sleep medicine, they represent only a subset of the analyses that might benefit from a PSG foundation model, such as insomnia phenotyping, periodic limb movement detection, or long-term risk prediction.

8.4 Future Work

Several directions follow naturally from these limitations. A first step is to broaden the objective space beyond contrastive learning. Extending LOOC with masked reconstruction or related autoencoding tasks could encourage the encoder to model fine-grained waveform structure, improve calibration, and make the learned representations easier to interpret.

Addressing dataset imbalance is another priority. Future work could explore cohort-aware sampling schemes, explicit loss reweighting at the study level, or strategies that over-sample under-represented cohorts.

On the evaluation side, a more systematic re-examination of the scratch baselines, including checks of class weighting and optimization settings, would help confirm that the observed transfer gains are not confounded by implementation details.

On the architectural side, it would be valuable to systematically compare different downstream heads such as pure Transformer-based sleep-staging models. Finally, we only touched two downstream tasks. Applying the same encoder to a broader task portfolio, such as arousal and insomnia classification, or prediction of cardiovascular outcomes, would test how far a single PSG foundation model can be pushed in practice.

Chapter 9

Conclusion

This thesis set out to explore how far a single, contrastively pretrained foundation encoder can be pushed for heterogeneous PSG data. Building on the SleepFM set-then-sequence architecture, we trained a multimodal encoder on ten public cohorts and evaluated its frozen representations on two downstream tasks on SHHS: sleep staging and sleep apnea detection.

We found that scaling pretraining from one to over ten thousand subjects led to consistent gains in downstream sleep staging, with F1, balanced accuracy, and minority-stage performance all improving and showing diminishing but not saturated returns at the largest scale. We then showed that, for sleep staging, an LSTM trained on frozen BAS embeddings achieved substantially higher balanced accuracy and F1 than an otherwise identical BAS-only baseline trained from scratch, demonstrating clear benefits of foundation-style pretraining. For apnea detection, however, the scratch Transformer remained competitive or even superior on AHI-centric metrics, suggesting that our encoder and head are not yet optimally matched to this task.

We then investigated class imbalance in sleep staging, where we compared class-weighted CE and focal loss. The models yielded similar global metrics but different trade-offs: focal loss improved F1 for N1 and REM at the cost of slightly lower balanced accuracy. Nonetheless, addressing the imbalance was crucial for fair performance across all sleep stages. In our next experiment, we compared modality configurations and showed a strong task dependence: for staging, BAS (EEG+EOG) alone already captured most of the useful information, while additional EMG, respiratory, or ECG channels provided at best modest gains. In contrast, respiratory signals were indispensable for apnea detection. Configurations lacking RESP performed poorly on AHI screening, whereas BAS+RESP (with or without EMG/ECG) markedly improved both token-level and night-level metrics. Lastly, we compared LOOC-only pretraining with a LOOC+SimCLR objective. Both achieved similar downstream staging performance, with SimCLR primarily enabling inclusion of single-modality cohorts rather than yielding clear accuracy gains.

Overall, these findings support the promise of PSG foundation models while also clarifying their current limitations. Contrastive pretraining across heterogeneous cohorts substantially improves sleep staging and offers a label-efficient path toward clinically useful models. At the same time, performance remains tied to the target dataset, the choice of downstream head, and task-specific modality needs. Future work should therefore explore masked reconstruction and other autoencoding objectives, alternative downstream architectures, true cross-cohort testing, and additional tasks such as arousal, insomnia, or long-term risk prediction, moving toward a single PSG foundation model that robustly serves diverse clinical and research applications.

Appendix A

Appendix

Supplementary Tables for Datasets

Table A.1: MESA CHANNELS AND ACQUISITION SUMMARY. *MESA (Multi-Ethnic Study of Atherosclerosis) PSG channels grouped by modality and sampling rate (Chen et al., 2015; Zhang et al., 2018).*

Modality	Sensors	Sampling rate (Hz)
BAS	EEG (Fz-Cz, Cz-Oz, C4-M1), EOG (L-Fpz, R-Fpz)	256
EMG	Chin EMG, leg EMG (piezo limb movement)	256 (chin), 32 (leg)
EKG	ECG	256
RESP	Nasal pressure flow, thorax and abdomen effort belts, position, thermistor, snore microphone, SpO_2	32 (flow, belts, position, thermistor, snore), 1 (SpO_2)
OTHER	Plethysmography (pulse waveform), heart rate (derived)	256 (plethysmography), 1 (heart rate)

Table A.2: MROS CHANNELS AND ACQUISITION SUMMARY. *MrOS Sleep Study PSG channels and acquisition summary (Blackwell et al., 2011; Zhang et al., 2018).*

Modality	Sensors	Sampling rate (Hz)
BAS	EEG (C3-Fpz, C4-Fpz), EOG (LOC-Fpz, ROC-Fpz), A1, A2	256
EMG	Chin EMG (L, R, center), leg EMG (L, R)	256 (chin), 64 (leg)
EKG	ECG (L, R)	512
RESP	Airflow (thermistor), thorax and abdomen effort belts, sum, SpO_2	16 (airflow, belts, sum), 1 (SpO_2)
OTHER	Heart rate, oximetry status	1

Table A.3: MASS SS3 CHANNELS AND ACQUISITION SUMMARY. *MASS SS3 (Montreal Archive of Sleep Studies, session 3) channels and acquisition summary. Sampling rates follow O'Reilly et al. (2014).*

Modality	Sensors	Sampling rate (Hz)
BAS	EEG (4-20 channels, lab-dependent), EOG (L, R)	256
EMG	Chin EMG, leg EMG	256
EKG	ECG/EKG	256
RESP	Thermistor, RIP bands (where present), SpO_2	256 or lower (varies)
OTHER	Position (where present)	varies

Table A.4: SLEEP-EDF CHANNELS AND ACQUISITION SUMMARY. *Sleep-EDF (Expanded) channels and sampling rates (Kemp et al., 2000; Goldberger et al., 2000).*

Modality	Sensors	Sampling rate (Hz)
BAS	EEG (Fpz-Cz; Pz-Oz)	100
BAS	EOG (horizontal, LOC-ROC)	100
EMG	Chin EMG (submental)	100 (ST) / 1 (SC)
RESP	Oro-nasal thermistor airflow	1 (SC subset)
OTHER	Rectal body temperature	1 (SC subset)
OTHER	Event marker (subject button)	1

Table A.5: WSC CHANNELS AND ACQUISITION SUMMARY. *WSC (later recordings, Grass Comet PSG system) channels and acquisition summary (Young et al., 2009; Zhang et al., 2018).*

Modality	Sensors	Sample rate (Hz)
BAS	EEG (F3-M2, Fz-M2, Cz-M2, C3-M2, Pz-M2, O1-M2), EOG (E1-M2, E2-M1)	200
EMG	Chin EMG (bipolar, linked), Leg EMG (bilateral, linked)	200
EKG	ECG	200
RESP	Airflow (nasal pressure + thermistor), Thoracic RIP, Abdominal RIP, RIP sum	200
OTHER	Snore microphone, Body position sensor	200 (SpO_2 averaging)

Table A.6: MNC CHANNELS AND ACQUISITION SUMMARY. *MNC PSG channels and acquisition summary, based on common examples from the NSRR montage page (Stephansen et al., 2018; Zhang et al., 2018).*

Modality	Sensors	Sampling rate (Hz)
BAS	EEG (F3-Fpz, F4-Fpz, C3-Fpz, C4-Fpz, Cz-Fpz), EOG (E1-Fpz, E2-Fpz), M1, M2	128
EMG	Chin EMG (cchin, lchin, rchin), leg EMG (lleg, rleg, linked)	128-512 (chin), 128-256 (leg)
EKG	ECG, ECG1/2/3, linked ECG1_2	200-256
RESP	Airflow, nasal pressure, thermistor, thorax and abdomen effort belts, sum, plethysmography, SpO_2	16-100
OTHER	Heart rate, pulse transit time, position, light	varies

Table A.7: HMC CHANNELS AND ACQUISITION SUMMARY. *HMC (Hospital MC) PSG channels and acquisition summary (Alvarez-Estevez and Rijsman, 2021; Goldberger et al., 2000).*

Modality	Sensors	Sampling rate (Hz)
BAS	EEG (F4-M1, C4-M1, O2-M1, C3-M2), EOG (E1-M2, E2-M2)	256
EMG	Chin EMG	256
EKG	ECG	256

Table A.8: CAP CHANNELS AND ACQUISITION SUMMARY. *CAP (Cyclic Alternating Pattern) sleep database PSG channels and acquisition summary, based on the dataset overview (Terzano et al., 2001; Goldberger et al., 2000).*

Modality	Sensors	Sampling rate (Hz)
BAS	EEG (F3/F4, C3/C4, O1/O2 vs A1/A2), EOG (L, R)	varies by record
EMG	Chin EMG, tibial EMG (bilateral)	varies by record
EKG	ECG/EKG	varies by record
RESP	Airflow, thoracic and abdominal effort, SaO_2	varies by record

Table A.9: STAGES CHANNELS AND ACQUISITION SUMMARY. *STAGES (Sleep Training and Evaluation Study) PSG channels and acquisition summary (Zhang et al., 2018). Sampling characteristics vary across contributing sites.*

Modality	Sensors	Sampling rate (Hz)
BAS	EEG (multi-derivation), EOG (bilateral)	varies by site
EMG	Chin EMG, leg EMG	varies by site
EKG	ECG/EKG	varies by site
RESP	Nasal/oral airflow, chest movements	varies by site
OTHER	Position, SpO_2, questionnaires (non-PSG measures)	varies

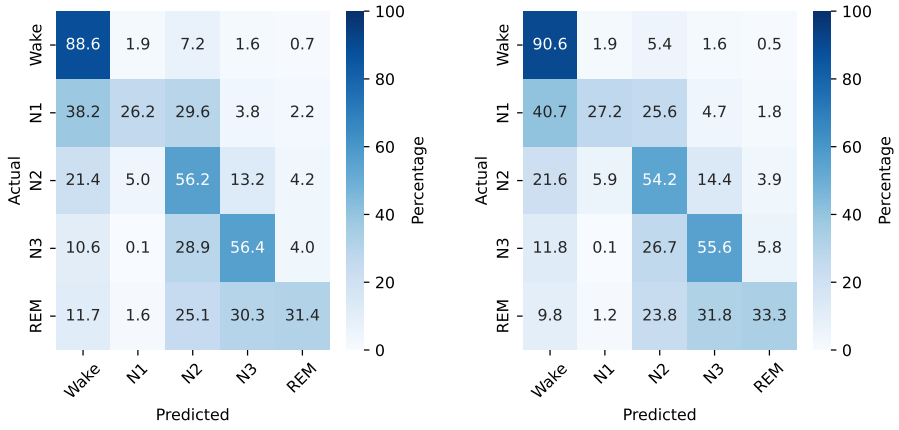
Table A.10: CFS CHANNELS AND ACQUISITION SUMMARY. *CFS (Cleveland Family Study) PSG channels and acquisition summary (Redline et al., 1995; Zhang et al., 2018).*

Modality	Sensors	Sampling rate (Hz)
BAS	EEG (C3-Fpz, C4-Fpz), EOG (LOC-Fpz, ROC-Fpz), A1-Fpz, A2-Fpz	128
EMG	Chin EMG (EMG1, EMG2, EMG3), leg EMG (L Leg-Fpz, R Leg-Fpz)	256 (chin), 64 (leg)
EKG	ECG1, ECG2	256
RESP	Airflow, thorax and abdomen effort belts, sum	32
OTHER	Snore, plethysmography, SpO_2, pulse, oximetry status, position	256 (snore), 128 (plethysmography), 1 (SpO_2, pulse, oximetry status, position)

Table A.11: SHHS CHANNELS AND ACQUISITION SUMMARY. *SHHS PSG channels and acquisition summary (Quan et al., 1997; Zhang et al., 2018).*

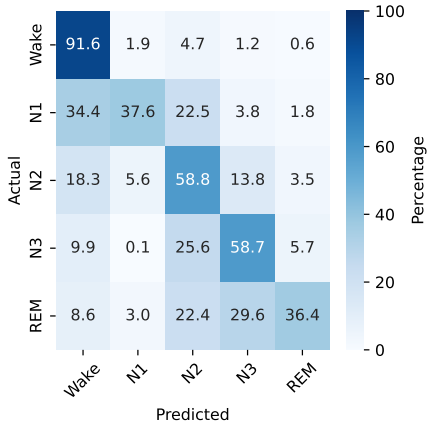
Modality	Sensors	Sampling rate (Hz)
BAS	EEG (C3-A2, C4-A1), EOG (L-PG1, R-PG1)	125 (EEG), 50 (EOG)
EMG	Chin EMG	125
EKG	ECG	125
RESP	Airflow (thermistor), thorax and abdomen effort belts	10
RESP	SpO_2	1
OTHER	Position, heart rate, oximetry status, light	1

Figures for RQ1

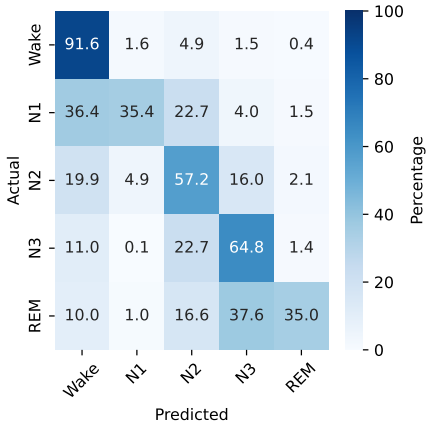


(a) 1k Subjects

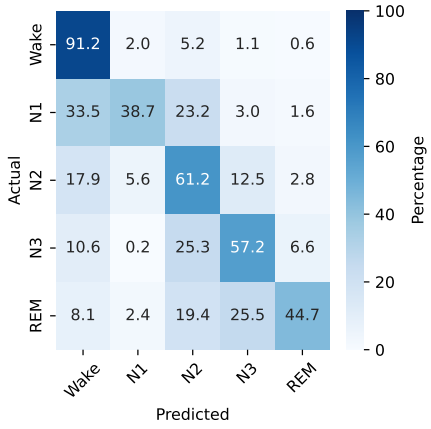
(b) 2k Subjects



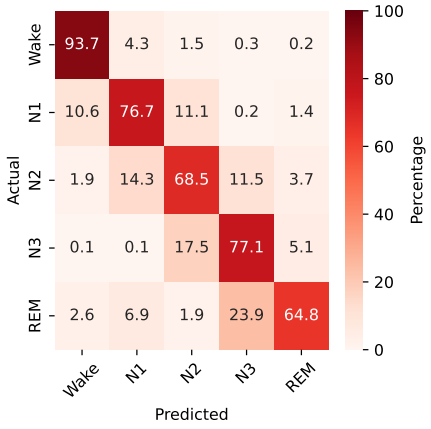
(c) 3k Subjects



(d) 5k Subjects



(e) 9k Subjects



(f) 13k Subjects (BestModel)

Figure A.1: RQ1: CONFUSION MATRICES ACROSS PRETRAINING SCALES. *Confusion matrices for sleep staging on the SHHS test split for different numbers of pretraining subjects. Subfigure (a)-(f) show the different pretraining split sizes. Each confusion matrix is row-normalized and shows percentages.*

Supplementary Figures

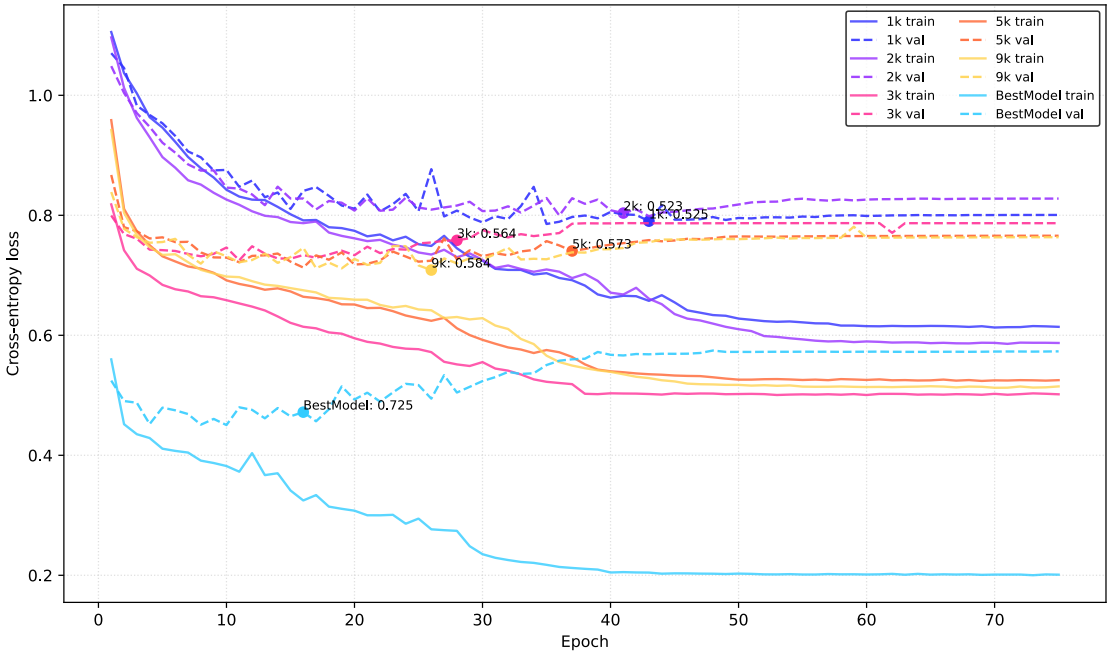
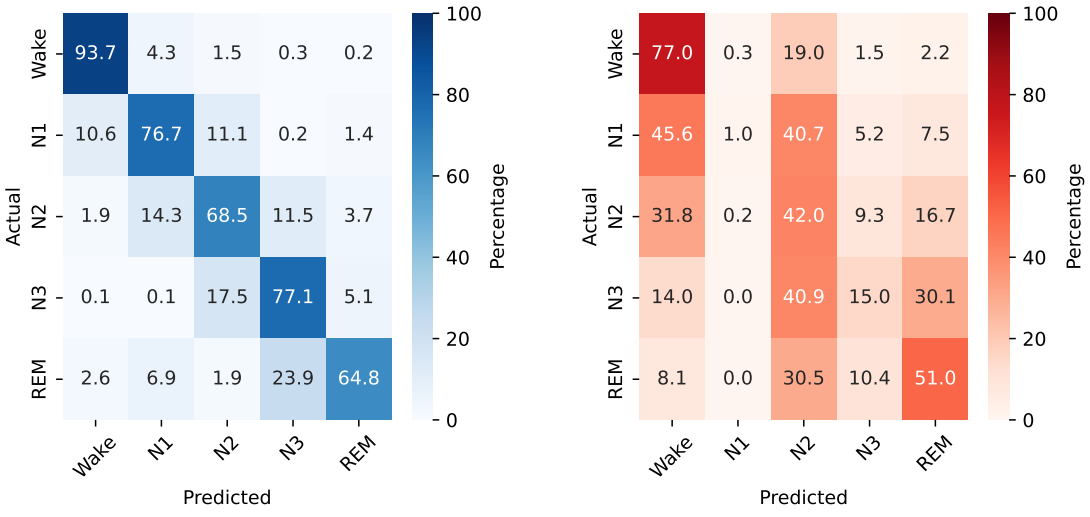


Figure A.2: DOWNSTREAM TRAINING AND VALIDATION. *Training and validation loss curves for downstream sleep staging models pretrained on increasing dataset sizes. BestModel shows lowest final loss and fastest convergence.*

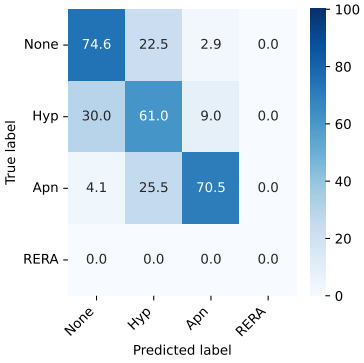
Figures for RQ2



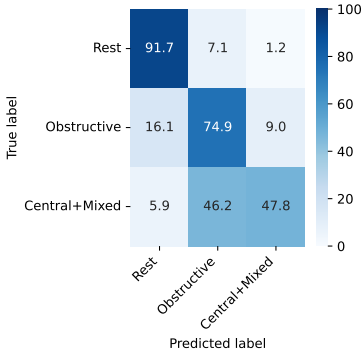
(a) Downstream LSTM

(b) Baseline LSTM

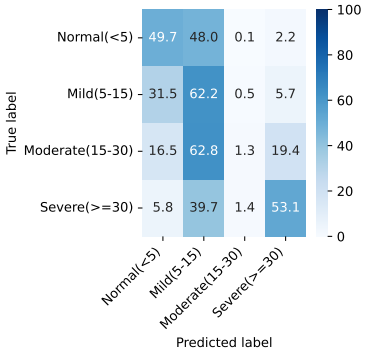
Figure A.3: RQ2: SLEEP STAGING CONFUSION MATRICES. *Confusion matrices on the SHHS hold-out set for the Downstream BAS model and the BAS-only LSTM baseline. Subfigure (a) shows the confusion matrix for the Downstream LSTM, and Subfigure (b) shows the confusion matrix for the Baseline LSTM. Each confusion matrix is row-normalized and shows percentages.*



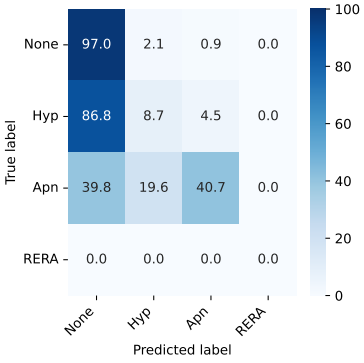
(a) Downstream Breathing Disorder (token)



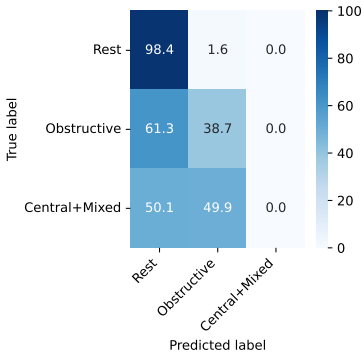
(b) Downstream Apnea Type (token)



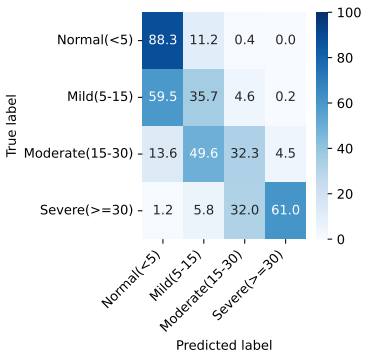
(c) Downstream Severity (subject)



(d) Baseline Breathing Disorder (token)



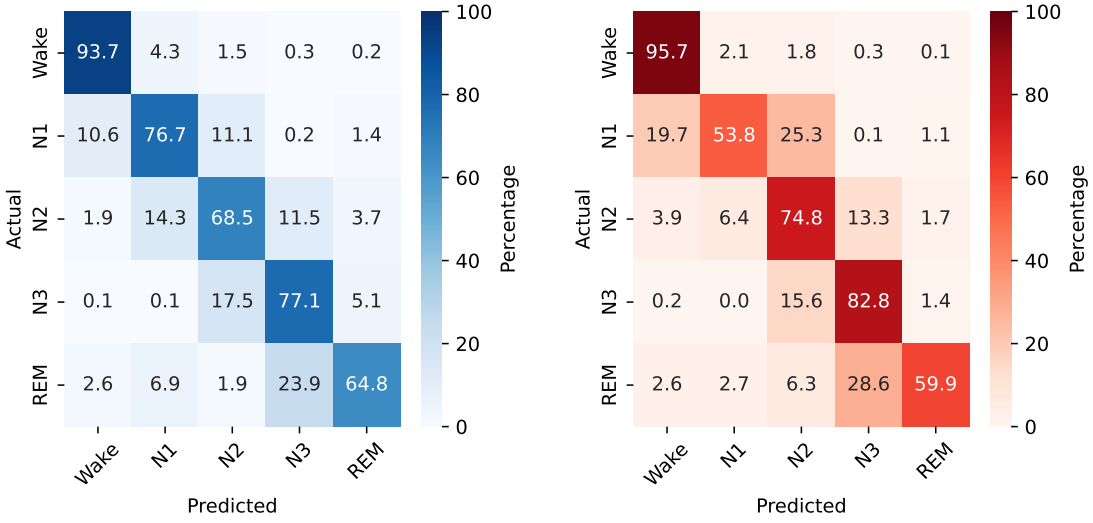
(e) Baseline Apnea Type (token)



(f) Baseline Severity (subject)

Figure A.4: RQ2: APNEA CONFUSION MATRICES. *Confusion matrices on the SHHS hold-out set for apnea-related tasks, comparing the Downstream BAS+RESP model (top row) and the BAS+RESP Transformer baseline trained from scratch (bottom row). Subfigures (a) and (d) show token-level breathing disorder classification, Subfigures (b) and (e) show token-level apnea type classification, and Subfigures (c) and (f) show subject-level severity classification (night-level AHI categories). All confusion matrices are row-normalized and show percentages.*

Figures for RQ3

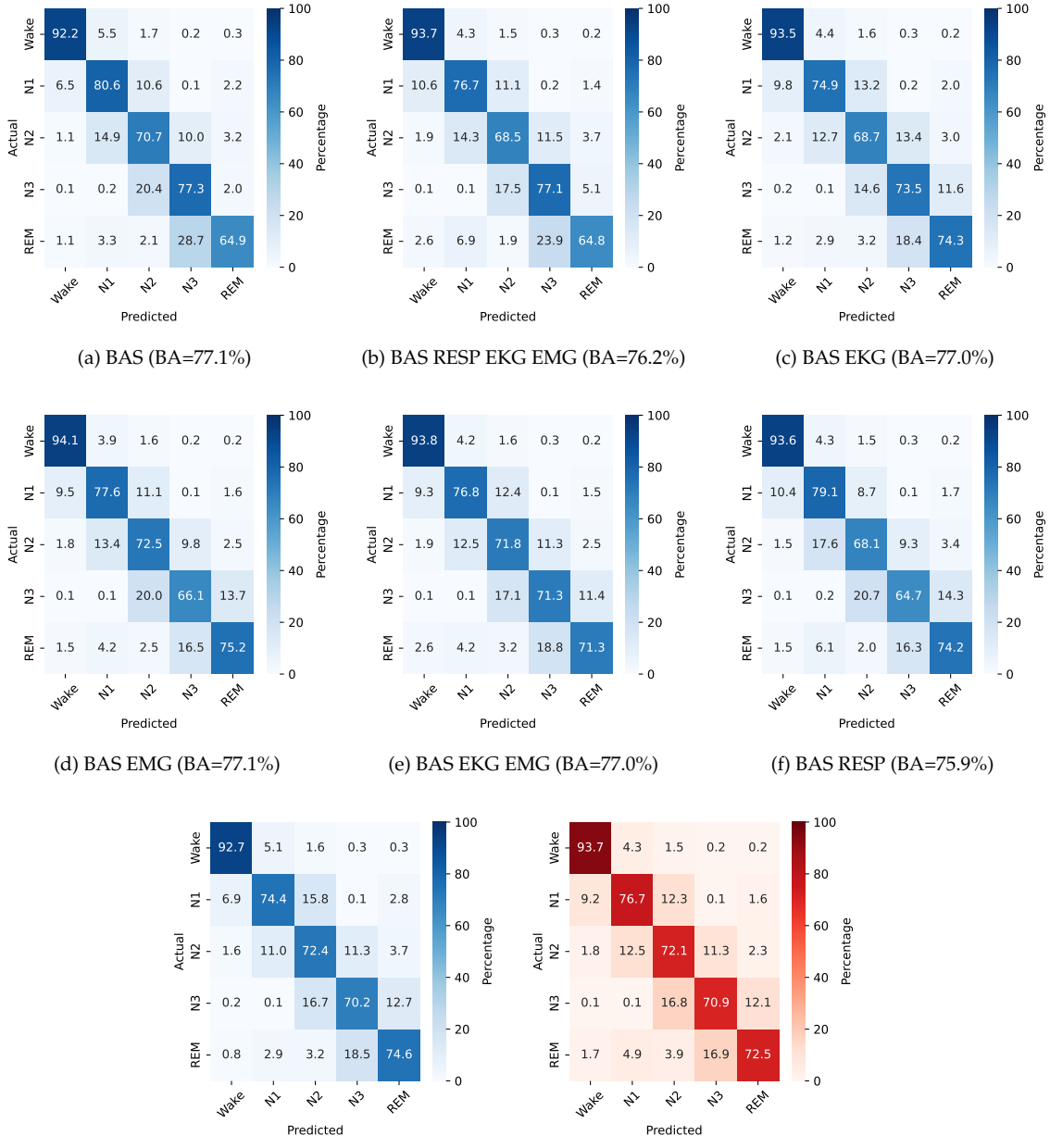


(a) BestModel with CE (BA=76.2%)

(b) BestModel with focal loss (BA=73.4%)

Figure A.5: RQ3: CONFUSION MATRICES FOR CE VS FOCAL LOSS. *Confusion matrices on the SHHS hold-out set for the cross-entropy model and the focal-loss model in sleep stage classification. Subfigure (a) shows the BestModel trained with cross-entropy, and Subfigure (b) shows the BestModel trained with focal loss. All values are row-normalized percentages.*

Figures for RQ4

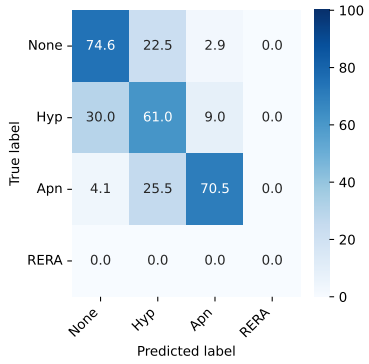


(a) BAS (BA=77.1%) (b) BAS RESP EKG EMG (BA=76.2%) (c) BAS EKG (BA=77.0%)

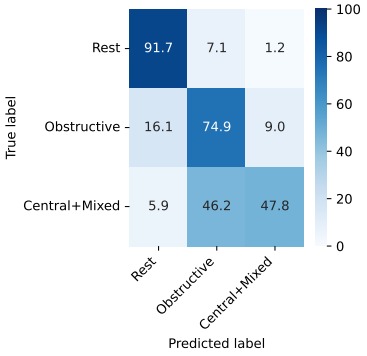
(d) BAS EMG (BA=77.1%) (e) BAS EKG EMG (BA=77.0%) (f) BAS RESP (BA=75.9%)

(g) BAS RESP EKG (BA=76.8%) (h) BAS RESP EMG (BA=77.2%)

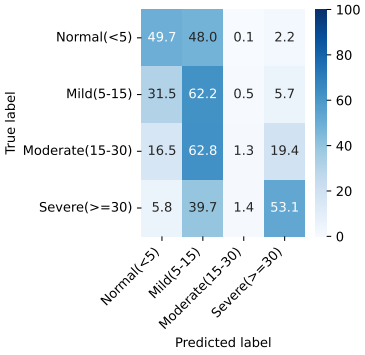
Figure A.6: RQ4: CONFUSION MATRICES FOR SLEEP STAGING BY MODALITY. *Confusion matrices on the SHHS hold-out set for selected modality configurations, illustrating how prediction errors are distributed across sleep stages. Subfigures (a)-(h) show different combinations of BAS, RESP, EKG, and EMG inputs, ordered from left to right and top to bottom, with the reported balanced accuracy for each configuration. All matrices are row-normalized and show percentages.*



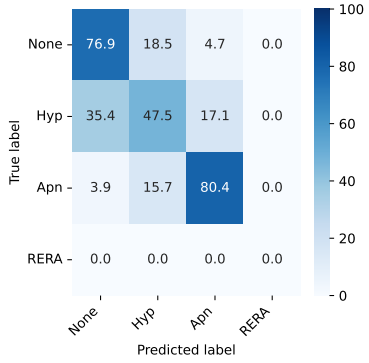
(a) BAS+RESP Breathing Disorder (token)



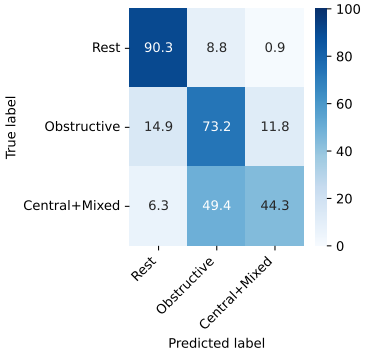
(b) BAS+RESP Apnea Type (token)



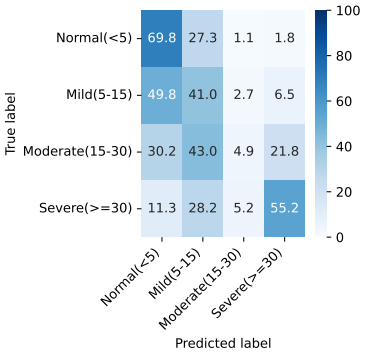
(c) BAS+RESP Severity (subject)



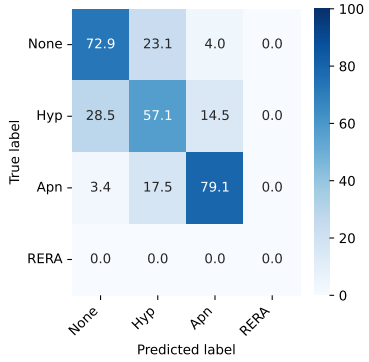
(d) BAS+RESP+EMG+EKG Breathing Disorder (token)



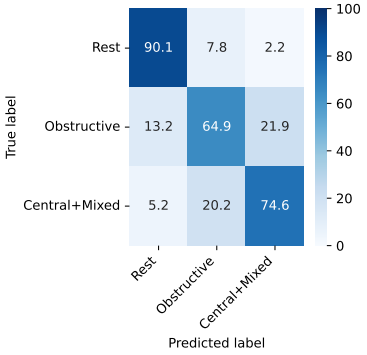
(e) BAS+RESP+EMG+EKG Apnea Type (token)



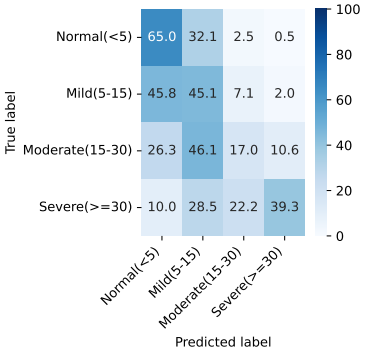
(f) BAS+RESP+EMG+EKG Severity (subject)



(g) BAS+RESP+EMG Breathing Disorder (token)

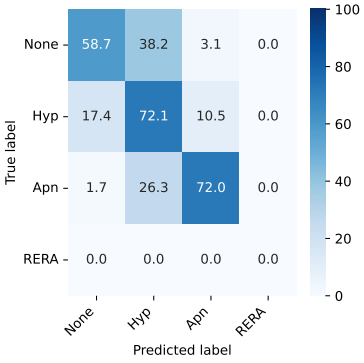


(h) BAS+RESP+EMG Apnea Type (token)

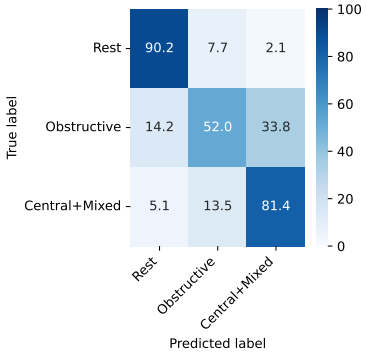


(i) BAS+RESP+EMG Severity (subject)

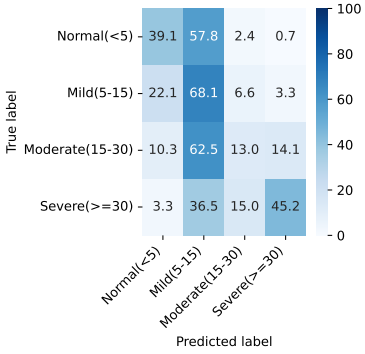
Figure A.7: RQ4: CONFUSION MATRICES FOR APNEA DETECTION BY MODALITY (PART 1 OF 3). *Confusion matrices for the 4-class sleep breathing disorder and 3-class apnea type tasks under different modality configurations, illustrating changes in error patterns when adding respiratory, EMG and EKG channels. This figure is the first of three (27 confusion matrices in total). Subfigures (a)-(c) show BAS+RESP inputs, Subfigures (d)-(f) show BAS+RESP+EMG+EKG, and Subfigures (g)-(i) show BAS+RESP+EMG. All matrices are row-normalized and show percentages.*



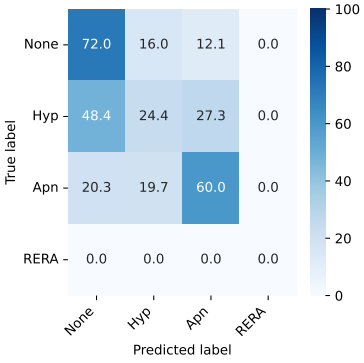
(a) BAS+RESP+EKG Breathing Disorder (token)



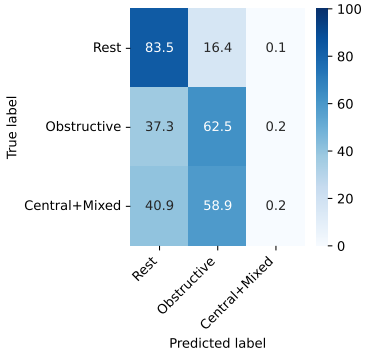
(b) BAS+RESP+EKG Apnea Type (token)



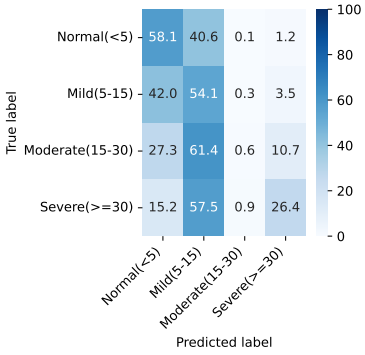
(c) BAS+RESP+EKG Severity (subject)



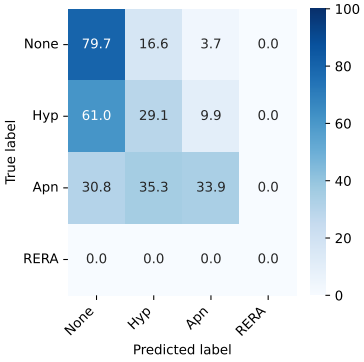
(d) BAS+EMG+EKG Breathing Disorder (token)



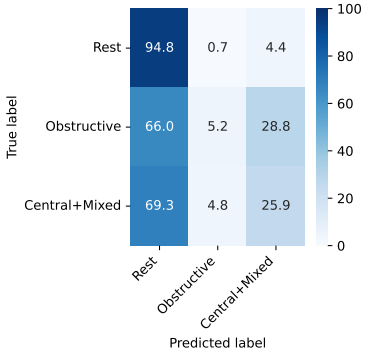
(e) BAS+EMG+EKG Apnea Type (token)



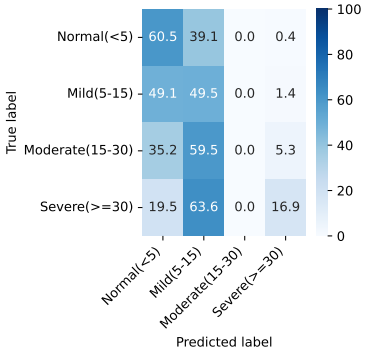
(f) BAS+EMG+EKG Severity (subject)



(g) BAS+EMG Breathing Disorder (token)

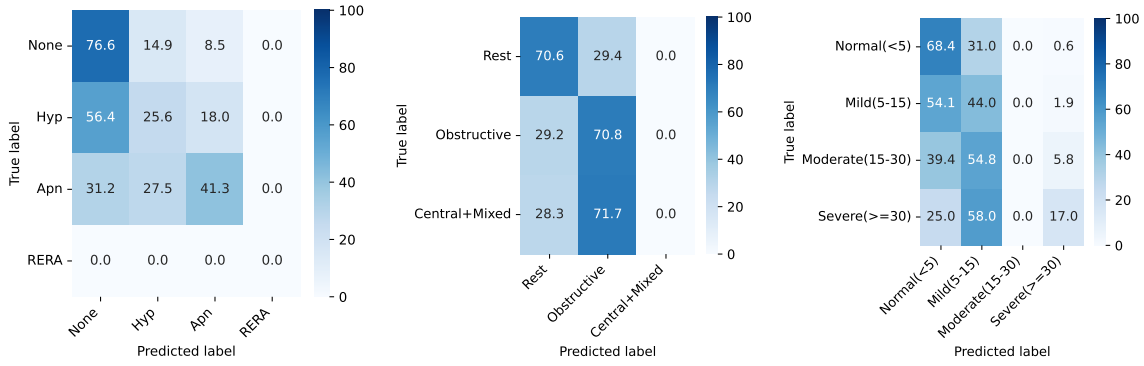


(h) BAS+EMG Apnea Type (token)

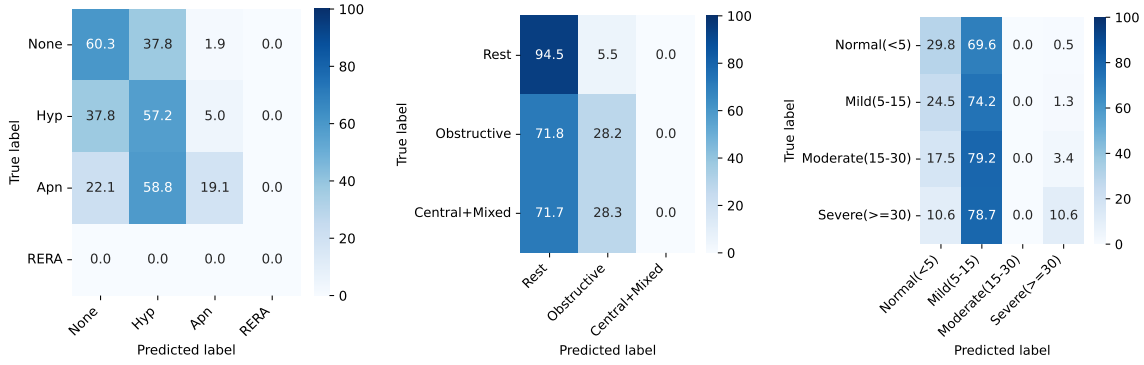


(i) BAS+EMG Severity (subject)

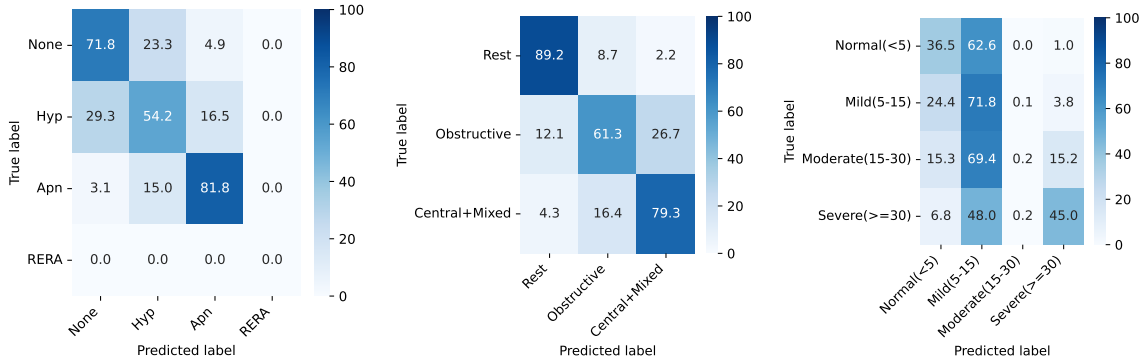
Figure A.8: RQ4: CONFUSION MATRICES FOR APNEA DETECTION BY MODALITY (PART 2 OF 3). *Confusion matrices for the 4-class sleep breathing disorder and 3-class apnea type tasks under different modality configurations, illustrating changes in error patterns when adding EMG and EKG channels. This figure is the second of three (27 confusion matrices in total). Subfigures (a)-(i) show BAS+RESP+EKG, BAS+EMG+EKG, and BAS+EMG configurations, ordered from left to right and top to bottom. All matrices are row-normalized and show percentages.*



(a) BAS+EKG Breathing Disorder (token) (b) BAS+EKG Apnea Type (token) (c) BAS+EKG Severity (subject)



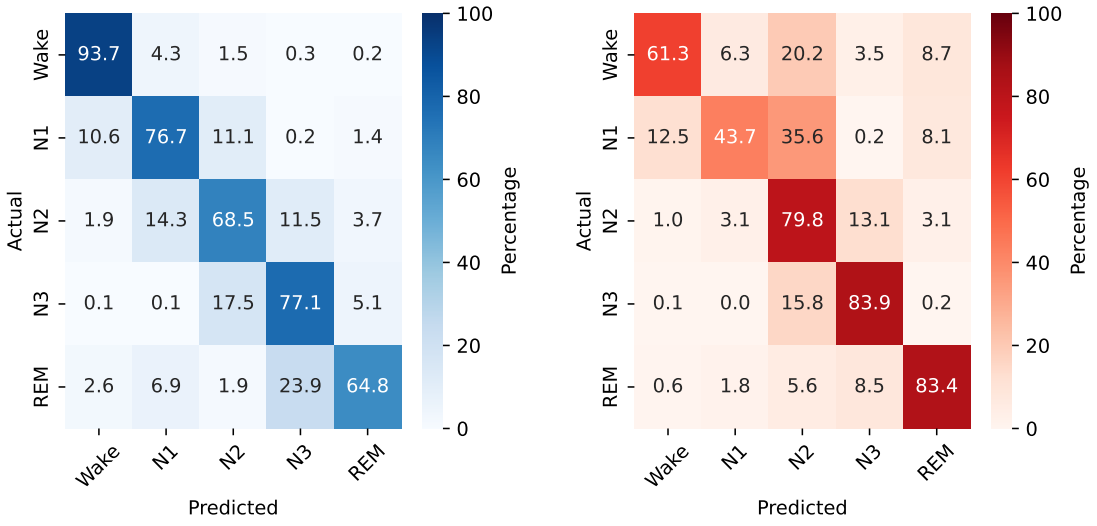
(d) BAS Breathing Disorder (token) (e) BAS Apnea Type (token) (f) BAS Severity (subject)



(g) RESP Breathing Disorder (token) (h) RESP Apnea Type (token) (i) RESP Severity (subject)

Figure A.9: RQ4: CONFUSION MATRICES FOR APNEA DETECTION BY MODALITY (PART 3 OF 3). *Confusion matrices for the 4-class sleep breathing disorder and 3-class apnea type tasks under different modality configurations. This figure is the third of three (27 confusion matrices in total). Subfigures (a)-(c) show BAS+EKG inputs, Subfigures (d)-(f) show BAS only, and Subfigures (g)-(i) show RESP only. All matrices are row-normalized and show percentages.*

Figures for RQ5



(a) LOOC

(b) SimCLR

Figure A.10: RQ5: CONFUSION MATRICES FOR DIFFERENT FOUNDATION MODEL ENCODERS ON DOWNSTREAM SLEEP STAGING. *Confusion matrices on the SHHS hold-out set for staging models using LOOC-only and LOOC+SimCLR encoders. Subfigures (a) and (b) compare the LOOC encoder and the SimCLR-pretrained encoder in terms of how prediction errors are distributed across sleep stages. All matrices are row-normalized and show percentages.*

List of Figures

List of Tables

Nomenclature

Indices

- b: batch index, $b \in \{1, \dots, B\}$
- c: channel index, $c \in \{1, \dots, C\}$
- s: temporal patch / token index within an epoch, $s \in \{0, \dots, S-1\}$
- t: labeled segment index (epoch or 10-second window) within a night, $t \in \{1, \dots, L_{\text{night}}\}$
- i, j: generic indices (token positions, confusion-matrix rows/columns)
- h: attention-head index, $h \in \{1, \dots, H\}$
- m: modality index, $m \in \mathcal{M}$
- n: training-example index in contrastive losses
- k: class index, $k \in \{1, \dots, K\}$

Sizes and Counts

- B: batch size
- C: number of channels
- T: number of time samples per segment / epoch
- P: patch length (samples per temporal patch)
- S: number of temporal patches per epoch, $S = \lfloor T/P \rfloor$
- E: token / model embedding dimension
- d_k, d_v: per-head query/key and value dimensions
- H: number of attention heads
- d_{hid}: hidden dimension in Transformer FFN
- L_{enc}: number of Transformer encoder layers
- L_{night}: number of labeled segments in one night
- K: number of classes
- N: number of labeled examples in a batch or evaluation set
- N_{train}: number of labeled training examples
- N_{tot}: number of evaluated examples for Cohen's κ
- L_{seq}: number of LSTM layers in sequence heads
- R: number of pooled outputs in PMA

Data Tensors and Embeddings

- $X \in \mathbb{R}^{B \times C \times T}$: batch of multichannel PSG segments (raw waveforms)
- $x_{b,c,s} \in \mathbb{R}^{P}$: raw waveform patch for batch element b, channel c, patch index s
- f_θ: CNN tokenizer mapping patches $x_{b,c,s}$ to token embeddings $z_{b,c,s}$
- $z_{b,c,s} \in \mathbb{R}^{E}$: token embedding for patch $x_{b,c,s}$
- $Z^{\text{ch}} \in \mathbb{R}^{B \times C \times S \times E}$: token tensor before channel aggregation
- $z_{b,s} \in \mathbb{R}^{E}$: fused token across channels at patch s for example b
- $Z \in \mathbb{R}^{B \times S \times E}$: sequence of fused tokens per epoch
- $H \in \mathbb{R}^{B \times S \times E}$: output of the temporal Transformer encoder
- $z_{\text{enc}} \in \mathbb{R}^{B \times S \times E}$: ℓ_2-normalized patch-level representations (from H)
- $z_{\text{fm}} \in \mathbb{R}^{B \times E}$: epoch-level representations (mean over patches of H)
- $z_{\text{fm},t} \in \mathbb{R}^{E}$: epoch-level representation for segment t
- $z_t \in \mathbb{R}^{E}$: generic segment-level embedding for downstream heads
- $z_{\text{pool}} \in \mathbb{R}^{1 \times E}$: pooled sequence representation from attention pooling

Attention and Transformer

- Q, K, V: query, key, and value tensors in attention
- $W_Q, W_K \in \mathbb{R}^{E \times d_k}$: projection matrices for queries and keys
- $W_V \in \mathbb{R}^{E \times d_v}$: projection matrix for values
- $W_O \in \mathbb{R}^{H d_v \times E}$: output projection matrix for multi-head attention
- $\mathrm{Attention}(\cdot)$: scaled dot-product attention
- $\mathrm{MHA}(\cdot)$: multi-head self-attention module
- $\mathrm{LN}(\cdot)$: layer-normalization operation
- $\mathrm{FFN}(\cdot)$: position-wise feed-forward network
- $\gamma, \beta \in \mathbb{R}^{E}$: learned scale and bias in layer normalization
- $\mu(x), \sigma^2(x)$: mean and variance of vector x
- ε: small constant for numerical stability in layer normalization
- $\odot$: element-wise product
- M: additive attention mask matrix (added to $QK^\top$; entries 0 or $-\infty$)
- $m_{ij} \in \{0, 1\}$: binary indicator whether query i may attend to key j
- $m_{b,c} \in \{0, 1\}$: binary mask indicating presence of channel c for example b

- $m_i \in \{0, 1\}$: binary mask for set elements in masked mean pooling
- $P_{\text{pos}} \in \mathbb{R}^{S \times E}$: sinusoidal positional encodings
- $Q_{\text{pool}} \in \mathbb{R}^{R \times E}$: learned query seeds for PMA with R outputs
- $\text{PMA}_R(\cdot)$: pooling-by-multi-head-attention operator with R outputs
- $q \in \mathbb{R}^E$: learned query for attention pooling over time
- [CLS]: learned special classification token

Modalities and Channel Sets

- $\mathcal{M} = \{\text{BAS}, \text{RESP}, \text{EKG}, \text{EMG}\}$: set of modality groups
- $m \in \mathcal{M}$: modality index (*e.g.* $m = \text{BAS}$)
- C_m: maximum number of channels for modality m (*e.g.* `BAS_CHANNELS`)

Contrastive Learning

- $z \in \mathbb{R}^E$: generic ℓ_2-normalized embedding
- $\text{sim}(u, v)$: cosine similarity between u and v (dot product for unit vectors)
- $\tau > 0$: temperature parameter in InfoNCE
- $A(i)$: index set of positives and negatives for anchor i in InfoNCE
- $j(i)$: index of the positive view for i in SimCLR
- $z_n^{(m)}$: embedding for modality m of sample n
- $\bar{z}_n^{(-m)}$: average embedding over all modalities except m for sample n
- $z_{\text{enc}}^{(a)}, z_{\text{enc}}^{(b)}$: embeddings of two augmented views of the same patch
- $\mathcal{L}_i$: InfoNCE loss for anchor i
- $\mathcal{L}_{\text{LOOC}}$: leave-one-out contrastive loss
- $\mathcal{L}_{\text{SimCLR}}$: SimCLR contrastive loss
- $\mathcal{L}_{\text{SSL}}$: combined self-supervised loss

Supervised Losses

- $a_{i,k}$: logit for sample i and class k
- $p_{i,k}$: predicted probability that sample i belongs to class k
- $y_{i,k} \in \{0, 1\}$: one-hot target indicator for sample i and class k
- α_k: class weight for class k
- $\gamma \geq 0$: focusing parameter in focal loss

- $y_i \in \mathbb{R}$: true continuous target (*e. g.* AHI) for sample i
- $\hat{y}_i \in \mathbb{R}$: predicted continuous target for sample i
- $\mathcal{L}_{\text{wCE}}$: class-weighted cross-entropy loss
- $\mathcal{L}_{\text{FL}}$: focal loss
- $\mathcal{L}_{\text{MSE}}$: mean-squared error loss

Metrics

- $\hat{y}_i$: predicted class label for sample i (argmax over $p_{i,k}$)
- $\text{TP}_k, \text{FP}_k, \text{FN}_k, \text{TN}_k$: true positives, false positives, false negatives, true negatives for class k
- P_k, R_k: precision and recall for class k
- $F1_k$: per-class F1 score for class k
- Macro-F1: averaged F1 across K classes
- Accuracy: overall fraction of correctly classified examples
- BA: balanced accuracy (mean recall across classes)
- n_{ij}: number of examples with true class i predicted as class j
- $n_{i:}, n_{:j}$: row and column sums of the confusion matrix
- p_o, p_e: observed and expected agreement in Cohen's κ
- κ: Cohen's kappa
- s_i: continuous score for binary classification (*e. g.* apnea probability)
- $\text{TPR}(t), \text{FPR}(t)$: true- and false-positive rates at threshold t
- $\text{Precision}(t), \text{Recall}(t)$: precision and recall at threshold t
- AUROC: area under the ROC curve
- AUPR/AP: area under the precision-recall curve / average precision

Bibliography

Alvarez-Estevez, D. and Rijsman, R. M. (2021). Inter-database validation of a deep learning approach for automatic sleep scoring. *PLOS ONE*, 16(8):e0256111. Publisher: Public Library of Science.

Ba, L. J., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *CoRR*, abs/1607.06450.

Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems*, volume 33, pages 12449–12460. Curran Associates, Inc.

Banville, H., Albuquerque, I., Hyvärinen, A., Moffat, G., Engemann, D.-A., and Gramfort, A. (2019). Self-supervised representation learning from electroencephalography signals. In *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. ISSN: 1551-2541.

Berry, R. B., Brooks, R., Gamaldo, C. E., Harding, S. M., Lloyd, R. M., Quan, S. F., Troester, M. M., and Vaughn, B. V. (2017). *The AASM Manual for the Scoring of Sleep and Associated Events*. American Academy of Sleep Medicine, Darien, IL.

Blackwell, T., Yaffe, K., Ancoli-Israel, S., Redline, S., Ensrud, K. E., Stefanick, M. L., Laffan, A., Stone, K. L., and Osteoporotic Fractures in Men Study Group (2011). Associations between sleep architecture and sleep-disordered breathing and cognition in older community-dwelling men: the osteoporotic fractures in men sleep study. *Journal of the American Geriatrics Society*, 59(12):2217–2225.

Brodersen, K. H., Ong, C. S., Stephan, K. E., and Buhmann, J. M. (2010). The balanced accuracy and its posterior distribution. In *2010 20th International Conference on Pattern Recognition*, pages 3121–3124, Istanbul, Turkey. IEEE.

Brunini, G. (2023). Deep learning with temporal context for sleep stage classification. Master's thesis, University of Zurich.

Cai, M. and Zeng, Y. (2024). MAE-EEG-transformer: A transformer-based approach combining masked autoencoder and cross-individual data augmentation pre-training for EEG classification. *Biomedical Signal Processing and Control*, 94:106131.

Carmel, A. (2023). Polysomnography: Overview of polysomnography, parameters monitored, staging of sleep. *eMedicine*. Publication: Medscape - eMedicine.

Chambon, S., Galtier, M. N., Arnal, P. J., Wainrib, G., and Gramfort, A. (2018). A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(4):758–769.

Chen, S., Yang, H., Yang, G., Cheng, M., Zhang, Z., Wu, Z., and Lin, J. (2025). Multiscale simulation study on the mechanical, electrical, and thermal properties of ZnSb semiconductor. *Journal of Materials Chemistry C*, 13(34):17692–17700. Publisher: The Royal Society of Chemistry.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Chen, X., Wang, R., Zee, P., Lutsey, P. L., Javaheri, S., Alcántara, C., Jackson, C. L., Williams, M. A., and Redline, S. (2015). Racial/ethnic differences in sleep disturbances: The multi-ethnic study of atherosclerosis (MESA). *Sleep*, 38(6):877–888.

Craik, A., He, Y., and Contreras-Vidal, J. L. (2019). Deep learning for electroencephalogram (EEG) classification tasks: a review. *Journal of Neural Engineering*, 16(3):031001. Publisher: IOP Publishing.

Cui, Y., Jia, M., Lin, T.-Y., Song, Y., and Belongie, S. (2019). Class-balanced loss based on effective number of samples. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9260–9269.

Dai, Y., Li, X., Liang, S., Wang, L., Duan, Q., Yang, H., Zhang, C., Chen, X., Li, L., Li, X., and Liao, X. (2023). MultiChannelSleepNet: A transformer-based model for automatic sleep stage classification with PSG. *IEEE Journal of Biomedical and Health Informatics*, 27(9):4204–4215.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dimofte, A., Bucagu, G. A., Ingolfsson, T. M., Wang, X., Cossettini, A., Benini, L., and Li, Y. (2025). CEReBrO: Compact encoder for representations of brain oscillations using efficient alternating attention. arXiv:2501.10885 [cs].

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Eldele, E., Ragab, M., Chen, Z., Wu, M., Kwoh, C. K., Li, X., and Guan, C. (2021). Time-series representation learning via temporal and contextual contrasting. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pages 2352–2359, Montreal, Canada. International Joint Conferences on Artificial Intelligence Organization.

Fox, B., Jiang, J., Wickramaratne, S., Kovatch, P., Suarez-Farinas, M., Shah, N. A., Parekh, A., and Nadkarni, G. N. (2025). A foundational transformer leveraging full night, multichannel sleep study data accurately classifies sleep stages. *Sleep*, 48(8). Publisher: Oxford Academic.

Fu, Z., Zhu, H., Zhao, Y., Huan, R., Zhang, Y., Chen, S., and Pan, Y. (2024). GMAEEG: A self-supervised graph masked autoencoder for EEG representation learning. *IEEE Journal of Biomedical and Health Informatics*, 28(11):6486–6497.

Goldberger, A. L., Amaral, L. A. N., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., and Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet. *Circulation*, 101(23):e215–e220. Publisher: American Heart Association.

Guo, Y., Nowakowski, M., and Dai, W. (2024). FlexSleepTransformer: a transformer-based sleep staging model with flexible input channel configurations. *Scientific Reports*, 14(1):26312. Publisher: Nature Publishing Group.

Hafner, M., Romanelli, R. J., Yerushalmi, E., and Troxel, W. M. (2023). The societal and economic burden of insomnia in adults: An international study. Technical report, Rand Europe.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. (2022). Masked autoencoders are scalable vision learners. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15979–15988.

Hong, S. and Baek, H. (2021). Drowsiness detection based on intelligent systems with nonlinear features for optimal placement of encephalogram electrodes on the cerebral area. *Sensors*, 21:1255.

Hu, S., Liu, J., Wang, Y., Fu, C., Zhu, J., Yu, H., and Yang, C. (2025). Transparent artificial intelligence-enabled interpretable and interactive sleep apnea assessment across flexible monitoring scenarios. *Nature Communications*, 16(1):7548. Publisher: Nature Publishing Group.

Irie, K. (2025). Why are positional encodings nonessential for deep autoregressive transformers? A petroglyph revisited. In Che, W., Nabende, J., Shutova, E., and Pilehvar, M. T., editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 551–559, Vienna, Austria. Association for Computational Linguistics.

Jaiswal, A., Babu, A. R., Zadeh, M. Z., Banerjee, D., and Makedon, F. (2021). A survey on contrastive self-supervised learning. *Technologies*, 9(1):2. Publisher: Multidisciplinary Digital Publishing Institute.

Kemp, B., Zwinderman, A., Tuk, B., Kamphuisen, H., and Oberye, J. (2000). Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the EEG. *IEEE Transactions on Biomedical Engineering*, 47(9):1185–1194.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. (2020). Supervised contrastive learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in neural information processing systems*, volume 33, pages 18661–18673. Curran Associates, Inc.

Kiyasseh, D., Zhu, T., and Clifton, D. A. (2021). CLOCS: Contrastive learning of cardiac signals across space, time, and patients. In *Proceedings of the 38th International Conference on Machine Learning*, pages 5606–5615. PMLR. ISSN: 2640-3498.

Lawhern, V. J., Solon, A. J., Waytowich, N. R., Gordon, S. M., Hung, C. P., and Lance, B. J. (2018). EEGNet: A compact convolutional network for EEG-based brain-computer interfaces. *Journal of Neural Engineering*, 15(5):056013.

Lee, J., Lee, Y., Kim, J., Kosiorek, A., Choi, S., and Teh, Y. W. (2019). Set Transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, pages 3744–3753. PMLR.

Lee, Y. J., Lee, J. Y., Cho, J. H., and Choi, J. H. (2022). Interrater reliability of sleep stage scoring: a meta-analysis. *Journal of Clinical Sleep Medicine*, 18(1):193–202.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2020). Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327.

Magalang, U. J., Chen, N.-H., Cistulli, P. A., Fedson, A. C., Gíslason, T., Hillman, D., Penzel, T., Tamisier, R., Tufik, S., Phillips, G., and Pack, A. I. (2013). Agreement in the scoring of respiratory events and sleep among international sleep centers. *Sleep*, 36(4):591–596.

Mahowald, M. W. and Schenck, C. H. (2005). Insights from studying human sleep disorders. *Nature*, 437(7063):1279–1285. Publisher: Nature Publishing Group.

Mander, B. A., Winer, J. R., and Walker, M. P. (2017). Sleep and human aging. *Neuron*, 94(1):19–36.

McHugh, M. L. (2012). Interrater reliability: the kappa statistic. *Biochemia Medica*, 22(3):276–282.

Montavon, G., Samek, W., and Müller, K.-R. (2018). Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15.

Moret-Bonillo, V., Alvarez-Estévez, D., Fernández-Leal, A., and Hernández-Pereira, E. (2014). Intelligent approach for analysis of respiratory signals and oxygen saturation in the sleep apnea/hypopnea syndrome. *The Open Medical Informatics Journal*, 8:1–19.

Ogg, M. and Coon, W. G. (2024). Self-supervised transformer model training for a Sleep-EEG foundation model. In *2024 46th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1–6.

Oord, A. V. D., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748.

O'Reilly, C., Gosselin, N., Carrier, J., and Nielsen, T. (2014). Montreal archive of sleep studies: an open-access resource for instrument benchmarking and exploratory research. *Journal of Sleep Research*, 23(6):628–635.

Peng, L., Ren, Y., Luan, Z., Chen, X., Yang, X., and Tu, W. (2023). SleepViTransformer: Patch-based sleep spectrogram transformer for automatic sleep staging. *Biomedical Signal Processing and Control*, 86:105203.

Perslev, M., Darkner, S., Kempfner, L., Nikolic, M., Jennum, P. J., and Igel, C. (2021). U-Sleep: resilient high-frequency sleep staging. *npj Digital Medicine*, 4(1):72. Publisher: Nature Publishing Group.

Perslev, M., Jensen, M., Darkner, S., rgen Jennum, P. J., and Igel, C. (2019). U-Time: A fully convolutional network for time series segmentation applied to sleep staging. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Phan, H., Andreotti, F., Cooray, N., Chén, O. Y., and De Vos, M. (2019). SeqSleepNet: End-to-end hierarchical recurrent neural network for sequence-to-sequence automatic sleep staging. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(3):400–410.

Phan, H., Mikkelsen, K., Chén, O. Y., Koch, P., Mertins, A., and De Vos, M. (2022). SleepTransformer: Automatic sleep staging with interpretability and uncertainty quantification. *IEEE Transactions on Biomedical Engineering*, 69(8):2456–2467.

Powers, D. (2008). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *Mach. Learn. Technol.*, 2.

Quan, S. F., Howard, B. V., Iber, C., Kiley, J. P., Nieto, F. J., O'Connor, G. T., Rapoport, D. M., Redline, S., Robbins, J., Samet, J. M., and Wahl, P. W. (1997). The sleep heart health study: design, rationale, and methods. *Sleep*, 20(12):1077–1085.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021). Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR. ISSN: 2640-3498.

Redline, S., Tishler, P. V., Tosteson, T. D., Williamson, J., Kump, K., Browner, I., Ferrette, V., and Krejci, P. (1995). The familial aggregation of obstructive sleep apnea. *American Journal of Respiratory and Critical Care Medicine*, 151(3 Pt 1):682–687.

Shaw, P., Uszkoreit, J., and Vaswani, A. (2018). Self-attention with relative position representations. In Walker, M., Ji, H., and Stent, A., editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

Shi, H., Gao, J., Ren, X., Xu, H., Liang, X., Li, Z., and Kwok, J. T.-Y. (2021). SparseBERT: Rethinking the importance analysis in self-attention. In *Proceedings of the 38th International Conference on Machine Learning*, pages 9547–9557. PMLR.

Shrivastava, D., Jung, S., Saadat, M., Sirohi, R., and Crewson, K. (2014). How to interpret the results of a sleep study. *Journal of Community Hospital Internal Medicine Perspectives*, 4(5):10.3402/jchimp.v4.24983.

Silber, M. H., Ancoli, I. S., Bonnet, M. H., Chokroverty, S., Grigg, D. M. M., Hirshkowitz, M., Kapen, S., Keenan, S. A., Kryger, M. H., Penzel, T., Pressman, M. R., and Iber, C. (2007). The visual scoring of sleep in adults. *Journal of Clinical Sleep Medicine*, 03(02):121–131. Publisher: American Academy of Sleep Medicine.

Sokolova, M. and Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437.

Stephansen, J. B., Olesen, A. N., Olsen, M., Ambati, A., Leary, E. B., Moore, H. E., Carrillo, O., Lin, L., Han, F., Yan, H., Sun, Y. L., Dauvilliers, Y., Scholz, S., Barateau, L., Hogl, B., Stefani, A., Hong, S. C., Kim, T. W., Pizza, F., Plazzi, G., Vandi, S., Antelmi, E., Perrin, D., Kuna, S. T., Schweitzer, P. K., Kushida, C., Peppard, P. E., Sorensen, H. B. D., Jennum, P., and Mignot, E. (2018). Neural network analysis of sleep stages enables efficient diagnosis of narcolepsy. *Nature Communications*, 9(1):5229.

Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., and Liu, Y. (2024). RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Supratak, A., Dong, H., Wu, C., and Guo, Y. (2017). DeepSleepNet: a model for automatic sleep stage scoring based on raw single-channel EEG. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(11):1998–2008.

Terzano, M. G., Parrino, L., Sherieri, A., Chervin, R., Chokroverty, S., Guilleminault, C., Hirshkowitz, M., Mahowald, M., Moldofsky, H., Rosa, A., Thomas, R., and Walters, A. (2001). Atlas, rules, and recording techniques for the scoring of cyclic alternating pattern (CAP) in human sleep. *Sleep Medicine*, 2(6):537–553.

Thapa, R., He, B., Kjær, M. R., Moore, H., Ganjoo, G., Mignot, E., and Zou, J. (2024). Sleepfm: multi-modal representation learning for sleep across brain activity, ecg and respiratory signals. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Thapa, R., Kjær, M. R., He, B., Covert, I., Moore, H., Hanif, U., Ganjoo, G., Westover, M. B., Jennum, P., Brink-Kjær, A., Mignot, E., and Zou, J. (2025). A multimodal sleep foundation model developed with 500K hours of sleep recordings for disease predictions.

Vallat, R. and Walker, M. P. (2021). An open-source, high-performance tool for automated sleep staging. *eLife*, 10:e70092. Publisher: eLife Sciences Publications, Ltd.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Xiong, R., Yang, Y., He, D., Zheng, K., Zheng, S., Xing, C., Zhang, H., Lan, Y., Wang, L., and Liu, T. (2020). On layer normalization in the transformer architecture. In *Proceedings of the 37th International Conference on Machine Learning*, pages 10524–10533. PMLR. ISSN: 2640-3498.

Young, T., Palta, M., Dempsey, J., Peppard, P. E., Nieto, F. J., and Hla, K. M. (2009). Burden of sleep apnea: rationale, design, and major findings of the wisconsin sleep cohort study. *WMJ: official publication of the State Medical Society of Wisconsin*, 108(5):246–249.

Yue, Z., Wang, Y., Duan, J., Yang, T., Huang, C., Tong, Y., and Xu, B. (2022). TS2Vec: Towards universal representation of time series. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(8):8980–8987.

Zaheer, M., Kottur, S., Ravanbhakhsh, S., Póczos, B., Salakhutdinov, R., and Smola, A. J. (2017). Deep sets. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 3394–3404, Red Hook, NY, USA. Curran Associates Inc.

Zhang, G.-Q., Cui, L., Mueller, R., Tao, S., Kim, M., Rueschman, M., Mariani, S., Mobley, D., and Redline, S. (2018). The national sleep research resource: towards a sleep data commons. *Journal of the American Medical Informatics Association: JAMIA*, 25(10):1351–1358.

Zhang, X., Zhang, X., Huang, Q., Lv, Y., and Chen, F. (2024). A review of automated sleep stage based on EEG signals. *Biocybernetics and Biomedical Engineering*, 44(3):651–673.

Zhang, Y., Zhou, L., Zhu, S., Zhou, Y., Wang, Z., Ma, L., Yuan, Y., Xie, Y., Niu, X., Su, Y., Liu, H., Hei, X., Shi, Z., Ren, X., and Shi, Y. (2025). Deep learning for obstructive sleep apnea detection and severity assessment: A multimodal signals fusion multiscale transformer model. *Nature and Science of Sleep*, 17:1–15.

Zhao, T., Cui, Y., Ji, T., Luo, J., Li, W., Jiang, J., Gao, Z., Hu, W., Yan, Y., Jiang, Y., and Hong, B. (2024). VAEEG: Variational auto-encoder for extracting EEG representation. *NeuroImage*, 304:120946.

Zong, Y., Wei, Q., Zhong, M., Liu, K., and Liu, Q. (2025). Automated sleep staging based on multi-physiological signals using RF and XGBoost. In *Proceedings of the 4th International Conference on Biomedical and Intelligent Systems*, IC-BIS '25, pages 125–130, New York, NY, USA. Association for Computing Machinery.