scientific reports

Check for updates

OPEN

FlexSleepTransformer: a transformer-based sleep staging model with flexible input channel configurations

Yanchen Guo[1], Maciej Nowakowski[2] & Weiying Dai[1]✉

Clinical sleep diagnosis traditionally relies on polysomnography (PSG) and expert manual classification of sleep stages. Recent advancements in deep learning have shown promise in automating sleep stage classification using a single PSG channel. However, variations in PSG acquisition devices and environments mean that the number of PSG channels can differ across sleep centers. To integrate a sleep staging method into clinical practice effectively, it must accommodate a flexible number of PSG channels. In this paper, we proposed FlexSleepTransformer, a transformer-based model designed to handle varying number of input channels, making it adaptable to diverse sleep staging datasets. We evaluated FlexSleepTransformer using two distinct datasets: the public SleepEDF-78 dataset and the local SleepUHS dataset. Notably, FlexSleepTransformer is the first model capable of simultaneously training on datasets with differing number of PSG channels. Our experiments showed that FlexSleepTransformer trained on both datasets together achieved 98% of the accuracy compared to models trained on each dataset individually. Furthermore, it outperformed models trained exclusively on one dataset when tested on the other dataset. Additionally, FlexSleepTransformer surpassed state-of-the-art CNN and RNN-based models on both datasets. Due to its adaptability with varying channels numbers, FlexSleepTransformer holds significant potential for clinical adoption, especially when trained with data from a wide range of sleep centers.

Keywords Automatic sleep staging, Transformer, Sequence-to-sequence, Multi-channel, Multi-dataset, Deep neural network

Sleep plays a pivotal role in human health. Various sleep-related disorders, such as sleep apnea, insomnia, and narcolepsy, significantly reduce the quality of life for a substantial number of individuals, as indicated by data from CDC, estimating around 70 million affected Americans. Moreover, they are linked to many chronic health problems, including diabetes, stroke, brain injury, Parkinson disease, depression, and Alzheimer's disease[1–7]. Accurate measurement of sleep quality is essential for diagnosing sleep disorders and facilitating early detection of other health conditions. Currently, clinical sleep diagnosis relies on polysomnography (PSG), which involves overnight electroencephalogram (EEG), electrooculogram (EOG), electrocardiogram (ECG), and electromyogram (EMG). The PSG is segmented into 30-second epochs, which sleep experts then classify into different sleep stages following guidelines such as those provided by the American Academy of Sleep Medicine (AASM)[8]. Sleep stages typically include wake, three non-rapid eye movement (NREM: N1-N3), and rapid eye movement (REM) stages. However, manual sleep stage classification is a labor intensive and challenging process that relies on the subjective experience of sleep experts[9]. Recognizing this, recent sleep studies have shifted their focus towards the development of automatic sleep staging methods from conventional machine learning techniques to advanced deep learning approaches, including CNN, RNN, and transformer-based models. This progression reflects the growing demand for more efficient, accurate, and automated sleep stage identification methods.

Machine learning-based sleep staging

Conventional machine learning-based approaches to sleep staging rely heavily on manually extracting features from sleep epochs. Experts derive features from frequency domains — such as Delta, Theta and Alpha activities — alongside characteristic waveforms like K-complexes, sleep spindles and saw-tooth waves. Additionally,

[1]School of Computing, State University of New York at Binghamton, Binghamton, NY 13902, USA. [2]Sleep Medicine, United Health Services Hospitals, Inc, Binghamton, NY 13902, USA. ✉email: wdai@binghamton.edu

handcrafted transition rules are used to describe how sleep stages progress across consecutive epochs. These features, once extracted, are then used to train shallow classifiers like naive bayes[10], k-nearest neighbor[11,12], support vector machine[13–15], or random forest[16,17] to assign sleep stages. While these models achieve reasonable performance, they require substantial prior knowledge of and sleep physiology and are dependent on the quality of feature engineering, which can be intricate and time-consuming.

CNN & RNN-based sleep staging

The introduction of deep learning, particularly convolutional neural networks (CNN), has significantly transformed the field of sleep staging by enabling automatic feature extraction directly from raw sleep signals, CNNs are particularly effective at capturing localized patterns within both time and frequency domains, thus eliminating the need for manual feature engineering. To further enhance model performance, recurrent neural networks (RNNs), especially long short-term memory (LSTM) networks, were introduced to account for temporal dependency between consecutive epochs[8], thereby improving accuracy by effectively modeling the sequential nature of sleep stages.

Several prominent models have been developed within this framework. DeepSleepNet[18] uses two CNNs with different kernel sizes to extract temporal and frequency features from raw single-channel PSG data, combined with a bidirectional LSTM networks to capture sequential dependencies. TinySleepNet[19], a streamlined version of DeepSleepNet, integrates one CNN and LSTM architectures while employing data augmentation techniques to improve model robustness. SleepEEGNet[20] leverages CNNs to extract time-invariant features and bidirectional RNNs to capture dependencies from a series of epochs. The combined CNN and LSTM networks have also been used to investigate the effects of sleep apnea severity on classification accuracy[21]. EEGSNet[22] uses CNNs and bidirectional LSTMs to process EEG spectrogram inputs, capitalizing on time-frequency representations of sleep data. Jadhav et al.[23] uses the time-frequency CNN to develop a less complex and smaller model. SeqSleepNet[24] adopts an end-to-end RNN architecture to model sequences of consecutive epochs, thereby capturing inter-epoch relationships more effectively. Lastly, AttnSleep[25] enhances model performance through multi-resolution CNNs and adaptive feature recalibration, incorporating a temporal context encoder to optimize feature learning.

Attention-based sleep staging

Inspired by the seminal transformer network[26], more recent sleep scoring network architectures have focused on attention-based mechanisms and transformer-based models, which are adept at capturing both local and global dependencies within the data. Attention mechanisms allow models to prioritize important features across epochs, enhancing their ability to understand inter- and intra-epoch relationships. Researchers[25,27,28] combined CNNs with attention mechanisms, using CNNs to capture local signal characteristics and attention layers to refine feature learning across epochs. RobustSleepNet[29] applied attention to integrate features from multiple channels, enhancing the robustness of sleep stage predictions. MRASleepNet[30] introduced a multi-resolution attention (MRA) module to focus on different feature resolutions, supported by a feature extraction module and a gated multilayer perceptron (gMLP) to model temporal relationships. SleepTransformer[31] adopts the original transformer architecture[26], which is inherently suited to sequence modeling, to process single-channel sleep data with contextual input, allowing for more sophisticated temporal modeling. However, it is limited to processing only single-channel EEG data. Recently, MultiChannelSleepNet[32] extended the concept of SleepTransformer to multi-channel inputs, using transformers to process data from multiple EEG channels. However, its network architecture does not support the input from a sequence of epochs.

Our contribution

Despite all this progress, we found that (1) not many transformer-based deep learning (DL) models are dealing with multi-channel input, while the manual scoring from sleep experts typically considers multiple-channel PSG data, as well as sleep stages from adjacent epochs. To align more closely with expert sleep scoring, there is a pressing need for a transformer-based sleep network architecture capable of accepting input from a sequence of multiple-channel PSG data. We hypothesize that DL models utilizing multiple-channel PSG data will outperform those using single-channel PSG data. (2) Different sleep centers may have different numbers of PSG channels located at different brain sites and home-based sleep centers may have only a single PSG channel. It is preferable to build a uniform transformer-based model that allows for a flexible number of input channels. In this case, the model can be used to learn the sleep stage scoring using the data from different sleep centers, enabling the transfer learning from different hospitals with different electrode setups and different number of channels. We hypothesize that training a DL model with a diverse range of datasets from various PSG acquisition devices and environments will enable it to become a generalizable tool for automatic sleep stage classification. (3) The input signals can be from time domain[18,25,33] and time-frequency domain[24,31] PSG signals. The time-frequency images are obtained with time-frequency analysis, such as short-time Fourier transform (STFM), from raw time domain signals. Time-frequency images include both time domain and frequency domain features of the raw signals, which manual scoring of sleep experts consider. We hypothesize that the input with time-frequency images will utilize the information from both time and frequency domains and thereby with enhanced classification accuracy.

To address and validate the points discussed, we propose a new sleep stage classification model, FlexSleepTransformer. This model advances and extends the SleepTransformer network. Compared to previous works, especially the SleepTransformer, our contribution can be summarized as follows:

1. Dataset Versatility: FlexSleepTransformer is designed to perform effectively across a range of datasets, even those with varying numbers of PSG channels from different sleep centers. Unlike earlier models that are

specialized to single datasets, FlexSleepTransformer's adaptability to different input channel configurations may improve its transfer-learning capabilities on larger datasets.
2. Channel Handling: FlexSleepTransformer supports multi-channel staging, offering a more comprehensive analysis compared to the previous SleepTransformer, which is limited to single-channel staging.
3. Model Architecture: FlexSleepTransformer leverages transformer-based architecture, which has shown superior performance in domains like computer vision and language modeling, surpassing the capabilities of CNNs and RNNs. This represents a significant advancement from earlier models that did not incorporate transformer technology.
4. Sequence Processing: FlexSleepTransformer employs a sequence-to-sequence framework[24,34] combined with a self-attention mechanism, allowing it to efficiently capture context information from neighboring epochs. This represents a notable advancement over previous methods in handling sequence information.
5. Clinical Application: FlexSleepTransformer incorporates a local dataset from a sleep clinic in its experiments to better validate its clinical utility, particularly in assessing its performance across various sleep centers.

Results and discussion

Model performance

To illustrate qualitative performance of FlexSleepTransformer, we present a comparison between the ground truth hypnogram and the predicted hypnogram for a typical subject (an accuracy of 81.83% for this subject, which aligns with the 10-fold cross validation accuracy for the dataset) from the SleepUHS dataset (Fig. 1). The comparison reveals a high degree of similarity between the ground truth and predicted hypnograms. Additionally, we are using 2-dimensional t-SNE visualization of the 5-element vector before softmax to demonstrate the effectiveness of FlexSleepTransformer and TinySleepNet on the SleepUHS dataset (Fig. 2). Both TinySleepNet and FlexSleepTransformer reveal five sleep stages. However, in the t-SNE visualization of TinySleepNet, the points (sleep epochs) within the clusters for N1, N2, and N3 stages are more dispersed. This increased dispersion may be indicative of reduced performance in TinySleepNet. Table 1 shows the comprehensive and quantitative performance results of our proposed FlexSleepTransformer under various experimental setups. The performance metrics such as Accuracy, Cohen's kappa, Macro-F1 and class-wise Macro-F1 were assessed for FlexSleepTransformer and TinySleepNet on two datasets.

Intra-model comparison

Figure 3 illustrates the subject-wise cross-validation accuracy comparison of single channel input and multi-channel inputs for each model (TinySleepNet, FlexSleepTransformer/SSE, and FlexSleepTransformer/TF) on two datasets. In this intra-model comparison, different input channel types (single channel vs. multiple channels) are compared within the same model.

For TinySleepNet, multi-channel input significantly improves accuracy compared to single-channel input across both the SleepEDF and SleepUHS datasets. For FlexSleepTransformer/SSE, both multi-channel concatenation and multi-channel random fusion inputs demonstrate significantly higher accuracy than single-channel input in both datasets. Notably, multi-channel random fusion achieves higher accuracy than multi-channel concatenation. For FlexSleepTransformer/TF, multi-channel random fusion achieves higher accuracy than both multi-channel concatenation and single channel input across both datasets. However, multi-channel concatenation shows significantly higher accuracy compared to single-channel input only in the SleepUHS dataset, not in the SleepEDF dataset. Overall, for FlexSleepTransformer, multi-channel input with random fusion demonstrates superior performance than multi-channel input with concatenation, with both multi-channel

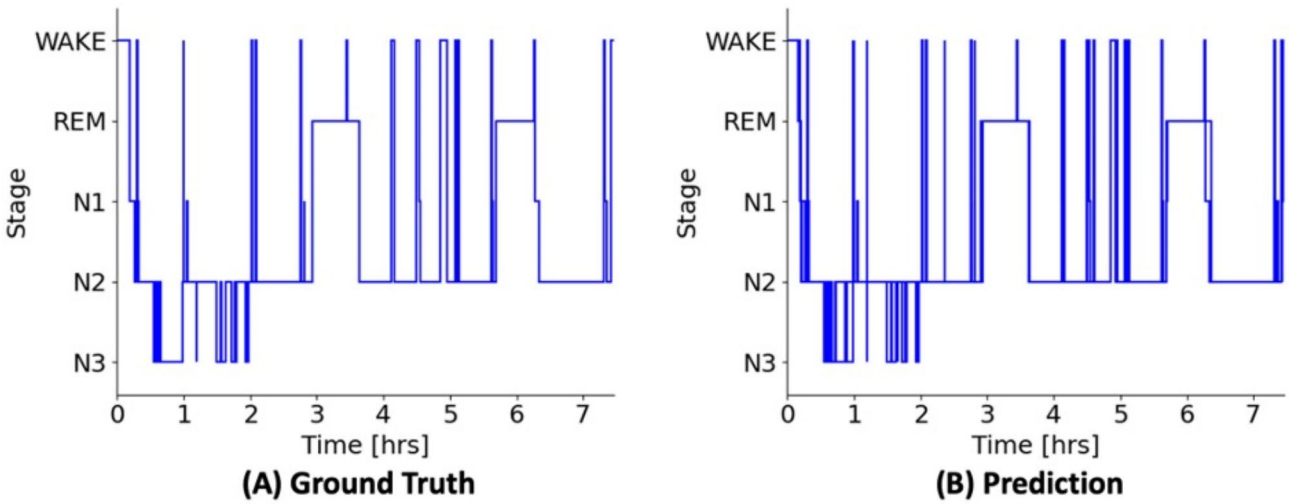

Fig. 1. Comparison between (**A**) the ground truth hypnogram and (**B**) the FlexSleepTransformer-predicted hypnogram for a typical test subject from the SleepUHS dataset. FlexSleepTransformer achieves an accuracy of 81.83% for this subject, which aligns with the 10-fold cross validation accuracy for the dataset.

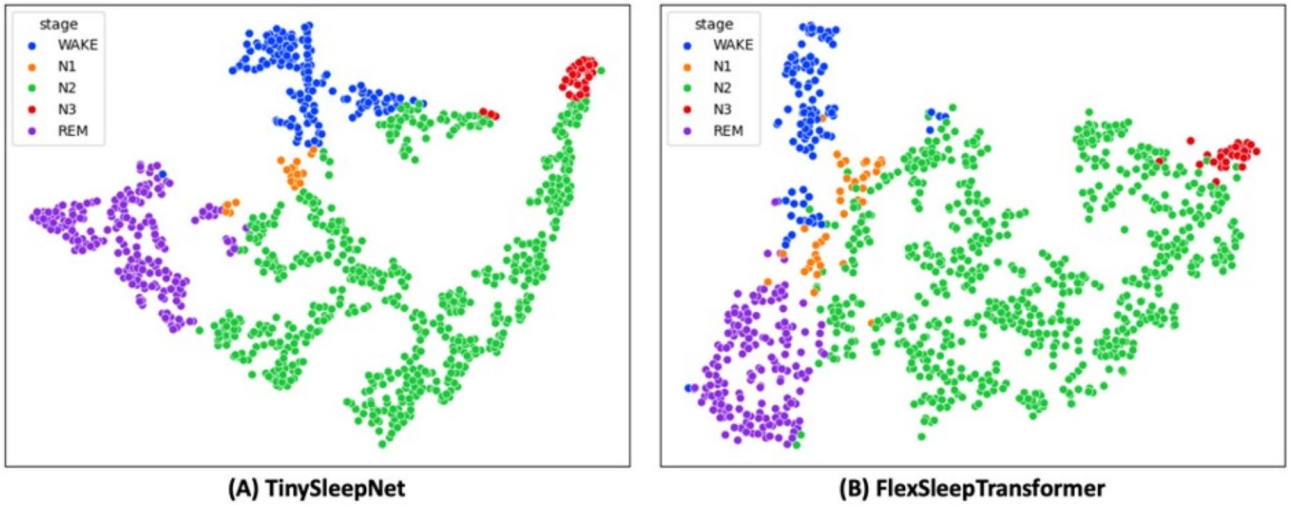

Fig. 2. t-SNE visualization of FlexSleepTransformer and TinySleepNet performance on the SleepUHS dataset. Panel (**A**) shows the t-SNE visualization of the 5-element vector before softmax for the predicted clusters using TinySleepNet. Panels (**B**) displays the corresponding visualization for FlexSleepTransformer.

Database	Method	Channel	Overall metrics			Class-wise MF1				
			Accuracy	Cohen's kappa	MF1	W	N1	N2	N3	REM
SleepEDF-78	TinySleepNet	Single[a]	80.4	72.54	71.8	91.79	31.6	83.9	80.43	71.25
		Multiple[b]	81.8	74.59	74.73	91.99	39.38	84.08	79.76	78.41
	FlexSleepTransformer/SSE	Single	80.37	72.75	73.29	91.93	38.81	82.84	78.37	74.52
		Concatenation[c]	81.16	73.8	74.46	91.85	42.11	82.92	76.16	79.26
		Random fusion[d]	81.95	74.86	75.56	92.27	43.53	83.51	77.35	81.17
		Mixed dataset[e]	81.27	73.99	74.69	91.87	42.23	83.05	76.32	80
	FlexSleepTransformer/TF	Single	82	74.96	75.55	92.07	42.6	84.56	**79.84**	78.7
		Concatenation	82.46	75.57	76.33	91.89	44.9	84.26	77.28	83.34
		Random fusion	**83.02**	**76.34**	**76.78**	**92.62**	**44.99**	**84.59**	77.8	**83.89**
		Mixed dataset	81.32	74.04	75.23	91.77	42.89	82.36	76.53	82.59
SleepUHS	TinySleepNet	Single	79.48	67.11	61.16	84.04	12.74	86.52	46.8	75.69
		Multiple	81.67	**71.14**	**68.93**	85.08	26.98	**88.07**	**64.05**	80.46
	FlexSleepTransformer/SSE	Single	78.27	65.41	62.17	82.78	20.8	86.22	46.8	74.29
		Concatenation	80.25	68.39	65.45	84.94	23.52	87.02	52.88	78.86
		Random fusion	81.69	70.88	67.37	85.98	27.15	87.81	52.76	**83.15**
		Mixed dataset	80.75	69.52	65.86	85.48	26.64	87.14	46.96	83.08
	FlexSleepTransformer/TF	Single	76.59	61.91	58.02	80.29	18.54	85.41	38.18	67.66
		Concatenation	80.98	69.64	66.77	85.29	**27.72**	87.34	52.65	80.83
		Random fusion	**81.83**	71.05	68.37	**86.07**	26.48	87.68	58.94	82.67
		Mixed dataset	80.2	68.92	66.41	84.43	27.97	86.46	50.63	82.54

Table 1. The performance comparison between CNN & RNN-based model TinySleepNet and variants of FlexSleepTransformer using the same cross-validation subject grouping in term of different input channel arrangements. The models with the highest accuracy values for each dataset are highlighted in bold. [a]Single channel refers to the Fpz-Cz EEG in the SleepEDF-78 dataset and the F3-A2 EEG in the SleepUHS dataset. [b]Multiple channels involves using more than one channels, i.e., two channels (Fpz-Cz EEG and horizontal EOG) in the SleepEDF-78 dataset and three channels (F3-A2 EEG, F4-A1 EEG and ROC-A2 EOG) in the SleepUHS dataset. [c]Concatenation entails using the concatenated multi-channel sub-epoch input, while [d]random fusion involves using the randomly fused multi-channel sub-epoch input. The concept of a [e]mixed dataset involves training with both SleepEDF-78 and SleepUHS, utilizing randomly fused multi-channel sub-epoch input. It is worth noting that TinySleepNet, due to inability to process the sub-epoch input, doesn't yield results for concatenation and random fusion.

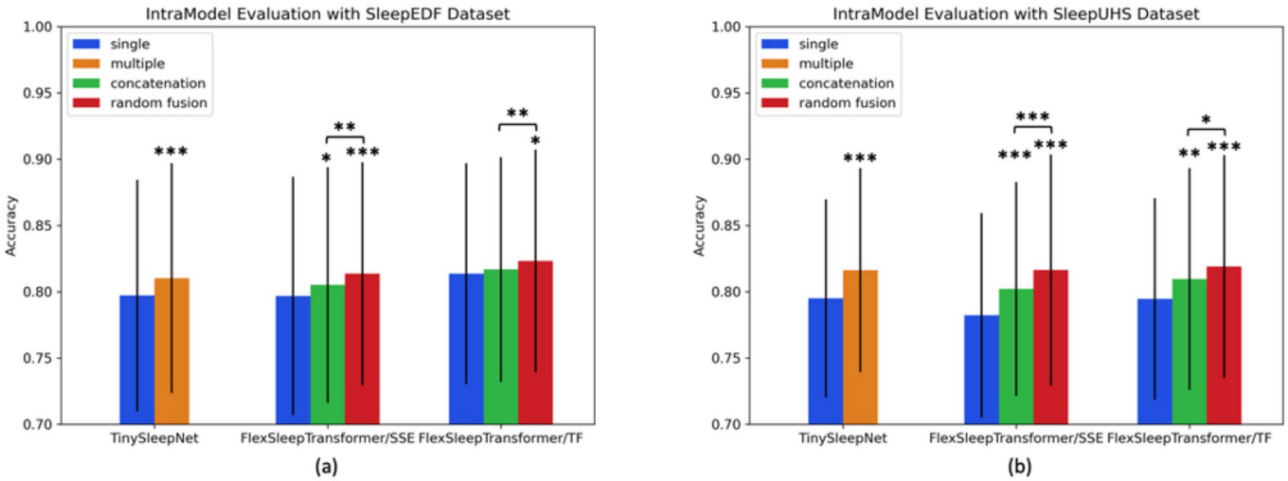

Fig. 3. Intra-model comparison of single-channel, multi-channel, multi-channel concatenation, multi-channel random fusion inputs using (**a**) the SleepEDF-78 dataset and (**b**) the SleepUHS dataset. Ten-fold subject-wise cross-validation was used for evaluation. Across both datasets, the accuracy of multi-channel input (depicted by the orange bar for multi-channel, green bar for multi-channel concatenation or red bar for multi-channel random fusion), except the multi-channel concatenation using the FlexSleepTransformer/TF model for the SleepEDF-78 dataset, is significantly higher than the single-channel input (depicted by the blue bar for 'single'); the accuracy of multi-channel concatenation is significantly higher than multi-channel concatenation for both the SleepEDF-78 and SleepUHS datasets. Single means using the single channel input, Fpz-Cz EEG in the SleepEDF-78 dataset and the F3-A2 EEG in the SleepUHS dataset. Multiple means using the multi-channel input, the Fpz-Cz EEG and horizontal EOG in the SleepEDF-78 dataset and the F3-A2 EEG, F4-A1 EEG and ROC-A2 EOG in the SleepUHS dataset. Concatenation and random fusion are two methods for processing sub-epochs in the FlexSleepTransformer with multi-channel input. Concatenation means using concatenated multi-channel sub-epoch input. Random fusion means using randomly fused multi-channel sub-epoch input. Single-channel accuracy serves as the reference within each model. The significance, in comparison to single-channel accuracy, is denoted by asterisks directly above the error bar. Significance between other pairs of comparisons is depicted separately with connecting lines. Non-significant differences are not shown. *Indicates significance at the $0.01 \leq p < 0.05$ level. **Indicates significance at the $0.001 \leq p < 0.01$ level. ***Indicates significance at the $p < 0.001$ level.

approaches outperforming single-channel input. This highlights the advantage of using multiple channels in FlexSleepTransformer models, especially with the random fusion method.

Inter-model comparison

Figure 4 presents the subject-wise cross-validation accuracy comparison of three models, TinySleepNet, FlexSleepTransformer/SSE, and FlexSleepTransformer/TF on two datasets, focusing on inter-model significance. This comparison involves assessing different input channel types (single channel or multiple channels) across the models.

On the SleepEDF-78 dataset, FlexSleepTransformer/TF with single-channel input and multi-channel random fusion input achieves significantly higher accuracy compared to TinySleepNet with the same input channels. Additionally, FlexSleepTransformer/TF with single-channel input, multi-channel concatenation input, and multi-channel random fusion input demonstrates significantly higher accuracy compared to FlexSleepTransformer/SSE with the corresponding input channels. On the SleepUHS dataset, FlexSleepTransformer/TF with single-channel input shows significantly higher accuracy compared to FlexSleepTransformer/SSE with single-channel input. The performance of FlexSleepTransforme/TF input is slightly higher than FlexSleepTransformer/SSE and TinySleepNet for multi-channel input but does not reach statistical significance. Overall, FlexSleepTransformer/TF with multi-channel random fusion input outperforms both TinySleepNet with multi-channel input and FlexSleepTransformer/SSE with random fusion input. This underscores the benefits of using the transformer-based model and incorporating the additional frequency information.

Multi-dataset training and testing

Figure 5 presents the FlexSleepTransformer's transfer learning ability using multiple datasets during the training and testing. When utilizing two datasets (referred to as the mixed dataset in Fig. 3) in the training of FlexSleepTransformer with multi-channel random fusion input, we observed 98% of the accuracy, a slight decrease in test accuracy on both datasets compared to models trained and tested on the same dataset (referred to as learning from scratch, LFS, in Fig. 3). However, the accuracy using the mixed dataset in the training remained significantly superior compared to the direct transfer (referred to as DT in Fig. 3), where the model is trained on the SleepEDF-78 dataset and tested on the SleepUHS dataset, or vice versa. FlexSleepTransformer enables simultaneous training on multiple datasets with differing number of PSG channels. This capability allows it to

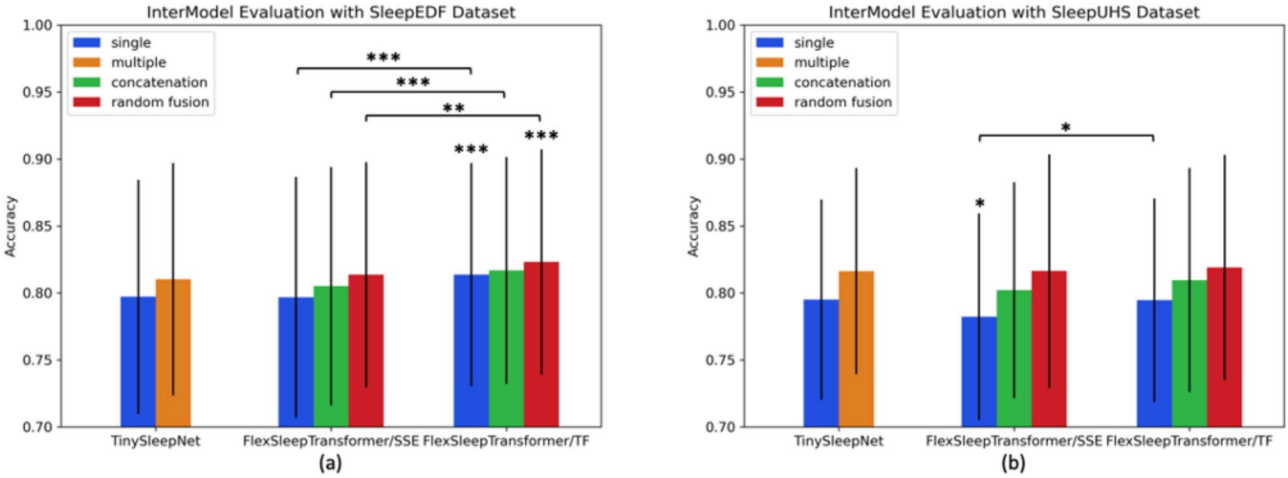

Fig. 4. Inter-model comparison of TinySleepNet, FlexSleepTransformer/SSE, and FlexSleepTransformer/TF using (**a**) the SleepEDF-78 dataset and (**b**) the SleepUHS dataset. Ten-fold subject-wise cross-validation was used for evaluation. FlexSleepTransformer/TF input significantly outperforms both FlexSleepTransformer/SSE input and TinySleepNet for both single and multi-channel input on the SleepEDF-78 dataset. On SleepUHS dataset, FlexSleepTransformer/TF significantly outperforms FlexSleepTransformer /SSE for single channel input. The performance of FlexSleepTransformer/TF is slightly higher than FlexSleepTransformer/SSE input and TinySleepNet for multi-channel input but does not reach statistical significance. Single means using the single channel input, Fpz-Cz EEG in the SleepEDF-78 dataset, and the F3-A2 EEG in the SleepUHS dataset. Multiple means using the multi-channel input, the Fpz-Cz EEG and horizontal EOG in the SleepEDF-78 dataset and the F3-A2 EEG, F4-A1 EEG and ROC-A2 EOG in the SleepUHS dataset. Concatenation and random fusion are two methods for processing sub-epochs in the FlexSleepTransformer with multi-channel input. Concatenation means using concatenated multi-channel sub-epoch input. Random fusion means using randomly fused multi-channel sub-epoch input. For inter-model comparison, TinySleepNet serves as the reference. Significance, denoted by asterisks directly above the error bar, shows comparison with the same input channel type but from a different model (i.e., FlexSleepTransformer with TF input). TinySleepNet with multi-channel input is compared with the FlexSleepTransformer with both multi-channel concatenation and multi-channel random fusion in the inter-model comparison. The significance between FlexSleepTransformer/SSE and FlexSleepTransformer/TF is shown separately with a connecting line. Non-significant differences are not shown. * refers to $0.01 \leq p < 0.05$. ** refers to $0.001 \leq p < 0.01$. *** refers to $p < 0.001$.

train on a variety of sleep datasets from different PSG acquisition devices and environments, greatly enhancing its potential for clinical adoption.

Comparison with existing models

To compare FlexSleepTransformer with existing models, we add its accuracy (FlexSleepTransformer/TF with multi-channel random fusion) alongside other methods in Table 2. FlexSleepTransformer achieves among the highest accuracy, surpassing SleepTransformer. Our direct comparison (see Table 1 for accuracy of FlexSleepTransformer/TF with multi-channel random fusion vs. single channel) confirms that FlexSleepTransformer outperforms SleepTransformer. We noticed slightly higher accuracy from TinySleepNet and MultichannelSleepNet. The higher accuracy reported from the original TinySleepNet may benefit from its adopted data augmentation technique. From our direct comparison, superior performance of FlexSleepTransformer than TinySleepNet is evident with the same input data and the same preprocessing technique. Although we did not implement MultichannelSleepNet for direct comparison, FlexSleepTransformer's sequence-to-sequence design theoretically provides an advantage due to additional context from sequence input. The higher accuracy of MultichannelSleepNet may be due to its data preprocessing method, which normalizes the time-frequency image to zero mean and unit variance. Further direct comparisons with MultichannelSleepNet are necessary to fully assess relative advantages and disadvantages of their model architectures.

Benefits of our proposed model

This study introduces FlexSleepTransformer, a transformer-based model tailored for multi-channel sleep staging. It is designed to handle varying number of input channels across different sleep centers. The results highlight several key insights regarding its performance across various configurations and comparative analyses. In intra-model comparisons, the multi-channel input consistently outperformed the single-channel input, with the randomly fused input delivering the best performance. This suggests that random fusion allows for a more effective integration of multi-channel information compared to the concatenated input. Inter-model comparisons further reinforced the superiority of FlexSleepTransformer. FlexSleepTransformer with TF input outperformed both FlexSleepTransformer with SSE input and the CNN & RNN-based TinySleepNet model, particularly on the SleepEDF-78 dataset. While the performance differences were less significant in the SleepUHS

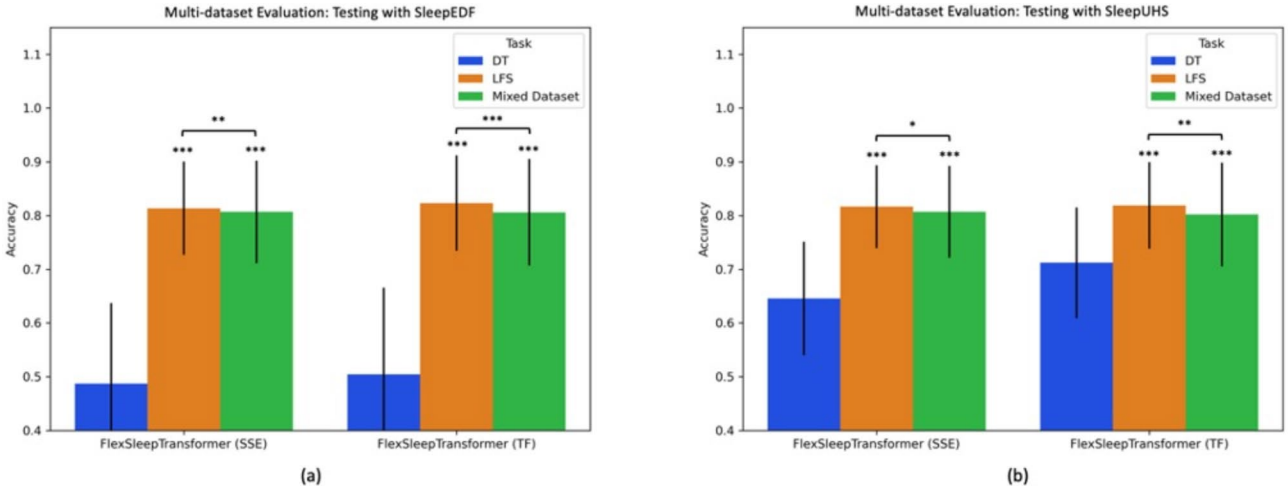

Fig. 5. Intra-model comparison of direct transfer (DT), learning from scratch (LFS), and mixed dataset approaches for FlexSleepTransformer/SSE and FlexSleepTransformer/TF using (**a**) the SleepEDF-78 dataset and (**b**) the SleepUHS dataset as test datasets. FlexSleepTransformer with multi-channel random fusion was used for all three training strategies. Ten-fold subject-wise cross-validation was used for evaluation. When mixing two datasets in the training, we observed 98% of the accuracy, a slight decrease in accuracy on both datasets compared to LFS models trained and tested on the same dataset. However, the accuracy using the mixed dataset in the training remained significantly superior compared to the DT, where the model is trained on one dataset and tested on the other dataset. DT means training on one dataset and testing on the other dataset. LFS means training and testing on the same dataset. Mixed dataset means training and testing with combined datasets, specifically using 9 folds from both SleepEDF-78 and SleepUHS for training, and 1 fold from each dataset for testing. DT serves as the reference within each model. The significance, in comparison to DT accuracy, is denoted by asterisks directly above the error bar. Significance between other pairs of comparisons is depicted separately with connecting lines. * refers to $0.01 \leq p < 0.05$. ** refers to $0.001 \leq p < 0.01$. *** refers to $p < 0.001$.

Model/Author	Year	Signal	Method	Evaluation	Accuracy (%)
DeepSleepNet[18] (Supratak et al.)	2017	EEG (1 channel)	CNNs + BiLSTM	5-fold CV	78.50
SeqSleepNet[24] (Phan et al.)	2019	EEG & EOG (multiple channel)	End-to-End Hierarchical RNN	5-fold CV	78.20
RobustSleepNet[29] (Guillot et al.)	2021	EEG & EOG (multiple channel)	Attention Mechanism, Transfer learning	5-fold CV	76.30
AttnSleep[25] (Eldele et al.)	2021	EEG (1 channel)	Multi-resolution CNN Adaptive feature recalibration (AFR) Temporal context encoder (TCE)	20-fold CV	81.30
Jadhav et al.[23]	2022	EEG (1 channel)	1D-CNN, SWT-CNN, STFT-CNN	10-fold CV	82.50
EEGSNet[22] (Li et al.)	2022	EEG (1 channel)	CNNs + BiLSTM	20-fold CV	83.02
MRASleepNet[30] (Yu et al.)	2022	EEG (1 channel)	feature extraction (FE) module multi-resolution attention (MRA) module gated multilayer perceptron (gMLP) module	10-fold CV	81.40
TinySleepNet[19] (Supratak et al.)	2020	EEG (1 channel)	CNN + LSTM, Data augmentation	10-fold CV	83.10
Zhu et al.[27]	2020	EEG (1 channel)	CNNs + Attention	20-fold CV	82.80
SleepTransformer[31] (Phan et al.)	2022	EEG (1 channel)	Transformer	10-fold CV	81.40
MultiChannelSleepNet[32] (Dai et al.)	2023	EEG & EOG (multiple channel)	Transformer	10-fold CV	83.80
FlexSleepTransformer (Ours)	2024	EEG & EOG (multiple channel)	Transformer, flexible input	10-fold CV	83.02

Table 2. Performance comparison between previous methods and FlexSleepTransformer on the SleepEDF-78 dataset.

dataset, FlexSleepTransformer continued to demonstrate a notable advantage. In multi-dataset training scenarios, FlexSleepTransformer achieved 98% of accuracy (a slight decrease in accuracy) compared to single-dataset training. Nonetheless, its performance remained markedly superior to direct transfer, underscoring its strong capability to generalize across diverse datasets with varying channel configurations. In summary,

FlexSleepTransformer is the first model designed to train simultaneously on datasets with varying numbers of input channels. Its flexibility and robust performance across multiple datasets and input configurations make it a promising candidate for clinical adoption, particularly when trained with data from a broad spectrum of clinical sleep centers.

Extending FlexTransformer to magnetoencephalography (MEG) data

Due its superior temporal and spatial resolution, the spectral features of MEG signals have been extensively linked to specific sleep events, such as sleep spindles[35], cognitive changes during sleep[36], and general sleep stages[37–39]. Recently, Brancaccio et al. demonstrated that those spectral features of brain MEG activity vary with different sleep stages and across various brain regions[37]. This suggests that MEG signals in frequency domain, especially from multiple brain locations, provide crucial information about sleep staging. Given the capability of the FlexSleepTransformer to address variations in EEG channels across different sleep centers, our model could be extended to explore the feasibility of using MEG signals from various locations for sleep staging. Additionally, it has the potential to integrate EEG and MEG signals, thereby enhancing the accuracy of sleep staging and deepening our understanding of sleep phenomena.

Limitations and future research

We evaluated the feasibility of our proposed FlexSleepTransformer using only two datasets to assess its performance with varying numbers of sleep channels and its transfer learning capabilities. Our focus was primarily on enhancing the model architecture, without exploring the benefits of data preprocessing and data augmentation techniques. Future work will aim to further explore the transfer learning potential of FlexSleepTransformer by incorporating additional sleep datasets from diverse sleep centers. Considering the variations in bio-signals arising from different sleep measurement equipment and settings across sleep centers, we will investigate the feasibility of building a truly generalizable model capable of sleep staging on data from previously unseen sleep centers. Additionally, we will investigate methods for improving performance through advanced data preprocessing and augmentation techniques.

Conclusion

We proposed the FlexSleepTransformer/TF with multi-channel random fusion input as a promising model for clinical sleep staging. It accommodates a flexible number of PSG channels and demonstrates significant improvements over existing methods. FlexSleepTransformer/TF with multi-channel random fusion input notably outperforms SleepTransformer (the single-channel version of FlexSleepTransformer/TF) with statistical significance ($p < 0.05$ for the SleepEDF-78 dataset and $p < 0.001$ for the SleepUHS dataset). It also surpasses TinySleepNet with multi-channel input, a CNN & RNN-based model, with $p < 0.001$ for the SleepEDF-78 dataset. FlexSleepTransformer is the first model designed to train on multiple sleep datasets with varying numbers of PSG channels from different sleep centers. When trained on multiple datasets simultaneously, FlexSleepTransformer achieves 98% accuracy — a slight drop compared to single dataset training — but significantly outperforms direct transfer, where training and testing are done on different datasets ($p < 0.001$ for both the SleepEDF-78 and SleepUHS datasets). Future research should focus on improving performance through advanced data preprocessing and augmentation techniques, as well as exploring the model's generalizability across diverse datasets from a variety of clinical sleep centers.

Methods

Database

SleepEDF-78

The SleepEDF-78 is from the study Sleep-Cassette[40,41], a publicly available dataset. It includes 153-night records from 78 healthy subjects from 25 to 101 years old. The first nights of subjects 36 and 52, and the second night of subject 13, were lost due to the failing cassette. The study was scored using the R&K standards and we merged the N3 and N4 stages into N3 stage to be comparable with other studies using the AASM guideline. To only consider the in-bed time, only 30 min of wake epochs were included before fall asleep and after waking up. We used the Fpz-Cz EEG and horizontal EOG in this study.

SleepUHS

The SleepUHS data is from a local sleep center of United Health Services (UHS) hospitals. The SleepUHS data was obtained following IRB approval with IRB00003573, approved on 5/15/2021. All methods were performed in accordance with the relevant guidelines and regulations. It contains 107-night records for 97 patients aged 24 to 84 with sleep disorders. 83.8% of patients received a confirmed diagnosis of sleep apnea. The scoring of the study followed the AASM standards by two sleep experts. Each sleep record was scored by one sleep expert. To focus solely on in-bed time, only 30 min of wake epochs were included before fall asleep and after waking up. The SleepUHS data comprises a comprehensive set of PSG. However, for the proof of a concept, we utilized only the F3-A2 EEG, F4-A1 EEG and ROC-A2 EOG in this study.

FlexSleepTransformer

Figure 6 illustrates the comprehensive architecture of the proposed model, FlexSleepTransformer. It contains a two-level transformer-based sequence-to-sequence deep learning model for sleep stage classification, closely following the SleepTransformer framework. O_1 to $O_M \in \mathbb{R}^5$ represent the predicted sleep stages for the corresponding consecutive input sequence of sleep epochs T_1 to $T_M \in \mathbb{R}^{N \times F}$. Initially, each sleep epoch input $T_i (i = 1, \cdots, M)$ undergoes positional and channel encoding and is then directed to the epoch-level

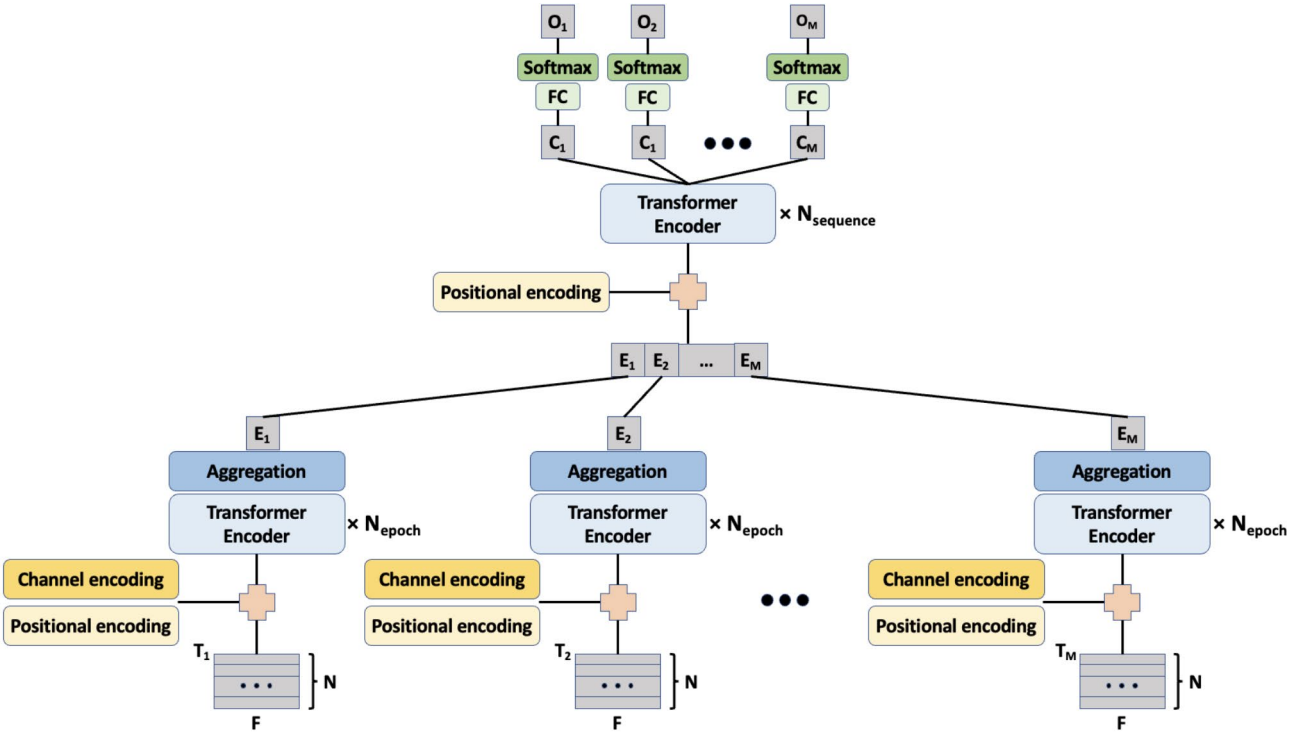

Fig. 6. The architecture of our proposed FlexSleepTransformer. It is a sequence-to-sequence transformer-based sleep staging model. O_1 to $O_M \in \mathbb{R}^5$ are the predicted sleep stages for the corresponding consecutive input sequence of sleep epochs T_1 to $T_M \in \mathbb{R}^{N \times F}$. In the time domain input, each sleep epoch (30s, comprising 3000 time points) is separated into N sub-epochs ($N = 30$), each having a length of F (F = 100 time points), as elaborated in sequence-to-sequence input from a single channel part. The FlexSleepTransformer contains a two-level transformer encoder, including both epoch-level and sequence-level encoders. Each sleep epoch input is encoded with positional and channel information first and sent to the epoch-level transformer encoder (a stack of N_{epoch} consistent transformer encoders, as detailed in transformer encoder & aggregation layer part) and aggregation layer (used to aggregate sub-epoch features) to get the epoch-level feature $E_i \in \mathbb{R}^F\ (i = 1, \cdots, M)$. The sequence of epoch-level features is encoded with positional information and then sent to the sequence-level transformer encoder (a stack of $N_{sequence}$ consistent transformer encoders, as detailed in transformer encoder & aggregation layer part) to further learn the neighboring sleep epochs information, the output of the sequence-level transformer encoder C_1 to $C_M \in \mathbb{R}^F$ is sent to the fully connected (FC) layer and Softmax layer to get the final predicted sequence of sleep stages O_1 to O_M.

transformer encoder—a stack of N epoch-consistent transformer encoders, as detailed in following transformer encoder & aggregation layer part. Subsequently, an aggregation layer is employed to aggregate sub-epoch features, yielding the epoch-level feature $E_i \in \mathbb{R}^F\ (i = 1, \cdots, M)$. The epoch-level features are concatenated and encoded with positional information and subsequently fed into the sequence-level transformer encoder—a stack of N sequence-consistent transformer encoders, as outlined in following transformer encoder & aggregation layer part. This step aims to enhance the model's understanding of contextual dependency from the adjacent sleep epochs. The resulting output from the sequence-level transformer encoder, denoted as C_1 to $C_M \in \mathbb{R}^F$, is then directed to both the fully connected (FC) layer and the Softmax layer. This flow enables the model to generate the final predicted sequence of sleep stages, represented as O_1 to O_M. In the FlexSleepTransformer, we innovatively introduce an efficient multi-channel fusion approach, detailed in following enabling multi-channel Sequence-to-sequence sub-epoch input part, designed to accommodate inputs from varying numbers of channels. Drawing inspiration from a computer vision model named Masked Autoencoder (MAE)[42], this design enhancement empowers the transformer-based sleep staging model to handle a multi-channel input with different numbers of channels, resulting in significantly improved performance compared to a straightforward implementation.

Transformer encoder and aggregation layer

We follow the same design as the original transformer encoder[26] (Fig. 7). Figure 7a illustrates the multi-head attention mechanism with H heads. The inputs consist of query, key, and value vectors denoted as Q, K and V, respectively, where each of $Q,\ K,\ V \in \mathbb{R}^{N \times F}$ represents a sequence of vectors. Within each vector sequence, individual vectors are partitioned into H segments (due to H heads). There segments are then linearly projected differently to yield parallel queries, keys, and values denoted as $Q_i,\ K_i,\ V_i \in \mathbb{R}^{N \times (\frac{F}{H})}$. The attention mechanism is subsequently applied to those parallel queries, keys, and values, following the formula:

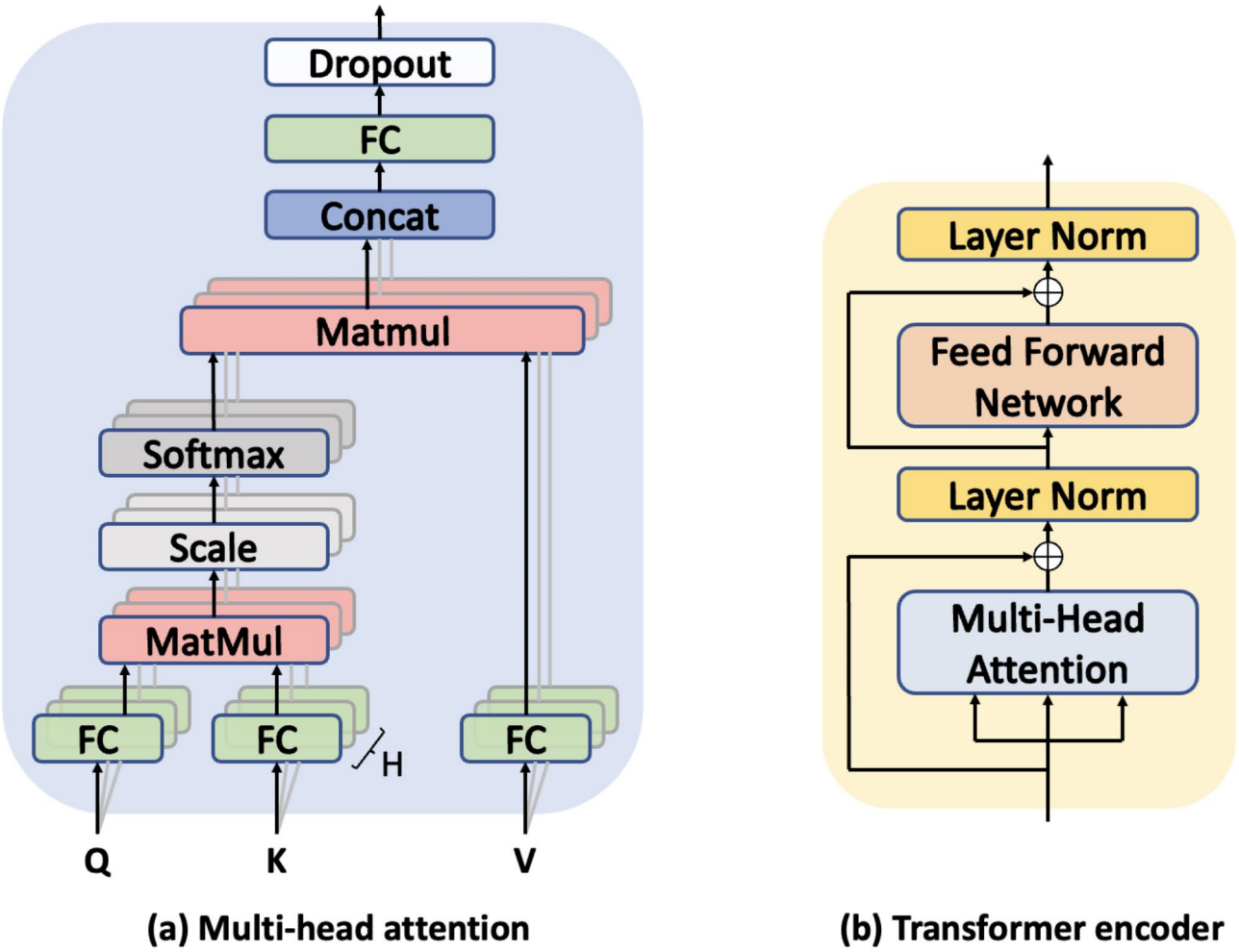

Fig. 7. Architecture of (**a**) multi-head attention and (**b**) transformer encoder. In multi-head attention, Query (Q), Key (K), and Value (V) matrices are computed from the input sequence. The attention score is calculated by comparing Q with K, determining the focus on different parts of the input (values). Multiple attention heads run in parallel, capturing various relationships between tokens, and their outputs are combined to create richer representations. The transformer encoder consists of stacked layers, each with Multi-Head Self-Attention, allowing tokens to attend to all others in the sequence and Feed-Forward Neural Network, refining each token's representation after attention processing.

$$Attention\,(Q_i,\ K_i,\ V_i) = softmax\left(\frac{Q_i {K_i}^T}{\sqrt{F}}\right) V_i$$

Then, the attention results from the H heads are concatenated and mapped to the same dimension as the input vector by another linear layer and processed through a dropout layer. Figure 7b illustrates the architecture of the transformer encoder. It comprises a multi-head attention layer, a feed forward layer (consisting of two linear layers separated by ReLU activation) and two residual and normalization layers. The transformer encoder processes a sequence of vectors by duplicating it into three copies, serving as input (Q, K, V) for the multi-head attention layer. Those copies undergo the multi-head attention process to capture dependencies among the vectors. The element-wise addition of input and the output from the multi-head attention layer is then directed to the layer normalization and the subsequent feed forward network. The element-wise addition of input and the output from feed forward network is further processed by the last layer normalization.

The aggregate layer in Fig. 4 is an attention-based feature aggregation layer designed to compress a sequence of vectors $X \in \mathbb{R}^{N\times F}$ into a single vector $\widetilde{X} \in \mathbb{R}^{F}$ via a weighted combination, employing an attention weight for each vector:

$$\widetilde{X} = X^{\mathrm{T}}\alpha$$

Where $\alpha \in \mathbb{R}^{N}$ is the attention weight vector learned by a softmax attention layer, as outlined in[24]:

$$\alpha_i = \frac{\exp\left(y_i{}^{\mathrm{T}} y_e\right)}{\sum_{i=0}^{N} \exp\left(y_i{}^{\mathrm{T}} y_e\right)}$$

$$y_i = \tanh(w_a X_i + b_a)$$

Where $w_a \in \mathbb{R}^{A\times F}$ and $b_a \in \mathbb{R}^{A}$ are a learnable weight matrix and bias. $y_e \in \mathbb{R}^{A}$ is the trainable epoch level context vector. A is the attention size.

Sequence-to-sequence input from a single channel

For the sequence-to-sequence input from a single channel, the FlexSleepTransformer model takes as input a temporal context of M successive sleep epochs (T_1, …, T_M, M = 10, in our implementation), aiming to predict the corresponding sleep stages for these M epochs. Each sleep epoch, treated as a 1D bio-signal (because of a single channel) with a fixed length, undergoes subdivision. Drawing inspiration from the Vision Transformer (ViT)[43] in computer vision, which divides a 2D image into small image patches as the input to the transformer-based image encoder, we apply a similar concept to the 1D bio-signal. Each sleep epoch is further segmented into sub-epochs (see below for detailed segmentation in the time domain or time-frequency domain). This subdivision serves as the input to the transformer-based 1D signal encoder. In either domain, a 30-s sleep epoch is divided into a sequence of N consecutive sub-epochs, each having a length of F. Consequently, for a single channel, the sequence-to-sequence input can be represented as $Z \in \mathbb{R}^{N\times F\times M}$.

Sub-sleep-epoch (SSE) input in the time domain

In the time domain, we partition a 30-s epoch sampled at 100 Hz (containing 3000 time points) into N successive SSEs ($N = 30$ in our implementation), each having a length of F sampling points ($F = 100$).

Time frames (TF) input in the time-frequency domain

To generate a 2D time-frequency image, a 30-second epoch sampled at 100 Hz (containing 3000 time points) was decomposed into two-second frames with a 50% overlap. These frames were then multiplied by a Hamming window and subjected to a 256-point short-time Fourier Transform (STFT) for frequency domain transformation. This process yields a 2D time-frequency image with $N = 29$ time frames and $F = 128$ frequency bins. Consistent with the approach in SleepTransformer, we exclude the 0-th frequency bin and maintain the same values of N and F for the time frames input.

Enabling multi-channel sequence-to-sequence sub-epoch input

To enable the multi-channel (for instance, C channels) sequence-to-sequence data $\mathbb{R}^{C\times N\times F\times M}$, we introduced an effective approach to fuse multiple channels, along with a simple concatenation method for comparison.

Concatenation

Concatenation serves as the straightforward method to facilitate multi-channel input for transformer-based sleep staging models. In this approach, for each epoch, we can stack each of C channels' 2D data $\mathbb{R}^{N\times F}$ on top of each other along the first dimension. This process yields the concatenated multi-channel sequence-to-sequence sub-epoch input $Z_C \in \mathbb{R}^{(C*N)\times F\times M}$. Figure 8 shows an example of concatenated multi-channel sub-epoch input in the time domain.

Random fusion

In the context of masked autoencoders (MAE) within the 2D image-domain transformer-based autoencoder, pre-training involves masking out a substantial random subset of image patches (e.g., 75%). Inspired by this approach, in our scenario for each sub-epoch (out of N sub-epochs), we randomly choose its signal from one of the available channels (C rows). By fusing the randomly chosen signals for each sub-epoch, as illustrated in Fig. 9, we derive the multi-channel sequence-to-sequence sub-epoch input $Z_{RF} \in \mathbb{R}^{N\times F\times M}$.

Positional encoding and channel encoding

As the transformer encoder doesn't consider the order of sub-epochs and the location of EEG electrodes for the bio-signal source, we introduced the positional encodings (PE) and channel encodings (CE) to each sub-epoch input $X \in \mathbb{R}^{N\times F}$.

$$\widetilde{X} = X + PE + CE$$

$$\text{Where } \widetilde{X} \in \mathbb{R}^{N\times F},\ PE \in \mathbb{R}^{N\times F},\ CE \in \mathbb{R}^{N\times F}.$$

We employ sine and cosine functions, following the seminal work[26], to encode positional and channel information. For positional encoding, where the i^{th} row and the $(2j)^{th}$ or the $(2j+1)^{th}$ column is specified, the encoding is determined as follows:

$$PE_{i,\,2j} = \sin\left(\frac{i}{10000^{\frac{2j}{F}}}\right)$$

$$PE_{i,\,2j+1} = \cos\left(\frac{i}{10000^{\frac{2j}{F}}}\right)$$

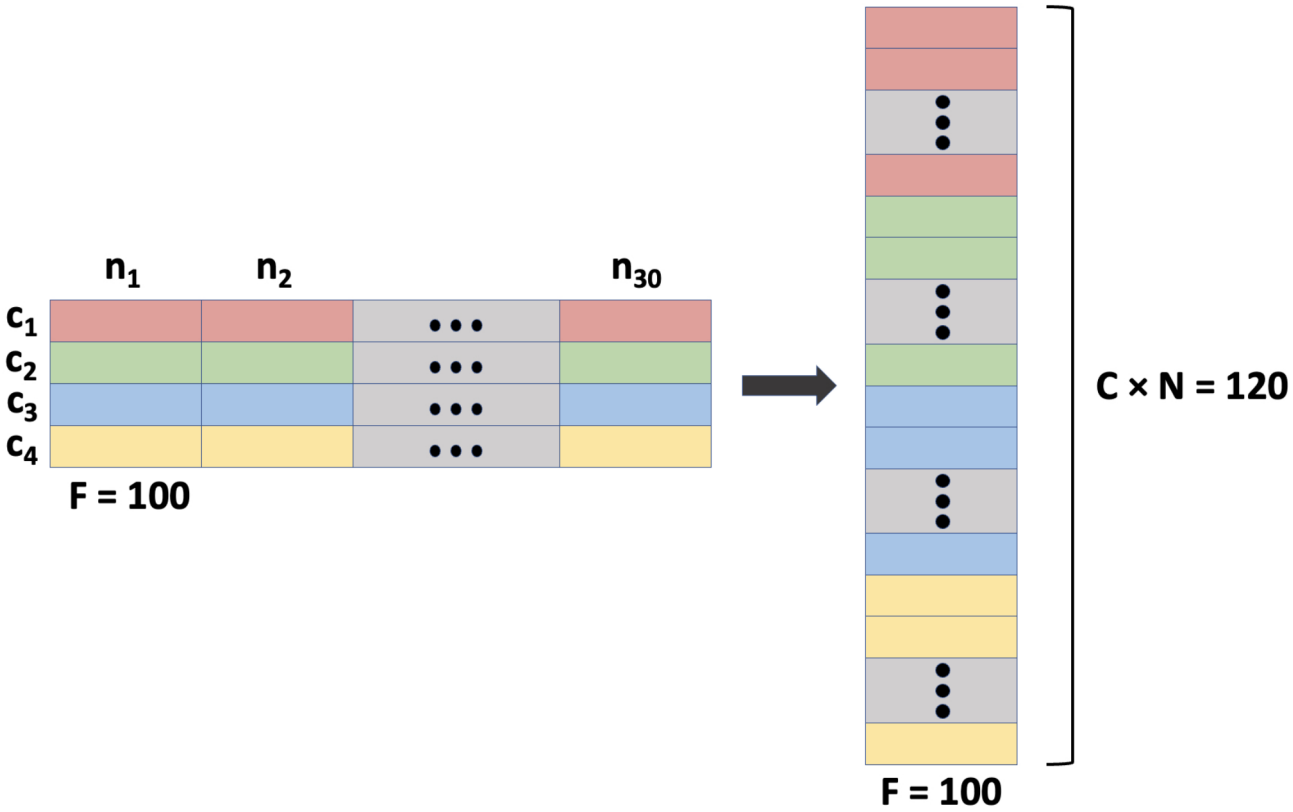

Fig. 8. The multi-channel sub-epoch concatenation in the time domain for a single sleep epoch. Left part of the figure illustrates a single epoch with multiple channels (C = 4), divided into N sub-epochs ($N = 30$), each having a length of F time points (F = 100). As shown in the right part, all the sub-epochs (n_1, n_2, …, n_{30}) of the first channel c_1 are stacked, followed by those of c_2, and so on. This results in a concatenated multi-channel sub-epoch input with C * $N = 120$ sub-epochs from a single sleep epoch.

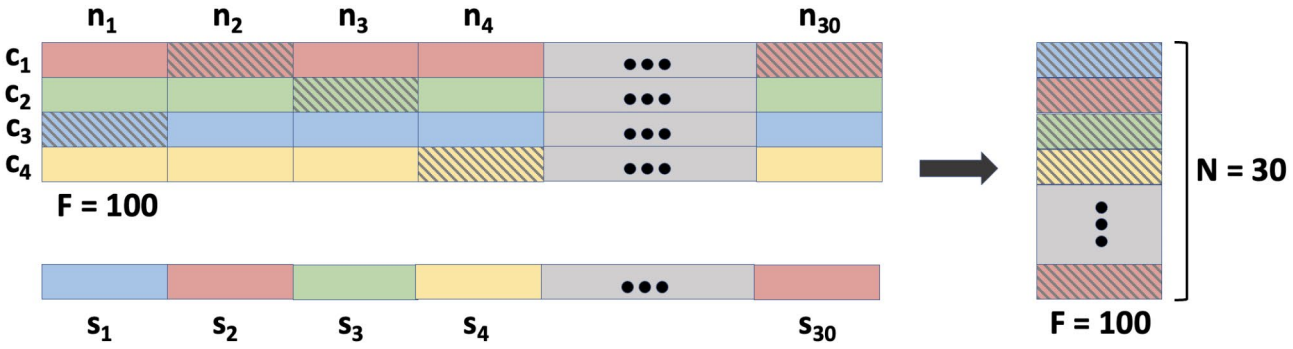

Fig. 9. The multi-channel sub-epoch random fusion in the time domain for a single sleep epoch. The upper left part of the figure illustrates a single epoch with multiple channels (C = 4), divided into N sub-epochs ($N = 30$), each having a length of F time points (F = 100). In each sub-epoch, a channel is randomly selected out of all available channels, denoted by S_1 to S_{30} with different colors. The selected sub-epochs are visually indicated with diagonal stripes. All the selected sub-epochs are then stacked in sequence, resulting in a randomly fused multi-channel sub-epoch input with $N = 30$ sub-epochs from a single sleep epoch.

Where i is the sub-epoch index, $0 \leq i < N$, and $2j$ or $2j+1$ is the relative index from the start of the sub-epoch, $0 \leq 2j \leq 2j+1 < F$.

Regarding channel encoding, we enumerate all the available C channels sequentially and apply the same formula used for positional encoding. This process yields the unified channel encoding $CE \in \mathbb{R}^{C \times F}$. Subsequently, the distinct encoding for each individual channel can be extracted from the unified channel encoding CE.

$$CE_{i,\ 2j} = \sin\left(\frac{i}{10000^{\frac{2j}{F}}}\right)$$

$$CE_{i,\ 2j+1} = \cos\left(\frac{i}{10000^{\frac{2j}{F}}}\right)$$

Where i is the channel index, $0 \leq i < C$, and $2j$ or $2j+1$ is the relative index from the start of the sub-epoch, $0 \leq 2j \leq 2j+1 < F$. For the channel encoding CE of a single-channel epoch, we use the same formula as above with the channel index i = 0.

Experimental setup

We employed nearly identical FlexSleepTransformer designs for both SSE and TF inputs, FlexSleepTransformer/SSE and FlexSleepTransformer/TF. The input epoch sequence consisted of a fixed number of consecutive epochs (M = 10) and the number of transformer layers for epoch-level and sequence-level transformer encoder was fixed to $N_{epoch} = N_{sequence} = 4$. For SSE input, number of attention heads in a transformer layer was fixed at $H_{SSE} = 10$, while for TF input, was fixed at $H_{TF} = 8$ because the dimension F (F_{SSE} = 100; F_{TF} = 128) for each epoch input $X \in \mathbb{R}^{N \times F}$ should be a multiple of attention heads. The number of hidden units in a feed forward layer was consistently set to $d = 200$.

We trained the FlexSleepTransformer using the cross-entropy loss function and the Adam (Adaptive Moment Estimation) optimizer with a learning rate of 1×10^{-4}. The momentum parameters were set to $B_1 = 0.9$ and $B_2 = 0.999$, with a minibatch size of 32. The model was trained for a maximum of 50 epochs, and we selected the model with the best validation accuracy for evaluation. The model was implemented using Pytorch and was trained on an Nvidia GeForce RTX 4090 GPU and Intel i7-8700k CPU, 3.7 GHz.

We implemented three training strategies, learning from scratch (LFS), mixed dataset, and direct transfer (DT), to assess the transfer learning capabilities of our proposed models, FlexSleepTransformer/SSE and FlexSleepTransformer/TF. FlexSleepTransformer with multi-channel random fusion was used for all three training strategies. It is worth noting that TinySleepNet cannot be compared with our proposed FlexSleepTransformer because it does not support varying numbers of PSG channels within the training data or between the training and testing dataset. LFS involves training and testing within a single dataset. To mitigate subject bias, we conducted a 10-fold subject-wise cross-validation for each dataset, allocating a fixed proportion (10%) of subjects for validation in each fold, while the remaining subjects participated in the training. In the mixed dataset approach, the model was trained and tested using 10-fold subject-wise cross-validation in LFS across both datasets with each fold comprising 10% of subjects from the SleepEDF-78 dataset and 10% of subjects from the SleepUHS dataset. For direct transfer (DT), the model underwent LFS training on one dataset (source dataset) but was tested on the other dataset (target dataset). we utilized the first fold model out of 10-fold cross-validation in LFS training on source dataset, and all the subjects from target dataset were included in the test set.

To compare the performance of FlexSleepTransformer with existing state-of-the-art methods, we listed the accuracy of these existing methods in Table 2. TinySleepNet achieves the highest accuracy among CNN and RNN-based methods. Therefore, we implemented it to benchmark against FlexSleepTransformer using the same datasets. To test our hypothesize that deep learning models utilizing multiple-channel PSG data can outperform those using single-channel PSG data, we evaluated the performance of FlexSleepTransformer with both multiple and single channels. The FlexSleepTransformer's transformer-based design allows each 30-second epoch to be divided into sub-epochs. We compared two methods of handling these sub-epochs for FlexSleepTransformer with multiple channels: one where sub-epochs are concatenated and anthother where they are randomly fused. Additionally, we assessed the performance of TinySleepNet with both multiple and single channels. Since TinySleepNet was designed for single-channel PSG data, the multiple-channel input was created by stacking data from all channels on top of each other (like RGB channels for 3-channel input). In this study, "single channel" refers to the Fpz-Cz EEG in the SleepEDF-78 dataset and the F3-A2 EEG in the SleepUHS dataset, while "multiple channels" includes the Fpz-Cz EEG and horizontal EOG in the SleepEDF-78 dataset and the F3-A2 EEG, F4-A1 EEG and ROC-A2 EOG in the SleepUHS dataset.

Data availability

After publishing our main findings, requests for data (email to wdai@binghamton.edu) will be evaluated on a case-by-case basis. Before sharing data, we will make sure that all data are free of identifiers that could directly or indirectly link information to an individual and that all sharing is compliant with institutional and IRB policies.

Received: 27 June 2024; Accepted: 11 October 2024
Published online: 01 November 2024

References

1. Bianchi, M. T., Cash, S. S., Mietus, J., Peng, C. K. & Thomas, R. Obstructive sleep apnea alters sleep stage transition dynamics. *PLoS One* **5**, e11356. https://doi.org/10.1371/journal.pone.0011356 (2010).
2. Mantua, J. et al. A systematic review and meta-analysis of sleep architecture and chronic traumatic brain injury. *Sleep. Med. Rev.* **41**, 61–77. https://doi.org/10.1016/j.smrv.2018.01.004 (2018).
3. Pallayova, M., Donic, V., Gresova, S., Peregrim, I. & Tomori, Z. Do differences in sleep architecture exist between persons with type 2 diabetes and nondiabetic controls? *J. Diabetes Sci. Technol.* **4**, 344–352. https://doi.org/10.1177/193229681000400215 (2010).
4. Siengsukon, C., Al-Dughmi, M., Al-Sharman, A. & Stevens, S. Sleep parameters, functional status, and time post-stroke are associated with offline motor skill learning in people with chronic stroke. *Front. Neurol.* **6**, 225. https://doi.org/10.3389/fneur.2015.00225 (2015).

5. Stefani, A. & Hogl, B. Sleep in Parkinson's disease. *Neuropsychopharmacology* **45**, 121–128. https://doi.org/10.1038/s41386-019-0448-y (2020).
6. Tsuno, N., Besset, A. & Ritchie, K. Sleep and depression. *J. Clin. Psychiatry* **66**, 1254–1269. https://doi.org/10.4088/jcp.v66n1008 (2005).
7. Zhang, F. et al. Alteration in sleep architecture and electroencephalogram as an early sign of Alzheimer's disease preceding the disease pathology and cognitive decline. *Alzheimers Dement.* **15**, 590–597. https://doi.org/10.1016/j.jalz.2018.12.004 (2019).
8. Iber, C., Ancoli-Israel, S., Chesson, A. L. & Quan, S. F. *The AASM manual for the scoring of sleep and associated events: rules, terminology and technical specifications*. (2007).
9. Rosenberg, R. S. & Van Hout, S. The American Academy of Sleep Medicine inter-scorer reliability program: sleep stage scoring. *J. Clin. Sleep. Med.* **9**, 81–87. https://doi.org/10.5664/jcsm.2350 (2013).
10. Dimitriadis, S. I., Salis, C. & Linden, D. A novel, fast and efficient single-sensor automatic sleep-stage classification based on complementary cross-frequency coupling estimates. *Clin. Neurophysiol.* **129**, 815–828. https://doi.org/10.1016/j.clinph.2017.12.039 (2018).
11. Gunes, S., Polat, K. & Yosunkaya, S. Efficient sleep stage recognition system based on EEG signal using k-means clustering based feature weighting. *Expert Syst. Appl.* **37**, 7922–7928 (2010).
12. Abdulla, S., Diykh, M., Laft, R. L., Saleh, K. & Deo, R. C. Sleep EEG signal analysis based on correlation graph similarity coupled with an ensemble extreme machine learning algorithm. *Expert Syst. Appl.* **138**, 112790 (2019).
13. Koley, B. & Dey, D. An ensemble system for automatic sleep stage classification using single channel EEG signal. *Comput. Biol. Med.* **42**, 1186–1195. https://doi.org/10.1016/j.compbiomed.2012.09.012 (2012).
14. Alickovic, E. & Subasi, A. Ensemble SVM method for automatic sleep stage classification. *IEEE Trans. Instrum. Meas.* **67**, 1258–1265 (2018).
15. Sharma, M., Goyal, D., Achuth, P. V. & Acharya, U. R. An accurate sleep stages classification system using a new class of optimally time-frequency localized three-band wavelet filter bank. *Comput. Biol. Med.* **98**, 58–75. https://doi.org/10.1016/j.compbiomed.2018.04.025 (2018).
16. Li, X. et al. HyCLASSS: a hybrid classifier for automatic sleep stage scoring. *IEEE J. Biomed. Health Inf.* **22**, 375–385. https://doi.org/10.1109/JBHI.2017.2668993 (2018).
17. Memar, P. & Faradji, F. A novel multi-class EEG-based sleep stage classification system. *IEEE Trans. Neural Syst. Rehabil Eng.* **26**, 84–95. https://doi.org/10.1109/TNSRE.2017.2776149 (2018).
18. Supratak, A., Dong, H., Wu, C., Guo, Y. & DeepSleepNet: A model for automatic sleep stage scoring based on raw single-channel EEG. *IEEE Trans. Neural Syst. Rehabil. Eng.* **25**, 1998–2008 (2017).
19. Supratak, A. & Guo, Y. TinySleepNet: an efficient deep learning model for sleep stage scoring based on raw single-channel EEG. *42nd Annual International Conference of the IEEE Engineering in Medicine*, 641–644 (2020).
20. Mousavi, S., Afghah, F., Acharya, U. R. & SleepEEGNet Automated sleep stage scoring with sequence to sequence deep learning approach. *PLoS One* **14**, e0216456. https://doi.org/10.1371/journal.pone.0216456 (2019).
21. Korkalainen, H. et al. Accurate deep learning-based sleep staging in a clinical population with suspected obstructive sleep apnea. *IEEE J. Biomed. Health Inf.* **24**, 2073–2081. https://doi.org/10.1109/JBHI.2019.2951346 (2020).
22. Li, C. et al. A deep learning method approach for sleep stage classification with EEG spectrogram. *Int. J. Environ. Res. Public. Health* **19**. https://doi.org/10.3390/ijerph19106322 (2022).
23. Jadhav, P. & Mukhopadhyay, S. Automated sleep stage scoring using time-frequency spectra convolution neural network. *IEEE Trans. Instrum. Meas.* **71**, 1–9 (2022).
24. Phan, H., Andreotti, F., Cooray, N., Chen, O. Y. & De Vos, M. SeqSleepNet: end-to-end hierarchical recurrent neural network for sequence-to-sequence automatic sleep staging. *IEEE Trans. Neural Syst. Rehabil Eng.* **27**, 400–410. https://doi.org/10.1109/TNSRE.2019.2896659 (2019).
25. Eldele, E. et al. An attention-based deep learning approach for sleep stage classification with single-channel EEG. *IEEE Trans. Neural Syst. Rehabil Eng.* **29**, 809–818. https://doi.org/10.1109/TNSRE.2021.3076234 (2021).
26. Vaswani, A. et al. Attention is all you need. Conference on *Neural Inf. Process. Syst. (NIPS)* (2017).
27. Zhu, T., Luo, W. & Yu, F. Convolution-and attention-based neural network for automated sleep stage classification. *Int. J. Environ. Res. Public. Health* **17**. https://doi.org/10.3390/ijerph17114152 (2020).
28. Qu, W. et al. A residual based attention model for EEG based sleep staging. *IEEE J. Biomed. Health Inf.* **24**, 2833–2843. https://doi.org/10.1109/JBHI.2020.2978004 (2020).
29. Guillot, A. & Thorey, V. RobustSleepNet: transfer learning for automated sleep staging at scale. *IEEE Trans. Neural Syst. Rehabil Eng.* **29**, 1441–1451. https://doi.org/10.1109/TNSRE.2021.3098968 (2021).
30. Yu, R., Zhou, Z., Wu, S., Gao, X. & Bin, G. MRASleepNet: a multi-resolution attention network for sleep stage classification using single-channel EEG. *J. Neural Eng.* **19**. https://doi.org/10.1088/1741-2552/aca2de (2022).
31. Phan, H. et al. SleepTransformer: automatic sleep staging with interpretability and uncertainty quantification. *IEEE Trans. Biomed. Eng.* **69**, 2456–2467. https://doi.org/10.1109/TBME.2022.3147187 (2022).
32. Dai, Y. et al. MultiChannelSleepNet: a transformer-based model for automatic sleep stage classification with PSG. *IEEE J. Biomed. Health Inf.* **27**, 4204–4215. https://doi.org/10.1109/JBHI.2023.3284160 (2023).
33. Chambon, S., Galtier, M. N., Arnal, P. J., Wainrib, G. & Gramfort, A. A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series. *IEEE Trans. Neural Syst. Rehabil Eng.* **26**, 758–769. https://doi.org/10.1109/TNSRE.2018.2813138 (2018).
34. Phan, H. et al. Towards more accurate automatic sleep staging via deep transfer learning. *IEEE Trans. Biomed. Eng.* **68**, 1787–1798. https://doi.org/10.1109/TBME.2020.3020381 (2021).
35. Klinzing, J. G. et al. Spindle activity phase-locked to sleep slow oscillations. *Neuroimage* **134**, 607–616. https://doi.org/10.1016/j.neuroimage.2016.04.031 (2016).
36. Wang, X., Inui, K., Qiu, Y. & Kakigi, R. Cortical responses to noxious stimuli during sleep. *Neuroscience* **128**, 177–186. https://doi.org/10.1016/j.neuroscience.2004.06.036 (2004).
37. Brancaccio, A., Tabarelli, D., Bigica, M. & Baldauf, D. Cortical source localization of sleep-stage specific oscillatory activity. *Sci. Rep.* **10**, 6976. https://doi.org/10.1038/s41598-020-63933-5 (2020).
38. Ioannides, A. A., Kostopoulos, G. K., Liu, L. & Fenwick, P. B. MEG identifies dorsal medial brain activations during sleep. *Neuroimage* **44**, 455–468. https://doi.org/10.1016/j.neuroimage.2008.09.030 (2009).
39. Ioannides, A. A., Liu, L., Poghosyan, V. & Kostopoulos, G. K. Using MEG to understand the progression of light sleep and the emergence and functional roles of spindles and K-complexes. *Front. Hum. Neurosci.* **11**, 313. https://doi.org/10.3389/fnhum.2017.00313 (2017).
40. Goldberger, A. L. et al. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *Circulation* **101**, E215–220. https://doi.org/10.1161/01.cir.101.23.e215 (2000).
41. Kemp, B., Zwinderman, A. H., Tuk, B., Kamphuisen, H. A. & Oberye, J. J. Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the EEG. *IEEE Trans. Biomed. Eng.* **47**, 1185–1194. https://doi.org/10.1109/10.867928 (2000).
42. He, K. et al. Masked autoencoders are scalable vision learners. *Conference on Computer Vision and Pattern Recognition (CVPR)* (2022).
43. Dosovitskiy, A. et al. An image is worth 16x16 Words: transformers for image recognition at scale. *The International Conference on Learning Representations (ICLR)* (2021).

Acknowledgements
This work is supported by National Science Foundation CMMI-2123061, National Institute on Aging R01AG066430, National Institute of Mental Health R21MH126260, and Transdisciplinary Areas of Excellence (TAE) seed grant at Binghamton University.

Author contributions
YG, and WD conceived and designed the experiments. YG performed the experiments and analyzed the data. YG and WD wrote the paper. MN contributed to the discussion and revised the manuscript.

Declarations

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to W.D.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.