Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

SleePyCo: Automatic sleep scoring with feature pyramid and contrastive learning

Seongju Lee [a], Yeonguk Yu [a], Seunghyeok Back [a], Hogeon Seo [b,c], Kyoobin Lee [a,*]

[a] *Gwangju Institute of Science and Technology (GIST), Cheomdangwagi-ro 123, Buk-gu, Gwangju, 61005, Republic of Korea*
[b] *Korea Atomic Energy Research Institute (KAERI), Daedeok-daero 989, Yuseong-gu, Daejeon, 34057, Republic of Korea*
[c] *University of Science and Technology (UST), Gajung-ro 217, Yuseong-gu, Daejeon, 34113, Republic of Korea*

ARTICLE INFO

Keywords:
Automatic sleep scoring
Multiscale representation
Feature pyramid
Supervised contrastive learning
Single-channel EEG

ABSTRACT

Automatic sleep scoring is essential for the diagnosis and treatment of sleep disorders and enables longitudinal sleep tracking in home environments. Conventionally, learning-based automatic sleep scoring on single-channel electroencephalogram (EEG) is actively studied because obtaining multi-channel signals during sleep is difficult. However, learning representation from raw EEG signals is challenging owing to the following issues: (1) sleep-related EEG patterns occur on different temporal and frequency scales and (2) sleep stages share similar EEG patterns. To address these issues, we propose an automatic ***Slee***p scoring framework that incorporates (1) a feature ***Py***ramid and (2) supervised ***Co***ntrastive learning, named ***SleePyCo***. For the feature pyramid, we propose a backbone network named *SleePyCo-backbone* to consider multiple feature sequences on different temporal and frequency scales. Supervised contrastive learning allows the network to extract class discriminative features by minimizing the distance between intra-class features and simultaneously maximizing that between inter-class features. Comparative analyses on four public datasets demonstrate that *SleePyCo* consistently outperforms existing frameworks based on single-channel EEG. Extensive ablation experiments show that *SleePyCo* exhibited an enhanced overall performance, with significant improvements in discrimination between sleep stages, especially for N1 and rapid eye movement (REM). Source code is available at https://github.com/gist-ailab/SleePyCo.

1. Introduction

Sleep scoring, also referred to as “sleep stage classification” or ”sleep stage identification”, is critical for the accurate diagnosis and treatment of sleep disorders (Wulff, Gatti, Wettstein, & Foster, 2010). Individuals suffering from sleep disorders are at risk of fatal complications such as hypertension, heart failure, and arrhythmia (Torabi-Nami, Mehrabi, Borhani-Haghighi, & Derman, 2015). In this context, polysomnography (PSG) is considered the gold standard for sleep scoring and is used in the prognosis of typical sleep disorders (e.g., sleep apnea, narcolepsy, and sleepwalking) (Berthomier et al., 2007). PSG consists of the biosignals associated with bodily activities such as brain activity (electroencephalogram, EEG), eye movement (electrooculogram, EOG), heart rhythm (electrocardiogram, ECG), and chin, facial, or limb muscle activity (electromyogram, EMG). Generally, experienced sleep experts examine these recordings based on sleep scoring rules and assign 20- or 30-s segments of the PSG data (called “epoch”) to a sleep stage. Rechtschaffen and Kales (R&K) (Rechtschaffen, 1968) and American Academy of Sleep Medicine (AASM) (Berry et al., 2012) standards serve as typical sleep scoring rules. The R&K standard classifies sleep stages into wakefulness (W), rapid eye movement (REM), and non-REM (NREM). NREM is further subdivided into S1, S2, S3, and S4 or N1, N2, N3, and N4. In the AASM rule, N3 and N4 are merged into N3, and it categorizes the PSG epochs into five sleep stages. Recently, the improved version of R&K – the AASM rule – has been widely utilized in manual sleep scoring. According to this rule, sleep experts should visually analyze and categorize the epochs of the entire night to form a hypnogram. Thus, manual sleep scoring is an arduous and time-consuming process (Malhotra et al., 2013). By contrast, machine learning algorithms require less than a few minutes for sleep scoring (Phan et al., 2021), and their performance is comparable to that of sleep experts (Stephansen et al., 2018). Therefore, automatic sleep scoring is highly desired for fast and accurate healthcare systems.

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

[*] Corresponding author.
E-mail addresses: lsj2121@gist.ac.kr (S. Lee), yeon_guk@gm.gist.ac.kr (Y. Yu), shback@gm.gist.ac.kr (S. Back), hogeony@hogeony.com (H. Seo), kyoobinlee@gist.ac.kr (K. Lee).

https://doi.org/10.1016/j.eswa.2023.122551
Received 28 June 2023; Received in revised form 9 November 2023; Accepted 9 November 2023
Available online 18 November 2023

Several methods have been developed for automatic sleep scoring based on deep neural networks. Basic one-to-one schemes that utilize a single EEG epoch and produce its corresponding sleep stage, were proposed as early methods (Phan, Andreotti, Cooray, Chén, & De Vos, 2018a, 2018b). In addition, one-to-many (*i.e.*, multitask learning) (Phan, Andreotti, Cooray, Chén, & De Vos, 2018c) schemes have been presented. Because the utilization of multiple EEG epochs offers advantageous performance, several many-to-one methods (Chambon, Galtier, Arnal, Wainrib, & Gramfort, 2018; Seo et al., 2020; Sors, Bonnet, Mirek, Vercueil, & Payen, 2018; Tsinalis, Matthews, Guo, & Zafeiriou, 2016), the method of predicting the sleep stage of the target epoch with the given PSG signals, and many-to-many (*i.e.*, sequence-to-sequence) (Phan, Andreotti, Cooray, Chén, & De Vos, 2019; Phan et al., 2021; Supratak, Dong, Wu, & Guo, 2017) methods have been proposed for automatic sleep scoring. The existing methods are generally based on convolutional neural networks (CNNs) (Andreotti et al., 2018; Chambon et al., 2018; Phan et al., 2018b, 2018c; Sors et al., 2018; Tsinalis et al., 2016; Vilamala, Madsen, & Hansen, 2017), recurrent neural networks (RNNs) (Phan et al., 2019), deep belief networks (DBNs) (Längkvist, Karlsson, & Loutfi, 2012), convolutional recurrent neural networks (CRNNs) (Korkalainen et al., 2020; Mousavi, Afghah, & Acharya, 2019; Phan et al., 2021; Seo et al., 2020; Sun, Chen, Li, Fan, & Chen, 2019; Supratak et al., 2017; Supratak & Guo, 2020), fully convolutional networks (FCNs) (Jia et al., 2021; Perslev, Jensen, Darkner, Jennum, & Igel, 2019), transformer (Phan et al., 2022), CNN+Transformer (Phan et al., 2018a; Qu et al., 2020), and other network architectures (Huang, Ren, Zhou, & Yan, 2022; Sun, Fan, Chen, Li, & Chen, 2019).

To obtain an improved representation from EEG, the architectures are designed to extract multiscale features with varying temporal and frequency scales (Fiorillo, Favaro, & Faraci, 2021; Huang et al., 2022; Phan et al., 2018b; Qu et al., 2020; Sun, Chen, et al., 2019; Supratak et al., 2017; Wang et al., 2022). Phan et al. (2018b) and Qu et al. (2020) used two distinct widths of max-pooling kernels on the spectrogram. Supratak et al. (2017), Fiorillo et al. (2021), and Huang et al. (2022) designed two-stream networks with two distinct filter widths of the convolutional layer in representation learning. Further, Sun, Chen, et al. (2019) and Wang et al. (2022) utilized convolutional layers with two or more distinguished filter widths in parallel. These studies utilized feature maps with different receptive field sizes to obtain richer information from given input signals. However, they could not obtain the advantages of multi-level features, which represent broad temporal scales and frequency characteristics.

Automatic scoring methods based on batch contrastive approaches have been proposed (Jiang, Zhao, Du, & Yuan, 2021; Mohsenvand, Izadi, & Maes, 2020; Ye et al., 2021) to improve the representation of PSG signals without labeled data, as multiple self-supervised contrastive learning frameworks have been proposed for visual representation (Caron et al., 2020; Chen, Kornblith, Norouzi, & Hinton, 2020; He, Fan, Wu, Xie, & Girshick, 2020). These batch-based approaches have been extensively studied because they outperform the traditional contrastive learning methods (Hadsell, Chopra, & LeCun, 2006) such as the triplet (Schroff, Kalenichenko, & Philbin, 2015) and N-pair (Sohn, 2016) strategies. Mohsenvand et al. (2020) proposed self-supervised contrastive learning for electroencephalogram classification. Jiang et al. (2021) proposed self-supervised contrastive learning for EEG-based automatic sleep scoring. CoSleep (Ye et al., 2021) presented self-supervised learning for multiview EEG representation between the raw signal and spectrogram for automatic sleep scoring. These studies only solve the problem of lack of labeled PSG data and do not focus on accurate automatic scoring. Furthermore, they do not leverage the large amount of annotated PSG data.

To address the aforementioned limitations, we propose ***SleePyCo***, a novel automatic ***Slee***p scoring framework that jointly utilizes a feature ***Py***ramid (Lin et al., 2017) and supervised ***Co***ntrastive learning (Khosla et al., 2020). To incorporate the feature pyramid into automatic sleep scoring, we present the *SleePyCo-backbone*. This enables the feature pyramid to consider various temporal scales and frequency characteristics by leveraging multi-level features, resulting in enhanced discrimination between sleep stages. Our training framework, based on supervised contrastive learning, enables the network to extract class discriminative features by simultaneously minimizing the distance between intra-class features and maximizing the distance between inter-class features. To verify the effectiveness of the feature pyramid and supervised contrastive learning, we conducted an ablation study on the Sleep-EDF (Goldberger et al., 2000; Kemp, Zwinderman, Tuk, Kamphuisen, & Oberye, 2000) dataset. The results show that *SleePyCo* exhibits an improved overall performance in automatic sleep scoring by enhancing the discrimination between sleep stages, especially for N1 and REM stages. The results of extensive experiments and comparative analyses conducted on four public datasets further corroborate the performance of *SleePyCo*. *SleePyCo* achieves state-of-the-art performance by exploiting the representation power of the feature pyramid and supervised contrastive learning. The main contributions are summarized as follows:

- We present a novel framework named *SleePyCo* that jointly utilizes a feature pyramid and supervised contrastive learning for automatic sleep scoring.
  - We incorporate the feature pyramid for automatic sleep scoring and propose the *SleePyCo-backbone* to account for diverse temporal and frequency scales present in raw single-channel EEG, resulting in improved discrimination between sleep stages.
  - We propose a training framework based on supervised contrastive learning, marking the first application of this approach to automatic sleep scoring. Our framework aims to mitigate the ambiguity of sleep stage by extracting class discriminative features.
- We demonstrate that *SleePyCo* outperforms the state-of-the-art frameworks on four public datasets via comparative analyses.

2. Model architecture

2.1. Problem statement

The proposed model is designed to classify L successive single-channel EEG epochs into sleep stages for the Lth input EEG epoch (called the target EEG epoch). Formally, we define L successive single-channel EEG epochs sampled at F Hz as $\mathbf{X}^{(L)} \in \mathbb{R}^{1\times E\cdot F\cdot L}$, where E is the duration of an EEG epoch in seconds. The corresponding ground truth is denoted as $\boldsymbol{y}^{(L)} \in \{0,1\}^{N_c}$, where $\boldsymbol{y}^{(L)}$ denotes the one-hot encoding label of the target EEG epoch and $\sum_{j=1}^{N_c} y_j^{(L)} = 1$. N_c indicates the number of classes and was set to 5, following the five-stage sleep classification in the AASM rule (Berry et al., 2012), which is briefly summarized in Table 1. $\mathbf{X}^{(L)}$ can be described as $\mathbf{X}^{(L)} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_L\}$, where $\boldsymbol{x}_i \in \mathbb{R}^{1\times E\cdot F}$ is a E-second single EEG epoch sampled at F Hz.

2.2. Model components

The network architecture of *SleePyCo* was inspired by *IITNet* (Seo et al., 2020). As reported in Seo et al. (2020), considering the temporal relations between EEG patterns in intra- and inter-epoch levels is important for automatic sleep scoring because sleep experts analyze the PSG data in the same manner. However, EEG patterns exhibit various frequencies and temporal characteristics. For instance, the sleep spindles in N2 occur in the frequency range of 12–14 Hz between 0.5–2 s, whereas the slow wave activity in N3 occur in the frequency range of 0.5–2 Hz throughout the N3 stage. To address this, we incorporated a feature pyramid into our model to enable multiscale representation. This is because a feature pyramid can consider various temporal scales

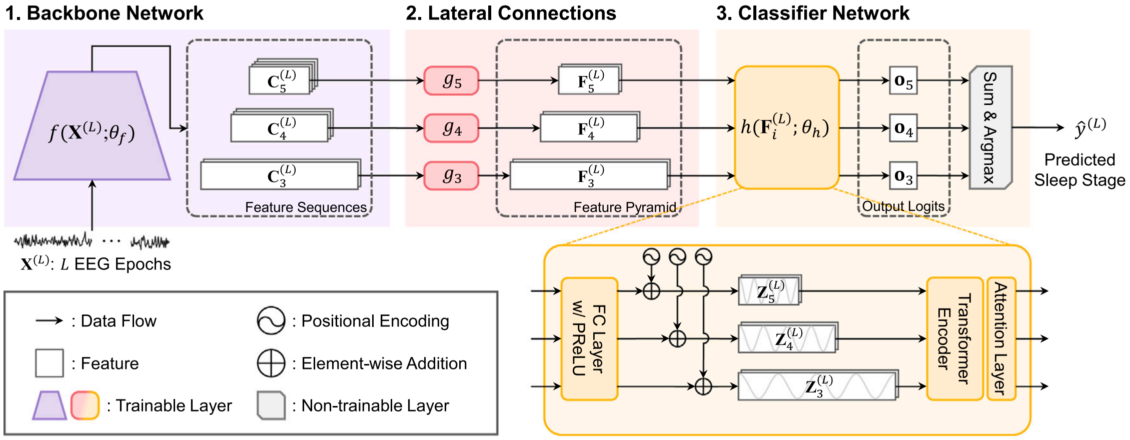


Fig. 1. Model Architecture of *SleePyCo*. The purple, pink, and yellow regions indicate the backbone network, lateral connections, classifier network of *SleePyCo*, and their corresponding outputs, respectively.

Table 1
EEG characteristics of each sleep stage according to the AASM rule (Berry et al., 2012); **bold** indicates the fundamental rationale that scores the corresponding sleep stage.

Sleep stage	EEG characteristics
Wake	• **More than 50% of alpha rhythm (8–13 Hz)** • **Beta rhythm (13–30 Hz)**
N1	• **Vertex sharp waves (5–14 Hz)** • **Low amplitude, mixed frequency activity (4–7 Hz)** • Less than 50% of alpha rhythm (8–13 Hz)
N2	• **K complex (8–16 Hz)** • **Sleep spindle (12–14 Hz)** • Low amplitude, mixed frequency activity (4–7 Hz) • Less than 50% of alpha rhythm (8–13 Hz)
N3	• **More than 20% of slow wave activity (0.5–2 Hz)** • Sleep spindle (12–14 Hz)
REM	• **Sawtooth waves (2–6 Hz)** • **Low amplitude, mixed frequency activity (4–7 Hz)** • Alpha rhythm (8–13 Hz) • K complex (8–16 Hz) • Sleep spindle (12–14 Hz)

and frequency characteristics, thereby enhancing discrimination between sleep stages. This is based on the rationale that a feature pyramid considers various frequency components and spatial scales, such as shape and texture, in computer vision (Hermann, Chen, & Kornblith, 2020).

As illustrated in Fig. 1, the proposed *SleePyCo* model comprises three major components: backbone network, lateral connections, and classifier network. The backbone network extracts feature sequences from raw EEG signals with multiple temporal scales and channel dimensions. Thus, we designed a shallow network based on previous studies because they achieved state-of-the-art performance (Perslev et al., 2019; Phan et al., 2021; Supratak & Guo, 2020). The lateral connections transform feature sequences with different channel dimensions into the same channel dimension via a single convolutional layer, resulting in a feature pyramid. For the classifier, a transformer encoder is employed to capture the sequential relations of EEG patterns on different temporal and frequency scales at sub-epoch levels.

2.2.1. Backbone network

To facilitate the feature pyramid, we propose a backbone network, named *SleePyCo-backbone*, containing five convolutional blocks, as proposed in Perslev et al. (2019), Seo et al. (2020). Each convolutional block is formed by the repetition of unit convolutional layers in the sequence of 1-D convolutional layer, 1-D batch normalization layer, and parametric rectified linear unit (PReLU) (He, Zhang, Ren, & Sun, 2015). All convolutional layers have a filter width of 3, stride length of 1, and padding size of 1 to maintain the temporal dimension within the same convolutional block. Max-pooling is performed between convolutional blocks to reduce the temporal dimension of feature sequences. Additionally, a squeeze and excitation module (Hu, Shen, & Sun, 2018) is included before the last activation function (PReLU) of each convolutional block. The details of the parameters, such as filter size, number of channels, and max-pooling size, are presented in Section 4.3.

Formally, *SleePyCo-backbone* takes L successive EEG epochs as input, obtaining the following set of feature sequences:

$$\{\mathbf{C}_3^{(L)}, \mathbf{C}_4^{(L)}, \mathbf{C}_5^{(L)}\} = f(\mathbf{X}^{(L)}; \theta_f), \tag{1}$$

where $\mathbf{C}_i^{(L)}$ denotes the output of the ith convolutional block of the backbone network, $f(\cdot)$ represents the backbone network, and θ_f indicates its trainable parameters. The size of the feature sequence can be denoted as $\mathbf{C}_i^{(L)} \in \mathbb{R}^{c_i \times \lceil 3000L/r_i \rceil}$, where $i \in \{3, 4, 5\}$ represents the stage index of the convolutional block, $c_i \in \{192, 256, 256\}$ denotes the channel dimension of the ith feature sequence, $r_i \in \{5^2, 5^3, 5^4\}$ denotes the temporal reduction ratio, and $\lceil \cdot \rceil$ signifies the ceiling operation. Notably, the temporal reduction ratio r_i is derived from the ratio of the length of the input $\mathbf{X}^{(L)}$ to that of the feature sequence $\mathbf{C}_i^{(L)}$. The feature sequences from the first and second convolutional blocks were excluded from the feature pyramid owing to their large memory allocation. Thus, the stage indices of 1 and 2 were not considered in this study.

2.2.2. Lateral connections

Lateral connections were attached at the end of the 3rd, 4th, and 5th convolutional blocks to form pyramidal feature sequences (*i.e.*, feature pyramid) with the same channel dimension. Importantly, the channel dimension of the feature pyramid should be identical because all pyramidal feature sequences share a single classifier network. Because the feature vectors in the feature pyramid represent an assorted frequency meaning but the same semantic meaning (EEG patterns), the application of a shared classifier is appropriate in our methods. Formally, the feature pyramid $\{\mathbf{F}_3^{(L)}, \mathbf{F}_4^{(L)}, \mathbf{F}_5^{(L)}\}$ can be obtained using the following equation:

$$\mathbf{F}_i^{(L)} = g_i(\mathbf{C}_i^{(L)}; \theta_{g,i}), \tag{2}$$

where $g_i(\cdot)$ denotes the lateral connection for the feature sequence $\mathbf{C}_i^{(L)}$ with the trainable parameter $\theta_{g,i}$. Each lateral connection consists of one convolutional layer with a filter width of 1 and results in a pyramidal feature sequence $\mathbf{F}_i^{(L)}$ that describes the same temporal scale with $\mathbf{C}_i^{(L)}$. Thus, the size of the pyramidal feature sequences is formulated as $\mathbf{F}_i^{(L)} \in \mathbb{R}^{d_f \times \lceil 3000L/r_i \rceil}$, where d_f is the channel dimension of feature pyramid.

2.2.3. Classifier network

The classifier network of *SleePyCo* can analyze the temporal context in the feature pyramid and output the predicted sleep stage of the target epoch $\hat{y}^{(L)}$. Formally, we denote the classifier network as $h(\mathbf{F}_i^{(L)}; \theta_h)$, where θ_h is the trainable parameter. We utilized the encoder part of the Transformer (Vaswani et al., 2017) for sequential modeling of the feature pyramid extracted from the raw single-channel EEG. Overall, recurrent architectures such as LSTM (Hochreiter & Schmidhuber, 1997) and GRU (Chung, Gulcehre, Cho, & Bengio, 2014) have been extensively utilized in automatic sleep scoring. Interestingly, the transformer delivered a remarkable performance in various sequential modeling tasks, including automatic sleep scoring (Phan et al., 2022; Qu et al., 2020; Shi, Chen, & Zhang, 2021). Owing to the large number of parameters of the original transformer, we reduced the model dimension d_m (*i.e.*, dimension of the query, key, and value in self-attention and output dimension of multi-head attention) and the feed-forward network dimension d_{FF} in comparison to the original ones. We set the number of heads N_h and the number of encoder layers N_e to be the same as the original transformer. The parameters are detailed in Section 4.3.

Prior to the transformer encoder, a shared fully connected layer with PReLU is employed to transform the channel dimension of the feature pyramid d_f into the model dimension d_m. Specifically, we denote the transformed feature pyramid as $\tilde{\mathbf{F}}_i^{(L)} \in \mathbb{R}^{d_m \times \lceil 3000L/r_i \rceil}$. This layer maps EEG patterns from various convolutional stages into the same feature space, and thus, the shared classifier considers the temporal context, regardless of the feature level. Subsequently, because the transformer encoder is a recurrent-free architecture, the positional encoding should be added to the input feature sequences to blend the temporal information:

$$\mathbf{Z}_i^{(L)} = \tilde{\mathbf{F}}_i^{(L)} + \mathbf{P}_i^{(L)}, \tag{3}$$

where $\mathbf{P}_i^{(L)} \in \mathbb{R}^{d_m \times \lceil 3000L/r_i \rceil}$ denotes the positional encoding matrix for the ith feature sequence. Herein, sinusoidal positional encoding was performed following a prior study (Vaswani et al., 2017). However, because the same time indices are applicable at both ends of the pyramidal feature sequences, we modified the positional encoding to coincide with the absolute temporal position between them by hopping the temporal index of positional encoding. Thus, the element of $\mathbf{P}_i^{(L)}$ at the temporal index t and dimension index k can be defined as

$$\mathbf{P}_i^{(L)}(t,k) = \begin{cases} \sin\left(\dfrac{tR^{(i-3)} + \lfloor R^{(i-3)}/2 \rfloor}{10000^{k/d_m}}\right), & \text{if } k \text{ is even,} \\ \cos\left(\dfrac{tR^{(i-3)} + \lfloor R^{(i-3)}/2 \rfloor}{10000^{k/d_m}}\right), & \text{otherwise,} \end{cases} \tag{4}$$

where $\lfloor \cdot \rfloor$ denotes the floor operation and $R = r_i/r_{i-1}$ indicates a temporal reduction factor (set as 5 in our model). $\mathbf{P}_i^{(L)}(t,k)$ degenerates into the original sinusoidal positional encoding for $i = 3$.

The output hidden states $\mathbf{H}_i^{(L)} \in \mathbb{R}^{d_m \times \lceil 3000L/r_i \rceil}$ for the ith pyramidal feature sequence can be formulated as

$$\mathbf{H}_i^{(L)} = \mathrm{TransformerEncoder}(\mathbf{Z}_i^{(L)}), \tag{5}$$

where $\mathrm{TransformerEncoder}(\cdot)$ denotes the encoder component of the transformer. To consolidate the output hidden states from the transformer encoder into a unified feature vector, we utilized the attention layer (Bahdanau, Cho, & Bengio, 2015; Luong, Pham, & Manning, 2015) as employed in Phan et al. (2019). Foremostly, the output hidden states $\mathbf{H}_i^{(L)}$ were projected into the attentional hidden states $\mathbf{A}_i^{(L)} = \{\mathbf{a}_{i,1}, \mathbf{a}_{i,2}, \ldots, \mathbf{a}_{i,T_i}\}$, where T_i is $\lceil 3000L/r_i \rceil$, via a single fully-connected layer. Thereafter, the attentional feature vector $\bar{\mathbf{a}}_i$ of the ith pyramidal feature sequence can be formulated via a weighted summation of the attentional hidden states along the temporal dimension:

$$\bar{\mathbf{a}}_i = \sum_{t=1}^{T_i} \alpha_{i,t} \mathbf{a}_{i,t}, \tag{6}$$

where $\alpha_{i,t}$ is the attention weight at time step t and $\mathbf{a}_{i,t}$ is the attentional hidden state at time step t. The attention weight at time step t is obtained by applying the softmax function to the attention score over the temporal dimension:

$$\alpha_{i,t} = \frac{\exp(\mathbf{W}_{\text{att}} \mathbf{a}_{i,t})}{\sum_t \exp(\mathbf{W}_{\text{att}} \mathbf{a}_{i,t})}, \tag{7}$$

where $\mathbf{W}_{\text{att}} \in \mathbb{R}^{1 \times d_m}$ represents the trainable weight matrix used to map the attentional hidden state to an attention score.

Upon using the attention feature vector $\bar{\mathbf{a}}$ obtained from Eq. (6), the output logits of the ith pyramidal feature sequence, $\mathbf{o}_i$, can be formulated as follows:

$$\mathbf{o}_i = \mathbf{W}_{\text{a}} \bar{\mathbf{a}}_i + \mathbf{b}_{\text{a}}, \tag{8}$$

where $\mathbf{W}_{\text{a}} \in \mathbb{R}^{N_c \times d_m}$ and $\mathbf{b}_{\text{a}} \in \mathbb{R}^{N_c}$ denote the trainable weight and bias, respectively. Eventually, the sleep stage $\hat{y}^{(L)}$ of the target epoch was predicted based on the following equation:

$$\hat{y}^{(L)} = \mathrm{argmax}\Big(\sum_{i \in \{3,4,5\}} \mathbf{o}_i\Big), \tag{9}$$

where $\mathrm{argmax}(\cdot)$ is the operation that returns the index of the maximum value.

3. Training procedure

As illustrated in Fig. 2, our learning framework involves two training steps. The first step involves contrastive representation learning (*CRL*) to pretrain the backbone network $f(\cdot)$ via supervised contrastive learning (Khosla et al., 2020). In this step, the backbone network $f(\cdot)$ is trained to extract the class discriminative features based on the supervised contrastive loss (Khosla et al., 2020). The second step involves multiscale temporal context learning (*MTCL*) to learn sequential relations in feature pyramid. For this purpose, we acquire the learned weights of $f(\cdot)$ from *CRL* and freeze them to conserve the class discriminative features. The remaining parts of the network ($g(\cdot)$ and $h(\cdot)$) are trained to learn the multiscale temporal context by predicting the sleep stage of the target epoch.

To prevent overfitting during the training procedures, we performed early stopping in both *CRL* and *MTCL*. Thus, validation was performed to track the validation cost (*i.e.*, validation loss) at every certain training iteration (*i.e.*, validation period, ψ), and the training was terminated if the validation loss did not progress more than a certain number of times (*i.e.*, early stopping patience, ϕ). Early stopping in our learning framework results in better pretrained parameters of the backbone network and prevents overfitting at the *MTCL* step. The details of the hyperparameters used in the training procedure are summarized in Section 4.4. Note that different validation periods, ψ_1 and ψ_2, were used in *CRL* and *MTCL*, respectively. The specifics of the training framework are described in the following sections and Algorithm 1.

3.1. Contrastive representation learning

In *CRL*, we adapted the training strategy of supervised contrastive learning (Khosla et al., 2020) to extract class discriminative features from a single EEG epoch. As illustrated in Fig. 2(a), the *CRL* aimed to maximize the similarity between the two projected vectors based on two different views of a single EEG epoch. Simultaneously, the similarity between two projected vectors from different sleep stages was minimized as the optimization of supervised contrastive loss (Khosla et al., 2020). Thus, we focused on reducing the ambiguous frequency characteristics by extracting distinguishable representations of the sleep stage. Accordingly, a single EEG epoch was initially transformed by two distinct augmentation functions. Thereafter, the encoder network and projection network mapped them into the hypersphere, which produced a latent vector $\boldsymbol{z}$. The details are explained in the following

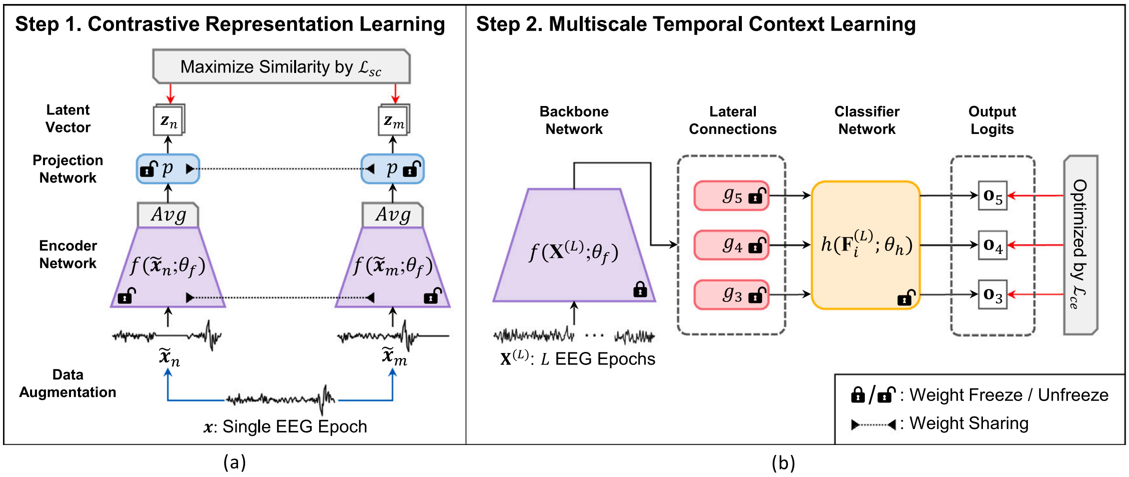

Step 1. Contrastive Representation Learning
Maximize Similarity by $\mathcal{L}_{sc}$
Latent Vector
z_n
z_m
Projection Network
p
p
Encoder Network
Avg
Avg
$f(\tilde{x}_n;\theta_f)$
$f(\tilde{x}_m;\theta_f)$
$\tilde{x}_n$
$\tilde{x}_m$
Data Augmentation
x: Single EEG Epoch
(a)
Step 2. Multiscale Temporal Context Learning
Backbone Network
Lateral Connections
Classifier Network
Output Logits
$f(\mathbf{X}^{(L)};\theta_f)$
g_5
g_4
g_3
$h(\mathbf{F}_i^{(L)};\theta_h)$
$\mathbf{o}_5$
$\mathbf{o}_4$
$\mathbf{o}_3$
Optimized by $\mathcal{L}_{ce}$
$\mathbf{X}^{(L)}$: L EEG Epochs
: Weight Freeze / Unfreeze
: Weight Sharing
(b)

Fig. 2. Illustration of the proposed training framework. (a) Contrastive representation learning (*CRL*); (b) multiscale temporal context learning (*MTCL*); red arrow indicates first backward path and blue arrow indicates $Aug(\cdot)$.

Algorithm 1 Training Algorithm

```
Input: $data_{train}$, $data_{val}$, early stopping patience $\phi$, learning rate $\eta$, validation
    period $\psi_1$, $\psi_2$, early stopping counter $p$, iteration counter $q$, and trainable
    parameter $\theta_f, \theta_p, \theta_g, \theta_h$
    /* Step 1: Contrastive Representation Learning */
$p \leftarrow 0$, $q \leftarrow 0$
while $p \leq \phi$ do
    Sample a minibatch $(\mathbf{X}^{(1)}, \mathbf{y}^{(1)}) \in data_{train}$
    Calculate $\mathcal{L}_{sc}$ as Eq. (11)
    Update $\theta_f$ and $\theta_p$ w.r.t $\mathcal{L}_{sc}$ by Adam with $\eta$
    $q \leftarrow q + 1$
    if $q$ mod $\psi_1 = 0$ then
        Calculate $\mathcal{L}_{sc}$ for $data_{val}$
        if $\mathcal{L}_{sc} >$ previous $\mathcal{L}_{sc}$ then
            $p \leftarrow p + 1$
        else
            Store $\theta_f$, $p \leftarrow 0$
        end if
    end if
end while
    /* Step 2: Multiscale Temporal Context Learning */
$p \leftarrow 0$, $q \leftarrow 0$
Restore $\theta_f$ of the lowest $\mathcal{L}_{sc}$, then freeze $\theta_f$
while $p \leq \phi$ do
    Sample a minibatch $(\mathbf{X}^{(L)}, \mathbf{y}^{(L)}) \in data_{train}$
    Calculate $\mathcal{L}_{ce}$ as Eq. (12)
    Update $\theta_g$ and $\theta_h$ w.r.t $\mathcal{L}_{ce}$ by Adam with $\eta$
    $q \leftarrow q + 1$
    if $q$ mod $\psi_2 = 0$ then
        Calculate $\mathcal{L}_{ce}$ for $data_{val}$
        if $\mathcal{L}_{ce} >$ previous $\mathcal{L}_{ce}$ then
            $p \leftarrow p + 1$
        else
            Store $\theta_g$ and $\theta_h$, $p \leftarrow 0$
        end if
    end if
end while
return trainable parameter $\theta_f, \theta_g, \theta_h$
```

Table 2
Data augmentation pipeline.

Transformation pipeline	Min	Max	Probability
amplitude scaling	0.5	2	0.5 each
time shift (samples)	−300	300	
amplitude shift (μV)	−10	10	
zero-masking (samples)	0	300	
additive Gaussian noise (σ)	0	0.2	
band-stop filter (2 Hz width) (lower bound frequency, Hz)	0.5	30.0	

sections: *Augmentation Module*, *Encoder Network*, *Projection Network*, and *Loss Function*, which constitute the major components of *CRL*.

Data Augmentation Module: The data augmentation module, $Aug(\cdot)$, transforms a single epoch of EEG signal $\boldsymbol{x}$ into a slightly different but semantically identical signal $\tilde{\boldsymbol{x}}$ (*i.e.* a different perspective on a single EEG epoch $\boldsymbol{x}$) as follows:

$$\tilde{\boldsymbol{x}} = Aug(\boldsymbol{x}). \tag{10}$$

With a given set of randomly sampled data $\{\boldsymbol{x}_p, \boldsymbol{y}_p\}_{p=1,\ldots,N_b}$ (*i.e.*, batch), two different $Aug(\cdot)$ result in $\{\tilde{\boldsymbol{x}}_q, \tilde{\boldsymbol{y}}_q\}_{q=1,\ldots,2N_b}$, called a multiviewed batch (Khosla et al., 2020) as illustrated in Fig. 2(a), where N_b is the batch size and $\tilde{\boldsymbol{y}}_{2p-1} = \tilde{\boldsymbol{y}}_{2p} = \boldsymbol{y}_p$. Because the augmentation set is crucial for contrastive learning, we adapted the verified transformation functions from previous studies (Mohsenvand et al., 2020; Supratak & Guo, 2020). We sequentially applied six transformations, namely, *amplitude scale, time shift, amplitude shift, zero-masking, additive Gaussian noise*, and *band-stop filter* with the probability of 0.5. In addition, we modified the hyperparameter of the transformation functions by considering the sampling rate and characteristics of the EEG signal. Table 2 lists the data augmentation pipeline details.

Encoder Network: The encoder network transforms an augmented single-EEG epoch $\tilde{\boldsymbol{x}}$ into a representation vector $\boldsymbol{r} \in \mathbb{R}^{c_5}$. The encoder network contains the sequence of the backbone network $f(\cdot)$ and the global average pooling operation $Avg(\cdot)$. Thus, the backbone network initially transforms an augmented single-EEG epoch $\tilde{\boldsymbol{x}}$ into the feature sequence $\mathbf{F}_5^{(1)}$. Then, the representation vector is obtained from the feature sequence via global average pooling along the temporal dimension. Formally, the representation vector is evaluated as $\boldsymbol{r} = Avg(\mathbf{F}_5^{(1)})$.

Projection Network: The projection network is vital for mapping the representation vector $\boldsymbol{r}$ into the normalized latent vector $\boldsymbol{z} = \frac{\boldsymbol{z}'}{\|\boldsymbol{z}'\|_2} \in \mathbb{R}^{d_z}$, where $\boldsymbol{z}' = p(\boldsymbol{r}; \theta_p)$, $p(\cdot)$ denotes the projection network, θ_p represents its trainable parameter, and d_z indicates the dimension of the latent vector. We use a multilayer perceptron (Hastie, Tibshirani, & Friedman, 2001) with a single hidden layer of size 128 to obtain a latent vector of size $d_z = 128$ as the projection network. For sequential modeling, the $p(\cdot)$ is removed at the *MTCL*.

Loss Function: For *CRL*, we employed the supervised contrastive loss as proposed in Khosla et al. (2020). The supervised contrastive loss simultaneously maximizes the similarity between positive pairs and promotes the deviations across negative pairs. In this study, samples annotated with the same sleep stage in a multiviewed batch are regarded as positives, and negatives otherwise. Formally, the supervised contrastive loss function can be formulated as

$$\mathcal{L}_{sc} = -\sum_{n=1}^{2N_b} \frac{1}{N_p^{(n)}} \sum_{m=1}^{2N_b} \log \frac{\mathbb{1}_{[n\neq m]}\mathbb{1}_{[\tilde{\boldsymbol{y}}_n=\tilde{\boldsymbol{y}}_m]}\exp(\boldsymbol{z}_n\cdot \boldsymbol{z}_m/\tau)}{\sum_{a=1}^{2N_b}\mathbb{1}_{[n\neq a]}\exp(\boldsymbol{z}_n\cdot \boldsymbol{z}_a/\tau)}, \tag{11}$$

where $N_p^{(n)}$ denotes the number of positives for the nth sample in a multiviewed batch excluding itself, $\mathbb{1}_{[\cdot]}$ denotes the indicator function, $\tilde{\boldsymbol{y}}_n$ denotes the ground truth corresponding to $\boldsymbol{z}_n$, $\cdot$ operation denotes the inner product between two vectors, and $\tau \in \mathbb{R}^+$ denotes a scalar temperature parameter ($\tau = 0.07$ in all present experiments).

3.2. Multiscale temporal context learning

As illustrated in Fig. 2(b), the second step of *SleePyCo* executes L successive EEG epochs $\mathbf{X}^{(L)}$ to analyze both the intra- and inter-epoch temporal contexts ($L = 10$ in this study). The performance obtained considering intra- and inter-epoch temporal contexts (Seo et al., 2020) is better than that considering only the intra-epoch. However, it is difficult for the network to capture the EEG patterns of the previous epochs, because only the label of the target epoch is provided. To resolve this, we fixed the parameters of the backbone network $f(\cdot)$ to maintain and utilize the class discriminative features learned from *CRL*. By contrast, the remaining parts of the network, which are lateral connections $g(\cdot)$ and classifier network $h(\cdot)$, are learned to score the sleep stage of the target epoch based on the following loss function:

$$\mathcal{L}_{ce} = -\sum_{i\in\{3,4,5\}} \sum_{j=1}^{N_c} y_j^{(L)} \log\left(\frac{\exp(o_{i,j})}{\sum_{k=1}^{N_c}\exp(o_{i,k})}\right), \tag{12}$$

where $y_j^{(L)}$ denotes the jth element of one-hot encoding label, and $o_{i,j}$ represents the jth element of output logits from the ith convolutional stage. This loss function follows the summation of cross entropy over the output logits from each convolutional block.

Because all pyramidal feature sequences share a single classifier network, the classifier network considers feature sequences across a broad scale. Thus, Eq. (12) facilitates the classifier network to analyze the temporal relation between the EEG patterns at different temporal scales and frequency characteristics. Consequently, the classifier network $h(\cdot)$ models the temporal information based on the pyramidal feature sequences $\mathbf{F}_i^{(L)}$ derived from analyzing the intra- and inter-epoch temporal contexts.

4. Experiments

4.1. Datasets

Four public datasets, including PSG recordings and their associated sleep stages, were utilized to assess the performance of *SleePyCo*: Sleep-EDF (2018 version) (Goldberger et al., 2000; Kemp et al., 2000), Montreal Archive of Sleep Studies (MASS) (O'reilly, Gosselin, Carrier, & Nielsen, 2014), Physionet2018 (Ghassemi et al., 2018; Goldberger et al., 2000), and Sleep Heart Health Study (SHHS) (Quan et al., 1997; Zhang et al., 2018). The number of subjects, utilized EEG channels, evaluation scheme, number of held-out validation subjects, and sample distribution are presented in Table 3. In this study, the duration of an EEG epoch was set to 30-s ($E = 30$), and all EEG signals, except for the Sleep-EDF dataset were downsampled to 100 Hz ($F = 100$) following previous works (Perslev et al., 2019; Phan et al., 2021). We did not employ preprocessing to EEG signals except for downsampling. For all datasets, we discarded the EEG epochs with annotations that were not related to the sleep stage, such as MOVEMENT class. In addition, we merged N3 and N4 into N3 to consider the five-class problem for datasets annotated with R&K (Rechtschaffen, 1968).

Sleep-EDF: The Sleep-EDF Expanded dataset (Goldberger et al., 2000; Kemp et al., 2000) (2018 version) includes 197 PSG recordings containing EEG, EOG, chin EMG, and event markers. The Sleep-EDF dataset comprises two kinds of studies: SC for 79 healthy Caucasian individuals without sleep disorders and ST for 22 subjects of a study on the effects of Temazepam on sleep. In this study, the SC recordings (subjects aged 25–101 years) were used based on previous studies (Mousavi et al., 2019; Perslev et al., 2019; Phan et al., 2019, 2021, 2022). According to the R&K rule (Rechtschaffen, 1968), sleep experts score each half-minute epoch with one of the eight classes {WAKE, REM, N1, N2, N3, N4, MOVEMENT, UNKNOWN}. Owing to the larger size of the class WAKE group compared to others, we included only 30 min of WAKE epochs before and after the sleep period.

MASS: The Montreal Archive of Sleep Studies (MASS) dataset (O'reilly et al., 2014) contains PSG recordings from 200 individuals (97 males and 103 females). This dataset includes five subsets (SS1, SS2, SS3, SS4, and SS5) that are classified based on the research purpose and data acquisition protocols. The AASM standard (Berry et al., 2012) (SS1 and SS3 subsets) or the R&K standard (Rechtschaffen, 1968) (SS2, SS4, and SS5 subsets) was used for the manual annotation. Specifically, the SS1, SS2, and SS4 subsets were annotated with 20-s EEG epochs instead of SS3 and SS5 subsets. Because 30-s EEG samples are used in *CRL*, 5-s segments of signals before and after the EEG epoch were considered for the case of SS1, SS2, and SS4. In *MTCL*, an equal length of EEG signals was used for all subsets (300 s in this study).

Physio2018: The Physio2018 dataset is contributed by the Computational Clinical Neurophysiology Laboratory at Massachusetts General Hospital and was applied to detect arousal during sleep in the 2018 Physionet challenge (Ghassemi et al., 2018; Goldberger et al., 2000). Owing to the unavailability of annotation for the evaluation set, we used the training set containing PSG recordings for 994 subjects aged 18–90. Thereafter, these recordings were annotated with the AASM rules (Berry et al., 2012), and we employed only C3–A2 EEG recordings for the single-channel EEG classification.

SHHS: The SHHS dataset (Quan et al., 1997; Zhang et al., 2018) is a multicenter cohort research that is designed to examine the influence of sleep apnea on cardiovascular diseases. The collection is composed of two rounds of PSG records: Visit 1 (SHHS-1) and Visit 2 (SHHS-2), wherein each record contains two-channel EEGs, two-channel EOGs, a single-channel EMG, a single-channel ECG, and two-channel respiratory inductance plethysmography, as well as the data from location sensors, light sensors, pulse oximeters, and airflow sensors. Each epoch was assigned a value of W, REM, N1, N2, N3, N4, MOVEMENT, and UNKNOWN according to the R&K rule (Rechtschaffen, 1968). In this study, the single-channel EEG (C4–A1) was analyzed from 5793 SHHS-1 recordings.

4.2. Backbone networks for ablation study

A direct comparison between the automatic sleep scoring methods would not be justified depending on the experimental settings such as the data processing method and training framework. For fair comparison with other state-of-the-art backbones, we implemented five backbones in our training framework: *DeepSleepNet* (Supratak et al., 2017), *TinySleepNet* (Supratak & Guo, 2020), *IITNet* (Seo et al., 2020), *U-Time* (Perslev et al., 2019), and *XSleepNet* (Phan et al., 2021). Additionally, we designed two experimental setups: the single-scale setting and the multiscale setting, to examine the performance of the *SleePyCo-backbone* w/ and w/o the influence of the feature pyramid.

Specifically, in the single-scale setting, we utilized only the feature sequence from the last layer of the backbone network during the *MTCL* step. In the multiscale setting, three feature sequences are utilized as the feature pyramid, ordered in ascending order based on

Table 3
Experimental settings and dataset statistics.

Dataset	No. of subjects	EEG channel	Experimental setting		Class distribution					
			Evaluation scheme	Held-out validation set	Wake	N1	N2	N3	REM	Total
Sleep-EDF	79	Fpz-Cz	10-fold CV	7 subjects	69,824 (35.8%)	21,522 (10.8%)	69,132 (34.7%)	13,039 (6.5%)	25,835 (13.0%)	199,352
MASS	200	C4-A1	20-fold CV	10 subjects	31,184 (13.6%)	19,359 (8.5%)	107,930 (47.1%)	30,383 (13.3%)	40,184 (17.5%)	229,040
Physio2018	994	C3-A2	5-fold CV	50 subjects	157,945 (17.7%)	136,978 (15.4%)	377,870 (42.3%)	102,592 (11.5%)	116,877 (13.1%)	892,262
SHHS	5,793	C4-A1	Train/Test: 0.7:0.3	100 subjects	1,691,288 (28.8%)	217,583 (3.7%)	2,397,460 (40.9%)	739,403 (12.6%)	817,473 (13.9%)	5,863,207

their temporal dimension. The *DeepSleepNet*, *TinySleepNet*, and *IITNet* backbones were considered only in the single-scale setting owing to their large memory footprint and structural limitation. All backbones were trained and evaluated in the proposed framework on the same datasets (Sleep-EDF, MASS, Physio2018, and SHHS).

DeepSleepNet Backbone: The structure of *DeepSleepNet* (Supratak et al., 2017) consists of a dual path CNN for representation learning and two layers of LSTM for sequential learning. To compare the representation power of the backbone network, we considered only the dual-path CNN component of *DeepSleepNet*. The filter width of a single path is smaller for capturing certain EEG patterns, and that of the other path is larger to consider the frequency components from the EEG. To aggregate features from each CNN path, interpolation is performed after the CNN path with larger filters. Thereafter, we concatenated these two features and applied the two convolutional layers following (Seo et al., 2020). The output size was 128 × 16 with a single EEG epoch.

TinySleepNet Backbone: *TinySleepNet* (Supratak & Guo, 2020) is composed of four layers of CNN and two layers of LSTM for representation learning and sequential learning, respectively, similar to the architecture of *DeepSleepNet* (Supratak & Guo, 2020). For the comparison of backbone network, we utilized CNN component of *TinySleepNet*. With given a single EEG epoch, the output size of *TinySleepNet* backbone was 128 × 4.

IITNet Backbone: *IITNet* (Seo et al., 2020) uses the modified 1-D ResNet-50 for extracting the representation of the raw EEG signal. Similar to the backbone network of *SleePyCo*, the backbone network of *IITNet* forms a five-block structure. However, the backbone network of *IITNet* has a deep architecture (49 convolutional layers), whereas that of *SleePyCo* is shallow (13 convolutional layers). Given a single EEG epoch of size 1 × 3000, the backbone network of *IITNet* outputs feature sequences of sizes 16 × 1500, 64 × 750, 64 × 375, 128 × 94, and 128 × 47 from each convolutional block.

U-Time Backbone: *U-Time* (Perslev et al., 2019) is a fully convolutional network for time-series segmentation applied for automatic sleep scoring. Similar to previous fully convolutional networks (Long, Shelhamer, & Darrell, 2015; Ronneberger, Fischer, & Brox, 2015), *U-Time* is the encoder–decoder structure, with the encoder for feature extraction and the decoder for time-series segmentation. Accordingly, we implemented only the encoder component to capture EEG patterns from raw EEG signals. The *U-Time* backbone comprises five convolutional blocks, similar to the proposed backbone network. However, the output from the last convolutional block could not be calculated from the single-epoch EEG because the encoder was designed to analyze 35 epochs of the EEG signals. To solve this problem, we lengthened the temporal dimension of the feature sequences by modifying the filter width of the max-pooling layer between the convolutional blocks from $\{10, 8, 6, 4\}$ to $\{8, 6, 4, 2\}$, respectively. Consequently, the sizes of the feature sequences were 16 × 3000, 32 × 375, 64 × 62, 128 × 15, and 256 × 7 from each convolutional block for a single EEG epoch.

XSleepNet Backbone: *XSleepNet* (Phan et al., 2021) is a multi-viewed model that acquires raw signals and time–frequency images as inputs. Thus, *XSleepNet* includes two types of encoder network: one for the raw signals and the other for the time–frequency images. To compare the raw signal domain, we used the encoder component on raw signals in the ablation study. This encoder was composed of nine one-dimensional convolutional layers with a filter width of 31 and stride length of 2. The sizes of the output feature sequences were 16 × 1500, 16 × 750, 32 × 325, 32 × 163, 64 × 82, 64 × 41, 128 × 21, 128 × 11, and 256 × 6, as obtained from 3000 samples of input EEG epochs.

4.3. Model specifications

The details of the components of *SleePyCo-backbone* are presented in Table 4. In addition, we used the one-dimensional operations of the convolutional layer, batch normalization layer, and max-pooling layer. All convolutional layers in *SleePyCo-backbone* had a filter width of 3, stride size of 1, and padding size of 1 to maintain the temporal dimension in the convolutional block. The max-pooling layers were utilized with a filter width of 5 between each convolutional block to reduce the temporal dimension of the feature sequence. As explained in Section 2.2.2, the lateral connections that follow the backbone network had a filter width of 1. The channel dimension of the feature pyramid d_f was set to 128. For the transformer encoder of the classifier network, the number of heads N_h was 8, the number of encoder layers N_e was 6, the model dimension d_m was 128, and the feed-forward network dimension d_{FF} was 128. The number of parameters in our model was 2.37 M (2.37×10^6).

4.4. Model training

The networks were trained using the Adam optimizer (Kingma & Ba, 2015) with a learning rate of $\eta = 1 \times 10^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1 \times 10^{-8}$. L2-weight regularization was used with a factor of 1×10^{-6} to prevent overfitting. Because a large batch size benefits contrastive learning (Chen et al., 2020; Khosla et al., 2020; Mohsenvand et al., 2020), a batch size of 1024 was employed in the *CRL*, whereas that of 64 was used in *MTCL*. For all datasets except SHHS, validation was conducted to track the validation loss used for early termination at every 50-th and 500-th training iterations (*i.e.*, validation period, ψ) in the *CRL* and *MTCL*, respectively. For the larger dataset SHHS, the validation period was 500 and 5000 in *CRL* and *MTCL*, respectively. Early stopping was employed by tracking the validation loss, such that the training was terminated if the validation loss did not decrease for 20 validation steps, (*i.e.*, early stopping patience, ϕ). At each cross validation, the model with the lowest validation loss was evaluated on the test set. The networks were trained on NVIDIA GeForce RTX 3090. Python 3.8.5 and PyTorch 1.7.1 (Paszke et al., 2019) were utilized in this study. In this development environment, the total training time of *SleePyCo* was approximately 22.5 h for Sleep-EDF, 76.5 h for MASS, 31.5 h for Physio2018, and 23 h for the SHHS dataset. The inference time of our model was 14.5 ms for a single forward pass, computed by averaging over 2000 iterations. The training and inference times, including the training time per fold, are summarized in Table 5.

Table 4
Model Specification of *SleePyCo-backbone*. C: channel dimension, T: temporal dimension, BN: Batch Normalization and SE: Squeeze and Excitation (Hu et al., 2018).

Layer name	Composition	Output size [$C \times T$]	
		$L = 1$	$L = 10$
Input	–	1 × 3000	1 × 30000
Conv1	3 Conv + BN + PReLU 3 Conv + BN + SE + PReLU	64 × 3000	64 × 30000
Max-pool1	5 Max-pool	64 × 600	64 × 6000
Conv2	3 Conv + BN + PReLU 3 Conv + BN + SE + PReLU	128 × 600	128 × 6000
Max-pool2	5 Max-pool	128 × 120	128 × 1200
Conv3	3 Conv + BN + PReLU 3 Conv + BN + PReLU 3 Conv + BN + SE + PReLU	192 × 120	192 × 1200
Max-pool3	5 Max-pool	192 × 24	192 × 240
Conv4	3 conv + BN + PReLU 3 Conv + BN + PReLU 3 Conv + BN + SE + PReLU	256 × 24	256 × 240
Max-pool4	5 Max-pool	256 × 5	256 × 48
Conv5	3 Conv + BN + PReLU 3 Conv + BN + PReLU 3 Conv + BN + SE + PReLU	256 × 5	256 × 48

Table 5
Training and inference times for *SleePyCo*: The training time is an approximate value, with the values in parentheses indicating training time per fold.

Dataset	Training time			Inference time
	CRL	MTCL	Total	
Sleep-EDF	2.5 h (15 m/fold)	20 h (2 h/fold)	22.5 h (2.25 h/fold)	14.5 ms
MASS	7 h (20 m/fold)	69.5 h (3.5 h/fold)	76.5 h (4 h/fold)	
Physio2018	3.5 h (40 m/fold)	28 h (5.5 h/fold)	31.5 h (6 h/fold)	
SHHS	3 h	20 h	23 h	

4.5. Model evaluation

4.5.1. Evaluation scheme

To assess the performance of *SleePyCo*, we conducted k-fold cross validation for the Sleep-EDF, MASS, and Physio-2018 datasets. Given that the number of subjects in a dataset is N_s, the records with N_s/k subjects were used for model evaluation (*i.e.*, test set), and the other records were classified into training and validation data. The selection of subjects for model evaluation was performed over all subjects by sequentially changing on k folds. As listed in Table 3, we utilized k as 10, 20, and 5 for the Sleep-EDF, MASS, and Physio2018 dataset, respectively. The held-out validation set refers to the number of subjects used for the validation set in a fold. For instance, subjects of the MASS dataset were categorized into 180, 10, and 10 recordings for the training, validation, and test set, respectively. Unlike these datasets, the SHHS dataset was randomly divided in a ratio of 0.7 to 0.3 for training and validation, respectively. As performed in Phan et al. (2021), we used 100 subjects for the validation.

4.5.2. Evaluation criteria

As the evaluation criteria, the overall accuracy (ACC), macro F1-score (MF1), and Cohen's Kappa (κ) (Sokolova & Lapalme, 2009) were used for the overall performance measurement and per-class F1-score (F1) was used for class-specific performance measurement. The respective equations for the evaluation criteria are as follows:

$$\mathrm{ACC} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{TN}+\mathrm{FN}}, \tag{13}$$

$$\mathrm{MF1} = \frac{1}{N_c}\sum_{j=1}^{N_c}\mathrm{F1}_j = \frac{1}{N_c}\sum_{j=1}^{N_c}\frac{2\times\mathrm{PR}_j\times\mathrm{RE}_j}{\mathrm{PR}_j+\mathrm{RE}_j}, \tag{14}$$

$$\kappa = \frac{\mathrm{ACC}-\mathrm{P}_e}{1-\mathrm{P}_e} = 1 - \frac{1-\mathrm{ACC}}{1-\mathrm{P}_e}, \tag{15}$$

where TP is true positive, TN is true negative, FP is false positive, FN is false negative, and $\mathrm{F1}_j$, PR_j, and RE_j are per-class F1-score, per-class precision, and per-class recall of the jth class, respectively. In Eq. (15), P_e represents the hypothetical probability of chance agreement. Typically, precision (PR) and recall (RE) can be defined as follows:

$$\mathrm{PR} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \tag{16}$$

$$\mathrm{RE} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}. \tag{17}$$

ACC is the intuitive performance indicator that is generally considered in several classification tasks. However, the F1-score indicating the harmonic mean of precision and recall is more valuable in class-imbalanced tasks such as sleep stage classification. In addition, the F1-score per class indicates the class-specific performance of the F1-score by calculating Eq. (14) without averaging. Cohen's Kappa κ indicates the agreement by chance for imbalanced proportions of various classes with a maximum value of 1.0 for ideal classification.

The mean Silhouette Coefficient (Rousseeuw, 1987) across all test data (referred to as the Silhouette Coefficient, S) was employed to evaluate the class discrimination ability of *SleePyCo*. This metric quantifies intra-class similarity and inter-class dissimilarity using the following equation:

$$S = \frac{1}{|\mathcal{D}|}\sum_{i\in\mathcal{D}} s(i), \quad \text{where} \quad s(i) = \frac{b(i)-a(i)}{\max\{a(i), b(i)\}}. \tag{18}$$

In Eq. (18), $\mathcal{D}$ is a set of data, $|\mathcal{D}|$ represents its cardinality, and $s(i)$ denotes the Silhouette Coefficient for the ith data. $a(i)$ and $b(i)$ represent intra-class similarity and inter-class dissimilarity, respectively, and are computed using the following equations:

$$a(i) = \frac{1}{|\mathcal{D}_I|-1}\sum_{j\in\mathcal{D}_I, i\neq j} d(i,j), \tag{19}$$

Table 6
Performance comparison between *SleePyCo* and state-of-the-art (SOTA) methods for automatic sleep scoring via deep learning; **bold** and underline indicate the best and second best, respectively. Furthermore, the results indicated by † were not directly comparable because they employed a subset distinct from the corresponding dataset. RS and SP denote Raw Signal and SPectrogram, respectively.

Method				Overall metrics			Per-class F1 score				
Dataset	System	Input	Subjects	ACC	MF1	κ	W	N1	N2	N3	REM
Sleep-EDF	**SleePyCo (Ours)**	RS	79	**84.6**	**79.0**	**0.787**	**93.5**	50.4	**86.5**	80.5	**84.2**
Sleep-EDF	XSleepNet (Phan et al., 2021)	RS + SP	79	84.0	77.9	0.778	93.3	49.9	86.0	78.7	81.8
Sleep-EDF	Korkalainen et al. (2020)	RS	79	83.7	–	0.77	–	–	–	–	–
Sleep-EDF	TinySleepNet (Supratak & Guo, 2020)	RS	79	83.1	78.1	0.77	92.8	**51.0**	85.3	**81.1**	80.3
Sleep-EDF	SeqSleepNet (Phan et al., 2019)	SP	79	82.6	76.4	0.760	–	–	–	–	–
Sleep-EDF	SleepTransformer (Phan et al., 2022)	SP	79	81.4	74.3	0.743	91.7	40.4	84.3	77.9	77.2
Sleep-EDF	U-Time (Perslev et al., 2019)	RS	79	81.3	76.3	0.745	92.0	**51.0**	83.5	74.6	80.2
Sleep-EDF	SleepEEGNet (Mousavi et al., 2019)	RS	79	80.0	73.6	0.73	91.7	44.1	82.5	73.5	76.1
MASS	**SleePyCo (Ours)**	RS	200	**86.8**	**82.5**	**0.811**	**89.2**	**60.1**	**90.4**	**83.8**	**89.1**
MASS	XSleepNet (Phan et al., 2021)	RS + SP	200	85.2	80.6	0.788	–	–	–	–	–
MASS	SeqSleepNet (Phan et al., 2019)	SP	200	84.5	79.8	0.778	–	–	–	–	–
MASS†	Sun, Chen, et al. (2019)	RS + SP	147	86.1	79.6	0.795	85.1	50.2	89.8	84.0	89.0
MASS†	Qu et al. (2020)	RS	62	86.5	81.0	0.799	87.2	52.8	91.5	87.0	86.6
MASS†	IITNet (Seo et al., 2020)	RS	62	86.3	80.5	0.794	85.4	54.1	91.3	86.8	84.8
MASS†	DeepSleepNet (Supratak et al., 2017)	RS	62	86.2	81.7	0.800	87.3	59.8	90.3	81.5	89.3
Physio2018	**SleePyCo (Ours)**	RS	994	**80.9**	**78.9**	**0.737**	**84.2**	**59.3**	**85.3**	**79.4**	**86.3**
Physio2018	XSleepNet (Phan et al., 2021)	RS + SP	994	80.3	78.6	0.732	–	–	–	–	–
Physio2018	SeqSleepNet (Phan et al., 2019)	SP	994	79.4	77.6	0.719	–	–	–	–	–
Physio2018	U-Time (Perslev et al., 2019)	RS	994	78.8	77.4	0.714	82.5	59.0	83.1	79.0	83.5
SHHS	**SleePyCo (Ours)**	RS	5,793	**87.9**	**80.7**	**0.830**	**92.6**	49.2	**88.5**	84.5	**88.6**
SHHS	SleepTransformer (Phan et al., 2022)	SP	5,791	87.7	80.1	0.828	92.2	46.1	88.3	**85.2**	**88.6**
SHHS	XSleepNet (Phan et al., 2021)	RS + SP	5,791	87.6	**80.7**	0.826	92.0	**49.9**	88.3	85.0	88.2
SHHS	Sors et al. (2018)	RS	5,728	86.8	78.5	0.815	91.4	42.7	88.0	84.9	85.4
SHHS	IITNet (Seo et al., 2020)	RS	5,791	86.7	79.8	0.812	90.1	48.1	88.4	**85.2**	87.2
SHHS	SeqSleepNet (Phan et al., 2019)	SP	5,791	86.5	78.5	0.81	–	–	–	–	–

$$b(i) = \min_{J \neq I} \frac{1}{|\mathcal{D}_J|} \sum_{j \in \mathcal{D}_J} d(i,j), \tag{20}$$

where $\mathcal{D}_I$ is a set of data that belongs to the same class as ith data, $\mathcal{D}_J$ is a set of data that belongs to a different class than ith data, and $d(i,j)$ is the distance between the ith and jth data, such as the Euclidean distance. $|\mathcal{D}_I|$ and $|\mathcal{D}_J|$ represent their respective cardinalities. $a(i)$ is the mean distance between the ith and all other data points in the same class, and $b(i)$ is defined as the smallest mean distance of ith data to all points in any other class to which ith data does not belong. The Silhouette Coefficient ranges between −1 and 1, with higher values close to 1 indicating better class discrimination.

5. Results and discussion

5.1. Performance comparison with state-of-the-art (SOTA) frameworks

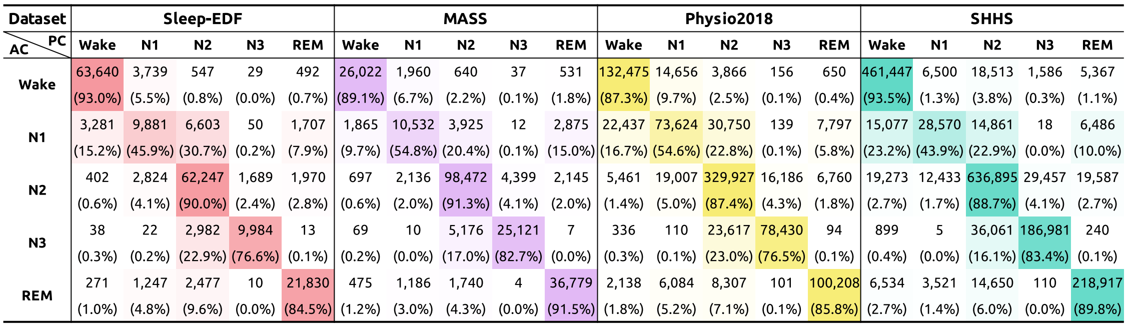
The performances of *SleePyCo* and SOTA frameworks are presented in Table 6 according to the datasets, system name, input types for the deep learning models, and number of subjects considered in the study. Fig. 3 illustrates the confusion matrices of *SleePyCo* on the Sleep-EDF, MASS, Physio2018, and SHHS datasets. Fig. 4 represents a hypnogram comparison of the best and worst scoring results predicted by *SleePyCo* and their corresponding ground truth for Sleep-EDF subjects. As shown in Figs. 3 and 4, predictions of *SleePyCo* exhibit a remarkable concurrence with the sleep stage scores annotated by human experts.

For all datasets, *SleePyCo* achieved state-of-the art performance compared with other models based on single-channel EEG. Quantitatively, *SleePyCo* delivered the best performance in terms of overall accuracy, MF1, and κ: 84.6%, 79.0%, 0.787 for Sleep-EDF, 86.8%, 82.5%, 0.811 for MASS, 80.9%, 78.9%, 0.737% for Physio2018, and 87.9%, 80.7%, 0.830 for SHHS, respectively. The performance differences between *SleePyCo* and the SOTA frameworks were +0.6%p, +1.1%p, +0.009 for Sleep-EDF, +1.6%p, +1.9%p, +0.023 for MASS, +0.6%p, +0.3%, +0.005 for Physio2018, and +0.2%p, +0.0%p, +0.002 for SHHS in overall accuracy, MF1, and κ, respectively. The proposed model achieved SOTA performance with the introduction of the feature pyramid and supervised contrastive learning. The network, particularly the classifier network, could learn the feature sequences with various temporal and frequency scales. Furthermore, contrastive learning enables the backbone network to extract class discriminative features, thereby reducing the ambiguity associated with EEG characteristics of sleep stages.

The major advantages of *SleePyCo* over other SOTA frameworks include *performance* and *use of single-channel EEG*, as indicated in Table 6. We achieved remarkable performance by solely utilizing raw single-channel EEG signals as input, without requiring any preprocessing or hand-crafted features. By contrast, existing SOTA frameworks utilize both raw signals and time–frequency images (*i.e.*, spectrogram) (Phan et al., 2021, 2022). Because the performance of automatic sleep scoring relies on several factors (Phan et al., 2021), the superiority of the raw EEG signal in comparison to the spectrogram could not be verified. However, the proposed model demonstrated SOTA performance by utilizing raw signals with no information loss compared to the time–frequency image. In addition, the number of parameters in *SleePyCo* is 2.37 M, which is 59% lower than the popular *XSleepNet* (5.8 M parameters) (Phan et al., 2021). Furthermore, under the same GPU conditions, its inference time (14.5 ms/sample) is 2 times faster than that of *XSleepNet* (29.7 ms/sample). Therefore, *SleePyCo* is suitable for real-time sleep scoring because it takes the target epoch and its previous epochs as input. This study can be expanded to classify other types of time-series data, such as sound and biosignals, to exploit the advantages of multiscale representation and supervised contrastive learning.

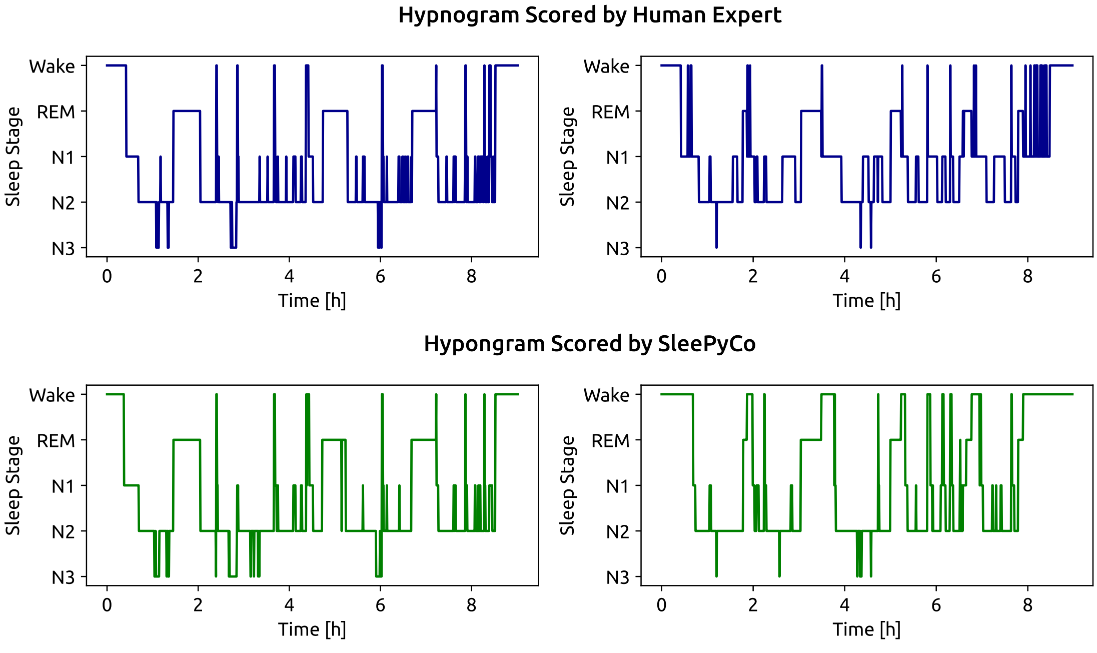
5.2. Ablation studies

To examine the effectiveness of *SleePyCo*, we conducted ablation studies and discussed on the following four aspects: the backbone network, feature pyramid (*FP*), contrastive representation learning (*CRL*), and Silhouette Coefficient, which are explained in Sections 5.2.1, 5.2.2, 5.2.3, and 5.2.4, respectively. Note that the *FP* is accompanied by the



Dataset	Sleep-EDF					MASS					Physio2018					SHHS				
AC \ PC	Wake	N1	N2	N3	REM	Wake	N1	N2	N3	REM	Wake	N1	N2	N3	REM	Wake	N1	N2	N3	REM
Wake	63,640 (93.0%)	3,739 (5.5%)	547 (0.8%)	29 (0.0%)	492 (0.7%)	26,022 (89.1%)	1,960 (6.7%)	640 (2.2%)	37 (0.1%)	531 (1.8%)	132,475 (87.3%)	14,656 (9.7%)	3,866 (2.5%)	156 (0.1%)	650 (0.4%)	461,447 (93.5%)	6,500 (1.3%)	18,513 (3.8%)	1,586 (0.3%)	5,367 (1.1%)
N1	3,281 (15.2%)	9,881 (45.9%)	6,603 (30.7%)	50 (0.2%)	1,707 (7.9%)	1,865 (9.7%)	10,532 (54.8%)	3,925 (20.4%)	12 (0.1%)	2,875 (15.0%)	22,437 (16.7%)	73,624 (54.6%)	30,750 (22.8%)	139 (0.1%)	7,797 (5.8%)	15,077 (23.2%)	28,570 (43.9%)	14,861 (22.9%)	18 (0.0%)	6,486 (10.0%)
N2	402 (0.6%)	2,824 (4.1%)	62,247 (90.0%)	1,689 (2.4%)	1,970 (2.8%)	697 (0.6%)	2,136 (2.0%)	98,472 (91.3%)	4,399 (4.1%)	2,145 (2.0%)	5,461 (1.4%)	19,007 (5.0%)	329,927 (87.4%)	16,186 (4.3%)	6,760 (1.8%)	19,273 (2.7%)	12,433 (1.7%)	636,895 (88.7%)	29,457 (4.1%)	19,587 (2.7%)
N3	38 (0.3%)	22 (0.2%)	2,982 (22.9%)	9,984 (76.6%)	13 (0.1%)	69 (0.2%)	10 (0.0%)	5,176 (17.0%)	25,121 (82.7%)	7 (0.0%)	336 (0.3%)	110 (0.1%)	23,617 (23.0%)	78,430 (76.5%)	94 (0.1%)	899 (0.4%)	5 (0.0%)	36,061 (16.1%)	186,981 (83.4%)	240 (0.1%)
REM	271 (1.0%)	1,247 (4.8%)	2,477 (9.6%)	10 (0.0%)	21,830 (84.5%)	475 (1.2%)	1,186 (3.0%)	1,740 (4.3%)	4 (0.0%)	36,779 (91.5%)	2,138 (1.8%)	6,084 (5.2%)	8,307 (7.1%)	101 (0.1%)	100,208 (85.8%)	6,534 (2.7%)	3,521 (1.4%)	14,650 (6.0%)	110 (0.0%)	218,917 (89.8%)

Fig. 3. Confusion matrix of *SleePyCo* for Sleep-EDF, MASS, Physio2018, and SHHS dataset. The values in parentheses indicate per-class recall. AC and PC denote Actual Class and Predicted Class, respectively.



(a) Best Case

(b) Worst Case

Fig. 4. Hypnogram scored by a human expert (top) and the hypnogram scored by *SleePyCo* (bottom). (a) and (b) represent the best and worst scoring results obtained from Sleep-EDF subjects using *SleePyCo*, respectively.

MTCL procedure. The results of the ablation studies are presented in Tables 7 and 8. In Section 5.2.1, we discuss the performance verification for *SleePyCo-backbone* by replacing it with other SOTA backbones. In Sections 5.2.2, 5.2.3, and 5.2.4, we demonstrate the effectiveness of *FP* and *CRL* by eliminating the components. Section 5.2.4 describes the Silhouette Coefficient analysis to demonstrate the enhancement in the class discrimination ability achieved by *SleePyCo*. Notably, *SleePyCo* that employs only $\mathbf{F}_5^{(L)}$, which was simultaneously trained from scratch, was set as the BaseLine, denoted as BL. Without *CRL*, the entire network of *SleePyCo* was trained from scratch using cross entropy loss. It is important to note that all experiments in the ablation studies were conducted under the identical conditions described in Section 4.4.

5.2.1. Performance of SleePyCo-backbone

The performances of *SleePyCo-backbone* and SOTA backbones on Sleep-EDF, MASS, Physio2018, and SHHS are compared in Table 7. In the single-scale setting, the proposed *SleePyCo-backbone* without feature pyramid (*i.e.*, w/o FP) displayed competitive or superior performance compared to the SOTA backbones. The performance differences between *SleePyCo-backbone* w/o FP and the best results of the single-scale backbones in terms of accuracy, MF1, and κ were +0.3%p, +0.2%p, +0.005 for Sleep-EDF, +0.0%p, +0.2%p, +0.000 for MASS, +0.2%p, +0.5%p, +0.004 for Physio2018, and −0.3%p, −1.0%p, −0.004 for SHHS, respectively. With the application of the feature pyramid, the performance of the proposed model was superior to that of the *U-Time* and *XSleepNet* backbones. The variations in overall accuracy, MF1, and κ between *SleePyCo-backbone* and other SOTA backbones were +0.2%p, +0.2%p, +0.004 for Sleep-EDF, +0.2%p, +0.5%p, +0.004 for MASS, +0.5%p, +0.6%p, +0.006 for Physio2018, and +0.1%p, +0.9%p, +0.002 for SHHS, respectively.

The results of the extensive experiments revealed the superior performance of the proposed *SleePyCo-backbone* compared to that of other network architectures. Exceptionally, the *IITNet* backbone demonstrated superior performance compared to the other backbones on SHHS dataset in the single-scale setting. This result indicates that the *IITNet* backbone, a deeper neural network with 49 convolutional layers, is capable of effectively extracting rich features from a large dataset, leading to improved performance compared to the other backbones (Seo et al., 2020), which have fewer convolutional layers (8, 4, 10, 9, and 13 for *DeepSleepNet*, *TinySleepNet*, *U-Time*, *XSleepNet*, and *SleePyCo*, respectively). In addition, the feature pyramid tended to improve the overall sleep scoring performance in the *U-Time* and *XSleepNet* backbones. These results imply that the feature pyramid

Table 7
Performance comparison of *SleePyCo-backbone* and SOTA backbones on experimental datasets; **bold** and underline indicate the first and second highest, respectively. SS and MS denote Single-Scale and MultiScale, respectively.

Backbone	Setting	Sleep-EDF			MASS			Physio2018			SHHS		
		ACC	MF1	κ	ACC	MF1	κ	ACC	MF1	κ	ACC	MF1	κ
DeepSleepNet	SS	83.8	78.2	0.775	86.2	81.4	0.801	79.6	77.3	0.721	87.1	79.4	0.818
TinySleepNet	SS	83.6	77.4	0.772	86.3	81.9	0.803	79.8	77.5	0.723	87.2	79.1	0.820
IITNet	SS	83.5	77.8	0.771	86.1	81.4	0.801	79.9	77.6	0.724	**87.5**	79.3	**0.824**
U-Time	SS	83.6	78.1	0.773	**86.4**	81.6	**0.804**	80.1	77.8	0.726	87.1	79.1	0.817
XSleepNet	SS	83.4	77.2	0.769	86.0	81.3	0.799	79.7	77.6	0.722	87.4	**79.5**	0.821
SleePyCo (Ours)	SS	**84.1**	**78.4**	**0.780**	**86.4**	**82.1**	**0.804**	**80.3**	**78.3**	**0.730**	87.2	78.5	0.820
U-Time	MS	84.4	78.8	0.783	86.6	82.0	0.807	80.4	78.3	0.731	87.8	79.5	0.828
XSleepNet	MS	83.5	77.5	0.771	86.2	81.8	0.803	80.2	78.0	0.728	87.7	79.8	0.826
SleePyCo (Ours)	MS	**84.6**	**79.0**	**0.787**	**86.8**	**82.5**	**0.811**	**80.9**	**78.9**	**0.737**	**87.9**	**80.7**	**0.830**

Table 8
Ablation study on Sleep-EDF; **bold** indicates the highest. BL, FP, and CRL indicate BaseLine, Feature Pyramid, and Contrastive Representation Learning, respectively.

Method	Overall metrics			Per-class F1 score					Silhouette coefficient
	ACC	MF1	κ	W	N1	N2	N3	REM	
BL	83.2	77.3	0.767	93.2	46.6	85.1	79.9	81.6	0.292
BL + FP	83.5	77.7	0.772	93.2	47.9	85.1	79.9	82.3	0.294
BL + CRL	84.1	78.4	0.780	93.2	49.3	86.1	79.7	83.5	0.317
BL + FP + CRL (Ours)	**84.6**	**79.0**	**0.787**	**93.5**	**50.4**	**86.5**	**80.5**	**84.2**	**0.325**

improves the sleep scoring performance by imparting richer features in *SleePyCo-backbone* as well as other architectures.

5.2.2. Influence of feature pyramid

Table 8 shows the ablation study results for *FP* and *CRL* on the Sleep-EDF dataset. The results indicate that the feature pyramid improves the sleep scoring performance, regardless of the *CRL*. Upon adding the feature pyramid to BL, the overall performances were enhanced by 0.3%p, 0.4%p, and 0.005 in ACC, MF1, and κ, respectively. In the case of application to *CRL*, the feature pyramid enhanced the sleep scoring performance by 0.5%p, 0.6%p, and 0.007 in ACC, MF1, and κ, respectively. Because the proposed model predicts the sleep stage with the summation of logits from each convolutional stage, the feature pyramid has the ensemble effect that enhances the performance based on predictions from various models. As reported in the literature (Lin et al., 2017), the feature pyramid provides richer information than the single-scale representation, which result in overall performance improvement.

In terms of per-class performance, the F1 scores of entire classes increased when feature pyramid was employed. On average, feature pyramid enhanced the F1 scores by 0.15%p for W, 1.2%p for N1, 0.2%p for N2, 0.4%p for N3, and 0.7%p for REM. The performance improvement for N1, N3, and REM was greater than that for the other sleep stages. This result indicates that low- and mid-level features contribute to the network classification performance for N1, N3, and REM. According to the AASM (Berry et al., 2012) rule, which is briefly described in Table 1, the fundamental scoring rationales of N1 and REM in EEG are low amplitude, mixed frequency (LAMF) activity as a common feature, and vertex sharp waves and sawtooth waves as a distinguishable feature, respectively. These rationales have a relatively mid-range frequency (4–7 Hz for LAMF activity, 5–14 Hz for vertex sharp waves, and 2–6 Hz for sawtooth waves) compared to the characteristics of the other sleep stages. Moreover, N3 is principally scored by the existence of slow wave activity that has a relatively low-range frequency between 0.5–2 Hz.

To determine the relationship between the feature pyramid and per-class performance, we evaluated the per-class F1 of each stage by separating the output logits from each stage, as described in Table 9. Consequently, the F1 of W and N2, which is scored according to mid- and high-frequency characteristics (W: alpha (8–13 Hz) and beta rhythm (13–30 Hz); N2: K complex (8–16 Hz) and sleep spindles (12–14 Hz); REM: mixed frequency (2–14 Hz)), was the highest at stage index 5. At the sleep stage of N1, where the mid-frequency range forms the dominant frequency component (theta wave with 4–7 Hz), the highest F1 occurs at stage index 4. In the case of N3, F1 is significantly high at stage index 3 because N1 exhibits low-frequency characteristics (slow wave activity with 0.5–2 Hz). The results indicate that the specific frequency components were intensively considered according to the feature level. These results demonstrate that the feature pyramid provides more information to enable the consideration of the AASM rules by the network instead of single-scale baselines. Furthermore, the feature type extracted from the CNN varies with the feature level in automatic sleep scoring as well as computer vision (Geirhos et al., 2019; Hermann et al., 2020).

5.2.3. Effect of contrastive representation learning

As observed in Table 8 by comparing BL with BL + CRL and BL + FP with BL + FP + CRL, the overall metrics were improved by *CRL*. When the baseline model w/o and w/ feature pyramid were trained on *CRL*, the overall performances in ACC, MF1, and κ were enhanced by 1.0%p, 1.2%p, and 0.014, respectively. In particular, the F1 scores of N1, N2, and REM were significantly improved by 2.6%p, 1.2%p, and 1.9%p, respectively, compared to the improvements of the other classes; 0.15%p for W, 0.2%p for N3 with arithmetic mean. Following the AASM rule, the categorization of sleep stages using only EEG is confusing owing to their similar frequency activities as described in Table 1. Specifically, N1, N2, and REM share a similar EEG characteristic, namely, LAMF activity. Thus, ambiguous EEG characteristics between the sleep stages is a primary factor that affects the sleep scoring performance on EEG signals. Based on these facts and experimental results, the proposed training framework enables the network to extract the discriminative features between sleep stages more effectively, especially for N1, N2, and REM, than the vanilla-supervised strategy. Primarily, these factors contribute toward the improvement of the overall performance.

5.2.4. Silhouette coefficient analysis

The first right-hand column of Table 8 lists Silhouette Coefficients of attentional feature vector at the same convolutional level ($\bar{\mathbf{a}}_5$). Silhouette Coefficients were computed by averaging the values across all folds using the test set. When FP was added to the BL and BL + CRL, the Silhouette Coefficients were enhanced by +0.002 and +0.008, respectively. This demonstrates that the feature pyramid enhances the network, particularly the classifier network, by extracting class discriminative features through the provision of rich information at various

Table 9
Per-class F1 score evaluated on each stage index $i \in \{3, 4, 5\}$; **bold** indicates the highest except values in the last row.

Stage index	Overall metrics			Per-class F1 score				
	ACC	MF1	κ	W	N1	N2	N3	REM
3	83.9	78.1	0.776	93.2	48.7	85.9	**80.3**	82.5
4	**84.4**	**78.7**	**0.783**	93.4	**50.4**	86.2	80.0	**83.7**
5	**84.4**	78.5	**0.783**	**93.5**	49.7	**86.3**	79.4	**83.7**
3, 4, 5	84.6	79.0	0.787	93.5	50.4	86.5	80.5	84.2

temporal and frequency scales. In other words, the feature pyramid improved the quality of information for discriminating EEG signals with mixed frequencies. When adding *CRL* to BL and BL + FP, the Silhouette Coefficients increased by +0.025 and +0.031, respectively. Because the objective of *CRL* is to increase both the intra-class similarity and inter-class dissimilarity. The obtained results are straightforward.

In summary, the introduction of the feature pyramid and supervised contrastive learning enhanced the overall performance of the baseline by a considerable margin of 1.4%p, 1.7%p, and 0.02 in ACC, MF1, and κ, respectively. For the per-class performance, the *F1 scores of N1 and REM were significantly improved by 3.8*%p *and 2.6*%p*, respectively*, compared to the other classes (0.3%p for W, 1.4%p for N2, and 0.6%p for N3). The Silhouette Coefficient, a measure of discrimination power, was enhanced by 0.033 with the introduction of the proposed *SleePyCo* compared with the baseline. Furthermore, a higher increase in the Silhouette Coefficient was observed when utilizing both elements simultaneously as opposed to using them individually. These results demonstrate the synergistic effect of the feature pyramid and supervised contrastive learning in enhancing the discrimination ability of raw EEG signals. The feature pyramid enriches the information with various temporal and frequency scales, thereby improving the capacity of the network, particularly the classifier network, to distinguish EEG signals with ambiguous patterns. Simultaneously, the proposed learning framework, based on supervised contrastive learning, enables the backbone network to extract discriminative features that are beneficial for scoring the target epochs.

5.3. Limitations and future work

The limitations of this study include the chronic problem affecting the transformer, as reported in Carion et al. (2020), Zhuang et al. (2023). Specifically, training memory resources increase exponentially in proportion to the feature size, which is related to the sequence length (L). A detailed description of memory utilization is provided in Appendix. As future work, we aim to achieve more accurate sleep scoring via supervised contrastive learning for multi-channel PSG signals.

6. Conclusion

We presented *SleePyCo*, which incorporates a feature pyramid and supervised contrastive learning for accurate automatic sleep scoring. Inspired by the evidence that EEG patterns reflecting the sleep stage can be observed over various temporal and frequency scales, we proposed a deep learning model based on the feature pyramid. The proposed training framework, based on supervised contrastive learning, reduces ambiguities between sleep stages by extracting discriminative features. In ablation studies, *SleePyCo-backbone* outperformed SOTA backbones and the feature pyramid and supervised contrastive learning exhibited a synergistic effect in classifying the raw EEG signals. Therefore, the feature pyramid improved the overall performance and discrimination capacity of classifier network by considering various temporal and frequency scales of the feature sequences. According to the feature levels and the analysis of the frequency characteristics of the sleep stage, the proposed model exhibited a per-class performance effect. The supervised contrastive learning contributes toward overall performance improvement, which is attributed to the significantly enhanced prediction of N1 and REM by reducing the ambiguity between sleep stages. The comparative analysis demonstrated the SOTA performance of *SleePyCo* on four public datasets of varying sizes: Sleep-EDF, MASS, Physio2018, and SHHS. Furthermore, *SleePyCo* can be expanded to categorize other types of time-series data, which should focus on various temporal and frequency scales and similar frequency characteristics between classes.

CRediT authorship contribution statement

Seongju Lee: Conceptualization, Methodology, Software, Validation, Data curation, Writing – original draft, Visualization, Project administration. **Yeonguk Yu:** Conceptualization, Software, Validation, Writing – original draft, Writing – review & editing. **Seunghyeok Back:** Conceptualization, Validation, Data curation, Writing – review & editing. **Hogeon Seo:** Validation, Data curation, Writing – review & editing. **Kyoobin Lee:** Conceptualization, Resources, Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Kyoobin Lee reports financial support was provided by Korea Ministry of Science and ICT. Kyoobin Lee reports financial support was provided by Korea Ministry of Trade Industry and Energy.

Data availability

We shared the URL about our study in the manuscript. Readers can download public dataset and reproduce our results.

Acknowledgments

This research was supported by a grant from the Institute of Information and Communications Technology Planning and Evaluation (IITP) funded by the Korean government (MSIT) (No. 2020-0-00857, Development of cloud robot intelligence augmentation, sharing and framework technology to integrate and enhance the intelligence of multiple robots). Furthermore, this research was partially supported by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant funded by the Korean government (MOTIE) (No. 20202910100030).

Appendix. Memory utilization of SleePyCo

In this section, we provide further details about the memory utilization of *SleePyCo*. Based on our implementation, the memory usage amounts to 22.3 GB for *CRL* and 52.9 GB for *MTCL*. During inference, only 1.5 GB of memory was required. A single GPU is required for both inference and *CRL*, whereas three GPUs are required for *MTCL*. Fig. 5 illustrates the memory utilization during *MTCL* as it varies with the sequence length (L). As pointed out in Section 5.3, memory resources increase exponentially in proportion to the sequence length (L), as illustrated by the blue bar chart in Fig. 5. To address this issue, Rabe and Staats (2021) Rabe and Staats (2021) introduced a memory-efficient self-attention mechanism that demands only $O(\sqrt{n})$ memory, in contrast to the original self-attention approach by Vaswani

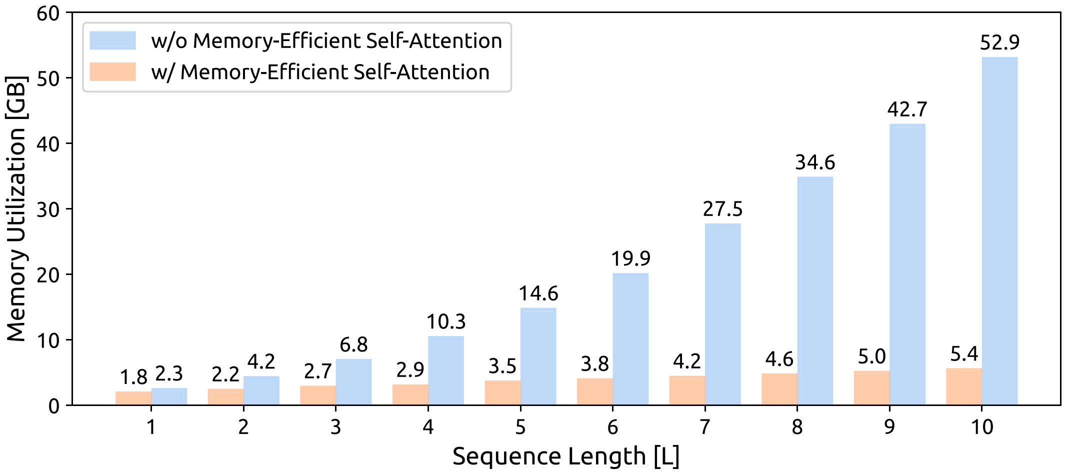


Fig. 5. Memory utilization of *SleePyCo* during *MTCL* varies with the sequence length (L) from 1 to 10. The orange and blue bar charts indicate memory requirements when memory-efficient self-attention is applied and when it is not, respectively.

et al. (2017) which requires $O(n^2)$ memory. This method has been incorporated into the latest versions of PyTorch (≥ v. 2.0). We integrated this memory-efficient attention in *SleePyCo* (as illustrated in the orange bar chart in Fig. 5). As a result, the training memory requirement for *SleePyCo* under *MTCL* has been reduced to 5.4 GB, marking a 90 % reduction from the original 52.9 GB (without the memory-efficient self-attention).

References

Andreotti, F., Phan, H., Cooray, N., Lo, C., Hu, M. T., & De Vos, M. (2018). Multichannel sleep stage classification and transfer learning using convolutional neural networks. In *2018 40th annual international conference of the IEEE engineering in medicine and biology society* (pp. 171–174). IEEE.

Bahdanau, D., Cho, K. H., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *3rd international conference on learning representations*.

Berry, R. B., Brooks, R., Gamaldo, C. E., Harding, S. M., Marcus, C., Vaughn, B. V., et al. (2012). The AASM manual for the scoring of sleep and associated events. *Rules, Terminology and Technical Specifications, Darien, Illinois, American Academy of Sleep Medicine*, *176*, 2012.

Berthomier, C., Drouot, X., Herman-Stoïca, M., Berthomier, P., Prado, J., Bokar-Thire, D., et al. (2007). Automatic analysis of single-channel sleep EEG: validation in healthy individuals. *Sleep*, *30*(11), 1587–1595.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. In *European conference on computer vision* (pp. 213–229). Springer.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., & Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, *33*, 9912–9924.

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

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *International conference on machine learning* (pp. 1597–1607). PMLR.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 workshop on deep learning*.

Fiorillo, L., Favaro, P., & Faraci, F. D. (2021). Deepsleepnet-lite: A simplified automatic sleep stage scoring model with uncertainty estimates. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *29*, 2076–2085.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., & Brendel, W. (2019). ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International conference on learning representations*. URL: https://openreview.net/forum?id=Bygh9j09KX.

Ghassemi, M. M., Moody, B. E., Lehman, L.-W. H., Song, C., Li, Q., Sun, H., et al. (2018). You snooze, you win: the physionet/computing in cardiology challenge 2018. In *2018 computing in cardiology conference, vol. 45* (pp. 1–4). IEEE.

Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., et al. (2000). PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *Circulation*, *101*(23), e215–e220.

Hadsell, R., Chopra, S., & LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In *2006 IEEE computer society conference on computer vision and pattern recognition, vol. 2* (pp. 1735–1742). IEEE.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning. Springer series in statistics*. New York, NY, USA.

He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9729–9738).

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision* (pp. 1026–1034).

Hermann, K., Chen, T., & Kornblith, S. (2020). The origins and prevalence of texture bias in convolutional neural networks. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems, vol. 33* (pp. 19000–19015). Curran Associates, Inc., URL: https://proceedings.neurips.cc/paper/2020/file/db5f9f42a7157abe65bb145000b5871a-Paper.pdf.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7132–7141).

Huang, J., Ren, L., Zhou, X., & Yan, K. (2022). An improved neural network based on SENet for sleep stage classification. *IEEE Journal of Biomedical and Health Informatics*.

Jia, Z., Lin, Y., Wang, J., Wang, X., Xie, P., & Zhang, Y. (2021). SalientSleepNet: Multimodal salient wave detection network for sleep staging. arXiv preprint arXiv:2105.13864.

Jiang, X., Zhao, J., Du, B., & Yuan, Z. (2021). Self-supervised contrastive learning for EEG-based sleep staging. In *2021 international joint conference on neural networks* (pp. 1–8). IEEE.

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

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., et al. (2020). Supervised contrastive learning. *Advances in Neural Information Processing Systems*, *33*.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In Y. Bengio, & Y. LeCun (Eds.), *3rd international conference on learning representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, conference track proceedings*. URL: http://arxiv.org/abs/1412.6980.

Korkalainen, H., Aakko, J., Nikkonen, S., Kainulainen, S., Leino, A., Duce, B., et al. (2020). Accurate deep learning-based sleep staging in a clinical population with suspected obstructive sleep apnea. *IEEE Journal of Biomedical and Health Informatics*, *24*(7), 2073–2081. http://dx.doi.org/10.1109/JBHI.2019.2951346.

Längkvist, M., Karlsson, L., & Loutfi, A. (2012). Sleep stage classification using unsupervised feature learning. *Advances in Artificial Neural Systems*, *2012*.

Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2117–2125).

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431–3440).

Luong, M. T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 1412–1421).

Malhotra, A., Younes, M., Kuna, S. T., Benca, R., Kushida, C. A., Walsh, J., et al. (2013). Performance of an automated polysomnography scoring system versus computer-assisted manual scoring. *Sleep*, *36*(4), 573–582.

Mohsenvand, M. N., Izadi, M. R., & Maes, P. (2020). Contrastive representation learning for electroencephalogram classification. In *Machine learning for health* (pp. 238–253). PMLR.

Mousavi, S., Afghah, F., & Acharya, U. R. (2019). SleepEEGNet: Automated sleep stage scoring with sequence to sequence deep learning approach. *PLoS One*, *14*(5), Article e0216456.

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

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, *32*.

Perslev, M., Jensen, M., Darkner, S., Jennum, P. J., & Igel, C. (2019). U-time: A fully convolutional network for time series segmentation applied to sleep staging. *Advances in Neural Information Processing Systems*, *32*, 4415–4426.

Phan, H., Andreotti, F., Cooray, N., Chén, O. Y., & De Vos, M. (2018a). Automatic sleep stage classification using single-channel EEG: Learning sequential features with attention-based recurrent neural networks. In *2018 40th annual international conference of the IEEE engineering in medicine and biology society* (pp. 1452–1455). IEEE.

Phan, H., Andreotti, F., Cooray, N., Chén, O. Y., & De Vos, M. (2018b). DNN filter bank improves 1-max pooling CNN for single-channel EEG automatic sleep stage classification. In *2018 40th annual international conference of the IEEE engineering in medicine and biology society* (pp. 453–456). IEEE.

Phan, H., Andreotti, F., Cooray, N., Chén, O. Y., & De Vos, M. (2018c). Joint classification and prediction CNN framework for automatic sleep stage classification. *IEEE Transactions on Biomedical Engineering*, *66*(5), 1285–1296.

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

Phan, H., Chén, O. Y., Tran, M. C., Koch, P., Mertins, A., & De Vos, M. (2021). XSleepNet: Multi-view sequential model for automatic sleep staging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

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

Qu, W., Wang, Z., Hong, H., Chi, Z., Feng, D. D., Grunstein, R., et al. (2020). A residual based attention model for eeg based sleep staging. *IEEE Journal of Biomedical and Health Informatics*, *24*(10), 2833–2843.

Quan, S. F., Howard, B. V., Iber, C., Kiley, J. P., Nieto, F. J., O'Connor, G. T., et al. (1997). The sleep heart health study: design, rationale, and methods. *Sleep*, *20*(12), 1077–1085.

Rabe, M. N., & Staats, C. (2021). Self-attention does not need $O(n^2)$ memory. arXiv preprint arXiv:2112.05682.

Rechtschaffen, A. (1968). A manual for standardized terminology, techniques and scoring system for sleep stages in human subjects. *Brain Information Service*.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International conference on medical image computing and computer-assisted intervention* (pp. 234–241). Springer.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65.

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 815–823).

Seo, H., Back, S., Lee, S., Park, D., Kim, T., & Lee, K. (2020). Intra-and inter-epoch temporal context network (IITNet) using sub-epoch features for automatic sleep scoring on raw single-channel EEG. *Biomedical Signal Processing and Control*, *61*, Article 102037.

Shi, G., Chen, Z., & Zhang, R. (2021). A transformer-based spatial-temporal sleep staging model through raw EEG. In *2021 international conference on high performance big data and intelligent systems* (pp. 110–115). IEEE.

Sohn, K. (2016). Improved deep metric learning with multi-class n-pair loss objective. *Advances in Neural Information Processing Systems*, *29*.

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

Sors, A., Bonnet, S., Mirek, S., Vercueil, L., & Payen, J. F. (2018). A convolutional neural network for sleep stage scoring from raw single-channel EEG. *Biomedical Signal Processing and Control*, *42*, 107–114.

Stephansen, J. B., Olesen, A. N., Olsen, M., Ambati, A., Leary, E. B., Moore, H. E., et al. (2018). Neural network analysis of sleep stages enables efficient diagnosis of narcolepsy. *Nature Communications*, *9*(1), 1–15.

Sun, C., Chen, C., Li, W., Fan, J., & Chen, W. (2019). A hierarchical neural network for sleep stage classification based on comprehensive feature learning and multi-flow sequence learning. *IEEE Journal of Biomedical and Health Informatics*, *24*(5), 1351–1366.

Sun, C., Fan, J., Chen, C., Li, W., & Chen, W. (2019). A two-stage neural network for sleep stage classification based on feature learning, sequence learning, and data augmentation. *IEEE Access*, *7*, 109386–109397.

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

Supratak, A., & Guo, Y. (2020). TinySleepNet: An efficient deep learning model for sleep stage scoring based on raw single-channel eeg. In *2020 42nd annual international conference of the IEEE engineering in medicine & biology society* (pp. 641–644). IEEE.

Torabi-Nami, M., Mehrabi, S., Borhani-Haghighi, A., & Derman, S. (2015). Withstanding the obstructive sleep apnea syndrome at the expense of arousal instability, altered cerebral autoregulation and neurocognitive decline. *Journal of Integrative Neuroscience*, *14*(02), 169–193.

Tsinalis, O., Matthews, P. M., Guo, Y., & Zafeiriou, S. (2016). Automatic sleep stage scoring with single-channel EEG using convolutional neural networks. arXiv preprint arXiv:1610.01683.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems* (pp. 5998–6008).

Vilamala, A., Madsen, K. H., & Hansen, L. K. (2017). Deep convolutional neural networks for interpretable analysis of EEG sleep stage scoring. In *2017 IEEE 27th international workshop on machine learning for signal processing* (pp. 1–6). IEEE.

Wang, H., Lu, C., Zhang, Q., Hu, Z., Yuan, X., Zhang, P., et al. (2022). A novel sleep staging network based on multi-scale dual attention. *Biomedical Signal Processing and Control*, *74*, Article 103486.

Wulff, K., Gatti, S., Wettstein, J. G., & Foster, R. G. (2010). Sleep and circadian rhythm disruption in psychiatric and neurodegenerative disease. *Nature Reviews Neuroscience*, *11*(8), 589–599.

Ye, J., Xiao, Q., Wang, J., Zhang, H., Deng, J., & Lin, Y. (2021). CoSleep: A multi-view representation learning framework for self-supervised learning of sleep stage classification. *IEEE Signal Processing Letters*, *29*, 189–193.

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

Zhuang, B., Liu, J., Pan, Z., He, H., Weng, Y., & Shen, C. (2023). A survey on efficient training of transformers. arXiv preprint arXiv:2302.01107.