SleepTransformer: Automatic Sleep Staging with Interpretability and Uncertainty Quantification

Huy Phan*, Kaare Mikkelsen, Oliver Y. Chén, Philipp Koch, Alfred Mertins, and Maarten De Vos

Abstract—*Background:* Black-box skepticism is one of the main hindrances impeding deep-learning-based automatic sleep scoring from being used in clinical environments. *Methods:* Towards interpretability, this work proposes a sequence-to-sequence sleep-staging model, namely SleepTransformer. It is based on the transformer backbone and offers interpretability of the model's decisions at both the epoch and sequence level. We further propose a simple yet efficient method to quantify uncertainty in the model's decisions. The method, which is based on entropy, can serve as a metric for deferring low-confidence epochs to a human expert for further inspection. *Results:* Making sense of the transformer's self-attention scores for interpretability, at the epoch level, the attention scores are encoded as a heat map to highlight sleep-relevant features captured from the input EEG signal. At the sequence level, the attention scores are visualized as the influence of different neighboring epochs in an input sequence (i.e. the context) to recognition of a target epoch, mimicking the way manual scoring is done by human experts. *Conclusion:* Additionally, we demonstrate that SleepTransformer performs on par with existing methods on two databases of different sizes. *Significance:* Equipped with interpretability and the ability of uncertainty quantification, SleepTransformer holds promise for being integrated into clinical settings.

Index Terms—Automatic sleep staging, transformer, interpretability, uncertainty estimation, deep neural network, sequence-to-sequence.

I. INTRODUCTION

Sleep deprivation is prevalent and sleep disorders affect millions of people worldwide [1], posing an huge burden on public health. Current practice of sleep diagnosis and assessment is still heavily dependent on human expertise. Machine intelligence, which is disrupting various application fields, holds huge potential for automating current sleep annotation. Although it is not the goal that machine intelligence will entirely replace human sleep experts [2]–[4], we envision it could work alongside and assist human experts to facilitate their jobs and scale up sleep assessment and diagnosis.

Sleep staging, the first and fundamental step in sleep diagnosis and assessment, is a typical application where machine intelligence can excel. In practice, this task of assigning a sleep stage to a 30-second sleep epoch is still being done manually, following a predefined set of rules, such as the American Academy of Sleep Medicine (AASM) guideline [5]. On average, a sleep expert needs to spend two hours to complete scoring an overnight polysomnography (PSG) recording [6], making manually handling millions of sleep recordings infeasible. Automating this labor-intensive and routine process will free up a huge amount of time and efforts from sleep experts as a machine can complete the same task in a few seconds. Furthermore, automatic sleep scoring is indispensable when it comes to longitudinal sleep monitoring in home environments with novel mobile-EEG devices [7], [8].

Significant progress has been made towards automatic sleep staging in the last few years. The availability of large-scale public sleep databases with hundreds [9] to thousands of subjects [10], [11] has stimulated and enabled the exploration of deep learning paradigms in solving this problem [12]–[17]. Early attempts tried to use vanilla deep network architectures, such as deep neural networks (DNNs) [18], convolutional neural networks (CNNs) [17], [19]–[22], and recurrent neural networks (RNNs) [23]. Replacing more conventional machine learning methods with these vanilla networks in simple one-to-one or many-to-one frameworks resulted in limited success, owing to the limitation of the short input context. Since the seminal work in [15], the sequence-to-sequence sleep staging approach has grown in popularity for the task. Using this framework, handling a long context of 20-30 consecutive PSG epochs simultaneously, various advanced architectures have been proposed, for example CNN+RNN [24], [25], hierarchical RNN [15], [26], and CNN+Transformer [27]. It further allows extensions from different angles, such as transfer learning [28]–[30], model personalization [31], [32], and multi-view learning [16]. These advances have significantly pushed the performance of machine sleep scoring to be on par with human scoring [12], [15], [16], [26].

Despite all this progress, we have not yet seen automatic sleep staging widely adopted clinically. Unofficial communications with leading sleep experts point to the scepticism of deep learning models being a black box, which is a common criticism when it comes to the application of artificial intelligence in healthcare and medicine [33]. We argue that two overarching obstacles need to be addressed for a machine scoring system to work alongside practitioners in an interactive and collaborative manner: (1) interpretability [34], [35] and (2) uncertainty quantification [36]. Interpretability is the ability of a model to explain how its decision is made given a certain

H. Phan is with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, UK and the Alan Turing Institute, London NW1 2DB, UK. K. Mikkelsen is with the Department of Electrical and Computer Engineering, Aarhus University, Aarhus 8200, Denmark. O. Y. Chén is with the School of Economics, Finance and Management, University of Bristol, Bristol BS8 1TU, UK. P. Koch and A. Mertins are with the Institute for Signal Processing, University of Lübeck, Lübeck 23562, Germany and with the German Research Center for Artificial Intelligence (DFKI), Lübeck 23562, Germany. M. De Vos is with the Department of Electrical Engineering and with the Department of Development and Regeneration, KU Leuven, 3001 Leuven, Belgium.
*Corresponding author: `h.phan@qmul.ac.uk`

input, to be understood by a human. Inspired by the way a sleep expert performs manual scoring [5], interpretability in automatic sleep scoring is reasonably about (but not limited to) what features the model learns from the input signal, whether these features are relevant to and underpin the sleep stages, and how the decision on a target epoch is made under the influence of its neighboring epochs. Interpretability is particularly important due to the fact that sleep stages are ambiguous and even different human experts tend to disagree at a certain extend [37], [38]. Also, due to this ambiguity, quantifying uncertainty in the model's decisions is equally important. Simply put, we are in need of a simple and concrete metric, ideally a single number, for quantifying the model's uncertainty. Using this metric, epochs that are scored with low confidence by the model can be deferred to sleep experts for further inspection [39].

In this work, we propose a sleep staging model, namely SleepTransformer, as a stepping stone towards addressing the two above-mentioned obstacles. SleepTransformer adheres to the sequence-to-sequence sleep staging framework [15], [28]. However, different from most (if not indeed all) existing works, SleepTransformer is convolution- and recurrent-free. Instead, it relies on the transformer concept [40] as the backbone for both epoch- and sequence-level modelling. The transformer construction is solely based on a self-attention mechanism whose attention scores will be leveraged for the model's interpretability at both the epoch and sequence level. On the one hand, the attention scores at the epoch level will be used as a heat map applied to the EEG signal input to highlight the features the model attends to. On the other hand, the attention scores at the sequence level is interpreted as the influence of different neighboring epochs to the recognition of a target epoch in an input sequence. We also propose to use entropy of the multi-class probability distribution outputted by the model to neatly quantify uncertainty in its decisions. We show that the estimated uncertainty allows us to identify most of the model's mistakes. Experimental results on two public databases, Sleep Heart Health Study (SHHS) and Sleep-EDF Expanded, of varying size also show that SleepTransformer performs comparably to existing state-of-the-art models on the two databases.

Our major contributions are summarized as follows.

- The proposed SleepTransformer is a transformer-based sequence-to-sequence model which achieves state-of-the-art performance on automatic sleep scoring. To the best of our knowledge, this is the first sequence-to-sequence model solely relying on the transformer architecture proposed for the task.
- We address interpretability of a sleep-staging model in a natural way at both the epoch and sequence level by leveraging the attention scores of the transformer's self-attention module.
- We propose an entropy-based method to elegantly quantify uncertainty in the model's decisions as a concrete number.

The rest of the article is organized as follows. We outline the used databases in Section II. We then describe the architecture of the transformer backbone in Section III, followed by the proposed SleepTransformer in Section IV. We elaborate the interpretability and uncertainty quantification of the model in Section V. Details about the experiments will be presented in Section VI. We conclude the article in Section VII.

II. MATERIALS

The following two databases will be used for experiments in this work:

Sleep Heart Health Study (SHHS): This is a large-scale database collected from multiple centers to study the effect of sleep-disordered breathing on cardiovascular diseases [10], [11]. The data was collected as part of the clinical trial "Sleep Heart Health Study (SHHS)", ClinicalTrials.gov number, NCT00005275. It has two rounds of PSG records, namely Visit 1 (SHHS-1) and Visit 2 (SHHS-2). The former, consisting of 5,791 subjects aged 39-90, was employed in this work. Manual scoring was completed using the R&K guideline [41]. Similar to other databases annotated with the R&K rule, N3 and N4 stages were merged into N3 stage and MOVEMENT and UNKNOWN epochs were discarded. We adopted C4-A1 EEG in the experiments.

SleepEDF-78: This database is the 2018 version of the Sleep-EDF Expanded dataset [42], [43], consisting of 78 healthy Caucasian subjects aged 25-101. Two consecutive day-night PSG recordings were collected for each subject, except subjects 13, 36, and 52 whose one recording was lost due to device failure. Manual scoring was done by sleep experts according to the R&K standard [41] and each 30-second PSG epoch was labeled as one of eight categories {W, N1, N2, N3, N4, REM, MOVEMENT, UNKNOWN}. N3 and N4 stages were merged into N3 stage. MOVEMENT and UNKNOWN epochs were excluded. We used the Fpz-Cz EEG in this study. Of note, we adhere to the common setting where a recording was trimmed starting from 30 minutes before to 30 minutes after its *in-bed* part [16].

III. TRANSFORMER

Transformer [40], a sequence model solely based on self-attention, has shown compelling results on various sequential modelling tasks. The transformer is composed of an encoder and a decoder sharing the same model architecture. However, the decoder is a left-context-only version which is tasked for generation purpose. To avoid confusion, the transformer used in this work is the encoder part. It comprises two core modules: multi-head attention and position-wise feed-forward network.

The attention mechanism used in the multi-head attention module is scaled dot-product attention, as illustrated in Figure 1 (a). It associates elements at different positions of an input sequence to derive the output sequence which is computed as a weighted sum of the input values, where the weight for each value is computed by an attention function of the query with the corresponding keys. Multi-head attention is composed of H scaled dot-product attention modules, as illustrated in Figure 1 (b). Firstly, H different learnable linear projections are applied to the input and map it to parallel queries, keys, and

values. Then, the scaled dot-product attention is performed on these mapped queries, keys, and values simultaneously. The H attention heads are then concatenated, followed by a linear projection to produce the attentive output. All these steps can be formulated as follows:

$$\mathbf{Q}_i = \mathbf{Z}\mathbf{W}_i^Q, \ \mathbf{K}_i = \mathbf{Z}\mathbf{W}_i^K, \ \mathbf{V}_i = \mathbf{Z}\mathbf{W}_i^V, \ 1 \leq i \leq H, \quad (1)$$

$$\mathbf{H}_i = \text{Attention}(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i)$$

$$= \text{softmax}(\frac{\mathbf{Q}_i\mathbf{K}_i^{\mathsf{T}}}{\sqrt{d}})\mathbf{V}_i, \quad (2)$$

$$\tilde{\mathbf{Z}} = \text{Concat}(\mathbf{H}_1, \ldots, \mathbf{H}_H)\mathbf{W}^Z. \quad (3)$$

Here, $\mathbf{Z} \in \mathbb{R}^{l \times d}$ is the input with length l and dimension d. $\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i \in \mathbb{R}^{l \times \frac{d}{H}}$ are the mapped queries, keys, and values. $\mathbf{H}_i \in \mathbb{R}^{l \times \frac{d}{H}}$ is the i-th attention head. $\mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V \in \mathbb{R}^{d \times \frac{d}{H}}$ and $\mathbf{W}^Z \in \mathbb{R}^{d \times d}$ are the learnable weight matrices. $\tilde{\mathbf{Z}} \in \mathbb{R}^{l \times d}$ is the attentive output.

The position-wise feed-forward network is a fully connected feed-forward network. It is comprised of two linear transformations with a ReLU activation in between. Besides the two main modules, the transformer also includes several residual and normalization layers as illustrated in Figure 1 (c). As a whole, it can be formulated as follows:

$$\tilde{\mathbf{Z}} = \text{MultiHeadAttention}(\mathbf{Z}), \quad (4)$$

$$\mathbf{Z}_{mid} = \text{LayerNorm}(\mathbf{Z} + \tilde{\mathbf{Z}}), \quad (5)$$

$$\mathbf{Z}_{FF} = \text{ReLU}(\mathbf{Z}_{mid}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2, \quad (6)$$

$$\mathbf{O} = \text{LayerNorm}(\mathbf{Z}_{mid} + \mathbf{Z}_{FF}). \quad (7)$$

Here, $\mathbf{Z}_{FF}$ denotes the output of the position-wise feed-forward network, in which $\mathbf{W}_1 \in \mathbb{R}^{d \times d_{FF}}$, $\mathbf{W}_2 \in \mathbb{R}^{d_{FF} \times d}$, $\mathbf{b}_1 \in \mathbb{R}^{d_{FF}}$, and $\mathbf{b}_2 \in \mathbb{R}^d$ are learnable weight matrices and biases, respectively.

IV. SLEEPTRANSFORMER

Given a training set $\{\mathcal{S}_n\}_{n=1}^N$ of size N where $\mathcal{S}_n = (\{\mathbf{S}_1^{(n)}, \mathbf{y}_1^{(n)}\}, \ldots, \{\mathbf{S}_L^{(n)}, \mathbf{y}_L^{(n)}\})$ is the n-th sequence of L sleep epochs. $\mathbf{S}_i^{(n)} \in \mathbb{R}^{T \times F}$, $1 \leq i \leq L$ represents a time-frequency image of $T = 29$ time frames and $F = 128$ frequency bins extracted from the i-th 30-second EEG epoch in the n-th
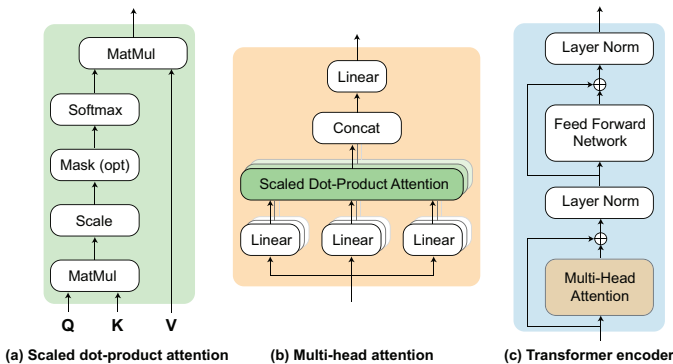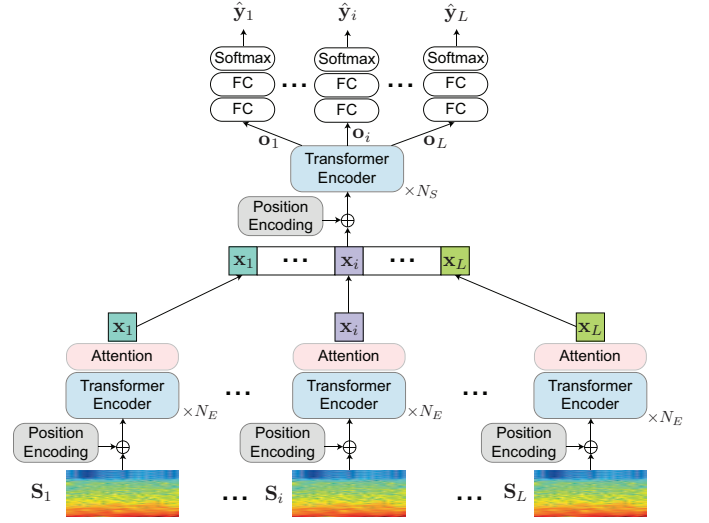

Figure 2: Illustration of SleepTransformer.

sequence (see Section VI-A). $\mathbf{y}_i^{(n)} \in \{0, 1\}^C$ denotes the one-hot encoding label of the i-th EEG epoch in the n-th sequence, where $C = 5$ as we are dealing with 5-stage sleep staging.

The proposed SleepTransformer, illustrated in Figure 2, adheres to the sequence-to-sequence sleep staging framework proposed in [15]. It uses the transformer described in Section III as the backbone network for both intra-epoch (i.e. epoch level) and inter-epoch (i.e. sequence level) processing. It is, therefore, free of convolutional and recurrent components which are the main ingredients in existing deep-learning models for sleep staging, such as [12], [13], [15]–[17], [44]–[46].

A. Epoch transformer

The epoch transformer plays the role of a feature map that transforms a 30-second EEG epoch into a feature vector for representation. This feature map has been commonly realized either by a CNN [13], [16] or an RNN [15]. Orthogonally, SleepTransformer realizes this map using N_E transformers.

A time-frequency image $\mathbf{S}$ is treated as a sequence of T spectral columns. Without confusion, we omit the superscript and subscript for simplicity. We aim to encode this sequence by a heap of N_E transformers which are denoted as *EpochTransformer*. As the transformer itself cannot encode the order information which is vital for both intra-epoch and inter-epoch processing in a sequence-to-sequence sleep-scoring network [15], we firstly add positional encodings to the input to introduce order information:

$$\tilde{\mathbf{S}} = \mathbf{S} + \mathbf{P}^{\text{ep}}. \quad (8)$$

In (8), $\mathbf{P}^{\text{ep}} \in \mathbb{R}^{T \times F}$ denotes the positional encoding matrix. We use sine and cosine functions as in the seminal work [40] for positional encoding purpose where the i-th row and the $(2j)$-th or the $(2j + 1)$-th column is given as

$$p_{i,2j} = \sin\left(\frac{i}{10000^{2j/F}}\right), \quad (9)$$

$$p_{i,2j+1} = \cos\left(\frac{i}{10000^{2j/F}}\right). \quad (10)$$

Figure 1: Architecture of (a) scaled dot-product attention, (b) multi-head attention, and (c) transformer encoder.

The sequence of spectral columns of $\tilde{\mathbf{S}}$ is then modelled by

$$\mathbf{X}^{(i)} = \text{EpochTransformer}\left(\mathbf{X}^{(i-1)}\right), \qquad (11)$$

where $\mathbf{X}^{(i)} \in \mathbb{R}^{T \times F}$, $1 \leq i \leq N_E$, and $\mathbf{X}^{(0)} \equiv \tilde{\mathbf{S}}$. In order to reduce $\mathbf{X}^{(N_E)}$, the output of the last transformer in the heap, to a compact feature vector for epoch-wise representation, we combine its columns $(\mathbf{x}_1^{(N_E)}, \ldots, \mathbf{x}_T^{(N_E)})$ via a weighted combination:

$$\mathbf{x} = \sum\nolimits_{t=1}^{T} \alpha_t \mathbf{x}_t^{(N_E)}. \qquad (12)$$

In (12), $\mathbf{x} \in \mathbb{R}^F$ denotes the derived feature vector that represents the input epoch. $\alpha_1, \ldots, \alpha_T$ are the attention weights learned by a softmax attention layer as in [15], [23]:

$$\alpha_t = \frac{\exp(\mathbf{a}_t^\mathsf{T} \mathbf{a}_e)}{\sum_{t=1}^{T} \exp(\mathbf{a}_t^\mathsf{T} \mathbf{a}_e)}, \qquad (13)$$

$$\mathbf{a}_t = \tanh(\mathbf{W}_a \mathbf{x}_t + \mathbf{b}_a), \qquad (14)$$

where $\mathbf{W}_a \in \mathbb{R}^{A \times F}$ and $\mathbf{b}_a \in \mathbb{R}^A$ are a learnable weight matrix and bias, respectively. $\mathbf{a}_e \in \mathbb{R}^A$ is the trainable epoch-level context vector. A is the so-called attention size.

B. Sequence transformer

Via the epoch transformer in Section (IV-A), an input sequence $(\mathbf{S}_1, \ldots, \mathbf{S}_L)$ has now been transformed into a sequence of epoch-wise feature vectors $(\mathbf{x}_1, \ldots, \mathbf{x}_L)$, where $\mathbf{x}_i, 1 \leq i \leq L$, is given in (12). In existing work complying to the sequence-to-sequence sleep staging framework, the resulting epoch-wise feature vectors $(\mathbf{x}_1, \ldots, \mathbf{x}_L)$ were typically processed by a bidirectional RNN for inter-epoch modelling [15]. Here, we employ a heap of N_S transformers, denoted as *SequenceTransformer*, for this purpose.

Similar to the epoch transformer, positional encoding is firstly carried out via sine-and-cosine functions:

$$\tilde{\mathbf{X}} = \mathbf{X} + \mathbf{P}^{\text{seq}}, \qquad (15)$$

where $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_L) \in \mathbb{R}^{L \times F}$. $\mathbf{P}^{\text{seq}} \in \mathbb{R}^{L \times F}$ denotes the positional encoding matrix whose elements are computed using (9) and (10). $\tilde{\mathbf{X}}$ is then processed by the heap of N_S *SequenceTransformer*:

$$\mathbf{O}^{(i)} = \text{SequenceTransformer}\left(\mathbf{O}^{(i-1)}\right), \qquad (16)$$

where $\mathbf{O}^{(i)} \in \mathbb{R}^{L \times F}$, $1 \leq i \leq N_S$, and $\mathbf{O}^{(0)} \equiv \tilde{\mathbf{X}}$.

Given the output of the last *SequenceTransformer*, $\mathbf{O}^{(N_S)} = (\mathbf{o}_1^{(N_S)}, \ldots, \mathbf{o}_L^{(N_S)})$, the vectors $\mathbf{o}_i^{(N_S)}$, $1 \leq i \leq L$, are eventually presented to two fully-connected (FC) layers with ReLU activation, followed by a softmax layer to obtain the output sequence $(\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_L)$. As in [15], [16], SleepTransformer is trained to minimize the cross-entropy loss over the sequence:

$$\mathcal{L} = -\frac{1}{L} \sum\nolimits_{i=1}^{L} \mathbf{y}_i \log(\hat{\mathbf{y}}_i). \qquad (17)$$

V. INTERPRETABILITY AND CONFIDENCE QUANTIFICATION

A. Interpretability via self-attention

Self-attention (cf. Figure 1 (a)) learns a representation by relating the input elements at different positions in the input sequence. From the given query $\mathbf{Q}$, the machine learns the relation between the query and keys $\mathbf{K}$ to compute attention scores, and multiply the attention scores to the values $\mathbf{V}$. Finally, the sum of attended values composes the semantics of the given query. The attention scores can be leveraged to interpret the model. We propose two different visualizations for interpretation: (1) EEG attention heat map that shows where in the input EEG signal the model pays more attention to, and (2) epoch influence as a bar chart which qualifies the contribution of neighboring epochs to predicting the sleep stage of a target epoch in the input sequence.

EEG attention heat map. To understand the behavior of the model, it is important to know what parts of the EEG input the model pays more attention to. To this end, we sum the attention scores from each attention head of *EpochTransformer* for visualization. Attention score $\mathbf{A}$ of a single attention head is given as:

$$\mathbf{A} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\mathsf{T}}{\sqrt{d}}\right), \qquad (18)$$

where $\mathbf{A} \in \mathbb{R}^{l \times l}$. The element $a_{i,j}$ at i-th row and j-th column indicates how much the input at index j attributes to the representation at index i. The matrix $\mathbf{A}$ is, therefore, summed in the first dimension, followed by normalization to the range $[0, 1]$ to obtain a score vector whose i-th element indicates how much the input at index i attributes to the representations at all other indices.

A potential pitfall of the above-mentioned heat map is that summing the attention scores across the attention heads may dismiss attention structures of the attention heads. As an alternative, to further gain insight into the representation learned by the network, we use the attention score matrices to transform a time-frequency input and obtain a time-frequency output right after the last *EpochTransformer*, omitting all other non-linear operations. Inverse short-time Fourier Transform (ISTFT) is then applied the time-frequency output to construct the raw EEG signal which can be visualized to exhibit the features learned by the network. Of note, we use the original phase of the time-frequency input for this construction.

Epoch influence bar chart. To further shed light on the behavior of the model, it is equally important to know which neighboring epochs the model pays more attention to while scoring the target epoch in the input sequence. Given the attention score matrix $\mathbf{A}$ of a *SequenceTransformer*, the element $a_{i,j}$ at i-th row and j-th column indicates how much the epoch j in the input sequence is attributing to the representation of the target epoch i. We argue that it closely resembles the way a clinician performs manual scoring. Specifically, when the target epoch does not show much evidence of sleep-relevant features, the neighboring epochs in the context will be attended to, providing evidence in support of the scoring [5].

B. Entropy-based confidence quantification

In a general multi-class classification problem, a deep neural network outputs a vector whose elements are probabilities, one for each target class of interest. For the 5-stage sleep staging we are dealing with, an output $\hat{\mathbf{y}}$ from SleepTransformer consists of C probability values ($C = 5$ in this case) corresponding to C sleep stages. Typically, the sleep stage with respect to the maximum probability is considered the network's prediction. However, the predicted discrete label does not tells us how much the network is confident about its decision whereas the multi-class probability distribution $\hat{\mathbf{y}}$ is too complex.

In fact, the multi-class probability distribution over the sleep stages encoded in $\hat{\mathbf{y}}$ can provide a more refined measure of confidence in the network prediction. In one extreme, when $\hat{\mathbf{y}}$ assigns probability 1 to one class and probability 0 to the remaining classes, we expect the network to be very confident in its decision. In the other extreme, when the distribution is flat, i.e. all elements in $\hat{\mathbf{y}}$ are equal, the network has no confidence in its decision. All other distributions indicate varying levels of confidence between these two extremes. The entropy of the discrete probability distribution, an information-theoretic measure of uncertainty [47], appears to be a natural way to measure the network's intrinsic uncertainty. In turn, the network's confidence can be quantified as a concrete number. To this end, we propose to use normalized entropy:

$$H(\hat{\mathbf{y}}) = -\sum_{c=1}^{C} \hat{y}_c \frac{\log(\hat{y}_c)}{\log C}, \qquad (19)$$

to normalize the range of the uncertainty to $[0, 1]$, assuming $0 \times \log 0 = 0$. In turn, the network confidence is quantified as

$$Conf(\hat{\mathbf{y}}) = 1 - H(\hat{\mathbf{y}}). \qquad (20)$$

For 5-stage classification, $H(\hat{\mathbf{y}}) = 1$ and $Conf(\hat{\mathbf{y}}) = 0$ when $\hat{\mathbf{y}} = (\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$. $H(\hat{\mathbf{y}}) = 0$ and $Conf(\hat{\mathbf{y}}) = 1$ when $\hat{\mathbf{y}}$ contains exactly one probability 1, for example $\hat{\mathbf{y}} = (0, 1, 0, 0, 0)$. All other possible values of $\hat{\mathbf{y}}$ will result in $0 < Conf(\hat{\mathbf{y}}) < 1$.

Given the estimated confidence, we envision that a low-confidence epoch can be deferred for further manual verification and correction by human experts. The filtering can be accomplished via either thresholding the confidences with a predefined threshold or simply selecting a certain percentage of epochs with lowest confidences.

VI. EXPERIMENTS

A. Extraction of time-frequency image

As described, SleepTransformer ingests time-frequency images as input. To extract a time-frequency image, a 30-second EEG epoch sampled at 100 Hz was decomposed into two-second frames with 50% overlap, multiplied with a Hamming window, and transformed to the frequency domain using a 256-point short-time Fourier Transform (STFT). This procedure resulted in an image $\mathbf{S} \in \mathbb{R}^{T \times F}$ with $T = 29$ time frames and $F = 128$ frequency bins. Of note, we excluded the 0-th frequency bin to keep $F = 128$ which is a multiple of the number of attention heads in the epoch transformer in Section

IV-A. The amplitude spectrum was then log-transformed. The time-frequency images extracted from a database were normalized to zero mean and unit variance along the frequency dimension given the normalization parameters computed using the training data.

B. Parameters

We experimented with different values $\{11, 21, 31, 41, 51\}$ for the input sequence length L and 21 was found to be best. This result confirms the finding reported in other works like [13], [15]. Thus, we fixed $L = 21$ for further experiments here. The network was designed to have $N_E = 4$ *EpochTransformer*s for intra-epoch processing and $N_S = 4$ *SequenceTransformer*s for inter-epoch processing. In a transformer, the number of attention heads was fixed to $H = 8$ and the number of hidden units of a feed forward layer was fixed to $d_{FF} = 1024$. The FC layers of the network was also of size 1024. A common dropout rate of 0.1 was applied to the transformer, including the self-attention layers and the feed forward layers, as well as the FC layers.

The experiments were conducted on the two databases SHHS and SleepEDF-78 individually. We carried out 10-fold cross-validation on the SleepEDF-78 database as in prior works [16], [25], [31]. In each iteration, seven subjects were left out from the training subjects as the validation set. For the large-scale database, SHHS, we randomly split the subject into 70% for training and 30% for testing as in [16], [22]. 100 subjects were left out from the training set as the validation set. Of note, following [22], those recordings without all five sleep stages were excluded. The network was trained using Adam optimizer [50] with a learning rate of 10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-7}$. A minibatch size of 32 was used for training. The model was validated on the validation set every 100 training steps. Early stopping was applied and the training was stopped after 200 validation steps without improvement on the validation data. Particularly, for SHHS, the model was trained for at least 5000 validation steps before early stopping was activated.

C. Experimental results

1) Sleep staging performance: Table I shows the performance on the experimental databases obtained by SleepTransformer in comparison to those reported in previous works. Accuracy, Cohen's kappa (κ) [51], macro F1-score (MF1) [52], average sensitivity, and average specificity are used as overall performance metrics while class-specific performance is assessed using class-wise MF1.

On the large-scale database, SHHS, SleepTransformer achieves an overall accuracy of 87.7% and κ of 0.828. On the one hand, compared to the seminal sequence-to-sequence counterpart, SeqSleepNet [15], SleepTransformer results in an improvement of 1.2% absolute in terms of accuracy and 0.017 in terms of κ. This result suggests that the transformer backbone is more advantageous than the recurrent backbone used in SeqSleepNet. On the other hand, SleepTransformer's performance is on par with those of the existing state-of-the-art XSleepNets [16] (i.e. XSleepNet1 and XSleepNet2 in

Table I: Performance comparison between SleepTransformer and previous works on the experimental databases. The superscript * indicates the model was initialized by the model pretrained on the SHHS database. † The results are not directly comparable either due to the differences in the data split and the channels used (Olesen *et al.* [44] and U-Sleep [46]) or due to the use of a small subset of healthy subjects (Eldele *et al.* [48]).

Database	System	Overall metrics					Class-wise MF1				
		Acc.	κ	MF1	Sens.	Spec.	W	N1	N2	N3	REM
SHHS	**SleepTransformer**	**87.7**	**0.828**	80.1	78.7	**96.5**	92.2	46.1	88.3	**85.2**	88.6
	XSleepNet2	87.6	0.826	80.7	79.7	**96.5**	92.0	49.9	88.3	85.0	88.2
	XSleepNet1	87.5	0.826	**81.0**	**80.4**	**96.5**	91.6	51.4	**88.5**	85.0	88.4
	U-Sleep† [46]	–	–	80.0	–	–	93.0	51.0	87.0	76.0	**92.0**
	Olesen *et al.*† [44]	87.1	0.816	78.8	77.7	96.3	**94.1**	47.8	87.9	74.3	89.9
	SeqSleepNet [15]	86.5	0.811	78.5	76.9	96.1	91.4	43.3	87.4	82.9	87.3
	FCNN+RNN	86.7	0.813	79.5	78.1	96.2	91.1	48.7	88.0	82.6	87.1
	CNN [22]	86.8	0.810	78.5	–	95.0	–	–	–	–	–
	IITNet [24]	86.7	0.810	79.8	–	–	–	–	–	–	–
	AttnSleep† [48]	84.2	0.78	75.3	–	–	86.7	33.2	87.1	87.1	82.1
SleepEDF-78	**SleepTransformer***	**84.9**	**0.789**	78.8	**78.2**	**95.9**	**93.5**	48.5	**86.5**	80.9	84.6
	SleepTransformer	81.4	0.743	74.3	74.5	95.0	91.7	40.4	84.3	77.9	77.2
	XSleepNet2	84.0	0.778	77.9	77.5	95.7	93.3	49.9	86.0	78.7	81.8
	XSleepNet1	83.6	0.773	77.8	77.7	95.7	92.6	50.2	85.9	79.2	81.3
	SeqSleepNet [15]	82.6	0.760	76.4	76.3	95.4	92.2	47.8	84.9	77.2	79.9
	FCNN+RNN	82.8	0.761	76.6	75.9	95.4	92.5	47.3	85.0	79.2	78.9
	Zhu *et al.* [49]	82.8	–	77.8	–	–	90.3	47.1	86.0	82.1	83.2
	U-Time [45]	–	–	76.0	–	–	–	–	–	–	–
	U-Sleep† [46]	–	–	**79.0**	–	–	93.0	**57.0**	86.0	71.0	**88.0**
	CNN-LSTM [45]	–	–	73.0	–	–	–	–	–	–	–
	AttnSleep [48]	81.3	0.74	75.1	–	–	92.0	42.0	85.0	82.1	74.1
	SleepEEGNet [25]	80.0	0.730	73.6	–	–	–	–	–	–	–

Table I). Although the differences in their accuracy and κ are small, it is considerable given the fact that SleepTransformer only uses the time-frequency input (i.e. single view), has a smaller model footprint, and is computationally cheaper (see Table III). The class-wise MF1s further unravel their opposite patterns. SleepTransformer seems to favor the major sleep stages, i.e. Wake and REM, over the under-present stage N1. On the contrary, the class-wise MF1s are less skewed in case of XSleepNets, resulting in a better overall MF1 than SleepTransformer.

On the smaller database (SleepEDF-78) SleepTransformer's performance seems to be inferior to other competitors. However, this result does not correctly reflect its modelling capacity. In fact, we observed that SleepTransformer overfitted this database quite easily and thus requires a larger amount of data for training. Motivated by the sleep transfer learning approach in [28], we utilized the model trained on SHHS to initialize the network instead of random initialization. This simple trick significantly boosted the performance, improving the overall absolute accuracy, κ, and MF1 by 3.5%, 0.046, and 4.5%. With the achieved overall accuracy of 84.9%, κ of 0.789, and MF1 of 78.8, SleepTransformer outperforms all the previous works evaluated on SleepEDF-78, except for U-Sleep [46] which utilized an ensemble of models separately trained on EEG and EOG. Regarding the class-wise performance, similar conclusions as for SHHS can be drawn.

2) Confidence estimation: Let $\bar{\mathcal{A}}$ denote the set of low-confidence epochs (that need further manual verification and/or correction) whose confidences are below a predefined threshold. Alternatively, $\bar{\mathcal{A}}$ can also be selected as the set of epochs with lowest confidences, e.g. the set containing 20% of epochs with lowest confidences. In addition, let $\mathcal{A}$ denote the set of the remaining epochs, which is the complement of $\bar{\mathcal{A}}$.
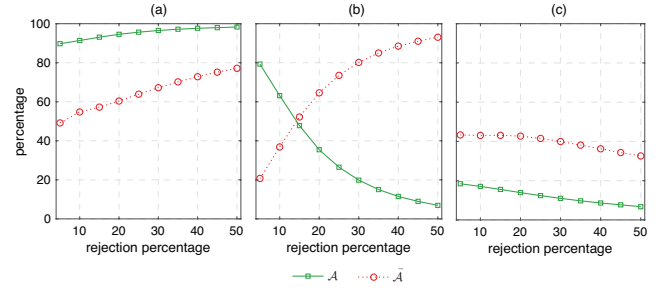

Figure 3: SleepEDF-78: **(a)** Accuracies of $\mathcal{A}$ and $\bar{\mathcal{A}}$; **(b)** percentages of misclassified epochs in $\mathcal{A}$ and $\bar{\mathcal{A}}$ out of all misclassified epochs; **(c)** percentages of transitioning epochs out of $\mathcal{A}$ and $\bar{\mathcal{A}}$.

Reasonably, the confidence metric is only useful and meaningful if it could help filter out misclassified epochs for further manual verification/correction. As shown in Figure 3 (a), the classification accuracy of $\bar{\mathcal{A}}$ remains much lower than $\mathcal{A}$ regardless of the size of $\bar{\mathcal{A}}$. For example, when $\bar{\mathcal{A}}$ constitutes 20% of all the epochs, it has an accuracy around 60%, meaning 40% of its epochs are misclassified ones. And these misclassified epochs constitute about 65% of all misclassified epochs as shown in Figure 3 (b). The percentage of misclassified epochs in $\bar{\mathcal{A}}$ increases sharply when it grows larger. When $\bar{\mathcal{A}}$ constitutes of 50% of all the epochs, we can segregate more than 90% of all misclassified epochs. All of this implies that the misclassified epochs are often associated with low confidences. For sleep particularly, the transitioning epochs (whose sleep stages are different from those of its preceding and/or succeeding neighbors [53]) are usually difficult ones as human scorers also tend to disagree and machine scoring systems often make mistakes. Interestingly, the percentage of

Figure 4: Confusion matrices of SleepEDF-78. A confidence threshold of 0.5 was used to separate the epochs into $\mathcal{A}$ and $\bar{\mathcal{A}}$.
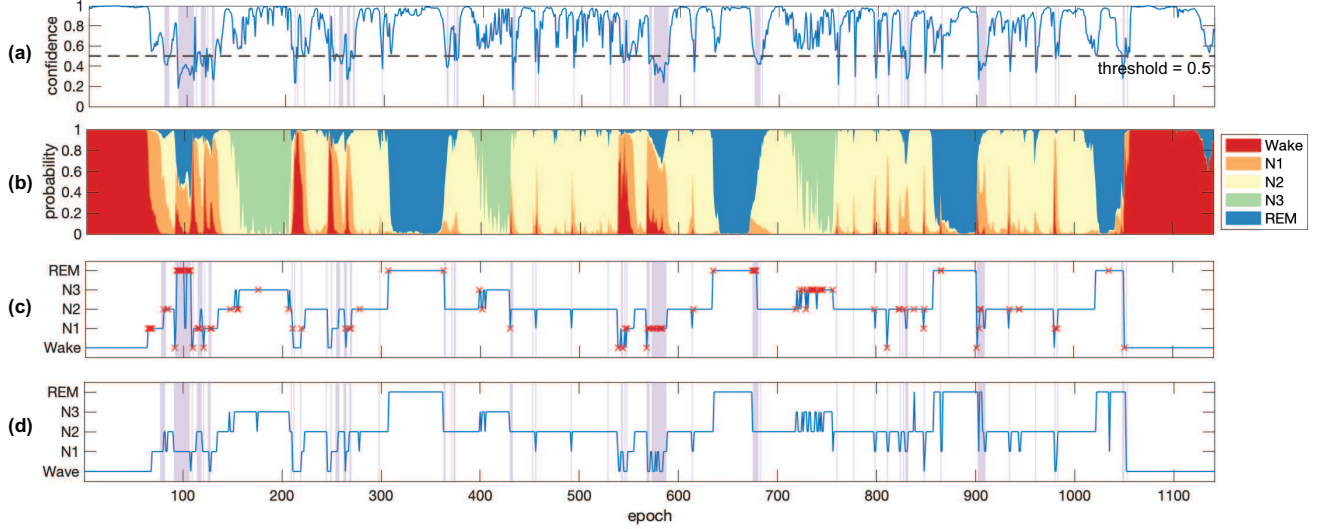

Figure 5: Visualization the estimated confidence for Subject 2 of SleepEDF-78. **(a)** The quantified confidence; **(b)** the probability output; **(c)** the output hypnogram where $\times$ indicates the misclassified epochs; and **(d)** the ground-truth hypnogram. The shaded regions correspond to the epochs with their confidence below the threshold.

transitioning epochs in $\bar{\mathcal{A}}$ is always considerably larger than in $\mathcal{A}$, as shown in Figure 3 (c). Moreover, the confusion matrices in Figure 4 further reveal that the majority of the epochs in $\bar{\mathcal{A}}$ are N1, leaving just a small portion of N1 epochs in $\mathcal{A}$. This is not a surprise since N1 is, in general, hard to be correctly recognized due to its under-presence and strong resemblance to Wake and N2.

We further showcase the above findings in Figure 5 where we portray the quantified confidence alongside the ground-truth hypnogram, the output hypnogram, and the multi-class probability output for a subject in SleepEDF-78 (i.e. Subject 2). In the figure, the confidence is thresholded by 0.5. It can be seen that, often, the low-confidence and transitioning epochs are misclassified. With the threshold value 0.5, roughly 90% (in case of SleepEDF-78) or more (in case of SHHS) of epochs per night have their confidences above the threshold, as further shown in Figure 6, and would not need to undergo manual verification.

3) Attention score visualization for interpretation: In light of the proposed approach for explainability in Section V-A, in Figures 7 and 8, we attempted to visualize the attention scores for two different input sequences at both the epoch and sequence level. For simplicity and clarity, we used the model with sequence length $L = 11$ and the attention scores of the
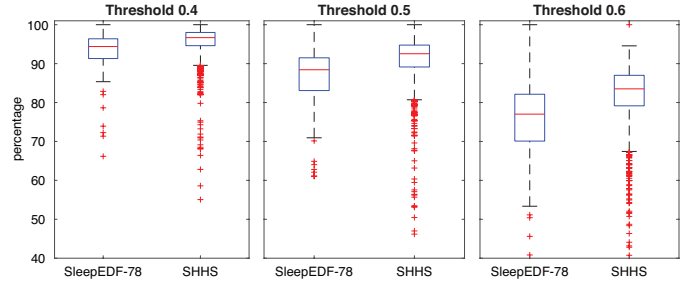

Figure 6: Percentages of epochs per night with confidence above a confidence threshold of 0.4, 0.5, and 0.6.

last *EpochTransformer* and *SequenceTransformer* for the EEG heat map and the sequence-level attention, respectively. We also included the predicted labels, the ground-truth labels, and the estimated confidences of the epochs in the sequences to aid interpretability. We additionally show enlarged versions of the raw EEG signals in Appendix for further details.

At the epoch-level, the heat map on the EEG signals in the figures suggests that the model indeed attended more to sleep related features. For instance, the K-complexes present in epochs 2 and 4 in Figure 7(b). This type of micro-event is notable in the sleep stage N2. The attention is more scattering
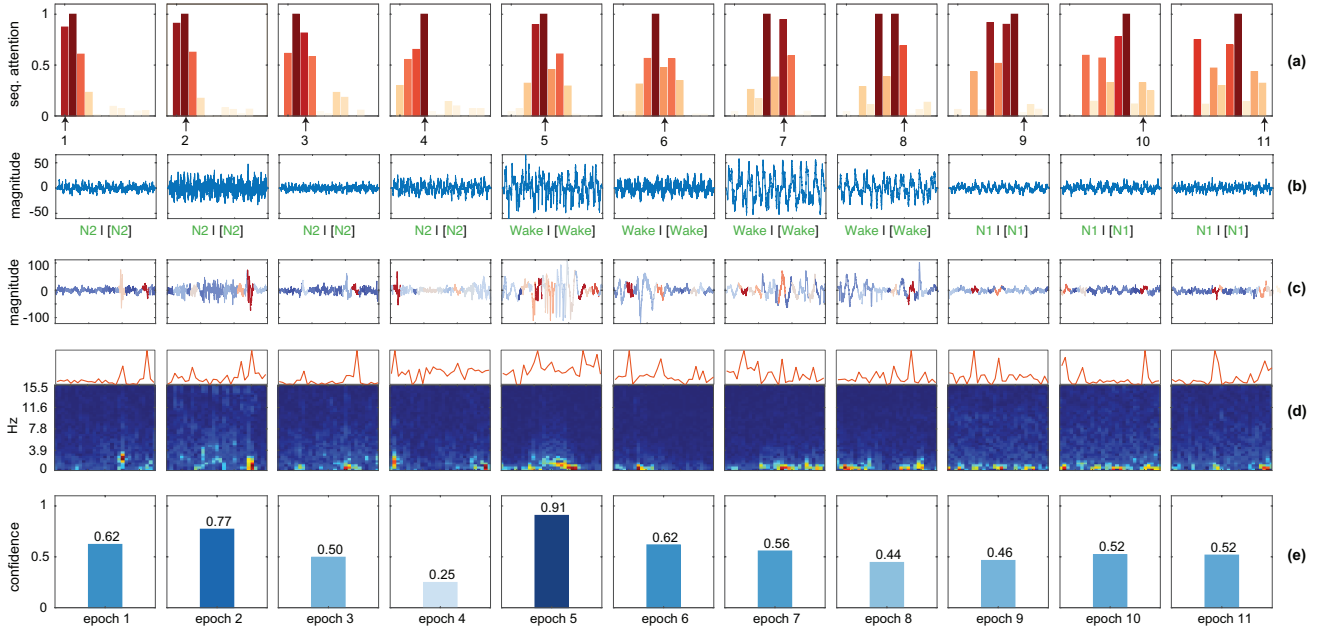
Figure 7: Self-attention visualization for interpretation of an input sequence with $L = 11$ stemming from SleepEDF-78 (Subject 2). The sequence involves two cross-stage transitions: (i) N2→Wake around epochs 4 and 5, and (ii) Wake→N1 around epochs 8 and 9. **(a)** The distribution of sequence-level attention scores where an arrow indicates the position of the current epoch in the distribution of the attention score. **(b)** The EEG features learned for epoch representation (constructed via ISTFT as described in Section V-A). For pair of labels, the output label is on the left and the ground-truth label is inside the brackets on the right. **(c)** The epoch-level attention scores represented by heat map on the raw EEG signals. **(d)** The spectrogram inputs and their epoch attention scores (i.e. the red curves) distributed over spectral columns. **(e)** The estimated confidences.
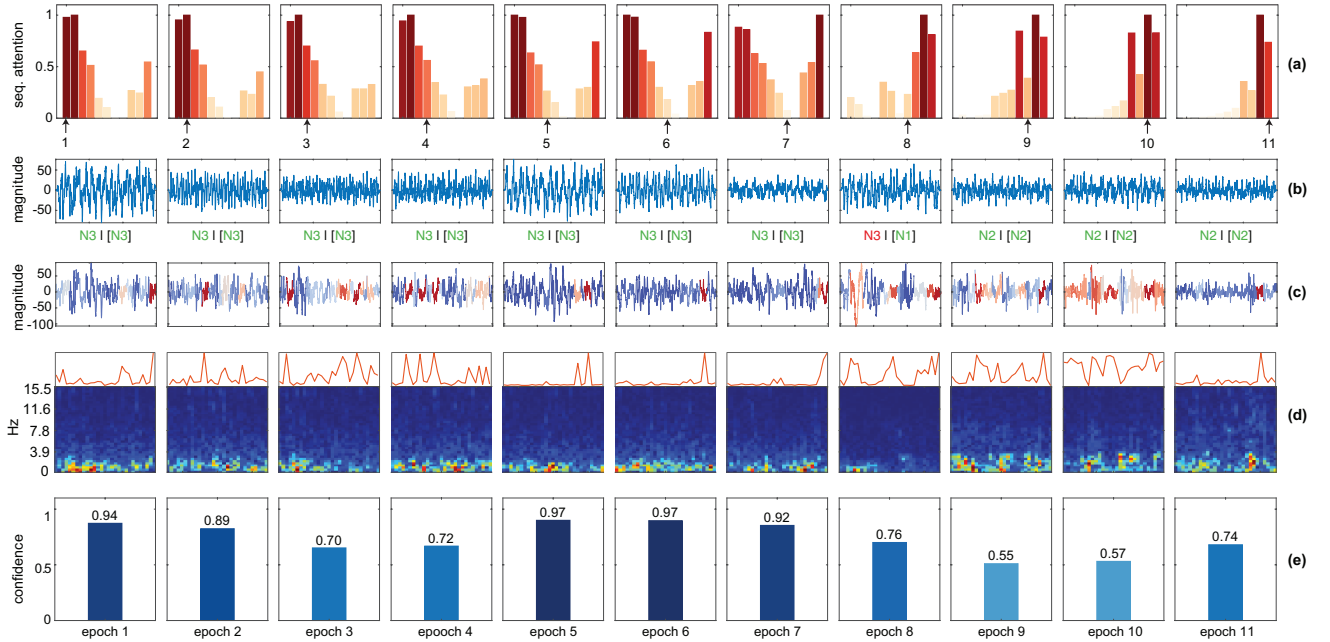

Figure 8: Self-attention visualization for interpretation of another input sequence with $L = 11$ stemming from SleepEDF-78 (Subject 2). The sequence involves two cross-stage transitions: (i) N3→N1 around epochs 7 and 8, and (ii) N1→N2 around epochs 8 and 9. **(a)** The distribution of sequence-level attention scores where an arrow indicates the position of the current epoch in the distribution of the attention score. **(b)** The EEG features learned for epoch representation (constructed via ISTFT as described in Section V-A). For pair of labels, the output label is on the left and the ground-truth label is inside the brackets on the right. **(c)** The epoch-level attention scores represented by heat map on the raw EEG signals. **(d)** The spectrogram inputs and their epoch attention scores (i.e. the red curves) distributed over spectral columns. **(e)** The estimated confidences.

Table II: Variation of SleepTransformer's performance on SHHS with different values of N_E, the number of transformer encoder block in the epoch encoder, and N_S, the number of transformer encoder block in the sequence encoder.

N_E	N_S	Overall metrics					Class-wise MF1				
		Acc.	κ	MF1	Sens.	Spec.	W	N1	N2	N3	REM
1	4	87.5	0.824	79.3	77.8	96.4	91.9	42.8	88.2	85.1	88.2
2	4	**87.7**	0.827	79.5	77.9	**96.5**	92.3	42.9	**88.3**	85.2	88.6
3	4	**87.7**	**0.828**	79.8	78.4	**96.5**	**92.4**	44.2	**88.3**	85.1	**88.9**
4	4	**87.7**	**0.828**	80.1	**78.7**	96.5	92.2	**46.1**	**88.3**	85.2	88.6
4	3	87.4	0.823	79.1	77.7	96.4	92.1	42.3	87.9	85.1	88.1
4	2	87.6	0.826	79.6	78.5	**96.5**	**92.4**	44.0	88.2	85.2	88.3
4	1	87.5	0.825	79.1	77.9	**96.5**	92.1	41.9	**88.3**	**85.5**	87.7

for N3 stage in epochs 1, 2, and 3 in Figure 7(b) given the omnipresence of Delta waves. Furthermore, the constructed EEGs resemble Alpha waves in N1 stage (epochs 9, 10, and 11 in Figure 7(c)), high-amplitude neural activities in Wake stage (epochs 5, 7, and 8 in Figure 7(c)), or Delta waves in N3 stage (epochs 1-6 in Figure 8(c)). These constructed EEGs exhibit distinguishable frequency distributions as shown in Figures A.1 and A.2.

At the sequence level, the attention scores act as the weights used to collectively combine features in different epochs in the sequence to classify a target epoch, featuring the benefit of sequence-to-sequence sleep scoring with self-attention. In the first example in Figure 7, apparently, those epochs on the left of the sequence containing useful features for recognizing the stage N2 are associated with strong weights whereas other epochs containing less relevant features are associated with smaller weights. Similarly, for Wake epochs, the attention scores concentrate more around the epochs in the middle. On the contrary, the attention scores of the N1 epochs on the right disperse due to the fact that the stage N1 shares similar features to both Wake and N2. In other words, even though the EEG signal of a target epoch does not show much useful features for recognizing the sleep stage N1, the model is still able to recognize it by leveraging the relevant features appearing in the context via the sequence-level attention.

In the second example in Figure 8, we note that, for both N2 and N3, considerably greater attention weights are placed on the epochs far away from the transitioning boundary which most likely convey more reliable features. Thus, the network is taking advantage of long-term structure in sleep to recognize those epochs close to the transitioning boundary. This particular example, on the other hand, demonstrates an interesting case at the 8-th epoch where the network misclassified, predicting N3 against the ground-truth N1. We argue that the ground-truth N1 in this case highlights the well-known subjectivity of human scoring since the transient transition N3→N1→N2 seems to be counter-intuitive. This epoch seems to contain mixed information of both N3 (delta activity as shown in the time-frequency image for the 8-th epoch in Figure A.2) and N2 (the big K-complex in the raw EEG of the 8-th epoch in Figure 8 (c)). Thus, it is more likely to be a N3→N2 transitioning epoch.

We argue that this visualization resembles the way human scoring is done, and therefore, would facilitate manual verification and correction of low-confidence epochs and provide a gateway for practitioners to interact with the model.

D. Discussion

For models using the transformer backbone like Sleep-Transformer, choosing an appropriate number of transformers can be crucial. To investigate the influence of the number of *EpochTransformer* N_E and the number of *SequenceTransformer* N_S, we repeated the experiments with N_E fixed to 4 and N_S varied in $\{1, 2, 3, 4\}$. After that, we repeated the experiments with N_S fixed to 4 and N_E varied in $\{1, 2, 3, 4\}$. Using SHHS for this investigation, the overall performance obtained with different values of N_E and N_S are shown in Table II. These results suggest the modest impact of both N_E and N_S on the overall accuracy. However, the class-wise MF1s suggest a large number of transformers is important to improve the performance (mostly the sensitivity, i.e. the true positive rate) on the under-present stage N1 which, in turn, improves the average MF1.

Concerning the model size and computational cost, even with $N_E = 4$ and $N_S = 4$, SleepTransformer has moderate model size and modest computational overhead as contrasted with some existing models (whose relevant information was previously reported) in Table III. In particular, compared to our recent developed model, XSleepNet [16], SleepTransformer's model size is just two third of it of XSleepNet [16] while it is 2.7 times faster to train. It is even faster than SeqSleepNet, the compact model proposed in our previous work [15], most likely because SleepTransformer is recurrent-free. Of note, we measured the training time of the models in the table using a common DGX-2 machine with NVIDIA Tesla V100 graphic card and Intel Xeon Platinum 8186 CPU, 2.7 GHz.

Future work can further address the following limitations of this work. First, the entropy-based uncertainty quantification proposed here is not only applicable for SleepTransformer but also any sleep-scoring model with probability output, such as SeqSleepNet [15] or XSleepNet [16]. Furthermore, alternative to entropy, the likelihood of the most likely class (i.e. the maximum probability in an output probability distribution) could also be used to obtain a measure of uncertainty [54]. Different from entropy, which depends on the entire distribution over classes to measure an overall uncertainty in the predictions, this measure of uncertainty is not affected by probabilities of other classes and may yield a more precise estimation. Lastly, we are only dealing with *knowledge uncertainty* (i.e. uncertainty in the model's predictions), leaving *data uncertainty* [36] (i.e. uncertainty arises due to the complexity, multi-modality and noise in the data) open for future works. Ideally, a method that could take into account both data

Table III: Model size and training time per 1000 training steps.

Model	#parameters	training time (s) per 1000 steps
SleepTransformer	3.70×10^6	308
SeqSleepNet [15]	1.64×10^5	379
U-Time [45]	1.10×10^6	–
U-Sleep [46]	3.10×10^6	–
XSleepNet [16]	5.74×10^6	828
DeepSleepNet [13]	2.30×10^7	–

uncertainty and model uncertainty, could be agnostic to the network architectures, and could be applied to already trained models would be much more useful. Second, our visualization attempt in Figure V-A is not necessarily the best and the only way to interpret the model's decisions. Further creativity and interaction with experts will be needed to leverage the information encoded in the attention scores before it can be embedded in daily sleep practise. Third, we employed the original transformer proposed in the seminal work of [40] as the backbone of SleepTransformer, more advanced variants of transformer can be further explored.

VII. Conclusions

We proposed SleepTransformer, a sequence-to-sequence sleep staging model relying solely on the transformer network. We showed that SleepTransformer performed comparably to state-of-the-art models on both SHHS, a large-scale database, and SleepEDF-78, a relative small database. We leveraged the attention scores of the transformer's self-attention module for interpretability. At the epoch level, the attention scores was applied to the EEG input as a heat map to highlight sleep-relevant features the model attended to. At the sequence level, the attention scores were interpreted as the contribution of different neighboring epochs to the recognition of a target epoch in the input sequence. We also used entropy of the multi-class probability distribution output to quantify uncertainty of the model's decisions as a concrete number which was shown to align well with the model's mistakes and successes.

References

[1] Institute of Medicine, *Sleep Disorders and Sleep Deprivation: An Unmet Public Health Problem*, Washington DC: The National Academies Press, 2006.

[2] O. Asan *et al.*, "Artificial intelligence and human trust in healthcare: Focus on clinicians," *Journal of Medical Internet Research*, vol. 22, no. 6, pp. e15154, 2020.

[3] M. Nagendran *et al.*, "Artificial intelligence versus clinicians: systematic review of design, reporting standards, and claims of deep learning studies," *BMJ*, vol. 368, pp. m689, 2020.

[4] G. Eysenbach and Q. Zeng, "Artificial intelligence and human trust in healthcare: Focus on clinicians," *Journal of Medical Internet Research*, vol. 22, no. 6, pp. e15154, 2020.

[5] C. Iber *et al.*, "The AASM manual for the scoring of sleep and associated events: Rules, terminology and technical specifications," *American Academy of Sleep Medicine*, 2007.

[6] A. Malhotra *et al.*, "Performance of an automated polysomnography scoring system versus computer-assisted manual scoring," *SLEEP*, vol. 36, no. 4, pp. 573–582, 2013.

[7] K. B. Mikkelsen *et al.*, "Machine-learning-derived sleep–wake staging from around-the-ear electroencephalogram outperforms manual scoring and actigraphy," *J. Sleep Res.*, vol. 28, no. 2, pp. e12786, 2019.

[8] K. B. Mikkelsen *et al.*, "Sleep monitoring using ear-centered setups: Investigating the influence from electrode configurations," *IEEE Transactions on Biomedical Engineering*, pp. 1–1, 2021.

[9] C. O'Reilly *et al.*, "Montreal archive of sleep studies: An open-access resource for instrument benchmarking & exploratory research," *J. Sleep Res.*, pp. 628–635, 2014.

[10] G. Q. Zhang *et al.*, "The national sleep research resource: towards a sleep data commons," *J Am Med Inform Assoc.*, vol. 25, no. 10, pp. 1351–1358, 2018.

[11] S. F. Quan *et al.*, "The sleep heart health study: design, rationale, and methods," *Sleep*, vol. 20, no. 12, pp. 1077–1085, 1997.

[12] J. B. Stephansen *et al.*, "Neural network analysis of sleep stages enables efficient diagnosis of narcolepsy," *Nature Communications*, vol. 9, no. 1, pp. 5229, 2018.

[13] A. Supratak *et al.*, "DeepSleepNet: A model for automatic sleep stage scoring based on raw single-channel EEG," *IEEE Trans. Neural Syst. Rehabilitation Eng.*, vol. 25, no. 11, pp. 1998–2008, 2017.

[14] S. Biswal *et al.*, "Expert-level sleep scoring with deep neural networks," *J Am Med Inform Assoc.*, vol. 25, no. 12, pp. 1643–1650, 2018.

[15] H. Phan *et al.*, "SeqSleepNet: end-to-end hierarchical recurrent neural network for sequence-to-sequence automatic sleep staging," *IEEE Trans. Neural Syst. Rehabilitation Eng.*, vol. 27, no. 3, pp. 400–410, 2019.

[16] H. Phan *et al.*, "XSleepNet: Multi-view sequential model for automatic sleep staging," *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, 2021.

[17] S. Chambon *et al.*, "A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series," *IEEE Trans. Neural Syst. Rehabilitation Eng.*, vol. 26, no. 4, pp. 758–769, 2018.

[18] H. Dong *et al.*, "Mixed neural network approach for temporal sleep stage classification," *IEEE Trans. Neural Syst. Rehabilitation Eng.*, vol. 26, no. 2, pp. 324–333, 2018.

[19] O. Tsinalis *et al.*, "Automatic sleep stage scoring with single-channel EEG using convolutional neural networks," *arXiv preprint arXiv:1610.01683*, 2016.

[20] H. Sun *et al.*, "Large-scale automated sleep staging," *SLEEP*, vol. 40, no. 10, pp. zsx139, 2017.

[21] H. Phan *et al.*, "DNN filter bank improves 1-max pooling CNN for single-channel EEG automatic sleep stage classification," in *Proc. EMBC*, 2018, pp. 453–456.

[22] A. Sors *et al.*, "A convolutional neural network for sleep stage scoring from raw single-channel eeg," *Biomed Signal Process Control*, vol. 42, pp. 107–114, 2018.

[23] H. Phan *et al.*, "Automatic sleep stage classification using single-channel eeg: Learning sequential features with attention-based recurrent neural networks," in *Proc. EMBC*, 2018, pp. 1452–1455.

[24] H. Seo *et al.*, "Intra- and inter-epoch temporal context network (iitnet) using sub-epoch features for automatic sleep scoring on raw single-channel eeg," *Biomed Signal Process Control*, 2020.

[25] S. Mousavi *et al.*, "SleepEEGNet: Automated sleep stage scoring with sequence to sequence deep learning approach," *PLoS One*, vol. 14, no. 5, pp. e0216456, 2019.

[26] A. Guillot and V. Thorey, "Robustsleepnet: Transfer learning for automated sleep staging at scale," *arXiv preprint arXiv:2101.02452*, 2021.

[27] J. Fan *et al.*, "EOGNET: a novel deep learning model for sleep stage classification based on single-channel eog signal," *Front. Neurosci.*, vol. 15, pp. 573194, 2021.

[28] H. Phan *et al.*, "Towards more accurate automatic sleep staging via deep transfer learning," *IEEE Trans. Biomed. Eng.*, vol. 68, no. 6, pp. 1787–1798, 2021.

[29] H. Phan *et al.*, "Deep transfer learning for single-channel automatic sleep staging with channel mismatch," in *Proc. EUSIPCO*, 2019.

[30] N. Banluesombatkul *et al.*, "MetaSleepLearner: A pilot study on fast adaptation of bio-signals-based sleep stage classifier to new individual subject using meta-learning," *IEEE Journal of Biomedical and Health Informatics (JBHI)*, vol. 25, no. 6, pp. 1949–1963, 2021.

[31] H. Phan *et al.*, "Personalized automatic sleep staging with single-night data: a pilot study with KL-divergence regularization," *Physiological Measurement*, vol. 41, no. 6, pp. 064004, 2020.

[32] K. Mikkelsen and M. De Vos, "Personalizing deep learning models for automatic sleep staging," *arXiv Preprint arXiv:1801.02645*, 2018.

[33] J. Amann *et al.*, "Explainability for artificial intelligence in healthcare: a multidisciplinary perspective," *BMC Medical Informatics and Decision Making*, vol. 20, pp. 310, 2020.

[34] A. Vilamala *et al.*, "Deep convolutional neural networks for interpretable analysis of EEG sleep stage scoring," in *Proc. MLSP*, 2017.

[35] T. Lee *et al.*, "Trier: Template-guided neural networks for robust and interpretable sleep stage identification from eeg recordings," *arXiv Preprint arXiv:2009.05407*, 2020.

[36] K. B. Mikkelsen *et al.*, "Predicting sleep classification performance without labels," in *Proc. EMBC*, 2020, pp. 645–648.

[37] A. Guillot *et al.*, "Dreem open datasets: Multi-scored sleep datasets to compare human and automated sleep staging," *IEEE Trans. Neural Systems and Rehabilitation Engineering (TNSRE)*, vol. 28, no. 9, pp. 1955–1965, 2020.

[38] H. Danker-Hopfe *et al.*, "Interrater reliability for sleep scoring according to the Rechtschaffen & Kales and the new AASM standard," *J. Sleep Res.*, vol. 18, pp. 74–84, 2009.

[39] T. Becker *et al.*, "Classification with a deferral option and low-trust filtering for automated seizure detection," *Sensors*, vol. 21, no. 4, pp. 1046, 2021.

[40] A. Vaswani *et al.*, "Attention is all you need," in *Proc. NIPS*, 2017, p. 5998–6008.

[41] J. A. Hobson, "A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects," *Electroencephalography and Clinical Neurophysiology*, vol. 26, no. 6, pp. 644, 1969.

[42] B. Kemp *et al.*, "Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the EEG," *IEEE. Trans. Biomed. Eng.*, vol. 47, no. 9, pp. 1185–1194, 2000.

[43] A. L. Goldberger *et al.*, "Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, pp. e215–e220, 2000.

[44] A. N. Olesen *et al.*, "Automatic sleep stage classification with deep residual networks in a mixed-cohort setting," *SLEEP*, vol. 44, no. 1, pp. zsaa161, 2021.

[45] M. Perslev *et al.*, "U-time: A fully convolutional network for time series segmentation applied to sleep staging," in *Proc. NeurIPS*, 2019, pp. 4417–4428.

[46] M. Perslev *et al.*, "U-Sleep: resilient high-frequency sleep staging," *npj Digital Medicine*, vol. 4, no. 72, 2021.

[47] T. M. Cover and J. A. Thomas, *Elements of information theory*, John Wiley & Sons, 2006.

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

[49] T. Zhu *et al.*, "Convolution- and attention-based neural network for automated sleep stage classification," *International Journal of Environment Research and Public Health*, vol. 7, pp. 4152, 2020.

[50] D. P. Kingma and J. L. Ba, "Adam: a method for stochastic optimization," in *Proc. ICLR*, 2015, number 1-13.

[51] M. L. McHugh, "Interrater reliability: the kappa statistic," *Biochemia Medica*, 2012.

[52] Y. Yang and X. Liu, "A re-examination of text categorization methods," in *Proc. SIGIR*, 1999, vol. 99, pp. 42–49.

[53] H. Phan *et al.*, "Joint classification and prediction CNN framework for automatic sleep stage classification," *IEEE. Trans. Biomed. Eng.*, vol. 66, no. 5, pp. 1285–1296, 2019.

[54] K. B. Mikkelsen *et al.*, "Accurate whole-night sleep monitoring with dry-contact ear-EEG," *Sci. Rep.*, vol. 9, no. 1, pp. 1–12, 2019.

APPENDIX A
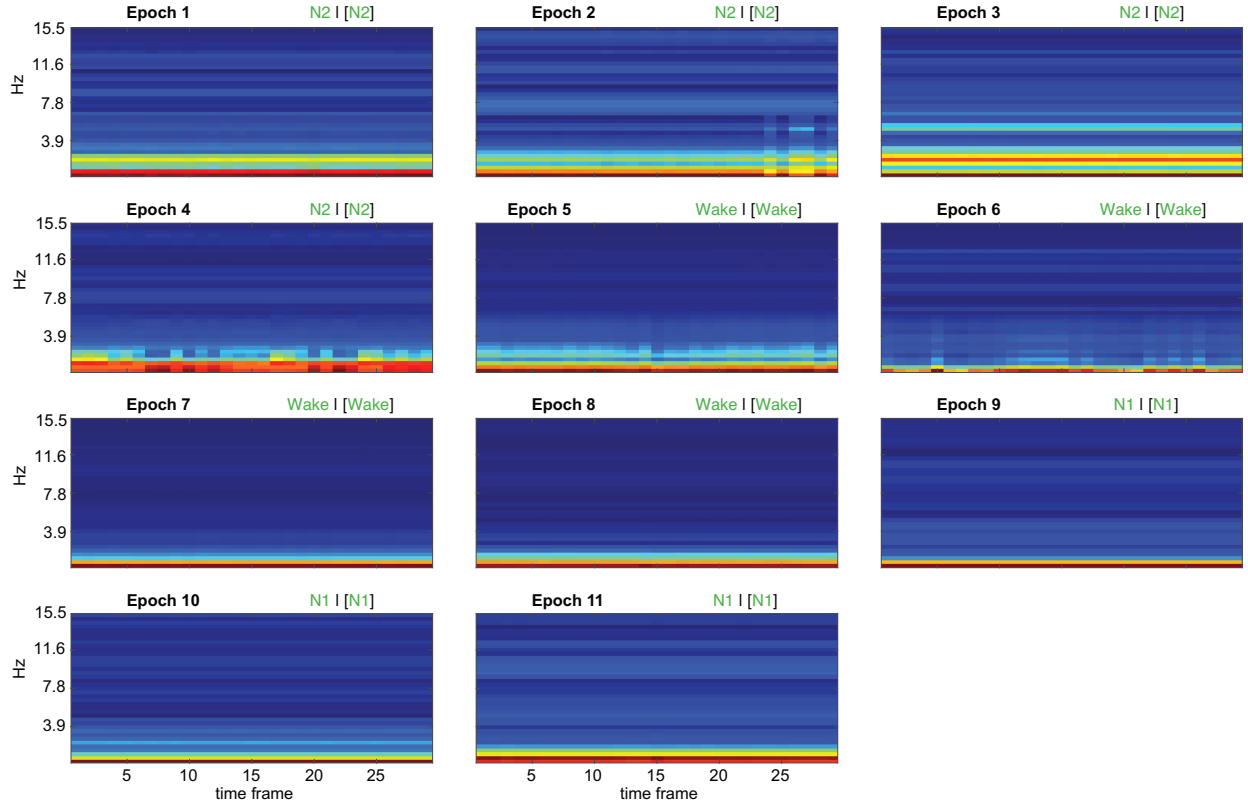## TIME-FREQUENCY REPRESENTATION CORRESPONDING TO THE CONSTRUCTED EEGS

Figure A.1: Time-frequency representation corresponding to the constructed EEG in Figure 7 (b). Of note, we only show frequency bins 1-40 corresponding to frequency range $(0, 15.5]$ Hz.
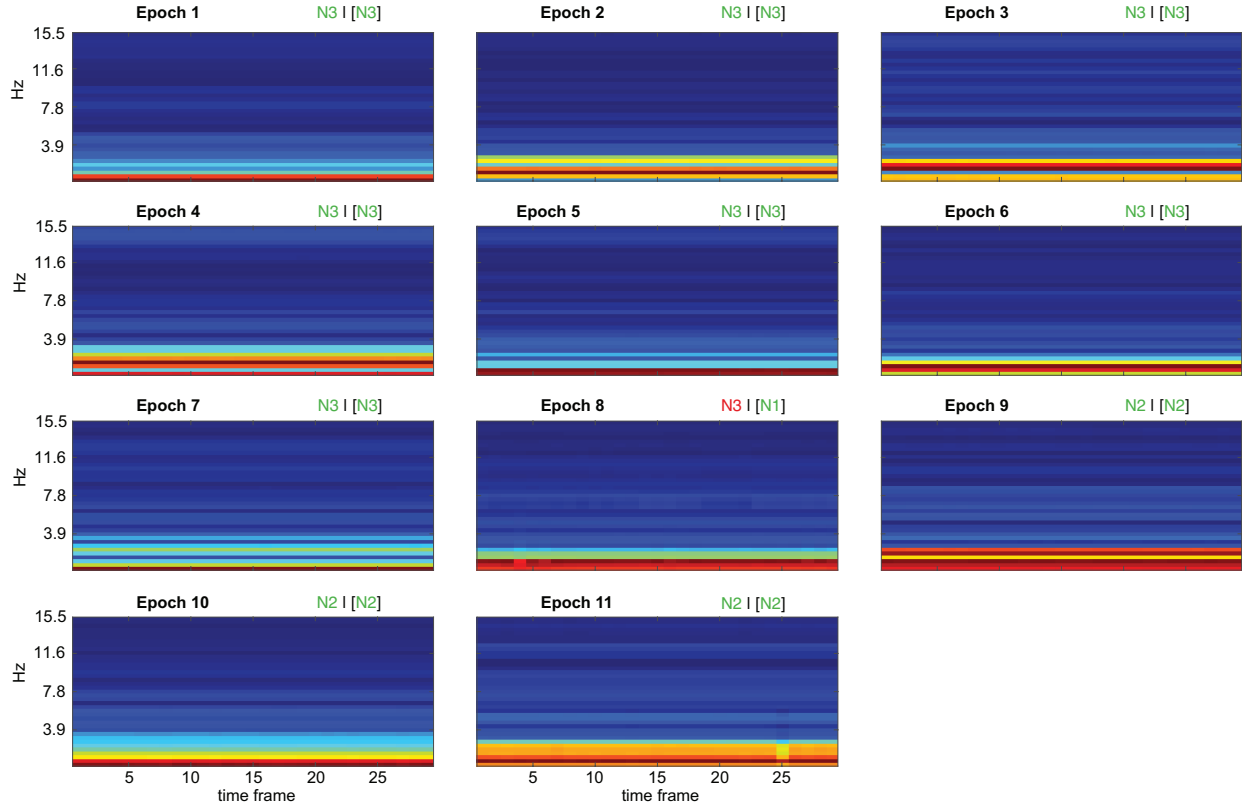

Figure A.2: Time-frequency representation corresponding to the constructed EEG in Figure 8 (b). Of note, we only show frequency bins 1-40 corresponding to frequency range $(0, 15.5]$ Hz.