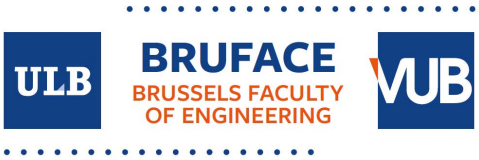

Sleep Spindle Detection from Electroencephalography via Sleep Foundation Models

Alexis Bollengier

Academic year 2025–2026

Master's thesis submitted in order to be awarded the Master's Degree in
Electrical Engineering, Focus Electronics and Information Technologies

Supervisors
Dr. André Anjos
Prof. Hichem Sahli

Co-Supervisors
Dr. Abel Díaz Berenguer
Dr. Olivier Pallanca

ECOLE
POLYTECHNIQUE
DE BRUXELLES

DOCUMENT TO BE INCLUDED IN THE MASTER THESIS

CONSULTATION OF THE MASTER THESIS

I, the undersigned

NAME ..... Bollengier .....

FIRST NAME ..... Alexis .....

MASTER PROGRAM ..... Electrical Engineering, Focus Electronics and Information Technologies .....

MASTER THESIS TITLE ..... Sleep Spindle Detection from Electroencephalography via Sleep Foundation Models .....

.....

.....

.....

.....

AUTHORIZE*

~~REFUSE~~*

(IN CASE OF REFUSAL, THIS DOCUMENT MUST BE COUNTERSIGNED BY THE MASTER THESIS SUPERVISOR)

(Delete as appropriate)*

Consultation of this master thesis by users of the libraries of the Université libre de Bruxelles.

If consultation is authorised, the undersigned hereby grants the Université libre de Bruxelles a free and non-exclusive licence, for the duration of the legal term of protection of the work, to reproduce and communicate to the public the above-mentioned work, on graphic or electronic media, in order to enable users of the libraries of the ULB and other institutions to consult it within the framework of inter-library loan.

Brussels, ... 13/08/2026 .....

Signature of the student

Signature of the supervisor
(only if consultation is refused)

Abstract

Sleep Spindle Detection from Electroencephalography via Sleep Foundation Models by Alexis Bollengier, Master in Electrical Engineering, Focus Electronics and Information Technologies, Université libre de Bruxelles, academic year 2025-2026.

Sleep spindle analysis offers quantitative insights into sleep physiology. However, large-scale studies require reliable automatic detection and precise event localization. Sleep foundation models generate pretrained representations from extensive polysomnographic datasets, yet their token-level temporal representations may be too coarse for the dense localization of brief electrophysiological events.

This thesis examines the adaptation of a pretrained sleep foundation model for dense sleep spindle detection from EEG data. The proposed architecture employs a frozen SleepFM encoder to supply contextual temporal information and incorporates a task-specific frequency branch to preserve localized spectral structure. These representations are integrated using residual cross-attention, after which a dense decoder maps the fused representation to sample-level predictions.

Experimental results indicate that the pretrained temporal representation alone lacks sufficient localization information, while the frequency representation provides the fine-grained evidence necessary for accurate event reconstruction. The final model achieves an event-level F1 score of 0.7950 at IoU 0.2 on the held-out OPA test set and achieves the highest F1 score among the evaluated methods under the common OPA protocol, outperforming BLAST and SUMO. However, cross-dataset experiments demonstrate that superior in-domain performance does not necessarily result in stronger generalization, and transfer is strongly influenced by the annotation source. These findings suggest that adapting sleep foundation models to dense event detection requires using pretrained temporal representations as contextual information while preserving local temporal and spectral information below the token scale.

Key words: Sleep spindle detection, Electroencephalography, Foundation models, Deep learning, Time-frequency modeling, Temporal localization.

Acknowledgements

I would like to sincerely thank Prof. Hichem Sahli, Dr. Abel Díaz Berenguer, Dr. André Anjos and Dr. Olivier Pallanca for their guidance, trust, and involvement throughout this work.

I am particularly grateful to Prof. Hichem Sahli and Dr. Abel Díaz Berenguer for the time and energy they dedicated to this thesis during the second semester. They took the time to understand the architecture in depth, challenge its limitations, and continuously search for ways to improve both the model and the quality of the work. Their expertise in deep learning had a direct impact on the final results. Beyond the technical aspects, they constantly encouraged me to be scientifically ambitious and pushed me beyond my comfort zone, with the objective of producing work that could eventually reach publication quality. Their confidence in me and the working environment they provided at the VUB were also greatly appreciated.

I would equally like to thank Dr. André Anjos and Dr. Olivier Pallanca, who have accompanied this project since it began in September as a semester project during my exchange at EPFL. I particularly appreciated André's clear explanations, the precision of his expectations and the trust he placed in my work. His expertise in deep learning and his guidance have been essential throughout the development of the project. Olivier provided a complementary medical perspective that continuously brought the work back to the physiology and characteristics of sleep spindles. His input helped ensure that the model was not designed solely around numerical performance but also around the properties of the events it was intended to detect.

The numerous discussions with all four supervisors helped shape the research questions, refine the architecture and progressively raise the scientific quality of this work. I feel deeply thankful for having had the chance to work in such an environment.

Finally, I would like to thank the Idiap Research Institute for the computational resources and working environment that made this project possible. More importantly, this experience allowed me to discover scientific research from within and strongly reinforced my desire to continue along this path in the future.

Contents

List of Figures

List of Tables

List of Acronyms

Acronym	Definition
AUC	Area Under the Curve
BiLSTM	Bidirectional Long Short-Term Memory
CCA	Canonical Correlation Analysis
CFS	Cleveland Family Study
CNN	Convolutional Neural Network
CUDA	Compute Unified Device Architecture
CWT	Continuous Wavelet Transform
DTW	Dynamic Time Warping
ECG	Electrocardiography
EEG	Electroencephalography
EMG	Electromyography
EOG	Electrooculography
FFN	Feed-Forward Network
FFT	Fast Fourier Transform
FN	False Negative
FP	False Positive
GELU	Gaussian Error Linear Unit
GPU	Graphics Processing Unit
HDF5	Hierarchical Data Format Version 5
InfoNCE	Information Noise-Contrastive Estimation
IoU	Intersection over Union
KL	Kullback–Leibler
LN	Layer Normalization
MAE	Mean Absolute Error
MASS	Montreal Archive of Sleep Studies
MESA	Multi-Ethnic Study of Atherosclerosis
MLP	Multilayer Perceptron
MNC	Mignot Nature Communications
MODA	Massive Online Data Annotation
MrOS	Osteoporotic Fractures in Men Study
NREM	Non-Rapid Eye Movement

Acronym	**Definition**
OPA	Olivier Pallanca Annotation Set
OSF	Open Sleep Foundation Model
PSG	Polysomnography
REM	Rapid Eye Movement
ROC	Receiver Operating Characteristic
RUSBoost	Random Under-Sampling Boosting
SE	Squeeze-and-Excitation
SGD	Stochastic Gradient Descent
SHHS	Sleep Heart Health Study
STAGES	Stanford Technology Analytics and Genomics in Sleep
STFT	Short-Time Fourier Transform
TCN	Temporal Convolutional Network
TP	True Positive
WSC	Wisconsin Sleep Cohort

Nomenclature

Symbols are grouped by the stage of the pipeline in which they are introduced. Numerical values are reported at their first relevant specification and summarized where appropriate in Section 5.1.

Signal and supervision

Symbol	Definition
X	EEG recording, $X \in \mathbb{R}^{C \times T}$
X_k	EEG segment of index k, $X_k \in \mathbb{R}^{C \times T_0}$
x_c	EEG signal associated with channel c
C	Number of EEG channels
T	Number of samples in a recording
T_0	Number of samples in a segment
f_s	Sampling frequency
B	Batch size
$t_{\text{on}}, t_{\text{off}}$	Onset and offset of the spindle event enclosing a positive sample
y_t	Ground-truth spindle mask at sample t
$\mathbf{b}_t$	Boundary target at sample t, $(\Delta^t_{\text{on}}, \Delta^t_{\text{off}})$
$\Delta^t_{\text{on}}, \Delta^t_{\text{off}}$	Offsets from sample t to the onset and offset of the enclosing event
c_t	Temporal class target: background, boundary, or interior
Ω^+	Set of samples belonging to an annotated spindle
$\hat{y}_t$	Predicted spindle probability at sample t
$\hat{\mathbf{b}}_t$	Predicted boundary offsets at sample t
$\hat{\Delta}^t_{\text{on}}, \hat{\Delta}^t_{\text{off}}$	Predicted offsets to the onset and offset of the enclosing event
$\hat{\mathbf{c}}_t$	Predicted temporal-class probabilities at sample t
θ	Trainable parameters of the detector

Tokenization and temporal branch

Symbol	Definition
$\mathcal{M}$	Set of regional EEG groups used during leave-one-out pretraining
m	Index of an EEG regional group
$z_m^{(b)}$	Normalized representation of regional group m for batch element b
$z_{-m}^{(b)}$	Aggregate representation of the regional groups complementary to m for batch element b
L_{LOO}	Leave-one-out pretraining objective
ℓ_{NCE}	Symmetric InfoNCE loss used during leave-one-out pretraining
$S_{bb'}^m$	Contrastive logit between regional representations of batch elements b and b'
τ_{pre}	Learnable logit-scale parameter used during leave-one-out pretraining and $e^{\tau_{\mathrm{pre}}}$ scales the contrastive logits
W	Token window length, in samples
S_{tok}	Token stride, in samples
N	Number of tokens extracted from a segment
a_n	First sample of token n
E	Embedding dimension of the pretrained encoder
Z_{tmp}	Temporal token representation, $\mathbb{R}^{B\times N\times E}$

Frequency branch

Symbol	Definition
n_c	Number of oscillation cycles of the Morlet wavelet
$f_{\min}, f_{\max}$	Lower and upper frequencies of the analyzed frequency grid
f_j	Analyzed frequency of index j
F	Number of analyzed frequencies
Δf	Spacing of the frequency grid
σ_f	Temporal spread of the Morlet wavelet at frequency f
σ_f^{Hz}	Spectral spread of the Morlet wavelet at frequency f
κ	Truncation of the Gaussian envelope, in standard deviations
w	Wavelet kernel length, in samples
p	Symmetric zero padding applied to the wavelet convolution
ψ_j	Complex Morlet kernel associated with frequency f_j
$\widetilde{\psi}_j$	Centered and unit-energy normalized Morlet kernel
H_{sc}	Frame stride of the scalogram, in samples
K	Number of scalogram frames in a segment

Symbol	Definition
S	Scalogram, $S \in \mathbb{R}^{C\times K\times F}$
ε	Small positive constant ensuring numerical stability
L	Number of scalogram frames in one token-aligned patch
P	Stack of token-aligned spectral patches, $P \in \mathbb{R}^{B\times N\times C\times L\times F}$
$P_{b,n}$	Spectral patch associated with batch element b and token n
$r_{\text{freq},j}$	Frequency-wise summary at frequency index j
r_{freq}	Frequency summary vector, $r_{\text{freq}} \in \mathbb{R}^F$
W_1, W_2, b_1, b_2	Parameters of the frequency-wise recalibration projection
a	Frequency gate produced by the recalibration module, $a \in (0,1)^F$
$\widetilde{P}$	Recalibrated spectral patch
$U^{(\ell)}$	Feature map produced by convolutional layer ℓ of the frequency encoder
U	Output of the convolutional stack, $U \in \mathbb{R}^{M\times L\times F}$
M	Channel width of the last convolutional layer
F_b	Frequency indices belonging to frequency band b
N_b	Number of frequency bands used by the structured readout
T_s	Number of temporal segments used by the structured readout
I_s	Scalogram-frame indices belonging to temporal segment s
$\mu[m,b,s]$	Mean activation of convolutional channel m, frequency band b, and temporal segment s
v	Token descriptor produced by the structured readout
D	Width of the token descriptor, $D = N_b M T_s$
W_p, b_p	Weight matrix and bias used to project the structured descriptor to the frequency embedding
z_{freq}	Frequency embedding associated with one token, $z_{\text{freq}} \in \mathbb{R}^E$
Z_{freq}	Frequency token representation, $\mathbb{R}^{B\times N\times E}$

Fusion and auxiliary objectives

Symbol	Definition
Q_f	Queries obtained from the frequency stream
K_t, V_t	Keys and values obtained from the temporal stream
W_Q, W_K, W_V	Linear projections used to construct queries, keys, and values
W_O	Output projection of the multi-head attention module
n_h	Number of attention heads
head_i	Output of attention head i
A	Concatenated multi-head attention output after projection
Z_{attn}	Representation after cross-attention and its residual connection

Symbol	Definition
Z_{fused}	Fused token representation, $\mathbb{R}^{B\times N\times E}$
h_{align}	Shared projection head used by the auxiliary alignment objective
$q_{\text{tmp}}, q_{\text{freq}}$	Diagonal Gaussian distributions associated with the temporal and frequency streams
μ_q, σ_q^2	Mean and variance parameters of stream q in the alignment space
d_{align}	Dimensionality of the latent alignment space
Ω_{tok}	Set of batch-token pairs used by the consistency objective
D_{KL}	Kullback–Leibler divergence between the temporal and frequency latent distributions
L_{cons}	Token-level representation consistency loss
Ω_{evt}	Set of predicted event candidates used by the frequency constraint
ρ_e	Sigma prominence of event candidate e
ρ_{th}	Target sigma prominence
$E_{\sigma,e}$	Energy of event e in the sigma frequency band
$E_{\text{ctx},e}$	Energy of event e in the surrounding frequency bands
L_{freq}	Spectral prominence penalty
L_{align}	Complete auxiliary objective, $L_{\text{align}} = L_{\text{cons}} + L_{\text{freq}}$

Dense temporal reconstruction

Symbol	Definition
d	Width of the sample-level decoder representation
K_a	Number of Morlet atoms in the synthesis bank
P_a	Number of candidate positions of each atom within a token window
Ψ	Fixed atom bank, $\Psi \in \mathbb{R}^{K_a\times P_a\times W}$
z_n	Fused representation of token n
C_n	Atom coefficients predicted from token n
L_n	Atom-position logits predicted from token n
f_C, f_L	Multilayer perceptrons predicting atom coefficients and position logits
$\pi_{n,k,m}$	Placement probability of atom k at candidate position m for token n
$A_{n,k,w}$	Localized form of atom k at position w within token n
W_b, b_b	Weight matrix and bias of the token-level base projection
$X_{n,w}$	Reconstructed feature vector at position w of token n
g	Hann window used by the overlap-add reconstruction

Symbol	Definition
I_t	Set of token-window positions contributing to sample t
Z_{sample}	Sample-level sequence produced by the wavelet synthesis
Φ	Intra-token frequency frames retained by the structured readout, $\mathbb{R}^{B\times N\times T_s\times Q}$
Q	Width of one intra-token frame, $Q = N_b M$
W_f, b_f	Weight matrix and bias used to project the intra-token frames to the decoder width
$\overline{\Phi}$	Intra-token frames projected to the decoder width, $\mathbb{R}^{B\times N\times T_s\times d}$
$\mathrm{Up}(\cdot)$	Nearest-neighbor expansion of intra-token frames to the token sample grid
Z_{frame}	Frame-aware representation aligned on the sample axis
W_s, b_s	Weight matrix and bias used to combine the reconstructed and frame-aware sample-level representations
Z_{comb}	Combination of the reconstructed and frame-aware representations
N_r	Number of residual blocks in the temporal convolutional network
δ_ℓ	Dilation of residual block ℓ
$h^{(\ell)}$	Sample-level representation after residual block ℓ
H	Sample-level representation returned by the decoder, $\mathbb{R}^{B\times T_0\times d}$
H_t	Decoder representation associated with sample t

Prediction heads and training objective

Symbol	Definition
ϕ	Gaussian Error Linear Unit activation
$\sigma(\cdot)$	Sigmoid activation function
W_g, b_g	Weight matrix and bias of the shared prediction-head projection
G_t	Shared features used by the prediction heads at sample t
f_{mask}	Prediction head producing the spindle probability
f_{bnd}	Prediction head producing the boundary offsets
f_{cls}	Prediction head producing the temporal-class probabilities
Ω	Set of sample indices used by the dense training losses
L_{seg}	Segmentation loss of the spindle-mask head
L_{focal}	Focal component of the segmentation loss
L_{dice}	Dice component of the segmentation loss
p_t	Probability assigned to the ground-truth binary class at sample t in the focal loss

Symbol	Definition
α_t	Class-balancing factor of the focal loss at sample t
γ	Focusing parameter of the focal loss
L_{bnd}	Boundary regression loss
ℓ_β	Smooth ℓ_1 penalty used for boundary regression
β	Transition point between the quadratic and linear regions of the smooth ℓ_1 loss
L_{cls}	Temporal-class cross-entropy loss
w_c	Weight associated with temporal class c
w_{c_t}	Class weight selected according to the target class c_t
$\lambda_{\text{seg}}, \lambda_{\text{bnd}}$	Weights of the segmentation and boundary-regression objectives
$\lambda_{\text{cls}}, \lambda_{\text{align}}$	Weights of the temporal-class and auxiliary objectives
$\mathcal{L}$	Complete detector training objective

Inference and evaluation

Symbol	Definition
η	Probability threshold used to convert spindle probabilities into binary sample-level predictions
g_{merge}	Maximum temporal gap below which neighboring candidate intervals are merged
s_t, e_t	Onset and offset proposals generated by sample t during boundary refinement
w_t	Weight assigned to the boundary proposals of sample t, with $w_t = \hat{y}_t$
$\hat{t}_{\text{on}}, \hat{t}_{\text{off}}$	Refined onset and offset of a predicted spindle event
median_w	Weighted-median operator used to aggregate boundary proposals
τ	Intersection-over-Union threshold used for event matching
$\mathcal{G}$	Set of reference spindle events
$\mathcal{P}$	Set of predicted spindle events
N_G	Number of reference spindle events
N_P	Number of predicted spindle events
N_M	Number of one-to-one matched prediction–reference event pairs

Chapter 1

Introduction

1.1 Motivation and Context

Sleep spindles offer quantitative insights into brain activity during sleep and are associated with cognitive processes, including memory consolidation [1]. These oscillatory events facilitate memory stabilization by coordinating with slow oscillations and hippocampal ripples [2]. Because they are generated by the thalamocortical circuit, spindle characteristics reflect the integrity of that circuit. Changes in spindle characteristics have been reported in neuropsychiatric conditions such as schizophrenia [1] and major depressive disorder [3]. They have also been reported in sleep disorders such as obstructive sleep apnea [4]. Consequently, accurate sleep spindle detection enables the reliable quantification of these electrophysiological biomarkers and may support the characterization of neurological and sleep-related disorders.

Accurate extraction of these measures depends on the precise identification and localization of individual spindle events in electroencephalographic recordings. Manual expert annotation is time-intensive. It does not scale to large sleep studies. It is also subject to variability in event identification and boundary placement. Reliable automatic detection is therefore essential for overcoming annotation limitations and enabling reproducible analysis of spindle characteristics across large populations.

Deep learning methods have advanced automatic spindle detection by learning representations directly from EEG recordings. However, most existing detectors are trained specifically for spindle detection and depend on expert annotations for representation learning. Sleep foundation models [5, 6] present an alternative by learning reusable physiological representations from extensive collections of polysomnographic recordings prior to adaptation for downstream tasks. This approach offers the possibility of transferring knowledge acquired during large-scale pretraining to dense EEG event detection.

1.2 Problem Statement and Objectives

Adapting a sleep foundation model for spindle detection introduces a temporal resolution mismatch. SleepFM [5] encodes several seconds of EEG into compact token-level

embeddings, while spindle boundaries require recovery at a much finer temporal scale. Increasing output resolution after tokenization does not guarantee the restoration of local information that may no longer be explicitly represented within the tokens.

Spindle detection further depends on localized spectral activity. While a pretrained temporal representation may implicitly encode this information, it does not directly preserve the evolution of spectral features within each token. The central objective of this thesis is to determine how an existing pretrained sleep representation can be adapted for dense spindle localization.

This study investigates whether the temporal representation learned by SleepFM [5] retains sufficient information for precise event detection. It also examines whether a frequency representation offers complementary local evidence and explores how both representations should interact prior to reconstruction at the original sampling resolution. Finally, it evaluates how annotation source and model architecture affect cross-dataset generalization.

1.3 Contributions

This thesis develops a dense spindle detector based on a frozen EEG representation derived from SleepFM. A task-specific frequency branch complements the pretrained representation by capturing localized spectral information. Both streams share the same token grid. They are integrated through residual cross-attention, where the temporal representation provides context to the frequency stream.

The frequency branch employs a Morlet representation and a structured readout to preserve intra-token temporal information. A dense decoder then reconstructs the fused representation at sample resolution, reintegrating these retained features.

Ablation studies evaluate the primary architectural choices related to frequency modeling, representation alignment, fusion and dense reconstruction. The final detector is further assessed based on event localization and physiological consistency. Cross-dataset experiments investigate the impact of the annotation source. They also test whether strong in-domain performance generalizes to broader datasets.

1.4 Thesis Organization

Chapter 2 introduces the physiological and signal-processing concepts relevant to spindle detection. Chapter 3 reviews automatic spindle detection and sleep foundation models, and identifies the research gap addressed in this thesis. Chapter 4 details the proposed architecture and training procedure. Chapter 5 evaluates the architectural choices and analyzes the final detector and its cross-dataset performance. Chapter 6 summarizes the main findings and limitations, and outlines directions for future research.

Chapter 2

Background

2.1 Sleep Physiology and Polysomnography

2.1.1 Polysomnographic Signals

Polysomnography (PSG) is the standard clinical and research method for objectively assessing sleep. It records multiple physiological signals overnight, allowing cerebral activity to be interpreted alongside ocular movements, muscle tone, cardiac activity and respiration [7]. This multimodal approach is essential because sleep stages are defined by combined physiological patterns rather than by a single signal.

Electroencephalography (EEG) records electrical activity generated by the brain via electrodes placed on the scalp. Electrooculography (EOG) measures eye movements using electrodes positioned near the eyes and electromyography (EMG) assesses skeletal muscle activity, most commonly beneath the chin. Electrocardiography (ECG) records cardiac electrical activity, while respiratory sensors monitor airflow and breathing effort. The synchronized interpretation of these signals forms the physiological basis for sleep-stage scoring and the analysis of shorter events occurring within individual stages.

2.1.2 Sleep Architecture and Stage Scoring

Sleep is conventionally categorized into wakefulness, non-rapid eye movement (NREM) sleep and rapid eye movement (REM) sleep. NREM sleep is further divided into stages N1, N2 and N3. N1 marks the transition from wakefulness to sleep, N2 is characterized by transient EEG events such as K-complexes and sleep spindles and N3 is dominated by high-amplitude slow-wave activity. REM sleep is characterized by rapid eye movements alongside markedly reduced skeletal muscle tone [7].

Standard sleep scoring divides the PSG recording into consecutive 30-second epochs, assigning each epoch to Wake, N1, N2, N3, or REM based on the physiological patterns observed during that interval. Representative EEG, EOG and EMG epochs are presented in Figure 2.1. The N2 segment contains a K-complex and a sleep spindle, N3 exhibits prominent slow-wave activity and REM sleep is characterized by rapid eye movements with reduced background muscle tone [8].

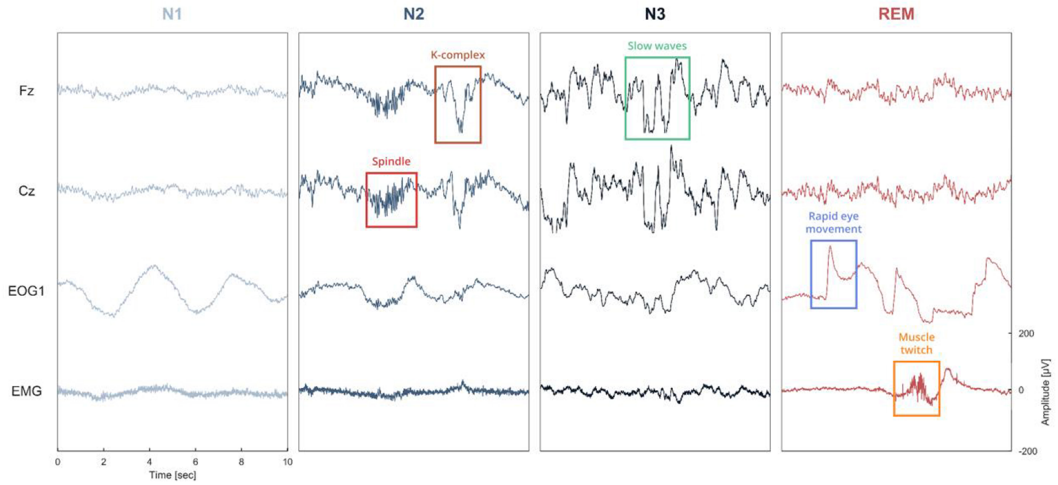

Figure 2.1: Representative EEG (F_z, C_z), EOG and EMG patterns during N1, N2, N3 and REM sleep. Reproduced from [8].

Epoch-level scoring effectively describes the global organization of sleep but lacks the temporal precision necessary to localize brief electrophysiological events. The spindle highlighted in the N2 excerpt occupies only a small fraction of the corresponding 10-second excerpt and may begin and end without affecting the sleep stage assigned to the surrounding interval. This distinction separates sleep macrostructure from microstructure, as localizing transient events requires signal representations and annotations with much finer temporal resolution than conventional stage labels.

2.1.3 EEG Acquisition and Spatial Organization

Among the physiological signals recorded during PSG, EEG is the primary modality considered in this thesis, as sleep spindles are defined by transient patterns of cerebral electrical activity. Scalp EEG records voltage fluctuations generated by the spatially aggregated activity of large neuronal populations, which are propagated through the tissues separating the brain from the electrodes.

An EEG channel represents the potential difference between two electrodes. In a referential montage, the voltage at an active electrode is compared with a common or local reference, whereas a bipolar montage measures the difference between two recording electrodes. Consequently, the resulting waveform depends not only on the underlying neural activity but also on the selected electrode positions and reference configuration.

Standardized electrode conventions establish a spatial framework for EEG acquisition. Figure 2.2 relates major cerebral regions to representative scalp electrode positions. Electrode prefixes indicate frontal, central, temporal, parietal, or occipital locations. Odd and even indices typically correspond to the left and right hemispheres, respectively, while the suffix z denotes the cranial midline. Channel names such as $C3$, $C4$ and Cz are thus interpreted according to their approximate scalp positions [9].

(a) Main cerebral regions used to describe EEG electrode locations.

(b) Representative EEG electrode positions distributed across the scalp.

Figure 2.2: Simplified spatial organization of scalp EEG acquisition. Adapted from [9].

The selected montage influences the appearance of physiological events in the recorded signal. Electrical activity is projected differently to each electrode and reference, which can alter the apparent amplitude, polarity and morphology of the same event across channels. Comparisons between recordings must therefore account for channel location and reference configuration, especially when a detector is trained and evaluated on data acquired with different montages.

EEG contains patterns across multiple temporal scales. Slow changes assist in identifying global sleep states, whereas brief oscillatory events occur locally within the continuous waveform. Sleep spindles fall into this latter category.

2.1.4 Sleep Spindles

A sleep spindle is a short burst of rhythmic EEG activity observed primarily during NREM stage N2, though it may also occur during N3 sleep. The waveform typically exhibits regular oscillations with a waxing and waning envelope. Spindles generally occupy the sigma frequency range of approximately 11 to 16 Hz and commonly last between 0.5 and 3 seconds [10, 11]. Their amplitude, dominant frequency and spatial expression vary across individuals and recording channels.

A common distinction separates slow and fast spindles within the sigma range. Slow spindles typically occupy lower frequencies around 11 to 13 Hz and are more prominent over frontal regions, whereas fast spindles generally occur around 13 to 16 Hz and show a stronger central distribution [11]. This division is approximate rather than absolute, since spindle frequency and topography vary continuously across subjects and recording conditions. It nevertheless shows that spindle activity is not characterized by a single frequency or scalp location, making both spectral content and channel selection relevant to its analysis.

The spindle illustrated in Figure 2.1 appears as a transient oscillation embedded within the surrounding N2 EEG. As the event gradually emerges from and fades into background activity, its onset and offset are not defined by sharp physiological bound-

aries. Other transient patterns and fluctuations in sigma-band energy may also resemble portions of a spindle, making precise boundary placement uncertain.

Sleep spindles originate from coordinated activity within thalamocortical circuits involving the thalamic reticular nucleus, thalamocortical relay cells and cortical populations [10, 11]. Rhythmic activity generated in the thalamus propagates to the cortex, while corticothalamic feedback modulates and synchronizes this activity. Figure 2.3 presents a simplified illustration of these interactions.

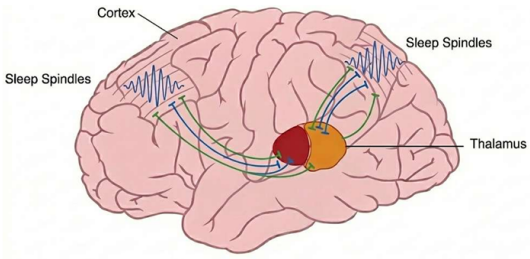

Figure 2.3: Simplified representation of the thalamo-cortical interactions involved in sleep spindle generation.

Sleep spindle analysis is relevant from both neuroscientific and clinical perspectives. Spindles reflect coordinated thalamocortical activity and are associated with processes such as memory consolidation, synaptic plasticity, sensory processing and sleep maintenance [10, 11]. Clinical studies have associated atypical spindle characteristics with schizophrenia [1] and major depressive disorder [3]. Altered spindle characteristics have also been reported in obstructive sleep apnea [4]. Their density, duration, amplitude and oscillatory frequency are therefore increasingly investigated as potential electrophysiological biomarkers of brain and sleep health.

Reliable estimation of these characteristics depends on accurate event detection and localization. Missed or duplicate detections directly affect spindle density, whereas inaccurate boundaries can bias duration and other event-level measurements, thereby influencing their physiological and clinical interpretation.

From a signal analysis perspective, a spindle combines a localized temporal waveform with a characteristic spectral signature. The raw EEG preserves its morphology and temporal structure, whereas a time-frequency representation highlights the transient concentration of sigma-band energy.

2.2 EEG Signal Representation for Spindle Analysis

EEG recordings generally contain several channels corresponding to different scalp derivations. The signal representations introduced in this section are applied independently to each EEG channel. A channel c is therefore represented by the one-dimensional signal x_c

and its temporal, Fourier, or wavelet representation is computed without combining information from other channels. Multichannel information can subsequently be combined by a detection model, as described later in this thesis.

2.2.1 Discrete EEG Signals and Preprocessing

An EEG recording originates from continuous electrical signals measured at the scalp, which are subsequently converted into discrete sequences by the acquisition system. The continuous signal from channel c is denoted as $s_c(t)$. Sampling at frequency f_s yields

$$x_c[n] = s_c\left(\frac{n}{f_s}\right), \qquad n = 0, \ldots, N-1$$

where N represents the total number of acquired samples. Consecutive samples are separated by $1/f_s$ seconds, meaning the sampling frequency defines the temporal grid for representing event boundaries.

Raw scalp EEG signals are subject to both physiological and acquisition-related variability. Preprocessing mitigates these sources of variability by limiting the signal to a relevant frequency range, resampling recordings to a common temporal grid and normalizing amplitudes across subjects and sessions.

These preprocessing operations can affect event localization by modifying brief oscillatory activity, temporal representation, or amplitude scale. Therefore, preprocessing should render recordings comparable without distorting spindle waveforms or shifting their boundaries.

2.2.2 Fourier and Short-Time Fourier Representations

Sleep EEG signals are non-stationary, as both their waveform and spectral composition change over time. While a temporal representation preserves the progression of the waveform and the timing of local changes, it does not indicate which oscillatory components generate the observed patterns.

The discrete Fourier transform provides a global frequency representation of a finite segment. For a sequence $x_c[n]$ of length N,

$$X_c[k] = \sum_{n=0}^{N-1} x_c[n] e^{-j2\pi kn/N}, \qquad f_k = \frac{k f_s}{N}$$

The magnitude of $X_c[k]$ reflects the contribution of each frequency to the analyzed segment. Since the transform summarizes the entire observation window, it can reveal sigma-band activity but does not specify when this activity occurs [12].

Temporal localization is introduced by applying the transform to successive portions of the signal. For a window $w[n]$ of length L, a hop size H and a transform size N_{FFT}, the short-time Fourier transform is

$$X_c[m,k] = \sum_{n=0}^{L-1} x_c[n+mH]\, w[n]\, e^{-j2\pi kn/N_{\mathrm{FFT}}}$$

where m indexes temporal frames and k frequency bins. The same transformation is applied independently to each channel and the spectrogram is obtained from the squared magnitude $S_c[m,k] = |X_c[m,k]|^2$.

The window shape and duration influence the representation of transient events by controlling spectral leakage and temporal resolution [13]. Successive windows are typically overlapped, allowing an event to be described from multiple nearby temporal positions.

Figure 2.4 illustrates both representations for the same event. In the temporal signal, the spindle manifests as a brief oscillatory burst with varying amplitude, whereas in the spectrogram, the same interval is depicted as a localized concentration of energy within a restricted frequency range.

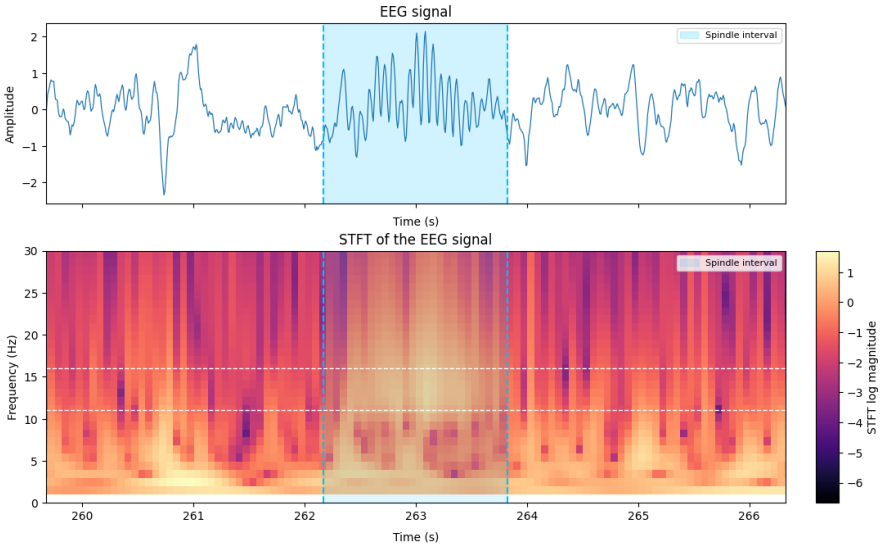

Figure 2.4: Example of a sleep spindle in the temporal and time-frequency domains. The top panel shows the EEG signal and spindle interval, while the bottom panel shows the corresponding spectrogram with sigma-band activity indicated.

2.2.3 Wavelet Representations

The short-time Fourier transform employs the same analysis window for all frequencies, resulting in uniform resolution across the spectrum. A window of sufficient length to separate closely spaced components within the sigma range imposes equivalent temporal smearing on both slow waves and fast oscillations, despite their differing time scales.

Wavelet transforms replace the fixed window by a family of functions obtained by dilating a single prototype ψ. The continuous wavelet transform (CWT) of x_c is

$$W_c(a,t) = \frac{1}{\sqrt{a}} \int x_c(u)\, \psi^* \left(\frac{u-t}{a} \right) du$$

where $a > 0$ denotes the scale and ψ^* is the complex conjugate of the mother wavelet [14]. Small scales analyze the signal over short intervals and capture rapid variations, whereas large scales encompass longer intervals. Because the scale is inversely related to the analyzed frequency, the duration of analysis adapts to the frequency under examination rather than remaining fixed.

For neurophysiological signals the prototype is commonly a complex Morlet wavelet, a complex exponential tapered by a Gaussian envelope. Expressed as a function of frequency,

$$\psi_f(t) = A_f \exp\left(-\frac{t^2}{2\sigma_f^2} \right) \exp\left(j2\pi f t \right), \qquad \sigma_f = \frac{n_c}{2\pi f}$$

where A_f is a normalization constant and n_c represents the number of oscillation cycles retained within the envelope. Parametrizing the envelope by the number of cycles, rather than by absolute duration, is the standard convention in this field, although reporting the full width at half maximum of the envelope has been suggested as less ambiguous [15].

Figure 2.5 illustrates this representation for a spindle event, where the transient oscillatory activity appears as an increased concentration of energy within the sigma band.

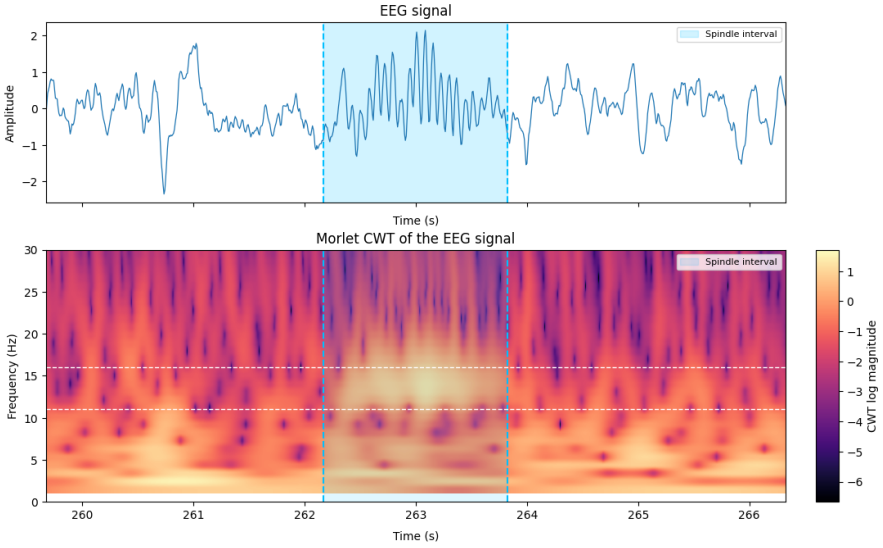

Figure 2.5: Example of a sleep spindle in the temporal and wavelet domains. The top panel shows the EEG signal and spindle interval, while the bottom panel shows the corresponding Morlet CWT, with sigma-band activity indicated.

2.2.4 Resolution and Complementary Views

Both transforms are subject to the same constraint. A short analysis localizes brief events precisely but resolves nearby frequencies poorly, while a longer one does the opposite [13]. For a Morlet wavelet, the temporal spread σ_f and the spectral spread $\sigma_f^{\mathrm{Hz}} = f/n_c$ satisfy $\sigma_f \, \sigma_f^{\mathrm{Hz}} = 1/2\pi$, a constant product that no parameter choice can circumvent.

The two families differ in how they distribute this fixed area over the time–frequency plane, as illustrated in Figure 2.6. The fixed absolute window duration of the Short-Time Fourier Transform (STFT) produces resolution cells with identical dimensions across the spectrum. In contrast, the fixed number of oscillation cycles used by a wavelet family causes its cells to contract in time and widen in frequency as the analyzed frequency increases. Within the sigma range, the Gaussian temporal spread $\sigma_f = n_c/(2\pi f)$ is typically on the order of several tens of milliseconds, whereas the duration corresponding to n_c complete oscillation cycles is n_c/f.

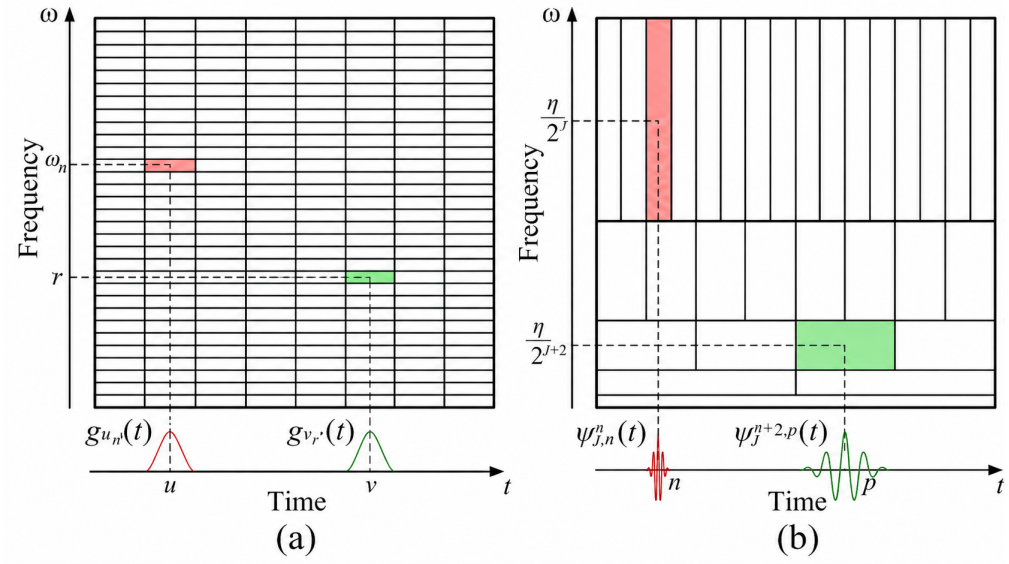

Figure 2.6: Time-frequency resolution cells of (a) the STFT and (b) a multiresolution wavelet decomposition. Reproduced from [16].

Sampling density is distinct from resolution. The frame spacing H/f_s and the bin spacing f_s/N_{FFT} can both be refined, yet a smaller hop only produces more overlapping frames and zero-padding only a denser grid, neither adding information beyond that contained in the analysis window. Once the frame spacing is set, both transforms yield coefficients on a regular temporal grid, so the choice between them rests on resolution behavior rather than on alignment.

2.3 Sleep Spindle Annotation and Dense Temporal Labels

2.3.1 Expert Annotation and Boundary Uncertainty

Sleep spindles are typically identified through visual inspection of the EEG by expert scorers. Upon recognition of a transient oscillatory pattern, the scorer assigns onset and offset times to delineate the event [7].

For a recording containing N_e annotated spindles, the intervals can be represented as

$$\mathcal{E} = \left\{ \left(t_i^{\text{on}}, t_i^{\text{off}} \right) \right\}_{i=1}^{N_e}$$

where t_i^{on} and t_i^{off} denote the onset and offset of event i. In contrast to epoch-level labels, this representation preserves both the temporal location and duration of each spindle.

These boundaries are not perfectly objective because spindle activity typically emerges gradually from the background EEG and subsequently fades. As a result, scorers may identify the same event but select different boundaries, or disagree on whether to annotate weak oscillatory activity. The resulting intervals also depend on the scoring procedure and EEG derivation, since the same event may appear differently across channels and scorers may apply the visual criteria with varying strictness. Therefore, expert annotations should be interpreted as temporal references defined by a particular protocol, rather than as precise physiological boundaries.

Consensus annotation mitigates reliance on individual decisions by aggregating the input of multiple scorers. For example, the Massive Online Data Annotation (MODA) [17] dataset derives spindle labels from repeated annotations and group agreement. This approach yields a more stable reference, although it does not eliminate the uncertainty associated with gradual event boundaries.

Annotation variability can influence both training and evaluation. A detector trained on one reference may not replicate annotations produced under a different scoring procedure, even if the underlying EEG is unchanged [18]. Consequently, small localization errors near an onset or offset should be interpreted in the context of this variability.

2.3.2 Conversion from Event Intervals to Dense Labels

Dense spindle detection requires target labels that are aligned with the EEG sampling grid. Each annotated interval is thus converted into a binary sequence that distinguishes spindle samples from the surrounding background, as illustrated in Figure 2.7.

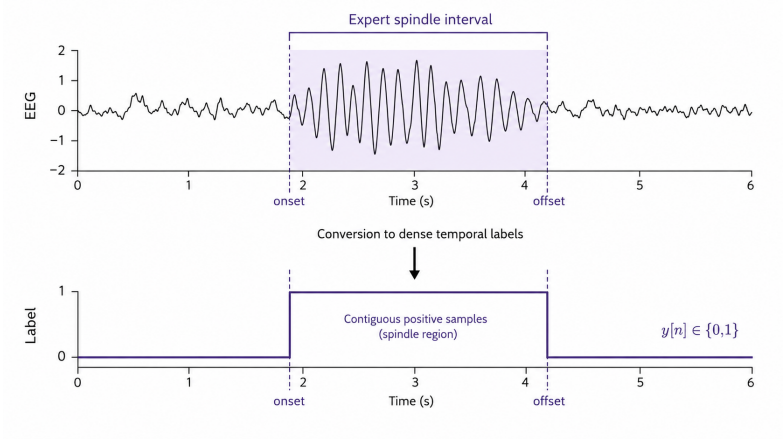

Figure 2.7: Conversion of an expert-annotated spindle interval into dense temporal labels.

For an EEG signal $x[n]$ sampled at frequency f_s, the time associated with sample n is

$$t_n = \frac{n}{f_s}$$

The dense label sequence is then defined as

$$y[n] = \begin{cases} 1 & \text{if } t_n \in \left[t_i^{\text{on}}, t_i^{\text{off}}\right] \text{ for at least one event } i \\ 0 & \text{otherwise} \end{cases}$$

Consecutive positive samples preserve the temporal extent of the annotated spindle, whereas transitions between label values define its onset and offset on the discrete sampling grid. This dense representation preserves the temporal structure needed to recover event boundaries and duration.

2.3.3 Class Imbalance in Continuous EEG

Sleep spindles constitute only a small fraction of continuous EEG recordings. Even during NREM stage N2, most samples correspond to background activity, resulting in a substantial predominance of negative labels over positive ones.

This imbalance renders overall accuracy an uninformative metric, as a detector that predicts background for nearly all samples may achieve high accuracy while failing to identify most spindle events. It also affects optimization, as errors accumulated over the background can dominate the learning objective and encourage overly conservative predictions.

Once point-wise predictions are reconstructed into events, missed spindles reduce recall, while additional predicted intervals reduce precision. Therefore, evaluation should consider both sample-level predictions and the events derived from them.

2.4 Evaluation of Sleep Spindle Detection

Automatic spindle detection is evaluated at both the sample and event levels. Sample-level metrics assess whether each temporal position is assigned to the correct class, while event-level metrics determine whether predicted intervals correspond to individual expert annotations. These perspectives are complementary because accurate sample classification does not guarantee correct event reconstruction.

2.4.1 Sample-Level Evaluation

A probability sequence $\hat{y}$ is converted into a binary sequence $\hat{\mathbf{Y}}$ by applying a decision threshold η,

$$\hat{Y}[n] = \begin{cases} 1 & \text{if } \hat{y}[n] \geq \eta, \\ 0 & \text{otherwise.} \end{cases}$$

The predicted labels are compared with the reference sequence $y[n]$. At the sample level, a true positive is a position labeled as spindle in both sequences, a false positive is predicted as spindle but belongs to the background and a false negative is part of an annotated spindle but classified as background.

Sample-level precision and recall are defined as

$$\text{Precision}_{\text{sample}} = \frac{TP_{\text{sample}}}{TP_{\text{sample}} + FP_{\text{sample}}}$$

$$\text{Recall}_{\text{sample}} = \frac{TP_{\text{sample}}}{TP_{\text{sample}} + FN_{\text{sample}}}$$

The combination of precision and recall yields

$$F_{1,\text{sample}} = 2\frac{\text{Precision}_{\text{sample}}\,\text{Recall}_{\text{sample}}}{\text{Precision}_{\text{sample}} + \text{Recall}_{\text{sample}}}$$

These metrics quantify the degree to which predicted positive samples align with annotated spindle regions and are sensitive to boundary displacement. Overall accuracy is less informative because the predominance of background EEG can dominate the result.

Sample-level agreement does not fully characterize event reconstruction. A long prediction may overlap multiple annotated spindles, while a single spindle may be divided into several predicted regions. Therefore, point-wise metrics should be complemented by evaluations that treat contiguous positive regions as individual events [19].

2.4.2 Event Matching and Intersection over Union

Let the reference spindle intervals be denoted by

$$\mathcal{G} = \{G_i\}_{i=1}^{N_G}$$

and let the intervals reconstructed from the binary prediction sequence be

$$\mathcal{P} = \{P_j\}_{j=1}^{N_P}$$

The temporal overlap between a predicted event P_j and a reference event G_i can be measured using the Intersection over Union,

$$\mathrm{IoU}(P_j, G_i) = \frac{|P_j \cap G_i|}{|P_j \cup G_i|}$$

where $|\cdot|$ denotes interval duration. The IoU equals zero when the events do not overlap and approaches one as their boundaries become increasingly similar.

A predicted interval is eligible for matching when

$$\mathrm{IoU}(P_j, G_i) \geq \tau$$

where τ is the event-overlap threshold. Matching is performed under a one-to-one constraint, ensuring that each prediction and each reference event contribute to at most one pair. This prevents a broad prediction from being counted as multiple correct detections and avoids assigning several predictions to the same annotated spindle.

The selected value of τ determines the strictness of the evaluation. A lower threshold accepts events with partial temporal overlap, while a higher threshold requires closer agreement between event boundaries. Consequently, performance is reported at multiple IoU thresholds.

2.4.3 Event-Level Precision, Recall and F1 Score

Let N_M denote the number of matched prediction-reference pairs. Event-level precision and recall are defined as

$$\mathrm{Precision}_{\mathrm{event}} = \frac{N_M}{N_P}$$

$$\mathrm{Recall}_{\mathrm{event}} = \frac{N_M}{N_G}$$

The corresponding event-level F1 score is

$$F_{1,\mathrm{event}} = 2\frac{\mathrm{Precision}_{\mathrm{event}}\,\mathrm{Recall}_{\mathrm{event}}}{\mathrm{Precision}_{\mathrm{event}} + \mathrm{Recall}_{\mathrm{event}}}$$

An unmatched predicted interval is considered a false positive, while an unmatched reference interval is considered a false negative.

The notation $F_1@\tau$ indicates that event matching was performed using the IoU threshold τ. Consequently, $F_1@0.2$ refers to an event-level F1 score for which a prediction is considered eligible for matching when its IoU with a reference event is at least 0.2. This overlap threshold should not be confused with the probability threshold η, which is applied earlier to convert the model output into a binary temporal sequence.

The aggregation procedure also influences the reported results. Pooling all events assigns greater weight to recordings with many spindles, while computing metrics separately for each subject preserves inter-subject variability. The chosen convention must remain consistent when comparing models.

2.4.4 Consistency of Detected Spindle Statistics

Event-level F1 measures detection accuracy but does not ensure that reconstructed events preserve physiologically meaningful spindle characteristics.

For a recording containing N_e detected events over an analyzed duration T seconds, spindle density can be expressed in events per minute as

$$D = \frac{60N_e}{T}$$

The mean event duration is

$$\overline{d} = \frac{1}{N_e} \sum_{i=1}^{N_e} \left(t_i^{\text{off}} - t_i^{\text{on}} \right)$$

Missed or additional detections modify spindle density, while inaccurate boundaries affect the estimated duration. The same intervals may also be used to derive amplitude or frequency characteristics from the enclosed EEG waveform.

Agreement between predicted and expert-derived statistics can be measured across S recordings. For a spindle characteristic q, the mean absolute error is

$$\text{MAE}(q) = \frac{1}{S} \sum_{s=1}^{S} |\hat{q}_s - q_s|$$

where q_s and $\hat{q}_s$ denote the reference and predicted values for recording s. Correlation provides complementary information by indicating whether between-subject variation is preserved, although systematic bias may persist even when correlation is high.

Recent spindle detectors evaluate not only localization performance but also whether predicted density and duration are consistent with expert-derived estimates [20]. Combining sample-level overlap, event-level matching and spindle-statistic consistency yields a more comprehensive assessment than any single metric alone.

Chapter 3

Related Work

3.1 Automatic Sleep Spindle Detection

Automatic sleep spindle detection has become important in large-scale sleep studies because manual scoring is both time-consuming and difficult to apply systematically. Measures such as spindle density, duration, amplitude and frequency are widely used in sleep research because they have been associated with memory consolidation, cognitive function and neurological or psychiatric conditions [10, 11]. The primary challenges include the short duration of spindles, inter-subject variability and their occurrence within non-stationary EEG signals. These challenges have driven the development of successive generations of automatic detectors, ranging from physiologically motivated signal-processing methods to learned models that enhance localization and robustness.

3.1.1 Signal-Processing Detectors

Signal-processing detectors translate prior knowledge of spindle physiology into explicit procedures applied to the EEG. Most methods first isolate activity within a predefined sigma frequency range, then compute local measures using sliding windows. Commonly extracted features include signal amplitude or envelope, sigma power, spectral ratios and oscillatory regularity metrics. Candidate intervals are identified when these features exceed predefined thresholds and are then merged or removed according to duration and temporal-separation constraints [21].

Threshold-based methods differ primarily in the features used to define candidate spindles and in whether decision thresholds are fixed, adaptive, or normalized relative to the surrounding EEG. The A7 detector, proposed by Lacourse et al. [22], exemplifies this methodology. A7 analyzes a single EEG channel and integrates four complementary metrics derived from both broadband and sigma-filtered signals: absolute sigma power, relative sigma power, sigma covariance and sigma correlation. The metrics are computed on a short sliding window and a candidate is retained only where all four simultaneously exceed their thresholds. This multi-criterion formulation preserves physiological interpretability instead of relying on a single amplitude or power threshold, yet detection

remains directly influenced by the selected frequency range, normalization, thresholds and post-processing rules.

A second category of methods seeks to isolate spindle-related oscillations from other EEG components prior to event detection. For instance, DETOKS decomposes the observed signal into transient, low-frequency and oscillatory components through sparse optimization, with spindle candidates identified from the oscillatory component [23]. Related techniques rely on wavelet decompositions or matching pursuit. Spindler applies this idea with Gabor atoms and selects its detection parameters from stable regions of spindle-property surfaces rather than from agreement with a single annotation set [24]. These decomposition-based methods represent spindle morphology more finely than direct thresholding, although their outputs still depend on the chosen dictionary, frequency constraints and selection thresholds.

In summary, signal-processing detectors can operate without learning a high-capacity model from large annotated datasets. Their explicit reliance on sigma-band activity facilitates alignment with the physiological definition of a spindle. However, fixed signal assumptions may not generalize effectively across subjects, age groups, recording devices, or clinical populations. Furthermore, the estimation of event onset and offset is highly sensitive to window length and other parameters. These limitations have led to the development of methods that preserve interpretable EEG features while learning decision rules from annotated data.

3.1.2 Classical Machine Learning Approaches

Classical machine learning methods preserve an explicit feature extraction stage but replace fixed decision thresholds with a classifier learned from data. For an EEG window x_i, a deterministic mapping ϕ first produces a feature vector $\mathbf{z}_i = \phi(x_i) \in \mathbb{R}^d$, which may contain temporal, spectral, morphological, or time-frequency descriptors. A classifier g_θ then estimates the probability that the window contains a spindle, $\hat{p}_i = g_\theta(\mathbf{z}_i)$. This formulation enables multiple EEG characteristics to contribute jointly to the decision and allows the classification boundary to be estimated from expert annotations rather than specified through independent thresholds [21].

Supervised approaches commonly employ support vector machines, decision trees, nearest-neighbor methods, or ensemble classifiers. MUSSDET, for example, applies feature selection, normalization and multivariate classification to single-channel EEG epochs [25]. Its support vector machine combines several descriptors into a single discriminative decision, which is more flexible than requiring each descriptor to cross an independently selected threshold. However, the reported performance of MUSSDET differs across the Montreal Archive of Sleep Studies (MASS) [26] and the DREAMS Sleep Spindles [27] database. Although these results are not directly comparable because the datasets and annotation protocols differ, they illustrate the difficulty of transferring fixed feature distributions and learned decision boundaries across recording conditions.

Other approaches combine time-frequency feature extraction with classifiers designed for imbalanced data. Kinoshita et al. use the synchrosqueezed wavelet transform to obtain a concentrated time-frequency representation and apply RUSBoost to distinguish

spindle from non-spindle observations [28]. Random undersampling reduces the dominance of background windows. This directly addresses a central property of continuous EEG recordings, in which spindle intervals represent only a small fraction of the signal. Nevertheless, the learned classifier remains dependent on the selected wavelet representation, the construction of positive and negative examples and the temporal window used to describe each candidate event.

Unsupervised and probabilistic approaches reduce direct dependence on event labels while retaining a feature-based representation. Patti et al. [29] model spindle candidates using multivariate Gaussian mixture models from sigma-band envelope features obtained through a Hilbert transform. These methods estimate probability distributions rather than discriminative boundaries, but the correspondence between mixture components and physiologically valid spindle events is not guaranteed and remains constrained by the selected features and distributional assumptions.

Overall, classical machine learning methods represent an intermediate approach between explicit signal-processing rules and end-to-end neural models. They can learn combinations of handcrafted features and address class imbalance, but their representational capacity remains limited by the information retained by ϕ. Fixed-window classification also yields coarse temporal decisions, with event boundaries typically recovered by merging neighboring positive windows and applying duration constraints. These limitations motivated models that learn both the representation and decision function directly from EEG signals.

3.1.3 Transition to Deep Learning-Based Detection

The transition from classical machine learning to deep learning primarily reflects a change in how the representation is learned. In feature-based pipelines, an explicit mapping ϕ transforms each EEG window into a predefined vector of temporal, spectral, or morphological descriptors prior to classification. In contrast, deep models learn an encoder E_θ jointly with the prediction function, producing a task-adapted representation $\mathbf{h}_i = E_\theta(x_i)$ and spindle probability $\hat{p}_i = g_\psi(\mathbf{h}_i)$. This approach reduces reliance on manually selected features and enables the representation to be optimized based on expert spindle annotations [21].

The transition was gradual rather than a complete replacement of signal-processing knowledge. Early deep spindle detectors often combined learned features with physiological inputs or handcrafted descriptors. SpindleNet, for example, processes raw EEG together with the envelope of the sigma-band signal and combines the learned representation with power features for spindle classification [30]. Similarly, Pan et al. integrate convolutional feature extraction with wavelet-informed processing and decision-tree validation for spindle detection in patients with acute disorders of consciousness [31]. These hybrid designs preserve explicit spindle-related information while allowing neural networks to learn discriminative waveform patterns that are difficult to encode through fixed rules. More recent approaches increasingly formulate spindle detection as an end-to-end temporal prediction problem, in which representation learning, contextual modeling and event localization are optimized jointly [19, 20].

Deep learning enhances representation flexibility and enables joint optimization of feature extraction, temporal context modeling and event prediction. However, these benefits increase dependence on labeled data, annotation quality, model capacity and training distribution. Learned features are also less interpretable than handcrafted physiological descriptors and strong performance on a single dataset does not guarantee robustness across different recording conditions. The architectural strategies for addressing local morphology, temporal context, dense localization and long-range dependencies are discussed in the following section.

3.2 Neural Architectures for EEG Event Detection

Modern neural detectors progressively transform the EEG from local waveform patterns into temporally contextualized representations that can support precise event localization. Convolutional layers capture spindle morphology at several temporal scales, while recurrent processing extends this information beyond the local receptive field. Encoder-decoder structures then restore the resolution required for point-wise prediction and attention mechanisms regulate which intermediate features are transferred to the decoder. Figure 3.1 provides a representative example through BLAST [20], where raw and sigma-filtered EEG are encoded, contextualized with bidirectional recurrent layers, filtered through attention-gated skip connections and decoded into dense spindle predictions.

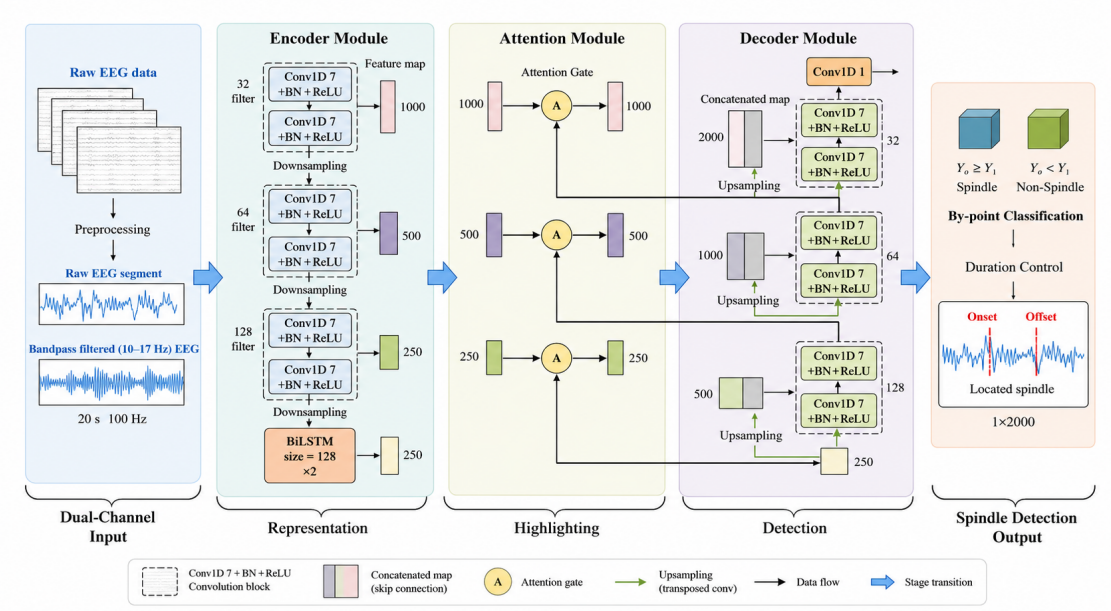

Figure 3.1: Representative integration of convolutional encoding, bidirectional temporal modeling, attention-gated skip connections and dense decoding for sleep spindle localization. Adapted from [20].

3.2.1 Convolutional Feature Extraction

Convolutional neural networks learn local filters that are shared across the temporal dimension of the EEG [32]. For a multichannel input $x_c[t]$, a one-dimensional convolutional feature map can be expressed as

$$h_m[t] = \sigma\left(b_m + \sum_{c=1}^{C}\sum_{\tau=0}^{K-1} w_{m,c}[\tau]\, x_c[t-\tau]\right) \tag{3.1}$$

where K is the kernel length, $w_{m,c}$ denotes the learned filter coefficients, b_m is a bias term and σ is a nonlinear activation function. Weight sharing allows the same filter to detect local waveform patterns regardless of their absolute temporal position. This property is particularly suitable for spindle detection, as spindle morphology can appear at various locations while maintaining characteristic oscillatory, amplitude and envelope features.

Convolutional detectors vary in the extent to which physiological preprocessing is incorporated alongside learned feature extraction. SpindleNet [30] processes raw EEG together with the envelope of a sigma-band signal and integrates the learned representation with power-related descriptors. Pan et al. [31] similarly combine convolutional feature extraction with wavelet-based signal processing and decision-tree validation in a clinical population with acute disorders of consciousness. The left panel of Figure 3.1 illustrates another hybrid approach. BLAST [20] receives both raw EEG and a band-pass filtered signal targeting spindle-related frequencies, enabling the encoder to jointly process broadband waveform morphology and an enhanced oscillatory component. These architectures show that deep feature learning does not necessarily replace physiological priors. Instead, handcrafted transformations can be retained as complementary input channels.

The BLAST [20] encoder shown in Figure 3.1 progressively increases the number of convolutional filters while reducing the temporal resolution of the feature sequence. This hierarchical structure expands the effective receptive field, enabling deeper layers to integrate short waveform fragments into wider event representations. Similar multiscale principles are evident in SpindleU-Net [19], which employs relatively large temporal kernels to capture dependencies extending across spindle waveforms.

This expanded context introduces a fundamental trade-off. Pooling and strided convolutions improve computational efficiency and increase the receptive field, but they also reduce the temporal resolution required for precise onset and offset localization. Consequently, while convolutional encoders are effective for learning local and multiscale EEG morphology, additional mechanisms are generally necessary to model longer temporal dependencies and to reconstruct boundaries at the original signal resolution.

3.2.2 Recurrent Models and Temporal Context

Recurrent neural networks complement convolutional feature extraction by maintaining a hidden representation that evolves across a sequence. Gated recurrent architectures

such as long short-term memory networks regulate the preservation and removal of information through learned gates, which improves the modeling of dependencies extending beyond the local receptive field of individual convolutions.

Temporal context is essential because increased sigma-band activity alone does not reliably identify all spindles. Background EEG, artifacts and other transient oscillations may share similar local properties. The temporal evolution surrounding a candidate event can indicate whether the oscillation exhibits a coherent onset, sustained body and termination. SpindleNet [30] integrates convolutional feature extraction with recurrent processing in a system designed for real-time spindle detection. This architecture exemplifies a common division of functions, where convolutional layers detect local signal patterns and recurrent layers model the evolution of these patterns across consecutive temporal positions.

Bidirectional recurrent models extend this principle by processing the feature sequence in both temporal directions,

$$\overrightarrow{\mathbf{h}}_t = f_{\rightarrow}\left(\mathbf{x}_t, \overrightarrow{\mathbf{h}}_{t-1}\right), \qquad \overleftarrow{\mathbf{h}}_t = f_{\leftarrow}\left(\mathbf{x}_t, \overleftarrow{\mathbf{h}}_{t+1}\right) \tag{3.2}$$

The forward and backward states can then be combined to represent each position using both preceding and subsequent EEG activity. In Figure 3.1, two BiLSTM layers are placed at the lowest-resolution stage of the encoder. This arrangement allows BLAST [20] to contextualize the multiscale convolutional representation before it is transferred to the decoder. Access to both temporal directions is particularly relevant for offline localization because the complete morphology surrounding a candidate event is available when estimating its boundaries.

The choice between unidirectional and bidirectional recurrence reflects a practical trade-off in the context available to the model. A unidirectional model reads each position from present and preceding samples only, so it can operate on a streaming signal without access to future samples. A bidirectional model instead reads the full sequence in both directions. It thus incorporates future context to refine each estimate, but it requires the complete segment to be available and cannot run on a streaming signal.

Recurrent context alone cannot recover information lost due to encoder downsampling. While a BiLSTM can enrich low-resolution features with temporal dependencies, it does not directly reconstruct a sample-level prediction sequence. Consequently, recurrent modules are typically embedded within encoder-decoder architectures that separately address contextual modeling and temporal resolution recovery.

3.2.3 Encoder-Decoder and Dense Segmentation Models

Dense temporal segmentation formulates spindle detection as the assignment of a probability to each temporal position. Given an EEG segment $\mathbf{x}$ of length T, the model estimates

$$\hat{y} = f_\theta(\mathbf{x}), \qquad \hat{y} \in [0,1]^T \tag{3.3}$$

where $\hat{y}[t]$ represents the probability that temporal position t belongs to a spindle. Contiguous positive regions can subsequently be converted into events with estimated onset,

offset and duration. This approach contrasts with fixed-window classification, in which a single decision is assigned to an entire segment and event boundaries must be inferred indirectly.

U-Net architectures [33] provide a natural framework for dense temporal prediction because they combine a contracting encoder with an expanding decoder. The encoder progressively transforms the input into lower-resolution features with increasingly broad receptive fields. The decoder then upsamples these representations until the temporal resolution required for point-wise prediction is recovered. Skip connections transfer higher-resolution encoder features to corresponding decoder stages, thereby preserving local details that may be lost at the bottleneck.

SpindleU-Net [19] adapts U-Net to one-dimensional single-channel EEG and generates point-wise spindle labels from variable-length inputs. The encoder captures broader temporal context, while the decoder restores the resolution necessary for localization. Skip connections preserve local details and attention mechanisms filter the transferred features before prediction. The principal contribution of this architecture is the formulation of spindle detection as dense temporal segmentation.

The decoder module in Figure 3.1 illustrates how this resolution recovery is implemented in BLAST. At each stage, lower-resolution features are upsampled and concatenated with features transferred from the corresponding encoder level. A final one-dimensional convolution produces a point-wise classification sequence, after which duration constraints are used to derive localized spindle intervals [20].

Subsequent U-Net-based detectors extend the dense segmentation framework to address limitations arising from annotation quality and clinical variability. SUMO [34] applies a slim U-Net to the MODA dataset and reports event-level agreement exceeding that of the reference A7 detector [22] and of most individual experts. Building on this line, Tian and Wei introduce a label optimization module for a detector trained using the same MODA annotations [35]. BLAST [20] integrates BiLSTM layers and attention-gated skip connections into the encoder-decoder structure. Pan et al. [31] combine neural detection with a separate decision-tree validation stage in a clinical setting.

Dense encoder-decoder models remain constrained by the resolution and supervision available during training. Downsampling may blur short transitions and upsampling cannot fully reconstruct information discarded by the encoder. Sample-level targets are derived from expert event annotations, with onset and offset times that may vary between scorers. While dense segmentation is well suited to spindle localization, boundary accuracy continues to depend on preserving temporal detail and managing annotation uncertainty.

3.2.4 Attention-Based and Transformer Models

Attention mechanisms apply data-dependent weighting to features rather than transferring or combining all information uniformly. For query, key and value matrices Q, K

and V, scaled dot-product attention is defined as

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{3.4}$$

where d_k is the dimensionality of the key vectors [36]. The similarity between queries and keys determines the contribution of the associated values, allowing the representation to emphasize information considered relevant to each prediction.

In spindle-specific architectures, attention is commonly used to refine convolutional or encoder-decoder features rather than replace the complete detection pipeline. SpindleU-Net [19] incorporates attention to increase the contribution of features associated with spindle-relevant regions. BLAST [20] applies attention gates to the skip connections linking its encoder and decoder. The central panel of Figure 3.1 shows one gate at each temporal scale. Encoder features are filtered before concatenation with the decoder representation, which limits the direct transfer of background activity while preserving higher-resolution information relevant to the predicted event.

Gated attention differs from global self-attention because it filters an existing encoder feature map using decoder context.

Transformers stack multi-head self-attention and feed-forward layers [36]. Self-attention allows each token to interact directly with distant positions, while positional information preserves temporal order. For EEG, the waveform is typically compressed into a shorter token sequence before Transformer processing.

This broader context is useful for sleep analysis, but temporal compression may reduce boundary precision and does not inherently capture local spindle morphology. As a result, specialized spindle detectors more frequently combine attention with convolutional, recurrent, or U-Net components [21]. Transformers are more prominent in sleep foundation models, where large-scale pretraining supports longer-range and multimodal representations.

The architectures reviewed in this section progressively enrich the EEG representation before dense prediction. Local waveform information is first extracted and placed within a broader temporal context, after which the signal resolution is recovered for event localization. Attention can further control which intermediate features contribute to the final prediction, as illustrated by BLAST in Figure 3.1. Despite these advances, such representations are still learned mainly from task-specific spindle annotations. Sleep foundation models offer a different approach by learning reusable representations from large-scale PSG data before adaptation to fine-grained temporal events.

3.3 Sleep Foundation Models

A foundation model is a large model pretrained on broad unlabeled data that learns general-purpose representations reusable across many downstream tasks. Rather than training a separate model for each task, a single pretrained encoder is adapted with limited task-specific data.

The specialized detectors reviewed in previous sections instead learn representations from task-specific annotations. Sleep foundation models bring the foundation-model paradigm to physiological signals. They pretrain reusable encoders on extensive collections of PSG recordings before adaptation to downstream tasks. This approach is especially pertinent to sleep analysis. Expert annotations are expensive to obtain, while unannotated physiological signals are abundant.

3.3.1 Pretraining Objectives and Large-Scale Sleep Representation Learning

Self-supervised learning obtains supervisory signals directly from recorded data. Contrastive objectives align related observations, whereas reconstruction objectives require the model to recover masked or corrupted information. The selected objective influences which physiological properties remain accessible during downstream adaptation.

SleepFM [5] was pretrained on over 585000 hours of multimodal PSG data collected from approximately 65000 participants. The recordings are segmented into five-second tokens and contextualized over longer intervals. The leave-one-out contrastive objective aligns one modality with information aggregated from the remaining signals, thereby encouraging the representation to capture physiological structure shared across the PSG.

The resulting embeddings transfer to several downstream tasks ranging from sleep analysis to subject-level clinical prediction [5]. However, these tasks are primarily evaluated at the epoch, recording, or subject level. SleepFM is not explicitly trained to preserve the boundaries of short EEG events.

SleepGPT [6] simultaneously pretrains temporal and frequency-domain representations. Contrastive alignment and matching objectives relate these two domains, while masked reconstruction preserves local signal information. This approach is particularly relevant for spindle detection, as spindle morphology is defined by both temporal evolution and oscillatory content.

OSF [37] evaluates multiple self-supervised objectives within a unified evaluation framework. The results suggest that gains from scaling also depend on the model's ability to accommodate heterogeneous datasets and varying channel configurations.

3.3.2 Multimodal and Channel-Adaptive Representations

PSG datasets rarely share identical sensor configurations. EEG derivations can vary across cohorts and other physiological modalities may be absent or recorded with different devices. Consequently, a model constrained to a fixed montage may transfer poorly to new datasets.

SleepFM [5] addresses this challenge by using modality-aware encoding and channel-agnostic attention pooling. Channel-level representations are aggregated without assuming a fixed ordering and leave-one-out contrastive learning encourages information from one modality to be predictable from the others. As a result, the model can benefit from multimodal pretraining even when only EEG data are available during downstream adaptation.

This property is particularly relevant to the present thesis. SleepFM was pretrained on multimodal PSG, while the spindle detector primarily utilizes EEG. Therefore, the potential benefit of multimodal pretraining lies in the structure learned by the EEG representation, rather than in requiring all modalities during inference.

SleepGPT also accommodates varying montages by processing channels separately prior to Transformer interaction and masking unavailable inputs [6]. However, OSF [37] demonstrates that exposure to heterogeneous recordings does not inherently confer robustness to missing channels. Channel adaptability may therefore need to be explicitly encouraged during pretraining.

3.3.3 Temporal Granularity and Adaptation to Downstream Tasks

The utility of a pretrained representation partly depends on the temporal scale preserved by the encoder. Sleep staging typically operates over thirty-second epochs, whereas disease prediction may aggregate information across an entire night. In contrast, spindle detection requires a finer temporal representation, as events last only a few seconds and their boundaries must be localized precisely.

SleepFM [5] retains a sequence of five-second tokens prior to longer-range contextualization. These tokens preserve greater temporal detail than pooled epoch or recording representations, yet remain coarser than the sample-level output required for spindle localization. Consequently, a dedicated decoder is necessary to transform the pretrained sequence into dense predictions.

SleepGPT [6] provides direct evidence that a sleep foundation model can support spindle detection. Its pretrained representations are linked to a lightweight prediction head that generates spindle probabilities over time.

This result demonstrates that general-purpose sleep pretraining can be adapted for tasks beyond coarse classification. However, SleepGPT simultaneously introduces pretraining, time-frequency modeling and a dedicated spindle head. As a result, its experiments do not fully isolate the contribution of each individual component.

SleepMaMi [38] addresses temporal granularity using separate micro- and macro-level encoders. Short physiological segments are first encoded locally, followed by a second model that captures dependencies across the entire night. This hierarchical approach is conceptually relevant to spindle detection, as local events occur within a broader sleep context. However, SleepMaMi has not been directly evaluated on spindle localization.

Overall, large-scale PSG pretraining can produce transferable physiological representations, but the reviewed models also reveal a mismatch between the temporal granularity learned during pretraining and the precision required for spindle localization. SleepGPT shows that this gap can be reduced through task-specific adaptation, while SleepMaMi emphasizes the importance of preserving local information within broader sleep context. For SleepFM, the remaining question is whether its pretrained temporal representation retains sufficient spectral and boundary information for dense spindle detection. This motivates the frequency-aware modeling discussed in the following section.

3.4 Joint Time-Frequency Modeling

Adapting a sleep foundation model for spindle detection raises two distinct questions. The first concerns how explicit spectral information should interact with a pretrained temporal representation. The second addresses the compatibility of features generated by independently trained encoders. Fusion governs the exchange of information between streams, while alignment constrains the correspondence between their representations.

Sleep spindles are defined by both their waveform evolution and localized sigma-band activity. Temporal EEG retains event morphology and local temporal structure, whereas a time-frequency representation explicitly highlights the evolution of spectral energy. Although both representations derive from the same signal, they emphasize distinct properties.

SleepGPT [6] provides a direct example of joint time-frequency modeling. As shown in Figure 3.2, temporal and frequency-domain PSG signals are initially tokenized using separate channel-wise convolutions. The resulting embeddings are concatenated and processed within a unified Transformer architecture. Lower layers maintain domain-specific transformations, while subsequent layers facilitate shared processing between the two token types.

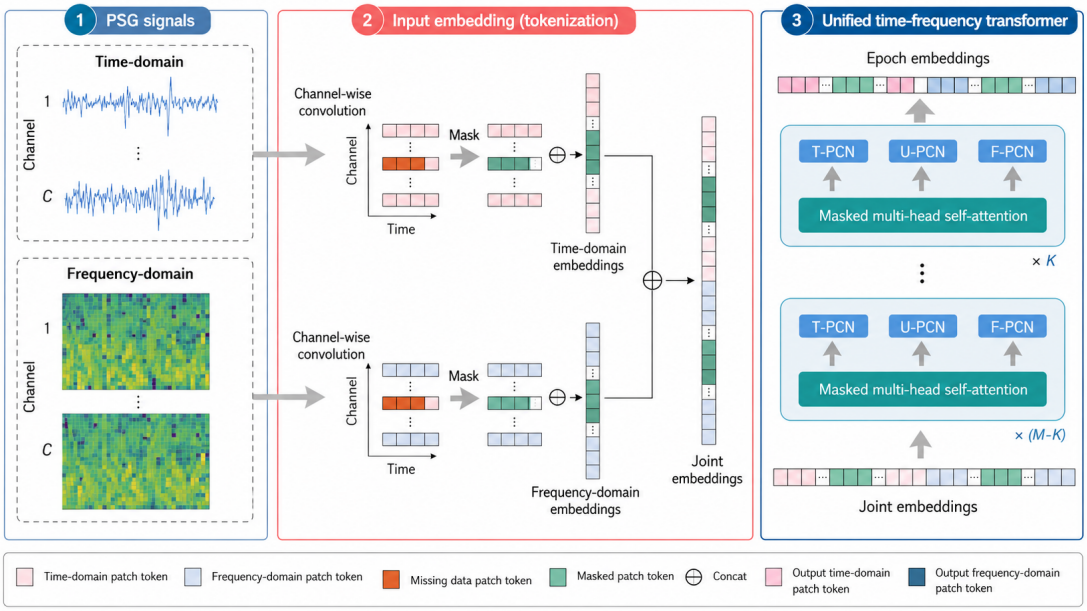

Figure 3.2: Unified time-frequency processing in SleepGPT. Temporal and frequency-domain PSG signals are tokenized separately and processed within a shared Transformer with domain-specific and unified processing layers. Adapted from [6].

The approach in this thesis differs from SleepGPT in that the temporal representation is inherited from SleepFM, whereas the frequency stream is learned specifically for spindle detection. The objective is not to pretrain a new unified backbone, but rather

to augment an existing temporal representation with task-specific spectral evidence.

3.4.1 Fusion of Pretrained and Task-Specific Representations

A straightforward approach involves concatenating temporal and frequency features prior to prediction. This method preserves information from both streams and serves as a useful baseline, but it does not explicitly model correspondences between temporal and frequency features.

Cross-attention enables context-dependent interactions between two representation streams. One stream provides the queries, while the other supplies the keys and values, allowing each position to retrieve complementary information according to its current context. The resulting sequence preserves the resolution of the querying stream while incorporating relevant information from the other representation space.

MulT [39] applies this directional principle to learn interactions between sequences without enforcing a predefined one-to-one alignment. This is particularly relevant when temporal and frequency features differ in tokenization or effective receptive field, since attention weights determine which spectral positions contribute to each temporal feature rather than assuming that identical indices contain equivalent information.

The strength of the interaction also matters when one branch is already pretrained. Flamingo [40] introduces new information via gated cross-attention while preserving a pretrained backbone. The gating mechanism regulates the magnitude of the added representation, which can limit disruption of the pretrained features during adaptation.

3.4.2 Temporal Correspondence and Latent Alignment

Feature interaction does not ensure compatibility between independently produced embeddings. Temporal and frequency streams may differ in scale, statistical distribution, or effective temporal resolution, even when derived from the same EEG signal.

Correlation-based objectives address this mismatch by encouraging both encoders to retain shared information without requiring their features to be numerically identical. Deep Canonical Time Warping extends this approach by learning nonlinear projections that are jointly correlated and temporally aligned [41]. Hadji et al. [42] further demonstrate that differentiable global alignment and cycle-consistency can constrain the ordering and coherence of correspondences across entire sequences.

Drop-DTW [43] further enables the exclusion of positions without reliable counterparts from the alignment. This approach prevents the forced matching of every feature in one stream to information in the other.

Another alternative proposed in previous work is to align representations probabilistically rather than enforce pointwise equality of features. Features have been mapped to latent distributions and compared using information-theoretic divergences, such as KL divergence, enabling a teacherstudent formulation in which a fixed representation guides another branch without requiring identical embeddings [44, 45]. Related variational distillation approaches transfer information in a similar manner via probabilistic latent representations [46].

3.5 Robustness and Cross-Dataset Generalization

Despite advancements in dense deep learning detectors for spindle localization, achieving robustness across diverse datasets remains a significant challenge. Variability in EEG acquisition conditions, population demographics and annotation protocols can impact detector performance. Consequently, a detector that demonstrates strong results on a specific benchmark may not generalize effectively to other cohorts or reliably reproduce spindle statistics beyond its training distribution.

Public benchmarks reveal multiple facets of this issue. The MASS dataset is an open resource comprising whole-night polysomnography recordings collected across various research protocols and sleep laboratories [26]. The MASS-SS2 subset includes recordings from 19 young, healthy participants, accompanied by expert annotations of sleep spindles and K-complexes [47]. While one expert annotated all recordings, spindle annotations from a second expert are available for only 15 subjects. However, its limited sample size and dependence on specific scorers constrain the generalizability of findings to wider clinical populations.

Massive Online Data Annotation addresses annotation reliability by deriving spindle labels based on the consensus of multiple scorers [17]. This dataset comprises 5342 consensus spindles from 180 participants and includes both younger and older cohorts. However, the EEG signals in MODA were sampled from N2 stage segments of the MASS archive, rather than being collected independently. Therefore, comparisons between MASS-SS2 and MODA primarily assess sensitivity to population composition and annotation methodology, rather than evaluating transferability across different acquisition centers or recording systems.

Recent detection methods increasingly assess robustness beyond a single training split. SEED [18] employs pretraining to reduce reliance on expert annotations and investigates transferability to new annotation settings. BLAST [20] evaluates performance across multiple datasets and compares predicted spindle characteristics with those derived by experts. OpenSpindleNet [48] extends this analysis to transitions between scalp and intracranial EEG.

Robust evaluation should specify whether generalization is assessed across held-out subjects, across annotation protocols within the same archive, or across independently acquired datasets. While event-level performance is essential, it should be supplemented by analyses of whether spindle density, duration and other derived characteristics remain consistent across different recording contexts.

3.6 Research Gap and Thesis Positioning

Specialized spindle detectors have demonstrated that dense temporal prediction is an effective approach for event localization. Models such as SpindleU-Net [19], SEED [18] and BLAST [20] learn representations directly from spindle or sleep-event annotations and generate temporally resolved predictions. While their task-specific design supports accurate localization, it remains unclear to what extent representations learned from

large-scale unlabeled PSG data can be effectively reused for this purpose.

Sleep foundation models approach representation learning from a distinct perspective. For example, SleepFM [5] learns transferable multimodal PSG representations, but its downstream evaluations primarily operate at temporal resolutions coarser than spindle boundaries. SleepMaMi [38] explicitly separates local physiological information from broader sleep context, but it has not been assessed for spindle localization. SleepGPT [6] offers the closest precedent by integrating time-frequency pretraining with a spindle detection task. Although SleepGPT demonstrates that foundation-model representations can support fine-grained sleep-event prediction, its unified pretraining architecture, time-frequency interaction and task-specific head are introduced simultaneously. As a result, the individual contributions of these components to dense localization remain unclear.

The central research gap is therefore not whether a sleep foundation model can detect spindles, but rather how an existing pretrained sleep representation should be adapted for this task. SleepGPT couples time-frequency pretraining, cross-domain interaction and a task-specific head in a single architecture, preventing the contribution of each component to dense localization from being isolated from its results. SleepFM instead provides a pretrained temporal representation without an explicit frequency branch or spindle-specific head. It was released well before SleepGPT and served as the backbone from the start of this work. This makes it a starting point where each component can be added and evaluated on its own.

This thesis therefore employs SleepFM as a frozen temporal backbone and adapts its token-level representation for dense spindle prediction. A task-specific frequency branch is incorporated to explicitly capture local spectral structure and a temporal decoder reconstructs the resolution necessary for event localization. The study compares increasingly structured fusion mechanisms and assesses whether adaptive interaction enhances performance beyond the pretrained temporal representation alone and whether auxiliary alignment facilitates or hinders the integration of complementary information.

Chapter 4

Methodology

4.1 Problem Formulation and Notation

Let

$$X \in \mathbb{R}^{C \times T} \tag{4.1}$$

represent an EEG recording comprising C channels and T temporal samples. This recording is partitioned into shorter segments,

$$X_k \in \mathbb{R}^{C \times T_0}, \qquad T_0 < T \tag{4.2}$$

where T_0 is the number of samples in each input segment. For each position $t \in \{1, \ldots, T_0\}$, the ground-truth spindle mask is defined as

$$y_t \in \{0, 1\} \tag{4.3}$$

where $y_t = 1$ designates that sample t belongs to an annotated spindle, whereas $y_t = 0$ denotes background EEG. Each positive sample is also assigned a boundary target,

$$\mathbf{b}_t = \left(\Delta^t_{\text{on}},\ \Delta^t_{\text{off}}\right) \tag{4.4}$$

which encodes the position of the sample relative to the onset t_{on} and offset t_{off} of the corresponding spindle event,

$$\Delta^t_{\text{on}} = t - t_{\text{on}}, \qquad \Delta^t_{\text{off}} = t_{\text{off}} - t \tag{4.5}$$

Consequently, boundary supervision is restricted to the set

$$\Omega^+ = \{t \mid y_t = 1\} \tag{4.6}$$

A temporal class target

$$c_t \in \{0, 1, 2\} \tag{4.7}$$

further distinguishes among background, spindle-boundary and spindle-interior samples. A positive sample is assigned to the boundary class when $\Delta^t_{\text{on}} \leq 16$ or $\Delta^t_{\text{off}} \leq 16$,

corresponding to a distance of 0.125 s from the annotated onset or offset at $f_s = 128$ Hz, and to the interior class otherwise. Given an EEG segment X_k, the model predicts

$$f_\theta(X_k) = \left\{ \hat{y}_t, \hat{\mathbf{b}}_t, \hat{\mathbf{c}}_t \right\}_{t=1}^{T_0} \tag{4.8}$$

where θ represents the trainable parameters. The output $\hat{y}_t \in [0, 1]$ specifies the probability of spindle presence at sample t, while

$$\hat{\mathbf{b}}_t = \left(\hat{\Delta}_{\text{on}}^t, \hat{\Delta}_{\text{off}}^t \right) \in \mathbb{R}^2 \tag{4.9}$$

yields estimates of the corresponding boundary offsets. The output

$$\hat{\mathbf{c}}_t \in [0, 1]^3 \tag{4.10}$$

contains the predicted probabilities associated with each of the three temporal classes. The model therefore generates sample-level predictions for spindle presence, event boundaries and temporal structure.

4.2 Proposed Architecture Overview

The proposed detector poses sleep spindle detection as dense temporal prediction. Given a multichannel EEG segment, it estimates the spindle probability, boundary offsets and temporal class defined in Section 4.1 at every sample.

Figure 4.1 summarizes the proposed architecture.

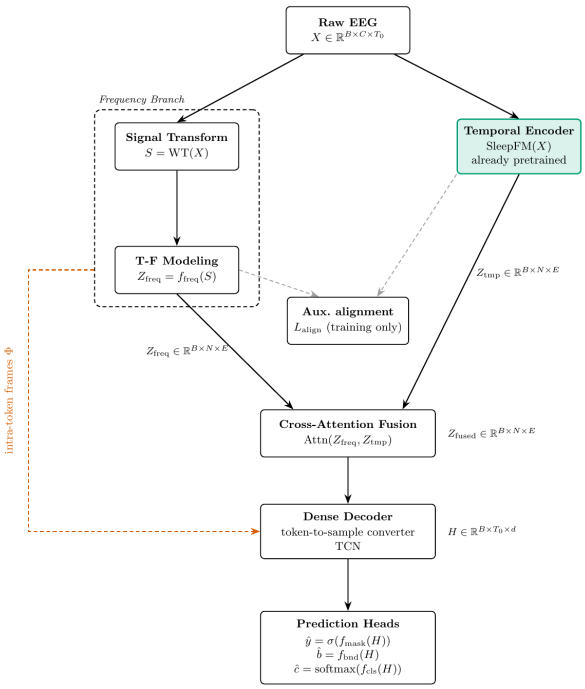

Figure 4.1: Overview of the proposed detector.

4.2.1 Overall Processing Pipeline

Each EEG segment is processed by two parallel branches sharing a common token grid. The frozen temporal encoder of Section 4.3 produces Z_{tmp} and contributes contextual information about the surrounding signal. The frequency branch of Section 4.4 extracts Z_{freq} from a Morlet scalogram and supplies the localized spectral evidence that the token discards. Its structured readout retains the T_s intra-token frames Φ, so sub-token temporal structure is preserved rather than pooled away.

The two streams are integrated by the fusion module of Section 4.5, in which the frequency stream queries the temporal one. The fused sequence Z_{fused} is returned to the sampling grid by the dense decoder of Section 4.6, which reinjects the retained frames Φ by frame-aware skip fusion. The reconstructed representation is then provided to the prediction heads of Section 4.7.

This organization differs from both lines of work reviewed in Chapter 3. Encoder-decoder detectors such as BLAST learn the entire representation from annotations and transfer their own encoder activations through skip connections. Here the encoder is frozen and general-purpose and what is transferred is the intra-token structure of a branch trained for the task. SleepGPT couples pretraining, time-frequency interaction and the detection head within a unified backbone. Keeping the temporal branch frozen while the frequency branch is trained separates these contributions, so each component can be substituted independently in the ablations of Chapter 5.

4.3 Temporal Encoder

The temporal branch of the proposed detector utilizes an adapted version of SleepFM, the sleep foundation model reviewed in Section 3.3. Instead of the published checkpoint, this approach employs an encoder pretrained exclusively on EEG signals, with a token duration shorter than in the original formulation. After pretraining, the encoder parameters are frozen and only the task-specific modules are optimized using spindle annotations.

4.3.1 EEG-Only Pretraining

The published SleepFM design learns multimodal representations using leave-one-out contrastive learning, aligning the representation of one physiological modality with information aggregated from the others. In the current setting, where only EEG is available, the required multimodal structure is reconstructed by decomposing the EEG channels into regional groups.

The available EEG channels are organized into frontal, central and posterior groups, which serve as three complementary views of the same EEG interval. This partition follows the topographic organization of spindle activity described in Section 2.1.4, where slow spindles predominate over frontal regions and fast spindles show a stronger central distribution.

Writing z_m for the representation of regional group m within a temporal window and z_{-m} for the aggregate of the remaining groups, the pretraining objective over the set $M = \{\text{frontal}, \text{central}, \text{posterior}\}$ is

$$L_{\text{LOO}} = \frac{1}{|M|} \sum_{m \in M} \ell_{\text{NCE}}(z_m, z_{-m}) \tag{4.11}$$

where ℓ_{NCE} denotes the symmetric InfoNCE loss used during pretraining. The encoder parameters are shared across the three regional groups, so the decomposition determines which representations are compared without requiring region-specific encoders. The detailed similarity construction, temperature scaling and bidirectional contrastive formulation are provided in Appendix A.

EEG montages vary across pretraining cohorts and the correspondence between recorded derivations and regional groups is defined separately for each cohort to ensure that each recording contributes all three views. The same regional organization is maintained when integrating the pretrained encoder into the spindle detector, with each input chunk represented by the available frontal, central and posterior activity.

4.3.2 Input Tokenization

Before tokenization, the pretraining signals were resampled to $f_s = 128\,\text{Hz}$ and band-pass filtered. The adapted encoder produces token embeddings of dimension $E = 128$.

Instead of operating at sample resolution, the tokenizer maps a window of W consecutive samples to a single temporal token using a convolutional encoder. Its six convolutional layers each apply a stride of two, contracting the input by a factor of $2^6 = 64$ before pooling into one embedding. The value retained here is

$$W = 384 \text{ samples} = 3 \text{ s} \tag{4.12}$$

which replaces the five-second tokens used in [5]. A shorter window reduces the amount of background EEG summarized within a single token. Like E and f_s, the window length W is fixed when the encoder is pretrained and therefore constrains every downstream use of the representation.

4.3.3 Token Grid and Temporal Embedding Extraction

The pretrained encoder defines the duration of a token but not the positions at which tokens are placed along a recording. Extracting a temporal representation from an EEG chunk $X_k \in \mathbb{R}^{C \times T_0}$ therefore requires a scanning grid, whose stride S_{tok} is a hyperparameter of the detector rather than a property of the pretrained model. Successive windows are placed at

$$a_n = nS_{\text{tok}}, \qquad n = 0, \ldots, \left\lfloor \frac{T_0 - W}{S_{\text{tok}}} \right\rfloor \tag{4.13}$$

so that token n covers the interval $[a_n, a_n + W)$. Choosing $S_{\text{tok}} < W$ produces overlapping windows, so a given sample contributes to several tokens and the effective density of the

token sequence increases without modifying the encoder. Figure 4.2 illustrates the grid obtained for $S_{\text{tok}} = W/2$.

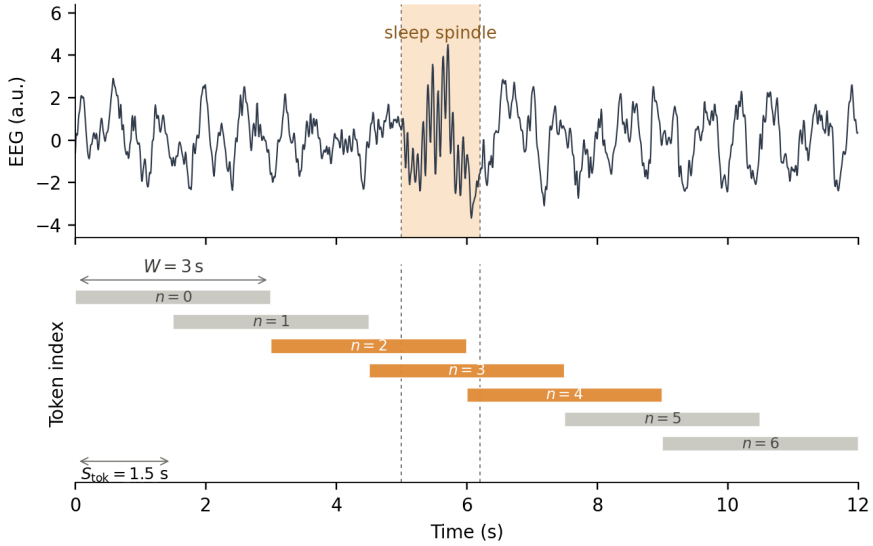

Figure 4.2: Token grid of the pretrained temporal encoder. Windows of length W are placed every S_{tok} samples, here with $S_{\text{tok}} = W/2$ as an illustration.

When $T_0 - W$ is not a multiple of S_{tok}, a final right-aligned window starting at $T_0 - W$ is appended so that every sample remains associated with at least one temporal window. Window positions are retained alongside the token embeddings and are used to reconstruct token-level information on the original sample axis. The exact token-count expression is provided in Appendix A.

Following channel pooling and temporal contextualization, the pretrained encoder generates both a pooled representation of the complete input and a temporally ordered sequence of token embeddings. Only the token-level representation

$$Z_{\text{tmp}} \in \mathbb{R}^{B \times N \times E} \tag{4.14}$$

is retained, where B denotes the batch size and N the number of temporal tokens associated with each input chunk.

Because each token summarizes a window of duration W, comparable to the longest spindles, Z_{tmp} describes the signal at a coarser resolution than localization requires. Reducing S_{tok} refines the grid on which the position of an event is expressed without changing the extent of signal summarized by any individual token, so precise onset and offset may not be reliably recovered from Z_{tmp} alone.

The encoder remains frozen during detector training. Excluding its parameters from gradient-based optimization preserves the representations learned from the larger pretraining dataset.

4.4 Frequency Branch

The pretrained temporal encoder compresses each EEG window into a single contextual token, preserving information about the surrounding signal. However, this approach does not explicitly capture the localized spectral activity associated with sleep spindles. The frequency branch addresses this limitation by extracting time-frequency patterns from the EEG and retaining their approximate position within each token, thereby complementing Z_{tmp}.

For an EEG chunk $X_k \in \mathbb{R}^{C\times T_0}$, where C denotes the number of EEG channels as defined in Section 4.1, the frequency branch computes a fixed Morlet scalogram across the entire chunk. Token-aligned patches are then extracted from this representation and processed by a convolutional encoder, producing $Z_{\text{freq}} \in \mathbb{R}^{B\times N\times E}$. The embedding dimension $E = 128$ matches that of the pretrained encoder, ensuring that Z_{freq} and Z_{tmp} share both the token grid and feature dimension.

4.4.1 Morlet Scalogram

The time-frequency representation is computed using the complex Morlet wavelets introduced in Section 2.2.3, with n_c controlling the trade-off between temporal and frequency resolution.

The wavelet family contains $F = 40$ kernels uniformly distributed between $f_{\min} = 1$ Hz and $f_{\max} = 40$ Hz,

$$f_j = f_{\min} + (j-1)\Delta f, \qquad \Delta f = \frac{f_{\max} - f_{\min}}{F-1}, \qquad j = 1, \ldots, F \tag{4.15}$$

All kernels share a common support determined by the lowest analyzed frequency. The Gaussian envelope of each wavelet is truncated at $\kappa = 3$ standard deviations, resulting in

$$w = 2\left\lceil \kappa \sigma_{f_{\min}} f_s \right\rceil + 1 \tag{4.16}$$

where $f_s = 128\,\text{Hz}$.

Let ψ_j denote the sampled complex kernel associated with frequency f_j. The kernel is centered and normalized to unit energy,

$$\widetilde{\psi}_j = \frac{\psi_j - \overline{\psi_j}}{\left\|\psi_j - \overline{\psi_j}\right\|_2} \tag{4.17}$$

where $\overline{\psi_j}$ denotes its sample mean. The kernels remain fixed throughout detector training.

Each EEG channel is convolved with the real and imaginary components of the wavelets using a frame stride H_{sc} and symmetric zero padding $p = (w-1)/2$. Frame k is centered on sample kH_{sc} and the scalogram coefficient is

$$S[c,k,j] = \frac{1}{2}\log\left(\left|\left(x_c * \widetilde{\psi}_j\right)[kH_{\text{sc}}]\right|^2 + \varepsilon\right) \tag{4.18}$$

where ε prevents numerical instability when coefficients approach zero. The modulus removes phase information, while logarithmic compression reduces the dynamic range.

For a chunk containing T_0 samples, the number of scalogram frames is

$$K = \left\lfloor \frac{T_0 + 2p - w}{H_{\text{sc}}} \right\rfloor + 1 \tag{4.19}$$

and the complete representation satisfies $S \in \mathbb{R}^{C \times K \times F}$, with temporal spacing H_{sc}/f_s between consecutive frames.

4.4.2 Token-Aligned Spectral Patches

The scalogram is computed once for the entire EEG chunk and then divided according to the windows defined by the SleepFM tokenizer. This avoids recomputing the transform for each token and preserves a consistent time-frequency grid across the chunk.

Let a_n denote the starting sample of token n. Since each token spans W samples, its spectral patch begins at frame a_n/H_{sc} and contains

$$L = 1 + \left\lfloor \frac{W - 1}{H_{\text{sc}}} \right\rfloor \tag{4.20}$$

frames, indexed from a_n/H_{sc} to $a_n/H_{\text{sc}} + L - 1$.

Exact alignment requires every token start to coincide with the scalogram grid. Denoting the token stride by S_{tok}, this condition is

$$S_{\text{tok}} \equiv 0 \pmod{H_{\text{sc}}}, \qquad T_0 - W \equiv 0 \pmod{H_{\text{sc}}} \tag{4.21}$$

where the first condition aligns the regular tokens and the second the additional end-anchored token introduced in Section 4.3.3. No rounding or interpolation is required when both conditions are satisfied.

Stacking the extracted patches over the batch and token dimensions gives

$$P \in \mathbb{R}^{B \times N \times C \times L \times F} \tag{4.22}$$

where each patch $P_{b,n}$ contains the spectral activity corresponding to one SleepFM token. At chunk boundaries, zero padding may attenuate coefficients whose wavelet support extends beyond the recorded signal. This effect is limited to the first and last patches.

4.4.3 Proposed Frequency Encoder

Before applying the learned encoder, the token dimension is folded into the batch, giving $P \in \mathbb{R}^{BN \times C \times L \times F}$. Each token is processed as a time-frequency map, with input channels corresponding to the EEG channels. The encoder first recalibrates the frequency bins, then extracts localized patterns using two-dimensional convolutions and finally summarizes the feature map through a structured readout.

Frequency-Wise Recalibration

A squeeze-and-excitation mechanism [49] assigns a data-dependent weight to each analyzed frequency. For one token, the input patch is averaged over the EEG channels and temporal frames,

$$r_{\text{freq}_j} = \frac{1}{CL} \sum_{c=1}^{C} \sum_{k=1}^{L} P[c, k, j], \qquad r_{\text{freq}} \in \mathbb{R}^F \tag{4.23}$$

A two-layer projection with reduction ratio 4 transforms this summary into a frequency gate,

$$a = \sigma \left(W_2 \phi \left(W_1 r_{\text{freq}} + b_1 \right) + b_2 \right) \tag{4.24}$$

with $W_1 \in \mathbb{R}^{(F/4) \times F}$ and $W_2 \in \mathbb{R}^{F \times (F/4)}$. The function ϕ denotes the GELU activation [50], while σ maps the output to $a \in (0, 1)^F$. The recalibrated patch is

$$\widetilde{P}[c, k, j] = a_j P[c, k, j]. \tag{4.25}$$

The same frequency weights are applied to all EEG channels before they are combined by the convolutional encoder.

Convolutional Feature Extraction

The recalibrated patch is processed by three two-dimensional convolutional layers following the formulation described in Section 3.2.1, with channel widths $C \to 32 \to 64 \to M$, where $M = 64$,

$$U^{(l)} = \phi \left(\text{Conv}_{3\times3}^{(l)} \left(U^{(l-1)} \right) \right), \qquad l = 1, 2, 3, \qquad U^{(0)} = \widetilde{P} \tag{4.26}$$

Each layer uses a 3×3 kernel, unit stride, unit zero padding and a GELU activation. The temporal and frequency dimensions are preserved throughout the stack, producing $U = U^{(3)} \in \mathbb{R}^{M \times L \times F}$.

The first layer combines the input EEG channels, after which the representation is no longer channel-resolved. The resulting feature map provides one shared spectral representation per token, consistent with the channel-pooled temporal representation Z_{tmp}, while the two-dimensional kernels capture patterns localized jointly in time and frequency.

Band-Structured Readout

Directly flattening U would make the projection size dependent on the complete time-frequency grid, while convolution alone does not explicitly preserve the absolute frequency position of a detected pattern. The readout therefore groups the feature map into fixed physiological bands before constructing the token descriptor.

Partitioning the frequency axis into physiological bands is motivated by [51, 52]. The partition used here isolates the sigma range and contains five disjoint intervals: low frequency $[1, 8)$ Hz, alpha $[8, 11)$ Hz, sigma $[11, 16]$ Hz, beta $(16, 30]$ Hz and high frequency $(30, 40]$ Hz. Let F_b denote the frequency indices associated with band b.

The temporal axis is divided into T_s ordered segments of equal length, which requires T_s to divide L. Let I_s denote the frame indices assigned to segment s,

$$I_s = \left\{ k \;\middle|\; \frac{sL}{T_s} \le k < \frac{(s+1)L}{T_s} \right\}, \qquad s = 0, \dots, T_s - 1 \tag{4.27}$$

so each segment contains L/T_s frames. Retaining the segment index localizes spectral activity to a fraction $1/T_s$ of the token rather than to the token as a whole.

For each convolutional channel, frequency band and temporal segment, the readout computes the mean activation,

$$\mu[m, b, s] = \frac{1}{|I_s||F_b|} \sum_{k \in I_s} \sum_{j \in F_b} U[m, k, j]. \tag{4.28}$$

Concatenating these statistics over the $N_b = 5$ frequency bands, the $M = 64$ convolutional channels and the T_s temporal segments produces

$$v \in \mathbb{R}^D, \qquad D = N_b M T_s = 320 T_s. \tag{4.29}$$

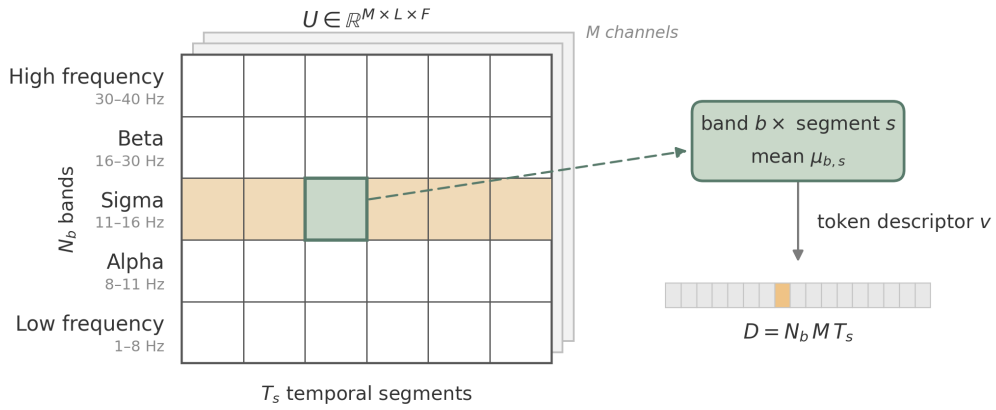

Figure 4.3: Band-structured readout. Each tile formed by a frequency band and a temporal segment is summarized by its mean and concatenated across all tiles and channels into the descriptor v.

The descriptor preserves coarse frequency organization and intra-token temporal position.

Frequency Embedding Projection

The descriptor is projected to the dimension of the pretrained temporal representation,

$$z_{\text{freq}} = \text{LN}\left(W_p v + b_p\right), \qquad W_p \in \mathbb{R}^{E\times D} \tag{4.30}$$

where LN denotes layer normalization [53]. Restoring the batch and token dimensions gives

$$Z_{\text{freq}} \in \mathbb{R}^{B\times N\times E}. \tag{4.31}$$

The frequency sequence thus shares the token index and embedding dimension of Z_{tmp}, allowing both representations to be combined directly by the fusion module.

4.5 Time-Frequency Fusion

The EEG signal is represented using two complementary approaches. The temporal branch models the progression of the waveform and its contextual information, while the frequency branch focuses on localized spectral features. Both representations are constructed on a shared token grid, with $Z_{\text{tmp}}, Z_{\text{freq}} \in \mathbb{R}^{B\times N\times E}$.

Consequently, tokens with the same index correspond to the same EEG interval. However, this temporal alignment does not ensure that the two embeddings exhibit identical latent structures, since each branch extracts distinct signal properties. The fusion module enables one representation to attend to the other and an auxiliary objective is introduced to promote consistency between corresponding tokens.

4.5.1 Cross-Attention Fusion

Although the temporal and frequency representations are defined on the same token grid, they encode different properties of the EEG. Accordingly, the fusion module uses the frequency stream as the query sequence and the temporal stream as the source of contextual information. Each frequency token can thus retrieve the temporal features most relevant to its spectral content.

The query, key and value sequences are obtained through independent linear projections,

$$Q_f = Z_{\text{freq}} W_Q^\top, \qquad K_t = Z_{\text{tmp}} W_K^\top, \qquad V_t = Z_{\text{tmp}} W_V^\top \tag{4.32}$$

where $W_Q, W_K, W_V \in \mathbb{R}^{E\times E}$.

A fixed sinusoidal positional encoding is added to the queries and keys prior to computing attention scores. This encoding preserves the temporal order of the token sequence and enables the similarity function to distinguish identical features at different positions.

The projected sequences are divided into $n_h = 8$ heads of width $E/n_h = 16$, as illustrated in Figure 4.4. For head i, scaled dot-product attention is defined as

$$\text{head}_i = \text{softmax}\left(\frac{Q_f^{(i)}\left(K_t^{(i)}\right)^\top}{\sqrt{E/n_h}}\right)V_t^{(i)} \tag{4.33}$$

The attention weights determine the contribution of the temporal stream to each frequency token and therefore allow this contribution to vary across positions [36].

The outputs of the n_h heads are concatenated and projected back to width E,

$$A = \text{Concat}\left(\text{head}_1, \ldots, \text{head}_{n_h}\right) W_O^\top, \qquad W_O \in \mathbb{R}^{E\times E} \tag{4.34}$$

The attended representation is then added through a residual connection with dropout [54],

$$Z_{\text{attn}} = \text{LN}\left(Z_{\text{freq}} + \text{Dropout}(A)\right) \tag{4.35}$$

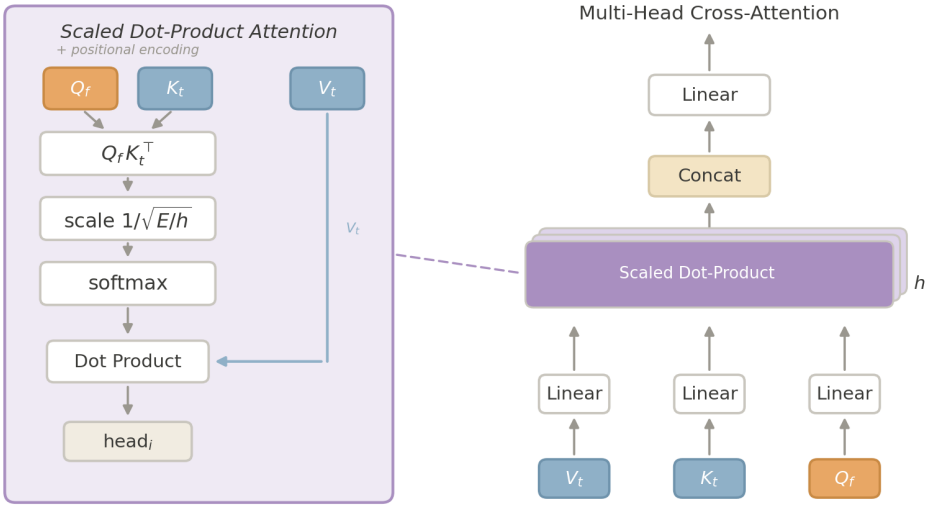

Figure 4.4: Cross-attention fusion. The frequency stream provides the queries and the temporal stream the keys and values, over n_h heads.

The residual path maintains the frequency representation, while the cross-attention mechanism introduces temporal context where it is useful. A position-wise feed-forward network further transforms the resulting features,

$$Z_{\text{fused}} = \text{LN}\left(Z_{\text{attn}} + \text{Dropout}\left(\text{FFN}\left(Z_{\text{attn}}\right)\right)\right) \tag{4.36}$$

where the inner width of the feed-forward network is $4E$. As all operations preserve the token dimension, $Z_{\text{fused}} \in \mathbb{R}^{B\times N\times E}$ remains defined on the original grid.

4.5.2 Auxiliary Alignment

The two branches characterize the same interval from complementary views and the cross-attention of Equation 4.33 lets the frequency stream draw on the temporal one. This exchange may benefit from consistency between the two representations, a property not enforced by attention alone. Enforcing identical sequences would make the frequency branch redundant, so the auxiliary objective operates on distributions rather than individual vectors and learns a variance for each latent dimension.

A shared projection head maps both sequences to diagonal Gaussian distributions [44],

$$\left(\mu_q, \log \sigma_q^2\right) = h_{\text{align}}\left(Z_q\right), \qquad q \in \{\text{tmp}, \text{freq}\} \tag{4.37}$$

Using the same projection head for both branches establishes a shared comparison space. Writing $\Omega_{\text{tok}} = \{1, \ldots, B\} \times \{1, \ldots, N\}$ for the token indices of a batch, the consistency loss is defined as

$$L_{\text{cons}} = \frac{1}{|\Omega_{\text{tok}}|} \sum_{(b,n)\in\Omega_{\text{tok}}} D_{\text{KL}}\left(q_{\text{tmp}}^{(b,n)} \| q_{\text{freq}}^{(b,n)}\right) \tag{4.38}$$

For diagonal Gaussians, this divergence with $d_{\text{align}} = 64$ has the closed form

$$D_{\text{KL}}\left(q_{\text{tmp}} \| q_{\text{freq}}\right) = \frac{1}{2} \sum_{i=1}^{d_{\text{align}}} \left[\log \frac{\sigma_{\text{freq},i}^2}{\sigma_{\text{tmp},i}^2} + \frac{\sigma_{\text{tmp},i}^2 + \left(\mu_{\text{tmp},i} - \mu_{\text{freq},i}\right)^2}{\sigma_{\text{freq},i}^2} - 1\right] \tag{4.39}$$

where the log-variances are bounded to $[-6, 2]$ for numerical stability. Placing the temporal stream first leverages the asymmetry of the divergence, penalizing a frequency distribution that is narrower than its reference.

The Gaussian parameters of the temporal stream are detached after the head, so the temporal stream acts as a fixed teacher and the shared head is trained only through the frequency branch. This teacher-student arrangement [45] employs a probabilistic feature representation related to variational distillation [46].

4.6 Dense Decoder

The fused representation $Z_{\text{fused}} \in \mathbb{R}^{B\times N\times E}$ is defined on the token grid, while the prediction targets are specified at the original sampling resolution. The decoder therefore maps it to a sample-level representation $H \in \mathbb{R}^{B\times T_0\times d}$, with $d = 64$.

This reconstruction integrates wavelet-based token synthesis, intra-token frequency features preserved by the structured readout and residual temporal refinement. The complete decoder is shown in Figure 4.5.

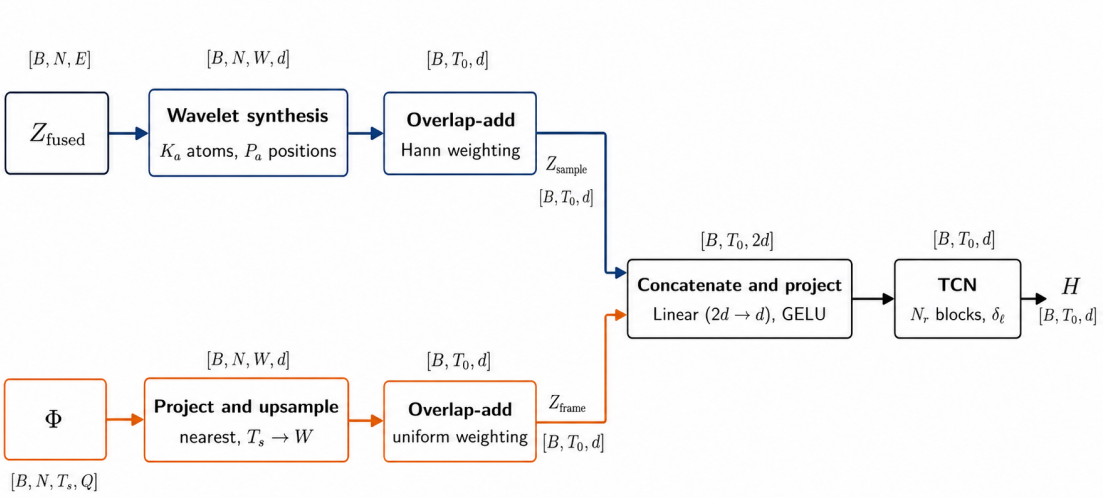

Figure 4.5: Dense decoder. The fused tokens are reconstructed at sample resolution by a localized wavelet synthesis, while the intra-token frames are expanded by nearest-neighbor upsampling. Both branches are recomposed by overlap-add, concatenated and refined by the temporal convolutional network.

4.6.1 Token-to-Sample Wavelet Construction

Each fused token is reconstructed over its original window of $W = 384$ samples using a fixed bank $\Psi \in \mathbb{R}^{K_a \times P_a \times W}$ of $K_a = 16$ Morlet atom types, each evaluated at $P_a = 6$ candidate positions within the window. The complete construction of the fixed atom bank is provided in Appendix B.

For each token $z_n = Z_{\text{fused}}^{(n)}$, two task-specific multilayer perceptrons predict the atom coefficients and position logits,

$$C_n = f_C(z_n) \in \mathbb{R}^{K_a \times d}, \qquad L_n = f_L(z_n) \in \mathbb{R}^{K_a \times P_a} \tag{4.40}$$

where both MLPs follow

$$f_q(z) = W_{q,2}\text{Dropout}\left(\phi\left(W_{q,1}\text{LN}(z) + b_{q,1}\right)\right) + b_{q,2}, \qquad q \in \{C, L\} \tag{4.41}$$

A softmax operation over the P_a positions yields a placement distribution for each atom. The localized form of atom k is

$$A_{n,k,w} = \sum_{m=1}^{P_a} \pi_{n,k,m} \Psi_{k,m,w} \tag{4.42}$$

The reconstructed feature vector at position w of token n combines a token-level base projection with the weighted atoms,

$$X_{n,w} = \mathrm{LN}\left(W_b z_n + b_b\right) + \sum_{k=1}^{K_a} A_{n,k,w} C_{n,k}, \qquad X_{n,w} \in \mathbb{R}^d \tag{4.43}$$

The atom bank remains fixed, while atom coefficients and position probabilities are predicted from each token. The reconstructed windows are aligned on the sample axis according to the token start positions a_n. Let $I_t = \{(n, w) \mid a_n + w = t\}$ denote the window positions contributing to sample t. Since the token windows overlap, each sample is covered by several of them. Following the standard overlap-add reconstruction [12], the windows are weighted by a Hann window g of length W,

$$Z_{\text{sample}}[t] = \frac{\sum_{(n,w)\in I_t} g_w X_{n,w}}{\sum_{(n,w)\in I_t} g_w}, \qquad Z_{\text{sample}}[t] \in \mathbb{R}^d \tag{4.44}$$

4.6.2 Frame-Aware Skip Fusion

The band-structured readout of Section 4.4.3 retains the mean activation along the temporal segment index s, providing each token with T_s intra-token frames,

$$\Phi \in \mathbb{R}^{B\times N\times T_s\times Q} \tag{4.45}$$

where each frame concatenates the mean across the N_b frequency bands and M convolutional channels, resulting in $Q = N_b M = 320$.

These segment-level features are projected to the decoder width,

$$\overline{\Phi} = \mathrm{LN}\left(\Phi W_f^\top + b_f\right) \in \mathbb{R}^{B\times N\times T_s\times d} \tag{4.46}$$

with $W_f \in \mathbb{R}^{d\times Q}$. For each token, nearest-neighbor upsampling expands the T_s segments to the W sample positions. The expanded windows are subsequently averaged on the original timeline,

$$Z_{\text{frame}}[t] = \frac{1}{|I_t|} \sum_{(n,w)\in I_t} \mathrm{Up}\left(\overline{\Phi}_n\right)_w \in \mathbb{R}^d \tag{4.47}$$

Nearest-neighbor expansion preserves each segment value without interpolation and no Hann weighting is applied because tapering would modulate these piecewise-constant features within the token.

$$Z_{\text{comb}} = \phi\left(\left[Z_{\text{sample}}\,; Z_{\text{frame}}\right] W_s^\top + b_s\right) \in \mathbb{R}^{B\times T_0\times d} \tag{4.48}$$

where $W_s \in \mathbb{R}^{d\times 2d}$.

4.6.3 Dense Temporal Refinement

The fused sample-level sequence is refined using a temporal convolutional network [55], consisting of $N_r = 5$ residual blocks of dilated one-dimensional convolutions. Each block applies two convolutions with kernel size 7 and dilation δ_ℓ,

$$h^{(\ell)} = \mathrm{LN}\left(h^{(\ell-1)} + \mathrm{Conv}_{7,\delta_\ell}\left(\mathrm{Dropout}\left(\phi\left(\mathrm{Conv}_{7,\delta_\ell}\left(h^{(\ell-1)}\right)\right)\right)\right)\right) \tag{4.49}$$

with $h^{(0)} = Z_{\mathrm{comb}}$. The dilation schedule enlarges the temporal receptive field without reducing sample resolution. No recurrent or attention block is applied. The decoder output is

$$H = h^{(N_r)} \in \mathbb{R}^{B\times T_0\times d} \tag{4.50}$$

4.7 Prediction Heads

The dense decoder produces the sample-level representation H defined in Equation 4.50, where $H_t \in \mathbb{R}^d$ describes sample t. A shared projection first maps this representation to features used by all prediction tasks,

$$G_t = \mathrm{Dropout}\left(\phi\left(W_g\mathrm{LN}(H_t) + b_g\right)\right) \tag{4.51}$$

where ϕ denotes the GELU activation. This shared projection provides common features to the three prediction tasks.

Three task-specific multilayer perceptrons (MLPs) [56] then estimate the spindle probability, boundary offsets and temporal class introduced in Section 4.1,

$$\hat{y}_t = \sigma\left(f_{\mathrm{mask}}(G_t)\right), \qquad \hat{\mathbf{b}}_t = f_{\mathrm{bnd}}(G_t), \qquad \hat{\mathbf{c}}_t = \mathrm{softmax}\left(f_{\mathrm{cls}}(G_t)\right) \tag{4.52}$$

Each MLP applies a linear layer, GELU activation and dropout [54], followed by a task-specific output layer. The module is applied independently at every sample. Its organization is shown in Figure 4.6.

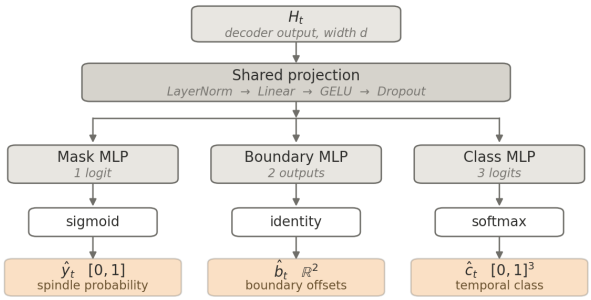

Figure 4.6: Organization of the prediction heads. A shared projection is applied to the decoder output, after which three task-specific perceptrons produce the spindle probability, the boundary offsets and the temporal class.

4.8 Training Objective

The detector is trained using three dense outputs, as detailed in Section 4.7, together with auxiliary constraints that promote representation consistency and regulate predicted spindle properties.

4.8.1 Dense Prediction Losses

All dense losses are computed across samples $\Omega = \{1, \ldots, T_0\}$ within each segment. Supervision of the spindle mask is accomplished by jointly applying focal and Dice losses.

$$L_{\text{seg}} = L_{\text{focal}} + L_{\text{dice}} \tag{4.53}$$

The focal loss reduces the impact of confidently classified samples and emphasizes positions that are difficult to distinguish [57].

$$L_{\text{focal}} = -\frac{1}{|\Omega|} \sum_{t \in \Omega} \alpha_t \, (1 - p_t)^{\gamma} \log p_t, \qquad p_t = \begin{cases} \hat{y}_t, & y_t = 1 \\ 1 - \hat{y}_t, & y_t = 0 \end{cases} \tag{4.54}$$

In this context, α_t balances positive and negative samples, while γ modulates the down-weighting of easily predicted samples. The Dice loss directly supervises the overlap between predicted and reference spindle regions.

$$L_{\text{dice}} = 1 - \frac{2 \sum_{t \in \Omega} \hat{y}_t y_t + \varepsilon}{\sum_{t \in \Omega} \hat{y}_t + \sum_{t \in \Omega} y_t + \varepsilon} \tag{4.55}$$

This loss is computed with probabilities rather than thresholded predictions, which reduces the dominance of background samples. As a result, the two losses provide complementary supervision at both the sample and region levels.

Boundary regression is restricted to the positive sample set Ω^+ defined by the ground truth. Predicted offsets are optimized using the smooth ℓ_1 loss [58]. The offsets are normalized by f_s, so that the residual is measured in seconds and the transition of ℓ_β occurs at an error of β seconds.

$$L_{\text{bnd}} = \frac{1}{2|\Omega^+|} \sum_{t \in \Omega^+} \sum_{u \in \{\text{on,off}\}} \ell_\beta \left(\frac{\hat{\Delta}_u^t - \Delta_u^t}{f_s} \right) \tag{4.56}$$

where

$$\ell_\beta(x) = \begin{cases} \frac{x^2}{2\beta} & |x| < \beta \\ |x| - \frac{1}{2}\beta & \text{otherwise} \end{cases} \tag{4.57}$$

The quadratic region enables precise boundary refinement, whereas the linear region constrains the influence of large offset errors.

The temporal class head is supervised using weighted cross-entropy,

$$L_{\text{cls}} = -\frac{1}{|\Omega|} \sum_{t \in \Omega} w_{c_t} \log \hat{c}_t^{(c_t)} \tag{4.58}$$

The class weights w_c are automatically computed from the training-set class distribution. Boundary and spindle-interior samples receive larger weights because they occur less frequently than background samples.

4.8.2 Auxiliary Objectives

The auxiliary objective includes the token consistency term described in Section 4.5.2, together with a frequency constraint imposed on predicted event candidates.

For an event e, the sigma prominence is defined as

$$\rho_e = \frac{E_{\sigma,e}}{E_{\sigma,e} + E_{\text{ctx},e}} \tag{4.59}$$

Here, $E_{\sigma,e}$ represents the energy in $[11, 16]$ Hz and $E_{\text{ctx},e}$ represents the energy in $[5, 11) \cup (16, 25]$ Hz. Events with prominence below the reference level are penalized using

$$L_{\text{freq}} = \frac{1}{|\Omega_{\text{evt}}|} \sum_{e \in \Omega_{\text{evt}}} \max\left(0, \rho_{\text{th}} - \rho_e\right) \tag{4.60}$$

with $\rho_{\text{th}} = 0.70$.[1] This ratio reduces sensitivity to amplitude variations across different recordings. When no predicted event candidates are present, i.e., $|\Omega_{\text{evt}}| = 0$, L_{freq} is set to zero.

The complete auxiliary term is

$$L_{\text{align}} = L_{\text{cons}} + L_{\text{freq}} \tag{4.61}$$

[1] Computed based on the mean prominence of the ground truth.

4.8.3 Combined Objective

The final training objective combines the dense prediction losses with the auxiliary constraints,

$$L = \lambda_{\text{seg}} L_{\text{seg}} + \lambda_{\text{bnd}} L_{\text{bnd}} + \lambda_{\text{cls}} L_{\text{cls}} + \lambda_{\text{align}} L_{\text{align}} \tag{4.62}$$

The coefficients set the contribution of each term to the gradient. Parameters of the auxiliary head are used exclusively during training.

Chapter 5

Experiments

5.1 Experimental Setup

5.1.1 Dataset and Annotation Protocol

The experiments used the MASS-SS2 dataset [26], re-annotated for sleep spindles by Olivier Pallanca. This annotation set, hereafter denoted OPA, is distinct from the publicly available MASS-SS2 annotations and had not previously been used to evaluate automatic detectors. The public MASS-SS2 annotations were not used for the primary experiments, which relied on the OPA annotations because their morphology and boundaries were more consistent with the spindle criteria adopted in this study (Section 2.1.4). They were nevertheless used later in the published protocol and annotation source comparison experiments. Three EEG channels were extracted for each participant: C3, F3 and O1, corresponding to the central, frontal and occipital regions defined in Section 2.1.3. All nineteen participants provide these three channels and the reference spindle annotations were defined on C3.

The OPA annotations contain 29573 spindle events scored across sleep stages N2 and N3. As they come from a single scorer, their boundaries remain subject to the uncertainty discussed in Section 2.3.1. The onset and offset times were converted into sample-aligned targets following Section 2.3.2.

5.1.2 Data Preparation and Subject-Wise Splits

Each EEG channel was read from the source recording at its original sampling rate and band-pass filtered between 0.3 and 40 Hz using a fourth-order zero-phase Butterworth filter. The filtered signal was then resampled to 128 Hz through polyphase filtering.

Each selected channel was standardized over the complete recording before the analyzed intervals were extracted as $\widetilde{x}_s[n] = (x_s[n] - \mu_s)/\sigma_s$, where μ_s and σ_s denote the mean and standard deviation of the corresponding derivation for participant s. After resampling, each annotated time t, expressed in seconds, was mapped to the 128 Hz grid by multiplying it by 128 and rounding to the nearest sample.

The recordings were partitioned into contiguous, non-overlapping 300-second chunks containing 38400 samples. A fixed subject-wise split was used throughout the primary OPA experiments, balancing spindle count, total spindle duration and available EEG duration across the training, validation and test subsets. All chunks from a participant remained in the same subset, preventing subject-level overlap. Table 5.1 summarizes the resulting distribution and the exact subject split is given in Appendix F.

Table 5.1: Distribution of the OPA-annotated MASS-SS2 data.

Subset	**Participants**	**Chunks**	**Duration (h)**	**N2**	**N3**	**Total**
Training	13	1264	105.3	16833	3143	19976
Validation	3	272	22.7	3595	667	4262
Test	3	332	27.7	4134	1201	5335
Total	19	1868	155.7	24562	5011	29573

5.1.3 Training

The SleepFM backbone remained frozen, while all task-specific modules were optimized jointly using the objective defined in Section 4.8. Hyperparameters were selected with Optuna [59] using 29 trials that maximized $F_1@0.2 + 0.5F_1@0.5$ on the validation subset. The best configuration was retained.

Table 5.2: Training configuration used across the experiments.

Parameter	**Value**	**Parameter**	**Value**
Optimizer	AdamW	Focal loss α	0.561
Training epochs	40	Focal loss γ	2.416
Learning rate	4×10^{-5}	Smooth ℓ_1 transition β	0.065
Weight decay	10^{-5}	λ_{seg}	0.755
Batch size	16	λ_{bnd}	1.817
Gradient clipping	1.0	λ_{cls}	1.0
Dropout	0.2	λ_{align}	0.462
Learning-rate reduction factor	0.527	Scheduler patience	3
S_{tok}	192 samples	Random seed	42

5.1.4 Inference and Evaluation Protocol

The sample-level spindle probabilities were thresholded to form contiguous candidate regions. Within each region, the predicted boundary offsets generated onset and offset proposals that were aggregated using probability-weighted medians. Boundary offsets

are expressed in samples during decoding, while their training residual is normalized by f_s. The complete decoding procedure is detailed in Appendix C.

The probability threshold was searched from 0.40 to 0.70 in increments of 0.01, while the merge gap was searched from 0 to 0.20 seconds in increments of 0.05. The complete grid was evaluated on the validation subset after each epoch and the checkpoint and operating point maximizing validation event-level F1@0.2 were retained. The selected configuration was then applied unchanged to the test subset, which was not involved in model or parameter selection. No duration-based filtering was applied.

Predicted and reference events were matched one-to-one using the greedy strategy described in [24]. A prediction and a reference event were matched when their IoU was greater than or equal to a fixed threshold. Precision, recall and F1 score were computed at IoU thresholds of 0.2, 0.3 and 0.5, following Section 2.4. True positives, false positives and false negatives were pooled over all test chunks before computing these metrics, so the reported F1 score is computed across individual spindle events rather than averaged per subject.

5.1.5 Implementation and Computational Environment

The models were implemented in Python using PyTorch 2.0.1 with CUDA 11.7 and executed on the Idiap computing cluster through Slurm. Most experiments used a single NVIDIA GeForce RTX 3090 GPU with 24 GB of memory, while NVIDIA Tesla V100S GPUs with 32 GB were used when required. Training was conducted in single-precision floating-point arithmetic without automatic mixed precision. A complete 40-epoch run of the final configuration required approximately ten hours on an RTX 3090, including validation decoding after each epoch and final evaluation.

5.2 Ablation Studies

The following experiments examine the contribution of the main design choices introduced in Chapter 4. Unless explicitly stated otherwise, each experiment modifies a single component while preserving the remaining architecture, training procedure, and decoding protocol. This controlled setting limits the influence of simultaneous changes and allows the observed differences to be associated with the component under investigation. All ablation results are reported on the validation subset, which is used for architectural analysis and model selection. The held-out test subset remains excluded from this process and is reserved for the final evaluation in Section 5.3 and the comparison with existing methods in Section 5.4.

5.2.1 Temporal Encoder

The proposed architecture uses a frozen SleepFM encoder for temporal modeling. To assess its contribution, it is compared with a Transformer, BiLSTM and TCN trained from scratch on the same token grid.

Table 5.3: Effect of the temporal encoder. The frozen pretrained SleepFM encoder is compared with three encoders trained from scratch.

F1 at IoU	**Temporal encoder**			
	SleepFM	Transformer	BiLSTM	TCN
0.2	0.7950	**0.7954**	0.7902	0.7869
0.3	0.7918	**0.7923**	0.7878	0.7846
0.5	0.7589	**0.7602**	0.7529	0.7541

A from-scratch Transformer achieves performance comparable to the frozen SleepFM encoder, while the recurrent and convolutional variants remain slightly below it. At this scale, pretraining provides no clear performance advantage over a temporal encoder trained from scratch. SleepFM is nevertheless retained because it served as the backbone from the beginning of this work, before SleepGPT was released and the architecture was developed around its representation. This experiment does not assess pretraining at the full capacity of SleepFM. It only shows that, in the present configuration, a smaller Transformer trained from scratch is sufficient to match it.

5.2.2 Frequency Representation

The frequency branch exposes the localized spectral features that the pretrained tokens discard. Its main architectural choices are evaluated through ablations of the signal transform, encoder input, convolutional encoder and structured readout.

Signal Transform

The Morlet scalogram of Section 4.4.1 is compared with the STFT of Section 2.2.2 and a learned SincNet filterbank [60].

Table 5.4: Effect of the signal transform.

F1 at IoU	**Transform**		
	Morlet	STFT	SincNet
0.2	**0.7950**	0.7904	0.7899
0.3	**0.7918**	0.7856	0.7858
0.5	**0.7589**	0.7464	0.7503

The Morlet representation achieves the highest F1 score at every IoU threshold. The difference becomes more apparent at IoU 0.5, where it exceeds the STFT and SincNet by 0.0125 and 0.0086, respectively. This advantage under stricter boundary

matching is consistent with the frequency-dependent temporal resolution of the Morlet representation.

Encoder Input

Three input modifications are compared with the unmodified scalogram. Frequency recalibration uses either the retained frequency-wise squeeze-and-excitation module (f-SE) or a finer time-frequency variant (TF-SE) [49]. A second family introduces A7-inspired physiological descriptors [22], either through a projected token or direct concatenation. The final variant appends fixed frequency-coordinate maps encoding absolute frequency position and the sigma range [61]. Their exact construction is reported in Appendix D.1.

Table 5.5: Effect of the encoder input.

F1 at IoU	**Recalibration**			**Physiological descriptors**		**Frequency coordinates**
	None	f-SE	TF-SE	Token	Direct	✓
0.2	0.7935	**0.7950**	0.7893	0.7897	0.7942	0.7932
0.3	0.7898	**0.7918**	0.7866	0.7860	0.7917	0.7907
0.5	0.7577	**0.7589**	0.7531	0.7527	0.7572	0.7578

Frequency-wise recalibration gives the highest scores, although its gain over the unmodified scalogram remains modest. TF-SE, handcrafted descriptors and fixed frequency coordinates provide no improvement. The simpler frequency-wise gate is therefore retained.

Frequency Encoder

Frame injection is disabled for the encoder and readout ablations because the alternative configurations do not produce the descriptor required by this connection. Their reference is therefore the retained model without frame injection, which reaches $F_1@0.2 = 0.7905$, rather than the complete-model score of 0.7950.

The retained convolutional encoder of Section 4.4.3 is compared with an alternative that divides the representation into three physiological bands and combines their features through attention [52].

Table 5.6: Effect of the frequency encoder. The reference is the retained model without frame injection.

F1 at IoU	**Encoder**	
	CNN	Attention on 3 bands
0.2	**0.7905**	0.6797
0.3	**0.7871**	0.6391
0.5	**0.7524**	0.5293

The three-band attention encoder substantially reduces performance, with a larger gap at stricter IoU thresholds. At IoU 0.5, the convolutional encoder reaches 0.7524 against 0.5293. This result supports preserving the complete frequency axis during encoding and introducing physiological band organization only at the readout.

Structured Readout

The structured readout of Section 4.4.3 is compared with flattening, global pooling and band pooling. Global pooling averages over both temporal and frequency axes, band pooling preserves coarse frequency organization while averaging over time and frame-wise band pooling additionally retains the T_s intra-token temporal segments defined in Equation 4.27.

Table 5.7: Effect of the readout. The reference is the retained model without frame injection.

F1 at IoU	**Readout**			
	Flatten	Global pooling	Band pooling	Frame-wise band pooling
0.2	0.7765	0.7727	0.7291	**0.7905**
0.3	0.7729	0.7690	0.7114	**0.7871**
0.5	0.7356	0.7274	0.6293	**0.7524**

Frame-wise band pooling achieves the highest performance at every IoU threshold. Its large advantage over band pooling shows that preserving frequency organization alone is insufficient when temporal variation within the token is averaged away. Flattening also remains below the structured readout, supporting an explicit organization of both frequency and intra-token position.

Parameter Sensitivity

The sensitivity of the retained frequency branch to H_{sc}, n_c, encoder depth and T_s was additionally evaluated. The retained values remained close to the best-performing con-

figurations across all IoU thresholds and no parameter exhibited a sharp optimum. The complete analysis is reported in Appendix D.2.

5.2.3 Auxiliary Alignment

The temporal and frequency branches share the same token grid but are produced by independently trained encoders and may therefore have incompatible latent structures. Jointly fine-tuning SleepFM could reduce this mismatch, but the temporal backbone is kept frozen to preserve its pretrained representation and avoid the cost of adapting the full encoder. Instead, Z_{tmp} and Z_{freq} are regularized during training by the auxiliary objective defined in Section 4.8, without modifying inference.

The retained Gaussian KL formulation is compared with four alternatives. In this comparison, the frequency constraint L_{freq} is kept unchanged across all configurations, so only the representation-consistency component is varied. TemporalAlign applies a residual two-layer temporal convolution, CCA encourages correlated projections [62], InfoNCE attracts corresponding tokens while separating non-corresponding pairs [63] and ℓ_2 matching directly minimizes their distance.

Table 5.8: Effect of the auxiliary objective.

F1 at IoU	**Alignment objective**				
	Gaussian KL	TemporalAlign	CCA	InfoNCE	ℓ_2 matching
0.2	**0.7950**	0.7775	0.7848	0.7847	0.7885
0.3	**0.7918**	0.7736	0.7824	0.7806	0.7836
0.5	**0.7589**	0.7369	0.7464	0.7467	0.7499

The Gaussian KL objective achieves the highest F1 score at every IoU threshold. TemporalAlign reduces F1 by 0.0175, 0.0182 and 0.0220, indicating that the additional temporal convolution is unnecessary once both branches share the same grid. CCA and InfoNCE also reduce performance, while direct ℓ_2 matching remains intermediate. These results suggest that alignment is most useful when it makes the representations compatible without forcing them to become identical.

5.2.4 Time-Frequency Fusion

The fusion module combines the temporal and frequency token representations before dense reconstruction. Its design is evaluated through controlled ablations in which only the fusion component under study is modified.

Fusion Mechanism

The retained mechanism is the residual cross-attention of Section 4.5.1, where Z_{freq} provides the queries and Z_{tmp} the keys and values. It is compared with concatena-

tion [64], gated cross-attention [40] and a selective variant combining head-specific positional bias [65] with a learned attention temperature.

Table 5.9: Effect of the fusion mechanism.

F1 at IoU	**Fusion mechanism**			
	Cross-attention	Concatenation	Gated	Selective
0.2	**0.7950**	0.7876	0.7877	0.7906
0.3	**0.7918**	0.7841	0.7844	0.7868
0.5	**0.7589**	0.7497	0.7535	0.7558

Residual cross-attention achieves the highest F1 score at every IoU threshold. Concatenation and gated attention remain below it, while the selective mechanism is the strongest alternative. Neither residual gating nor modification of the attention distribution improves over the standard cross-attention formulation.

Query Direction and Branch Contribution

The query direction determines which representation is preserved by the residual pathway. The retained configuration uses Z_{freq} as the query stream, while reversing the direction preserves Z_{tmp}. Frame injection is disabled throughout this experiment because otherwise the temporal-only configuration would still receive frequency information through Φ (Section 4.6.2). The comparison therefore isolates the fused token representation and the contribution of each branch.

Table 5.10: Effect of the query direction and individual branch contribution without frame injection.

F1 at IoU	**Query stream**		**Single branch**	
	Frequency	Temporal	Frequency only	Temporal only
0.2	**0.7905**	0.5978	0.7871	0.5533
0.3	**0.7871**	0.5613	0.7835	0.5195
0.5	**0.7524**	0.4528	0.7475	0.4196

Supplying Z_{freq} directly to the decoder reduces F1 by only 0.0034, 0.0036 and 0.0049 across the three IoU thresholds, showing that the frequency branch alone retains most of the information required for dense localization. In contrast, using Z_{tmp} alone reduces F_1@0.5 from 0.7524 to 0.4196. Reversing the query direction is only marginally better, reaching 0.4528 despite retaining access to frequency features as keys and values.

These results show that the frozen temporal representation is insufficiently localized when it forms the main pathway to the decoder. The frequency representation instead

provides the primary localization information, while the temporal branch contributes most effectively as contextual information. This asymmetry also motivates the frame-injection analysis of Section 5.2.5.

5.2.5 Dense Decoder

The dense decoder reconstructs a sample-level representation from Z_{fused}, reinjects the intra-token features Φ and refines the resulting sequence temporally. The following experiments examine the token-to-sample converter, the temporal refiner, its receptive field and the frame injection.

Token-to-Sample Converter

The localized wavelet converter of Section 4.6.1 is compared with simpler reconstruction schemes. The overlap-add MLP replaces wavelet synthesis with a learned window projection, while the wavelet-midpoint and midpoint-MLP variants retain only the central region of each reconstructed window, following the central-crop strategy of SleepGPT [6]. Two additional variants incorporate the intra-token descriptors directly through either wavelet construction or per-frame concatenation.

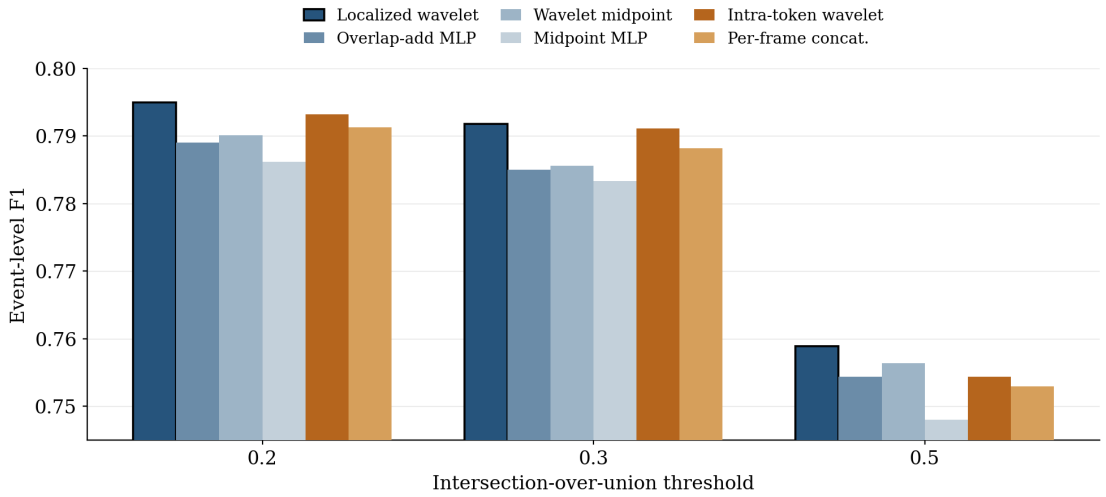

Figure 5.1: Effect of the token-to-sample converter across the evaluated IoU thresholds.

The localized wavelet converter achieves the highest F1 score at every IoU threshold, with the intra-token wavelet as the closest alternative at IoU 0.2 and 0.3. The overlap-add MLP remains consistently below it, while the midpoint MLP produces the largest decrease, particularly at IoU 0.5. These results support structured wavelet synthesis with overlapping reconstruction rather than central-window selection.

Temporal Refiner

The TCN of Section 4.6.3 is compared with a BiLSTM [20], a one-dimensional U-Net [33], a multi-scale dilated architecture [66] and a windowed temporal encoder [36]. The converter and frame injection remain unchanged across configurations.

Table 5.11: Effect of the temporal refiner.

F1 at IoU	**Temporal refiner**				
	TCN	BiLSTM	U-Net	Multi-scale	Temporal encoder
0.2	**0.7950**	0.7873	0.7879	0.7909	0.7732
0.3	**0.7918**	0.7841	0.7847	0.7872	0.7699
0.5	**0.7589**	0.7511	0.7530	0.7550	0.7294

The TCN achieves the highest F1 score at every IoU threshold, with the multi-scale variant as the closest alternative. The BiLSTM and U-Net remain slightly below it, while the windowed temporal encoder produces the largest decrease. These results support retaining the TCN for sample-level temporal refinement.

Receptive Field

The TCN receptive field was additionally varied by modifying its dilation schedule. The retained 1.51-second configuration provides the highest F1 score at IoU 0.5, while longer receptive fields offer no consistent improvement. The complete analysis is reported in Appendix D.

Frame Injection

The frame-aware connection of Section 4.6.2 reinjects the intra-token features Φ after sample-level reconstruction. Its contribution is evaluated by removing this connection while leaving the remaining decoder unchanged.

Table 5.12: Effect of the frame injection.

F1 at IoU	**Frame injection**		
	Enabled	Disabled	Difference
0.2	**0.7950**	0.7905	+0.0045
0.3	**0.7918**	0.7871	+0.0047
0.5	**0.7589**	0.7524	+0.0065

Frame injection improves F1 at every IoU threshold, with a slightly larger gain under stricter event matching. Although the gain is modest with the complete fused

representation, the branch ablation of Section 5.2.4 shows that Φ provides an additional source of localized frequency information when the temporal representation is supplied to the decoder.

5.2.6 Optimization Configuration

The reference model was trained with AdamW. To assess its sensitivity to the optimization strategy, the same architecture and data split were retrained with SGD [67]. SGD used a learning rate of 10^{-2}, a momentum of 0.9 and no Nesterov acceleration. The model was trained over 80 epochs. The longer training accounts for the slower convergence of SGD.

Table 5.13: Effect of the optimization configuration.

F1 at IoU	AdamW	SGD
0.2	**0.7950**	0.7893
0.3	**0.7918**	0.7865
0.5	**0.7589**	0.7530

AdamW achieves the highest F1 score. The best SGD checkpoint was selected at epoch 40, whereas the AdamW checkpoint was selected at epoch 39. These results indicate limited sensitivity to the optimization configuration and support retaining AdamW as the reference setting.

5.3 Results Analysis

The retained model is analyzed on the OPA test subjects (Appendix F.1) in terms of detection performance, temporal localization, physiological consistency and failure modes.

5.3.1 Detection and Localization Quality

Table 5.14 reports event-level performance at the three IoU thresholds used throughout the experiments. Precision and recall remain closely balanced. With the validation-selected threshold $\eta = 0.59$ and merge gap of 0.2 seconds, F1 decreases from 0.7950 at IoU 0.2 to 0.7575 at IoU 0.5.

Table 5.14: Event-level performance on the OPA test set.

IoU	Precision	Recall	F1	TP	FP	FN
0.2	0.7914	0.7987	0.7950	4261	1123	1074
0.3	0.7883	0.7955	0.7919	4244	1140	1091
0.5	0.7541	0.7610	0.7575	4060	1324	1275

Figure 5.2 extends the comparison across the complete range of matching thresholds. Performance remains stable under moderate overlap requirements and decreases more rapidly only when close agreement between predicted and reference boundaries is imposed.

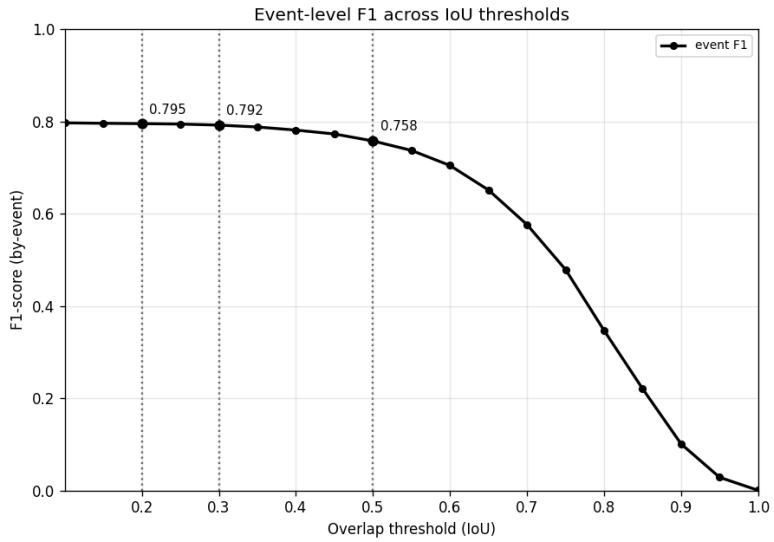

Figure 5.2: Event-level F1 as a function of the IoU threshold on the OPA test set.

Among pairs matched at IoU 0.2, the median overlap is 0.78. The interquartile range is 0.69-0.86. In addition, 95% of matched pairs exceed an IoU of 0.5 and 72% exceed 0.7. The complete distribution is reported in Appendix D.4.

False-positive predictions have a median event score of 0.98, indicating that they are not due to weak model outputs. Since OPA provides a single annotation set, these events cannot be clearly interpreted as either plausible unlabeled spindles or confident model errors.

5.3.2 Boundary Placement

Figure 5.3 shows the localization errors of matched events. Predicted onsets occur a median of 8 ms after the reference and offsets 31 ms before it, producing a median duration error of −55 ms. The limited median displacement indicates little systematic bias, while

the broader event-to-event variability explains the performance reduction at stricter IoU thresholds.

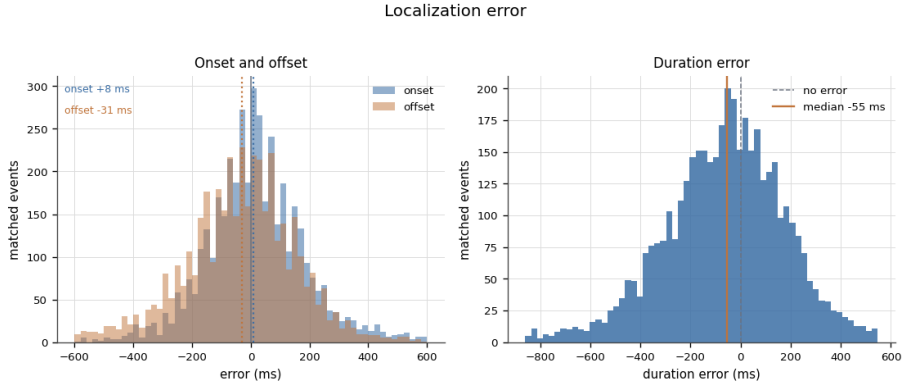

Figure 5.3: Localization errors for matched event pairs. The left panel shows the signed onset and offset errors. The right panel shows the resulting duration error.

5.3.3 Physiological Consistency

Figure 5.4 compares spindle density and duration with the OPA annotations using the spindle statistics defined in Section 2.4.4. Density is underestimated for two subjects and overestimated for the third, indicating no consistent directional bias. Predicted durations remain slightly shorter and less dispersed, consistent with the boundary analysis.

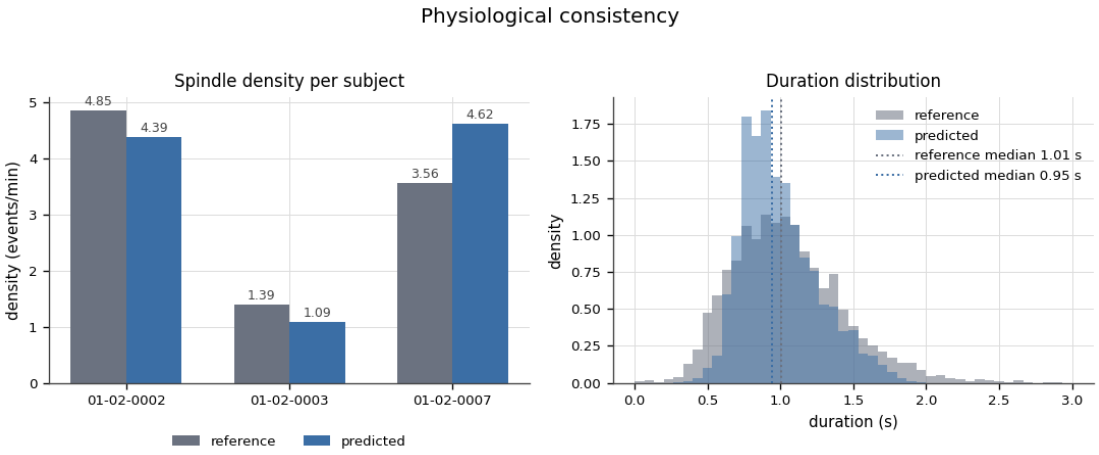

Figure 5.4: Physiological consistency of the predicted events. The left panel compares spindle density for each test subject. The right panel compares the predicted and reference duration distributions.

5.3.4 Error Characterization

Recall varies strongly with sigma amplitude, increasing from 0.42 in the lowest amplitude quartile to 0.96 in the highest. Missed detections are therefore concentrated among events with weak oscillatory evidence. Duration has a less pronounced effect, with recall increasing from 0.64 in the shortest quartile to 0.92 in the longest. Within the evaluated test set, low sigma amplitude is more strongly associated with missed detections than short duration.

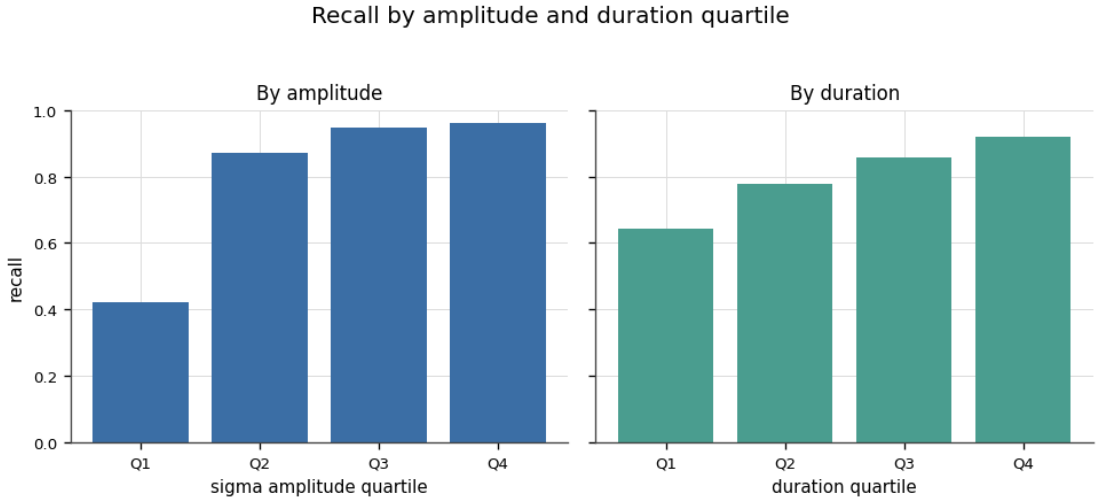

Figure 5.5: Recall stratified by sigma-amplitude and duration quartiles. Recall varies more strongly with sigma amplitude than with event duration.

5.3.5 Additional Diagnostics

Matched spindles exhibit important temporal overlap and limited recurring boundary displacement. Duration is recovered with a mean absolute error of 206 ms and a correlation of $r = 0.71$, while spindle density reaches a mean absolute error of 0.60 events/min and a correlation of $r = 0.91$. Median peak sigma frequency is 13.2 Hz for the reference events and 13.3 Hz for the predictions.

Fragmentation and merging remain below 1% and subject-level F1 ranges from 0.75 to 0.83, indicating that pooled performance is not driven by a single recording. At the sample level, the model reaches an ROC AUC of 0.980 and a false-positive rate of 0.051 at 90% sensitivity. The complete diagnostic statistics are reported in Appendix D.5.

5.4 Comparison and Generalization with the State of the Art

This section compares the proposed architecture with SUMO [34], BLAST [20] and SleepGPT [6] under common, published and cross-dataset evaluation settings.

5.4.1 Comparison Protocol

Detection scores depend on the evaluated subjects, reference annotations and event-matching procedure. The evaluation is therefore organized into three complementary settings. Event matching and event-level metrics follow Section 5.1.4.

Common-protocol comparison. The proposed model, SUMO [34] and BLAST [20] are trained and evaluated on the same OPA split using identical annotations, validation-based threshold selection and event matching. The hyperparameters of SUMO and BLAST are optimized on the validation set to ensure a fair comparison, with the retained values reported in Appendix E. This setting provides the direct architectural comparison.

Published Protocols comparison. The proposed architecture is also evaluated under the protocols of the three reference methods. The SUMO protocol is reproduced directly from its released MODA partition (Appendix F.2). The original BLAST and SleepGPT fold assignments are unavailable and are therefore reconstructed following their reported procedures. BLAST and the proposed architecture are retrained on reconstructed MASS-SS2 E1∪E2 folds (Appendix F.3). SleepGPT is not retrained. The proposed architecture is evaluated under reconstructed five-fold E1 and E2 protocols and compared with the published SleepGPT scores (Appendix F.4). The proposed architecture hyperparameters remain fixed and are not re-optimized for each annotation source. Following the reported SleepGPT protocol, the validation and test folds coincide, so these results should be interpreted as protocol reconstructions rather than exact replications.

Cross-dataset comparison. Zero-shot transfer is evaluated on the 166-subject MODA subset obtained after excluding the 14 MASS-SS2-derived recordings, leaving 4938 reference events. Decoding parameters are selected on MASS-SS2 validation data and applied to MODA without fine-tuning or recalibration. The first experiment varies the MASS-SS2 training annotations among OPA, E1, E2 and E1∪E2 using a common split (Appendix F.5). The second reuses the proposed model, SUMO and BLAST from the common-protocol comparison in Section 5.4.2 and evaluates them directly on MODA.

5.4.2 Common-Protocol Comparison

Table 5.15 compares the proposed model, BLAST and SUMO under the common OPA protocol, with η selected independently on the validation set for each method.

Table 5.15: Common-protocol comparison on the OPA test split.

Method	η	Precision@0.2	Recall@0.2	F1@0.2
Proposed	0.59	0.7914	**0.7987**	**0.7950**
BLAST	0.56	0.7922	0.7703	0.7811
SUMO	0.68	**0.8207**	0.7208	0.7675

The proposed model achieves the highest F1 score (0.7950), exceeding BLAST by 0.0139 and SUMO by 0.0275. It achieves the highest recall, while SUMO obtains the highest precision. Figure 5.6 shows how this comparison evolves as the matching criterion becomes stricter.

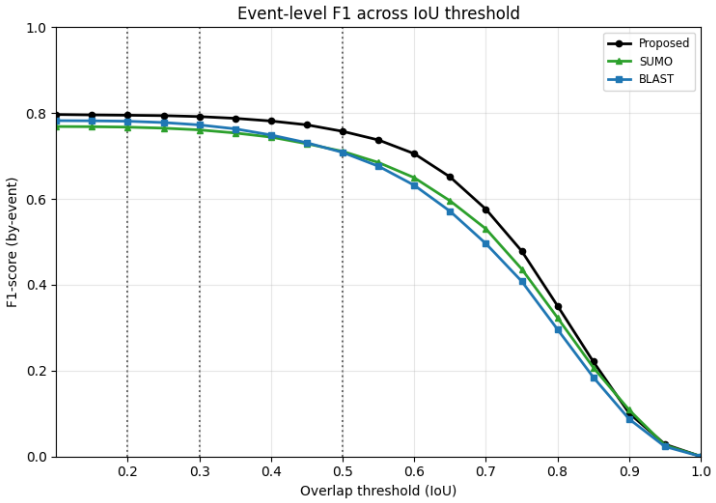

Figure 5.6: Event-level F1 as a function of the IoU threshold under the OPA protocol.

5.4.3 Published Protocol

Table 5.16 compares the proposed architecture with the reference methods under their respective published or reconstructed protocols.

Table 5.16: Event-level F1 at IoU 0.2 under each reference protocol. Reproduced and proposed scores are mean ± standard deviation across folds.

Protocol	Model	F1@0.2
SUMO (MODA)	SUMO (published)	0.820
	SUMO (reproduced)	0.8136 ± 0.0060
	Proposed	0.8201 ± 0.0029
BLAST (MASS-SS2, E1∪E2)	BLAST (published)	0.853 ± 0.056
	BLAST (reproduced)	0.7278 ± 0.0506
	Proposed	0.6706 ± 0.1128
SleepGPT E1 (MASS-SS2)	SleepGPT (published)	0.717
	Proposed	0.5876 ± 0.0885
SleepGPT E2 (MASS-SS2)	SleepGPT (published)	0.791
	Proposed	0.7137 ± 0.0421

The proposed model performs comparably to SUMO under the MODA protocol. Under the reconstructed BLAST protocol, reproduced BLAST outperforms the proposed architecture. The proposed model also remains below the published SleepGPT scores on both E1 and E2. The published BLAST score corresponds to training without data augmentation and no standard deviation is available for the published SUMO and SleepGPT results.

5.4.4 Cross-Dataset Generalization

The following presents the two cross-dataset experiments described in Section 5.4.1.

Effect of the Training Annotation Source

Because E2 and E1∪E2 are available only for the fifteen subjects scored by the E2 expert, all four models use the same fifteen-subject split. The OPA model is therefore retrained for this experiment and selects $\eta = 0.63$, rather than the 0.59 obtained on the full nineteen-subject OPA split.

Table 5.17: Effect of the MASS-SS2 training annotation source on zero-shot transfer to MODA.

Training labels	η	Precision	Recall	F1@0.2	F1@0.3	F1@0.5
OPA	0.63	**0.6747**	**0.6328**	**0.6531**	**0.6447**	**0.5837**
E2	0.70	0.3217	0.1902	0.2390	0.1456	0.0417
E1 ∪ E2	0.70	0.3038	0.1705	0.2184	0.1123	0.0327
E1	0.70	0.3437	0.0561	0.0964	0.0581	0.0233

OPA provides the strongest zero-shot transfer at every IoU threshold (Figure 5.7). E2 performs slightly better than E1∪E2, while E1 yields the lowest performance, primarily because of its low recall. The substantially stronger transfer from OPA indicates greater compatibility with the consensus-based MODA annotations.

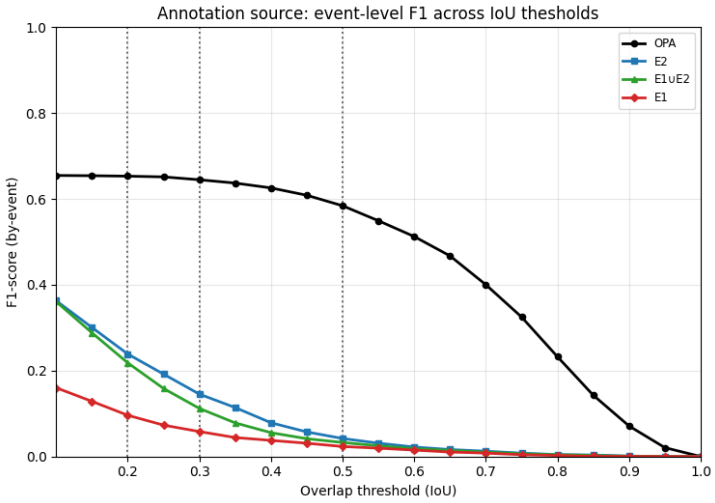

Figure 5.7: Event-level F1 as a function of the IoU threshold for models trained with different MASS-SS2 annotation sources and evaluated zero-shot on MODA.

Effect of the Detection Architecture

Table 5.18 compares the zero-shot transfer performance of the three architectures, each trained on the same OPA data.

Table 5.18: Effect of the detection architecture on zero-shot OPA-to-MODA transfer.

Architecture	η	Precision	Recall	F1@0.2
Proposed	0.59	0.7540	0.6077	0.6730
SUMO	0.68	**0.7880**	**0.6977**	**0.7401**
BLAST	0.56	0.7832	0.6644	0.7190

The cross-dataset ranking differs from the common OPA comparison. SUMO transfers best, followed by BLAST and the proposed model. This reversal shows that stronger in-domain performance does not necessarily imply better zero-shot generalization. Figure 5.8 compares the three architectures across IoU thresholds.

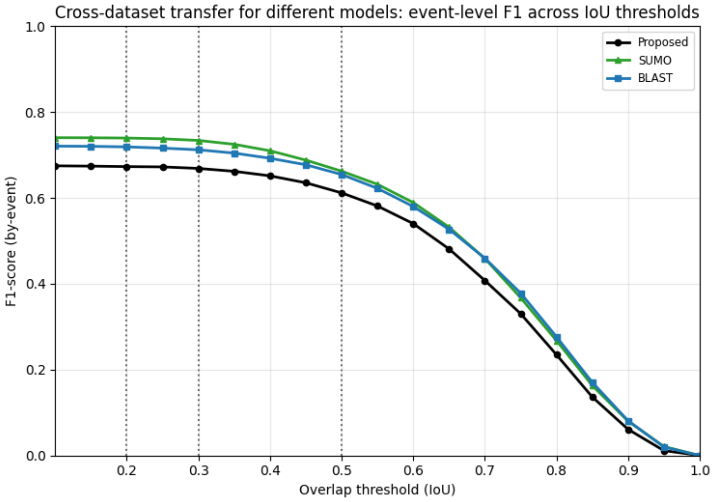

Figure 5.8: Event-level F1 as a function of the IoU threshold for the proposed model, SUMO and BLAST under zero-shot OPA-to-MODA transfer.

5.4.5 Summary

Under the common OPA protocol, the proposed model achieves the highest F1 score, but this advantage does not generalize to every evaluation setting. It performs comparably to SUMO under the MODA protocol, remains below BLAST under the reconstructed BLAST protocol and transfers less effectively than both SUMO and BLAST from OPA to MODA.

The cross-dataset experiments also reveal a strong dependence on the annotation source. OPA-trained models transfer substantially better to the consensus-based MODA annotations than models trained on E1, E2, or their union. These results indicate that reported detector performance depends not only on architecture but also on annotation compatibility, motivating the joint consideration of in-domain performance and cross-dataset robustness.

Chapter 6

Conclusion and Future Work

This thesis investigated the adaptation of a pretrained sleep foundation model for dense sleep spindle detection. A principal challenge is the discrepancy between the token-level representation produced by SleepFM and the sample-level resolution required for precise event localization. The proposed architecture addresses this challenge by utilizing the pretrained temporal representation as contextual information while preserving localized spectral information through a task-specific frequency branch.

6.1 Summary of Findings

Experimental results indicate that the pretrained temporal representation alone does not provide sufficient localization for precise spindle detection. In contrast, the frequency branch retains the information necessary for accurate event reconstruction. The best performance is obtained when the frequency representation functions as the primary localization stream, with the pretrained temporal representation providing complementary context. These findings suggest that effective adaptation of a sleep foundation model for dense event detection requires the preservation of local information at a resolution finer than the token scale.

The final model achieves an event-level F1 score of 0.7950 at an IoU threshold of 0.2 on the held-out OPA test set. Detected events are generally well localized and retain essential spindle characteristics. Under the common OPA protocol, the proposed architecture outperforms both BLAST and SUMO. However, this performance advantage does not consistently extend to cross-dataset evaluation. SUMO and BLAST exhibit superior transferability from OPA to MODA, indicating that higher in-domain performance does not necessarily correspond to improved generalization.

Cross-dataset experiments further highlight the significance of the annotation source. Models trained on OPA exhibit substantially greater compatibility with the MODA consensus reference compared to those trained on public MASS-SS2 annotations.

6.2 Limitations

The OPA test set comprises 5335 spindle events from only three participants, which restricts the ability to draw conclusions regarding inter-subject variability. Additionally, annotations were produced by a single expert scorer, precluding independent adjudication of residual disagreements.

Most architectural ablations were assessed using single experimental runs, so large performance differences provide stronger evidence than minor improvements. Cross-dataset evaluation is further constrained because MODA recordings are derived from the MASS archive. Consequently, the experiments primarily evaluate changes in population and annotation protocol rather than transfer across entirely independent acquisition settings.

6.3 Future Work

A first direction concerns the supervision of the current detector. The OPA and MODA annotations provide event-specific weights that are not exploited in the present training procedure. Incorporating these weights into the loss could allow each spindle to contribute according to the confidence associated with its annotation. This may be particularly useful for weak or borderline events, for which the residual analysis showed greater detection difficulty.

A more fundamental direction concerns the foundation model itself. The findings of this thesis suggest that pretrained representations provide useful contextual information for spindle detection, but that precise localization also requires fine temporal and spectral structure to remain accessible. This motivates a foundation model approach that preserves such information directly during pretraining rather than a task-specific model, because the goal is a reusable representation for many downstream tasks, not only spindle detection. The idea would be to combine the complementary principles of SleepGPT [6] and SleepMaMi [38]. SleepGPT jointly learns temporal and frequency-domain representations, while SleepMaMi models sleep at both micro and macro temporal scales. A hierarchical time-frequency foundation model could therefore learn local temporal and spectral features before progressively integrating them into representations of longer sleep context. This could provide the fine-grained information required for dense micro-event detection while retaining the wider organization of sleep.

This framework could later be evaluated on other transient events such as K-complexes. Validation on independently acquired datasets with consensus annotations would also provide a stronger assessment of robustness across populations and recording conditions.

Appendix A

Implementation Details of SleepFM Pretraining

This appendix complements the pretraining procedure introduced in Section 4.3.1. The main methodology chapter defines the regional views, the leave-one-out objective and the token-level representation retained by the spindle detector. The following sections focus on the practical details required to reproduce pretraining, including the corpus composition, channel harmonization, input construction, encoder configuration and optimization. The exact similarity formulation used by the leave-one-out objective and the edge handling of the downstream token grid are also reported here.

A.1 Pretraining Dataset

The encoder was pretrained using recordings from seven sleep cohorts. Each recording was stored in a separate HDF5 file, while a shared split file assigned it to either pretraining or validation. The pretraining partition contained 6633 recordings and an additional 1780 recordings were reserved for validation. Table A.1 summarizes the pretraining partition.

Table A.1: Composition of the corpus employed for pretraining.

Cohort	**Pretraining recordings**
Sleep Heart Health Study (SHHS)	3802
Wisconsin Sleep Cohort (WSC)	1094
MrOS Sleep Study (MrOS)	562
Multi-Ethnic Study of Atherosclerosis (MESA)	528
STAGES	304
MNC	234
Cleveland Family Study (CFS)	109
Total	**6633**

No cohort-specific sampling weights were applied. Each cohort therefore contributed approximately in proportion to its number of pretraining recordings, while differences in acquisition protocols and EEG montages were handled through the channel-harmonization procedure described in Section A.2.

The available metadata identify the recordings assigned to each split but do not provide their individual durations. The total pretraining duration and the exact number of complete 5-min sequences could therefore not be determined reliably.

A.2 Channel Harmonization and Regional Assignment

The contributing cohorts use different electrode montages, reference schemes and naming conventions. A shared channel dictionary maps the available derivation names to the frontal, central and posterior views introduced in Section 4.3.1. Assignment is based on the complete derivation name rather than an electrode prefix inferred during loading.

The dictionary contains 136 frontal aliases, 67 central aliases and 21 posterior aliases. These sets are mutually exclusive and account for the reference conventions encountered across the pretraining datasets. Table A.2 summarizes the principal electrode families covered by each regional group.

Table A.2: Regional EEG groups used during leave-one-out pretraining.

View	Principal electrodes	Reference conventions	Aliases
Frontal	$F3$, $F4$, $F7$, $F8$, $Fp1$, $Fp2$, Fpz, Fz	$A1/A2$, $M1/M2$, linked-ear, contralateral-ear and averaged references	136
Central	$C3$, $C4$, Cz	$A1/A2$, $M1/M2$, linked-ear, contralateral-ear and averaged references	67
Posterior	$O1$, $O2$, Oz, Pz-Oz	$A1/A2$, $M1/M2$, linked-ear, contralateral-ear and averaged references	21

The channels available within a regional view may differ across recordings. Rather than enforcing a fixed montage, the encoder processes the derivations assigned to each region and combines their embeddings through attention pooling. Recordings with different channel counts can therefore share the same encoder while preserving the regional structure required by the leave-one-out objective.

A.3 Pretraining Input Construction

The data loader operates on EEG signals stored at a common sampling frequency of 128 Hz. The inspected repository contains the harmonized HDF5 files and the runtime preprocessing procedure, but not the complete upstream conversion pipeline. The available implementation therefore does not establish whether amplitude normalization was applied per channel, recording, or cohort before storage.

During loading, each signal is band-limited through an FFT-domain operation. Frequency bins outside the 0.5–40 Hz interval are set to zero before the signal is reconstructed through the inverse transform. This operation preserves the phase of the retained coefficients and avoids the delay introduced by causal temporal filtering.

Each recording is divided into non-overlapping 5-min chunks. At 128 Hz, a complete chunk contains 38400 samples. With the 384-sample tokenization defined in Section 4.3.2, each chunk yields 100 non-overlapping tokens. No sleep-stage restriction is applied during pretraining, so the sequences preserve the natural succession of sleep stages in the original recordings.

Table A.3: Construction of the inputs used during pretraining.

Setting	**Value**
Input format	One harmonized HDF5 file per recording
Sampling frequency	128 Hz
Runtime frequency range	0.5–40 Hz
Filtering operation	FFT-domain spectral masking followed by inverse transformation
Chunk duration	5 min
Samples per complete chunk	38400
Chunk overlap	None
Token duration	3 s
Samples per token	384
Tokens per complete chunk	100
Sleep-stage restriction	None

A.4 Encoder Configuration

The architecture follows the processing sequence introduced in Section 4.3, while Table A.4 reports the additional implementation details. The convolutional tokenizer is applied independently to each available derivation. Its feature width increases progressively before projection to the embedding dimension $E = 128$.

Attention pooling first aggregates the channel embeddings available for each temporal token. Positional information is then introduced before contextualization by a six-layer Transformer encoder. A second attention-pooling operation summarizes the contextualized sequence into the chunk-level representation used by the pretraining objective. The ordered sequence before this final pooling operation provides the token representation retained by the downstream detector.

Table A.4: Configuration of the encoder used during pretraining.

Component	Configuration
Convolutional kernel	Size 5, stride 2, padding 2
Convolutional feature widths	$1 \to 4 \to 8 \to 16 \to 32 \to 64$
Tokenizer normalization	Batch normalization followed by layer normalization
Embedding dimension	$E = 128$
Regional aggregation	Attention pooling over the available channel set
Contextual encoder	6 Transformer encoder layers
Attention heads	8
Transformer normalization	Pre-normalization
Transformer dropout	0.3
Pretraining readout	Attention pooling over the contextualized token sequence
Representation reused downstream	Ordered token sequence before temporal pooling

A.5 Implementation of the Leave-One-Out Objective

The leave-one-out objective introduced in Equation 4.11 is applied to the chunk-level representation obtained after temporal attention pooling, whereas the spindle detector reuses the token-level output described in Section 4.3.3.

Writing $z_m^{(b)}$ for the L_2-normalized embedding of regional group m in the b-th of B batch segments, the complementary representation is formed by averaging the normalized embeddings of the other two regional groups,

$$z_{-m}^{(b)} = \frac{1}{|M| - 1} \sum_{m' \neq m} z_{m'}^{(b)} \tag{A.1}$$

The positive pair associates one regional view with the complement obtained from the same EEG chunk, while the remaining chunks in the batch provide the negatives.

The similarity matrix for regional group m is

$$S_{bb'}^{m} = e^{\tau_{\text{pre}}} \left\langle z_m^{(b)}, z_{-m}^{(b')} \right\rangle \tag{A.2}$$

where τ_{pre} is a learnable log-scale parameter and $e^{\tau_{\text{pre}}}$ is the resulting positive logit scale. Cross-entropy is applied in both directions of S^m,

$$\ell_{\text{NCE}}(z_m, z_{-m}) = -\frac{1}{2B} \sum_{b=1}^{B} \left[\log \frac{\exp S_{bb}^{m}}{\sum_{b'} \exp S_{bb'}^{m}} + \log \frac{\exp S_{bb}^{m}}{\sum_{b'} \exp S_{b'b}^{m}} \right] \tag{A.3}$$

This symmetric formulation favors the diagonal entries, which pair regional views from the same EEG chunk, over representations originating from the other chunks in the batch.

Gradients propagate through the selected representation and its complementary aggregate because no stop-gradient operation is applied. Negatives are restricted to the current batch, without cross-worker gathering or an external memory bank. The encoder parameters are shared across all three regional groups.

The learnable parameter τ_{pre} is initialized to zero, corresponding to an initial logit scale $e^{\tau_{\text{pre}}} = 1$ and is optimized jointly with the shared encoder.

A.6 Downstream Token-Grid Edge Handling

During pretraining, the 5-min chunks are divided into non-overlapping 384-sample tokens. During downstream spindle detection, the same pretrained tokenizer is scanned with the detector stride S_{tok}, as described in Section 4.3.3. When $T_0 - W$ is not an exact multiple of S_{tok}, the regular grid does not cover the final samples of the chunk. A final right-aligned token is therefore appended at

$$a_{N-1} = T_0 - W \tag{A.4}$$

The resulting number of tokens is

$$N = \begin{cases} \left\lfloor \dfrac{T_0 - W}{S_{\text{tok}}} \right\rfloor + 1 & (T_0 - W) \bmod S_{\text{tok}} = 0, \\ \left\lfloor \dfrac{T_0 - W}{S_{\text{tok}}} \right\rfloor + 2, & \text{otherwise} \end{cases} \tag{A.5}$$

The overlap between the last two windows can therefore exceed the regular overlap, but every sample remains associated with at least one temporal window. The corresponding start positions are retained and later used to reconstruct token-level information on the original sample axis.

A.7 Optimization Configuration

The shared encoder and learnable logit scale are optimized jointly through stochastic gradient descent. The learning rate is halved every 15 epochs, without an initial warm-up period. The training loop does not apply gradient clipping, gradient accumulation, or automatic mixed precision.

Table A.5: Optimization configuration used for leave-one-out pretraining.

Setting	**Value**	**Setting**	**Value**
Optimizer	Stochastic gradient descent	Batch size	32 chunks
Learning rate	3×10^{-4}	Training duration	60 epochs
Momentum	0.9	Learning-rate scheduler	StepLR with a factor of 0.5 every 15 epochs
Weight decay	1×10^{-5}	Learning-rate warm-up	None
Dropout	0.3	Data-loader workers	16
Random seed	42	Logit scale	Learnable scalar initialized to 0
Gradient clipping	None	Gradient accumulation	None

Appendix B

Fixed Wavelet Atom Bank

The token-to-sample reconstruction described in Section 4.6.1 uses a fixed bank of $K_a = 16$ wavelet atom types, evaluated at $P_a = 6$ candidate positions within each $W = 384$-sample token.

For atom type $k \in \{0, \ldots, K_a - 1\}$, the carrier-cycle parameter is

$$c_k = 1 + \frac{3k}{K_a - 1} \tag{B.1}$$

giving 16 values uniformly distributed from 1 to 4 cycles over the token support. The corresponding carrier frequency is

$$f_k = \frac{c_k}{W - 1} f_s \tag{B.2}$$

with $f_s = 128$ Hz. The resulting frequencies range from approximately 0.334 to 1.337 Hz, corresponding to temporal periods from approximately 3.0 to 0.75 s. These frequencies do not represent the sigma oscillation itself. Instead, they define event-scale temporal variations within the 3-s token, over a range comparable to typical spindle durations.

The Gaussian temporal widths are logarithmically spaced between

$$\sigma_{\min} = \frac{W}{16} = 24, \qquad \sigma_{\max} = \frac{W}{2.5} = 153.6 \tag{B.3}$$

samples, with each pair (f_k, σ_k) defining one atom type.

The $P_a = 6$ candidate centers are placed at

$$p_m \in \{0.20,\ 0.32,\ 0.44,\ 0.56,\ 0.68,\ 0.80\}, \qquad \mu_m = p_m(W - 1) \tag{B.4}$$

This corresponds to approximately

$$\{76.60,\ 122.56,\ 168.52,\ 214.48,\ 260.44,\ 306.40\}$$

samples, or

$$\{0.598,\ 0.957,\ 1.317,\ 1.676,\ 2.035,\ 2.394\}\ \text{s}$$

relative to the beginning of the token.

Each atom follows the Gaussian-modulated oscillatory formulation introduced in Section 2.2.3, centered at μ_m with carrier frequency f_k and temporal width σ_k. The waveform is evaluated directly over the finite token support without padding or integer translation.

Each atom is standardized along the temporal dimension to have zero mean and unit variance. The complete bank is then normalized at each temporal position by the root-mean-square amplitude across all atom types and candidate positions. This bank-wise normalization is applied twice, with temporal standardization repeated after each iteration.

The resulting bank satisfies

$$\Psi \in \mathbb{R}^{16\times 6\times 384} \tag{B.5}$$

corresponding to 16 atom types, 6 candidate positions and 384 token-local samples. The bank therefore contains 96 localized waveforms and receives no gradients during training. Only the atom coefficients and candidate-position probabilities predicted from each fused token are learned.

Appendix C

Decoding Procedure

This appendix describes the process for converting dense predictions into event intervals, as summarized in Section 5.1.4. The procedure utilizes only the mask probability and boundary offsets. Samples satisfying $\hat{y}_t \geq \eta$ constitute the candidate intervals. Candidates separated by less than $g_{\mathbf{merge}}$ seconds are merged to restore events fragmented by transient probability drops prior to extent estimation. The boundaries of each candidate are then refined. The refinement process incorporates every sample within a candidate interval. Each sample proposes an onset and offset based on its predicted distances,

$$s_t = t - \hat{\Delta}^t_{\text{on}}, \qquad e_t = t + \hat{\Delta}^t_{\text{off}} \tag{C.1}$$

The onset and offset proposals are aggregated using a weighted median, with each proposal weighted by the mask probability of its corresponding sample,

$$\hat{t}_{\text{on}} = \text{median}_w \{s_t\}, \qquad \hat{t}_{\text{off}} = \text{median}_w \{e_t\}, \qquad w_t = \hat{y}_t \tag{C.2}$$

A median is employed instead of a mean because a single sample with an erroneous predicted distance disproportionately affects the mean, while it alters the median by only one rank. The refined boundaries are rounded to the nearest sample and may extend beyond the original candidate interval. Consequently, events with progressively decaying amplitude may be assigned an extent larger than the region exceeding the threshold.

Appendix D

Additional Experimental Analyses

This appendix reports implementation details, sensitivity analyses and complementary diagnostic results supporting the experiments of Chapter 5.

D.1 Frequency-Branch Input Variants

The encoder-input ablation of Section 5.2.2 evaluates three families of modifications to the Morlet scalogram.

The first family applies squeeze-and-excitation recalibration [49]. The retained frequency-wise variant (f-SE) predicts one weight per frequency, as described in Section 4.4.3. The finer time-frequency variant (TF-SE) instead predicts a separate weight for each frequency and frame.

The second family injects handcrafted spindle descriptors inspired by the signal characteristics used by the A7 detector of Lacourse et al. [22]. Each patch is represented by

$$v = [\log E_\sigma, e_{\text{rel}}, r_\sigma]$$

where $\log E_\sigma$ is the log-energy in the sigma band, e_{rel} measures sigma energy relative to its spectral context and r_σ is the Pearson correlation between the raw signal and its sigma-filtered counterpart. The descriptor is supplied either through a projected physiology token or by direct concatenation with the learned representation.

The third family appends two fixed feature maps to the input patch [61]. A normalized frequency ramp encodes the absolute position of each frequency bin, while a Gaussian profile centered on the sigma range provides an explicit spindle spectral prior.

D.2 Frequency-Branch Parameter Sensitivity

The architectural ablations in Section 5.2.2 are complemented by a sensitivity analysis of the main numerical hyperparameters of the retained frequency branch.

The frame stride H_{sc} in Equation 4.19 governs the temporal sampling density of the transform. The number of oscillation cycles n_c sets the trade-off between temporal

and frequency resolution of the Morlet kernels. Encoder depth is assessed by appending a fourth convolutional layer to the three-stage hierarchy defined in Equation 4.26. Finally, the number of temporal segments T_s in Equation 4.27 determines the intra-token resolution of the structured readout.

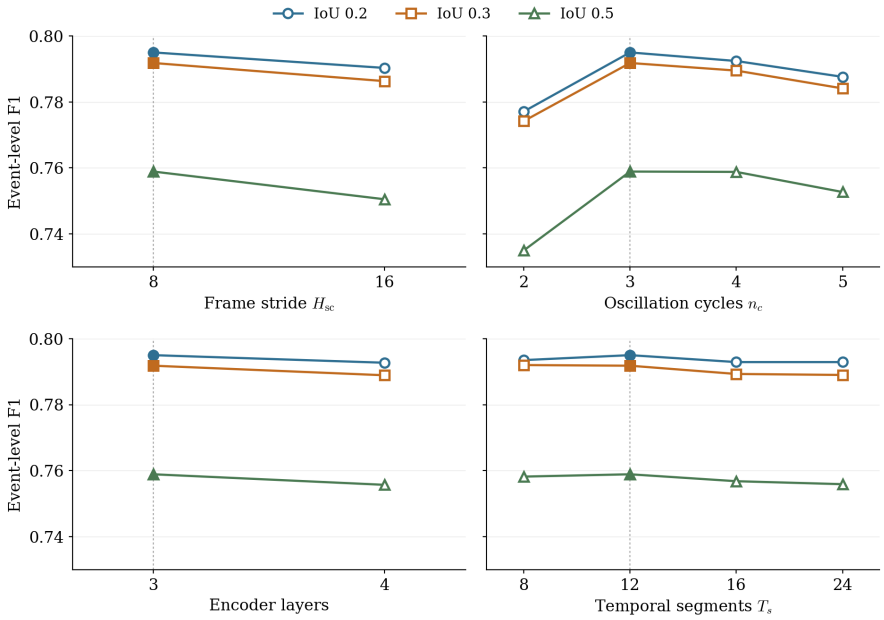

Figure D.1: Parameter sensitivity of the frequency branch. Filled markers denote the retained value and the dashed line marks its position.

The retained stride $H_{\text{sc}} = 8$ always outperforms $H_{\text{sc}} = 16$, confirming the benefit of the denser temporal sampling used by the final model. Three oscillation cycles provide the best overall trade-off, although $n_c = 4$ reaches an almost identical result at IoU 0.5. Adding a fourth convolutional layer does not improve performance. For the structured readout, $T_s = 12$ gives the highest F1 scores at IoU 0.2 and 0.5, while $T_s = 8$ is higher by only 0.0002 at IoU 0.3. Increased resolution beyond 12 segments provides no benefit. Each panel varies smoothly around its retained value and none of the curves exhibits a sharp optimum, indicating that the branch is not finely tuned to a particular setting.

D.3 Temporal Receptive-Field Sensitivity

The temporal-refiner comparison of Section 5.2.5 changes both the refinement mechanism and the organization of temporal context. A separate experiment therefore modifies only the dilation schedule of the TCN. The retained schedule produces a receptive field of 1.51 seconds, while the alternatives extend it to 3.01 and 4.04 seconds.

Table D.1: Effect of the temporal receptive field.

F1 at IoU	**Receptive field**		
	1.51 s	3.01 s	4.04 s
0.2	0.7950	**0.7964**	0.7938
0.3	0.7918	**0.7921**	0.7900
0.5	**0.7589**	0.7571	0.7548

The 3.01-second configuration slightly improves F_1@0.2 and F_1@0.3, but reduces F_1@0.5. Extending the receptive field to 4.04 seconds further decreases performance across all thresholds. Since the decoder is primarily responsible for precise boundary reconstruction, the stricter IoU 0.5 criterion is given greater importance. The 1.51-second receptive field is therefore retained because it provides the best localization accuracy.

D.4 IoU Distribution of Matched Events

Figure D.2 shows the IoU distribution for event pairs matched at IoU 0.2. The median overlap is 0.78, with an interquartile range of 0.69-0.86. Among these pairs, 95% exceed an IoU of 0.5, while 72% exceed 0.7.

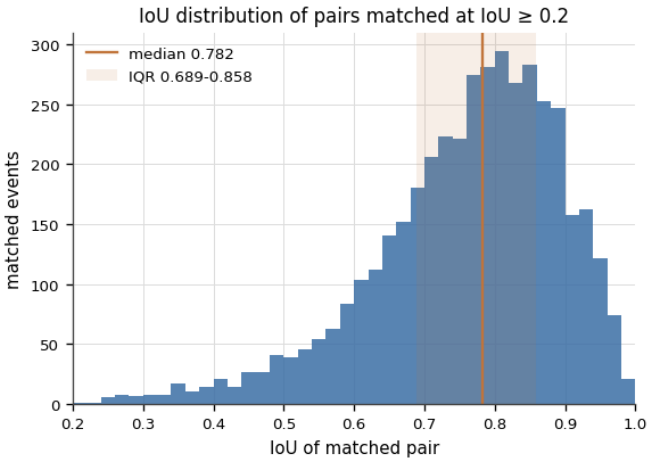

Figure D.2: Distribution of IoU values for event pairs matched at IoU 0.2.

D.5 Additional Diagnostic Statistics

Table D.2 reports complementary localization, physiological, event-structure, confidence and sample-level statistics on the OPA test set.

Table D.2: Additional diagnostic statistics on the OPA test set.

Analysis	**Statistic**	**Value**
Localization	Mean absolute duration error	206 ms
	Duration correlation	$r = 0.71$
Physiological consistency	Density mean absolute error	0.60 events/min
	Density correlation	$r = 0.91$
	Median peak sigma frequency, reference	13.2 Hz
	Median peak sigma frequency, predicted	13.3 Hz
Event structure	Fragmented reference events	0.5%
	Merged predicted events	0.6%
	Subject-level F1 range	0.75–0.83
Detection confidence	Median unmatched-event score	0.98
Sample-level evaluation	ROC AUC	0.980
	False-positive rate at 90% sensitivity	0.051

Appendix E

Baseline Hyperparameter Optimization

To ensure a fair common-protocol comparison on the OPA annotations, the training hyperparameters of BLAST and SUMO were optimized separately with Optuna using the OPA training and validation sets. Their architectures remained unchanged. The retained hyperparameters are reported below.

Table E.1: Retained hyperparameters for BLAST.

Hyperparameter	**Retained value**
Learning rate	4.4606×10^{-3}
OneCycle warm-up fraction	0.3591
Initial LR divisor	5.9473
Final LR divisor	139.9267
Weight decay	2.2424×10^{-4}
Maximum momentum	0.9695
Positive-class weight	1.7239
Batch size	32
Early-stopping patience	10

Table E.2: Retained hyperparameters for SUMO.

Hyperparameter	**Retained value**
Learning rate	1.0290×10^{-3}
Weight decay	9.9765×10^{-4}
Dice smoothing	2.9142×10^{-6}
Batch size	24
Dice class weighting	True

Appendix F

Dataset Splits

This appendix details the subject membership for each dataset split referenced in Chapter 5. The experimental design utilizes two subject universes: the OPA re-annotation encompasses all nineteen MASS-SS2 subjects, while the comparison with public expert annotations (E1, E2) is limited to the fifteen subjects evaluated by both experts, since E2 and E1∪E2 are undefined for the remaining four subjects (0004, 0008, 0015, 0016). The proposed architecture is trained separately with OPA annotations on the two subject sets, selecting $\eta = 0.63$ for the fifteen-subject annotation-source split and $\eta = 0.59$ for the full nineteen-subject OPA split. Altering the annotation source among E1, E2 and their union does not affect the partition. MASS-SS2 subject identifiers share the prefix `01-02-`. Therefore, tables report only the four-digit suffix. The SUMO dataset spans five MODA cohorts (`01-01` to `01-05`).

F.1 OPA Common-Protocol Split

Table F.1: OPA common-protocol split (19 subjects), used for the common-protocol comparison and the temporal-encoder ablation.

Partition	Subjects (`01-02-`)
Train (13)	0001, 0004, 0005, 0006, 0010, 0012, 0013, 0014, 0015, 0016, 0017, 0018, 0019
Validation (3)	0008, 0009, 0011
Test (3)	0002, 0003, 0007

F.2 SUMO Folds

The fixed test set (36 subjects, identical across all six folds) is `01-01-`{0004, 0023, 0028, 0029, 0043, 0044, 0046, 0048, 0049}; `01-02-`{0006, 0010}; `01-03-`{0003, 0004, 0012, 0014, 0023, 0026, 0027, 0031, 0033, 0040, 0051, 0059, 0060, 0062, 0064}; `01-04-`{0007, 0008, 0013, 0015, 0016, 0029}; `01-05-`{0002, 0018, 0020, 0024}. Each fold trains on the 120 subjects formed by the 144 non-test subjects minus its validation set.

Table F.2: Rotating 24-subject validation folds for the SUMO comparison.

Fold	Validation (24)
00	01-01-{0003, 0008, 0021, 0024, 0026, 0031, 0035}; 01-02-{0011, 0013}; 01-03-{0005, 0007, 0020, 0044, 0046, 0053, 0054, 0058}; 01-04-{0011, 0012, 0030, 0038}; 01-05-{0008, 0014, 0023}
01	01-01-{0005, 0014, 0017, 0019, 0025, 0040, 0050}; 01-02-{0002, 0012}; 01-03-{0011, 0019, 0025, 0035, 0041, 0050, 0063}; 01-04-{0002, 0019, 0021, 0023, 0036}; 01-05-{0003, 0017, 0025}
02	01-01-{0012, 0013, 0016, 0018, 0042, 0047, 0051}; 01-02-{0001, 0017}; 01-03-{0017, 0021, 0028, 0037, 0042, 0048, 0057}; 01-04-{0018, 0020, 0022, 0035, 0037}; 01-05-{0001, 0005, 0019}
03	01-01-{0007, 0011, 0022, 0030, 0037, 0041}; 01-02-{0004, 0019}; 01-03-{0002, 0008, 0015, 0018, 0022, 0030, 0056, 0061}; 01-04-{0003, 0005, 0017, 0033, 0034}; 01-05-{0006, 0016, 0022}
04	01-01-{0006, 0010, 0020, 0032, 0038, 0052}; 01-02-{0005, 0015}; 01-03-{0006, 0016, 0029, 0032, 0034, 0038, 0047, 0052}; 01-04-{0009, 0014, 0025, 0028}; 01-05-{0007, 0009, 0010, 0013}
05	01-01-{0001, 0002, 0027, 0033, 0036, 0053}; 01-02-{0003, 0014}; 01-03-{0001, 0009, 0010, 0013, 0024, 0036, 0039, 0045}; 01-04-{0006, 0026, 0031, 0040}; 01-05-{0004, 0011, 0012, 0015}

F.3 BLAST Leave-One-Subject-Out Folds

Folds 08 and 09 share the validation pair (0006, 0010) and differ only in the test subject, following the stratified fold construction.

Table F.3: Leave-one-subject-out folds for the BLAST reconstruction (12/2/1). Training sets are the complement of the listed subjects within the fifteen-subject pool.

Fold	Test	Validation
00	0001	0005, 0014
01	0002	0001, 0003
02	0003	0002, 0017
03	0005	0009, 0014
04	0006	0005, 0018
05	0007	0013, 0018
06	0009	0001, 0003
07	0010	0007, 0012
08	0011	0006, 0010
09	0012	0006, 0010
10	0013	0009, 0019
11	0014	0011, 0013
12	0017	0002, 0012
13	0018	0007, 0019
14	0019	0011, 0017

F.4 SleepGPT Folds

Table F.4: Five-fold splits for the SleepGPT reconstruction (faithful protocol, validation = test, so selection is performed on the test fold).

Fold	Test = Validation	Train (12)
00	0001, 0013, 0019	0002, 0003, 0005, 0006, 0007, 0009, 0010, 0011, 0012, 0014, 0017, 0018
01	0007, 0009, 0018	0001, 0002, 0003, 0005, 0006, 0010, 0011, 0012, 0013, 0014, 0017, 0019
02	0005, 0010, 0011	0001, 0002, 0003, 0006, 0007, 0009, 0012, 0013, 0014, 0017, 0018, 0019
03	0002, 0006, 0014	0001, 0003, 0005, 0007, 0009, 0010, 0011, 0012, 0013, 0017, 0018, 0019
04	0003, 0012, 0017	0001, 0002, 0005, 0006, 0007, 0009, 0010, 0011, 0013, 0014, 0018, 0019

F.5 Annotation-Source Split

Table F.5: Annotation-source split (15 both-expert subjects), used for the OPA, E1, E2 and E1∪E2 comparison.

Partition	Subjects (`01-02-`)
Train (10)	0001, 0002, 0003, 0005, 0007, 0009, 0014, 0017, 0018, 0019
Validation (3)	0006, 0010, 0013
Test (2)	0011, 0012

Declaration of Generative AI and AI-assisted Technologies in the Writing Process

During the preparation of this thesis, the author used ChatGPT and DeepL to support English-language writing and improve the clarity of the manuscript. OpenAI Codex was used to assist with code development and debugging. After using these tools, the author reviewed and edited the generated content as needed. All scientific claims, experimental designs, results, interpretations and conclusions were verified and approved by the author, who takes full responsibility for the final content of this thesis.

Bibliography

[1] Dara S. Manoach and Robert Stickgold. "Abnormal Sleep Spindles, Memory Consolidation, and Schizophrenia". In: *Annual Review of Clinical Psychology* 15 (2019), pp. 451–479. DOI: 10.1146/annurev-clinpsy-050718-095754.

[2] Bernhard P. Staresina et al. "Hierarchical Nesting of Slow Oscillations, Spindles and Ripples in the Human Hippocampus During Sleep". In: *Nature Neuroscience* 18.11 (2015), pp. 1679–1686. DOI: 10.1038/nn.4119.

[3] Jorge Lopez, Robert Hoffmann, and Roseanne Armitage. "Reduced Sleep Spindle Activity in Early-Onset and Elevated Risk for Depression". In: *Journal of the American Academy of Child & Adolescent Psychiatry* 49.9 (2010), pp. 934–943. DOI: 10.1016/j.jaac.2010.05.014.

[4] Diego Z. Carvalho et al. "Loss of Sleep Spindle Frequency Deceleration in Obstructive Sleep Apnea". In: *Clinical Neurophysiology* 125.2 (2014), pp. 306–312. DOI: 10.1016/j.clinph.2013.07.005.

[5] Rahul Thapa et al. "A Multimodal Sleep Foundation Model for Disease Prediction". In: *Nature Medicine* 32.2 (2026), pp. 752–762. DOI: 10.1038/s41591-025-04133-4.

[6] Weixuan Huang et al. "A Unified Time-Frequency Foundation Model for Sleep Decoding". In: *Nature Communications* 17 (2026), p. 1198. DOI: 10.1038/s41467-025-67970-4.

[7] Michael H. Silber et al. "The Visual Scoring of Sleep in Adults". In: *Journal of Clinical Sleep Medicine* 3.2 (2007), pp. 121–131. DOI: 10.5664/jcsm.26814.

[8] Etienne Combrisson et al. "Sleep: An Open-Source Python Software for Visualization, Analysis, and Staging of Sleep Data". In: *Frontiers in Neuroinformatics* 11 (2017), p. 60. DOI: 10.3389/fninf.2017.00060.

[9] Seunghyeok Hong and Hyun Jae Baek. "Drowsiness Detection Based on Intelligent Systems with Nonlinear Features for Optimal Placement of Encephalogram Electrodes on the Cerebral Area". In: *Sensors* 21.4 (2021), p. 1255. DOI: 10.3390/s21041255.

[10] Luigi De Gennaro and Michele Ferrara. "Sleep Spindles: An Overview". In: *Sleep Medicine Reviews* 7.5 (2003), pp. 423–440. DOI: 10.1053/smrv.2002.0252.

[11] Laura M. J. Fernandez and Anita Lüthi. "Sleep Spindles: Mechanisms and Functions". In: *Physiological Reviews* 100.2 (2020), pp. 805–868. DOI: 10.1152/physrev.00042.2018.

[12] Alan V. Oppenheim and Ronald W. Schafer. *Discrete-Time Signal Processing*. 3rd ed. Pearson, 2010.

[13] Leon Cohen. *Time-Frequency Analysis*. Prentice Hall, 1995.

[14] Stéphane Mallat. *A Wavelet Tour of Signal Processing: The Sparse Way*. 3rd. Academic Press, 2008.

[15] Michael X. Cohen. "A Better Way to Define and Describe Morlet Wavelets for Time-Frequency Analysis". In: *NeuroImage* 199 (2019), pp. 81–86. DOI: 10.1016/j.neuroimage.2019.05.048.

[16] Guomin Luo et al. "Time–Frequency Entropy-Based Partial-Discharge Extraction for Nonintrusive Measurement". In: *IEEE Transactions on Power Delivery* 27.4 (Oct. 2012), pp. 1919–1927. DOI: 10.1109/TPWRD.2012.2200911.

[17] Karine Lacourse et al. "Massive Online Data Annotation, Crowdsourcing to Generate High Quality Sleep Spindle Annotations from EEG Data". In: *Scientific Data* 7 (2020), p. 190. DOI: 10.1038/s41597-020-0533-4.

[18] Nicolás I. Tapia-Rivas, Pablo A. Estévez, and José A. Cortes-Briones. "A Robust Deep Learning Detector for Sleep Spindles and K-Complexes: Towards Population Norms". In: *Scientific Reports* 14 (2024), p. 263. DOI: 10.1038/s41598-023-50736-7.

[19] Jiaxin You et al. "SpindleU-Net: An Adaptive U-Net Framework for Sleep Spindle Detection in Single-Channel EEG". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 29 (2021), pp. 1614–1623. DOI: 10.1109/TNSRE.2021.3105443.

[20] Chenyun Guo et al. "BLAST: Towards Robust Detection of Sleep Spindles Across Clinical Settings". In: *Neurocomputing* 653 (2025), p. 131107. DOI: 10.1016/j.neucom.2025.131107.

[21] Waqas Ahmed, Pekka Toivanen, and Keijo Haataja. "Toward Sleep Spindle Detection: A Comparative Survey of State-of-the-Art, Challenges and Future Research". In: *IEEE Access* 13 (2025), pp. 182821–182845. DOI: 10.1109/ACCESS.2025.3622316.

[22] Karine Lacourse et al. "A Sleep Spindle Detection Algorithm That Emulates Human Expert Spindle Scoring". In: *Journal of Neuroscience Methods* 316 (2019), pp. 3–11. DOI: 10.1016/j.jneumeth.2018.08.014.

[23] Ankit Parekh et al. "Detection of K-Complexes and Sleep Spindles (DETOKS) Using Sparse Optimization". In: *Journal of Neuroscience Methods* 251 (2015), pp. 37–46. DOI: 10.1016/j.jneumeth.2015.04.006.

[24] John LaRocco et al. “Spindler: A Framework for Parametric Analysis and Detection of Spindles in EEG with Application to Sleep Spindles”. In: *Journal of Neural Engineering* 15.6 (2018), p. 066015. DOI: 10.1088/1741-2552/aadc1c.

[25] Daniel Lachner-Piza et al. “A Single Channel Sleep-Spindle Detector Based on Multivariate Classification of EEG Epochs: MUSSDET”. In: *Journal of Neuroscience Methods* 297 (2018), pp. 31–43. DOI: 10.1016/j.jneumeth.2017.12.023.

[26] Christian O’Reilly et al. “Montreal Archive of Sleep Studies: An Open-Access Resource for Instrument Benchmarking and Exploratory Research”. In: *Journal of Sleep Research* 23.6 (2014), pp. 628–635. DOI: 10.1111/jsr.12169.

[27] Stéphanie Devuyst et al. “Automatic Sleep Spindles Detection — Overview and Development of a Standard Proposal Assessment Method”. In: *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2011, pp. 1713–1716. DOI: 10.1109/IEMBS.2011.6090491.

[28] Takafumi Kinoshita et al. “Sleep Spindle Detection Using RUSBoost and Synchrosqueezed Wavelet Transform”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 28.2 (2020), pp. 390–398. DOI: 10.1109/TNSRE.2020.2964597.

[29] Carlo R. Patti, Thomas Penzel, and Dean Cvetkovic. “Sleep Spindle Detection Using Multivariate Gaussian Mixture Models”. In: *Journal of Sleep Research* 27.4 (2018), e12614. DOI: 10.1111/jsr.12614.

[30] Prathamesh M. Kulkarni et al. “A Deep Learning Approach for Real-Time Detection of Sleep Spindles”. In: *Journal of Neural Engineering* 16.3 (2019), p. 036004. DOI: 10.1088/1741-2552/ab0933.

[31] Jiahui Pan et al. “Deep Learning-Augmented Sleep Spindle Detection for Acute Disorders of Consciousness: Integrating CNN and Decision Tree Validation”. In: *IEEE Transactions on Biomedical Engineering* 72.10 (2025), pp. 3120–3132. DOI: 10.1109/TBME.2025.3562067.

[32] Yann LeCun et al. “Gradient-Based Learning Applied to Document Recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Springer, 2015, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28.

[34] Lars Kaulen et al. “Advanced Sleep Spindle Identification with Neural Networks”. In: *Scientific Reports* 12.1 (2022), p. 7686. DOI: 10.1038/s41598-022-11210-y.

[35] Jihao Tian and Yan Wei. “Sleep Spindle Auto Detection Based on U-Net with Label Optimization Module”. In: *Proceedings of the 2025 2nd International Conference on Generative Artificial Intelligence and Information Security (GAIIS 2025)*. ACM, 2025, pp. 382–391. DOI: 10.1145/3728725.3728786.

[36] Ashish Vaswani et al. "Attention Is All You Need". In: *Advances in Neural Information Processing Systems*. Vol. 30. 2017. DOI: 10.48550/arXiv.1706.03762.

[37] Zitao Shuai et al. *OSF: On Pre-training and Scaling of Sleep Foundation Models*. 2026. DOI: 10.48550/arXiv.2603.00190. arXiv: 2603.00190 [cs.LG].

[38] Keondo Park et al. *SleepMaMi: A Universal Sleep Foundation Model for Integrating Macro- and Micro-structures*. 2026. DOI: 10.48550/arXiv.2602.07628. arXiv: 2602.07628 [cs.AI].

[39] Yao-Hung Hubert Tsai et al. "Multimodal Transformer for Unaligned Multimodal Language Sequences". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019, pp. 6558–6569. DOI: 10.18653/v1/P19-1656.

[40] Jean-Baptiste Alayrac et al. "Flamingo: A Visual Language Model for Few-Shot Learning". In: *Advances in Neural Information Processing Systems*. Vol. 35. 2022, pp. 23716–23736. DOI: 10.48550/arXiv.2204.14198. arXiv: 2204.14198.

[41] George Trigeorgis et al. "Deep Canonical Time Warping". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 5110–5118. DOI: 10.1109/CVPR.2016.552.

[42] Isma Hadji, Konstantinos G. Derpanis, and Allan D. Jepson. "Representation Learning via Global Temporal Alignment and Cycle-Consistency". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 11068–11077. DOI: 10.1109/CVPR46437.2021.01092.

[43] Nikita Dvornik et al. "Drop-DTW: Aligning Common Signal Between Sequences While Dropping Outliers". In: *Advances in Neural Information Processing Systems*. Vol. 34. 2021, pp. 13782–13793. DOI: 10.48550/arXiv.2108.11996. arXiv: 2108.11996 [cs.CV].

[44] Diederik P. Kingma and Max Welling. "Auto-Encoding Variational Bayes". In: *International Conference on Learning Representations (ICLR)*. 2014. DOI: 10.48550/arXiv.1312.6114. arXiv: 1312.6114.

[45] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. "Distilling the Knowledge in a Neural Network". In: *arXiv preprint arXiv:1503.02531* (2015). DOI: 10.48550/arXiv.1503.02531. arXiv: 1503.02531.

[46] Sungsoo Ahn et al. "Variational Information Distillation for Knowledge Transfer". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 9155–9163. DOI: 10.1109/CVPR.2019.00938.

[47] CEAMS. *SS2 Sleep Annotations*. Borealis, the Canadian Dataverse Repository. Version V1. 2022. DOI: 10.5683/SP3/Y889CS.

[48] Michal Sejak et al. "OpenSpindleNet: An Open-Source Deep Learning Network for Reliable Sleep Spindle Detection in Scalp and Intracranial EEG". In: *Computers in Biology and Medicine* 197 (2025), p. 110854. DOI: 10.1016/j.compbiomed.2025.110854.

[49] Jie Hu, Li Shen, and Gang Sun. "Squeeze-and-Excitation Networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 7132–7141. DOI: 10.1109/CVPR.2018.00745.

[50] Dan Hendrycks and Kevin Gimpel. "Gaussian Error Linear Units (GELUs)". In: *arXiv preprint arXiv:1606.08415* (2016). DOI: 10.48550/arXiv.1606.08415. arXiv: 1606.08415 [cs.LG].

[51] Rajeswari Jayaraj and Jagannath Mohan. "Classification of Sleep Apnea Based on Sub-Band Decomposition of EEG Signals". In: *Diagnostics* 11.9 (2021), p. 1571. DOI: 10.3390/diagnostics11091571.

[52] Zheng Chen et al. "Automated Sleep Staging via Parallel Frequency-Cut Attention". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 31 (2023), pp. 1974–1985. DOI: 10.1109/TNSRE.2023.3243589.

[53] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. "Layer Normalization". In: *arXiv preprint arXiv:1607.06450* (2016). DOI: 10.48550/arXiv.1607.06450. arXiv: 1607.06450 [stat.ML].

[54] Nitish Srivastava et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958.

[55] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling". In: *arXiv preprint arXiv:1803.01271* (2018). DOI: 10.48550/arXiv.1803.01271.

[56] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning Representations by Back-Propagating Errors". In: *Nature* 323.6088 (1986), pp. 533–536. DOI: 10.1038/323533a0.

[57] Tsung-Yi Lin et al. "Focal Loss for Dense Object Detection". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 2980–2988. DOI: 10.1109/ICCV.2017.324.

[58] Ross Girshick. "Fast R-CNN". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 1440–1448. DOI: 10.1109/ICCV.2015.169.

[59] Takuya Akiba et al. "Optuna: A Next-Generation Hyperparameter Optimization Framework". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 2019, pp. 2623–2631. DOI: 10.1145/3292500.3330701.

[60] Mirco Ravanelli and Yoshua Bengio. "Speaker Recognition from Raw Waveform with SincNet". In: *2018 IEEE Spoken Language Technology Workshop (SLT)*. 2018, pp. 1021–1028. DOI: 10.1109/SLT.2018.8639585.

[61] Rosanne Liu et al. "An Intriguing Failing of Convolutional Neural Networks and the CoordConv Solution". In: *Advances in Neural Information Processing Systems*. Vol. 31. 2018. DOI: 10.48550/arXiv.1807.03247.

[62] Galen Andrew et al. "Deep Canonical Correlation Analysis". In: *International Conference on Machine Learning (ICML)*. 2013, pp. 1247–1255.

[63] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. "Representation Learning with Contrastive Predictive Coding". In: *arXiv preprint arXiv:1807.03748* (2018). DOI: 10.48550/arXiv.1807.03748. arXiv: 1807.03748.

[64] Tadas Baltruaitis, Chaitanya Ahuja, and Louis-Philippe Morency. "Multimodal Machine Learning: A Survey and Taxonomy". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.2 (2019), pp. 423–443. DOI: 10.1109/TPAMI.2018.2798607.

[65] Ofir Press, Noah A. Smith, and Mike Lewis. "Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation". In: *International Conference on Learning Representations*. 2022. DOI: 10.48550/arXiv.2108.12409.

[66] Jianing Li et al. "Global-Local Temporal Representations for Video Person Re-Identification". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 3958–3967. DOI: 10.1109/ICCV.2019.00406.

[67] Léon Bottou. "Large-Scale Machine Learning with Stochastic Gradient Descent". In: *Proceedings of COMPSTAT'2010*. Ed. by Yves Lechevallier and Gilbert Saporta. Physica-Verlag HD, 2010, pp. 177–186. DOI: 10.1007/978-3-7908-2604-3_16.