AccSleepNet: An Axis-Aware Hybrid Deep Fusion Model for Sleep Stage Classification Using Wrist-Worn Accelerometer Data

Guanjie Huang1, Ye Yuan2, Guohong Cao1, Fenglong Ma1
1 Pennsylvania State University, United States
1{gzh8, gxc27, fenglong}@psu.edu
2 JD Intelligent Cities Research, China
2 yuanye48@jd.com

Abstract—Numerous people are suffering from sleep-related problems. To diagnose them, a prerequisite is to divide the polysomnography (PSG) data into different sleep stages. Thus, sleep stage classification is an essential step, but collecting PSG data is expensive, time-consuming, and even belated. To address this issue, using accelerometers that are widely used in smartwatches is treated as an alternative way to monitor people’s sleep conditions. However, the flexibility of deep learning models by purely using wrist-worn accelerometer data for sleep stage classification has not been investigated by researchers. To explore the answer, in this paper, we design a novel axis-aware hybrid fusion-based deep learning model, named AccSleepNet, which purely uses accelerometer data for the sleep stage classification task compared with state-of-the-art baselines. Moreover, an ablation study validates the necessity of leveraging three axes’ accelerometer data and the superiority of the designed axis-aware hybrid fusion mechanism.

Index Terms—sleep stage classification, deep learning, cross attention, accelerometer.

I. INTRODUCTION

Sleep is an essential component of everyday human lives as it keeps our bodies stable on both physical and emotional levels. Unfortunately, a large number of people are suffering from sleep-related problems around the world [1]. In clinical practice, doctors usually diagnose these disorders by analyzing sleep data, and classifying sleep stages is conducted as a prerequisite yet essential step.

According to the American Academy of Sleep Medicine (AASM) scoring rule, sleep is classified into four stages, i.e., N1-N3 (NREM) and REM (rapid eye movement) stages, which are determined by analyzing the Polysomnography (PSG) data. However, the collection of PSG is complicated, which requires the patient to stay at a clinic setup with video recording during the whole night under the monitoring of a clinical expert, and more than 20 sensors are attached to the patient’s body to collect a variety of physiological data (e.g., electroencephalogram (EEG), electrocardiogram (ECG), and pulse oximeter). Clinical experts then visually inspect the PSG data to score the sleep stages, which usually takes 2 to 4 hours for one night PSG data [2]. What is worse, patients usually visit the hospital only after they already notice their symptoms of sleep problems, which means that the diagnosis procedures are always belated. Although studies [3–6] have developed methods to classify sleep stages, they either use EEG or ECG data, which can not be easily collected in daily life without professional devices.

To address this issue, in this paper, we aim to purely use an accelerometer to identify different sleep stages. Existing work [7] only derives physical activity features from the accelerometer data, which are further considered as the inputs of traditional machine learning models such as k-nearest neighbors and random forest for detecting at most three sleep stages. However, it is hard to achieve satisfactory performance, especially when the number of classes increases to 5, i.e., Wake, N1, N2, N3, and REM. Besides, the accelerometer data are significantly different from the PSG data, which use three axes to measure acceleration. This leads to the failure of directly applying those deep learning approaches [8–11] for sleep stage classification using the PSG data.

Intuitively, 3-axis accelerometer data can be considered as a special type of multi-view data, and several multi-view learning approaches can be directly applied to aggregate 3-axis data. [12] concatenates the deep features from different views without considering interactions between them. [13] applies a 2D convolutional operation across different views to fuse the information, but it still ignores the interaction over temporal/spatial dimension. [14] calculates a similarity-based attention score, but such 3-axis data do not guarantee to have similar waveforms when certain activity happens. Thus, it is essential to design new mechanisms to deal with the unique 3-axis accelerometer data.

Towards this end, we propose a novel axis-aware hybrid fusion-based deep learning model, call AccSleepNet, which purely uses accelerometer data for the sleep stage classification.

1The source code of the proposed AccSleepNet can be found via https://bit.ly/3Q2Xmnz4
task as shown in Figure 1. In particular, the preprocessing module first preprocesses the raw accelerometer data axis by axis, which is then taken as the inputs of the proposed AccSleepNet model. A convolutional neural network (CNN) based feature extractor is used to learning a latent feature representation for each axis data. Then, since the accelerometer data from different axes mutually influence each other, a novel hybrid fusion mechanism is proposed to them collaboratively by considering the relationship between axes in order to automatically and efficiently classify different sleep stages via the classification module. We conduct experiments to show the effectiveness of the proposed AccSleepNet model and the utility of the wrist-worn accelerometer data for the sleep stage classification task. To sum up, the major contributions of this paper are as follows:  

- To the best of our knowledge, this is the first work to develop a deep learning model by purely using wrist-worn accelerometer data to identify five sleep stages (i.e., Wake, N1, N2, N3, and REM) with decent performance.  
- We propose a novel axis-aware hybrid fusion mechanism to learn deep features from the accelerometer data from its three axes collaboratively, which generates a self attention map and a cross-axis attention map to accentuate more significant features and suppress irrelevant features.  
- We demonstrate state-of-the-art classification performance on two publicly available datasets, Sleep-Accel and Newcastle-Accel, outperforming all other benchmark methods, including feature-based machine learning and deep learning ones. The results also reveal a valid connection between sleep stages and the wrist-worn accelerometer data, and it can be better captured by the deep features than the hand-crafted features.

II. RELATED WORK

Sleep Stage Classification. Most of the studies [15] adopt a single-channel EEG signal for sleep stage classification. Very few studies use the wrist-worn accelerometer to detect sleep problems. [7,16] employ traditional machine learning classifiers (e.g., KNN, RF) with hand-crafted features, and achieve decent performance for 2-class (Wake/Sleep) or 3-class (Normal/NREM/REM) sleep stage classification. Besides the wrist-worn accelerometer, several studies also develop an accelerometer attached to chest [17,18], abdomen [19], or head[20] for detecting sleep problems.

Some commercial sleep-tracking apps [21,22] have been developed based on accelerometer, microphones, and heart rate sensors, but they may not achieve satisfactory performance.

From the technical perspective, most studies require domain knowledge to extract the hand-crafted features. However, such a feature extraction process is quite challenging. Thus, several studies design the deep learning-based methods to automatically learn features from the raw data directly for sleep stages classification [3,4,23–25]. The deep learning-based methods usually achieve better performance than the traditional feature-based machine learning methods through the result comparison of these studies.

Multi-View Feature Learning & Attention. Another line of related work is the feature learning for multi-view data, which are also known as multi-modality or multi-variate data. [12] designs a deep learning approach to automatically exploit features from multi-variate PSG signals separately for predicting the sleep stages. [13] applies a 2D CNN to exploit the interactions of the frequency components of different views and an RNN to learn temporal relationships.

On the other hand, with the development of the attention mechanism, cross attention recently has been proposed to take advantage of the features derived from multi-view data, which generates an attention mask from different views collaboratively. [26] calculates attention maps for each pair of class features and queries sample features in order to highlight particular regions and improve the discrimination of the extracted features. [27] proposes to use a single view (LiDAR) to create an attention mask that regulates the spatial features of another modality (HSI). [14] proposes a Transformer-based cross-attention, which uses one view as key/value and another view as query to learn the relationships between them.

Different from all the aforementioned studies, in this paper, we propose a novel deep learning model and purely use accelerometer data for classifying sleep stages. We also design a novel axis-wise hybrid fusion mechanism to boost the classification performance.

III. METHODOLOGY

In this section, we first introduce how to preprocess the raw data and then provide the details of the proposed AccSleepNet model.
Sleep

Feature Extractor

attention map to accentuate more valuable features. Lastly, module, which generates a self attention map and a cross-axis of the three axes are passed to the axis-aware hybrid fusion ically learn the deep features. Second, the extracted features of each axis are fed to a feature extractor module to automat-ically learn the deep features for each axis. Since the ac-

celerometer data have three axes, we employ three feature extractors, i.e., $\mathcal{F}_x, \mathcal{F}_y, \mathcal{F}_z$, respectively. Each feature extractor consists of two convolutional blocks. As shown in Fig. 2, each block contains two one-dimensional (1D) convolutional layers ($\text{Conv}(\# \text{ of filters}, \text{ filter size}, \text{ stride})$), two batch normalization layers ($\text{BN}$), two ReLU activation layers ($\text{ReLU}$), one 1D max-pooling layer ($\text{MaxPool}(\text{pool size})$), and one dropout layer ($\text{Dropout}(\text{rate})$). The 1D convolutional layer can automatically extract the features using multiple different filters. Its output will be a stack of the extracted features from the all filters. We use stacking here, instead of flattening, since it can retain the relative temporal order of original features, which is essential for the subsequent axis-aware hybrid fusion.

Fig. 2: The architecture of our proposed AccSleepNet model. The feature extractors compute the deep features for each axis automatically. In the axis-aware hybrid fusion, the cross-axis attention component generates attention maps based on the other two axes and self attention component calculates attention maps based on itself. With these attention maps, the proposed hybrid fusion can enhance or suppress the learned features corresponding to the waveform from three axes collaboratively.

A. Preprocessing

To prepare the data for the subsequent deep learning model, the raw accelerometer data are first passed through a Finite Impulse Response (FIR) low-pass filter with a cutoff frequency of 5 Hz. It will reduce the noise to prevent the deep learning model from overfitting to them. Then, we resample the data to a lower sampling rate as the sampling rate has high impact on the battery life [28], which impairs the user experience and data collection. After that, the data are normalized by subtracting the mean and then dividing by its standard deviation. The normalization helps to reduce the individual difference from the data and thereby boost the prediction performance of the system.

B. AccSleepNet Architecture

Due to the importance of modeling interactions between these axes of the accelerometer data, we propose the AccSleepNet model to learn more effective features from three axes collaboratively. Specifically, the proposed model is made of three modules as shown in Fig. 2. First, the input data of each axis are fed to a feature extractor module to automatically learn the deep features. Second, the extracted features of the three axes are passed to the axis-aware hybrid fusion module, which generates a self attention map and a cross-axis attention map to accentuate more valuable features. Lastly, the attention-weighted features are sent to the classification module to predict the probabilities of target classes.

We assume the input $\mathcal{X} = \{x_1^n, x_2^n, x_3^n\}_{n=1}^N$ with the ground truth $\mathcal{Y} = \{y_i\}_{i=1}^n$, where $x = [x_1, x_2, \ldots, x_T]$ is the accelerometer data with time length $T$ after the preprocessing, $n$ represents the number of samples, and the superscripts denote the axis of the accelerometer. The label $y_i \in \{1, 2, \ldots, K\}$, where $K$ is the number of ground truth classes. To obtain the predicted label $\hat{y}$, $\mathcal{X}$ is passed through every module in the proposed AccSleepNet model that we will discuss as follows.

1) Feature Extractor: The feature extractor is used to extract the deep feature representation from each axis. Since the accelerometer data have three axes, we employ three feature extractors, $\mathcal{F}_x, \mathcal{F}_y, \mathcal{F}_z$, respectively.
The feature extractor can be represented as $\mathcal{F}(x; \theta_x)$, where $x$ denotes the input data and $\theta_x$ denotes the parameters of the module. After feature extraction, we can obtain the deep features $d_i \in \mathbb{R}^{T \times F}$. $T$ represents the length of each extracted feature and $F$ denotes feature dimension, which corresponds to the number of filters of the convolutional layer.

$$d_i^x = \mathcal{F}_x(x_i^x; \theta_{F_x}), d_i^y = \mathcal{F}_y(x_i^y; \theta_{F_y}), d_i^z = \mathcal{F}_z(x_i^z; \theta_{F_z}).$$  

(1)

2) Axis-Aware Hybrid Fusion: Since the contribution of each deep feature to the classification task is always not equivalent, we propose an axis-aware hybrid fusion mechanism to highlight the important features. Specifically, it contains two sub-components: cross-axis attention and self attention.

**Cross-Axis Attention.** The accelerometer is used to describe a movement by recording the component of acceleration on its 3 orthogonal axes simultaneously. Intuitively, these three axes mutually influence each other. Thus, the cross-axis attention is designed to accentuate features by considering the mutual effects between axes. To take X-axis as an example, we can obtain its cross-axis attention maps from Y-axis $A_i^{|y|} \in \mathbb{R}^{T \times F}$ and Z-axis $A_i^{|z|} \in \mathbb{R}^{T \times F}$ as follows:

$$A_i^{|y|} = \sigma(W_{xy}^{|y|} \circ d_i^y + b^{|y|}),$$

$$A_i^{|z|} = \sigma(W_{xz}^{|z|} \circ d_i^z + b^{|z|}),$$

(2)

where $\circ$ represents the convolutional operation, and $\sigma$ represents the sigmoid function. $W \in \mathbb{R}^{M \times F}$ and $b \in \mathbb{R}^F$ are the trainable parameters of the convolutional operation, and the superscript denotes the attention relationship. For example, “$|y|$” denotes the cross attention from Y-axis on X-axis. The deep features of the Y and Z axes are passed through convolutional operations with $M$ filter size and 1 stride. With the convolutional operation, the model generates the attention map by only taking into account the neighboring features instead of fully considering all the features, which may scatter the attention distribution through capturing the neighboring features and waveform patterns. We then calculate the normalized importance weights through a sigmoid function. The value of each element in the normalized attention map will be in the range between 0 and 1. Thus, the normalized attention map will work as a gate, which allows the important feature to pass and suppress the irrelevant features. Similarly, we can obtain the cross-axis attention weights for Y-axis as follows:

$$A_i^{|y|} = \sigma(W_{yy}^{|y|} \circ d_i^y + b^{|y|}),$$

$$A_i^{|z|} = \sigma(W_{yz}^{|z|} \circ d_i^z + b^{|z|}),$$

(3)

and the cross-axis attention weights for Z-axis are

$$A_i^{|z|} = \sigma(W_{zx}^{|z|} \circ d_i^x + b^{|z|}),$$

$$A_i^{|y|} = \sigma(W_{yz}^{|y|} \circ d_i^y + b^{|y|}).$$

(4)

**Self Attention.** Besides the cross-axis attention, the model also needs to generate attention maps for each axis based on itself. With such a mechanism, the model can learn to focus on the useful features of one axis according to waveforms or patterns of itself. Hence, we again use the attention mechanism to control the self attention weights as follows:

$$A_i^{|x|} = \sigma(W_{xx}^{|x|} \circ d_i^x + b^{|x|}),$$

$$A_i^{|y|} = \sigma(W_{yy}^{|y|} \circ d_i^y + b^{|y|}),$$

$$A_i^{|z|} = \sigma(W_{zz}^{|z|} \circ d_i^z + b^{|z|}),$$

(5)

where $W \in \mathbb{R}^{M \times F}$ and $b \in \mathbb{R}^F$ are the trainable parameters of the convolutional operation. Eq. (5) shows that the model can learn to automatically adjust its self attention weights $A_i \in \mathbb{R}^{T \times F}$ and thereby control its own concentration proportion instead of completely depending on the cross-axis attention maps from other axes. In addition, with the self attention mechanism, the model becomes more robust to the different kinds of movements. Even if the accelerometer moves along one axis, which only changes the value of this axis, it can mainly focus on itself and ignore the cross-axis attention maps by self attention.

**Hybrid Fusion.** To simultaneously take into account itself and the other two axes, we further propose a hybrid fusion mechanism to combine the cross-axis attention and self attention maps. First, we calculate the features learned by the cross-axis attention mechanism as follows:

$$d_i^{|y|z} = d_i^x \cdot (A_i^{|y|} + A_i^{|z|}) / 2,$$

$$d_i^{|z|y} = d_i^y \cdot (A_i^{|z|} + A_i^{|y|}) / 2,$$

$$d_i^{|x|y} = d_i^z \cdot (A_i^{|x|} + A_i^{|y|}) / 2.$$  

(6)

We can observe that the cross-axis feature is calculated by multiplying the input deep features with the corresponding cross-axis attention maps. We then combine the deep feature and the cross-axis feature together via the proposed hybrid fusion mechanism to obtain the fused feature representation for each axis as follows:

$$d_i^{|xy} = A_i^{|x|} \cdot d_i^x + (1 - A_i^{|x|}) \cdot d_i^{|xy},$$

$$d_i^{|yz} = A_i^{|y|} \cdot d_i^y + (1 - A_i^{|y|}) \cdot d_i^{|yz},$$

$$d_i^{|zx} = A_i^{|z|} \cdot d_i^z + (1 - A_i^{|z|}) \cdot d_i^{|zx}.$$  

(7)

For each axis, it will be filtered by the proposed attention mechanism to accentuate the significant features and suppress the irrelevant features by considering all three axes synergistically.

3) Classification Module: The inputs to the classification modules $\text{clf}$ are the concatenation of the hybrid fused features, i.e., $[d_i^{|xy}; d_i^{|yz}; d_i^{|zx}] \in \mathbb{R}^{T \times 3F}$. This classification module consists of a global average pooling (GAP) and 3-layer fully-connected layers. All the fully-connected layers except the last one are using the ReLU activation function, while the last layer is operated on the softmax activation and its unit number must match the number of classes. The module will calculate the probabilities of each class. During training, its output will be used to compute categorical cross-entropy loss, which will be backpropagated to update the parameters of the entire AccSleepNet model. During testing, the class with the maximum probability will be considered as the final predicted class.
TABLE I: The overall performance of all approaches on Sleep-Accel and Newcastle-Accel datasets for 2-, 3-, 4-, and 5-class sleep stage classification via 5-fold cross-validation in terms of Accuracy and Weighted-F1 score. The standard deviations (i.e., std) of proposed methods are also provided. The results of the best baseline and best performer in each column are highlighted and boldfaced, respectively. * represents the significant difference with a p-value less than 0.005 compared with the best baselines.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Type</th>
<th>Method</th>
<th>5-class</th>
<th>4-class</th>
<th>3-class</th>
<th>2-class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sleep-Accel</td>
<td>Feature-Based Machine Learning</td>
<td>KNN</td>
<td>0.4743</td>
<td>0.3852</td>
<td>0.5144</td>
<td>0.4423</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RF</td>
<td>0.4734</td>
<td>0.3845</td>
<td>0.2907</td>
<td>0.2863</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MLP</td>
<td>0.2003</td>
<td>0.1377</td>
<td>0.2525</td>
<td>0.2232</td>
</tr>
<tr>
<td></td>
<td>Deep Learning</td>
<td>CNN</td>
<td>0.5039</td>
<td>0.4996</td>
<td>0.6008</td>
<td>0.5947</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LSTM-CNN</td>
<td>0.6567</td>
<td>0.6586</td>
<td>0.7105</td>
<td>0.7124</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ResNet</td>
<td>0.7081</td>
<td>0.7099</td>
<td>0.7883</td>
<td>0.7802</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SanD</td>
<td>0.7270</td>
<td>0.7243</td>
<td>0.7863</td>
<td>0.7846</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Multimodal-CNN</td>
<td>0.6763</td>
<td>0.6734</td>
<td>0.7603</td>
<td>0.7624</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CrossAttn</td>
<td>0.7437</td>
<td>0.7501</td>
<td>0.8018</td>
<td>0.8054</td>
</tr>
<tr>
<td></td>
<td>Proposed Method</td>
<td>AccSleepNet</td>
<td>0.7697*</td>
<td>0.7643*</td>
<td>0.8263*</td>
<td>0.8251*</td>
</tr>
<tr>
<td>Newcastle-Accel</td>
<td>Feature-Based Machine Learning</td>
<td>KNN</td>
<td>0.4133</td>
<td>0.3447</td>
<td>0.4437</td>
<td>0.3904</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RF</td>
<td>0.4632</td>
<td>0.4500</td>
<td>0.4910</td>
<td>0.4833</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MLP</td>
<td>0.2804</td>
<td>0.2875</td>
<td>0.3390</td>
<td>0.3465</td>
</tr>
<tr>
<td></td>
<td>Deep Learning</td>
<td>CNN</td>
<td>0.5214</td>
<td>0.5386</td>
<td>0.6031</td>
<td>0.6050</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LSTM-CNN</td>
<td>0.4884</td>
<td>0.5072</td>
<td>0.6101</td>
<td>0.6099</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ResNet</td>
<td>0.5243</td>
<td>0.5424</td>
<td>0.6247</td>
<td>0.6271</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SanD</td>
<td>0.5411</td>
<td>0.5594</td>
<td>0.6210</td>
<td>0.6220</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Multimodal-CNN</td>
<td>0.5389</td>
<td>0.5511</td>
<td>0.6277</td>
<td>0.6287</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DeepSense</td>
<td>0.5611</td>
<td>0.5663</td>
<td>0.6334</td>
<td>0.6349</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CrossAttn</td>
<td>0.5732</td>
<td>0.5840</td>
<td>0.6318</td>
<td>0.6329</td>
</tr>
<tr>
<td></td>
<td>Proposed Method</td>
<td>AccSleepNet</td>
<td>0.5994*</td>
<td>0.6122*</td>
<td>0.6571*</td>
<td>0.6578*</td>
</tr>
</tbody>
</table>

A. Datasets

We conduct experiments on two publicly available wrist-worn accelerometer datasets, Sleep-Accel [7] and Newcastle-Accel [29]. Sleep-Accel contains 2,028 epochs of Wake, 1,656 epochs of stage N1, 11,655 epochs of stage N2, 3,187 epochs of stage N3, and 4,951 epochs of REM. Newcastle-Accel contains 9,147 epochs of Wake, 1,461 epochs of stage N1, 10,682 epochs of stage N2, 5,552 epochs of stage N3, and 3,573 epochs of REM.

B. Baselines

We use the following traditional machine learning and deep learning approaches as baselines:

- **Traditional machine learning approaches:** [7] derives physical activity features from the accelerometer data and passes them to k-nearest neighbors (KNN), random forest (RF), multi-layer perceptron (MLP) separately for sleep stage classification.


The aforementioned methods consider the triaxial accelerometer data as one single input with three-dimensionalities. On the other hand, we can consider the data as three inputs with different dimensionalities. Hence, we also adopt several deep learning methods taking the multi-modal or multi-view data as their inputs for our comparison.

- **Multimodal-CNN** is a CNN-based model for learning features from multimodal data [12].

- **DeepSense** uses a 2D CNN to exploit the interactions of different input modalities and an RNN to learn temporal relationships [13].

- **CrossAttn** applies a query-key-value cross-attention mechanism to handle multi-modality inputs with learning the relationship between them [14]. We replace our hybrid fusion with its cross-attention.

Note that there are several approaches also considering multi-view or multimodal learning, but we do not list them as baselines (e.g., DeepMV [30], SalientSleepNet [25], MuVan [31]), since they require that each input is multidimensional. However, in our task, each axis data of the accelerometer only has one dimension.

C. Experiment Setup

We examine the proposed AccSleepNet model with the benchmark methods on two public datasets, Sleep-Accel and Newcastle-Accel, by 5-fold cross-validation for the sleep stage classification task. In addition, since the existing work without PSG simplified the sleep stage rule to obtain more acceptable results as discussed in Section II, we also examine our proposed method in terms of 2-class (Wake/Sleep), 3-class (Wake/NREM/REM), 4-class (Wake/Light sleep/Deep sleep/REM), and 5-class (Wake/N1/N2/N3/REM) sleep stage classification.

Authorized licensed use limited to: Penn State University. Downloaded on November 16, 2023 at 15:44:22 UTC from IEEE Xplore. Restrictions apply.
For multiple class classification tasks, accuracy and macro-F1 scores are commonly used as evaluation metrics. However, both of the sleep stage datasets used in the experiments are imbalanced. Thus, we choose to use the weighted-F1 score as the metric, which calculates the F1 score for each class and finds their average weighted by the number of samples for each class. Besides the overall performance, we further compute the per-class F1 score to show the performance for identifying each class. In the clinical experiments, the classification performance of each class is also valuable to discuss. For example, some minor sleep stages (e.g., N1 and N3) are more critical when diagnosing sleep disorders.

The proposed AccSleepNet model is trained on NVIDIA Tesla V100 GPU by the AMSGrad optimizer [32] for 1200 epochs with initial learning rate of 0.005. The parameters of the feature extractors are listed in Fig. 2 and are initialized using HeNormal initializer [33]. In the hybrid fusion, the convolutional operations adopt $M = 4$ as the filter size and $F = 128$ as the number of filters. For baselines, we use the recommended parameters listed in the original papers.

### D. Performance of Sleep Stage Classification

We compare the performance of AccSleepNet with state-of-the-art benchmark methods in terms of Accuracy and F1 scores, and the results of 5-fold cross validation are shown in Table I. We can observe that the performance of all the approaches on the Sleep-Accel dataset is much better than that on the Newcastle-Accel dataset. Since they use different devices to collect data, the data quality significantly depends on devices’ quality, which further affects the model performance. However, the proposed AccSleepNet achieves the best performance in most cases, which clearly demonstrates the effectiveness of AccSleepNet for the sleep stage classification task.

Besides the overall performance, the performance of each sleep stage are also evaluated, and the results are shown in Table II. The deep learning-based methods achieve the more decent performance of scoring the minority sleep stages, such as N1 and N3. Through these results, we can see that the N1 stage is hard to be recognized. It is reasonable as the definition of the N1 stage is a transition stage between wakefulness and N2 stages, which means that their waveforms are similar to some extent. Furthermore, [34] points out that N1 and REM stages are also very hard to differentiate by EEG as both of them contain relatively low voltage mixed (2-7 Hz) frequency EEG [35]. However, the proposed AccSleepNet using a wrist-worn accelerometer instead of EEG achieves a more competitive performance of differentiating N1 and REM stages.

We can also see that the overall performance of the deep learning-based methods is better than the feature-based machine learning methods. Although the work [7] designs some special physical activity features, the performance of this approach is still relatively weak, which indicates that the handcrafted features of accelerometer data cannot appropriately represent the characteristics from different sleep stages. Thus, it may require more expert knowledge to design specific features to capture the relationship between wrist movement/vibration and sleep stages. However, this knowledge is not included in the AASM rule and is not well studied by existing researches. In this context, the deep learning-based methods skip calculating hand-crafted features and automatically learning the appropriate features for the sleep stage classification successfully.

Besides, we notice that the SAnD [11] and CrossAttn [14] obtain better performance than other deep learning methods among baselines. These two methods both use attention mechanisms. Thus, we believe the attention mechanism helps the model to focus on the more discriminative part of the accelerometer data as most parts of the data are likely to be very similar during sleep. Moreover, our proposed AccSleepNet
Table III: Ablation study to validate the necessity of leveraging 3-axis data in the proposed method on the Sleep-Accel and Newcastle-Accel datasets for 2-, 3-, 4-, and 5-class sleep stage classification via 5-fold cross-validation.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>5-class (Wake/N1/N2/N3/REM)</th>
<th>4-class (Wake/Light/Deep/REM)</th>
<th>3-class (Wake/NREM/REM)</th>
<th>2-class (Wake/Sleep)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Acc</td>
<td>F1</td>
<td>Acc</td>
<td>F1</td>
</tr>
<tr>
<td>Sleep-Accel</td>
<td>AccSleepNet-Con</td>
<td>74.40</td>
<td>74.08</td>
<td>80.11</td>
<td>79.92</td>
</tr>
<tr>
<td></td>
<td>AccSleepNet-Avg</td>
<td>73.60</td>
<td>72.78</td>
<td>79.91</td>
<td>79.62</td>
</tr>
<tr>
<td></td>
<td>AccSleepNet-Mul</td>
<td>73.18</td>
<td>74.02</td>
<td>79.64</td>
<td>79.75</td>
</tr>
<tr>
<td></td>
<td>AccSleepNet</td>
<td>76.77</td>
<td>76.43</td>
<td>82.65</td>
<td>82.51</td>
</tr>
<tr>
<td>Newcastle-Accel</td>
<td>AccSleepNet-Con</td>
<td>54.16</td>
<td>55.47</td>
<td>62.63</td>
<td>62.72</td>
</tr>
<tr>
<td></td>
<td>AccSleepNet-Avg</td>
<td>55.16</td>
<td>56.63</td>
<td>63.67</td>
<td>63.65</td>
</tr>
<tr>
<td></td>
<td>AccSleepNet-Mul</td>
<td>53.49</td>
<td>55.11</td>
<td>61.46</td>
<td>61.50</td>
</tr>
<tr>
<td></td>
<td>AccSleepNet</td>
<td>59.94</td>
<td>61.22</td>
<td>65.71</td>
<td>65.78</td>
</tr>
</tbody>
</table>

Table IV: Comparison study to validate the effectiveness of the proposed axis-aware hybrid fusion module on the Sleep-Accel and Newcastle-Accel datasets for 2-, 3-, 4-, and 5-class sleep stage classification via 5-fold cross-validation.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>5-class (Wake/N1/N2/N3/REM)</th>
<th>4-class (Wake/Light/Deep/REM)</th>
<th>3-class (Wake/NREM/REM)</th>
<th>2-class (Wake/Sleep)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Acc</td>
<td>F1</td>
<td>Acc</td>
<td>F1</td>
</tr>
<tr>
<td>Sleep-Accel</td>
<td>AccSleepNet</td>
<td>0.6369</td>
<td>0.6401</td>
<td>0.7087</td>
<td>0.7112</td>
</tr>
<tr>
<td></td>
<td>AccSleepNet-Mul</td>
<td>0.6493</td>
<td>0.6537</td>
<td>0.7232</td>
<td>0.7270</td>
</tr>
<tr>
<td></td>
<td>AccSleepNet</td>
<td>0.6064</td>
<td>0.6098</td>
<td>0.6827</td>
<td>0.6857</td>
</tr>
<tr>
<td></td>
<td>AccSleepNet</td>
<td>0.7677</td>
<td>0.7643</td>
<td>0.8265</td>
<td>0.8251</td>
</tr>
<tr>
<td>Newcastle-Accel</td>
<td>AccSleepNet-Mul</td>
<td>0.3784</td>
<td>0.3989</td>
<td>0.4445</td>
<td>0.4500</td>
</tr>
<tr>
<td></td>
<td>AccSleepNet</td>
<td>0.3675</td>
<td>0.3872</td>
<td>0.4276</td>
<td>0.4314</td>
</tr>
<tr>
<td></td>
<td>AccSleepNet</td>
<td>0.3926</td>
<td>0.3519</td>
<td>0.4348</td>
<td>0.4450</td>
</tr>
<tr>
<td></td>
<td>AccSleepNet</td>
<td>0.5994</td>
<td>0.6122</td>
<td>0.6571</td>
<td>0.6578</td>
</tr>
</tbody>
</table>

E. Ablation Study

In the proposed AccSleepNet method, we use three axes' data simultaneously and propose a novel attention mechanism to fuse them together. In this ablation study, we first show the necessity of leveraging the data from all three axes instead of a single axis. Towards this goal, we simplify the proposed AccSleepNet model for taking only a single axis data as the input. Let AccSleepNet_\text{X}, AccSleepNet_\text{Y}, and AccSleepNet_\text{Z} denote the three models, respectively. They have the same network architecture, which consists of a CNN-based feature extractor to learn features from each axis data and a classifier \text{Cf} to directly make predictions using the output of the feature extractor. Note that we do not need to calculate the self attention for the single axis data.

Table III shows the results on the two datasets. We can observe that the performance of leveraging the data from all three axes, i.e., the proposed AccSleepNet model, is much better than that of considering single axis data, especially on the Newcastle-Accel dataset. Compared with the results listed in Table I, we can observe that the performance of single axis data is comparable to that of traditional machine learning-based baselines, but is worse than that of deep learning-based models. From these comparisons, we can conclude that it is essential and necessary to leveraging the data from all three axes together when using the accelerometer data for the sleep stage classification task.

F. Axis-Fusion Method Validation

The ablation study aims to demonstrate the necessity of considering three axes data together, and in this subsection, we conduct experiments to validate the effectiveness of the proposed hybrid fusion mechanism for fusing three axes' data. To this end, we compare the proposed AccSleepNet model with different fusion approaches, such as concatenation (AccSleepNet-Con), average (AccSleepNet-Avg), and multiplication (AccSleepNet-Mul). Note that these three approaches are totally different from the proposed AccSleepNet model, which uses a novel axis-aware hybrid fusion module to fuse the features from three separate axes, highlighting the significant features and suppressing the irrelevant features by learning the three axes collaboratively.

Table IV lists the comparison results. We can observe that the proposed model AccSleepNet fusion outperforms others under all the scenarios of sleep stage classification. With the proposed hybrid fusion mechanism, the discriminative features or patterns are highlighted and amplified. For other
fusion approaches, the information over each axis is extracted individually without sufficient emphasis.

V. CONCLUSION

Different from existing studies for classifying sleep stages by analyzing the polysonmography data, in this paper, we propose an axis-aware hybrid fusion-based deep model, named AccSleepNet, which purely utilizes wrist-worn accelerometer data. Since the accelerometer has been widely used in smartwatches and smartbands nowadays, a model using a wrist-worn accelerometer is more scalable than the other existing methods. Furthermore, in the proposed AccSleepNet model, we design a novel axis-aware hybrid fusion mechanism, which generates a self-attention map and a cross-axis attention map to accentuate more significant features and suppress irrelevant features synergistically. We conduct experiments on two public datasets to validate the effectiveness of the proposed AccSleepNet model. Experimental results show that AccSleepNet boosts the performance and outperforms baselines. Besides, ablation studies demonstrate that leveraging three axes together is essential and the designed axis-aware hybrid fusion is effective for the sleep stage classification task.

REFERENCES